

Chapter 4

Theory and Numerical Solution of Differential and Algebraic Riccati Equations

Peter Benner

Abstract Since Kalman’s seminal work on linear-quadratic control and estimation problems in the early 1960s, Riccati equations have been playing a central role in many computational methods for solving problems in systems and control theory, like controller design, Kalman filtering, model reduction, and many more. We will review some basic theoretical facts as well as computational methods to solve them, with a special emphasis on the many contributions Volker Mehrmann had regarding these subjects.

4.1 Introduction

The algebraic and differential Riccati equations (AREs/DREs) play a fundamental role in the solution of problems in systems and control theory. They have found widespread applications in applied mathematics and engineering, many of which can be found in the monographs [1, 36, 68, 86]. In this chapter, we focus on Riccati equations associated to control problems, as these have always inspired Volker Mehrmann’s work, and he has mainly focused on the resulting *symmetric* Riccati equations – symmetric in the sense that the associated Riccati operators map symmetric (Hermitian) matrices onto symmetric (Hermitian) matrices. Hence, also the solutions to the Riccati equations we will consider are expected to be symmetric (Hermitian). A class of nonsymmetric AREs that arises, e.g., in queuing theory, certain fluid flow problems, and transport theory (see, e.g., [36]) is of importance as well, but for conciseness, we will omit these AREs even though Volker has also contributed to this area [81].

In most of the literature on AREs and DREs, the motivation is taken from the classical linear-quadratic regulator (LQR) problem. This was the topic of Volker’s habilitation thesis [74], where, building upon earlier work by Bender and Laub [6, 7], he extended the LQR theory to so-called descriptor systems, i.e., systems with

P. Benner (✉)

Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, D-39106 Magdeburg, Germany

e-mail: benner@mpi-magdeburg.mpg.de

dynamics described by differential-algebraic equations (DAEs). Much of Volker's early work on AREs culminated in this thesis which later became the appraised book [76]. We therefore also start by formulating the LQR problems in continuous- and discrete time and how they relate to AREs. These optimal control problems can be formulated in various levels of generality. The setting we consider is:

minimize

$$\mathcal{J}(x^0, u) = \frac{1}{2} \int_0^{\infty} (y(t)^T Q y(t) + 2y(t)^T S u(t) + u(t)^T R u(t)) dt \quad (4.1)$$

subject to the linear time-invariant (LTI) system

$$E \dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x^0, \quad (4.2a)$$

$$y(t) = Cx(t), \quad (4.2b)$$

in the continuous-time case and

minimize

$$\mathcal{J}(x^0, u) = \frac{1}{2} \sum_{k=0}^{\infty} (y_k^T Q y_k + 2y_k^T S u_k + u_k^T R u_k) \quad (4.3)$$

subject to the discrete-time LTI system

$$E x_{k+1} = A x_k + B u_k, \quad x_0 = x^0, \quad (4.4a)$$

$$y_k = C x_k, \quad (4.4b)$$

in the discrete-time case. In both settings, $A, E \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $Q \in \mathbb{R}^{p \times p}$, $R \in \mathbb{R}^{m \times m}$, and $S \in \mathbb{R}^{p \times m}$. Furthermore, we assume Q and R to be symmetric. In both cases, the initial state $x^0 \in \mathbb{R}^n$ can be chosen freely if E is nonsingular and is constrained to a manifold in the descriptor case, see the Chap. 16 for more details on this. In the continuous-time case, u is considered as an element of an appropriate function space like k times continuously differentiable functions $C^k(0, \infty)$ or square-integrable functions $L_2(0, \infty)$. No further constraints are imposed in this setting. In discrete-time, u represents a sequence $(u_k)_{k=0}^{\infty}$ that should be (square-)summable in an appropriate sense. A formulation in complex arithmetic is possible, and most of the results considered here remain valid (cf. [68] for a detailed treatment of real and complex AREs), but as most practical applications are formulated using real data, we stick to this case here.

Under fairly general conditions, the LQR problems have solutions of the form $u(t) = \mathcal{F}_c(x(t))$ and $u_k = \mathcal{F}_d(x_k)$, respectively. As they appear in feedback form, that is, the current control input depends on the current state information, this

observation lays the basis for *modern (feedback) control* [54]. Possible sufficient conditions to obtain such feedback solutions are: E is nonsingular, Q and R are positive definite, S is “small” enough such that $\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}$ is positive semidefinite, and the matrix triples (E, A, B) and (E, A, C) are stabilizable and detectable, respectively. Here, *stabilizable* means that $\text{rank}[A - \lambda E, B] = n$ for all $\lambda \in \{z \in \mathbb{C} \mid \Re(z) \geq 0\}$ in the continuous-time case and for all $\lambda \in \{z \in \mathbb{C} \mid |z| \geq 1\}$ in the discrete-time case, while detectability of (E, A, C) is equivalent to stabilizability of (E^T, A^T, C^T) . Under these conditions, the LQR problems have unique solutions. These are given by the feedback laws

$$u(t) = -R^{-1}(B^T X_c E + S^T C)x(t) =: -F_c x(t), \quad t \geq 0, \quad (4.5)$$

in the continuous-time case and

$$u_k = -(R + B^T X_d B)^{-1}(B^T X_d A + S^T C)x_k =: -F_d x_k, \quad k = 0, 1, \dots, \quad (4.6)$$

in the discrete-time case. Now, the relation of the LQR problem to AREs becomes evident as X_c is the unique *stabilizing* solution to the (*generalized*) *continuous-time algebraic Riccati equation* (CARE)

$$0 = \mathcal{R}_c(X) := C^T Q C + A^T X E + E^T X A - (E^T X B + C^T S)R^{-1}(B^T X E + S^T C), \quad (4.7)$$

while X_d is the unique *stabilizing* solution of the (*generalized*) *discrete-time algebraic Riccati equation* (DARE)

$$0 = \mathcal{R}_d(X) := C^T Q C + A^T X A - E^T X E - (A^T X B + C^T S)(R + B^T X B)^{-1}(B^T X A + S^T C). \quad (4.8)$$

The solutions X_c and X_d are *stabilizing* in the sense that the feedback solutions generated by inserting (4.5) and (4.6) into (4.2) and (4.4), respectively, are *asymptotically stable*, i.e., $x(t), x_k \rightarrow 0$ for $t, k \rightarrow \infty$ and all initial values $x^0 \in \mathbb{R}^n$. Under the given assumptions, CAREs and DAREs have exactly one stabilizing solution, despite the fact that there may exist many other solutions [68]. These stabilizing solutions are symmetric and positive semidefinite, the latter property again uniquely identifies the solutions X_c, X_d in the respective solution sets.

In his work, Volker has often considered the case that E is singular. This is in particular the case in his habilitation thesis and the resulting book [76]. In general, in this case the relation between the AREs (4.7) and (4.8) and the feedback solutions to the LQR problems is lost. Several modifications of the AREs to re-establish this connection have been suggested in the literature. They usually require special conditions, and the resulting generalized AREs are often not easily solvable numerically. Only recently, efficient methods for a class of generalized

CAREs with singular E have been suggested in [35]. Similarly to [6, 7], Volker shows in [76, § 5] how the LQR problem for descriptor systems (i.e., E singular) can be reduced to a problem with nonsingular E for which the Riccati solutions exist. Nevertheless, this procedure requires strong conditions, in particular quite complicated consistency conditions for the initial values, that are often not satisfied in practice. This observation, among others, led Volker to work on alternative approaches to solve the LQR problem for descriptor systems, avoiding AREs completely. This is the topic of [63, 64, 66, 67], where also extensions to systems with time-varying coefficients are considered and the theory of *strangeness-free* DAEs [65] is applied to the LQR problem. We will not further discuss these fundamental contributions to optimal control for DAEs, as they do not directly relate to Riccati equations anymore. Recently, a solution concept for LQR problems based on purely algebraic considerations was derived in the Ph.D. thesis of Matthias Voigt [88]. This theory relates the feedback solution of the LQR problem for descriptor systems to dissipation inequalities, even matrix pencils (see the Chaps. 6 and 12), and the stabilizing solutions of Lur'e equations. This work may complete three decades of the quest for a general algebraic solution theory for the LQR problem that does not require a special index concept or restrictions on the index of the underlying DAE. Notably, Matthias is Volker's academic grandchild!

In [76], Volker also considers LQR problems with finite time-horizon, i.e., the cost functionals are replaced by finite integrals \int_0^T , where $0 < T < \infty$, or finite sums $\sum_{k=0}^{K-1}$, $K \in \mathbb{N}$. In this situation, usually also the final states $x(T)$, x_K are penalized in the cost functional. Often, the LQR problem is used for stabilization problems so that it is desired that the final state is close to zero, which suggests a positive definite quadratic weighting function. This is particularly the case when the modeling is done in such a way that x represents the deviation of the state of a dynamical system from a desired state. In the continuous-time case, the LQR problem can then be formulated as follows:

minimize

$$\mathcal{J}(x^0, u) = \frac{1}{2} \left(x(T)^T M x(T) + \int_0^T (y(t)^T Q y(t) + 2y(t)^T S u(t) + u(t)^T R u(t)) dt \right)$$

subject to (4.2)

with $M = M^T \in \mathbb{R}^{n \times n}$ positive semidefinite. Note that for consistency, one could write the penalty term at T in terms of $y(T)$ instead of $x(T)$. As we can replace $y(T)$ by $Cx(T)$ using (4.2), this is merely a notational issue.

Under similar conditions as in the infinite-time horizon case, the unique solution of the finite-time LQR problem is then given by the feedback control

$$u(t) = -R^{-1}(B^T X(t)E + S^T C)x(t) =: -F_c(t)x(t), \quad t \geq 0, \quad (4.9)$$

where $X(t)$ is the unique solution of the DRE

$$-E^T \dot{X}(t)E = \mathcal{R}_c(X(t)) \quad (4.10)$$

with terminal condition $E^T X(T)E = M$ and \mathcal{R}_c as in (4.7). The existence and uniqueness of the DRE solution is considered in [1, 86]. The theory can also be extended to time-varying systems, i.e., systems with $A = A(t)$, etc. In general, if E is singular, again the relation of the solution to the finite-time LQR problem and the DRE is lost, but is re-established using a reduction procedure described by Volker in [76, § 3]. We will re-visit this technique, as well as a numerical procedure to solve (4.10) as suggested by Mehrmann and Kunkel in [62], in the next section.

The finite horizon discrete-time LQR problem is also treated in [76]. This leads to the solution of *difference Riccati equations* of the form

$$-E^T X_k E = \mathcal{R}_d(X_{k+1}) \quad (4.11)$$

with terminal condition $E^T X_K E = M$. As in the other cases considered so far, a reduction to the case E nonsingular is necessary to establish the link between the LQR problem and (4.11). Such a reduction procedure is suggested again in [76, § 3]. A numerical procedure to solve (4.11) is sketched in [76, § 19]. As (4.11) is solved backwards starting from X_K , advancing from X_{k+1} to X_k can be achieved by simply evaluating $\mathcal{R}_d(X_{k+1})$ using, e.g., a Cholesky decomposition of $R + B^T X_{k+1} B$, and then solving a linear system of equations using, e.g., the LU decomposition of E . Due to its conceptual simplicity, we will not discuss this approach here any further.

The remainder of this chapter is organized as follows. As already mentioned above, we will focus on Volker's contributions to the solution of DREs in the next section. The relation between CAREs and DAREs is explored in Sect. 4.3. The bulk of this chapter summarizes Volker's contributions to the numerical solution of AREs and comprises Sect. 4.4, while we briefly touch upon Volker's passion to solve control problems avoiding AREs in Sect. 4.5. Final remarks are given in Sect. 4.6.

4.2 Numerical Methods for Differential Riccati Equations

In [62], the DRE (4.10) is considered with $S = 0$, and in the simplified representation

$$-E^T \dot{X}(t)E = F + A^T X(t)E + E^T X(t)A - E^T X(t)GX(t)E \quad (4.12)$$

with terminal condition $E^T X(T)E = M$, where even time-dependence of A, E, F, G is allowed under the assumption of sufficient smoothness.

The contributions of [62], and [76, § 3,19] are twofold. First of all, conditions are derived under which the solution of the finite-time horizon LQR problem is given via the feedback control law (4.9), defined by the solution of the DRE. In case of singular E , this requires a reduction procedure resulting in a DRE with

nonsingular E matrix. We will sketch this procedure briefly. Second, a numerical solution method for (4.12) is derived that resolves some singularity issues when (4.12) is solved in vectorized form as a DAE. In the notation derived later on by Kunkel and Mehrmann [65], it is assumed that the DAE $E\dot{x}(t) = Ax(t)$ is strangeness-free (see the Chap. 16 for details), (E, A) is a regular matrix pair, and $\text{rank}(E)$ is constant on $[0, T]$. For the ease of presentation, in the following we will only consider the time-invariant case, but the derivations in the time-varying case in [62] are formally completely analogous.

