Kalman Filtering and Applied Estimation

Lecture 9
Square Root Filters

Erik Blasch

(937) 904-9077
erik.blasch@wpafb.af.mil

Sponsor
WSU

Outline – Square Root

- **Efficient Processing**
  - **Processing the Ricatti Equation**
  - **Sensitivity to round-off errors (precision)**
- **Square Root Types**
  - **Joseph -**
  - **Potter – Cholesky, Householder**
  - **Carlson – Cholesky**
  - **Bierman – UD**
- **Comparisons**
  - **Speed versus accuracy**
    - **Memory is not a problem**

Square Root

- **Chapter 6**
- **Although Kalman filtering is suited for computer implementation, the computer is not ideally suited.**
- **The Kalman filter - in terms of covariance matrices - is particularly sensitive to round-off errors.**
  - **Alternative representations for the covariance matrix of estimation uncertainty, in terms of symmetric products of triangular factors.**
  - **Note: issues sparseness**
  - **Note: covariance, transpose, & inverse covariance computation**
- **The alternative Kalman filter implementations use these factors of the covariance matrix (or its inverse) in three types of filter operations:**
  1. **Temporal updates**
  2. **Observational updates**
  3. **Combined updates (temporal and observational)**

Square Root Approaches

- **Square Root Covariance Approach**
  - Cholesky products
  - Symmetry for easier processing
- **Triangluar Approach**
  - Upper, Lower Triangular
  - Diagonalize – Gram-Schmidt
- **Information Approach**
  - More efficient
  - Information versus track states
  - Shown to be useful for distributed systems

Square Root

- Chapter 6
  - Alternative representations for the covariance matrix of estimation uncertainty, in terms of symmetric products of triangular factors.
- 1. Square root covariance filters, which use a decomposition of the covariance matrix of estimation uncertainty as a symmetric product of triangular Cholesky factors:

$$P = CC^T$$

- 2. UD covariance filters, which use a modified (square-root-free) Cholesky decomposition of the covariance matrix:

$$P = UDU^T$$

- 3. Square root Information filters, which use a symmetric product factorization of the information matrix,

$$I = P^{-1}$$

Square Root

- Chapter 6
  - Alternative representations for the covariance matrix of estimation uncertainty, in terms of symmetric products of triangular factors.
- 1. ***Cholesky decomposition methods***, by which a symmetric positive definite matrix M can be represented as symmetric products of triangular matrix *C:*

$$M = CC^T \quad \text{or} \quad M = UDU^T.$$

- The Cholesky decomposition algorithms compute *C* (or *U* and *D*), given *M.*
- 2. ***Triangularizaton methods***, by which a symmetric product of a general matrix *A* can be represented as a symmetric product of a triangular matrix *C:*

$$AA^T = CC^T \quad \text{or} \quad AA^T = U\,D'\,U^T$$

These methods compute *C* (or *U* and *D*), given *A* (or *A* and D').

- 3. ***Rank One Modification Methods***, by which the sum of a symmetric product of a triangular matrix and scaled symmetric product of a vector (rank-one matrix) v can be represented by a symmetric product of a new triangular matrix *C:*

$$C'C'^T + s\,v\,v^T = CC^T \quad \text{or} \quad U'D'U'^T + s\,v\,v^T = UDU^T$$

These methods compute *C* (or *U* and *D*), given C' (or D' and U'), s, and v.

Square Root

- **KF Operations**

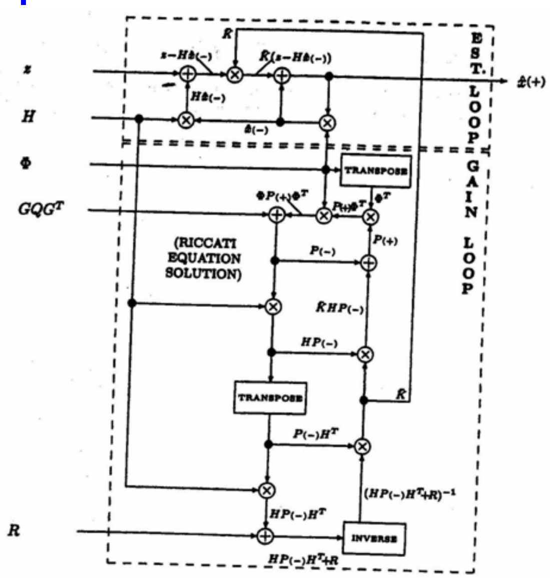
•TABLE 6.5 OPERATIONS FOR CONVENTIONAL KALMAN FILTER

- State dimension = *n*
  Measurement dimension = ℓ
- Operation flops

Operation	flops
$H \times P(-)$	$n^2\ell$
$H \times (HP(-))^T + R$	$(1/2)\, n\,\ell^2 + (1/2)\, n\,\ell$
$[H\,(HP(-))^T + R\,]^{-1}$	$\ell^3 + (1/2)\,\ell^2 + (1/2)\,\ell$
$[H\,(HP(-))^T + R]^{-1} \times [\,HP(-)\,]$	$n\,\ell^2$
$P(-) - [\,HP(-)\,] \times [\,H(HP(-))^T + R]^{-1}\,[HP(-)]$	$(1/2)\, n^2\ell + (1/2)\, n\,\ell$
Total	$(3/2)\, n^2\ell + (3/2)\, n\,\ell^2 + n\,\ell + \ell^3 + (1/2)\,\ell^2 + (1/2)\,\ell$

Square Root

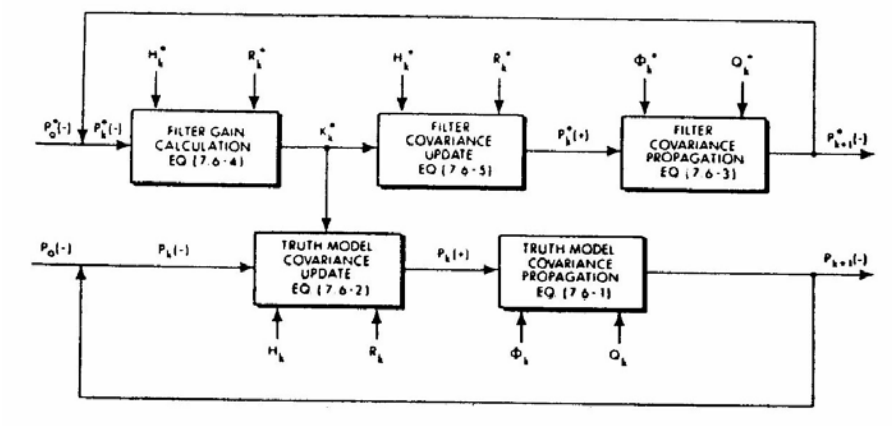
- **KF Operations**



Square Root

- **KF Operations**

Gelb, 1974

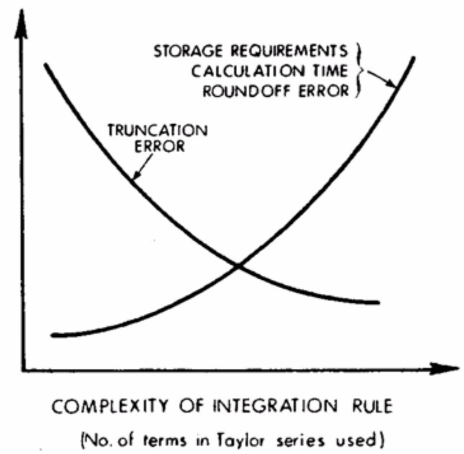


Square Root

- **KF Operations**

Gelb, 1974

•Integration time vs roundoff errors



Square Root

Gelb, 1974

- In the square-root formulation,

Matrix W is calculated instead of P,

where $\boxed{P = WW^T.}$

Thus, P(+) is always assured to be positive definite.
The square-root formulation gives the same accuracy in single precision as does the conventional formulation in double precision.

Unfortunately, the matrix W is not unique, and this leads to a proliferation of square-root algorithms The square-root algorithm takes the form:

Update $W(+) = W(-)\,[\,I - Z(U^T)^{-1}(U+V)^{-1}Z^T]$

Extrapolation $[d\,W(-)/d\,t] = FW(-) + (1/2)\,Q[W^T(-)]^{-1}$

where
$Z = W^T(-)\,H^T$ note: $Z = H\,x + v$
$UU^T = R + Z^T Z$
$VV^T = R$

Square Root

Gelb, 1974

- In the square-root formulation,

Matrix W is calculated instead of P,

where $P = WW^T.$

Differential equation which can be a difference equation

$$\dot{P} = \dot{W}W^T + W\dot{W}^T$$

$$\dot{P} = [\,FW + (1/2)Q(W^T)^{-1}\,]\,W^T + W[W^T F^T + (1/2)\,W^{-1}Q]$$

$$\dot{P} = F\,(WW^T) + (1/2)\,Q + (WW^T)\,F^T + (1/2)\,Q$$

$$\dot{P} = FP + P\,F^T + Q$$

Square Root

Gelb, 1974

Of course, a penalty is paid in that

the “square root” of certain matrices such as R must be calculated; a tedious process involving eigenvalue--eigenvector routines.

An indication of the number of extra calculations required to implement the square-root formulation can be seen from the sample case illustrated in Table 8.4-1, which is for a state vector of dimension 10 and a scalar measurement. This potential increase in computation time has motivated a search for efficient square root algorithms. Recently, Carlson (Ref. 53) has derived an algorithm which utilizes a lower triangle form for W to improve computation speed. Carlson demonstrates that this algorithm approaches or exceeds the speed of the conventional algorithm for low-order filters and reduces existing disadvantages of square-root filters for the high-order case.

