Chapter 5

Square-Root Arrays and Chandrasekhar Recursions

Using the observation that H^∞ filtering coincides with Kalman filtering in Krein space, in this chapter we develop square-root arrays and Chandrasekhar recursions for H^∞ filtering problems. These are the generalizations of the conventional square-root arrays and Chandrasekhar recursions to the Krein space setting. The H^∞ square-root algorithms involve propagating the indefinite square-root of the quantities of interest and have the property that the appropriate inertia of these quantities is preserved. For systems that are constant, or whose time-variation is structured in a certain way, the Chandrasekhar recursions allow a reduction in the computational effort per iteration from $O(n^3)$ to $O(n^2)$, where n is the number of states. The H^∞ square-root and Chandrasekhar recursions both have the interesting feature that one does not need to explicitly check for the positivity conditions required of the H^∞ filters. These conditions are built into the algorithms themselves so that an H^∞ estimator of the desired level exists if, and only if, the algorithms can be executed. [The results of this chapter were first reported in [HSK94c].]

5.1 Introduction

In Chapter 2 we presented a self-contained theory for linear estimation in Krein spaces with the objective of unifying the H^2, H^∞, risk-sensitive and game-theoretic approaches to estimation and control. There we claimed that the major bonus of this approach is that, apart from rather more transparent derivations of existing results (as done in Chapters 3 and 4), it shows a way to apply to the H^∞ (and these other) settings many of the results developed for Kalman filtering and LQG control over the last three decades. In this chapter we intend to support this claim by showing how the Krein space approach can be used to naturally extend the (numerically superior) square-root and (fast) Chandrasekhar array recursions of conventional Kalman filtering to H^∞ filtering.

The (so-called) square-root array algorithms were devised in the late 1960's [DM69, KBS71, Har72], and, for several reasons, are currently more often used to implement the conventional Kalman filter. These algorithms are closely related to the (so-called) QR method for solving systems of linear equations [BG66, GL89, HJ90, Str93], and have the properties of better conditioning, reduced dynamical range, and the use of orthogonal transformations, which typically leads to stabler algorithms.

Furthermore for constant systems, or in fact for systems where the time-variation is structured in a certain way, the Riccati recursions and the square-root recursions, both of which take $O(n^3)$ elementary computations (flops) per iteration (where n is the dimension of the state-space), can be replaced by more efficient recursions, which require only $O(n^2)$ flops per iteration [Kai72, MSK74, SK94a]. These algorithms are analogous to certain equations invented in 1943-47 by the astrophysicists Ambartsumian [Amb43] and Chandrasekhar [Cha48] (hence the name Chandrasekhar recursions), are also closely related to the concept of displacement structure [KKM79, Say92].

One immediate fall-out of our observation that H^∞ filtering coincides with Kalman filtering in Krein space is that it allows us to generalize these square-root arrays and Chandrasekhar recursions to the H^∞ setting. Both these algorithms involve propagating (indefinite) square-roots of the quantities of interest and guarantee that the proper inertia of these quantities is preserved. Furthermore, the condition required for the existence of the H^∞ filters is built into the algorithms — if the algorithms can be carried out, then an H^∞ filter of the desired level exists, and if they cannot be executed then such H^∞ filters do not exist. This can be a significant simplification of the existing algorithms.

The remainder of the chapter is organized as follows.

The conventional square-root array algorithms are introduced in Sec. 5.2 along with some of their properties. In Sec. 5.3 we begin the development of the H^∞ square-root array algorithms and mention why they are natural extensions of their conventional counterparts. We initially encounter some difficulties in generalizing these arrays to the Krein space setting, and in order to alleviate them we then introduce the concept of indefinite square-roots, and study the inertia properties of the Gramian matrices in the H^∞ filtering problem in some detail. These inertia properties are related to the triangularization of matrices via J-unitary transformations and will be crucial for the development of the H^∞ square-root array and Chandrasekhar recursions. Finally, the general form of the H^∞ a posteriori and a priori filters are given in Sec. 5.3.1 and the central filters in Sec. 5.3.2.

The conventional (fast) Chandrasekhar recursions, along with several of their properties, is given in Sec. 5.4. Sec. 5.5.1 extends these recursions to the H^∞ setting, and Sec. 5.5.2 gives the corresponding central H^∞ filters.

In closing this introduction we note that there are many variations to the conventional square-root arrays and Chandrasekhar recursions, of which only a few have been considered here. However, the approach adopted here is of sufficient generality that it should allow a reader to extend any other variation of these algorithms to the H^∞ setting.

A brief remark on the notation used in this chapter. To avoid confusion between the various gain vectors used in this chapter, we shall employ the following convention: $K_{p,i}$ will denote the gain vector in the usual Krein space (or Hilbert space) Kalman filter, $K_{f,i}$ the gain vector in the filtered form of the Krein space Kalman filter, and $K_{s,i}$ and $K_{a,i}$ will denote the gain vectors in the H^∞ a posteriori and a priori filters, respectively.

5.2 H^2 Square-Root Array Algorithms

In state-space estimation problems we begin with a (possibly) time-varying state-space model of the form

$$\begin{cases} x_{j+1} &= F_j x_j + G_j u_j, \quad x_0 \\ y_j &= H_j x_j + v_j \end{cases} \tag{5.2.1}$$

where the $\{u_j, v_j\}$ are disturbances whose nature depends on the criterion being used, and where the $\{y_j\}$ are the observed outputs. We shall be, typically, interested in obtaining estimates of some given linear combination of the states, say $s_i \stackrel{\Delta}{=} L_i x_i$, and, most frequently, filtered and predicted estimates, denoted by $\hat{s}_{j|j}$ and $\hat{s}_j$, respectively, that each use the observations $\{y_k, k \leq j\}$ and $\{y_k, k < j\}$.

In conventional Kalman filtering the $\{x_0, u_j, v_j\}$ are assumed to be zero-mean random variables with

$$E \begin{bmatrix} x_0 \\ u_i \\ v_i \end{bmatrix} \begin{bmatrix} x_0^* & u_j^* & v_j^* \end{bmatrix} = \begin{bmatrix} \Pi_0 & 0 & 0 \\ 0 & Q_i \delta_{ij} & \\ 0 & 0 & R_i \delta_{ij} \end{bmatrix} \geq 0.$$

Moreover, the output covariance of (5.2.1) is assumed to be positive-definite, *i.e.* $R_y > 0$, where $[R_y]_{ij} = E y_i y_j^*$.[1]

Using the H^2 criterion, the predicted and filtered estimates are given by $\hat{s}_j = L_j \hat{x}_j$ and $\hat{s}_{j|j} = L_j \hat{x}_{j|j}$, respectively, where $\hat{x}_j$ satisfies the predicted form of the conventional Kalman filter recursions

$$\hat{x}_{j+1} = F_j \hat{x}_j + K_{p,j}(y_j - H_j \hat{x}_j), \quad \hat{x}_0 = 0 \tag{5.2.2}$$

and $\hat{x}_{j|j}$ satisfies its filtered form,

$$\hat{x}_{j+1|j+1} = F_j \hat{x}_{j|j} + K_{f,j+1}(y_{j+1} - H_{j+1} F_j \hat{x}_{j|j}), \quad \hat{x}_{-1|-1} = 0 \tag{5.2.3}$$

[$\hat{x}_j$ denotes the predicted estimate of x_j, given $\{y_0, \ldots, y_{j-1}\}$, and $\hat{x}_{j|j}$ denotes its filtered estimate, given $\{y_0, \ldots, y_j\}$.] The gain vectors $K_{p,j}$ and $K_{f,j}$ can be computed

[1] One way to ensure the positive definiteness of the output covariance, R_y, is to assume that the measurement noise covariance matrix is full rank, *i.e.*, $R_i > 0$. This is often a very reasonable assumption.

in several ways. The most common method uses a certain Riccati recursion, viz.,

$$K_{f,j+1} = P_{j+1}H_{j+1}R_{e,j+1}^{-1} \quad , \quad K_{p,j} = F_j K_{f,j-1} \quad , \quad R_{e,j} = R_j + H_j P_j H_j^* \tag{5.2.4}$$

where P_j satisfies the Riccati recursion,

$$P_{j+1} = F_j P_j F_j^* + G_j Q_j G_j^* - K_{p,j} R_{e,j} K_{p,j}^*, \quad P_0 = \Pi_0. \tag{5.2.5}$$

[The invertibility of the $R_{e,j}$ is guaranteed by the positivity assumption on R_y.]

The matrix P_j appearing in this Riccati recursion has the physical meaning of being the variance of the state prediction error, $\tilde{x}_j = x_j - \hat{x}_j$, and therefore has to be positive (semi)definite. Round-off errors can cause a loss of positive-definiteness, thus throwing all the obtained results into doubt. For this, and other reasons (reduced dynamic range, better conditioning, stabler algorithms, etc.) attention has moved in the Kalman filtering community to the so-called square-root array (or factorized) estimation algorithms [DM69, KBS71] that propagate square-root factors of P_j, *i.e.* a matrix, $P_j^{1/2}$ say, with positive diagonal entries, and such that

$$P_j = P_j^{1/2}(P_j^{1/2})^* = P_j^{1/2}P_j^{*/2}.$$

Square roots can be similarly defined for the system covariances $\{Q_j, R_j\}$. Then it is in fact not hard to show the following.

Find any orthogonal transformation, say Θ_j,[2] that triangularizes the pre-array shown below

$$\begin{bmatrix} R_i^{1/2} & H_j P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j Q_j^{1/2} \end{bmatrix} \Theta_j = \begin{bmatrix} X & 0 & 0 \\ Y & Z & 0 \end{bmatrix}. \tag{5.2.6}$$

The resulting post-array entries can be checked, by taking squares and using the orthogonality of Θ_j, to obey

$$\begin{aligned} XX^* &= R_j + H_j P_j H_j^* = R_{e,j} \\ YX^* &= F_j P_j H_j^* \\ ZZ^* &= F_j P_j F_j^* + G_j Q_j G_j^* - YY^* \\ &= F_j P_j F_j^* + G_j Q_j G_j^* - YX^*(XX^*)^{-1}XY^* \end{aligned}$$

[2]By an orthogonal transformation, Θ, we mean one for which, $\Theta\Theta^* = \Theta^*\Theta = I$.

$$\begin{aligned} &= F_jP_jF_j^* + G_jQ_jG_j^* - F_jP_jH_j^*R_{e,j}^{-1}H_jP_jF_j^* \\ &= P_{j+1}. \end{aligned}$$

Therefore we can identify

$$Z = P_{j+1}^{1/2} \quad , \quad X = R_{e,j}^{1/2} \tag{5.2.7}$$

and also

$$Y = F_jP_jH_j^*R_{e,j}^{-*/2} = K_{p,j}R_{e,j}^{1/2}. \tag{5.2.8}$$

Thus the square-root algorithm not only propagates the square-roots of the Riccati variable, P_j, but also gives us quantities useful for the state estimation recursion

$$\hat{x}_{j+1} = F_j\hat{x}_j + K_{p,j}R_{e,j}^{-1/2}(y_j - H_j\hat{x}_j).$$

The unitary transformation Θ_j is highly nonunique and can be computed in many ways, the simplest ones being to construct it as a sequence of elementary (Givens or plane) rotations nulling one entry at a time in the pre-array, or as a sequence of elementary (Householder) reflections nulling out a block of entries in each row. We refer to [Hou53, GL89, Str93] for more details.[3] The numerical advantages of the square-root transformations arise from the length preserving properties of unitary transformations, and from the fact that the dynamic range of the entries in $P_j^{1/2}$ is roughly the square-root of the dynamic range of those in P_j. Moreover, regular computational (systolic) arrays can be designed to implement sequences of elementary unitary transformations [ML92].