4.2.1 Reduction to a Standard LQR Problem

Basically, the singular value decomposition of E ,

$$E = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^T \quad (4.13)$$

is the key step in the reduction procedure. Here, $U, V \in \mathbb{R}^{n \times n}$ are orthogonal, $\Sigma \in \mathbb{R}^{r \times r}$ is nonsingular, and $r = \text{rank}(E)$. Note that we do not necessarily need a reduction of Σ to diagonal form, a URV-type decomposition with a nonsingular Σ suffices. Therefore, we do not assume symmetry of Σ in the following. If we now insert the SVD of E in (4.2), multiply the first equation with U^T from the left, and using

$$U^T A V = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad U^T B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad C V = [C_1, C_2],$$

as well as the change of basis $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = V^T x$ with the partitioning implied by the SVD (4.13), the descriptor system (4.2) becomes

$$\Sigma \dot{x}_1(t) = A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t), \quad (4.14a)$$

$$0 = A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t), \quad (4.14b)$$

$$y(t) = C_1x_1(t) + C_2x_2(t). \quad (4.14c)$$

The strangeness-freeness assumption implies that A_{22} is nonsingular, so that we can solve the second of these equations for x_2 and insert the result in the first and third equations. This leads to the standard LTI system

$$\Sigma \dot{x}_1(t) = \underbrace{(A_{11} - A_{12}A_{22}^{-1}A_{21})}_{=: \hat{A}} x_1(t) + \underbrace{(B_1 - A_{12}A_{22}^{-1}B_2)}_{=: \hat{B}} u(t), \quad (4.15)$$

$$y(t) = \underbrace{(C_1 - C_2A_{22}^{-1}A_{21})}_{=: \hat{C}} x_1(t) + \underbrace{(-C_2A_{22}^{-1}B_2)}_{=: \hat{D}} u(t). \quad (4.16)$$

Now also performing the change of basis in the cost functional results in a standard finite-time horizon LQR problem which can be solved via the DRE

$$-\Sigma^T \hat{X}(t) \Sigma = \hat{Q} + \Sigma^T \hat{X} \hat{A} + \hat{A}^T \hat{X} \Sigma - \left(\hat{B}^T \hat{X} \Sigma + \hat{S}^T \right)^T \hat{R}^{-1} \left(\hat{B}^T \hat{X} \Sigma + \hat{S}^T \right) \quad (4.17)$$

with terminal condition $\Sigma^T \hat{X} \Sigma = \hat{M}$, where \hat{M} results from the terminal condition of the original LQR problem after the change of coordinates. Note that when $B_2 \neq 0$, $u(T)$ appears in the terminal penalty term which changes the nature of the problem. As M can usually be chosen freely, it is therefore convenient to assume in the partitioning of $V^T M V = \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{bmatrix}$ that $M_{12} = 0$, $M_{22} = 0$. The coefficient matrices \hat{Q} , \hat{S} , \hat{R} in (4.17) can be read off from the cost functional after applying the change of coordinates and inserting the modified output equation (4.16) in the first integrand:

$$\begin{aligned} y(t)^T Q y(t) &= (\hat{C} x_1(t) + \hat{D} u(t))^T Q (\hat{C} x_1(t) + \hat{D} u(t)) \\ &= x_1(t)^T \underbrace{\hat{C}^T Q \hat{C}}_{=: \hat{Q}} x_1(t) + 2x_1(t)^T \underbrace{\hat{C}^T Q \hat{D}}_{=: \hat{S}} u(t) + u(t)^T \hat{D}^T Q \hat{D} u(t). \end{aligned}$$

With $\hat{R} = R + \hat{D}^T Q \hat{D}$, and the assumption that $S = 0$, the LQR theory for LTI systems yields the feedback solution via the DRE (4.17). In order for this reduction procedure to work, it is of course necessary that the modified matrices defining the LTI system and cost functional inherit the properties like positive (semi)definiteness of \hat{Q} , \hat{R} as well as the stabilizability and detectability properties. In a numerical procedure, this needs to be checked before the solution of the original LQR problem can be derived from that of the reduced LQR problem. While often, these properties indeed carry over, a more severe restriction is caused by the consistency condition implied by (4.14b) for $t = 0$: if $\begin{bmatrix} x_1^0 \\ x_2^0 \end{bmatrix} = V^T x^0$, then (4.14b) implies

$$x_2^0 = -A_{22}^{-1} (A_{21} x_1^0 + B_2 u(0)).$$

If $B_2 \neq 0$, this is a restriction on the possible controls, while for $B_2 = 0$, it yields a consistency condition for the initial values. Whether or not this restricts the applicability of the reduction approach to the LQR problem for descriptor systems certainly depends on the application. It should also be noted that under certain conditions on the system matrices, higher-index DAE problems can be reduced to regular index-1 problems using output feedback. This topic is discussed in more detail in Chap. 15.

Remark 1 The reduction procedure using the SVD (4.13) of E can also be applied directly to the DRE (4.12) without considering the LQR background. This basically leads again to the DRE (4.17) with additional consistency and solvability conditions, see [62, Section 3].

4.2.2 Numerical Solution of DREs

In [62] and [76, § 19], it is suggested to solve the DRE (4.12) by *vectorization*. This uses the “vec” operator

$$\text{vec} : \mathbb{R}^{\ell \times k} \rightarrow \mathbb{R}^{\ell k} : X \mapsto \text{vec}(X),$$

which stacks the columns of a matrix on top of each other. Vectorization of expressions like AXB is simplified by using the Kronecker product \otimes , see, e.g., [61, 69]. In particular, the formulas

$$\text{vec}(AXB) = (B^T \otimes A) \text{vec}(X), \quad (A^T \otimes B^T) = (A \otimes B)^T \quad (4.18)$$

are useful for our purposes. Applying vectorization to (4.12) and using (4.18) yields the system of ordinary differential equation (ODEs) or DAEs

$$\begin{aligned} -(E \otimes E)^T \text{vec}(\dot{X}(t)) &= \text{vec}(F) + \left((E \otimes A)^T + (A \otimes E)^T \right) \text{vec}(X(t)) \\ &\quad - (E \otimes E)^T (X(t) \otimes X(t)) \text{vec}(G). \end{aligned} \quad (4.19)$$

The terminal condition yields

$$(E \otimes E)^T \text{vec}(X(T)) = \text{vec}(M). \quad (4.20)$$

Together, (4.19) and (4.20) form an initial value problem (in reverse time) for a system of ODEs/DAEs if E is nonsingular/singular with quadratic nonlinearity. (In case $n = 1$, a classical Riccati differential equation is obtained, thus the name “differential Riccati equation”.)

In case of E being nonsingular, this initial value problem can be solved by any integration method for ODEs. As (4.19) is then a system of n^2 ODEs, already one time step of an explicit integrator would require $\mathcal{O}(n^4)$ floating point operations (flops) in general for the matrix vector products. In an implicit integration technique, linear systems of equations need to be solved at a computational cost of $\mathcal{O}(n^6)$ flops. Therefore, this approach is limited to very small dimensions n , even if one exploits the symmetry of the DRE as suggested in [62] and only works with the $n(n+1)/2$ necessary equations, ignoring the redundant other ones (which is possible, but the data handling is cumbersome). After this initial work on numerical methods for DREs by Kunkel/Mehrmann, it was suggested in [48] to re-write ODE integrators in matrix form. This reduces the cost to $\mathcal{O}(n^3)$ flops in dense arithmetic. Due to the usual inherent stiffness of DREs, in the literature, mostly implicit integrators are discussed. In [48], in particular the backward differentiation formulas (BDF) were considered. This was followed up in [53], where an efficient implementation of the matrix-valued BDF for DREs was described. Later, this was extended to large-scale

problems employing sparsity in the coefficient matrices in [29]. Also other popular implicit integration techniques were investigated, e.g., the Rosenbrock methods in matrix form in [30, 70] by members of Volker's academic descendants family.

The case that E is singular is much more involved, though. Let $\tilde{E} := (E \otimes E)^T$ and $\tilde{A} := (E \otimes A)^T + (A \otimes E)^T$. Then it is shown in [62] that even in the simplest form of a strangeness-free DAE $E\dot{x} = Ax$, the matrix pencil $\lambda\tilde{E} - \tilde{A}$ is singular. If the nilpotency index of (E, A) is 1, then so is the nilpotency index of $\lambda\tilde{E} - \tilde{A}$, and the DRE (4.19) can be solved by simply omitting the singular "zero blocks". A procedure to achieve this is suggested in [62, Section 5]. Apart from the SVD (4.13) and the associated reduction, it includes consistency checks and requires additional algebraic equations to be solved. The reduced DRE is then solved by standard DAE integrators for index-1 systems. Also, a variant for time-dependent coefficients $A(t), E(t), \dots$ is provided in [62]. If the nilpotency index of the regular matrix pencil $\lambda E - A$ is larger than 1, then in addition to singularity of $\lambda\tilde{E} - \tilde{A}$, this matrix pencil may have even a larger nilpotency index, and also different kinds of singular blocks. Therefore, a numerical procedure for this situation cannot be derived easily, and this is an open problem up to now to the best of this author's knowledge. An interesting question for further research would be whether the singularity problem can be avoided in matrix-valued DRE solvers. Recently, a first step in this direction was proposed by another of Volker's many Ph.D. students, Jan Heiland. In his thesis [58], he suggests a matrix-valued implicit Euler scheme for a special index-2 DRE, which he calls a differential-algebraic Riccati equation and which arises in the finite-time horizon LQR problem for flow control problems. Further research in the area of matrix-valued solvers for DREs or differential-algebraic Riccati equations is certainly needed.

4.3 A Unified Treatment of CAREs and DAREs?

A unified treatment of discrete- and continuous-time algebraic Riccati equations is discussed in [77]. We will recall the main result from this paper related to AREs: a then new result on the solution of DAREs using the idea of a unified treatment of continuous- and discrete-time control problems. As this result was derived in complex arithmetic, we will (in contrast to the other sections) in this chapter also formulate the results in complex arithmetic. For this, we denote by M^H the complex conjugate transpose of a matrix or vector M , while $\bar{\mu}$ denotes as usual the complex conjugate of the scalar $\mu \in \mathbb{C}$.

In addition to the Hamiltonian and symplectic matrices introduced already in Chap. 1, we will need the following structures in this section.

Definition 1 Let $J = \begin{bmatrix} 0_n & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$ with $I_n, 0_n$ the identity and zero matrices of order n .

- (a) A matrix pencil $\lambda K - H$ is *Hamiltonian* iff $KJH^H + HJK^H = 0$.
- (b) A matrix pencil $\lambda T - S$ is *symplectic* iff $TJT^H = SJS^H$.

In this and the next section, a special class of n -dimensional subspaces in \mathbb{C}^{2n} plays a prominent role.

Definition 2 Let $\mathcal{L} \subset \mathbb{C}^{2n}$ and $\dim(\mathcal{L}) = n$. Then \mathcal{L} is *Lagrangian* if $x^H J y = 0$ for all $x, y \in \mathcal{L}$.

A typical example of a Lagrangian subspace is the graph subspace $\begin{bmatrix} I_n \\ M \end{bmatrix}$ for a Hermitian matrix M , see the Chap. 5 for more on the use of graph subspaces.

If we now consider the continuous- and discrete-time LQR problems from Sect. 4.1 with $S = 0$ for simplicity, and the associated AREs, then we have the following well-known observation: let $E = I_n$, and define $F := C^H Q C$, $G = BR^{-1}B^H$, and

$$H = \begin{bmatrix} A & G \\ F & -A^H \end{bmatrix}, \quad \lambda T - S := \lambda \begin{bmatrix} I_n & -G \\ 0_n & A^H \end{bmatrix} - \begin{bmatrix} A & 0_n \\ F & I_n \end{bmatrix}. \quad (4.21)$$

Then H is Hamiltonian, $\lambda T - S$ is symplectic. We then have the following result, which is a collection of results that can be found, e.g., in [68]:

Proposition 1 Let $E = I_n, S = 0$, and H, S, T as in (4.21).

- (a) Assume that $X_c = X_c^H$ is a solution to the CARE (4.7), then the columns of $\begin{bmatrix} I_n \\ -X_c \end{bmatrix}$ span a Lagrangian H -invariant subspace. On the other hand, if the columns of $\begin{bmatrix} U \\ V \end{bmatrix}$ span a Lagrangian H -invariant subspace with $U \in \mathbb{C}^{n \times n}$ invertible, then $X_c = -VU^{-1}$ is a Hermitian solution to the CARE (4.7).
- (b) Let A be nonsingular. Assume that $X_d = X_d^H$ is a solution to the DARE (4.8), then the columns of $\begin{bmatrix} I_n \\ -X_d \end{bmatrix}$ span a Lagrangian deflating subspace of $\lambda T - S$. On the other hand, if the columns of $\begin{bmatrix} U \\ V \end{bmatrix}$ span a Lagrangian deflating subspace of $\lambda T - S$ with $U \in \mathbb{C}^{n \times n}$ invertible, then $X_d = -VU^{-1}$ is a Hermitian solution to the DARE (4.8).

Note that the CARE (DARE) solutions requested in the LQR problems are associated to the stable invariant (deflating) subspaces of H ($\lambda T - S$), that is, those associated to the n eigenvalues in the open left half of the complex plane (inside the open unit disk).¹

This result can be used in order to link the CARE and DARE in the following way, employing the generalized Cayley transformation: given $\mu \in \mathbb{C}$ with $|\mu| = 1$,

¹This spectral dichotomy of the Hamiltonian matrix and symplectic pencil exists under the assumptions used in Sect. 4.1, i.e., there are no eigenvalues on the boundaries of these regions, that is the imaginary axis in the continuous-time and the unit circle in the discrete-time case.

$\mu^2 \neq -1$. Then the *generalized Cayley transformation* of a matrix pencil $\lambda L - M$ is given by the matrix pair

$$\mathcal{C}_\mu(L, M) := ((L - \mu M), (\mu L + M)). \quad (4.22)$$

With this, Volker proved the following result:

Lemma 1 ([77, Lemma 2])

- (a) If $\lambda T - S$ is a symplectic pencil, then $\mathcal{C}_\mu(T, S) = (K, H)$ defines a Hamiltonian pencil $\lambda K - H$.
- (b) If $\lambda K - H$ is a Hamiltonian pencil, then $\mathcal{C}_{-\mu}(K, H) = (T, S)$ defines a symplectic pencil $\lambda T - S$.