TABLE 8.4-1 COMPARISON OF THE NUMBER OF CALCULATIONS INVOLVED IN THE CONVENTIONAL AND SQUARE-ROOT FORMULATIONS OF THE KALMAN FILTER (REF. 11)

	Square Roots	M&D	+	A&S	Equivalent M&D
Update:					
Conventional	0	310		211	352
Square-Root	1	322		302	387
Extrapolation:	Square Roots	M&D	+	A&S	Equivalent M&D
Conventional	0	2100		2250	2550
Square-Root	10	4830		4785	5837

Note: M&D = multiplications and divisions, A&S = additions and subtractions

Square Root

Gelb, 1974

•Example 8.4-2 illustrates how SR algorithms minimize round-off error. Suppose

$H = [1 \ 0]$, $r = \varepsilon^2$ $\quad P(-) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

where $\varepsilon \leq 1$ and to simulate computer word length roundoff, we assume $1 + \varepsilon \neq 1$ but $1 + \varepsilon^2 \cong 1$. It follows that the *exact* value for P(+) is

$$P(+) = \begin{bmatrix} [\varepsilon^2/(1+\varepsilon^2)] & 0 \\ 0 & 1 \end{bmatrix}$$

whereas the value calculated in the computer using the standard Kalman filter algorithm is $P(+) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

Using the square-root algorithm is $P(+) = \begin{bmatrix} \varepsilon^2 & 0 \\ 0 & 1 \end{bmatrix}$

Since $\boxed{K = P(+)\, H^T R^{-1}}$ is:

Exact	$K = \begin{bmatrix} 1/(1+\varepsilon^2) \\ 0 \end{bmatrix}$
Conventional	$K = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
Square-Root	$K = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Exact	True
Conventional	Bad
Square-Root	Good

Clearly conventional formulation may lead to divergence problems.

Square Root

Gelb, 1974

•Storage Requirements

TABLE 8.4-4 KALMAN FILTER STORAGE REQUIREMENTS FOR LARGE N (PROGRAM INSTRUCTIONS NOT INCLUDED) (REF. 43)

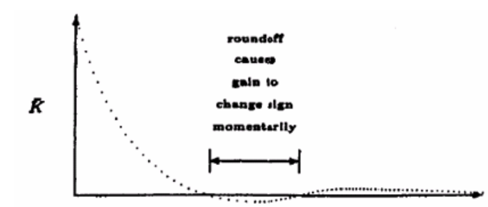
		Storage Locations			
Algorithm		$n \geq m$	$n \cong m$, $n \cong 1$	$n \cong m$, $n \geq 1$	$m \geq n$
Standard Kalman	[Eq. (8.4-1)]	$2.5\,n^2$	$3.5\,(n + 1.3)$	$3.5\,n^2$	m^2
Joseph	[Eq. (8.4-4)]	$2.5\,n^2$	$1.5\,n(n+1)$	$1.5\,n^2$	m^2
Andrews Square-Root		$3\,n^2$	$5.5\,n(n + 0.8)$	$5.5\,n^2$	$2.5\,m^2$
Standard Kalman	[Eq. (8.4-1)] (no symmetry)	$3\,n^2$	$5\,n\,(n + 0.6)$	$5\,n^2$	$2\,m^2$
Joseph	[Eq. (8.4-4)] (no symmetry)	$3n^2$	$6\,n\,(n+0.6)$	$6\,n^2$	$2\,m^2$

n = is the state vector dimension

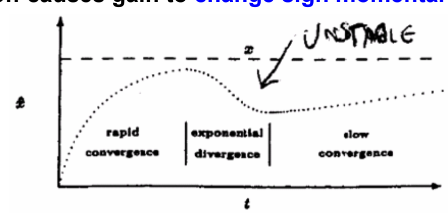
m = is the measurement vector dimension

Square Root

- Chapter 6 - Convergence



- Round-off causes gain to change sign momentarily



- Rapid convergence, exponential divergence, slow convergence

Square Root

•Chapter 6 - KF

Note the reduced computations (?)

TABLE 6.2 FIRST-ORDER ERROR PROPAGATION MODELS

Round-off error in filter variable	Error model (by filter type): Conventional Implementation	Square root covariance
$\delta x_{k+1}(-)$	$A_1[\delta x_k(-) + \delta P_k(-)\ A_2(z - H x_k(-))] + \Delta x_{k+1}$	
$\delta \bar{\bar{K}}_k$	$A_1\,\delta P_k(-)$	
$\delta P_{k+1}(-)$	$A_1\,\delta P_k(-)A_1^T + \Delta P_{k+1}$ $+ \Phi[\delta P_k(-) - \delta P_k^T(-)]\Phi^T - \Phi[\delta P_k(-) - \delta P_k^T(-)]A_1^T$	$A_1\,\delta P_k(-)A_1^T + \Delta P_{k+1}$

$A_1 = \Phi - \delta\bar{\bar{K}}_k H$

$A_2 = H^T[H P_k H^T + R]^{-1}$

The error propagation expression for the conventional Kalman filter includes extra terms that are proportional to the anti-symmetric part of *P*. Consequently, implementation methods maintains the symmetry of P will avoid error propagation. The same effect can be obtained by <u>computing only the unique elements of *P*</u>, which is fairly common practice anyway. (The square root covariance implementations maintain symmetry of *P*.)

Square Root

•Chapter 6 – Roundoff Errors

TABLE 6.3 UPPER BOUNDS ON ADDED ROUNDOFF ERRORS

Norm of added round-off errors	Upper bounds (by filter type): Conventional implementation	Square root covariance
$\lvert\Delta x_{k+1}(-)\rvert$	$\varepsilon_1(\lvert A_1\rvert\lvert x_{k+1}(-)\rvert + \lvert\bar{\bar{K}}_k\rvert\lvert z_k\rvert) + \lvert\Delta\bar{\bar{K}}_k\rvert(\lvert H\rvert\lvert x_k(-)\rvert + \lvert z_k\rvert)$	$\varepsilon_4[\lvert A_1\rvert\lvert x_k(-)\rvert + \lvert\bar{\bar{K}}_k\rvert\lvert z_k\rvert] + \lvert\Delta\bar{\bar{K}}_k\rvert(\lvert H\rvert\lvert x_k(-)\rvert + \lvert z_k\rvert)$
$\lvert\Delta\bar{\bar{K}}_k\rvert$	$\varepsilon_2\,\kappa^2(R^*)\lvert\bar{\bar{K}}_k\rvert$	$\varepsilon_5\,\kappa(R^*)(\lambda_m^{-1}(R^*)\lvert C_{P(k+1)}\rvert + \lvert\bar{\bar{K}}_k C_{R^*}\rvert + \lvert A_3\rvert/\lambda_1(R^*))$
$\lvert\Delta P_{k+1}(-)\rvert$	$\varepsilon_3\,\kappa^2(R^*)\lvert P_{k+1}(-)\rvert$	$\dfrac{\varepsilon_6(1+\kappa(R^*))\lvert P_{k+1}\rvert\lvert A_3\rvert}{\lvert C_{P(k+1)}\rvert}$

$\varepsilon_1, \ldots, \varepsilon_6$ are constant multiples of ε, the unit round-off error.

$A_1 = \Phi - \bar{\bar{K}}_k H.$

$A_3 = [(\bar{\bar{K}}_k C_{R^*})\ \lvert C_{P(k+1)}\rvert]$

$R^* = H P_k(-) H^T + R.$

$R^* = C_R \bullet C_{R^*}^T$ (triangular Cholesky decomposition).

$P_{k+1}(-) = C_{P(k+1)} C_{P(k+1)}^T$ (triangular Cholesky decomposition).

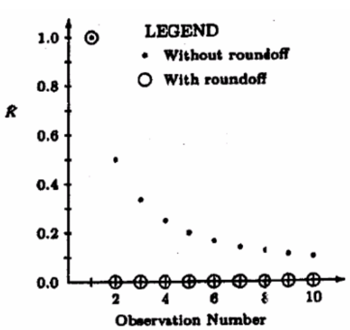
$\lambda_1(R^*) \geq \lambda_2(R^*) \geq \cdots \geq \lambda_m(R^*) \geq 0$ are the characteristic values of R^*.

$\kappa(R^*) = \lambda_1(R^*)/\lambda_m(R^*)$ is the condition number of R^*.