A final result will be useful before we summarize the above discussion in a theorem. Any unitary transformation Θ_j that triangularizes the pre-array in (5.2.6) also gives the, readily checked, identity,

$$\begin{bmatrix} -R_j^{-1/2}y_j & P_j^{-1/2}\hat{x}_j & 0 \end{bmatrix}\Theta_j = \begin{bmatrix} R_{e,j}^{-1/2}e_j & P_{j+1}^{-1/2}\hat{x}_{j+1} & \times \end{bmatrix} \tag{5.2.9}$$

where $\times$ denotes an entry whose exact form is not relevant at the moment.

We can summarize the above discussion as follows.

[3]The above square-root method is closely related to the QR (factorization) method for solving systems of linear equations.

Algorithm 5.2.1 (Conventional Square-Root Algorithm) *The gain vector $K_{p,j}$ necessary to obtain the state estimates in the conventional Kalman filter*

$$\hat{x}_{j+1} = F_j\hat{x}_j + K_{p,j}(y_j - H_j\hat{x}_j), \quad \hat{x}_0 = 0,$$

can be updated as follows

$$\begin{bmatrix} R_j^{1/2} & H_jP_j^{1/2} & 0 \\ 0 & F_jP_j^{1/2} & G_jQ_j^{1/2} \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j}^{1/2} & 0 & 0 \\ K_{p,j}R_{e,j}^{1/2} & P_{j+1}^{1/2} & 0 \end{bmatrix}, \tag{5.2.10}$$

where Θ_j is any unitary matrix that triangularizes the above pre-array. The algorithm is initialized with $P_0 = \Pi_0$.

Note that the quantities necessary to update the square-root array, and to calculate the state estimates, may all be found from the triangularized post-array.

It will also be useful to quote the filtered form of the square-root array algorithm, that can be verified in a fashion similar to what was done above.

Algorithm 5.2.2 (Conventional Square-Root Algorithm - Filtered Form) *The gain vector $K_{f,j}$ necessary to obtain the state estimates in the filtered form of the conventional Kalman filter*

$$\hat{x}_{j+1|j+1} = F_j\hat{x}_{j|j} + K_{f,j+1}(y_{j+1} - H_{j+1}F_j\hat{x}_{j|j}), \quad \hat{x}_{-1|-1} = 0,$$

can be updated as follows

$$\begin{bmatrix} R_j^{1/2} & H_jP_j^{1/2} \\ 0 & P_j^{1/2} \end{bmatrix} \Theta_j^{(1)} = \begin{bmatrix} R_{e,j}^{1/2} & 0 \\ K_{f,j}R_{e,j}^{1/2} & P_{j|j}^{1/2} \end{bmatrix}, \tag{5.2.11}$$

$$\begin{bmatrix} F_jP_{j|j}^{1/2} & G_jQ_j^{1/2} \end{bmatrix} \Theta_j^{(2)} = \begin{bmatrix} P_{j+1}^{1/2} & 0 \end{bmatrix} \tag{5.2.12}$$

where $\Theta_j^{(1)}$ and $\Theta_j^{(2)}$ are any unitary matrices that triangularize the above pre-arrays. The algorithm is initialized with $P_0 = \Pi_0$.

5.3 H^∞ Square-Root Array Algorithms

We now turn our attention to H^∞ filtering. Our goal here is to investigate whether it is possible to construct square-root array implementations of H^∞ filters, similar to what was done in the aforementioned H^2 case.

5.3.1 The General Case

Recall the parametrization of all possible H^∞ a posteriori filters, $\check{s}_{j|j} = \mathcal{F}_{f,j}(y_0, \ldots y_j)$, as given by Lemma 3.3.5. There we saw that all such $\check{s}_{j|j}$ are given by any choices that render

$$\sum_{j=0}^{k} \begin{bmatrix} y_j - H_j\hat{x}_j \\ \check{s}_{j|j} - L_j\hat{x}_j \end{bmatrix}^* R_{e,j}^{-1} \begin{bmatrix} y_j - H_j\hat{x}_j \\ \check{s}_{j|j} - L_j\hat{x}_j \end{bmatrix} \geq 0, \quad 0 \leq k \leq i \tag{5.3.1}$$

where $\hat{x}_j$ satisfies the recursion,

$$\hat{x}_{j+1} = F_j\hat{x}_j + K_{p,j} \begin{bmatrix} y_j - H_j\hat{x}_j \\ \check{s}_{j|j} - L_j\hat{x}_j \end{bmatrix}, \quad \hat{x}_0 = 0 \tag{5.3.2}$$

with

$$K_{p,j} = F_jP_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} R_{e,j}^{-1} \quad \text{and} \quad R_{e,j} = \begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix} + \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} \tag{5.3.3}$$

and

$$P_{j+1} = F_jP_jF_j^* + G_jQ_jG_j^* - K_{p,j}R_{e,j}K_{p,j}^*, \quad P_0 = \Pi_0. \tag{5.3.4}$$

As repeatedly mentioned earlier, the above solution looks very much like the conventional Kalman filter. The essential difference is in the Riccati recursion where we now have indefinite covariance matrices (such as $R_{e,i}$). Nonetheless, let us persist to see whether we can come up with a square-root implementation of the H^∞ Riccati recursion (5.3.4).

To this end, recall the conventional square-root array algorithm of Sec. 5.2,

$$\begin{bmatrix} R_j^{1/2} & H_jP_j^{1/2} & 0 \\ 0 & F_jP_j^{1/2} & G_jQ_j^{1/2} \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j}^{1/2} & 0 & 0 \\ K_{p,j}R_{e,j}^{1/2} & P_{j+1}^{1/2} & 0 \end{bmatrix}, \tag{5.3.5}$$

where $P_j^{1/2}P_j^{*/2} = P_j$, $Q_j^{1/2}Q_j^{*/2} = Q_j$, $R_j^{1/2}R_j^{*/2} = R_j$, and $R_{e,j}^{1/2}R_{e,j}^{*/2} = R_{e,j}$.

Using our earlier observation that H^∞ filtering is just Kalman filtering in Krein space, allows us to speculate that needs to be done is to come up with a Krein space generalization of the above square-root array algorithm. This is the approach that we shall take here.

The first problem that occurs if one wants to extend the square-root array algorithm to the Krein space setting (of which the H^∞ filtering problem is a special case) is that the matrices R_i, Q_i, P_i and $R_{e.i}$ are in general indefinite and square-roots may not exist. To alleviate this problem we need the notion of an indefinite square-root, as defined below.

Definition 5.3.1 (Indefinite Square-Roots) *Suppose A is an arbitrary Hermitian matrix. $A^{1/2}$ will be called an indefinite square-root of A if, and only, if*

$$A = A^{1/2} S A^{*/2}$$

*where S is a signature matrix (*i.e.* a diagonal matrix with diagonal elements either $+1$ or -1).*

However, in the Krein space case R_i, Q_i, P_i and $R_{e.i}$ may all have arbitrary inertia, *i.e.*

$$R_i = R_i^{1/2} S_i^{(1)} R_i^{*/2}, \quad Q_i = Q_i^{1/2} S_i^{(2)} Q_i^{*/2}, \quad P_i = P_i^{1/2} S_i^{(3)} P_i^{*/2}, \quad R_{e,i} = R_{e,i}^{1/2} S_i^{(4)} R_{e,i}^{*/2}$$

for arbitrary signature matrices $S_i^{(k)}$, $k = 1, 2, 3, 4$. It is thus not obvious how to incorporate all these different time-varying signature matrices into a square-root array algorithm of the type (5.3.5). Although this can be done in the general case, by either introducing non-Hermitian factorizations of the Gramians, or by keeping track of the inertia, we shall not pursue these lines of thought here. The reason is that, as it turns out, in the H^∞ problems that we have been studying, the Gramians satisfy certain inertia properties that allow us to extend the algorithm of (5.3.5) in a very natural way.

Indeed when a solution to the H^∞ filtering problem exists, we know that $P_i \geq 0$, and that $R_{e,i}$ and R_i have the same inertia (see *e.g.*, Lemmas 2.7.3 and 2.7.4). Moreover, $Q_i = I_m > 0$ and $R_i = \begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix}$ have constant inertia (and thus so does $R_{e,i}$) so that we may write,

$$R_i = R_i^{1/2} J R_i^{*/2}, \quad Q_i = Q_i^{1/2} Q_i^{*/2}, \quad P_i = P_i^{1/2} P_i^{*/2}, \quad R_{e,i} = R_{e,i}^{1/2} J R_{e,i}^{*/2} \quad (5.3.6)$$

with

$$R_i^{1/2} = \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} \quad \text{and} \quad J = \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}. \tag{5.3.7}$$

This suggests that in the H^∞ filtering problem, the prearray in (5.3.5) should be replaced by,

$$\begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{bmatrix}. \tag{5.3.8}$$

[Recall that in H^∞ estimation H_j is replaced by $\begin{bmatrix} H_j \\ L_j \end{bmatrix}$.] Now in the H^2 case the prearray in (5.3.5) was triangularized by a unitary transformation (or simply by a rotation). Since the H^2 estimation problem can be formulated in a Hilbert space, whereas the H^∞ estimation problem is most naturally formulated in a Krein space, it seems plausible that we should attempt to triangularize (5.3.8), not by an ordinary rotation, but by a hyperbolic rotation. To be more specific, we need to use a J-unitary transformation, as defined below.