As the Hamiltonian matrix H from (4.21) also defines a Hamiltonian pencil $\lambda I - H$, this lemma relates the Hamiltonian matrix and associated CARE to a symplectic pencil via Lemma 1(b), to which, using some cumbersome algebraic manipulations and some further assumptions on the Cayley shift μ , a DARE can be associated. Vice versa, the symplectic pencil associated to the DARE (4.8) can be related to a Hamiltonian pencil via Lemma 1(a). As it is easy to see that the Cayley transformation leaves deflating subspaces invariant, we thus can use numerical methods for computing deflating subspaces of Hamiltonian pencils to solve DAREs via Cayley transformation and Proposition 1. An algorithm for the Hamiltonian pencil eigenproblem with the desirable properties of being numerically backward stable, requiring only $\mathcal{O}(n^3)$ flops, and being structure-preserving in the sense that the symmetry of the spectrum is preserved exactly, was then derived later in [28]. This algorithm extends the method for Hamiltonian matrices based on the symplectic URV decomposition presented in the same paper and discussed in detail in Sect. 1.4. Unfortunately, this algorithm only computes eigenvalues, but no deflating subspaces. Already the extension of the method for Hamiltonian matrices to compute also invariant subspaces turned out to be rather complicated [27], so an extension for the Hamiltonian pencil algorithm was never derived. But it turned out a bit later that the better structure to consider was that of skew-Hamiltonian/Hamiltonian pencils (see the Chap. 2), or even more general, that of even/odd pencils, or palindromic pencils (see the Chap. 3). Numerically stable and efficient algorithms respecting these structures have been derived by Volker and co-workers in the last decades, see various chapters in this book and the recent overview [24].

The main contribution of [77] was to extend the use of the generalized Cayley transformation to more general situations than those discussed in Proposition 1 and Lemma 1. In particular, $E \neq I_n$ is allowed, and the quite restrictive assumption of A being nonsingular in the discrete-time case is dropped. In the situations where $E \neq I_n$, the matrix pencils associated to the CARE and DARE are no longer Hamiltonian and symplectic, respectively. Nevertheless, the same spectral symmetries are observed (which is evident when one considers, as done later, the associated even and palindromic pencil structures). A number of interesting

relations of the Cayley transformed pencils are derived in [77], and it is shown that controllability properties of the linear-time invariant systems associated to the Hamiltonian and symplectic matrix pencils before and after Cayley transformation remain invariant. This lead Volker to suggest an *implication scheme* which states that if one can prove “A \Rightarrow B” for a continuous-time system, one can prove the analogous result for the discrete-time system obtained via Cayley transformation (and vice versa). This eventually lead to the proof of the following result for the existence of DARE solutions which generalizes Proposition 1(b) and is proved using this implication scheme applied to a variant of Proposition 1(a).

Theorem 1 ([77, Theorem 14(b)]) *Consider the symplectic matrix pencil*

$$\lambda T - S := \lambda \begin{bmatrix} A & 0_n \\ F & I_n \end{bmatrix} - \begin{bmatrix} I_n & -BR^{-1}B^H \\ 0_n & A^H \end{bmatrix}.$$

Let the columns of $\begin{bmatrix} U \\ V \end{bmatrix}$ span an n -dimensional deflating subspace of $\lambda T - S$ with $U, V \in \mathbb{C}^{n \times n}$, $V^H U = U^H V$, and not containing eigenvectors corresponding to infinite eigenvalues. Suppose there exists $\mu \in \mathbb{C}$, $|\mu| = 1$, such that $I_n - \mu A$ and $\mu I + A - BR^{-1}B^H(\mu I_n - A^H)^{-1}F$ are invertible, as well as

$$\Psi(\mu) := R + B^H(A - \mu I_n)^{-H}F(A - \mu I_n)^{-1}B$$

is definite. Assume further that (A, B) is controllable (i.e., $\text{rank}([A - \lambda I_n, B]) = n$ for all $\lambda \in \mathbb{C}$).

Then U is invertible and $X_d := -VU^{-1}$ solves the DARE

$$0 = F + A^H X A - X + A^H X B (R + B^H X B)^{-1} B^H X A.$$

This result does not require A to be invertible. The nonsingularity assumptions in the theorem are needed to apply the generalized Cayley transformation with shift μ to $\lambda T - S$ and to convert the resulting Hamiltonian pencil to a Hamiltonian matrix, for which a variant of Proposition 1(a) can then be used to prove the assertions in the continuous-time case. The proof then follows from the implication scheme.

Highlighting this result gives a glimpse on the often observed interdisciplinary work of Volker Mehrmann, here linking systems and control theory with matrix analysis. It also demonstrates his keen interest in deriving fundamental and general principals that allow to solve classes of problems rather than just specific instances of a given problem.

4.4 Numerical Methods for AREs

We note that this section is based in large parts on [10] which concentrated on the CARE and continued the effort of providing surveys of ARE solvers given as in [4] and by Volker with co-workers [40]. We use an abridged and updated version of the survey given there, with pointers to corresponding methods for DAREs which appear throughout Volker's work. In this section, we restrict ourselves again to AREs in real arithmetic. Also, we will now always assume that E is nonsingular such that there is a clear relation of the LQR problem and the related ARE as outlined in the Introduction. For the ease of presentation, we will use the following simplified ARE versions. We consider CAREs of the form

$$0 = \mathcal{R}_c(X) := F + A^T X + XA - XGX, \quad (4.23)$$

and DAREs

$$0 = \mathcal{R}_d(X) = F + A^T XA - X - (A^T XB)(R + B^T XB)^{-1}(A^T XB)^T, \quad (4.24)$$

where in the LQR setting, $F := C^T QC$ and $G := BR^{-1}B^T$. The solution of (4.23) yielding the optimal feedback control then is the unique *stabilizing* solution X_c , i.e., $A - GX_c$ is stable in the sense that all its eigenvalues are in the open left half plane \mathbb{C}^- . Similarly, in the discrete-time case, the stabilizing solution X_d yields a stable *closed-loop matrix* $A - B(R + B^T X_d B)^{-1} B^T X_d A$ in the sense that all its eigenvalues are inside the open unit disk.

It should be noted that in the non-descriptor, non-singular case, i.e., when E and R are invertible, it is always possible to rewrite CAREs and DAREs in this simplified way. In practice, though, this should only be done in case all transformations are well-conditioned. See, e.g., [4, 9, 31, 76] for algorithms working in the more general formulations and avoiding inversions and matrix products in forming the coefficients as far as possible and necessary.

4.4.1 Methods Based on the Hamiltonian and Symplectic Eigenproblems

As discussed in the previous section, solving AREs can be achieved using methods to compute certain invariant or deflating subspaces of Hamiltonian matrices or symplectic pencils. If such a subspace $\begin{bmatrix} U \\ V \end{bmatrix}$ is Lagrangian with $U \in \mathbb{R}^{n \times n}$ invertible, the formula

$$X = -VU^{-1} \quad (4.25)$$

yields a symmetric solution to the CARE or DARE, respectively. Of course, for numerical stability the condition number of U should be small, which can be achieved by certain scaling procedures (see [9] for a comparison of several scaling strategies) or, more recently, using the principle of permuted graph matrices introduced by Mehrmann and Poloni, see the Chap. 5 for details of this approach.

The required solutions for solving the LQR problems are obtained if $\begin{bmatrix} U \\ V \end{bmatrix}$ corresponds to the stable eigenvalues of the corresponding Hamiltonian matrix or symplectic pencil as explained in the previous section. Due to spectral symmetries of Hamiltonian matrices H explained in Chap. 1, i.e., with λ also $-\bar{\lambda}$ is an eigenvalue of H , and the analogous property of symplectic pencils (with λ also $1/\bar{\lambda}$ is an eigenvalue), a sufficient condition for the existence of n -dimensional stable invariant subspaces is spectral *dichotomy*, i.e., no eigenvalues lie on the boundary of the respective stability region. This dichotomy property is usually satisfied under assumptions that lead to the unique solvability of LQR problems via AREs and is thus assumed throughout this section.

First, we will consider methods for CAREs based on solving the Hamiltonian eigenproblem for H as in (4.21). Using Proposition 1(a), in order to solve the CARE it is sufficient to find a nonsingular matrix $T \in \mathbb{R}^{2n \times 2n}$ such that

$$T^{-1}HT = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}, \quad (4.26)$$

where the eigenvalues of $H_{11} \in \mathbb{R}^{n \times n}$ are all in \mathbb{C}^- . Partitioning T analogously, the columns of $\begin{bmatrix} T_{11} \\ T_{12} \end{bmatrix}$ span the stable Lagrangian H -invariant subspace, and the desired stabilizing solution X_c of the CARE (4.23) is obtained via (4.25) setting $U = T_{11}$ and $V = T_{12}$.

Most algorithms for solving matrix eigenproblems, i.e., for computing eigenvalues and -vectors or invariant subspaces of some matrix $M \in \mathbb{R}^{\ell \times \ell}$, are based on the following approach:

Generic procedure to compute invariant subspaces:

1. Compute an initial transformation matrix $S_0 \in \mathbb{R}^{\ell \times \ell}$ in order to reduce M to some condensed form, i.e., compute

$$M_0 := S_0^{-1}MS_0. \quad (4.27)$$

2. Then construct a sequence of similarity transformations such that in each step

$$M_{j+1} := S_{j+1}^{-1}M_jS_{j+1}, \quad j = 0, 1, 2, \dots, \quad (4.28)$$

the reduced form is preserved and moreover, if we define $T_j := \prod_{k=0}^j S_k$, then $\lim_{j \rightarrow \infty} T_j = T$ and $\lim_{j \rightarrow \infty} M_j = M_*$ exist and eigenvalues and eigenvectors and/or M -invariant subspaces can be read off from M_* and T .

The purpose of the initial reduction to a condensed form and the preservation of this form throughout the iteration is twofold: first, such a reduction is usually necessary in order to satisfy the complexity requirements – an iteration step (4.28) on a reduced form can usually be implemented much cheaper than for a full matrix; second, using such a reduced form it is usually easier to track the progress of the iteration and detect if the problem can be decoupled (*deflation*) into smaller subproblems that can then be treated separately. For details see [56, Chapters 7–8].

For numerical stability, one usually requires all S_j and thus T_j, T to be orthogonal. Under this requirement, the real Schur decomposition of H is a reasonable choice in (4.26), and this was suggested in the seminal paper by Laub [71]. A disadvantage is that the structure of the Hamiltonian matrix is already destroyed in the initial Hessenberg reduction, leading to the loss of spectral symmetry in finite precision arithmetic. In the worst case, this may lead to violation of the spectral dichotomy as perturbations may move the eigenvalue to or across the boundary of the stability region. This can be avoided by using symplectic similarity transformations as then all iterates in the generic procedure above remain Hamiltonian and thus, the spectral symmetry is preserved. As already seen in Chap. 1, implementing the above procedure satisfying both demands, symplectic *and* orthogonal similarity transformations is only possible under certain assumptions due to the lack of an efficient procedure for computing the Hamiltonian Schur form, also called *Van Loan's curse*.

We will now briefly discuss the numerical methods for the Hamiltonian and symplectic eigenproblems that can be used to solve CAREs and DAREs. As most of them have already been discussed in Chap. 1, we keep this discussion short and highlight only Volker's contributions in more or less chronological order.

4.4.1.1 The SR Algorithm

In [41], Buse-Gerstner and Mehrmann suggest to use the SR algorithm to implement the generic procedure to compute the stable H -invariant subspace and then to solve the CARE via (4.25). The SR algorithm is described in some detail in Sect. 1.2. Its obvious advantage is the preservation of the Hamiltonian structure due to the exclusive use of symplectic similarity transformations. Another advantage is that it is fast: it requires only $\mathcal{O}(n)$ flops per iteration and generically converges with a cubic rate. As non-orthogonal transformations are required, numerical stability is lost and the computed CARE solution may not be as accurate as desired. A possible remedy for this problem is *defect correction* (see Sect. 4.4.2), which can be used to improve the accuracy of an approximate CARE solution.

Given that the method is not numerically backward stable, some variations have been suggested that compute the approximation to the CARE solution even faster by giving up orthogonal transformations altogether, see [42]. Another idea is to iterate directly on the approximation of the CARE rather than computing the Lagrangian invariant subspaces explicitly. For this, an updating procedure is suggested in [47].

Moreover, an SR algorithm for symplectic eigenvalue problems was also derived by Volker and co-workers, see [55] for a first variant and Chap. 1 for further developments of this method.

4.4.1.2 The Hamiltonian QR Algorithm

The ideal algorithm for the Hamiltonian and symplectic eigenvalue problems would be a method that computes the Hamiltonian Schur form using the generic method described above, where only symplectic and simultaneously orthogonal similarity transformations are used. The existence of the Hamiltonian Schur form under rather generic assumptions, in particular under the spectral dichotomy condition we assume here, was proved in [82]. The resulting quest for a Hamiltonian QR algorithm became known under the name *Van Loan's curse* and is described in some detail in Chap. 1. In summary, the lack of the existence of a Hamiltonian or symplectic QR algorithm of cubic complexity under the same general assumptions that allow for structured Schur forms is due to the non-existence of structured Hessenberg-like forms that stay invariant under structured QR iterations. Byers [43, 44] was able to derive such a Hamiltonian Hessenberg form for the special case that $\text{rank}(F) = 1$ or $\text{rank}(G) = 1$ in (4.21). This allows to compute the Hamiltonian Schur form using orthogonal and symplectic similarity transformations in $\mathcal{O}(n^3)$ flops and to solve the CARE (4.23) and the associated continuous-time LQR problem via (4.25).

Under the same rank assumptions on G or F for the symplectic pencil in (4.21), Volker was able to derive an analogous QZ-type algorithm [75]. This algorithm then was probably the most involved eigenvalue algorithm of QR-type, with a technically demanding sequence of elementary eliminations in the bulge-chasing process!

In the following, we will discuss the Hamiltonian QR algorithm and Volker's contribution to stop the search for a Hamiltonian Hessenberg form in the general situation. We omit the symplectic case as it is technically much more involved and refer to the original paper [75] for details.

As already noted, the Hamiltonian QR algorithm should be based on orthogonal and symplectic similarity transformations. This implies a special structure.