Square Root

•Chapter 6 – Roundoff Errors

(Round-off Errors Due to Large a *priori* Uncertainty)



Observation no.	Expression	Value: Exact	Rounded
1	P_0H^T	P_0	P_0
1	HP_0H^T	P_0	P_0
1	$HP_0H^T + R$	$P_0 + R$	P_0
1	$\bar{\bar{K}}_1 = P_0H^T(HP_0H^T + R)^{-1}$	$P_0/[P_0 + R]$	1
1	$P_1 = P_0 - \bar{\bar{K}}_1 H P_0$	$P_0R/[P_0 + R]$	0
:	:	:	:
k	$\bar{\bar{K}}_k = P_{k-1}H^T(HP_{k-1}H^T + R)^{-1}$	$P_0/[k P_0 + R]$	0
	$P_k = P_{k-1} - \bar{\bar{K}}_k H P_{k-1}$	$P_0R/[k P_0 + R]$	0

After computation – Error in analysis
Need to fix for covariance precision

Square Root

• Chapter 6

- **Joseph [13], who demonstrated improved numerical stability by rearranging the standard formulas for the observational update (given here for scalar measurements) into the following formats:**

$$\grave{z} = R^{-1/2} z$$

$$\grave{H} = \grave{x} H$$

$$\bar{\bar{K}} = [\grave{H} P(-)\grave{H}^T + 1]^{-1} P(-)\grave{H}^T$$

$$P(+) = (I - \bar{\bar{K}}\grave{H})(I - \bar{\bar{K}}\grave{H})^T + \bar{\bar{K}}\bar{\bar{K}}^T$$

Square Root

- **Chapter 6**
- **De Vries implementation**. at Rockwell International, is designed to reduce the computational complexity of Joseph formulation by judicious rearrangement of the matrix expressions and reuse of intermediate results

TABLE 6.6 DE VRIES-JOSEPH IMPLEMENTATION OF CO VARIANCE UPDATE

Operation	Complexity
Without using de-correlation	
$\mathcal{T}_1 = P(-)H^T$	ℓn^2
$\mathcal{T}_2 = H\mathcal{T}_1 + R$	$n\ell(\ell+1)/2$
$\boxed{\mathcal{UDU}^T = \mathcal{T}_2}$	$(1/6)\ell(\ell+1)(\ell+2)$ [UD factorization]
$\mathcal{UDU}^T K^T = \mathcal{T}_1{}^T$	$\ell^2 n$ [to solve for K]
$\mathcal{T}_3 = (1/2)K\mathcal{T}_2 - \mathcal{T}_1$	$\ell^2(n+1)$
$\mathcal{T}_4 = \mathcal{T}_3 K^T$	ℓn^2
$P(+) = P(-) + \mathcal{T}_4 + \mathcal{T}_4{}^T$	[Included above]
Total	$(1/6)\ell^3 + (3/2)\ell^2 + (1/3)\ell + (1/2)\ell n + (5/2)\ell^2 n + 2\ell n^2$
Using de-correlation	$(2/3)\ell^3 + \ell^2 - (5/3)\ell - (1/2)\ell n + (1/2)\ell^2 n$

Square Root

- **Chapter 6 - Factorization**
  - **Matrix factoring and decomposition.**
    - The terms *decomposition* and *factoring* (or *fractorization*) are used interchangeably to describe the process of transforming a matrix or matrix expression into an equivalent product of factors.
  - The term *decomposition* is somewhat more general. It is also used to describe non-product representations, such as the *additive decomposition* of a square matrix into its symmetric and anti-symmetric parts
  - $A = (1/2)(A + A^T) + (1/2)(A - A^T).$

Square Root

- **Chapter 6 - Factorization**
  - Another distinction between *decomposition* and *factorization* is made by Dongarra, Moler, Bunch, and Stewart [81], who use the term
  - *factotization* to refer to an arithmetic process for performing a product decomposition of a matrix in which not all factors are preserved.
  - The term *triangularization* is used in this book to indicate a *QR* factorization (in the sense of Dongarra *et al)* involving a triangular factor that is preserved and an orthogonal factor that is not preserved.
  - The more numerically stable implementations of the Kalman filter use one or more of the following techniques to solve the associated Riccati equation:

Square Root

- **Chapter 6 - Factorization**
  - Solve the associated Riccati equation:
  - 1. Factoring the covariance matrix of state estimation uncertainty (the dependent variable of the Riccati equation) into Cholesky factors (usually, triangu-lar factors), or into modified Cholesky factors (unit triangular and diagonal factors).
    - A *Cholesky factor* of a symmetric nonnegative definite matrix *M* is a matrix *C* such that $CC^T = M$. Cholesky decomposition algorithms solve for *C* that is either upper triangular or lower triangular.
    - The *modified Cholesky decomposition* algorithms solve for diagonal factors and either a lower triangular factor *L* or an upper triangular factor *U* such that $M = UDU^T = LDL^T$, where D and D are diagonal factors with nonnegative diagonal elements.

Square Root

•Chapter 6 - Factorization

- solve the associated Riccati equation:
- 2. Factoring the covariance matrix of measurement noise *R* to reduce the com-putational complexity of the observational update implementation, and factoring the plant noise covariance matrix Q to reduce the computational com-plexity of the temporal update implementation. (These methods effectively "de-correlate" the components of the measurement or plant noise vector.)
- 3. Taking the symmetric matrix square roots of elementary matrices. A symmetric elementary matrix has the form:

$$\boxed{I - \sigma v v^T}$$

where *I* is the $n \times n$ identity matrix, σ is a scalar, and *v* is an n-vector. The symmetric square root of an elementary matrix is also an elementary matrix with the same *v*, but a different value for σ.

Square Root

•Chapter 6 - Factorization

- Solve the associated Riccati equation:
- 4. Factoring general matrices as products of triangular and orthogonal matri-ces. Two general methods are used in Kalman filtering:
- (a) Triangularization (QR decomposition) methods were originally devel-oped for more numerically stable solutions of systems of linear equa-tions. They factor a matrix into the product of an orthogonal matrix Q and a triangular matrix *R.* In the application to Kalman filtering, only the triangular factor is needed. We will call the QR decomposition triangularization, because Q and *R* already have special meanings in Kalman filtering. The two triangularization methods used in Kalman filtering are:

Square Root

•Chapter 6 - Factorization

- Solve the associated Riccati equation:
- 4. Factoring general matrices as products of triangular and orthogonal matrices.
- (a) Triangularization (QR decomposition) methods
- The two triangularization methods used in Kalman filtering are:
  - i. Givens rotations [160] triangularize a matrix by operating on <u>one element at a time</u>. (A modified Givens method due to Gentleman [159] generates diagonal and unit triangular factors.)
  - ii. Householder transformations triangularize a matrix by operating <u>on one row or column at a time</u>.

Square Root

•Chapter 6 - Factorization

- Solve the associated Riccati equation:
- 4. Factoring general matrices as products of triangular and orthogonal matri-ces. Two general methods are
- (b) Gram-Schmidt ortho-normalization is another general method for fac-toring a general matrix into a product of an orthogonal matrix and a triangular matrix. Usually, the triangular factor is not saved. In the application to Kalman filtering, only the triangular factor is saved.

Square Root

•Chapter 6 - Factorization

- Solve the associated Riccati equation:
- 5. Rank one modification algorithms. A "rank one modification" of a symmetric positive definite *n* × n matrix *M* has the form $M \pm \nu\, \nu^T$, where *v* is a n-vector (and therefore has matrix rank equal to one).
- The algorithms compute a Cholesky factor of the modification $M \pm \nu\, \nu^T$, given ν and a Cholesky factor of *M.*

Square Root

•Chapter 6 - Factorization

- Solve the associated Riccati equation:
- 6. Block matrix factorizations of matrix expressions in the Riccati equation. The general approach uses two different factorizations to represents the two aides of an equation, such as
- $CC^T = AA^T + BB^T =$

$$CC^T = AA^T + BB^T = [A\ B]\begin{bmatrix} A^T \\ B^T \end{bmatrix}$$

- The alternative Cholesky factors *C* and [*A B*] must then be related by orthogonal transformations (triangularizations). A *QR* decomposition of [*A B*] will yield a <u>corresponding solution of the Riccati equation</u> in terms of a Cholesky factor of the covariance matrix.

Square Root

•Chapter 6 - Factorization

- Solve the associated Riccati equation:
- 6. Block matrix factorizations of Riccati equation.

$$CC^T = AA^T + BB^T = [A\ B]\begin{bmatrix} A^T \\ B^T \end{bmatrix}$$

- In the example used above, [*A B*] would be called a "1 × 2" block partitioned matrix, because there are <u>one row and two columns</u> of blocks (matri-ces) in the partitioning. Different block dimensions are used to solve different problems:
  - (a) The discrete-time temporal update equation is solved in "square root" form by using alternative 1 × 2 block partitioned Cholesky factors.
  - (b) The observational update equation is solved in "square root" form by using alternative 2 × 2 block partitioned Cholesky factors and modified Cholesky factors representing the observational update equation.
  - (c) The combined temporal/observational update equations are solved in "square root" form by using alternative 2 × 3 block partitioned Cholesky factors of the combined temporal and observational update equations.

Square Root

•Chapter 6 - Factorization

•Cholesky factors and modified Cholesky factors

- The product of a matrix *C* with its own transpose, in the form CCT = *M* is called the *symmetric product* of *C,* and *C* is called a Cholesky factor of M. Strictly speaking, a Cholesky factor is <u>*not*</u> *a* matrix square root, although the terms are often used interchangeably in the literature. (A matrix square root S of *M* is a solution of $M = SS = S^2$, without the transpose.)
- All symmetric nonnegative definite matrices (such as covariance matrices) have Cholesky factors, but the Cholesky factor of a given symmetric nonnegative definite matrix is *not unique*.
- For any *orthogonal matrix T* (i.e., such that $TT^T = I$), the product $\Gamma = C^T$ satisfies the equation $\Gamma\Gamma^T = CTT^TC^T = CC$. That is, Γ is also a Cholesky factor of *M.* However, for suitable constraints on the solution (e.g., being upper triangular or lower triangular with nonnegative diagonal elements), a unique *C* can be found.
- French geodesist André-Louis Cholesky (1875 - 1918) [135], and is called the Cholesky decomposition.

•Chapter 6 - Factorization

•Cholesky factors Use in Kalman Filtering

- Cholesky decom-position methods produce triangular matrix factors (Cholesky factors), and the sparseness of these factors can be exploited in the implementation of the Kalman filter equa-tions. These methods are used for the following purposes:
- 1. In the decomposition of covariance matrices (P, R, and Q) for implementa-tion of "square root" filters.
- 2. In "de-correlating" measurement errors between components of vector-valued measurements, so that the components may be processed sequentially as independent scalar-valued measurements. (See page 218.)
- 3. As part of a numerically stable method for computing matrix expressions containing the factor $(HPH^T + R)^{-1}$ in the conventional form of the Kalman filter. (This matrix inversion can be obviated by the de-correlation methods, however.)
- 4. In Monte Carlo analysis of Kalman filters by simulation. Cholesky factors are used for generating independent random sequences of vectors with pre--specified means and covariance matrices.