Definition 5.3.2 (J-unitary Matrices) *For any signature matrix, J, (a diagonal matrix with $+1$ and -1 diagonal elements) the matrix Θ will be called J-unitary if*

$$\Theta J \Theta^* = J. \tag{5.3.9}$$

Recall that unitary transformations (or ordinary rotations) preserve the length (or ordinary norm) of vectors. J-unitary transformations, on the other hand, preserve the (indefinite) J-norm of vectors. Indeed, if $b = a\Theta$, with Θ J-unitary, then

$$bJb^* = a\Theta J \Theta^* a^* = aJa^*.$$

The above discussions suggest that we should attempt to triangularize (5.3.8) via a J-unitary transformation, where

$$\begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} & & \\ & I_n & \\ & & I_m \end{bmatrix}. \tag{5.3.10}$$

Now it is well known that it is always possible to triangularize arrays using unitary transformations. But is this also true of J-unitary transformations? To see if this is the case, consider a much simpler example where we are given the (two-element) row vector

$$\begin{bmatrix} a & b \end{bmatrix},$$

and are asked to hyperbolically rotate it to that the resulting vector lies along the direction of the x-axis.[4] For the time being, assume that such a transformation can be found. Then we can write,

$$\begin{bmatrix} a & b \end{bmatrix} \Theta = \begin{bmatrix} c & 0 \end{bmatrix}, \tag{5.3.11}$$

where

$$\Theta J \Theta^* = J \quad \text{and} \quad J = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix}. \tag{5.3.12}$$

Since Θ is J-unitary this implies that,

$$\begin{bmatrix} a & b \end{bmatrix} J \begin{bmatrix} a^* \\ b^* \end{bmatrix} = \begin{bmatrix} c & 0 \end{bmatrix} J \begin{bmatrix} c^* \\ 0 \end{bmatrix}, \tag{5.3.13}$$

or more explicitly,

$$|a|^2 - |b|^2 = |c|^2 \geq 0. \tag{5.3.14}$$

Thus, $\begin{bmatrix} a & b \end{bmatrix}$ must have nonnegative J-norm. In other words, if the given $\begin{bmatrix} a & b \end{bmatrix}$ has negative J-norm (*i.e.*, $|a|^2 - |b|^2 < 0$) then it is impossible to hyperbolically rotate it to lie along the x-axis. [This fact is shown in Fig. 5.1. As can be seen, standard rotations move the vector $\begin{bmatrix} a & b \end{bmatrix}$ along the circle, $a^2 + b^2 =$ constant, whereas hyperbolic rotations move it along the hyperbola, $a^2 - b^2 =$ constant. Thus while it is always possible to rotate $\begin{bmatrix} a & b \end{bmatrix}$ to lie along the x-axis, if $|a|^2 - |b|^2 < 0$ then it is impossible to do so with a *hyperbolical* rotation. Indeed hyperbolic rotations cannot move vectors from the positive to negative subspaces of a Krein space, or vice versa.]

Thus it is quite obvious that it is not always possible to triangularize arrays using J-unitary transformations. The precise condition follows.

[4] Note in standard (two-dimensional) Euclidean space this can always be done.

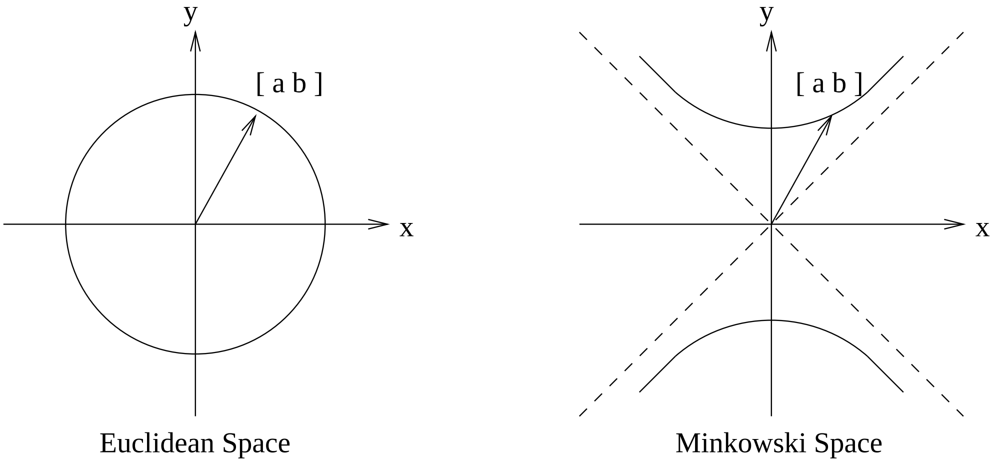

Figure 5.1: Standard rotations vs. hyperbolic rotations.

Lemma 5.3.1 (J-unitary Matrices and Triangularization) *Let A and B be arbitrary $n \times n$ and $n \times m$ matrices, respectively, and suppose $J = \begin{bmatrix} S_1 & \\ & S_2 \end{bmatrix}$ where S_1 and S_2 are $n \times n$ and $m \times m$ signature matrices. Then $\begin{bmatrix} A & B \end{bmatrix}$ can be triangularized by a J-unitary transformation Θ as*

$$\begin{bmatrix} A & B \end{bmatrix} \Theta = \begin{bmatrix} L & 0 \end{bmatrix}$$

with L lower triangular, if and only if, all leading submatrices of

$$S_1 \quad \text{and of} \quad AS_1A^* + BS_2B^*$$

have the same inertia.

Proof: To prove one direction suppose there exists a J-unitary transformation Θ that triangularizes $\begin{bmatrix} A & B \end{bmatrix}$. Consider an arbitrary partitioning of A, B and L, *i.e.*

$$\begin{bmatrix} A^{(1)} & B^{(1)} \\ A^{(2)} & B^{(2)} \end{bmatrix} \quad , \quad \begin{bmatrix} L^{(1)} & 0 \\ L^{(2)} & 0 \end{bmatrix}$$

where $A^{(1)}$, $B^{(1)}$ and $L^{(1)}$ have r rows. Now

$$\begin{bmatrix} A^{(1)} & B^{(1)} \\ A^{(2)} & B^{(2)} \end{bmatrix} \underbrace{\Theta J \Theta^*}_{=J} \begin{bmatrix} A^{(1)*} & A^{(2)*} \\ B^{(1)*} & B^{(2)*} \end{bmatrix} = \begin{bmatrix} L^{(1)} & 0 \\ L^{(2)} & 0 \end{bmatrix} J \begin{bmatrix} L^{(1)*} & L^{(2)*} \\ 0 & 0 \end{bmatrix}$$

so that

$$\begin{bmatrix} A^{(1)}S_1A^{(1)*} + B^{(1)}S_2B^{(1)*} & \times \\ \times & \times \end{bmatrix} = \begin{bmatrix} L^{(1)}S_1L^{(1)*} & \times \\ \times & \times \end{bmatrix} \tag{5.3.15}$$

where $\times$ denotes irrelevant entries. Moreover since L is lower triangular we have $L^{(1)} = \begin{bmatrix} L^{(11)} & 0 \end{bmatrix}$, where $L^{(11)}$ is lower triangular and $r \times r$. Thus if we denote by $S_1^{(1)}$ the leading $r \times r$ submatrix of S_1, equating the $(1,1)$ block entries in (5.3.15) yields

$$A^{(1)}S_1A^{(1)*} + B^{(1)}S_2B^{(1)*} = L^{(11)}S_1^{(1)}L^{(11)*} \tag{5.3.16}$$

The LHS of the above equation is the leading $r \times r$ submatrix of $AS_1A^* + BS_2B^*$. Thus (5.3.16) shows that the leading $r \times r$ submatrices of $AS_1A^* + BS_2B^*$ and S_1 have the same inertia. Since r was arbitrary the same is true for all leading submatrices.

To prove the other direction we assume that all leading submatrices of $AS_1A^* + BS_2B^*$ and S_1 have the same inertia. In particular the leading $(1,1)$ submatrices, so that

$$aS_1a^* + bS_2b^* = l_{11}^2 s \tag{5.3.17}$$

where a and b are the leading rows of A and B, s is the leading diagonal of S_1 and l_{11} is a scalar. Now define the vector

$$v = \begin{bmatrix} a & b \end{bmatrix} + l_{11}e_1$$

where $e_1 = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}$ is the first unit *row* vector. Consider the matrix

$$\Theta_1 = I - 2\frac{Jv^*v}{vJv^*}.$$

A straightforward calculation shows that $\Theta_1 J \Theta_1^* = J$ so that Θ_1 is J-unitary. Moreover another direct calculation shows that

$$\begin{bmatrix} a & b \end{bmatrix}\Theta_1 = l_{11}e_1 = \begin{bmatrix} l_{11} & 0 & \ldots & 0 \end{bmatrix}.$$

[Θ_1 may be recognized as an elementary Householder reflection in the J-metric.] We thus far have

$$\begin{bmatrix} A & B \end{bmatrix}\Theta_1 = \begin{bmatrix} \begin{bmatrix} l_{11} & 0 \end{bmatrix} & 0 \\ \begin{bmatrix} A_{21} & A_2 \end{bmatrix} & B_2 \end{bmatrix}. \tag{5.3.18}$$

Now if all leading submatrices of two given matrices have the same inertia, then their $(1,1)$ entries should have the same inertia and all leading submatrices of the *Schur complement* of their $(1,1)$ entries should have the same inertia. Now partition S_1 as

$$S_1 = \begin{bmatrix} s & \\ & S^{(1)} \end{bmatrix}$$

so that $S^{(1)}$ is the Schur complement of s in S_1. Likewise the Schur complement of the $(1,1)$ entry of $AS_1A^* + BS_2B^*$ is $A_2S^{(1)}A_2^* + B_2S_2B_2^*$ where A_2 and B_2 are defined in (5.3.18). Therefore all leading submatrices of $A_2S^{(1)}A_2^* + B_2S_2B_2^*$ and $S^{(1)}$ have the same inertia.We may now proceed as before and find a J-unitary matrix Θ_2 that rotates the first row of $\begin{bmatrix} A_{21} & A_2 & B_2 \end{bmatrix}$ to lie along the *second* unit vector. Continuing in a similar fashion will result in a J-unitary matrix $\Theta = \Theta_1\Theta_2\ldots\Theta_{n-1}$ that triangularizes $\begin{bmatrix} A & B \end{bmatrix}$.

■

Let us now apply the result of Lemma 5.3.1 to the triangularization of the array (5.3.8) using a J-unitary rotation with J given by (5.3.10). In fact, we need only consider the condition for the triangularization of the first block row (since setting the block $(2,3)$ entry of the post array to be zero can always be done via a standard unitary transformation). Thus we need only consider triangularizing

$$\begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{1/2} \end{bmatrix}, \tag{5.3.19}$$

using a J-unitary transformation with

$$J = \begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} & \\ & I_n \end{bmatrix}. \tag{5.3.20}$$

From Lemma 5.3.1, the condition obviously is that all leading submatrices of J and

$$\underbrace{\begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix}}_{R_j} + \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{1/2} P_j^{*/2} \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} = R_{e,j}, \tag{5.3.21}$$

have the same inertia. But this is precisely the condition required for the existence of an H^∞ a posteriori filter! (See Lemma 3.3.3 and Corollary 3.3.1). This result is quite amazing — it states that an H^∞ (a posteriori) filter exists if, and only if, the prearray can be triangularized, *i.e.*, if, and only if, the square-root algorithm can be performed and does not break down.