Lemma 2 ([82]) *If $U \in \mathbb{R}^{2n \times 2n}$ is orthogonal and symplectic, then*

$$U = \begin{bmatrix} U_1 & U_2 \\ -U_2 & U_1 \end{bmatrix}, \quad U_1, U_2 \in \mathbb{R}^{n \times n}. \quad (4.29)$$

Moreover, as the intersection of two matrix groups, orthogonal symplectic matrices form a matrix group \mathcal{US}_{2n} with respect to matrix multiplication.

As elements of \mathcal{US}_{2n} are determined by the $2n^2$ parameters given by the entries of U_1, U_2 , only these parameters need to be stored and updated throughout a sequence of similarity transformations.

Now, the following theorem raises the hope that it is possible to find an algorithm based on symplectic and orthogonal similarity transformations for solving CAREs.

Theorem 2 ([82]) *If H is Hamiltonian and $\Lambda(H) \cap i\mathbb{R} = \emptyset$, then there exists $U \in \mathcal{US}_{2n}$ such that*

$$U^T H U = \begin{bmatrix} \hat{H}_{11} & \hat{H}_{12} \\ 0 & -\hat{H}_{11}^T \end{bmatrix}, \quad \hat{H}_{11}, \hat{H}_{12} \in \mathbb{R}^{n \times n}, \quad (4.30)$$

where \hat{H}_{11} is in real Schur form and $\Lambda(\hat{H}_{11}) = \Delta$ (the stable part of the spectrum of H).

Partitioning U from (4.30) as in (4.29), we have from (4.25) that the stabilizing solution of the CARE (4.23) is given by $X_c = U_2 U_1^{-1}$.

Remark 2 The decomposition given in (4.30) is called the *Hamiltonian Schur form*. It can be shown that such a form may also exist if eigenvalues on the imaginary axis are present. They have to satisfy certain properties, the most obvious one is that their algebraic multiplicity needs to be even; see [72] and Chap. 6.

As the *QR* algorithm is considered to be the best method for solving the dense non-symmetric eigenproblem, it is straightforward to strive for a symplectic variant of the *QR* algorithm converging to the Hamiltonian Schur form given in (4.30). A framework for such an algorithm can easily be derived. Denote the iterates of such an algorithm by H_j . If we choose the *QR* decomposition performed in each step, i.e., $p_j(H_j) = S_{j+1} R_{j+1}$, such that all S_{j+1} are symplectic and orthogonal, then it follows that all iterates $H_{j+1} = S_{j+1}^T H_j S_{j+1}$ are Hamiltonian. Unfortunately, such a *symplectic QR decomposition* does not always exist. Sets of matrices in $\mathbb{R}^{2n \times 2n}$ for which it exists are described in [39]. In particular, it is also shown there (see [44] for a constructive proof) that if M is symplectic, then there exists $S \in \mathcal{US}_{2n}$ such that

$$M = S R = S \begin{bmatrix} R_{11} & R_{12} \\ 0_n & R_{11}^{-T} \end{bmatrix} = \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix}, \quad (4.31)$$

where $R_{11}, R_{12} \in \mathbb{R}^{n \times n}$. Uniqueness of this decomposition can be achieved by requiring all diagonal entries of R_{11} to be positive.

As the matrix R in (4.31) is permutationally similar to an upper triangular matrix and the Hamiltonian Schur form is similar to the real Schur form using the same permutations, it can be shown under mild assumptions that such a Hamiltonian QR algorithm converges to Hamiltonian Schur form if it exists. Moreover, as only similarity transformations in \mathcal{US}_{2n} are used, the algorithm can be shown to be strong backward stable in the sense of Bunch [38].

Byers shows in [44] that if the rational function p_j is chosen to be the Cayley shift $c_k(t) := (t - \mu_k)(t + \mu_k)^{-1}$, where μ_k is an approximate real eigenvalue of H , or $d_k(t) := (t - \mu_k)(t - \overline{\mu_k})(t + \mu_k)^{-1}(t + \overline{\mu_k})^{-1}$, where μ_k is an approximate complex eigenvalue of H , then $p_j(H_j)$ is symplectic, and hence, the symplectic QR decomposition of $p_j(H_j)$ exists. In case $\pm\mu_k$ are exact eigenvalues of H and hence of H_j , then deflation is possible, and we can proceed with the deflated problem of smaller dimension without ever being forced to invert a singular matrix. In this way, a double or quadruple Hamiltonian QR step can be implemented.

Unfortunately, the so derived algorithm is of complexity $\mathcal{O}(n^4)$ as each symplectic QR decomposition requires $\mathcal{O}(n^3)$ flops and usually, $\mathcal{O}(n)$ iterations are required (based on the experience that for each eigenvalue, 1–2 iterations are needed). The missing part that would bring the computational cost down to $\mathcal{O}(n^3)$ is an initial reduction analogous to the Hessenberg reduction in the QR algorithm that

- is invariant under the similarity transformation performed in each step of the Hamiltonian QR algorithm (the *Hamiltonian QR step*);
- admits an implementation of the Hamiltonian QR step using only $\mathcal{O}(n^2)$ flops.

In [44] Byers shows that such a form exists.

Definition 3 A Hamiltonian matrix $H \in \mathbb{R}^{2n \times 2n}$ is in *Hamiltonian Hessenberg form* if

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & -H_{11}^T \end{bmatrix} = \begin{bmatrix} \square & \square \\ * & \square \end{bmatrix}, \quad (4.32)$$

where $H_{ij} \in \mathbb{R}^{n \times n}$, $i, j = 1, 2$, H_{11} is upper Hessenberg, and $H_{21} = \varphi e_n e_n^T$ with $\varphi \in \mathbb{R}$ and e_n being the n th unit vector. The Hamiltonian Hessenberg matrix H is unreduced if $h_{i+1,i} \neq 0$, $i = 1, \dots, n-1$, and $\varphi \neq 0$.

Byers [44] shows that if H_j is in Hamiltonian Hessenberg form and the rational function p_j is chosen as a Cayley shift, then H_{j+1} is in Hamiltonian Hessenberg form again and the Hamiltonian QR step can be implemented in $\mathcal{O}(n^2)$ flops.

The crux of this algorithm is the initial reduction of a Hamiltonian matrix to Hamiltonian Hessenberg form. Byers shows how this can be achieved if one of the off-diagonal blocks of the Hamiltonian matrix H in (4.21) has rank 1. (This is related to control systems of the form (4.2) having only one input ($m = 1$), i.e., *single-input systems* and/or only one output ($p = 1$), i.e., *single-output systems*.) But unfortunately no algorithm is known for reducing a general Hamiltonian matrix

to Hamiltonian Hessenberg form. But the situation is even worse. Analogous to the standard QR algorithm where the QR step is performed on unreduced Hessenberg matrices (possibly zeros on the subdiagonal are used for deflation, i.e., splitting the problem in two or more subproblems consisting of unreduced Hessenberg matrices), the Hamiltonian QR algorithm works for unreduced Hamiltonian Hessenberg matrices. The following theorem due to Ammar and Mehrmann [3] shows that the situation is in general hopeless with respect to the existence of the unreduced Hamiltonian Hessenberg form.

Theorem 3 *If $H \in \mathbb{R}^{2n \times 2n}$ is Hamiltonian, then there exists an orthogonal and symplectic matrix transforming H to unreduced Hamiltonian Hessenberg form if and only if the nonlinear set of equations*

$$x^T x = 1 \quad \text{and} \quad x^T J H^{2k-1} x = 0 \quad \text{for } k = 1, \dots, n-1,$$

has a solution that is not contained in an H -invariant subspace of dimension n or less.

Obviously, if JH is positive definite, such a vector cannot exist, showing that there really exist situations in which the unreduced Hamiltonian Hessenberg form does not exist. A similar result holds in the symplectic case. Therefore, other approaches have been investigated during the last decade.

4.4.1.3 The Multishift Algorithm

From Theorem 3 we know that the reduction to Hamiltonian Hessenberg form which is necessary to efficiently implement the Hamiltonian QR algorithm is in general not possible. Nevertheless, the same paper [3] suggested a possible alternative method that became the topic of my diploma thesis [8] and eventually lead to the paper [2].

The basis of this idea is that by orthogonal symplectic similarity transformations, the following reduction due to Paige and Van Loan [82] can be achieved.

Theorem 4 *Let $H \in \mathbb{R}^{2n \times 2n}$. Then there exists $U \in \mathcal{US}_{2n}$ such that*

$$U^T H U = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \begin{bmatrix} \begin{array}{|c|c|} \hline \diagdown & \square \\ \hline \end{array} & \square \\ \hline \begin{array}{|c|c|} \hline \diagdown & \square \\ \hline \end{array} & \square \\ \hline \end{bmatrix}, \quad (4.33)$$

where $H_{11} \in \mathbb{R}^{n \times n}$ is upper Hessenberg and $H_{21} \in \mathbb{R}^{n \times n}$ is upper triangular. The transformation matrix U can be chosen such that

$$U = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & \tilde{U}_1 & 0 & \tilde{U}_2 \\ \hline 0 & 0 & 1 & 0 \\ 0 & -\tilde{U}_2 & 0 & \tilde{U}_1 \end{array} \right]. \quad (4.34)$$

If in addition, H is Hamiltonian, then

$$U^T H U = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & -H_{11}^T \end{bmatrix} = \begin{bmatrix} \diagdown & \square \\ \diagup & \square \end{bmatrix}, \quad (4.35)$$

i.e., H_{21} is diagonal and H_{12} is symmetric.

The reduced form (4.35) of a Hamiltonian matrix will be called *PVL form* in the following. An algorithm for computing the transformation given in (4.35) is derived in [82]. It can be implemented using a finite number of similarity transformations and requires $\mathcal{O}(n^3)$ flops. Unfortunately, the PVL form is not preserved under the Hamiltonian *QR* iteration and can therefore not serve for the initial reduction step of the Hamiltonian *QR* algorithm. In the following, we will see that the PVL form can be used in what can be considered as a Hamiltonian *multishift QR* algorithm.

First, we need some more theory. As before, we denote $J = \begin{bmatrix} 0_n & I_n \\ -I_n & 0_n \end{bmatrix}$.

Definition 4 A real subspace $\mathcal{L} \subset \mathbb{R}^{2n}$ is *isotropic* iff $x^T J y = 0$ for all $x, y \in \mathcal{L}$. If \mathcal{L} is maximal, i.e., not contained in an isotropic subspace of larger dimension, then \mathcal{L} is a *Lagrangian* subspace (cf. Definition 1).

By the definition of symplectic matrices, we obtain immediately the following lemma.

Lemma 3 Let $S \in \mathbb{R}^{2n \times 2n}$ be symplectic. Then the first r columns of S , $1 \leq r \leq n$, span an isotropic subspace of \mathbb{R}^{2n} . For $r = n$, this subspace is Lagrangian.

The basis for the multishift algorithm is contained in the following result.

Proposition 2 ([3]) Let $H \in \mathbb{R}^{2n \times 2n}$ be a Hamiltonian matrix with spectrum $\Lambda(H) = \Delta_n \cup (-\Delta_n)$, $\Delta_n \cap (-\Delta_n) = \emptyset$, and $\Delta_n = \overline{\Delta_n} = \{\lambda_1, \dots, \lambda_n\}$. Then the multishift vector

$$x = \alpha(H - \lambda_1 I_{2n}) \cdots (H - \lambda_n I_{2n}) e_1, \quad \alpha \in \mathbb{R}, \quad (4.36)$$

where $e_1 \in \mathbb{R}^{2n}$ is the first unit vector, is contained in the n -dimensional H -invariant subspace corresponding to $-\Delta_n$. Moreover, this subspace is Lagrangian. In particular, if $\Delta_n \subset \mathbb{C}^+ := \{z \in \mathbb{C} \mid \Re(z) > 0\}$, then this Lagrangian subspace is the stable H -invariant subspace.

So, once we know the spectrum of H , we can compute *one* vector that is contained in the subspace required for solving the corresponding CARE. This observation can be combined with the computation of the PVL form in order to derive a *multishift* step as follows – assuming for simplicity that H has no eigenvalues on the imaginary axis.

Using this approach, it is possible to get the whole stable H -invariant subspace. The following theorem will indicate how Algorithm 1 can be used to achieve this.

Algorithm 1 One step of the multishift algorithm for Hamiltonian matrices**Multishift step**

1. Compute the multishift vector as in (4.36) with $\lambda_j \in \mathbb{C}^+$, $j = 1, \dots, n$. Choose α in (4.36) such that $\|x\|_2 = 1$. (If this is not possible, i.e., $x = 0$, then exit.)
2. Compute $U_1 \in \mathcal{US}_{2n}$ such that $U_1^T x = \pm e_1$.
3. Set $H_1 = U_1^T H U_1$.
4. Compute the PVL form of H_1 , i.e., compute $U_2 \in \mathcal{US}_{2n}$ such that $H_2 = U_2^T H_1 U_2 = (U_1 U_2)^T H (U_1 U_2)$ is in PVL form.

Theorem 5 Let $H \in \mathbb{R}^{2n \times 2n}$ be Hamiltonian and let \mathcal{V}_n be an n -dimensional H -invariant Lagrangian subspace corresponding to $\Delta_n \subset \Lambda(H)$ with Δ_n as in Proposition 2. Further, let the multishift vector x from (4.36) be computed using $-\Delta_n = \{\lambda_1, \dots, \lambda_n\}$ as the shifts. If $1 \leq p \leq n$ is the dimension of the minimal isotropic H -invariant subspace \mathcal{V}_p containing x , then after Step 4 of the multishift step, H_2 has the form

$$H_2 = \left[\begin{array}{cc|cc} A_{11} & A_{12} & G_{11} & G_{21} \\ 0 & A_{22} & G_{21}^T & G_{22} \\ \hline 0 & 0 & -A_{11}^T & 0 \\ 0 & F_{22} & -A_{12}^T & -A_{22}^T \end{array} \right] \begin{array}{l} \} p \\ \} n-p \\ \} p \\ \} n-p \end{array}, \quad (4.37)$$

where $A_{11} \in \mathbb{R}^{p \times p}$, $\Lambda(A_{11}) \subset \Delta_n$, and the Hamiltonian submatrix

$$H_{22} := \begin{bmatrix} A_{22} & G_{22} \\ F_{22} & -A_{22}^T \end{bmatrix} \in \mathbb{R}^{2(n-p) \times 2(n-p)}$$

is in PVL form.