•Chapter 6 - Factorization

•Cholesky factors Ex

•Consider the 3 × 3 example for finding a *lower triangular* Cholesky factor $P = CC^T$ for symmetric P:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & 0 & 0 \\ c_{21} & c_{22} & 0 \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} c_{11} & 0 & 0 \\ c_{21} & c_{22} & 0 \\ c_{31} & c_{32} & c_{33} \end{bmatrix}^T$$

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33} \end{bmatrix} = \begin{bmatrix} c_{11}^2 & c_{11}c_{21} & c_{11}c_{31} \\ c_{11}c_{21} & c_{21}^2 + c_{22}^2 & c_{21}c_{31} + c_{22}c_{32} \\ c_{11}c_{31} & c_{21}c_{31} + c_{22}c_{32} & c_{31}^2 + c_{32}^2 + c_{33}^2 \end{bmatrix}$$

•Chapter 6 - Factorization

• Modified Cholesky factors Ex

•Uses the UD formulation

- Given M, a symmetric, positive definite $m \times m$ matrix, computes U and D, modified Cholesky factors of M, such that U is a unit upper triangular matrix, D is a diagonal matrix, and $M = UDU^T$.

•DECORRELATION

R = UDUT,

•Chapter 6 - Factorization

• Modified Cholesky factors Ex

• DECORRELATION *Value*

TABLE 6.15 OPERATIONS FOR SEQUENTIAL PROCESSING OF MEASUREMENTS

Operation	flops
$H \times P(-)$	n^2
$H \times (HP(-))^T + R$	n
$[H(HP(-))^T + R]^{-1}$	1
$[H(HP(-))^T + R]^{-1} \times [HP(-)]$	n
$P(-) - [HP(-)] \times [H(HP(-))^T + R]^{-1} [HP(-)]$	$(1/2)n^2 + (1/2)n$
Total (per component) $\times\, \ell$ components	$(3/2\, n^2 + 5/2\, n + 1) \times \ell$
+ de-correlation complexity	$(2/3)\ell^3 + \ell^2 - (5/3)\ell + (1/2)\ell^2 n - (1/2)\ell n$
Total	$(2/3)\ell^3 + \ell^2 - (2/3)\ell + (1/2)\ell^2 n + 2\ell n\, (3/2)\, \ell n^2$

The **computational advantage** of the de-correlation approach is

$$(1/3)\ell^3 - (1/2)\ell^2 + (7/6)\ell + \ell n + 2\ell^2 n + \ell n^2 \quad \text{flops.}$$

Square Root

•Chapter 6 – Square Root

• Potter

•An elementary matrix is a matrix of the form $I – s\,v\,w^{T}$, where

- **I is an identity matrix,**
- **s is a scalar, and**
- **v, w are column vectors of the same row dimension as *I*.**

- **Elementary matrices have the property that their products are also elementary matrices. Their squares are also elementary matrices, with the same vector values (v, w) but with different scalar values (s).**

Square Root

•Chapter 6 – Square Root

• Potter

•Symmetric elementary matrices. An elementary matrix is symmetric if *v* **=** w**.** The squares of such matrices have the same format:

$$(I - \sigma \mathbf{v}\mathbf{v}^T)^2 = (I - \sigma \mathbf{v}\mathbf{v}^T)(I - \sigma \mathbf{v}\mathbf{v}^T)$$

$$(I - \sigma \mathbf{v}\mathbf{v}^T)^2 = I - 2\sigma \mathbf{v}\mathbf{v}^T + \sigma^2 |\mathbf{v}|^2\, \mathbf{v}\mathbf{v}^T$$

$$(I - \sigma \mathbf{v}\mathbf{v}^T)^2 = I - (2\sigma - \sigma^2 |\mathbf{v}|^2)\, \mathbf{v}\mathbf{v}^T$$

$$(I - \sigma \mathbf{v}\mathbf{v}^T)^2 = I - s\,\mathbf{v}\mathbf{v}^T$$

$$s = (2\sigma - \sigma^2 |\mathbf{v}|^2)$$

Square Root

•Chapter 6 – Square Root

• Potter

•The symmetric square root of a symmetric elementary matrix. One can invert the last equation above and take the square root of the symmetric elementary matrix ($I - s\,v\,v^T$). This is done by solving the <u>scalar quadratic equation</u>

$$s = 2\sigma - \sigma^2 |\mathbf{v}|^2$$

$$\sigma^2 |\mathbf{v}|^2 - 2\sigma + s = 0$$

$$\sqrt{I - s\,\mathbf{v}\mathbf{v}^T} = I - \sigma \mathbf{v}\mathbf{v}^T$$

$$\sigma = \frac{1 + \sqrt{1 - s|\mathbf{v}|^2}}{|\mathbf{v}|^2}$$

to obtain the solution

In order that this square root be a real matrix, it is necessary that the radicand

$$\boxed{1 - s|\mathbf{v}|^2 \geq 0.}$$

Square Root

•Chapter 6 – Square Root

• Triangularization

• (QR decomposition) of $A = [\phi_k\, C_k(+) \mid C\, Q]$. It is a theorem of linear algebra that any general matrix *A* can be represented as a product

$$A = C_{k+1}(-)\, T$$

of a triangular matrix $C_{k+1}(-)$ and an orthogonal matrix T. This type of decomposition is called *QR decomposition or triangularization.* By means of this triangularization, the symmetric matrix product factorization

$$P_{k+1}(-) = A A^T$$

$$P_{k+1}(-) = (C_{k+1}(-)\, T)(C_{k+1}(-)\, T)^T$$

$$P_{k+1}(-) = C_{k+1}(-)\, T T^T\, C_{k+1}^T(-)$$

$$P_{k+1}(-) = C_{k+1}(-)\, (T T^T)\, C_{k+1}^T(-)$$

$$P_{k+1}(-) = C_{k+1}(-)\, C_{k+1}^T(-)$$

Square Root

•Chapter 6 – Square Root

- **Triangularization**
- **(USe)**

Temporal updates **of Cholesky factors of the covariance matrix of estimation uncertainty, as described above**

Observational updates **of Cholesky factors of the estimation information ma-trix, as described in §6.5**

Combined updates **(observational and temporal) of Cholesky factors of the covariance matrix of estimation uncertainty, as described in §6.7**

A modified Givens rotation due to W Morven Gentleman [159] is used for the temporal updating of modified Cholesky factors of the covariance matrix.

Square Root

•Chapter 6 – Square Root

- **Triangularization (Givens - Rotation)**

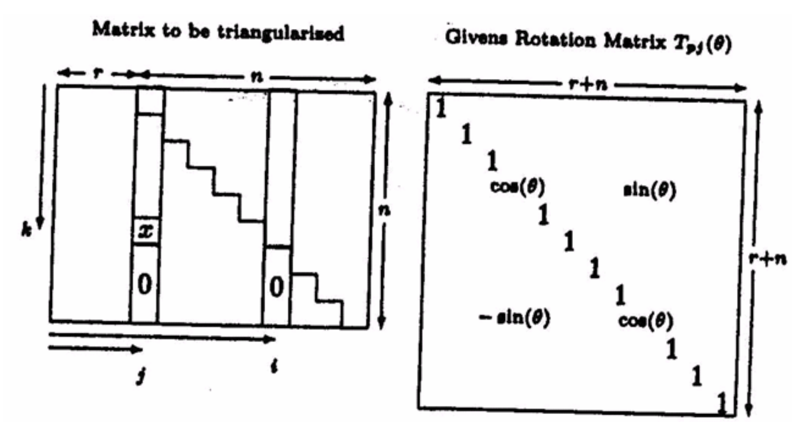
$$T_{ij}(\Theta) = \begin{array}{c|cc|} & j & i \\ \hline j & & \\ i & & \\ \hline \end{array}$$

$$T_{ij}(\Theta) = \begin{bmatrix} 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cos(\Theta) & 0 & \cdots & 0 & \sin(\Theta) & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -\sin(\Theta) & 0 & \cdots & 0 & \cos(\Theta) & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Square Root

•Chapter 6 – Square Root

- **Triangularization (Givens - Rotation)**



Square Root

•Chapter 6 – Square Root

- **Triangularization (Householder)**

•Alston S. Householder It uses an elementary matrix of the

$$T(v) = I - \frac{2}{v^T v} v v^T$$

where v is a column vector and I is the identity matrix of the same dimension. This particular form of the elementary matrix is called a ***Householder reflection*****,** ***House-holder transformation*****, or** ***Householder matrix.*** **Note that Householder transformation matrices are always** ***symmetric.*** **They are also** ***orthogonal*****, for**

$$T(v)\,T^T(v) = \left(I - \frac{2}{v^T v} v v^T\right)\left(I - \frac{2}{v^T v} v v^T\right)$$

$$T(v)\,T^T(v) = I - \frac{4}{v^T v} v v^T + \frac{4}{(v^T v)^2} v (v^T v) v^T$$

$$T(v)\,T^T(v) = I$$

They are called "reflections" because they transform any matrix a into its "mirror reflection" in the plane (or hyperplane) normal to the vector v