This observation should now reinforce our conviction that the most natural way of studying H^∞ problems is through the geometry of indefinite metric spaces. As just seen, this approach allowed us to immediately generalize the form of the conventional square-root array algorithm to the H^∞ setting (by introducing indefinite square-roots). Most importantly, once this was done, the conditions for triangularizing arrays, as forced upon us by the geometry of Krein spaces, directly led us to the conditions for the existence of H^∞ filters.[5] Thus in square-root implementations of H^∞ filters the existence conditions are built into the algorithms themselves, so that there is no need to check for them separately.

Now that we have settled the existence question, let us return to triangularizing the prearray (5.3.8), so that we can write,

$$\begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{bmatrix} \Theta_j = \begin{bmatrix} A & 0 & 0 \\ B & C & 0 \end{bmatrix}, \tag{5.3.22}$$

where A and C are lower triangular, and where Θ_j is J-unitary with J as in (5.3.10). The array on the left hand side of (5.3.22) is referred to as the *pre-array* and the array on the right hand side as the *post-array*. To identify the elements A, B and C in the post array let us square both sides of (5.3.22) and use the fact that Θ_j is J-unitary. Therefore

$$\begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{bmatrix} \underbrace{\Theta_j J \Theta_j^*}_{=J} \begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} & 0 \\ P_j^{*/2} \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} & P_j^{*/2} F_j^* \\ 0 & G_j^* \end{bmatrix}$$

[5] This phenomenon will occur again and again in this thesis. We shall shortly encounter it once more when we study the (fast) Chandrasekhar recursions.

$$= \begin{bmatrix} A & 0 & 0 \\ B & C & 0 \end{bmatrix} J \begin{bmatrix} A^* & B^* \\ 0 & C^* \\ 0 & 0 \end{bmatrix}. \tag{5.3.23}$$

Equating the $(1,1)$ blocks on the left hand and right hand sides of (5.3.23) yields

$$\begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix} + \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} = A \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} A^*.$$

The left hand side of the above relation is simply $R_{e,j}$. Therefore A is an indefinite square-root of the $R_{e,j}$, and we can write

$$A = R_{e,j}^{1/2} \quad , \quad R_{e,j}^{1/2} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} R_{e,j}^{*/2} = R_{e,j}. \tag{5.3.24}$$

Equating the $(2,1)$ blocks on the left hand and right hand sides of (5.3.23) yields

$$F_j P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} = B \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} A^*.$$

Therefore

$$B = F_j P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} R_{e,j}^{-*/2} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix},$$

so that we can write

$$B = \bar{K}_{p,j} = K_{p,j} R_{e,j}^{1/2}. \tag{5.3.25}$$

Equating the $(2,2)$ blocks on the left hand and right hand sides of (5.3.23) yields

$$\begin{aligned} F_j P_j F_j^* + G_j G_j^* &= B \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} B^* + CC^* \\ &= K_{p,j} R_{e,j}^{1/2} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} R_{e,j}^{*/2} K_{p,j}^* + CC^* \\ &= K_{p,j} R_{e,j} K_{p,j}^* + CC^*. \end{aligned}$$

Therefore

$$CC^* = F_j P_j F_j^* + G_j G_j^* - K_{p,j} R_{e,j} K_{p,j}^* = P_{j+1},$$

so that we may write

$$C = P_{j+1}^{1/2} \quad , \quad P_{j+1}^{1/2} P_{j+1}^{*/2} = P_{j+1}. \tag{5.3.26}$$

We can now summarize our results as follows.

Theorem 5.3.1 (H^∞ A Posteriori Square-Root Algorithm) *The H^∞ a posteriori filtering problem with level γ has a solution if, and only if, for all $j = 0, \ldots, i$ there exist J-unitary matrices (with J given by (5.3.10)), Θ_j, such that*

$$\begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j}^{1/2} & 0 & 0 \\ K_{p,j} R_{e,j}^{1/2} & P_{j+1}^{1/2} & 0 \end{bmatrix} \tag{5.3.27}$$

where the algorithm is initialized with, $P_0 = \Pi_0$. If this is the case, then all possible H^∞ a posteriori filters, $\check{s}_{j|j} = \mathcal{F}_{f,j}(y_0, \ldots y_j)$, are given by any choices that yield,

$$\sum_{j=0}^{k} \begin{bmatrix} y_j - H_j \hat{x}_j \\ \check{s}_{j|j} - L_j \hat{x}_j \end{bmatrix}^* R_{e,j}^{-1} \begin{bmatrix} y_j - H_j \hat{x}_j \\ \check{s}_{j|j} - L_j \hat{x}_j \end{bmatrix} \geq 0, \quad 0 \leq k \leq i$$

where $\hat{x}_j$ satisfies the recursion,

$$\hat{x}_{j+1} = F_j \hat{x}_j + K_{p,j} \begin{bmatrix} y_j - H_j \hat{x}_j \\ \check{s}_{j|j} - L_j \hat{x}_j \end{bmatrix}, \quad \hat{x}_0 = 0.$$

In the H^∞ a priori filtering problem we need to begin with the pre-array

$$\begin{bmatrix} \begin{bmatrix} \gamma I_q & 0 \\ 0 & I_p \end{bmatrix} & \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{bmatrix}, \tag{5.3.28}$$

and with

$$J = \begin{bmatrix} \begin{bmatrix} -I_q & 0 \\ 0 & I_p \end{bmatrix} & & \\ & I_n & \\ & & I_m \end{bmatrix}. \tag{5.3.29}$$

[Note the reversal of the order of the $\{H_j, L_j\}$ as compared to the a posteriori case.]

Using similar arguments we may prove the following result.

Theorem 5.3.2 **(**H^∞ **A Priori Square-Root Algorithm)** *The* H^∞ *a priori filtering problem with level* γ *has a solution if, and only if, for all* $j = 0, \ldots, i$ *there exist* J*-unitary matrices (with* J *given by (5.3.29)),* Θ_j*, such that*

$$\begin{bmatrix} \begin{bmatrix} \gamma I_q & 0 \\ 0 & I_p \end{bmatrix} & \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j}^{1/2} & 0 & 0 \\ K_{p,j} R_{e,j}^{1/2} & P_{j+1}^{1/2} & 0 \end{bmatrix} \tag{5.3.30}$$

where the algorithm is initialized with, $P_0 = \Pi_0$*. If this is the case, then all possible* H^∞ *a priori filters,* $\check{s}_j = \mathcal{F}_{f,j}(y_0, \ldots y_{j-1})$*, are given by any choices that yield,*

$$\sum_{j=0}^{k} \begin{bmatrix} \check{s}_j - L_j \hat{x}_j \\ y_j - H_j \hat{x}_j \end{bmatrix}^* R_{e,j}^{-1} \begin{bmatrix} \check{s}_j - L_j \hat{x}_j \\ y_j - H_j \hat{x}_j \end{bmatrix} \geq 0, \quad 0 \leq k \leq i$$

where $\hat{x}_j$ *satisfies the recursion,*

$$\hat{x}_{j+1} = F_j \hat{x}_j + K_{p,j} \begin{bmatrix} \check{s}_j - L_j \hat{x}_j \\ y_j - H_j \hat{x}_j \end{bmatrix}, \qquad \hat{x}_0 = 0.$$

Note that, as in the H^2*case*, the quantities necessary to update the square-root array and to calculate the desired estimates may all be found from the triangularized post-array.

In conventional Kalman filtering, square-root arrays are preferred since the positive-definiteness of the matrices is guaranteed, and since the Θ_j are unitary, which improves the numerical stability of the algorithm. In the H^∞ setting, the square-root arrays guarantee that the various matrices have their appropriate inertia; however, the Θ_j are no longer unitary but J-unitary. Therefore the numerical aspects need further investigation.

An interesting aspect of Theorems 5.3.1 and 5.3.2 is that there is no need to explicitly check for the existence conditions required of H^∞ filters (see Theorems 3.2.1 and 3.2.2). These conditions are built into the square-root algorithms themselves: if the algorithms can be performed an H^∞ estimator of the desired level exists, and if they cannot be performed such an estimator does not exist.

5.3.2 The Central Filters

In the previous section we obtained a square-root version of the parametrization of all H^∞ a posteriori and a priori filters. Perhaps the most important filters in these classes are the so-called central filters, which, as we have seen earlier, possess the additional properties of being maximum-entropy and risk-sensitive-optimal filters, as well as being the solution to certain quadratic dynamic games (see Chapters 1, 3 abd 4). In this section we shall develop square-root algorithms specifically for such central filters. As expected, the observer gains for the central filters turn out to be readily obtainable from the square-root algorithms of Theorems 5.3.1 and 5.3.2.

Let us begin by recalling (from Theorem 3.2.1) the central H^∞ a posteriori filter recursions,

$$\hat{x}_{j+1|j+1} = F_j\hat{x}_{j|j} + K_{s,j+1}(y_{j+1} - H_{j+1}F_j\hat{x}_{j|j}), \tag{5.3.31}$$

where the desired estimate is given by $\hat{s}_{j|j} = L_j\hat{x}_{j|j}$, and where the gain vector is given by

$$K_{s,j} = P_jH_j^*(I_p + H_jP_jH_j^*)^{-1}. \tag{5.3.32}$$

We will now show how to obtain the above gain vector from the a posteriori square-root recursions.