Furthermore, for $U_1, U_2 \in \mathcal{US}_{2n}$ from the multishift step we have

$$U := U_1 U_2 = [u_1, \dots, u_p, u_{p+1}, \dots, u_{2n}] \in \mathcal{US}_{2n}, \quad u_j \in \mathbb{R}^{2n} \text{ for } j = 1, \dots, 2n,$$

and $\text{span}\{u_1, \dots, u_p\} = \mathcal{V}_p \subset \mathcal{V}_n$.

The detailed proof of this result is contained in [10].

The theorem shows that if the multishift vector x from (4.36) has components in all directions of a Lagrangian H -invariant subspace, then after one multishift step, a basis for this invariant subspace is given by the first n columns of $U_1 U_2$. Otherwise, the first p columns of $U_1 U_2$ span a p -dimensional H -invariant subspace contained in this subspace and the problem decouples into two subproblems. Algorithm 1 can then repeatedly be applied to the resulting Hamiltonian submatrix $H_{22} \in \mathbb{R}^{2(n-p) \times 2(n-p)}$ until $p = n$. The implementation of this algorithm is described in detail in [2].

As only orthogonal symplectic similarity transformations are used, a multishift step is strongly backward stable. The computational cost of one multishift step for $p = 0$ is around 15 % of the Schur vector method [71]. The complete computational cost depends on the number of iteration steps necessary. In a worst case scenario, i.e., in each step only one basis vector of \mathcal{V}_n is found, the complexity of this algorithm becomes basically $\mathcal{O}(n^4)$. This is rarely observed in praxis, though. On the other hand, rounding errors during the computation, in particular while forming the multishift vector, and the fact that the eigenvalues are usually only known approximately, make it practically impossible that deflation occurs exactly. Often, some iteration steps are necessary to detect deflation when using finite precision arithmetic. Generally speaking, as long as the size of the problem is modest ($n \leq 100$), the method is feasible and the number of required iterations is acceptable.

When solving CAREs, usually the stable H -invariant subspace is required. In that case, Δ_n in Proposition 2 has to be chosen such that $\Re(\lambda_j) < 0$ for all $j = 1, \dots, n$. Note that the stable H -invariant subspace is Lagrangian; see, e.g., [3, 76]. But observe that in principle, the multishift algorithm can be used to compute the CARE solution corresponding to any Lagrangian H -invariant subspace. This is of particular importance in some applications, e.g., in some \mathcal{H}_∞ -control problems, ARE solutions exist and have to be computed if H has eigenvalues on the imaginary axis. As long as these eigenvalues permit a Lagrangian invariant subspace, the corresponding ARE solutions can be computed by the multishift algorithm.

The computation of the multishift vector in (4.36) requires the knowledge of the spectrum of H . Hence, what remains to show is how to obtain the eigenvalues of a Hamiltonian matrix H . One possibility is to run the QR algorithm without accumulating the transformations. But then the problems with eigenvalues close to the imaginary axis as mentioned above have to be expected. A different approach, which costs only one third of the QR algorithm and takes the symmetry of $\Lambda(H)$ into account, was suggested by Van Loan [87]. Consider $K := H^2$. Obviously, if $\lambda \in \Lambda(H)$, then $\lambda_K := \lambda^2 \in \Lambda(K)$. If $\Re(\lambda) \neq 0$, then λ_K is a double eigenvalue of K due to the symmetry of $\Lambda(H)$. Squared Hamiltonian matrices are *skew-Hamiltonian*, that is, they satisfy $KJ = -(KJ)^T$ and therefore have the explicit block structure

$$K = \begin{bmatrix} K_1 & K_2 \\ K_3 & K_1^T \end{bmatrix}, \quad K_2 = -K_2^T, \quad K_3 = -K_3^T. \quad (4.38)$$

The skew-Hamiltonian structure is preserved under symplectic similarity transformations [87]. Hence, computing the PVL form (4.33) for skew-Hamiltonian matrices yields

$$U^T H^2 U = U^T K U = \begin{bmatrix} \tilde{K}_1 & \tilde{K}_2 \\ 0 & \tilde{K}_1^T \end{bmatrix} = \begin{bmatrix} \square & \square \\ & \square \end{bmatrix}. \quad (4.39)$$

Hence, $\Lambda(K)$ can be obtained by computing the eigenvalues of the upper Hessenberg matrix \tilde{K}_1 , e.g., by applying the *QR* iteration to \tilde{K}_1 . Let $\Lambda(\tilde{K}_1) = \{\mu_1, \dots, \mu_n\}$, then $\Lambda(H) = \{\pm\sqrt{\mu_1}, \dots, \pm\sqrt{\mu_n}\}$. Note that no information about eigenvectors or invariant subspaces of H is obtained.

The resulting method is strongly backward stable for K and preserves the symmetry structures of $\Lambda(K)$ and $\Lambda(H)$. An implicit version of this algorithm is also suggested in [87]; U from (4.39) is applied directly to the Hamiltonian matrix such that $\tilde{H} := U^T H U$ is *square-reduced*, i.e., \tilde{H}^2 has the form given in (4.39). The disadvantage of Van Loan's method is that a loss of accuracy up to half the number of significant digits of the computed eigenvalues of H is possible. An error analysis in [87] shows that for a computed simple eigenvalue $\tilde{\lambda}$ corresponding to $\lambda \in \Lambda(H)$ we have

$$|\lambda - \tilde{\lambda}| \approx \min \left\{ \frac{\varepsilon \|H\|_2^2}{s(\lambda)|\lambda|}, \frac{\sqrt{\varepsilon} \|H\|_2}{s(\lambda)} \right\} = \varepsilon \frac{\|H\|_2}{s(\lambda)} \times \min \left\{ \frac{\|H\|_2}{|\lambda|}, \frac{1}{\sqrt{\varepsilon}} \right\}, \quad (4.40)$$

where $s(\lambda)$, the reciprocal condition number of λ , is the cosine of the acute angle between the left and right eigenvectors of H corresponding to λ . Basically, this error estimate indicates that eigenvalues computed by Van Loan's method are as accurate as those computed by a numerically backward stable method provided that $\lambda \approx \|H\|_2$ while for $\lambda \ll \|H\|_2$, the error grows with the ratio $\|H\|_2/|\lambda|$.

Usually, eigenvalues computed by Van Loan's method are satisfactory as shifts for the multishift algorithm and in most other practical circumstances. On the other hand, removing the possible $1/\sqrt{\varepsilon}$ loss of accuracy provides the motivation of the algorithm presented in the next section.

4.4.1.4 A Method Based on the Symplectic URV Decomposition

The method described in this subsection was the key to breaking Van Loan's curse as already described in Chap. 1. As we investigate it here in the context of solving CAREs, we will also need some details and therefore repeat the essential steps.²

The central problem of Van Loan's method is that squaring the Hamiltonian matrix leads to a possible loss of half of the accuracy. For products of general matrices, this possible loss of accuracy caused by forming the product can be circumvented by employing the *periodic* or *cyclic QR algorithm* [37, 59, 60].

²A personal remark: We derived this method over several months in 1995/1996, regularly meeting with Hongguo Xu, then a Humboldt fellow at TU Chemnitz-Zwickau in Volker's group, on Thursdays in Volker's office. The quest was to avoid the loss of accuracy due to the explicit squaring of H in Van Loan's method. When Hongguo wrote the key step, the symplectic URV decomposition, on Volker's blackboard one Thursday morning, this was one of the most beautiful and enlightening moments in my career as a mathematician.

If $A = A_1 \cdot A_2 \cdots A_p$, where $A_j \in \mathbb{R}^{n \times n}$, $j = 1, \dots, p$, then this algorithm computes the real Schur form of A without forming A explicitly. This is achieved by cyclically reducing the factors A_j to (quasi-)upper triangular form:

$$U^T A U = (U_1^T A_1 U_2)(U_2^T A_2 U_3) \cdots (U_p^T A_p U_1) = \begin{bmatrix} \diagdown & & \\ & \diagdown & \\ & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \diagdown & & \\ & \diagdown & \\ & & \ddots \end{bmatrix} \cdots \begin{bmatrix} \diagdown & & \\ & \diagdown & \\ & & \ddots \end{bmatrix}. \quad (4.41)$$

Here, $U_1^T A_1 U_2$ is in real Schur form while $U_j^T A_j U_{(j+1) \bmod p}$, $j = 2, \dots, p$, are upper triangular such that the product is in real Schur form. The eigenvalues are then obtained from computing the eigenvalues of the 1×1 and 2×2 blocks on the diagonal of the product in (4.41). This method is numerically backward stable and avoids the loss of accuracy in the eigenvalues as the product A is never formed explicitly.

The idea is now to employ this approach to H^2 by replacing the reduction of H^2 to PVL form by $U^T H^2 U = (U^T H V)(V^T H U)$, where $U, V \in \mathcal{US}_{2n}$. This can be achieved by the *symplectic URV-like decomposition* given in [28].

Proposition 3 For $H \in \mathbb{R}^{2n \times 2n}$ there exist $U, V \in \mathcal{US}_{2n}$ such that

$$V^T H U = \begin{bmatrix} H_1 & H_3 \\ 0 & -H_2^T \end{bmatrix} = \begin{bmatrix} \diagdown & \square \\ 0 & \diagdown \end{bmatrix}, \quad (4.42)$$

i.e., H_1 is upper triangular and H_2 is upper Hessenberg. If, in addition, H is Hamiltonian, then

$$U^T H^2 U = \begin{bmatrix} H_2 H_1 & H_2 H_3 - (H_2 H_3)^T \\ 0 & (H_2 H_1)^T \end{bmatrix} = \begin{bmatrix} \diagdown & \square \\ 0 & \diagdown \end{bmatrix} \quad (4.43)$$

and the eigenvalues of H are the positive and negative square roots of the eigenvalues of the upper Hessenberg matrix $H_2 H_1$.

That is, using the decomposition given in (4.42) we obtain the PVL form of H^2 without explicitly squaring H . In order to obtain the eigenvalues of H we then apply the periodic *QR* algorithm to $H_2 H_1$.

In [28] an algorithm for computing the decomposition given in (4.42) is presented. It requires a finite number of transformations. The combined cost of computing the decomposition (4.42) and applying the periodic *QR* algorithm to $H_2 H_1$ is about $48n^3$ flops – this is $1.5 \times$ the computational cost of Van Loan’s method and about 60 % of the cost of the *QR* algorithm applied to a non-symmetric $2n \times 2n$ matrix. The method is numerically backward stable as only orthogonal transformations are used. The symmetry property of $\Lambda(H)$ is preserved and in this sense the method can be considered to be strongly backward stable.

A detailed error analysis of the above method yields the following result [28]. Essentially (under mild assumptions), for a nonzero and simple eigenvalue λ of a Hamiltonian matrix $H \in \mathbb{R}^{2n \times 2n}$, the algorithm based on the symplectic URV-like decomposition followed by applying the periodic QR algorithm to $H_2 H_1$ from (4.42) yields a computed eigenvalue $\tilde{\lambda}$ satisfying

$$|\tilde{\lambda} - \lambda| \leq \frac{2\|H\|\varepsilon}{s(\lambda)} + \mathcal{O}(\varepsilon^2).$$

This is the accuracy to be expected from any backward stable method like the QR algorithm and shows that by avoiding to square H , we get the full possible accuracy.

Nevertheless, as Van Loan's method, the approach presented above does not provide the H -invariant subspaces. But based on (4.42) it is possible to derive an algorithm that can be used to compute the stable H -invariant subspace and the solution of the CARE (4.23) [27]. The basis for this algorithm is the following theorem.

Theorem 6 ([27]) *Let $A \in \mathbb{R}^{n \times n}$ and define $B = \begin{bmatrix} 0 & A \\ A & 0 \end{bmatrix}$. Then the spectra of A and B are related via $\Lambda(B) = \Lambda(A) \cup (-\Lambda(A))$. Further, let $\Lambda(A) \cap i\mathbb{R} = \emptyset$. If the columns of $[U_1^T, U_2^T]^T \in \mathbb{R}^{2n \times n}$ span an orthogonal basis for a B -invariant subspace such that*

$$B \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} R, \quad \Lambda(R) \subset \mathbb{C}^+ \cap \Lambda(B),$$

then $\text{range}(U_1 + U_2)$ is the A -invariant subspace corresponding to $\Lambda(A) \cap \mathbb{C}^+$ and $\text{range}(U_1 - U_2)$ is the stable A -invariant subspace.

An orthogonal basis for the subspace defined by $\text{range}(U_1 - U_2)$ can be obtained, e.g., from a rank-revealing QR decomposition of $U_1 - U_2$; see, e.g., [56].

In general it is of course not advisable to use the above result in order to obtain the stable invariant subspace of a matrix A as one would have to double the dimension and thereby increase the computational cost and required workspace significantly as compared to applying the QR algorithm to A . But we will see that for Hamiltonian matrices, the given structure makes this approach very attractive.

Let $H \in \mathbb{R}^{2n \times 2n}$ be Hamiltonian with $\Lambda(H) \cap i\mathbb{R} = \emptyset$. Define a permutation matrix $P \in \mathbb{R}^{4n \times 4n}$ by

$$P = \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & 0 & I_n & 0 \\ 0 & I_n & 0 & 0 \\ 0 & 0 & 0 & I_n \end{bmatrix}.$$

Then $P^T \begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix} P$ is a Hamiltonian matrix in $\mathbb{R}^{4n \times 4n}$. The basic idea is now to employ the decomposition (4.42) in order to make $P^T H P$ block-upper triangular.

