Exact Methodology for Testing Linear System Software Using Idempotent Matrices and Other Closed-Form Analytic Results*

Thomas H. Kerr III[a], CEO/Engineering Consultant

[a]TeK Associates, Lexington, Massachusetts, USA.

ABSTRACT

We alert the reader here to a variety of structural properties associated with idempotent matrices that make them extremely useful in the verification/validation testing of general purpose control and estimation related software. A rigorous general methodology is provided here along with its rationale to justify use of idempotent matrices in conjunction with other tests (for expedient full functional coverage) as the basis of a coherent general strategy of software validation for these particular types of applications. The techniques espoused here are universal and independent of the constructs of particular computer languages and were honed from years of experience in cross-checking Kalman filter implementations in several diverse commercial and military applications. While standard Kalman filter implementation equations were originally derived by Rudolf E. Kalman in 1960 using the Projection Theorem in a Hilbert Space context (with prescribed inner product related to expectations), there are now comparable Kalman filter results for systems described by partial differential equations (e.g., arising in some approaches to image restoration or with some distributed sensor situations for environmental toxic effluent monitoring) involving a type of Riccati-like PDE equation to be solved for the estimation error. The natural framework for such infinite dimensional PDE formulations is within a Banach Space (being norm-based) and there are generalizations of idempotent matrices similar to those offered herein for these spaces as well that allow closed-form test solutions for infinite dimensional linear systems to verify and confirm proper PDE implementations in S/W code. Other closed-form test case extensions discussed earlier by the author have been specifically tailored for S/W verification of multichannel maximum entropy power spectral estimation algorithms and of approximate nonlinear estimation implementations of Extended Kalman filtering and for Batch Least Squares (BLS) filters, respectively.

Keywords: Verification & Validation of Linear System Software Sub-Modules using Analytic Closed-Form Results, ODE's and PDE's, Trade-off's in Batch filter Use, Revised Assessment of Preferred Squareroot Filter implementations

1. INTRODUCTION

A Kalman filter (see Fig. 1) is an efficient and convenient computational scheme for providing the optimal linear estimate of the system state and an associated measure of the goodness of that estimate (the variance or covariance). In order to implement a KF, the actual continuous-time representation of the physical system must be adequately characterized by a linear (or linearized) ordinary differential equation (ODE) model, represented in state space at time t in terms of a vector $\mathbf{x}(t)$, and having associated initial conditions specified, and availing sensor output measurements $\mathbf{z}(t)$ (functions of the state plus additive measurement noise). It is mandatory that the KF itself actually contain within it an analytical mathematical model of the system and sensors in order to perform its computations (designated as a model-based estimator), and it must possess a statistical characterization of the covariance intensity level of the additive white Gaussian measurement and process noises present as well to enable an implementation.

Getting incorrect results at the output of a Kalman filter (KF) simulation or hardware implementation can be blamed on (1) use of faulty approximations in the implementation, or (2) on faulty coding/computer programming or (3) may actually be due to theoretical details of what should be implemented in the application being incorrectly specified by the analyst (especially since errors can arise, persist, and propagate in the published literature that the analyst makes use of as a starting point). Although situation (2) is initially impossible to distinguish from (1) or (3) for a new candidate KF software implementation; the handling of situations (1) and (3) have been

Research funded by TeK IRD Contract No. 96-105. Updates precursor version appearing in *Proceedings of* 16^{th} *Digital Avionics Systems Conference*, Irvine, CA, 26-30 Oct. 1997.

(Send correspondence to Thomas H. Kerr III, P.O. Box 459, 9 Meriam St., Suite 7-R, Lexington, MA 02420-5312, USA.)
Thomas H. Kerr III: e-mail: tkerr@tiac.net

Signal and Data Processing of Small Targets 2001, Oliver E. Drummond, Editor,
Proceedings of SPIE Vol. 4473 (2001) © 2001 SPIE · 0277-786X/01/$15.00