To this end, let us first note that we can rewrite the a posteriori square-root algorithm of Theorem 5.3.1 via the following two-step procedure,

$$\begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{1/2} \\ 0 & P_j^{1/2} \end{bmatrix} \Theta_j^{(1)} = \begin{bmatrix} R_{e,j}^{1/2} & 0 \\ K_{f,j}R_{e,j}^{1/2} & P_{j|j}^{1/2} \end{bmatrix}, \tag{5.3.33}$$

$$\begin{bmatrix} F_jP_{j|j}^{1/2} & G_j \end{bmatrix} \Theta_j^{(2)} = \begin{bmatrix} P_{j+1}^{1/2} & 0 \end{bmatrix} \tag{5.3.34}$$

where $\Theta_j^{(1)}$ is J-unitary, with $J = I_p \oplus (-I_q) \oplus I_n$, and $\Theta_j^{(2)}$ is unitary. [Note that the above two-step procedure is the H^∞ analog of Algorithm 5.2.2.] In the above recursions, we of course have

$$K_{f,j} = P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} R_{e,j}^{-1}, \tag{5.3.35}$$

with

$$R_{e,j} = \begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix} + \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix}. \tag{5.3.36}$$

Now in (5.3.33), $R_{e,j}^{1/2}$ can be any square-root of $R_{e,j}$. Let us study the consequences of choosing a lower triangular square-root. To do so, consider the following triangular factorization of $R_{e,j}$,

$$\begin{bmatrix} I_p & 0 \\ L_j P_j H_j^* (I_p + H_j P_j H_j^*)^{-1} & I_q \end{bmatrix} \begin{bmatrix} I_p + H_j P_j H_j^* & 0 \\ 0 & -\Delta_j \end{bmatrix} \begin{bmatrix} I_p & (I_p + H_j P_j H_j^*)^{-1} H_j P_j L_j^* \\ 0 & I_q \end{bmatrix}, \tag{5.3.37}$$

where we have defined the Schur complement,

$$-\Delta_j \triangleq -\gamma^2 I_q + L_j P_j L_j^* - L_j P_j H_j^* (I_p + H_j P_j H_j^*)^{-1} H_j P_j L_j^*.$$

Note that the inertia condition on $R_{e,j}$ implies that $\Delta_j > 0$, so that we may write

$$R_{e,j} = R_{e,j}^{1/2} S R_{e,j}^{*/2}, \tag{5.3.38}$$

with

$$\begin{aligned} R_{e,j}^{1/2} &= \begin{bmatrix} I_p & 0 \\ L_j P_j H_j^* (I_p + H_j P_j H_j^*)^{-1} & I_q \end{bmatrix} \begin{bmatrix} (I_p + H_j P_j H_j^*)^{1/2} & 0 \\ 0 & \Delta_j^{1/2} \end{bmatrix} \\ &= \begin{bmatrix} (I_p + H_j P_j H_j^*)^{1/2} & 0 \\ L_j P_j H_j^* (I_p + H_j P_j H_j^*)^{-*/2} & \Delta_j^{1/2} \end{bmatrix}, \end{aligned} \tag{5.3.39}$$

and $S = \begin{bmatrix} I_p & \\ & -I_q \end{bmatrix}$. Now the $(2,1)$ block entry in the postarray of (5.3.33) is given by

$$\begin{aligned} K_{f,j} R_{e,j}^{1/2} &= P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} R_{e,j}^{-*/2} \\ &= P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} \begin{bmatrix} (I_p + H_j P_j H_j^*)^{-*/2} & (I_p + H_j P_j H_j^*)^{-1} H_j P_j L_j^* \Delta_j^{-*/2} \\ 0 & \Delta_j^{-*/2} \end{bmatrix} \\ &= P_j \begin{bmatrix} H_j^* (I_p + H_j P_j H_j^*)^{-*/2} & \times \end{bmatrix}, \end{aligned} \tag{5.3.40}$$

where $\times$ denotes irrelevant entries.

Eqs. (5.3.39) and (5.3.40) now suggest how to compute the desired gain vector $K_{s,i}$. Indeed,

$$\begin{array}{c}\left(\text{first block column of } K_{f,j}R_{e,j}^{1/2}\right) \cdot \left((1,1) \text{ block entry of } R_{e,j}^{1/2}\right)^{-1} = \\ P_j H_j^*(I_p + H_j P_j H_j^*)^{-*/2} \cdot (I_p + H_j P_j H_j^*)^{-1/2} = \\ K_{s,j}.\end{array} \quad (5.3.41)$$

We are thus led to the following result.

Algorithm 5.3.1 (Central H^∞ A Posteriori Square-Root Algorithm) *The H^∞ a posteriori filtering problem with level γ has a solution if, and only if, for all $j = 0, \ldots, i$ there exist J-unitary (with $J = I_p \oplus (-I_q) \oplus I_n$) matrices, $\Theta_j^{(1)}$, such that*

$$\begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{1/2} \\ 0 & P_j^{1/2} \end{bmatrix} \Theta_j^{(1)} = \begin{bmatrix} R_{e,j}^{1/2} & 0 \\ K_{f,j}R_{e,j}^{1/2} & P_{j|j}^{1/2} \end{bmatrix}, \quad (5.3.42)$$

$$\begin{bmatrix} F_j P_{j|j}^{1/2} & G_j \end{bmatrix} \Theta_j^{(2)} = \begin{bmatrix} P_{j+1}^{1/2} & 0 \end{bmatrix} \quad (5.3.43)$$

with $R_{e,j}^{1/2}$ lower block triangular, and with $\Theta_j^{(2)}$ unitary. The gain vector $K_{s,j}$ needed to update the estimates in the central filter recursions

$$\hat{x}_{j|j} = F_{j-1}\hat{x}_{j-1|j-1} + K_{s,j}(y_j - H_j F_{j-1}\hat{x}_{j-1|j-1}), \quad \hat{x}_{-1|-1} = 0,$$

is equal to

$$K_{s,j} = \bar{K}_{s,j}(I + H_j P_j H_j^*)^{-1/2},$$

where $\bar{K}_{s,j}$ is given by the first block column of $\bar{K}_{f,j} = K_{f,j}R_{e,j}^{1/2}$, and $(I + H_j P_j H_j^*)^{1/2}$ is given by the $(1,1)$ block entry of $R_{e,j}^{1/2}$. The algorithm is initialized with $P_0 = \Pi_0$.

We can now proceed with a similar argument to find square-root recursions for the central H^∞ a priori filters. Let us first recall (from Theorem 3.2.2) that the central H^∞ a priori filter recursions are

$$\hat{x}_{j+1} = F_j\hat{x}_j + K_{a,j}(y_j - H_j\hat{x}_j), \quad (5.3.44)$$

where the desired estimate is given by $\hat{s}_j = L_j\hat{x}_j$, and where the gain vector is given by

$$K_{a,j} = F_j\tilde{P}_jH_j^*(I_p + H_j\tilde{P}_jH_j^*)^{-1}, \tag{5.3.45}$$

with

$$\tilde{P}_j = P_j - P_jL_j^*(-\gamma^2 I_q + L_jP_jL_j^*)^{-1}L_jP_j. \tag{5.3.46}$$

We will now show how to obtain the above gain vector from the a priori square-root recursions,

$$\begin{bmatrix} \begin{bmatrix} \gamma I_q & 0 \\ 0 & I_p \end{bmatrix} & \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_jP_j^{1/2} & G_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j}^{1/2} & 0 & 0 \\ K_{p,j}R_{e,j}^{1/2} & P_{j+1}^{1/2} & 0 \end{bmatrix} \tag{5.3.47}$$

Note now that,

$$K_{p,j} = F_jP_j \begin{bmatrix} L_j^* & H_j^* \end{bmatrix} R_{e,j}^{-1}, \tag{5.3.48}$$

with

$$R_{e,j} = \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix} + \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j \begin{bmatrix} L_j^* & H_j^* \end{bmatrix}. \tag{5.3.49}$$

As mentioned earlier, $R_{e,j}^{1/2}$ in (5.3.47) can be any indefinite square-root of R_e. Let us, once more, study the consequences of choosing a lower triangular square-root. To do so, consider the following block lower-diagonal-upper triangular factorization of the matrix, $R_{e,j}$,

$$\begin{bmatrix} I_q & 0 \\ -\gamma^{-2}H_j\tilde{P}_jL_j^* & I_p \end{bmatrix} \begin{bmatrix} -\gamma^2 I_q + L_jP_jL_j^* & 0 \\ 0 & I_p + H_j\tilde{P}_jH_j^* \end{bmatrix} \begin{bmatrix} I_q & -\gamma^{-2}L_j\tilde{P}_jH_j^* \\ 0 & I_p \end{bmatrix}, \tag{5.3.50}$$

where we have used the facts that,

$$H_jP_jL_j^*(-\gamma^2 I_q + L_jP_jL_j^*)^{-1} = -\gamma^{-2}H_j\tilde{P}_jL_j^*,$$

and, for the Schur complement,

$$I_p + H_jP_jH_j^* - H_jP_jL_j^*(-\gamma^2 I_q + L_jP_jL_j^*)^{-1}L_jP_jH_j^* = I_p + H_j\tilde{P}_jH_j^*.$$

Now the inertia conditions on $R_{e,j}$ require that, $-\gamma^{-2}H_j\tilde{P}_jL_j^* < 0$ and $I_p + H_j\tilde{P}_jH_j^* > 0$, so that we may write,

$$R_{e,j} = R_{e,j}^{1/2} S R_{e,j}^{*/2}, \tag{5.3.51}$$

with

$$R_{e,j}^{1/2} = \begin{bmatrix} (\gamma^2 I_q - L_jP_jL_j^*)^{1/2} & 0 \\ -\gamma^{-2}H_j\tilde{P}_jL_j^*(\gamma^2 I_q - L_jP_jL_j^*)^{1/2} & (I_p + H_j\tilde{P}_jH_j^*)^{1/2} \end{bmatrix}, \tag{5.3.52}$$

and $S = \begin{bmatrix} -I_q & \\ & I_p \end{bmatrix}$. Now the $(2,1)$ block entry in the postarray of (5.3.47) is given by

$$\begin{aligned} K_{p,j}R_{e,j}^{1/2} &= F_jP_j \begin{bmatrix} L_j^* & H_j^* \end{bmatrix} R_{e,j}^{-*/2} \\ &= F_jP_j \begin{bmatrix} L_j^* & H_j^* \end{bmatrix} \begin{bmatrix} (\gamma^2 I_q - L_jP_jL_j^*)^{-*/2} & \gamma^{-2}L_j\tilde{P}_jH_j^*(I_p + H_j\tilde{P}_jH_j^*)^{-*/2} \\ 0 & (I_p + H_j\tilde{P}_jH_j^*)^{-*/2} \end{bmatrix} \\ &= F_jP_j \begin{bmatrix} \times & (\gamma^{-2}L_j^*L_j\tilde{P}_j + I_n)H_j^*(I_p + H_j\tilde{P}_jH_j^*)^{-*/2} \end{bmatrix} \\ &= F_j \begin{bmatrix} \times & \tilde{P}_jH_j^*(I_p + H_j\tilde{P}_jH_j^*)^{-*/2} \end{bmatrix}, \end{aligned} \tag{5.3.53}$$

where in the last step we have used the (readily verified) identity,

$$P_j(\gamma^{-2}L_j^*L_j\tilde{P}_j + I_n) = \tilde{P}_j,$$

and where $\times$ denotes irrelevant entries.

Eqs. (5.3.52) and (5.3.53) now suggest how to compute the desired gain vector $K_{a,i}$. Indeed,

$$\begin{gathered} \left(\text{second block column of } K_{p,j}R_{e,j}^{1/2}\right) \cdot \left((2,2) \text{ block entry of } R_{e,j}^{1/2}\right)^{-1} = \\ F_j\tilde{P}_jH_j^*(I_p + H_j\tilde{P}_jH_j^*)^{-*/2} \cdot (I_p + H_j\tilde{P}_jH_j^*)^{-1/2} = \\ K_{a,j}. \end{gathered} \tag{5.3.54}$$

We are thus led to the following result.