Therefore, let $\hat{U}, \hat{V} \in \mathcal{US}_{2n}$ be as in Proposition 3 such that $\hat{V}^T H \hat{U}$ has the form given in (4.42). Then we apply the periodic QR algorithm to $H_2 H_1$. From this we obtain orthogonal matrices $V_1, V_2 \in \mathbb{R}^{n \times n}$ such that both, the product

$$(V_1^T H_2 V_2)(V_2^T H_1 V_1) =: \hat{H}_2 \hat{H}_1$$

and \hat{H}_2 , are in upper real Schur form while \hat{H}_1 is upper triangular. Define

$$U_1 := \hat{U} \begin{bmatrix} V_1 & 0 \\ 0 & V_1 \end{bmatrix}, \quad U_2 := \hat{V} \begin{bmatrix} V_2 & 0 \\ 0 & V_2 \end{bmatrix}, \quad \text{and} \quad U := \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}.$$

Then

$$B := P^T U^T \begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix} U P = \begin{bmatrix} 0 & \hat{H}_2 & 0 & \hat{H}_3^T \\ \hat{H}_1 & 0 & \hat{H}_3 & 0 \\ 0 & 0 & 0 & -\hat{H}_1^T \\ 0 & 0 & -\hat{H}_2^T & 0 \end{bmatrix}$$

is Hamiltonian and block upper triangular with \hat{H}_1 upper triangular, \hat{H}_2 in real Schur form, and $\hat{H}_3 = V_2^T (H_2 H_3 - H_3^T H_2^T) V_1$.

Now let U_3 be orthogonal such that

$$U_3^T \begin{bmatrix} 0 & \hat{H}_2 \\ \hat{H}_1 & 0 \end{bmatrix} U_3 = \begin{bmatrix} T_1 & T_3 \\ 0 & -T_2 \end{bmatrix} \quad (4.44)$$

is in upper real Schur form with $T_j \in \mathbb{R}^{n \times n}$, $j = 1, 2, 3$, and $\Lambda(T_1) = \Lambda(T_2) \subset \mathbb{C}^+$. Note that this is possible as the eigenvalues of $\begin{bmatrix} 0 & \hat{H}_2 \\ \hat{H}_1 & 0 \end{bmatrix}$ are exactly those of H^2 and $\Lambda(H) \cap i\mathbb{R} = \emptyset$. Hence,

$$\tilde{B} := \begin{bmatrix} U_3^T & 0 \\ 0 & U_3^T \end{bmatrix} B \begin{bmatrix} U_3 & 0 \\ 0 & U_3 \end{bmatrix} = \begin{bmatrix} T_1 & T_3 & R_1 & R_2 \\ 0 & -T_2 & R_2^T & R_3 \\ 0 & 0 & -\tilde{T}_1^T & 0 \\ 0 & 0 & -\tilde{T}_3^T & \tilde{T}_2^T \end{bmatrix}$$

is in Hamiltonian Schur form. In order to apply Theorem 6, we need to reorder the eigenvalues in the Hamiltonian Schur form such that all eigenvalues in the upper left $2n \times 2n$ block are in the open right half plane. This can be achieved, e.g., by the symplectic re-ordering algorithm due to Byers [43, 44]. With this algorithm it is possible to determine $\tilde{U} \in \mathcal{US}_{2n}$ such that

$$\tilde{U}^T \tilde{B} \tilde{U} = \begin{bmatrix} T_1 & \tilde{T}_3 & R_1 & \tilde{R}_2 \\ 0 & \tilde{T}_2 & \tilde{R}_2^T & R_3 \\ 0 & 0 & -\tilde{T}_1^T & 0 \\ 0 & 0 & -\tilde{T}_3^T & -\tilde{T}_2^T \end{bmatrix}, \quad \Lambda(\tilde{T}_2) = \Lambda(T_2).$$

Now define

$$S := P^T \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} P \begin{bmatrix} U_3 & 0 \\ 0 & U_3 \end{bmatrix} \tilde{U}. \quad (4.45)$$

Then $S \in \mathcal{US}_{4n}$ and

$$T := S^T P^T \begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix} P S =: \begin{bmatrix} T_{11} & T_{12} \\ 0 & -T_{11}^T \end{bmatrix} \quad (4.46)$$

is in Hamiltonian Schur form with $\Lambda(T_{11}) \subset \mathbb{C}^+$. Now we can apply Theorem 6 with A replaced by H and $R := T_{11}$.

Corollary 1 *Let $H \in \mathbb{R}^{2n \times 2n}$ be Hamiltonian with $\Lambda(H) \cap i\mathbb{R} = \emptyset$ and let S be as in (4.45) such that (4.46) holds. If $PS := \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$, with $S_{ij} \in \mathbb{R}^{2n \times 2n}$, then the n -dimensional, stable H -invariant subspace is given by $\text{range}(S_{11} - S_{21})$.*

The above transformations yielding S are described in more detail in [27]. The solution of the CARE can be obtained from an orthogonal basis of $\text{range}(S_{11} - S_{21})$ computed by a rank-revealing QR decomposition or directly from $S_{11} - S_{21}$; for details see [27]. The latter approach saves a significant amount of work such that the cost of the algorithm described above for computing the stabilizing solution of the CARE (4.23) is approximately 60% of the cost of the Schur vector method.

Remark 3 The transformation of $\begin{bmatrix} 0 & \hat{H}_2 \\ \hat{H}_1 & 0 \end{bmatrix}$ to real Schur form and the computation of the matrix U_3 in (4.44) can be efficiently implemented employing the available structure. An algorithm for this is given in [27].

It is shown in [27] that the algorithm presented above is strongly backward stable in $\mathbb{R}^{4n \times 4n}$. That is, if \tilde{S} is the analogue to S from (4.45) computed in finite precision arithmetic, then

$$\tilde{S}^T P^T \begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix} P \tilde{S} = T + E,$$

with T as in (4.46), $\|E\|_2 \leq c\varepsilon\|H\|_2$ for a small constant c and $E \in \mathbb{R}^{4n \times 4n}$ is Hamiltonian. Moreover it is shown in [27] that the computed invariant subspace is as accurate as the maximum of its condition number and the condition number of its complimentary (*antistable*) H -invariant subspace permit. This is to be expected from the fact that at the same time we compute the stable H -invariant subspace, by Theorem 6 we also compute the antistable H -invariant subspace. In that sense the algorithm is not optimal as we would like the accuracy of the computed subspace to be limited only by its own condition number.

The algorithms described in this section are implemented in Fortran 77, see [19], while an implementation of Van Loan's method with scaling is provided in [12]. All

these methods are also integrated in SLICOT,³ the Subroutine **L**ibrary in **C**ontrol **T**heory [26]. This subroutine library is based on a joint initiative of European researchers, in which Volker also was a driving force.

The invariant subspace computation as described above turned out to be not completely satisfactory. For some examples with eigenvalues close to the imaginary axis, unexpectedly large errors are encountered in the computed CARE solutions. Almost 10 years later, Volker and co-workers came up with the observation that the orthogonal and symplectic matrix U that puts H^2 into skew-Hamiltonian Schur form contains information that can be used to construct the stable H -invariant subspace in a recursive fashion [49]. This matrix U is obtained from the symplectic URV decomposition (4.42) with the additional step of accumulating the matrices obtained from the periodic QR algorithm applied to H_2H_1 into U, V . This approach is described in some detail in Sect. 1.4, we therefore refrain from recapitulating it here. An intuitive explanation of the method was given by Watkins in [89], and a block version of the algorithm that can better deal with clusters of eigenvalues is described in [79]. It can be concluded that Volker and co-workers have eventually found a method to solve CAREs via the approach based on invariant subspaces of the Hamiltonian eigenproblem that satisfies all desired properties regarding numerical stability, structure-preservation, and efficiency.

Analogous procedures for solving the DARE via the symplectic pencil approach have not yet been derived. On the other hand, by employing the generalized Cayley transformation approach described in the previous section, one may use the generalizations of this approach to skew-Hamiltonian/Hamiltonian or even/odd matrix pencils described by Volker and co-workers in [15, 24]. Nevertheless, there is still room for further improvement: the methods discussed in [15, 24] extend the approach from [27] rather than the more robust methods from [49, 79]. Also, the available software for structured matrix pencils [33, 34] is still based on the skew-Hamiltonian/Hamiltonian pencil structure rather than the more general structure of even or alternating matrix pencils.

4.4.2 Defect Correction

In this subsection, we re-visit a topic that is very important for obtaining solutions to CAREs and DAREs at highest possible accuracy. If a numerical solution \tilde{X} of the CARE (4.23) is computed, this process is prone to roundoff errors, thus \tilde{X} can only be an approximation to the solution X of (4.23). In particular for algorithms that are not numerically backward stable, like the SR algorithm discussed in Chap. 1 and the previous subsection, methods based on sign and disk functions, but also symplectic QR algorithms in case the final step of obtaining X via $X = -U_2U_1^{-1}$ as in (4.25) is ill-conditioned, these roundoff errors may lead to deteriorating accuracy. Improving

³Available at www.slicot.org

this approximation may be achieved using Newton's method, but alternatively also by *defect correction*. The necessary theory was derived in [80] and [76, Chapter 10]. We will provide this here in some detail, as the defect correction principle has, despite its importance for practice, received little attention since, and in particular new computing platforms including hardware accelerators may use this principle for obtaining fast and reliable algorithms, as has been already suggested for Lyapunov equation in [18].

The defect correction principle is most easily explained for linear systems of equations. Given an approximate solution \tilde{x} of the underlying problem

$$Ax = b,$$

given $A \in \mathbb{R}^{n \times n}$ nonsingular, $b \in \mathbb{R}^n$, we determine the *residual* $r = b - A\tilde{x}$, then compute a solution δ of the linear system

$$A\delta = r,$$

and set

$$x^+ = \tilde{x} + \delta.$$

This process can be iterated until a given stopping criterion is satisfied. Defect correction is successful as long as the residual can be computed accurately enough, i.e., with higher precision than \tilde{x} . Such higher precision may be obtained, e.g., using double precision in case \tilde{x} was computed in single precision. There is no need to compute \tilde{x} and δ by the same algorithm, providing great flexibility to the defect correction principle.

The basis for applying defect correction to CAREs is provided by the following theorem due to Mehrmann/Tan [80]

Theorem 7 *Let $X = X^T$ be a solution of (4.23), and \tilde{X} a symmetric approximation to this solution. Define $P = X - \tilde{X}$, $\tilde{A} = A - G\tilde{X}$, and let*

$$\tilde{F} = F + A^T \tilde{X} + \tilde{X} A - \tilde{X} G \tilde{X}$$

be the residual of (4.23) with respect to \tilde{X} . Then P satisfies the CARE

$$0 = \tilde{F} + P \tilde{A} + \tilde{A}^T P - PGP. \quad (4.47)$$

The theorem is proved by simple algebraic manipulations after inserting $X = \tilde{X} + P$ in (4.23).

Due to the non-uniqueness of CARE solutions, it is important to guarantee that the updated approximate solution $\tilde{X} + P$ is still the one of interest, usually the stabilizing solution in control applications. In order to show that this is indeed the case, first it is necessary to check whether the stabilizability property related to (4.23) carries over to (4.47).

Lemma 4 *Given (4.23) with (A, G) stabilizable, i.e.,*

$$\text{rank}[\lambda I - A, G] = n \quad \forall \lambda \in \mathbb{C} : \Re(\lambda) \geq 0.$$

Then $\tilde{A} = A - G\tilde{X}$ is stabilizable as well.

Proof

$$\begin{aligned} \text{rank}[\lambda I_n - \tilde{A}, G] &= \text{rank}[\lambda I_n - A + G\tilde{X}, G] \\ &= \text{rank} \begin{bmatrix} [\lambda I_n - A, G] & \begin{bmatrix} I_n & 0 \\ \tilde{X} & I_n \end{bmatrix} \end{bmatrix} = \text{rank}[\lambda I_n - A, G]. \quad \square \end{aligned}$$

As we have seen before, the positive semidefinite solution of the CARE (4.23) can be obtained from the invariant subspace of the Hamiltonian matrix

$$H = \begin{bmatrix} A & G \\ F & -A^T \end{bmatrix}$$

corresponding to the eigenvalues in \mathbb{C}^- . Forming the Hamiltonian matrix corresponding to the defect CARE (4.47),

$$\tilde{H} = \begin{bmatrix} \tilde{A} & G \\ \tilde{F} & -\tilde{A}^T \end{bmatrix},$$

we see that $\tilde{H} = Q^{-1}HQ$ using

$$Q = \begin{bmatrix} I & 0 \\ -\tilde{X} & I \end{bmatrix}.$$

This immediately implies that $\Lambda(H) = \Lambda(\tilde{H})$, so that also \tilde{H} has exactly n eigenvalues with negative real parts. Now assume the \tilde{H} -invariant subspace corresponding to these eigenvalues is spanned by $\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \in \mathbb{R}^{2n \times n}$, that is,

$$\begin{bmatrix} \tilde{A} & G \\ \tilde{F} & -\tilde{A}^T \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} Z, \quad (4.48)$$

where all eigenvalues of $Z \in \mathbb{R}^{n \times n}$ have negative real parts.

By Lemma 4, (\tilde{A}, G) is stabilizable. Using standard arguments of control theory [52], it can be shown then that Z_1 is invertible and that $Z_2 Z_1^{-1}$ is symmetric negative semidefinite. Inserting $P := -Z_2 Z_1^{-1}$ in (4.48), we obtain

$$\begin{bmatrix} \tilde{A} & G \\ \tilde{F} & -\tilde{A}^T \end{bmatrix} \begin{bmatrix} I \\ -P \end{bmatrix} = \begin{bmatrix} I \\ -P \end{bmatrix} \tilde{Z}, \quad (4.49)$$

where $\tilde{Z} = Z_1 Z Z_1^{-1}$ has only eigenvalues with negative real parts, too.