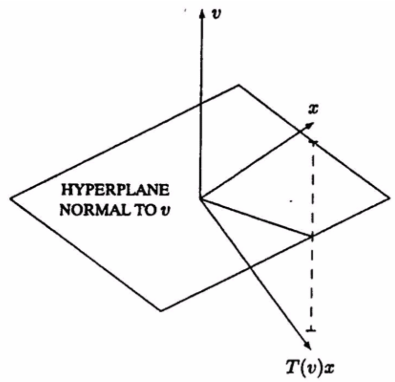
Square Root

•Chapter 6 – Square Root

- **Triangularization (Householder)**

•**Alston S. Householder It uses an elementary matrix of the**

$$T(v) = I - \frac{2}{v^T v} v v^T$$

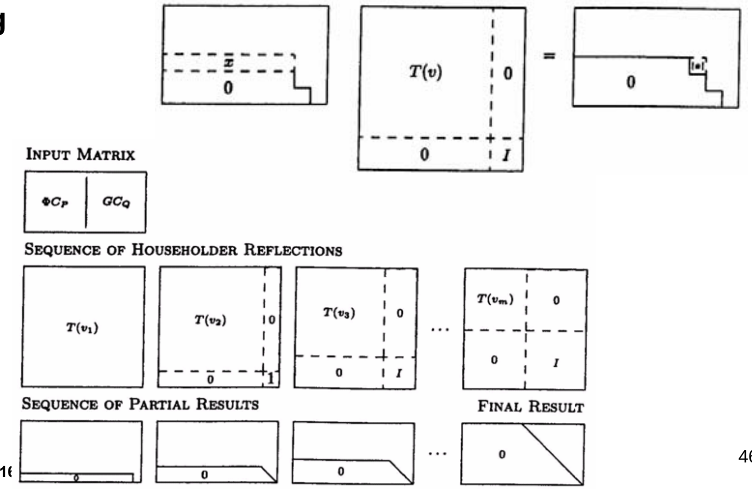


Square Root

•Chapter 6 – Square Root

- **Triangularization (Householder - Upper)**
- **Zeroing**

•**Upper**



Square Root

•Chapter 6 – Square Root

- **Triangularization (Potter)**

$$P(-) \stackrel{\text{def}}{=} C(-)\,C^T(-)$$

$$P(+) \stackrel{\text{def}}{=} C(+)\,C^T(+)$$

I_n is the $n \times n$ identity matrix
$V = C^T(-)H^T$ is an $n \times \ell$ general matrix
n is the dimension of the state vector
ℓ is the dimension of the measurement vector

so that the observational update equation

$$P(+) = P(-) - P(-)H^T\,[\,HP(-)H^T + R]^{-1}\,HP(-)$$

could be partially factored as

$$C(+)C^T(+) = C(-)C^T(-) - C(-)C^T(-)\;H^T\,[HC(-)C^T(-)\,H^T + R]^{-1}HC(-)C^T(-)$$

$$C(+)C^T(+) = C(-)C^T(-) - C(-)V\,[V^TV + R]^{-1}\,V^TC^T(-)$$

$$C(+)C^T(+) = C(-)\,\{I_n - V\,[V^TV + R]^{-1}\,V^T\}\,C^T(-)\,,$$

Square Root

•Chapter 6 – Square Root

- **Triangularization (Potter)**

$$\{I_n - V\,[V^TV + R]^{-1}\,V^T\}$$

For the case that the measurement is a scalar ($\ell = 1$), Potter was able to factor it in the form

$$I_n - V\,[V^TV + R]^{-1}\,V^T = WW^T,$$

so that the resulting equation,

$$C(+)C^T(+) = C(-)\,\{\,WW^T\,\}\,C^T(-) \qquad C(+)C^T(+) = \{C(-)W\}\,\{C(-)W\}^T$$

could be solved for the *a posteriori* Cholesky factor of P(+), as $\quad C(+) = C(-)W$

When the measurement is a scalar, the expression to be factored is a symmetric **elementary matrix** of the form

$$\boxed{I_n - \frac{v\,v^T}{R + |v|^2}}$$

where R is a positive scalar and $v = C^T(-)H^T$ *is* a column n-vector.

Square Root

•Chapter 6 – Square Root

- **Triangularization (Potter)**

•Potter square root observational update formula :

•the solution for the a *posteriori* Cholesky factor C(+) of the covariance matrix *P* can be expressed as the product

$$C(+)C^T(+) = P(+)$$
$$C(+)C^T(+) = C(-)\,(I - \sigma\,\mathbf{v}\,\mathbf{v}^T)\,C^T(-)$$
$$C(+)C^T(+) = C(-)\,(I - \sigma\,\mathbf{v}\,\mathbf{v}^T)\,(I - \sigma\,\mathbf{v}\,\mathbf{v}^T)^T\,C^T(-)$$

which can be factored as with

$$C(+) = C(-)\,(I - \sigma\,\mathbf{v}\,\mathbf{v}^T)$$

with

$$\sigma = \frac{1 + \sqrt{1 - s\,|\mathbf{v}|^2}}{|\mathbf{v}|^2} \qquad \sigma = \frac{1 + \sqrt{\dfrac{R}{R + |\mathbf{v}|^2}}}{|\mathbf{v}|^2}$$

Square Root

•Chapter 6 – Square Root

- **Triangularization (Carlson) - FAST**
- **This algorithm is due to Neal A. Carlson. It generates an upper triangular Cholesky factor *W* for the Potter factorization, and has generally lower computational complexity than the Potter algorithm. It is a specialized and simplified form of an al-gorithm used by Agee and Turner [103] for Kalman filtering. It is a rank-one mod-ification algorithm, like the Potter algorithm, but it produces a triangular Cholesky factor. It can be derived from the following lemma:**

•Lemma 4 (Carlson). If *W* is an upper triangular n × n matrix such that

$$W\,W^T = I - \frac{\mathbf{v}\,\mathbf{v}^T}{R + |\mathbf{v}|^2} \;\rightarrow\; \sum_{k=m}^{j} W_{ik}\,W_{mk} = \Delta_{im} - \frac{\mathbf{v}_i\,\mathbf{v}_m}{R + \sum_{k=1}^{j} \mathbf{v}_k^2}$$

- **for all *i, m, j* such that 1 ≤ i ≤ *m* ≤ j ≤ n.**

Square Root

•Chapter 6 – Square Root

- **Triangularization (Carlson) – FAST - UPPER**

$$W_{ij}\,W_{jj} = \Delta_{ij} - \frac{\mathbf{v}_i\,\mathbf{v}_j}{R + \sum_{k=1}^{j} \mathbf{v}_k^2}$$

$$W_{ij} = \begin{cases} 0 & i > j \\ \dfrac{\sqrt{R + \sum_{k=1}^{j-1} \mathbf{v}_k^2}}{\sqrt{R + \sum_{k=1}^{j} \mathbf{v}_k^2}} & i = j \\ \dfrac{-\mathbf{v}_i\,\mathbf{v}_j}{(R + \sum_{k=1}^{j-1} \mathbf{v}_k^2)(R + \sum_{k=1}^{j} \mathbf{v}_k^2)} & i < j \end{cases}$$

$$C_{ij}(+) = \sum_{k=i}^{j} C_{ik}(-)\,W_{kj} \;+ \text{terms with zero factors}$$

$$C_{ij}(+) = C_{ij}(-)\,W_{jj} + \sum_{k=i}^{j-1} C_{ik}(-)\,W_{kj}$$

$$\boxed{C_{ij}(+) = \frac{1}{\sqrt{R + \sum_{k=1}^{j} \mathbf{v}_k^2}}\left[C_{ij}(-)\sqrt{R + \sum_{k=1}^{j-1} \mathbf{v}_k^2} - \frac{\mathbf{v}_j \sum_{k=1}^{j} C_{ik}(-)\,\mathbf{v}_k}{\sqrt{R + \sum_{k=1}^{j-1} \mathbf{v}_k^2}} \right]}$$

Square Root

•Chapter 6 – Square Root

- **Triangularization (Bierman) – UD**

•Partial UD factorization of the covariance equations. In a manner simi-lar to the case with Cholesky factors for scalar-valued measurements, the conventional form of the observational update of the covariance matrix:

$$P(+) = P(-) - \frac{P(-)\,H^T H\,P(-)}{R + H\,P(-)\,H^T}$$

can be partially factored in terms of *UD* factors:

$$P(-) \overset{\text{def}}{=} U(-)\,D(-)\,U^T(-) \qquad \mathbf{v} = U^T(-)H^T$$
$$P(+) \overset{\text{def}}{=} U(+)\,D(+)\,U^T(+)$$
$$U(+)\,D(+)\,U^T(+) = U(-)\,D(-)\,U^T(-) - \frac{U(-)\,D(-)\,U^T(-)\,H^T H\,U(-)\,D(-)}{R + H\,U(-)\,D(-)U^T(-)H^T}\,U^T(-)$$
$$U(+)\,D(+)\,U^T(+) = U(-)\,D(-)\,U^T(-) - \frac{U(-)\,D(-)\,\mathbf{v}\,\mathbf{v}^T D(-)\,U^T(-)}{R + \mathbf{v}^T D(-)\,\mathbf{v}}$$
$$U(+)\,D(+)\,U^T(+) = U(-)\left[D(-) - \frac{D(-)\,\mathbf{v}\,\mathbf{v}^T D(-)}{R + \mathbf{v}^T D(-)\,\mathbf{v}} \right] U^T(-)$$

Square Root

•Chapter 6 – Square Root

- **Triangularization (Bierman) – UD**
  - *Proofs in Book*
- ***Lemma 5**. If for m such that $l < m \le n$, and for all i, j such that $l \le i \le j \le m$*

$$\sum_{k=j}^{m} B_{ik} D_{kk}(+) B_{jk} = D_{ij}(-) - \frac{D_{ii}(-)\,\mathbf{v}_i D_{jj}(-)\,\mathbf{v}_j}{R + \sum_{k=1}^{m} \mathbf{v}_k^2 D_{kk}(-)}$$

$$\sum_{k=j}^{m-1} B_{ik} D_{kk}(+) B_{jk} = D_{ij}(-) - \frac{D_{ii}(-)\,\mathbf{v}_i D_{jj}(-)\,\mathbf{v}_j}{R + \sum_{k=1}^{m-1} \mathbf{v}_k^2 D_{kk}(-)}$$

- **Lemma 6**. If

$$B\,D(+)\,B^T = D(-) - \frac{D(-)\,\mathbf{v}\,\mathbf{v}^T D(-)}{R + \mathbf{v}^T D(-)\,\mathbf{v}}$$

then Equation 6.104 holds for all *m* such that $1 < m \le n$.