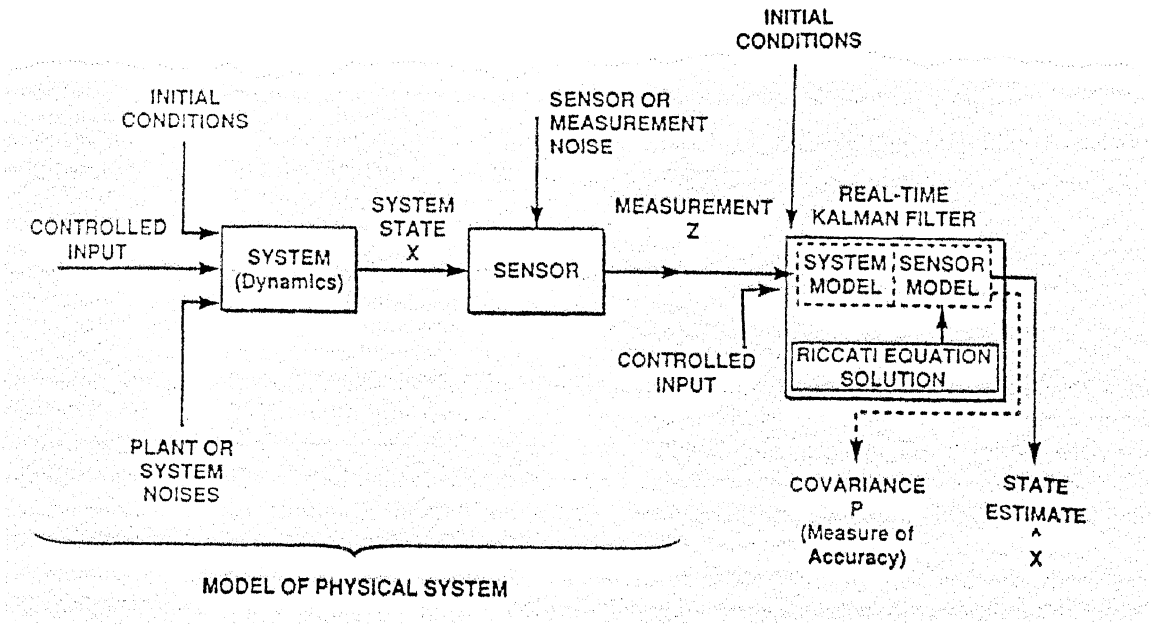

Figure 1. Overview functional block diagram of the internal structure of a Kalman filter

discussed elsewhere so that any residual problems present can be ferreted out by first eliminating (1) and (3) as possibilities for contamination and, further, by using certain test problems of known analytic closed-form solution for software calibration/check-out in the manner discussed here within the context of an original unique approach to Independent Verification and Validation (IV&V) for Kalman filter and LQG-like optimal control code. A rigorous general methodology is provided here along with its rationale to justify use of idempotent matrices in conjunction with other tests (for expedient full functional coverage) as the basis of a coherent general strategy for validation of software for these particular types of applications.

Using the notation $\mathcal{R}$ and $\mathcal{C}$ for the real and complex fields, respectively, as in [1], the following was offered [1, Corollary 3.5]: Let $A \epsilon \mathcal{C}^{n \times n}$ and suppose that

$$A^2 = A\,A = A \tag{1}$$

(as the property of matrix A being *idempotent*), then the matrix exponential is

$$e^A = I_{n \times n} + (e - 1)A\,. \tag{2}$$

While the above result is correct and somewhat straight forward, this form doesn't directly exhibit how to include the presence of a discrete-time step, Δ, or continuous-time, t, or for scaled (i.e., scalar multiples of) idempotent matrices in the use of this form for the computation of the system transition matrix, as is remedied here (Eq. 12) and, historically, in [2]–[4]. As in [2]–[4], we go further to also offer actual examples of idempotent matrices (Eq. 17) and show how to obtain them (as reviewed in Eqs. 14 to 17) and our alternative form (Eq. 12) for the result of Eq. 2 appears to be more useful in the application of software (S/W) verification that is considered in detail here for the first time. This approach to obtaining closed form solutions did not appear in any earlier collections such as in [5] nor in corresponding considerations of numerical stability [6]. The slightly unusual more elaborative style that we employ here is to also make our results fully accessible to interested *software implementers*, whose background and experience will differ from that of the usual readership.

We extend our novel results of [2]–[4] in offering a closed-form analytic expression for the discrete-time equivalent of continuous-time process noise Q_d (reviewed as Eq. 10) in the next section (reviewed as Eq. 20) and proceed to show as new insights offered here in subsequent sections how this same expression reoccurs time and again in the following other areas of linear system implementation and control:

1. Testing for system "*controllability*" and "*observability*" [7]–[9] (as regularity conditions that must be satisfied† before

† A less restrictive alternative is to satisfy mere "*stabilizability*" and "*detectability*", respectively, which are somewhat related regularity concepts [10].

proceeding to initiate any other items occurring below in this list);

2. Implementing a Kalman filter (KF) to correctly account for cross-correlations present due to system structure (and likewise for high fidelity Monte-Carlo simulations for algorithm verification);
3. Solving an associated matrix Lyapunov equation;
4. Implementing a particular open-loop [without feedback] finite horizon Linear Quadratic (LQ) Optimal Control solution;
5. Implementing a Linear Quadratic Gaussian (LQG) controller, which by virtue of the *separation theorem*, is just a repeat of items 2 and 4 above treated independently, then simply concatenated as the KF with an LQ controller appended to it.