Expanding terms in (4.49), the first row yields

$$\tilde{A} - GP = \tilde{Z}, \quad (4.50)$$

implying that P is stabilizing, and from the second row,

$$\tilde{F} + \tilde{A}^T P = -P \tilde{Z} = -P \tilde{A} + PGP,$$

we see that P is a solution of (4.47). In summary, P must be the unique symmetric positive semidefinite solution of (4.47).

Now with $\tilde{A} = A - G\tilde{X}$ it follows from (4.50), that

$$A - G(\tilde{X} + P) = \tilde{Z}$$

is stable, so that in summary we obtain the following result:

Theorem 8 *Suppose the invariant subspace of the Hamiltonian matrix*

$$\tilde{H} = \begin{bmatrix} \tilde{A} & G \\ \tilde{F} & -\tilde{A}^T \end{bmatrix}$$

corresponding to (4.47) is spanned by the columns of $\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \in \mathbb{R}^{2n \times n}$, and $P = -Z_2 Z_1^{-1}$, then $\tilde{X} + P$ is the unique symmetric positive semidefinite solution of (4.47).

Based on Theorems 7 and 8, we may now formulate a defect correction algorithm for CAREs.

As there are no specifications given for the methods employed in Steps 1 and 3 of Algorithm 2, one could use any numerical method to solve the CAREs, and possibly different ones in Steps 1 and 3. For instance, one could simply use a fast, but potentially unstable, method as the SR algorithm, as there is no need to have an \tilde{X} of high accuracy. As $\|P\|$ will be very small in general, the quadratic term in the defect CARE is basically negligible, Newton's method is a natural choice in Step 3. Another option is to employ an algorithm like the orthogonal symplectic multishift algorithm in which part of the computations from Step 1 can be re-used

Algorithm 2 Defect correction for the CARE (4.23)

Algorithm DC_CARE

Input. $A, G, F \in \mathbb{R}^{n \times n}$ as in (4.23); tolerance $\varepsilon > 0$ for the stopping criterion.

Output. Approximation \tilde{X} to the unique stabilizing solution of (4.23) and an error estimator $P \in \mathbb{R}^{n \times n}$ with $\|P\| \leq \varepsilon$.

Step 1. Compute stabilizing approximate solution \tilde{X} of (4.23) with the favorite CARE solver.

Step 2. Set $P := \tilde{X}$, $\tilde{X} := 0$.

Step 3.

while $\|P\| > \varepsilon$ **do**

 Set

$$\tilde{X} := \tilde{X} + P$$

$$\tilde{F} := F + A^T \tilde{X} + \tilde{X} A - \tilde{X} G \tilde{X}$$

$$\tilde{A} := A - G \tilde{X}$$

 Compute a stabilizing solution of the defect CARE

$$0 = \tilde{F} + \tilde{A}^T P + P \tilde{A} - P G P$$

end

in Step 3, so that the cost in Step 3 can be reduced. Such a variant is discussed in [2, 8].

The Algorithm DC_CARE is numerically backward stable in the sense that \tilde{X} is the exact solution of the following CARE with perturbed data:

$$0 = F - \tilde{F} + A^T X + X A - X G X,$$

where \tilde{F} is the residual from Theorem 7. If the leading significant digits of the defect P in Step 3 are computed correctly, the approximate solution \tilde{X} converges to the exact stabilizing solution X of (4.23). In practice, this will lead to reduced accuracy of the residual \tilde{F} due to cancelation which leads to a limitation of the obtainable accuracy. Nevertheless, the accuracy of the CARE is often greatly improved by 1–2 steps of defect correction, as several examples in [8] indicate.

A defect correction procedure for the DARE (4.24) can be derived in a completely analogous fashion; see [80] and [76, Chapter 10]. An interesting aspect of further research would be to derive a mixed precision CPU-GPU variant of Algorithm DC_CARE in the fashion of [18]. Also, in case of large-scale sparse solvers for CAREs as recently reviewed in [32], where low-rank approximations to the stabilizing solution are computed, it would be necessary to be able to represent \tilde{F} in low-rank format to use this concept. Whether this is possible or not is an open problem.

4.4.3 *Other Contributions to the Numerical Solution of AREs*

Volker's many contributions to the numerical solution of AREs go well beyond the described methods based on structured eigenproblems. A classical solution method of AREs is Newton's method, given their nature as nonlinear systems of equations. Volker has contributed to the convergence theory of Newton's method applied to the generalized CAREs (4.7) and DAREs (4.8), see [76, § 11]. He attributes these contributions mainly to Ludwig Elsner, but they had not been published before.

A notable contribution to understanding why and when the sign function method for CAREs [45] can be expected to yield accurate solutions was the perturbation analysis derived in [46].

Recently, Volker together with Federico Poloni has explored the use of permuted graph matrices to represent invariant and deflating subspaces of Hamiltonian matrices and pencils. This has often a tremendously positive effect on the numerical accuracy of iterative methods based on the inverse-free iteration/disk function method [5, 11, 31] and doubling-like algorithms [50, 51]. See the Chap. 5 for further details and references.

An important aspect in deriving numerical methods for AREs is to test them on challenging examples and to compare the performance to existing methods using well-defined benchmarks. Volker was the driving force in establishing benchmark collections for CAREs [20] and DAREs [21], see [22] for an overview. This had a very positive effect on the publication attitudes of new numerical methods for AREs as non competitive methods can be identified easily since these benchmark collections became available. Later on, he also inspired a number of other benchmark collections for computational control problems that became part of the SLICOT project, see also [25].

Together with Petko Petkov and Mihail Konstantinov, Volker also contributed and tested mathematical software for solving CAREs [84, 85].

Although Volker had not published on solving large-scale AREs until recently [78], he inspired much of the work in this area by putting his Ph.D. student Thilo Penzl on this track. His thesis [83] is now considered the starting point for many of the currently used methods for large-scale matrix equations, see also the later paper [23] that was published only 8 years after Thilo's unfortunate passing in 1999 due to a tragic accident. A survey on the developments of this prospering field was recently given in [32].

4.5 **Avoiding AREs**

Already in his early work related to his habilitation thesis and the book [76], Volker often made the point that in solving LQR problems, it may not be necessary to solve AREs explicitly. The basic idea of this can be presented using the continuous-time LQR problem. Borrowing the most recent representation, we can associate an even

matrix pencil to the LQR problem (4.1) for the descriptor or LTI system (4.2):

$$\lambda K - L := \lambda \begin{bmatrix} 0 & E & 0 \\ -E^T & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & A & B \\ A^T & C^T Q C & -C^T S \\ B^T & S^T C & R \end{bmatrix}. \quad (4.51)$$

This matrix pencil considered as a matrix polynomial has the leading coefficient skew-symmetric, the constant term is symmetric, and therefore is skew-symmetric/symmetric, or more general, an alternating matrix polynomial, see the Chaps. 2, 6, and 12 for more on these structures. Assuming for simplicity E, R nonsingular, the key observation now is that if the feedback solution $u(t) = F_c x(t)$ is sought, and an n -dimensional stable deflating subspace of the even pencil $\lambda K - L$ from (4.51) exists, spanned by the columns of $[V^T, U^T, W^T]^T$ with $U \in \mathbb{R}^{n \times n}$ invertible and V, W of size according to (4.51), then, following [76, § 6], we get

$$F_c = -R^{-1}(B^T X_c E + S^T C) = WU^{-1}.$$

Hence, the optimal feedback control law can be determined without computing the CARE solution! The latter is determined via $X_c E = VU^{-1}$ in this setting. Working with the even matrix pencil in (4.1) has the additional advantage that no linear systems with R need to be solved in forming the coefficients and thus, rounding errors in forming these coefficients are avoided [14]. This principle also carries over to the discrete-time case, see again [76, § 6] for the unstructured setting.

Avoiding the solution of AREs is even more desirable in H_∞ control. We will not go into the details of this problem, that has become a paradigm in robust control. It suffices to understand that the H_∞ -optimal controller is usually determined using the so called γ -iteration and subsequent controller formulas based on the output of this iteration. The γ -iteration is classically formulated in such a way that (in the continuous-time case) two CAREs need to be solved in each iteration step, and a spectral condition of their product is checked, see, e.g., [57, 90]. The crux is that these CAREs usually have indefinite quadratic terms which does not allow to solve them with Newton's method, and the eigenvalues of the associated Hamiltonian matrices are often close to the imaginary axis (in particular close to the "optimal" γ) which makes them difficult to solve by methods based on invariant or deflating subspaces of the associated Hamiltonian matrices. Moreover, often their norms tend to infinity which yields poorly scaled problems, implying additional numerical difficulties. For all these reasons, it was desirable to avoid the CAREs in the process. Volker and co-workers were able to derive a numerically robust method that achieves this, see [16, 17] for the LTI case and [73] for the descriptor case. Later, also controller formula based on this approach were presented in [13]. For details, we refer to [24], where also other applications of using extended pencil formulations in even or alternating form are discussed.

4.6 Concluding Remarks

Over the last 20 years, Volker and I have collaborated on methods for algebraic Riccati equations, and on how to avoid them. We often had differing opinions on whether or not the ARE is the right concept to use for solving certain control problems. These were always inspiring and fruitful discussions, and they certainly have been driving both of us to improve methods and ideas more and more. Today, we basically agree on which problems should be solved using AREs, and where one should avoid them by using structured matrix pencil methods as discussed, e.g., in Chaps. 1–3, and 5. Therefore, I foresee interesting research tasks in both directions for certainly another decade and more, as with increasing computer power, model complexity in control engineering problems is increasing, and will require new ideas and further development of the methods for DREs, AREs, and the related pencil problems at hand. I truly hope Volker and I will also be part of this development. In writing this chapter, a number of open problems and ideas already evolved, and I hope some of these can lead to improved methods and further inside in the theoretical and numerical treatment of DREs and AREs.

References

1. Abou-Kandil, H., Freiling, G., Ionescu, V., Jank, G.: *Matrix Riccati Equations in Control and Systems Theory*. Birkhäuser, Basel (2003)
2. Ammar, G.S., Benner, P., Mehrmann, V.: A multishift algorithm for the numerical solution of algebraic Riccati equations. *Electron. Trans. Numer. Anal.* **1**, 33–48 (1993)
3. Ammar, G.S., Mehrmann, V.: On Hamiltonian and symplectic Hessenberg forms. *Linear Algebra Appl.* **149**, 55–72 (1991)
4. Arnold, W.F., Laub, A.J.: Generalized eigenproblem algorithms and software for algebraic Riccati equations. *Proc. IEEE* **72**, 1746–1754 (1984)
5. Bai, Z., Demmel, J., Gu, M.: An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numer. Math.* **76**(3), 279–308 (1997)
6. Bender, D.J., Laub, A.J.: The linear-quadratic optimal regulator for descriptor systems. *IEEE Trans. Automat. Control* **32**, 672–688 (1987)
7. Bender, D.J., Laub, A.J.: The linear-quadratic optimal regulator for descriptor systems: discrete-time case. *Automatica* **23**, 71–85 (1987)
8. Benner, P.: Ein orthogonal symplektischer Multishift Algorithmus zur Lösung der algebraischen Riccatigleichung. Diplomarbeit, RWTH Aachen, Institut für Geometrie und Praktische Mathematik, Aachen (1993) (in German)
9. Benner, P.: Contributions to the Numerical Solution of Algebraic Riccati Equations and Related Eigenvalue Problems. Logos-Verlag, Berlin (1997). Also: Dissertation, Fakultät für Mathematik, TU Chemnitz-Zwickau (1997)
10. Benner, P.: Computational methods for linear-quadratic optimization. *Supplemento ai Rendiconti del Circolo Matematico di Palermo, Serie II* **58**, 21–56 (1999)
11. Benner, P., Byers, R.: Disk functions and their relationship to the matrix sign function. In: *Proceedings of the European Control Conference (ECC 97)*, Brussels, Paper 936. BELWARE Information Technology, Waterloo (1997). CD-ROM