Square Root

•Chapter 6 – Square Root

- **Triangularization (Bierman) – UD**
  - *Proofs in Book*
- ***Lemma 7. If***

$$B\,D(+)\,B^T = D(-) - \frac{D(-)\,\mathbf{v}\,\mathbf{v}^T D(-)}{R + \mathbf{v}^T D(-)\,\mathbf{v}}$$

then for $1 \le j \le n$

$$D_{jj}(+) = D_{ii}(-)\left[\frac{R + \sum_{k=1}^{j-1} \mathbf{v}_k^2 D_{kk}(-)}{R + \sum_{k=1}^{j} \mathbf{v}_k^2 D_{kk}(-)}\right]$$

and for $1 \le i < j \le n$,

$$B_{ij} = \frac{D_{ii}(-)\,\mathbf{v}_i\,\mathbf{v}_j}{R + \sum_{k=1}^{j-1} \mathbf{v}_k^2 D_{kk}(-)}$$

Square Root

•Chapter 6 – Square Root

- **Triangularization (??)**
  - ***A Square Partitioned Cholesky Factorization***
- **The following factorization of the observational update equations first appeared in a paper by Kaminski, Bryson, and Schmidt [179]. The derivation presented here has been modified to <u>yield upper triangular Cholesky factors</u>.**

Let C_N be a Cholesky factor of the covariance matrix of the innovations $v_k = z_k - H_k \hat{x}_k(-)$, defined as

$$N_k = E\langle (z_k - H_k \hat{x}_k(-))(z_k - H_k \hat{x}_k(-))^T \rangle \qquad (6.144)$$

$$N_k = H_k P_k(-) H_k^T + R_k \qquad (6.145)$$

[BLASCH] $N_k = S_k$

and let $C_{P(-)}$, $C_{P(+)}$, and C_R be Cholesky factors of $P_k(-)$, $P_k(+)$, and R_k, respectively. Then the symmetric partitioned $(n+\ell) \times (n+\ell)$ Cholesky factorization

$$\begin{bmatrix} P_k(-) & P_k(-)H_k^T \\ H_k P_k(-) & N_k \end{bmatrix} = \begin{bmatrix} C_{P(-)} & 0 \\ H_k C_{P(-)} & C_N \end{bmatrix} \begin{bmatrix} C^T_{P(-)} & C^T_{P(-)} H_k^T \\ 0 & C_N \end{bmatrix} \qquad (6.146)$$

$$\begin{bmatrix} P_k(-) & P_k(-)H_k^T \\ H_k P_k(-) & N_k \end{bmatrix} = \begin{bmatrix} C_{P(-)} & 0 \\ H_k C_{P(-)} & C_N \end{bmatrix} \begin{bmatrix} C_{P(-)} & 0 \\ H_k C_{P(-)} & C_N \end{bmatrix}^T \qquad (6.147)$$

Square Root

•Chapter 6 – Square Root

- **Triangularization (??)**
  - ***A Square Partitioned Cholesky Factorization***
  - **<u>yield upper triangular Cholesky factors</u>.**

$$\begin{bmatrix} P_k(-) & P_k(-)H_k^T \\ H_k P_k(-) & N_k \end{bmatrix} = \begin{bmatrix} C_{P(-)} & 0 \\ H_k C_{P(-)} & C_N \end{bmatrix} \begin{bmatrix} C^T_{P(-)} & C^T_{P(-)} H_k^T \\ 0 & C_N \end{bmatrix} \qquad (6.146)$$

$$\begin{bmatrix} P_k(-) & P_k(-)H_k^T \\ H_k P_k(-) & N_k \end{bmatrix} = \begin{bmatrix} C_{P(-)} & 0 \\ H_k C_{P(-)} & C_N \end{bmatrix} \begin{bmatrix} C_{P(-)} & 0 \\ H_k C_{P(-)} & C_N \end{bmatrix}^T \qquad (6.147)$$

:an also be factored as

$$\begin{bmatrix} P_k(-) & P_k(-)H_k^T \\ H_k P_k(-) & N_k \end{bmatrix} = \begin{bmatrix} C_{P(+)} & P_k(-)H_k^T C_N^{-T} \\ 0 & C_N \end{bmatrix} \begin{bmatrix} C^T_{P(+)} & 0 \\ C_N^{-1} H_k P^T_k(-) & C_N^T \end{bmatrix} \qquad (6.148)$$

$$\begin{bmatrix} P_k(-) & P_k(-)H_k^T \\ H_k P_k(-) & N_k \end{bmatrix} = \begin{bmatrix} C_{P(+)} & P_k(-)H_k^T C_N^{-T} \\ 0 & C_N \end{bmatrix} \begin{bmatrix} C_{P(+)} & P_k(-) H_k^T C_N^{-T} \\ 0 & C_N \end{bmatrix}^T \qquad (6.149)$$

Consequently, the two Cholesky factorizations are related by an orthogonal transformation. That is, if $\mathcal{O}$ is any orthogonal triangularizing transformation such that the conformably partitioned product

$$\begin{bmatrix} C_{P(-)} & 0 \\ H_k C_{P(-)} & C_N \end{bmatrix} \mathcal{O} = \begin{bmatrix} C_a & B \\ 0 & C_c \end{bmatrix} \qquad (6.150)$$

:s *upper triangular,* then the upper triangular sub-matrices $C_a = C_{P(+)}$ $C_c = C_N$

Square Root

•Chapter 6 – Square Root

• Triangularization (??)

• ***A Square Partitioned Cholesky Factorization***

• **<u>yield upper triangular Cholesky factors</u>.**

and the sub-matrix B satisfies the equation

$$B\, C_C^{-T} = [\,P_k(-)\, H_k^T\, C_N^{-T}\,]\, C_N^{-1}$$
$$B\, C_C^{-T} = P_k(-)\, H_k^T\, (C_N\, C_N^T)^{-1}$$
$$B\, C_C^{-T} = K_K$$

the Kalman gain. Note, however, that the observational update of the estimated state

$$\hat{x}_k(+) = \hat{x}_k(-) + \Delta\hat{x}_k$$

$$\Delta\hat{x}_k = K_k[\,z_k - H_k\,\hat{x}_k(-)\,]$$

$$\Delta\hat{x}_k = B\, C_C^{-T}\,[\,z_k - H_k\,\hat{x}_k(-)\,]$$

can be implemented more robustly by solving the triangular system

$$\boxed{C_C^T\, y = [\,z_k - H_k\,\hat{x}_k(-)\,]}$$

for y, using back substitution, then taking the product B y .

Square Root

•Chapter 6 – Square Root

• Triangularization (Kaliath)

• ***A Square Partitioned UD Factorization***

Let the UDU^T factorization of the covariance of innovations be

$$U_N D_N U_N^T = N_k$$

$$U_N D_N U_N^T = H_k\, P_k(-)\, H_k^T + R_k$$

Then the symmetric partitioned $(n+\ell)\times(n+\ell)$ matrix

$$\begin{bmatrix} P_k(-) & P_k(-)\,H_k^T \\ H_k\,P_k(-) & N \end{bmatrix} = \begin{bmatrix} U_{P(-)} & 0 \\ H_k\,U_{P(-)} & U_N \end{bmatrix}\begin{bmatrix} D_{P(-)} & 0 \\ 0 & D_N \end{bmatrix}\begin{bmatrix} U_{P(-)}^T & U_{P(-)}^T(-)\,H_k^T \\ 0 & U_N^T \end{bmatrix}$$

$$\begin{bmatrix} P_k(-) & P_k(-)\,H_k^T \\ H_k\,P_k(-) & N \end{bmatrix} = \begin{bmatrix} U_{P(-)} & 0 \\ H_k\,U_{P(-)} & U_N \end{bmatrix}\begin{bmatrix} D_{P(-)} & 0 \\ 0 & D_N \end{bmatrix}\begin{bmatrix} U_{P(-)} & 0 \\ H_k\,U_{P(-)} & U_N \end{bmatrix}^T$$

can also be factored as

$$\begin{bmatrix} P_k(-) & P_k(-)\,H_k^T \\ H_k\,P_k(-) & N \end{bmatrix} = \begin{bmatrix} U_{P(+)} & P_k(-)\,H_k^T\,U_N^{-1}\,D_N^{-1} \\ 0 & U_N \end{bmatrix}\begin{bmatrix} D_{P(+)} & 0 \\ 0 & D_N \end{bmatrix}\begin{bmatrix} U_{P(+)}^T & 0 \\ D_N^{-1}\,U_N^{-T}\,H_k P_k(-) & U_N^T \end{bmatrix}$$