Algorithm 5.3.2 (Central H^∞ A Priori Square-Root Algorithm) *The H^∞ a priori filtering problem with level γ has a solution if, and only if, for all $j = 0, \ldots, i$*

there exist J-unitary matrices (with $J = (-I_q) \oplus I_p \oplus I_n \oplus I_m$), Θ_j, such that

$$\left[\begin{array}{ccc} \begin{bmatrix} \gamma I_q & 0 \\ 0 & I_p \end{bmatrix} & \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{array} \right] \Theta_j = \begin{bmatrix} R_{e,j}^{1/2} & 0 & 0 \\ K_{p,j} R_{e,j}^{1/2} & P_{j+1}^{1/2} & 0 \end{bmatrix} \tag{5.3.55}$$

with $R_{e,j}^{1/2}$ lower block triangular. The gain vector $K_{a,i}$ needed to update the estimates in

$$\hat{x}_{j+1} = F_j \hat{x}_j + K_{a,j}(y_j - F_j \hat{x}_j), \quad \hat{x}_0 = 0,$$

is equal to

$$K_{a,j} = \bar{K}_{a,j}(I + H_j \tilde{P}_j H_j^*)^{-1/2},$$

where $\bar{K}_{a,j}$ is given by the second block column of $\bar{K}_{p,j} = K_{p,j} R_{e,j}^{1/2}$, and $(I + H_j \tilde{P}_j H_j^*)^{1/2}$ is given by the $(2,2)$ block entry of $R_{e,j}^{1/2}$. The algorithm is initialized with $P_0 = \Pi_0$.

5.4 H^2 Chandrasekhar Recursions

The conventional Kalman filter and square-root array recursions of Sec. 5.2 both require $O(n^3)$ operations per iteration (where n is the number of states in the state-space model). However, when the state-space model is time-invariant (or if the time-variation is structured in a certain way) the Chandrasekhar recursions offer an algorithm that requires $O(n^2)$ operations per iteration [Kai72, MSK74, SK94a].

In what follows we shall assume a time-invariant state-space model of the form

$$\begin{cases} x_{j+1} &= Fx_j + Gu_j, \quad x_0 \\ y_j &= Hx_j + v_j \end{cases} \tag{5.4.1}$$

where the $\{u_j, v_j\}$ are disturbances whose nature depends on the criterion being used, and where the $\{y_j\}$ are the observed outputs. In the H^2 case, where the $\{u_j, v_j\}$ are zero mean independent random variables. we shall also assume that the covariances of the $\{u_j, v_j\}$ are constant, *i.e.* $Q_j = Q \geq 0$ and $R_j = R > 0$, for all j. As before, we are interested in obtaining estimates of some linear combinations of the states, $s_j = L_j x_j$, and, in particular, the filtered estimates, $\hat{s}_{j|j} = L_j \hat{x}_{j|j}$, and predicted estimates, $\hat{s}_j = L_j \hat{x}_j$, that use the observations $\{y_k\}_{k=0}^{j}$ and $\{y_k\}_{k=0}^{j-1}$, respectively.

Under the aforementioned assumptions, it turns out that we can write

$$P_{j+1} - P_j = M_j S M_j, \qquad \forall j, \tag{5.4.2}$$

where M_j is a $n \times d$ matrix and S is a $d \times d$ signature matrix (*i.e.* a diagonal matrix with $+1$ and -1 on the diagonal). Thus, for time-invariant state-space models, $P_{j+1} - P_j$ has rank d for all j and in addition has constant inertia. In several important cases, d can be much less than n. When this is true, propagating the smaller matrices M_j, which is equivalent to propagating the P_j, can offer computational reductions. This what is done by the Chandrasekhar recursions (see [Kai81], App. II).

In the conventional Chandrasekhar recursions, one begins with the pre-array

$$\begin{bmatrix} R_{e,j}^{1/2} & HM_j \\ \bar{K}_{p,j} & FM_j \end{bmatrix}, \tag{5.4.3}$$

where $R_{e,j}^{1/2} R_{e,j}^{*/2} = R_{e,j} = R + HP_jH^*$ and $\bar{K}_{p,j} = K_{p,j} R_{e,j}^{-*/2}$, and triangularizes the array using a J-unitary matrix Θ_j where J is given by $\begin{bmatrix} I_p & \\ & S \end{bmatrix}$. The result of this triangularization gives us the various quantities of interest for propagating the Kalman filter recursions.

Algorithm 5.4.1 (Conventional Chandrasekhar Recursions) *The gain vector $K_{p,j} = \bar{K}_{p,j} R_{e,j}^{-1/2}$ necessary to obtain the state estimates in the conventional Kalman filter*

$$\hat{x}_{j+1} = F\hat{x}_j + K_{p,j}(y_j - H\hat{x}_j), \qquad \hat{x}_0 = 0,$$

can be computed using

$$\begin{bmatrix} R_{e,j}^{1/2} & HM_j \\ \bar{K}_{p,j} & FM_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{1/2} & 0 \\ \bar{K}_{p,j+1} & M_{j+1} \end{bmatrix}, \tag{5.4.4}$$

where Θ_j is any J-unitary matrix (with $J = I_p \oplus S$) that triangularizes the above pre-array. The algorithm is initialized with

$$R_{e,0} = R + H\Pi_0 H^*, \qquad \bar{K}_{p,0} = F\Pi_0 H^* R_{e,0}^{1/2},$$

and

$$P_1 - \Pi_0 = F\Pi_0 F^* + GQG^* - K_{p,0}R_{e,0}K_{p,0}^* - \Pi_0 = M_0 S M_0^*.$$

Thus, once more, the quantities necessary to update the arrays and to calculate the state estimates are all found from the triangularized post array.

The validity of the above algorithm can be readily verified by squaring both sides of the equation,

$$\begin{bmatrix} R_{e,j}^{1/2} & HM_j \\ \bar{K}_{p,j} & FM_j \end{bmatrix} \Theta_j = \begin{bmatrix} A & 0 \\ B & C \end{bmatrix}, \tag{5.4.5}$$

and using the J-unitarity of Θ_j, to find the entries of the post array. This leads to,

$$\underbrace{\underbrace{R_{e,j}^{1/2} R_{e,j}^{*/2}}_{R_{e,j}} + H \underbrace{M_j S M_j^*}_{P_{i+1}-P_i} H^*}_{R_{e,j+1}} = AA^*, \tag{5.4.6}$$

from which we conclude that $A = R_{e,j+1}^{1/2}$, and

$$\underbrace{\underbrace{\bar{K}_{p,j} R_{e,j}^{*/2}}_{FP_jH^*} + F \underbrace{M_j S M_j^*}_{P_{i+1}-P_i} H^*}_{FP_{i+1}H^*} = BA^*, \tag{5.4.7}$$

from which we conclude that $B = FP_{i+1}H^* R_{e,j+1}^{-*/2} = \bar{K}_{p,j+1}$. Finally, we have

$$\begin{aligned} CC^* &= FM_jSM_j^*F^* + \bar{K}_{p,j}\bar{K}_{p,j}^* - BB^* \\ &= F(P_{i+1} - P_i)F^* + \bar{K}_{p,j}\bar{K}_{p,j}^* - \bar{K}_{p,j+1}\bar{K}_{p,j+1}^* \\ &= FP_{i+1}F^* + GQG^* - \bar{K}_{p,j+1}\bar{K}_{p,j+1}^* - \left[FP_iF^* + GQG^* - \bar{K}_{p,j}\bar{K}_{p,j}^*\right] \\ &= P_{i+2} - P_{i+1}, \end{aligned}$$

from which we infer that, $C = M_{j+1}$.

If instead of defining, $M_jSM_j^* = P_{j+1} - P_j$, we had defined,

$$N_j S_f N_j^* = P_{j|j} - P_{j-1|j-1}, \tag{5.4.8}$$

where $P_{j|j} = E\tilde{x}_{j|j}\tilde{x}^*_{j|j}$ is the filtered state error variance, which satisfies the recursion,

$$P_{j|j} = FP_{j-1|j-1}F^* + GQG^* - K_{f,j}R_{e,j}K^*_{f,j}, \tag{5.4.9}$$

with $K_{f,j} = P_jHR^{-1}_{e,j}$, then it is also possible to obtain the following (so-called) filtered form of the Chandrasekhar recursions.[6]

Algorithm 5.4.2 (Conventional Chandrasekhar Recursions- Filtered Form) *The gain vector $K_{f,j} = \bar{K}_{f,j}R^{-1/2}_{e,j}$ necessary to obtain the state estimates in the filtered form of the conventional Kalman filter*

$$\hat{x}_{j|j} = F\hat{x}_{j-1|j-1} + K_{f,j}(y_j - HF\hat{x}_{j-1|j-1}), \quad \hat{x}_{-1|-1} = 0,$$

can be computed using

$$\begin{bmatrix} R^{1/2}_{e,j} & HFN_j \\ \bar{K}_{f,j} & FN_j \end{bmatrix} \Theta_j = \begin{bmatrix} R^{1/2}_{e,j+1} & 0 \\ \bar{K}_{f,j+1} & N_{j+1} \end{bmatrix}, \tag{5.4.10}$$

where Θ_j is any J-unitary matrix (with $J = I_p \oplus S_f$) that triangularizes the above pre-array. The algorithm is initialized with

$$R_{e,0} = R + H\Pi_0H^*, \qquad \bar{K}_{f,0} = \Pi_0H^*R^{1/2}_{e,0},$$

and

$$F^{-1}(P_1 - \Pi_0)F^{-*} = \Pi_0 + F^{-1}GQG^*F^{-*} - K_{f,0}R_{e,0}K^*_{f,0} - F^{-1}\Pi_0F^{-*} = N_0S_fN^*_0.$$

Note that compared to the square-root formulas, the size of the pre-array in the Chandrasekhar recursions has been reduced from $(p+n)\times(p+n+m)$ to $(p+n)\times(p+d)$ where m and p are the dimensions of the driving disturbance and output, respectively, and where n is the number of the states. Thus the number of operations for each iteration has been reduced from $O(n^3)$ to $O(n^2d)$, with d typically much less than n.

[6] Note that in this case it can also be shown that $P_{j+1|j+1} - P_{j|j}$ has constant inertia, given by the inertia of the signature matrix, S_f, for all j. This follows from the fact that $P_{j+1} = FP_{j|j}F^* + GQG^*$, so that $P_{j+1} - P_j = F(P_{j|j} - P_{j-1|j-1})F^*$.