12. Benner, P., Byers, R., Barth, E.: Algorithm 800. fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices I: the square-reduced method. *ACM Trans. Math. Softw.* **26**(1), 49–77 (2000)
13. Benner, P., Byers, R., Losse, P., Mehrmann, V., Xu, H.: Robust formulas for H_∞ optimal controllers. *Automatica* **47**(12), 2639–2646 (2011)
14. Benner, P., Byers, R., Mehrmann, V., Xu, H.: Numerical methods for linear-quadratic and H_∞ control problems. In: Picci, G., Gilliam, D.S. (eds.) *Dynamical Systems, Control, Coding, Computer Vision: New Trends, Interfaces, and Interplay*. Progress in Systems and Control Theory, vol. 25, pp. 203–222. Birkhäuser, Basel (1999)
15. Benner, P., Byers, R., Mehrmann, V., Xu, H.: Numerical computation of deflating subspaces of skew-Hamiltonian/Hamiltonian pencils. *SIAM J. Matrix Anal. Appl.* **24**(1), 165–190 (2002)
16. Benner, P., Byers, R., Mehrmann, V., Xu, H.: Robust numerical methods for robust control. Preprint 2004–06, Institut für Mathematik, TU Berlin, D-10623 Berlin (2004). <http://www.math.tu-berlin.de/preprints>
17. Benner, P., Byers, R., Mehrmann, V., Xu, H.: A robust numerical method for the γ -iteration in \mathcal{H}_∞ control. *Linear Algebra Appl.* **425**(2–3), 548–570 (2007)
18. Benner, P., Ezzatti, P., Kressner, D., Quintana-Ortí, E.S., Remón, A.: A mixed-precision algorithm for the solution of Lyapunov equations on hybrid CPU-GPU platforms. *Parallel Comput.* **37**(8), 439–450 (2011). doi:10.1016/j.parco.2010.12.002
19. Benner, P., Kressner, D.: Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices II. *ACM Trans. Math. Softw.* **32**(2), 352–373 (2006)
20. Benner, P., Laub, A.J., Mehrmann, V.: A collection of benchmark examples for the numerical solution of algebraic Riccati equations I: continuous-time case. Technical report SPC 95_22, Fakultät für Mathematik, TU Chemnitz–Zwickau, 09107 Chemnitz (1995). Available from <http://www.tu-chemnitz.de/sfb393/spc95pr.html>
21. Benner, P., Laub, A.J., Mehrmann, V.: A collection of benchmark examples for the numerical solution of algebraic Riccati equations II: discrete-time case. Technical report SPC 95_23, Fakultät für Mathematik, TU Chemnitz–Zwickau, 09107 Chemnitz (1995). Available from <http://www.tu-chemnitz.de/sfb393/spc95pr.html>
22. Benner, P., Laub, A.J., Mehrmann, V.: Benchmarks for the numerical solution of algebraic Riccati equations. *IEEE Control Syst. Mag.* **7**(5), 18–28 (1997)
23. Benner, P., Li, J.R., Penzl, T.: Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems. *Numer. Linear Algebra Appl.* **15**(9), 755–777 (2008)
24. Benner, P., Losse, P., Mehrmann, V., Voigt, M.: Numerical solution of eigenvalue problems for alternating matrix polynomials and their application in control problems for descriptor systems. Preprint MPIMD/13-24, Max Planck Institute Magdeburg (2013). <http://www2.mpi-magdeburg.mpg.de/preprints/2013/MPIMD13-24.pdf>
25. Benner, P., Mehrmann, V.: Collections of benchmark examples in control. *WGS Newsl.* **10**, 7–8 (1996)
26. Benner, P., Mehrmann, V., Sima, V., Huffel, S.V., Varga, A.: SLICOT – a subroutine library in systems and control theory. In: Datta, B.N. (ed.) *Applied and Computational Control, Signals, and Circuits*, vol. 1, chap. 10, pp. 499–539. Birkhäuser, Boston (1999)
27. Benner, P., Mehrmann, V., Xu, H.: A new method for computing the stable invariant subspace of a real Hamiltonian matrix. *J. Comput. Appl. Math.* **86**, 17–43 (1997)
28. Benner, P., Mehrmann, V., Xu, H.: A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils. *Numer. Math.* **78**(3), 329–358 (1998)
29. Benner, P., Mena, H.: BDF methods for large-scale differential Riccati equations. In: De Moor, B., Motmans, B., Willems, J., Van Dooren, P., Blondel, V. (eds.) *Proceedings of 16th International Symposium on Mathematical Theory of Network and Systems (MTNS 2004)*, Leuven, 12p. (2004)
30. Benner, P., Mena, H.: Rosenbrock methods for solving Riccati differential equations. *IEEE Trans. Autom. Control* **58**(11), 2950–2957 (2013)

31. Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Solving linear-quadratic optimal control problems on parallel computers. *Optim. Methods Softw.* **23**(6), 879–909 (2008). doi:[10.1080/10556780802058721](https://doi.org/10.1080/10556780802058721)
32. Benner, P., Saak, J.: Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey. *GAMM Mitt.* **36**(1), 32–52 (2013). doi:[10.1002/gamm.201310003](https://doi.org/10.1002/gamm.201310003)
33. Benner, P., Sima, V., Voigt, M.: FORTRAN 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems – part I: algorithms and applications. Preprint MPIMD/13-11, Max Planck Institute Magdeburg (2013). Available from <http://www.mpi-magdeburg.mpg.de/preprints/2013/11/>
34. Benner, P., Sima, V., Voigt, M.: FORTRAN 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems – part II: implementation and numerical results. Preprint MPIMD/13-12, Max Planck Institute Magdeburg (2013). Available from <http://www.mpi-magdeburg.mpg.de/preprints/2013/12/>
35. Benner, P., Stykel, T.: Numerical solution of projected algebraic Riccati equations. *SIAM J. Numer. Anal.* **52**(2), 581–600 (2014). doi:[10.1137/130923993](https://doi.org/10.1137/130923993)
36. Bini, D.A., Iannazzo, B., Meini, B.: Numerical Solution of Algebraic Riccati Equations. No. 9 in *Fundamentals of Algorithms*. SIAM, Philadelphia (2012)
37. Bojanczyk, A., Golub, G.H., Van Dooren, P.: The periodic Schur decomposition. Algorithms and applications. In: Luk, F.T. (ed.) *Advanced Signal Processing Algorithms, Architectures, and Implementations III*, San Diego. Proceeding SPIE, vol. 1770, pp. 31–42 (1992)
38. Bunch, J.R.: The weak and strong stability of algorithms in numerical algebra. *Linear Algebra Appl.* **88**, 49–66 (1987)
39. Bunse-Gerstner, A.: Matrix factorization for symplectic QR-like methods. *Linear Algebra Appl.* **83**, 49–77 (1986)
40. Bunse-Gerstner, A., Byers, R., Mehrmann, V.: Numerical methods for algebraic Riccati equations. In: Bittanti, S. (ed.) *Proceeding Workshop on the Riccati Equation in Control, Systems, and Signals*, Como, pp. 107–116 (1989)
41. Bunse-Gerstner, A., Mehrmann, V.: A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation. *IEEE Trans. Autom. Control* **31**, 1104–1113 (1986)
42. Bunse-Gerstner, A., Mehrmann, V., Watkins, D.: An SR algorithm for Hamiltonian matrices based on Gaussian elimination. *Methods Oper. Res.* **58**, 339–358 (1989)
43. Byers, R.: Hamiltonian and symplectic algorithms for the algebraic Riccati equation. Ph.D. thesis, Cornell University, Department of Computer Science, Ithaca (1983)
44. Byers, R.: A Hamiltonian QR-algorithm. *SIAM J. Sci. Stat. Comput.* **7**, 212–229 (1986)
45. Byers, R.: Solving the algebraic Riccati equation with the matrix sign function. *Linear Algebra Appl.* **85**, 267–279 (1987)
46. Byers, R., He, C., Mehrmann, V.: The matrix sign function method and the computation of invariant subspaces. *SIAM J. Matrix Anal. Appl.* **18**(3), 615–632 (1997)
47. Byers, R., Mehrmann, V.: Symmetric updating of the solution of the algebraic Riccati equation. *Methods Oper. Res.* **54**, 117–125 (1985)
48. Choi, C., Laub, A.J.: Efficient matrix-valued algorithms for solving stiff Riccati differential equations. *IEEE Trans. Autom. Control* **35**, 770–776 (1990)
49. Chu, D., Liu, X., Mehrmann, V.: A numerical method for computing the Hamiltonian Schur form. *Numer. Math.* **105**(3), 375–412 (2006)
50. Chu, E.K.W., Fan, H.Y., Lin, W.W.: A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations. *Linear Algebra Appl.* **396**, 55–80 (2005)
51. Chu, E.K.W., Fan, H.Y., Lin, W.W., Wang, C.S.: Structure-preserving algorithms for periodic discrete-time algebraic Riccati equations. *Int. J. Control* **77**(8), 767–788 (2004)
52. Datta, B.N.: *Numerical Methods for Linear Control Systems*. Elsevier Academic, Amsterdam/Boston (2004)
53. Dieci, L.: Numerical integration of the differential Riccati equation and some related issues. *SIAM J. Numer. Anal.* **29**(3), 781–815 (1992)
54. Dorf, R.C.: *Modern Control Systems*, 2nd edn. Addison Wesley, Reading (1974)

55. Flaschka, U., Mehrmann, V., Zywiets, D.: An analysis of structure preserving methods for symplectic eigenvalue problems. *RAIRO Autom. Prod. Inform. Ind.* **25**, 165–190 (1991)
56. Golub, G.H., Van Loan, C.F.: *Matrix Computations*, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
57. Green, M., Limebeer, D.J.N.: *Linear Robust Control*. Prentice-Hall, Englewood Cliffs (1995)
58. Heiland, J.: Decoupling and optimization of differential-algebraic equations with application in flow control. Ph.D. thesis, Fakultät II – Mathematik und Naturwissenschaften, TU Berlin (2014)
59. Hench, J.J., Laub, A.J.: An extension of the QR algorithm for a sequence of matrices. Technical report CCEC-92-0829, ECE Department, University of California, Santa Barbara (1992)
60. Hench, J.J., Laub, A.J.: Numerical solution of the discrete-time periodic Riccati equation. *IEEE Trans. Autom. Control* **39**, 1197–1210 (1994)
61. Horn, R.A., Johnson, C.R.: *Topics in Matrix Analysis*. Cambridge University Press, Cambridge (1991)
62. Kunkel, P., Mehrmann, V.: Numerical solution of Riccati differential algebraic equations. *Linear Algebra Appl.* **137/138**, 39–66 (1990)
63. Kunkel, P., Mehrmann, V.: The linear quadratic control problem for linear descriptor systems with variable coefficients. *Math. Control Signals Syst.* **10**, 247–264 (1997)
64. Kunkel, P., Mehrmann, V.: Analysis of over- and underdetermined nonlinear differential-algebraic systems with application to nonlinear control problems. *Math. Control Signals Syst.* **14**, 233–256 (2001)
65. Kunkel, P., Mehrmann, V.: *Differential-Algebraic Equations: Analysis and Numerical Solution*. Textbooks in Mathematics. EMS Publishing House, Zürich (2006)
66. Kunkel, P., Mehrmann, V.: Optimal control for unstructured nonlinear differential-algebraic equations of arbitrary index. *Math. Control Signals Syst.* **20**(3), 227–269 (2008)
67. Kunkel, P., Mehrmann, V., Rath, W.: Analysis and numerical solution of control problems in descriptor form. *Math. Control Signals Syst.* **14**, 29–61 (2001)
68. Lancaster, P., Rodman, L.: *The Algebraic Riccati Equation*. Oxford University Press, Oxford (1995)
69. Lancaster, P., Tismenetsky, M.: *The Theory of Matrices*, 2nd edn. Academic, Orlando (1985)
70. Lang, N., Mena, H., Saak, J.: On the benefits of the LDL^T factorization for large-scale differential matrix equation solvers. Preprint MPIMD/14-14, Max Planck Institute Magdeburg (2014). Available from <http://www.mpi-magdeburg.mpg.de/preprints/>
71. Laub, A.J.: A Schur method for solving algebraic Riccati equations. *IEEE Trans. Autom. Control* **24**, 913–921 (1979)
72. Lin, W.W., Mehrmann, V., Xu, H.: Canonical forms for Hamiltonian and symplectic matrices and pencils. *Linear Algebra Appl.* **301–303**, 469–533 (1999)
73. Losse, P., Mehrmann, V., Poppe, L.K., Reis, T.: The modified optimal H_∞ control problem for descriptor systems. *SIAM J. Control Optim.* **47**(6), 2795–2811 (2008)
74. Mehrmann, V.: *The linear quadratic control problem: theory and numerical algorithms*. Habilitationsschrift, Universität Bielefeld, Bielefeld (1987)
75. Mehrmann, V.: A symplectic orthogonal method for single input or single output discrete time optimal linear quadratic control problems. *SIAM J. Matrix Anal. Appl.* **9**, 221–248 (1988)
76. Mehrmann, V.: *The Autonomous Linear Quadratic Control Problem. Theory and Numerical Solution*. Lecture Notes in Control and Information Sciences, vol. 163. Springer, Heidelberg (1991)
77. Mehrmann, V.: A step toward a unified treatment of continuous and discrete time control problems. *Linear Algebra Appl.* **241–243**, 749–779 (1996)
78. Mehrmann, V., Poloni, F.: An inverse-free ADI algorithm for computing Lagrangian invariant subspaces. Preprint 14-2014, Institut für Mathematik, TU Berlin (2014)
79. Mehrmann, V., Schröder, C., Watkins, D.: A new block method for computing the Hamiltonian Schur form. *Linear Algebra Appl.* **431**(3–4), 350–368 (2009)
80. Mehrmann, V., Tan, E.: Defect correction methods for the solution of algebraic Riccati equations. *IEEE Trans. Autom. Control* **33**, 695–698 (1988)

81. Mehrmann, V., Xu, H.: Explicit solutions for a Riccati equation from transport theory. *SIAM J. Matrix Anal. Appl.* **30**(4), 1339–1357 (2008)
82. Paige, C.C., Van Loan, C.F.: A Schur decomposition for Hamiltonian matrices. *Linear Algebra Appl.* **41**, 11–32 (1981)
83. Penzl, T.: Numerische Lösung großer Lyapunov-Gleichungen. Logos-Verlag, Berlin, Germany (1998). Dissertation, Fakultät für Mathematik, TU Chemnitz (1998)
84. Petkov, P.H., Konstantinov, M.M., Gu, D.W., Mehrmann, V.: Numerical solution of matrix Riccati equations: a comparison of six solvers. SLICOT Working Note SLWN1999–10, (1999). Available from www.slicot.org
85. Petkov, P.H., Konstantinov, M.M., Mehrmann, V.: DGRSVX and DMSRIC: Fortran 77 subroutines for solving continuous-time matrix algebraic Riccati equations with condition and accuracy estimates. Technical report SFB393/98-16, Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz (1998)
86. Reid, W.T.: *Riccati Differential Equations*. Academic, New York (1972)
87. Van Loan, C.F.: A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix. *Linear Algebra Appl.* **61**, 233–251 (1984)
88. Voigt, M.: On linear-quadratic optimal control and robustness of differential-algebraic systems. Ph.D. thesis, Faculty for Mathematics, Otto-von-Guericke University, Magdeburg (2015)
89. Watkins, D.S.: On the reduction of a Hamiltonian matrix to Hamiltonian Schur form. *Electron. Trans. Numer. Anal.* **23**, 141–157 (2006)
90. Zhou, K., Doyle, J.C., Glover, K.: *Robust and Optimal Control*. Prentice-Hall, Upper Saddle River (1996)