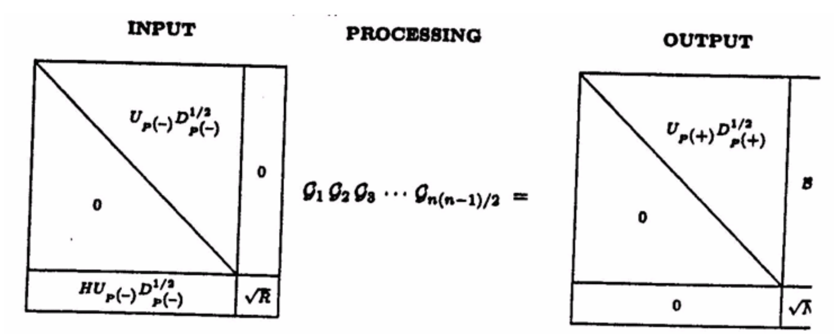
$$\begin{bmatrix} P_k(-) & P_k(-)\,H_k^T \\ H_k\,P_k(-) & N \end{bmatrix} = \begin{bmatrix} U_{P(+)} & P_k(-)\,H_k^T\,U_N^{-1}\,D_N^{-1} \\ 0 & U_N \end{bmatrix}\begin{bmatrix} D_{P(+)} & 0 \\ 0 & D_N \end{bmatrix}\begin{bmatrix} U_{P(+)} & P_k(-)H_k^T U_N^{-1}\,D_N^{-1} \\ 0 & U_N \end{bmatrix}^T$$

Square Root

•Chapter 6 – Square Root

• Triangularization (Kaliath)

• ***A Square Partitioned UD Factorization***



Square Root

•Chapter 6 – Square Root

• Triangularization (Schmidt)

•

A non-square, non-triangular Cholesky factor of $P_k(-)$. If C_P is a Cholesky factor of $P_{k-1}(-)$ and C_Q is a Cholesky factor of Q_k, then the partitioned $n \times (n+q)$ matrix

$$A = [\,G_{k-1}\,C_Q \mid \Phi_{k-1}\,C_P\,] \tag{6.181}$$

has the $n \times n$ symmetric matrix product value

$$AA^T = [\,G_{k-1}\,C_Q \mid \Phi_{k-1}\,C_P\,]\,[\,G_{k-1}\,C_Q \mid \Phi_{k-1}\,C_P\,]^T \tag{6.182}$$

$$AA^T = \Phi_{k-1}[\,C_P C_P^T\,]\,\Phi_{k-1}^T + G_{k-1}[\,C_Q\,C_Q^T]\,G_{k-1}^T \tag{6.183}$$

$$AA^T = \Phi_{k-1}\,[P_{k-1}(+)]\,\Phi_{k-1}^T + G_{k-1}Q\,G_{k-1}^T \tag{6.184}$$

$$AA^T = P_k(-) \tag{6.185}$$

That is, A is a *non-square, non-triangular* Cholesky factor of $P_k(-)$. If *only* it were square and triangular, it would be the kind of Cholesky factor we are looking for. It is not, but fortunately there are algorithmic procedures for modifying A to that format.

Square Root

•Chapter 6 – Square Root

• Triangularization (Thorton - MWGS)

•This *UD* factorization is due to Catherine Thornton [121]. It is also called *modi-fied weighted Gram-Schmidt (MWGS)* orthogonalization. [Gram-Schmidt *ortho-normalization* is a procedure for generating a set or "unit normal" vectors as linear combinations of a set of linearly independent vectors. That is, the resulting vectors are mutually orthogonal and have unit length. The procedure without the unit length property is called Gram-Schmidt *orthogonalization.* These algorithmic methods were derived by Jorgen Pedersen Gram (1850 - 1916) and Erhard Schmidt (1876 - 1959).] It uses a factorization algorithm due to Björck [137] that is actually quite different from the conventional Gram-Schmidt orthogonalization algorithm, and more robust against round-off errors. However, the algebraic properties of Gram-Schmidt orthogonalization are useful for deriving the factorization.

•

Square Root

•Chapter 6 – Square Root

• Triangularization (Thorton - MWGS)

• *UD*

The Thornton temporal update for UD factors uses triangularization of the Q matrix (if it is not already diagonal) in the form $Q = GD_QG^T$, where D_Q is a diagonal matrix. If we let the matrices

$$A = \begin{bmatrix} U_{k-1}^T(+)\,\Phi_{k-1}^T \\ G_{k-1}^T \end{bmatrix} \qquad L^T D_\beta L = P_{k-1}(-) \tag{6.198}$$

$$U_k(-) = L^T$$

$$D_w = \begin{bmatrix} D_{k-1}(+) & 0 \\ 0 & D_{Q_{k-1}} \end{bmatrix} \qquad D_k(-) = D_\beta \tag{6.199}$$

then the modified weighted Gram-Schmidt orthogonalization procedure will produce a unit lower triangular $n \times n$ matrix L^{-1} and a diagonal matrix D_β such that

$$A = BL \tag{6.200}$$
$$L^T D_\beta L = L^T B^T D_w B L \tag{6.201}$$
$$L^T D_\beta L = (BL)^T D_w B L \tag{6.202}$$
$$L^T D_\beta L = A^T D_w A \tag{6.203}$$
$$L^T D_\beta L = [\,\Phi_{k-1} U_{k-1}(+)\;\; G_{k-1}\,] \begin{bmatrix} D_{k-1}(+) & 0 \\ 0 & D_{Q_{k-1}} \end{bmatrix} \begin{bmatrix} U_{k-1}^T(+)\,\Phi_{k-1}^T \\ G_{k-1}^T \end{bmatrix} \tag{6.204}$$
$$L^T D_\beta L = \Phi_{k-1} U_{k-1}(+)\, D_{k-1}(+)\, U_{k-1}^T(+)\, \Phi_{k-1}^T + G_{k-1} D_{Q_{k-1}} G_{k-1}^T \tag{6.205}$$
$$L^T D_\beta L = \Phi_{k-1} P_{k-1}(+)\, \Phi_{k-1}^T + Q_{k-1} \tag{6.206}$$

Square Root

•Chapter 6 – Square Root

• Triangularization (Kailath - Observation)

• *UD*

Square Root

•Chapter 6 – Square Root

• Information Filter

•The inverse of the covariance matrix of estimation uncertainty is called the *Informati-on matrix*:

$$Y \stackrel{\text{def}}{=} P^{-1}$$

•This is also called the *Fisher Information matrix*, named after the English statistician Ronald Aylmer Fisher (1890 -1962). More generally, for distributions with differentiable probability density functions, the Information matrix is defined as the matrix of second-order derivatives of the logarithm of the probability density with respect to the variates. For Gaussian distributions, this equals the inverse of the covariance matrix.

•Implementation using Y (or its Cholesky factors) rather than *P* (or its Cholesky fac-tors) are called *Information filters*. (Implementations using *P* are also called *covari-ance filters*.)

Book Code (2)

CHAPTER5

- exam53.m demonstrates the extended Kalman filter in the estimation of position, velocity, and damping factor of a damped harmonic oscillator with constant forcing function using Example 5.3.
- exam53x.m and Exam53y.m descriptions are given in whatsup.doc.

CHAPTER6

- Shootout.m is a demonstration of the efficiency of nine alternative observational updates, using the ill-conditioned problem in Example 6.2.)
- Carlson.m - N. A. Carlson's observational update method.
- Utchol.m - upper triangular Cholesky decomposition (Matlab does lower)
- Potter.m - J. E. Potter's observational update method.
- Joseph.m - P. D. Joseph's observational update method.
- Josephb.m - P. D. Joseph's observational update, modified by Bierman.
- Josephdv.m - P. D. Joseph's observational update, modified by De Vries.
- Bierman.m - G. J. Bierman's observational update method.
- Thornton.m - C. L. Thornton's temporal update method.

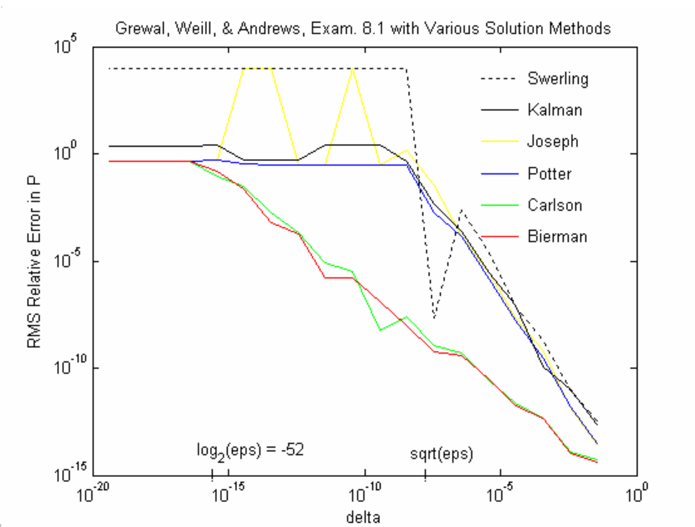
Square Root Comparison

•shootout.m demonstrates SR

```
% These alternative approaches are each evaluated for delta
% ranging from several order of magnitude larger than eps^(2/3)
% to several orders of magnitude smaller than eps^(2/3)
%
 for k=1:19,
 d(k) = delta*10^(10-k);
%
% COMMON PARAMETERS TO ALL METHODS
%
 P  = eye(3);
 D0 = eye(3);
 U0 = eye(3);
 R  = d(k)^2*eye(2);
 H  = [1,1,1; 1,1, 1+d(k) ];
 x0 = [1;1;-1];
 z  = H*x0 + d(k)*[randn;randn];
```