5.5 H^∞ Chandrasekhar Recursions

In this section we shall derive the H^∞ counterparts of the (fast) H^2 Chandrasekhar recursions of the previous section. We shall essentially see that, when the underlying state-space model is time-invariant, all the arguments necessary for the development of these algorithms go through, provided that we consider the geometry of indefinite spaces. We first give the general recursions, and then specialize them to obtain the central filters.

5.5.1 The General Case

Recall from Sec. 5.4 that the Chandrasekhar recursions apply to time-invariant state-space models. Therefore the Krein state-space models whose Kalman filters yield the H^∞ estimators must be time-invariant as well. Indeed for the H^∞ a posteriori filtering problem, we need to assume the following Krein state-space model

$$\left\{\begin{array}{rcl} \mathbf{x}_{i+1} & = & F\mathbf{x}_i + G\mathbf{u}_i \\ \begin{bmatrix} \mathbf{y}_i \\ \check{\mathbf{s}}_{i|i} \end{bmatrix} & = & \begin{bmatrix} H \\ L \end{bmatrix} \mathbf{x}_i + \mathbf{v}_i \end{array}\right. \qquad i \geq 0 \tag{5.5.1}$$

with

$$\left\langle \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \\ \mathbf{x}_0 \end{bmatrix}, \begin{bmatrix} \mathbf{u}_j \\ \mathbf{v}_j \\ \mathbf{x}_0 \end{bmatrix} \right\rangle = \begin{bmatrix} I_m\delta_{ij} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix}\delta_{ij} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Pi_0 \end{bmatrix}. \tag{5.5.2}$$

Suppose that the matrix Π_0 can be chosen such that $P_1 - \Pi_0$ has low rank. In other words,

$$P_1 - \Pi_0 = F\Pi_0F^* + GG^* - K_{p,0}R_{e,0}K_{p,0}^* - \Pi_0 = M_0SM_0^*, \tag{5.5.3}$$

where M_0 is a $n \times d$ matrix (typically $d \ll n$) and S is a $d \times d$ signature matrix, and, of course,

$$K_{p,j} = FP_j \begin{bmatrix} H^* & L^* \end{bmatrix} R_{e,j}^{-1} \quad \text{and} \quad R_{e,j} = \begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix} + \begin{bmatrix} H \\ L \end{bmatrix} P_j \begin{bmatrix} H^* & L^* \end{bmatrix}. \tag{5.5.4}$$

We shall presently show by induction that under the assumptions of a time-invariant state-space model, if the a posteriori H^∞ filtering problem has a solution for all j, then $P_{j+1}-P_j$ has rank d for all j and that we may actually write $P_{j+1}-P_j = M_j S M_j^*$.

Consider the following pre-array

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} H \\ L \end{bmatrix} M_j \\ \bar{K}_{p,j} & FM_j \end{bmatrix}, \tag{5.5.5}$$

which is the extension of the pre-array in (5.4.4) to the Krein state-space model (5.5.1) of the H^∞ a posteriori filtering problem. Now the H^∞ a posteriori filtering problem will have a solution if, and only if, all leading submatrices of R and $R_{e,j}$ (or $R_{e,j+1}$, for that matter) have the same inertia. In view of Lemma 5.3.1 this implies that the H^∞ a posteriori filtering problem with level γ will have a solution if, and only if, there exists a J-unitary matrix Θ_j that triangularizes (5.5.5) where

$$J = \begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} & \\ & S \end{bmatrix}. \tag{5.5.6}$$

Therefore we can write

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} H \\ L \end{bmatrix} M_j \\ \bar{K}_{p,j} & FM_j \end{bmatrix} \Theta_j = \begin{bmatrix} A & 0 \\ B & C \end{bmatrix}. \tag{5.5.7}$$

To identify the elements A, B and C in the post-array we square both sides of (5.5.7) and use the fact that Θ_j is J-unitary. Therefore

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} H \\ L \end{bmatrix} M_j \\ \bar{K}_{p,j} & FM_j \end{bmatrix} \underbrace{\Theta_j J \Theta_j^*}_{=J} \begin{bmatrix} R_{e,j}^{*/2} & \bar{K}_{p,j}^* \\ M_j^* \begin{bmatrix} H^* & L^* \end{bmatrix} & M_j F^* \end{bmatrix} = \begin{bmatrix} A & 0 \\ B & C \end{bmatrix} J \begin{bmatrix} A^* & B^* \\ 0 & C^* \end{bmatrix}. \tag{5.5.8}$$

Equating the $(1,1)$ blocks in (5.5.8) yields,

$$A \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} A^* \;=\; R_{e,j}^{1/2} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} R_{e,j}^{*/2} + \begin{bmatrix} H \\ L \end{bmatrix} M_j S M_j^* \begin{bmatrix} H^* & L^* \end{bmatrix}$$

$$\begin{aligned}
&= R_{e,j} + \begin{bmatrix} H \\ L \end{bmatrix} (P_{j+1} - P_j) \begin{bmatrix} H^* & L^* \end{bmatrix} \\
&= R_j + \begin{bmatrix} H \\ L \end{bmatrix} P_{j+1} \begin{bmatrix} H^* & L^* \end{bmatrix} \\
&= R_{e,j+1}.
\end{aligned}$$

Therefore A is the indefinite square-root of $R_{e,j+1}$,

$$A = R_{e,j+1}^{1/2}. \tag{5.5.9}$$

Equating the $(2,1)$ blocks in (5.5.8) yields,

$$\begin{aligned}
B \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} A^* &= \bar{K}_{p,j} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} R_{e,j}^{*/2} + F M_j S M_j^* \begin{bmatrix} H^* & L^* \end{bmatrix} \\
&= K_{p,j} R_{e,j} + F(P_{j+1} - P_j) \begin{bmatrix} H^* & L^* \end{bmatrix} \\
&= F P_j \begin{bmatrix} H^* & L^* \end{bmatrix} + F(P_{j+1} - P_j) \begin{bmatrix} H^* & L^* \end{bmatrix} \\
&= F P_{j+1} \begin{bmatrix} H^* & L^* \end{bmatrix}.
\end{aligned}$$

Therefore

$$\begin{aligned}
B &= F P_{j+1} \begin{bmatrix} H^* & L^* \end{bmatrix} A^{-*} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \\
&= F P_{j+1} \begin{bmatrix} H^* & L^* \end{bmatrix} R_{e,j+1}^{-*/2} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} = \bar{K}_{p,j+1}.
\end{aligned} \tag{5.5.10}$$

Equating the $(2,2)$ blocks in (5.5.8) yields

$$CSC^* + B \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} B^* = \bar{K}_{p,j} \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \bar{K}_{p,j}^* + F M_j S M_j^* F^*.$$

Therefore

$$CSC^* + K_{p,j+1} R_{e,j+1} K_{p,j+1}^* = K_{p,j} R_{e,j} K_{p,j}^* + F(P_{j+1} - P_j) F^*.$$

We can now write

$$\begin{aligned}
CSC^* &= F P_{j+1} F^* - K_{p,j+1} R_{e,j+1} K_{p,j+1}^* + GG^* - (F P_j F^* - K_{p,j} R_{e,j} K_{p,j}^* + GG^*) \\
&= P_{j+2} - P_{j+1},
\end{aligned}$$

and finally

$$C = M_{j+1}. \tag{5.5.11}$$

Note that our derivation of C also shows that if $P_1 - P_0 = M_0 S M_0^*$ then $P_{i+1} - P_i = M_i S M_i^*$ for all i.

We have thus established

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} H \\ L \end{bmatrix} M_j \\ \bar{K}_{p,j} & F M_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{1/2} & 0 \\ \bar{K}_{p,j+1} & M_{j+1} \end{bmatrix},$$

from which we can now give the following (fast) Chandrasekhar version of the parametrization of all H^∞ a posteriori filters.

Theorem 5.5.1 (H^∞ A Posteriori Chandrasekhar Recursions) *The H^∞ a posteriori filtering problem with level γ has a solution if, and only if, all leading submatrices of*

$$R = \begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix} \quad \text{and} \quad R_{e,0} = \begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix} + \begin{bmatrix} H \\ L \end{bmatrix} \Pi_0 \begin{bmatrix} H^* & L^* \end{bmatrix}$$

have the same inertia, and if for all $j = 0, \ldots, i$ there exist J-unitary matrices (with $J = I_p \oplus (-I_q) \oplus S$), Θ_j, such that

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} H \\ L \end{bmatrix} M_j \\ K_{p,j} R_{e,j}^{1/2} & F M_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{1/2} & 0 \\ K_{p,j+1} R_{e,j+1}^{1/2} & M_{j+1} \end{bmatrix} \tag{5.5.12}$$

where the algorithm is initialized with, $R_{e,0}$, $K_{p,0} = F\Pi_0 \begin{bmatrix} H^* & L^* \end{bmatrix} R_{e,0}^{-1}$, and

$$P_1 - \Pi_0 = F\Pi_0 F^* + GQG^* - K_{p,0} R_{e,0} K_{p,0}^* - \Pi_0 = M_0 S M_0^*. \tag{5.5.13}$$

If this is the case, then all possible H^∞ a posteriori filters, $\check{s}_{j|j} = \mathcal{F}_{f,j}(y_0, \ldots y_j)$, are given by any choices that yield,

$$\sum_{j=0}^{k} \begin{bmatrix} y_j - H_j \hat{x}_j \\ \check{s}_{j|j} - L_j \hat{x}_j \end{bmatrix}^* R_{e,j}^{-1} \begin{bmatrix} y_j - H_j \hat{x}_j \\ \check{s}_{j|j} - L_j \hat{x}_j \end{bmatrix} \geq 0, \quad 0 \leq k \leq i$$

where $\hat{x}_j$ satisfies the recursion,

$$\hat{x}_{j+1} = F_j\hat{x}_j + K_{p,j}\begin{bmatrix} y_j - H_j\hat{x}_j \\ \check{s}_{j|j} - L_j\hat{x}_j \end{bmatrix}, \qquad \hat{x}_0 = 0.$$

In the H^∞ a priori filtering problem, we need, instead, to start with the Krein state-space model,

$$\begin{cases} \mathbf{x}_{i+1} = F\mathbf{x}_i + G\mathbf{u}_i \\ \begin{bmatrix} \check{\mathbf{s}}_i \\ \mathbf{y}_i \end{bmatrix} = \begin{bmatrix} L \\ H \end{bmatrix}\mathbf{x}_i + \mathbf{v}_i \end{cases} \qquad i \geq 0 \tag{5.5.14}$$

with

$$\left\langle \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \\ \mathbf{x}_0 \end{bmatrix}, \begin{bmatrix} \mathbf{u}_j \\ \mathbf{v}_j \\ \mathbf{x}_0 \end{bmatrix} \right\rangle = \begin{bmatrix} I_m\delta_{ij} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix}\delta_{ij} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Pi_0 \end{bmatrix}, \tag{5.5.15}$$

and with the prearray,

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} L \\ H \end{bmatrix} M_j \\ K_{p,j}R_{e,j}^{1/2} & FM_j \end{bmatrix}, \tag{5.5.16}$$

where now

$$K_{p,j} = FP_j\begin{bmatrix} L^* & H^* \end{bmatrix}R_{e,j}^{-1} \quad \text{and} \quad R_{e,j} = \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix} + \begin{bmatrix} L \\ H \end{bmatrix}P_j\begin{bmatrix} L^* & H^* \end{bmatrix}. \tag{5.5.17}$$

[Note that the only difference with the a posteriori case is in the order of the matrices $\{H, L\}$.]