Square Root Comparison

•shootout.m demonstrates SR



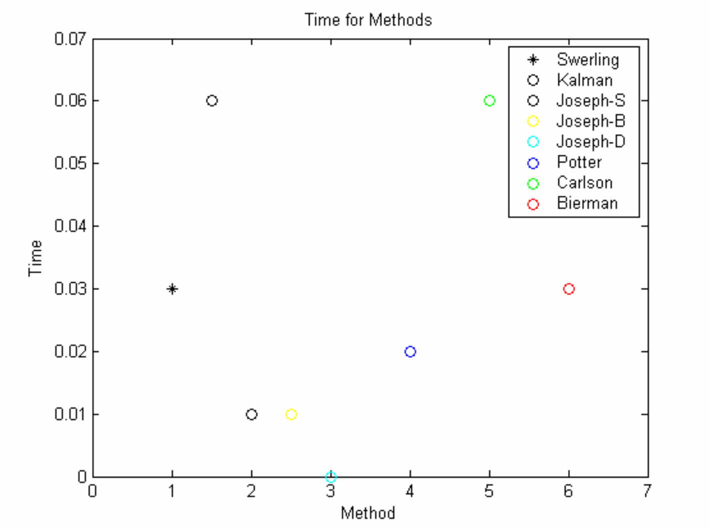
Square Root Comparison

•shootout.m CPUTIME

```
t1 = cputime;
  K  = P*H'/(H*P*H'+R);
  PK = P - K*H*P;
tCK = tCK + cputime - t1;
    xK = x0 + K*(z - H*x0);
%
% SWERLING IMPLEMENTATION
%
t2 = cputime;
  PS = inv(P + (H'/R)*H);
tSW = tSW + cputime - t2;
```

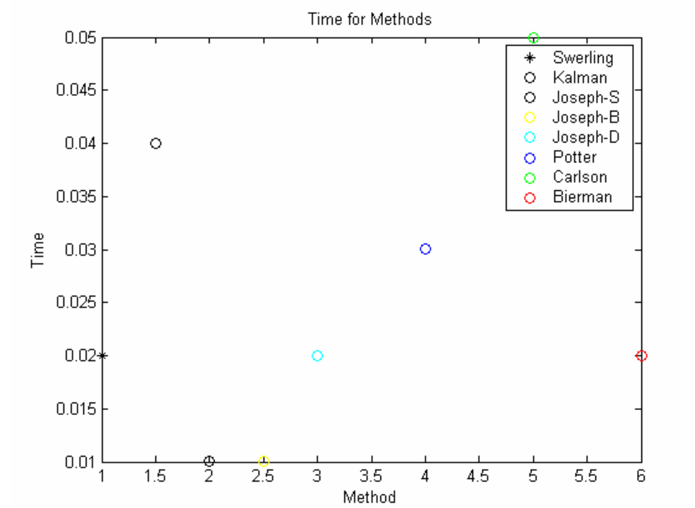
Square Root Comparison

- **shootout.m – plotted CPUTIME**



Square Root Comparison

- **shootout.m – plotted CPUTIME**



Square Root Comparison

- **1 Time Comp and 1 Obs. Update** Maybeck, 1979

Filter	Adds (all times $\frac{1}{6}$)	Multiplies (all times $\frac{1}{6}$)	Divides	Square roots
Conventional Kalman	$9n^3 + 3n^2(3m + s - 1) + n(15m + 3s - 6)$	$9n^3 + 3n^2(3m + s + 3) + n(27m + 9s)$	m	0
Joseph form Kalman	$18n^3 + 3n^2(5m + s - 10) + n(9m^2 + 6m + 3s) + 3m^3 - 6m^2 + 3m$	$18n^3 + 3n^2(5m + s + 4) + n(9m^2 + 24m + 9s) + 3m^3 + 9m^2 - 6m$	$2m - 1$	0
Potter covariance square root (MGS)	$12n^3 + 3n^2(6m + 2s) + n(6m - 6) + 6m$	$12n^3 + 3n^2(5m + 2s + 2) + n(24m + 6s) + 12m$	$n + 2m$	$n + m$
Potter covariance square root (Householder)	$10n^3 + 3n^2(6m + 2s - 1) + n(6m + 5) + 6m$	$10n^3 + 3n^2(5m + 2s + 2) + n(24m + 6s + 8) + 12m$	$n + 2m$	$n + m$
Carlson covariance square root (RSS)	$5n^3 + 3n^2(3m + s + 1) + n(9m + 3s - 14) + 2s^3 + 4s$	$5n^3 + 3n^2(4m + s + 3) + n(30m + 9s - 2) + 2s^3 + 6s^2 - 2s$	$2mn + s$	$mn + s$
Carlson covariance square root (MGS)	$9n^3 + 3n^2(3m + s - 1) + 3n(3m + 3s - 8) + 2s^3 + 6s^2 + 4s$	$9n^3 + 3n^2(4m + s + 2) + 3n(10m + 5s - 7) + 2s^3 + 12s^2 + 4s$	$2mn + s$	$mn + s$
Inverse covariance	$10n^3 + 3n^2(m + 3s + 2) + n(9m + 9s - 16)$	$10n^3 + 3n^2(m + 3s + 6) + n(15m + 21s - 10)$	$2s - 1$	0
Inverse covariance square root	$9n^3 + 3n^2(2m + 6s + 5) + n(12m + 6s - 6)$	$9n^3 + 3n^2(2m + 6s + 6) + n(12m + 24s + 3) + 6s$	$2n + 2s$	$n + s$
U–D factor	$9n^3 + 3n^2(3m + 2s + 2) + 3n(3m + 1)$	$9n^3 + 3n^2(3m - 2s + 7) + 3n(m + 4s - 4) - 6s$	$n(m + 1) - 1$	0

Square Root Comparison

- **Computation Time**

Maybeck, 1979

*Operations for One Total Filter Recursion**

Filter	Adds	Multiplies	Divides	Square roots	Time (msec)
Conventional Kalman	2340	2690	2	0	17.36
Joseph form Kalman	3631	4498	3	0	28.27
Potter covariance square root (MGS)	3612	3884	14	12	26.49
Potter covariance square root (Householder)	3247	3564	14	12	24.19
Carlson covariance square root (RSS)	2080	2560	50	30	18.24
Carlson covariance square root (MGS)	2830	3355	50	30	23.53
Inverse covariance	3520	3950	19	0	25.82
Inverse covariance square root	5080	5455	40	20	37.55
U–D factor	2935	3330	29	0	21.77

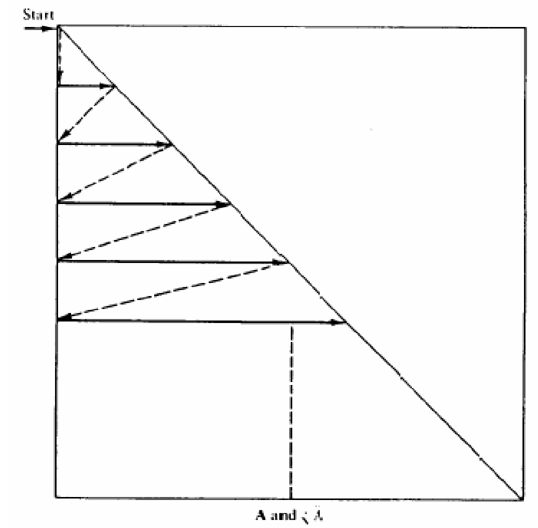
* $n = s = 10$ and $m = 2$.

time for addition = 2.7 μsec
time for multiplication = 4.1 μsec
time for division = 6.6 μsec
time for square root = 60.0 μsec

Square Root Comparison

- **Square Root**
- **Cholesky**

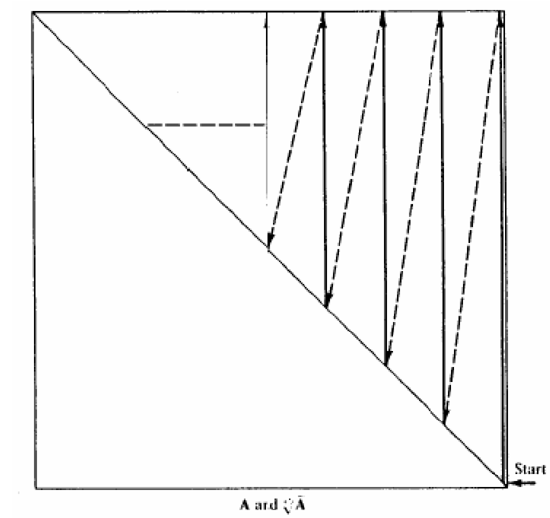
Maybeck, 1979



Square Root Comparison

- **Square Root**
- **Carlson**

Maybeck, 1979



Square Root Comparison

- **Square Root**
- **Potter (UD)**

Maybeck, 1979

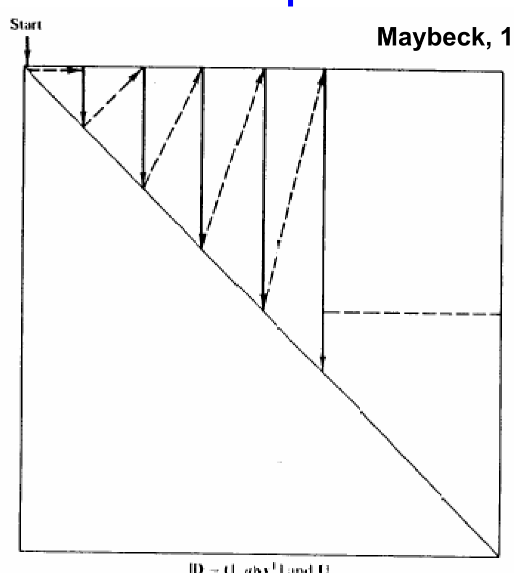


FIG. 7.3 Scanning of [D − (1 a)vv^T] and generation of Ū.