Proceeding with an argument similar to what was done in the a posteriori case, we can show the following result.

Theorem 5.5.2 (H^∞ A Priori Chandrasekhar Recursions) *The H^∞ a priori filtering problem with level γ has a solution if, and only if, all leading submatrices of*

$$R = \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix} \quad \textit{and} \quad R_{e,0} = \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix} + \begin{bmatrix} L \\ H \end{bmatrix}\Pi_0\begin{bmatrix} L^* & H^* \end{bmatrix}$$

have the same inertia, and if for all $j = 0, \ldots, i$ there exist J-unitary matrices (with $J = (-I_q) \oplus I_p \oplus S$), Θ_j, such that

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} L \\ H \end{bmatrix} M_j \\ K_{p,j} R_{e,j}^{1/2} & F M_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{1/2} & 0 \\ K_{p,j+1} R_{e,j+1}^{1/2} & M_{j+1} \end{bmatrix} \tag{5.5.18}$$

where the algorithm is initialized with, $R_{e,0}$, $K_{p,0} = F \Pi_0 \begin{bmatrix} L^* & H^* \end{bmatrix} R_{e,0}^{-1}$, and

$$P_1 - \Pi_0 = F \Pi_0 F^* + G Q G^* - K_{p,0} R_{e,0} K_{p,0}^* - \Pi_0 = M_0 S M_0^*. \tag{5.5.19}$$

If this is the case, then all possible H^∞ a priori filters, $\check{s}_j = \mathcal{F}_{f,j}(y_0, \ldots y_{j-1})$, are given by any choices that yield,

$$\sum_{j=0}^{k} \begin{bmatrix} \check{s}_j - L_j \hat{x}_j \\ y_j - H_j \hat{x}_j \end{bmatrix}^* R_{e,j}^{-1} \begin{bmatrix} \check{s}_j - L_j \hat{x}_j \\ y_j - H_j \hat{x}_j \end{bmatrix} \geq 0, \quad 0 \leq k \leq i$$

where $\hat{x}_j$ satisfies the recursion,

$$\hat{x}_{j+1} = F_j \hat{x}_j + K_{p,j} \begin{bmatrix} \check{s}_j - L_j \hat{x}_j \\ y_j - H_j \hat{x}_j \end{bmatrix}, \qquad \hat{x}_0 = 0.$$

Note that compared to the H^∞ square-root formulas, the size of the pre-array in the H^∞ Chandrasekhar recursions has been reduced from $(p+q+n) \times (p+q+n+m)$ to $(p+q+n) \times (p+q+d)$ where m, p and q are the dimensions of the driving disturbance, output and states to be estimated, respectively, and where n is the number of the states. Thus the number of operations for each iteration has been reduced from $O(n^3)$ to $O(n^2 d)$ with d typically much less than n.

As in the square-root case, the Chandrasekhar recursions do not require explicitly checking the positivity conditions of Theorems 3.2.1 and 3.2.2 — if the recursions can be carried out an H^∞ estimator of the desired level exists, and if not, such an estimator does not exist.

5.5.2 The Central Filters

The preceding section gave (fast) Chandrasekhar versions of all possible H^∞ a posteriori and a priori filters. Here we shall specialize these recursions to the central a posteriori and a priori filters. We shall show that the observer gains for these filters can be readily obtained from the post-arrays of Theorems 5.5.1 and 5.5.2, provided we insist on (block) lower triangular square-roots for $R_{e,j}^{1/2}$. The development closely follows that of Sec. 5.3.2 and uses the important facts (established in Sec. 5.3.2) that if $R_{e,j}^{1/2}$ is lower triangular, then $K_{s,j}$, the gain vector for the central a posteriori filter, is given by,

$$K_{s,j} = \left(\text{first block column of } K_{f,j} R_{e,j}^{1/2}\right) \cdot \left((1,1) \text{ block entry of } R_{e,j}^{1/2}\right)^{-1}, \quad (5.5.20)$$

and $K_{a,j}$, the gain vector for the central a priori filter, is given by,

$$K_{a,j} = \left(\text{second block column of } K_{p,j} R_{e,j}^{1/2}\right) \cdot \left((2,2) \text{ block entry of } R_{e,j}^{1/2}\right)^{-1}. \quad (5.5.21)$$

The following result are now readily establishable. (The proofs are straightforward and will be omitted for brevity.)

Algorithm 5.5.1 (Central H^∞ A Posteriori Chandrasekhar Recursions) *The H^∞ a posteriori filtering problem with level γ has a solution if, and only if, all leading submatrices of*

$$R = \begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix} \quad \text{and} \quad R_{e,0} = \begin{bmatrix} I_p & 0 \\ 0 & -\gamma^2 I_q \end{bmatrix} + \begin{bmatrix} H \\ L \end{bmatrix} \Pi_0 \begin{bmatrix} H^* & L^* \end{bmatrix}$$

have the same inertia, and if for all $j = 0, \ldots, i$ there exist J-unitary matrices, Θ_j, (where $J = I_p \oplus (-I_q) \oplus S_f$) such that

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} H \\ L \end{bmatrix} F N_j \\ K_{f,j} R_{e,j}^{1/2} & F N_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{1/2} & 0 \\ K_{f,j} R_{e,j+1}^{1/2} & N_{j+1} \end{bmatrix} \quad (5.5.22)$$

with $R_{e,j}^{1/2}$ and $R_{e,j+1}^{1/2}$ block lower triangular. The algorithm is initialized with, $R_{e,0}$, $K_{f,0} = \Pi_0 \begin{bmatrix} H^* & L^* \end{bmatrix} R_{e,0}^{-1}$, and

$$F^{-1}(P_1 - \Pi_0)F^{-*} = \Pi_0 + F^{-1}GQG^*F^{-*} - K_{f,0}R_{e,0}K_{f,0}^* - F^{-1}\Pi_0F^{-*} = N_0 S_f N_0^*.$$

The gain vector $K_{s,j}$ *needed to update the estimates in the central filter recursions*

$$\hat{x}_{j|j} = F_{j-1}\hat{x}_{j-1|j-1} + K_{s,j}(y_j - H_jF_{j-1}\hat{x}_{j-1|j-1}), \quad \hat{x}_{-1|-1} = 0,$$

is equal to

$$K_{s,j} = \bar{K}_{s,j}(I + H_jP_jH_j^*)^{-1/2},$$

where $\bar{K}_{s,j}$ *is given by the first block column of* $\bar{K}_{f,j} = K_{f,j}R_{e,j}^{1/2}$, *and* $(I + H_jP_jH_j^*)^{1/2}$ *is given by the* $(1,1)$ *block entry of* $R_{e,j}^{1/2}$.

Algorithm 5.5.2 (Central H^∞ A Priori Chandrasekhar Recursions) *The* H^∞ *a priori filtering problem with level* γ *has a solution if, and only if, all leading submatrices of*

$$R = \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix} \quad \text{and} \quad R_{e,0} = \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix} + \begin{bmatrix} L \\ H \end{bmatrix} \Pi_0 \begin{bmatrix} L^* & H^* \end{bmatrix}$$

have the same inertia, and if for all $j = 0, \ldots, i$ *there exist* J*-unitary matrices,* Θ_j, *(where* $J = (-I_q) \oplus I_p \oplus S$*) such that*

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} L \\ H \end{bmatrix} M_j \\ K_{p,j}R_{e,j}^{1/2} & FM_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{1/2} & 0 \\ K_{p,j+1}R_{e,j+1}^{1/2} & M_{j+1} \end{bmatrix} \tag{5.5.23}$$

with $R_{e,j}^{1/2}$ *and* $R_{e,j+1}^{1/2}$ *block lower triangular. The algorithm is initialized with,* $R_{e,0}$, $K_{p,0} = F\Pi_0 \begin{bmatrix} L^* & H^* \end{bmatrix} R_{e,0}^{-1}$ *and*

$$P_1 - \Pi_0 = F\Pi_0F^* + GQG^* - K_{p,0}R_{e,0}K_{p,0}^* - \Pi_0 = M_0SM_0^*.$$

The gain vector $K_{a,i}$ *needed to update the estimates in*

$$\hat{x}_{j+1} = F_j\hat{x}_j + K_{a,j}(y_j - F_j\hat{x}_j), \quad \hat{x}_0 = 0,$$

is equal to

$$K_{a,j} = \bar{K}_{a,j}(I + H_j\tilde{P}_jH_j^*)^{-1/2},$$

where $\bar{K}_{a,j}$ *is given by the second block column of* $\bar{K}_{p,j} = K_{p,j}R_{e,j}^{1/2}$, *and* $(I + H_j\tilde{P}_jH_j^*)^{1/2}$ *is given by the* $(2,2)$ *block entry of* $R_{e,j}^{1/2}$.

5.6 Conclusion

In this chapter, we obtained square-root array algorithms and Chandrasekhar recursions for the H^∞ a posteriori and a priori filtering problems. These algorithms involve propagating the indefinite square-roots of the quantities of interest, and have the interesting property that the appropriate inertia of these quantities is preserved. Moreover, the conditions for the existence of the H^∞ filters are built into the algorithms, so that filter solutions will exist if,and only if, the algorithms can be executed.

The conventional square-root arrays and Chandrasekhar recursions are preferred because of their numerical stability (in the case of square-root arrays) and their reduced computational complexity (in the case of the Chandrasekhar recursions). Since the H^∞ square-root arrays and Chandrasekhar recursions are the direct analogs of their conventional counterparts, they may be more attractive for numerical implementation of H^∞ filters. However, since J-unitary rather than unitary operations are involved, further numerical investigation is needed.

Our derivation of the H^∞ square-root arrays and Chandrasekhar recursions demonstrates a virtue of the Krein space approach to H^∞ estimation and control; the results appear to be more difficult to conceive and prove in the traditional H^∞ approaches. We should also mention that there are many variations of the conventional square-root array and Chandrasekhar recursions, *e.g.* for control problems, and the methods given here are directly applicable to extending these variations to the H^∞ setting as well. Finally, the algorithms presented here are equally applicable to risk-sensitive estimation and control problems, and to quadratic dynamic games.