The benefit of these closed-form solutions to be offered here is that the output computations of existing object oriented software modules (originally for the more general matrices anticipated to be encountered in various application scenarios) can be compared to our closed-form analytic results, which are exact when our hypothesized inputs (and the corresponding software inputs under user control) are the *same* identical idempotent matrices and so **outputs should ideally be identical as well** and we are alerted to the need for further debugging and more unit and/or integration testing *that should be performed* if they are **not** identical. These test cases can serve as a *necessary* gauge of how well the general purpose S/W modules perform in matching the template as a measure of achievement in reaching a *function point* benchmark goal, and may be directly observed by *any* user and not just by S/W Quality Assurance specialist (but may still need a further human evaluator's mitigating interpretation of how the S/W algorithms internally incur numerical round-off so any calculated answers encountered in a direct comparison that don't match up with the ideal may still be "close enough"). Unlike in [11], [12], this approach avoids judging software output fidelity by consensus of output agreement amongst several participating software packages (an inappropriate generalization of the *Delphi Method* of polling experts), where a dissenter could be unfairly punished even though it may be the only one offering the correct answer.

The well-known computer scientist/numerical analyst, Manuel Blum (UC, Berkeley), has long advocated use of something analogous to our closed-form solutions as a permanent resident on each machine or in pertinent software evaluation modules in order to calibrate software before initiating production runs [13]. Such an approach would immediately reveal any flaws present (that would adversely affect related more extensive computations before they're even initiated) when outputs no longer match our short templates as a quick, easy-to-understand sentinel that all users can rely upon. The need for such a watchdog should be apparent in the current fluctuating PC environment of changing Intel hardware 286 (16-bit) chips, 386 (32-bit), and upcoming 64-bit chips (recall the Pentium flaw encountered when it was a new chip); and changing/evolving operating systems (e.g., Microsoft DOS 6.22 and *Windows* 3.1, 3.11; *Windows* 95 (5 versions), *Windows* 98; *Windows* NT 3.5, 3.51, 4.0, 5.0, 2000; *Windows* CE &ME; and proprietary third party real-time versions of *Windows*; and IBM OS/2 and DOS 7.0, Linux, etc.) and BIOSes and updates to the Intel Math Kernel Library (such as occured within the past six months and is expected to continue to be somewhat in flux).

We explain why our proposed new idempotent matrix approach is superior in S/W verification to the well-known conventional approach of invoking specific similarity transformations to obtain closed-form solutions via conversion to a diagonally (and, perhaps, super-diagonally) banded *Jordan canonical form* by **(1)** being a lesser computational burden; **(2)** incurring less roundoff; and **(3)** exhibiting no ambiguity when repeated eigenvalues are encountered (unlike the Jordan approach, where encountering repeated eigenvalues is more challenging and more computationally burdensome to correctly decipher by requiring a *confluent* form to be invoked which is **seldom** straight forward). We also examine other numerical analysis issues associated with use of these examples (based on idempotent matrices) such as indications of ill-conditioning conveyed by somewhat controversial condition numbers and the evident insignificance of such indications.

Other useful extensions are also offered here in the last three sections including hints on how to appropriately generalize this same *idempotent matrix-based* methodology for software verification of ordinary differential equation (ODE) implementation to also handle S/W verification of infinite dimensional partial differential equation (PDE) implementations. Ideas for handling S/W verification of nonlinear estimation (also potentially infinite dimensional) have already been worked out [4].

2. CONSIDER COMMON MODULES UTILIZED WITHIN MOST LINEAR SYSTEMS SOFTWARE

First, further defining notation, the familiar time-invariant continuous-time state-variable representation of a linear system in terms of the matrix triple (F, G, H) is

$$\frac{dx(t)}{dt} = F\,x(t) + G\,u(t), \tag{3}$$

$$y(t) = H\,x(t)\,, \tag{4}$$

with $u(\cdot)$ being the input and $y(\cdot)$ being the measured output and F, G, and H being, respectively, $n \times n$, $n \times r$, and $m \times n$ matrices over the real field.

Because there have been numerous prior erroneous investigations into how to properly discretize a continuous-time system (e.g., [14]) and completely capture the behavior while seeking to prevent any anomalies from arising [15]–[17], we recall a fairly straight forward approach that has found widespread favor in industry, which proceeds from first principles by first looking at the form of the solution to Eq. 3 being:

$$x(t) = e^{F(t-s)}\,x(s) + \int_s^t e^{F(t-\tau)}\,G\,u(\tau)\,d\tau. \tag{5}$$

and then converting this into an *exact* formulation in discrete-time by proceeding as follows. For the upper and lower limits of the above integral selected to be the following particular specializations:

$$t = (k+1)\,\Delta\,;\;\; s = k\,\Delta\,,$$

with a constant, user specified Δ = time-step increment, the solution of Eq. 5 corresponds exactly to the following recursive iteration in discrete-time:

$$x(k+1) = \left[e^{F\,\Delta}\right]\,x(k) + \int_{k\,\Delta}^{(k+1)\,\Delta} e^{F(k\,\Delta-\tau)}\,G\,u(\tau)\,d\tau, \tag{6}$$

which, under the further assumption that $u(\tau)$ is essentially constant [‡] over the time-step from any $k\Delta$ to any other $(k+1)\Delta$, yields:

$$x(k+1) = \left[e^{F\,\Delta}\right]\,x(k) + \left[\int_{k\,\Delta}^{(k+1)\,\Delta} e^{F\,((k+1)\,\Delta-\tau)}\,G\,d\tau\right]\,u(k). \tag{7}$$

Upon making the change of variable $\tau = \tau' + k\,\Delta$ and substituting into the above integral yields:

$$x(k+1) = \left[e^{F\,\Delta}\right]\,x(k) + \left[e^{F\,\Delta}\right]\,\left[\int_0^{\Delta} e^{-F\,\tau'}\,d\tau'\right]\,G\,u(k).$$

While the previous two equations above correspond to the continuous-time formulation of Eq. 5 only by assuming $u(\tau)$ to be essentially constant over each small step-size Δ (or by incurring a minor approximation error if it isn't constant), the expression in Eq. 6 is the most general version of the discrete-time formulation that corresponds *exactly* to Eq. 5. In the case where the continuous-time input $u(t)$ is independent, white, Gaussian process noise of continuous-time covariance intensity level, Q (being symmetric and positive semi-definite), to then have *exact* adherence or correspondence to Eqs. 3 and 5 without any approximation incurred (or needing to utilize a zero-order hold), the discrete-time formulation should be:

$$x(k+1) = \left[e^{F\,\Delta}\right]\,x(k) + u'(k)\,, \tag{8}$$

where

$$u'(k) = \text{zero} - \text{mean Gaussian white noise}\,, \tag{9}$$

[‡] This assumption is sometimes enforced through use of a zero-order hold on the input $u(t)$.

having discrete-time covariance intensity level [8, p. 270], [9, p. 171, Eq. 4-127b]:

$$\begin{aligned} Q_d &= E\left[u'(k)(u'(j))^T\right] \\ &= e^{F\Delta}\left[\int_0^{\Delta} e^{-F\tau} G Q G^T e^{-F^T\tau}\, d\tau\right] e^{F^T\Delta}\delta_{kj} \end{aligned} \tag{10}$$

where δ_{kj} in the above is the usual Kronecker delta, which is 1 for $k = j$ and 0 otherwise. The above Q_d (as obtained directly from the right-hand term in Eq. 6 by post-multiplying it by its transpose and taking expectations throughout and utilizing the sifting property of the Dirac delta function) is the appropriate discrete-time process noise covariance level to use to have *exact* agreement between the discrete-time mechanization of Eq. 7 and the continuous-time formulation of Eqs 3 and 5. A well-known approximation for Q_d (due to R. E. Kalman and T. S. Englar) which is sometimes invoked is to take

$$Q_d^{'} = Q\,\Delta\,; \tag{11}$$

however, the deleterious effect of invoking this approximation varies and is uncalibrated, but for the case of a diagonal Q, it can easily be seen to be an unsatisfactory representation since the approximation of Eq. 11 suppresses all off-diagonal cross-correlation terms (i.e., the approximation Q_d' of Eq. 11 is merely diagonal if Q was and ignores any off-diagonal cross-correlations that *should* be present) as compared to the exact Q_d of Eq. 10, which preserves all inherent cross-correlations due to system structure (pre- and post-multiplying it) as a now nondiagonal matrix even if the original continuous-time Q was diagonal.

The effects due to incorrect representation of cross-correlation terms can be substantial [2, Eq. 40 ff] and is a significant issue for many applications [3] where fidelity is important as in developing adequate simulation testbeds for multichannel spectral estimation (where cross-channel leakage or feed-through is a concern) or for failure detection verification in an Inertial Navigation System or in Receiver Autonomous Integrity Monitoring (RAIM) in GPS (where isolation of effect to the culprit component level is the goal without false cross-contamination of failure signatures or being overly optimistic that they aren't); or within its mathematical dual in maneuver detection [68] while tracking noncooperative targets via radar or in other passive or active scenarios.

3. USING IDEMPOTENT MATRICES FOR CLOSED-FORM SOLUTIONS

A neat application of idempotent matrices is in constructing test matrices for verifying the transition matrix algorithmic implementations that are used for computer computation of e^{Ft}. The utility of these test matrices is that the resulting analytically derived expression for e^{Ft} is conveniently in closed-form for $F = \kappa A$, where κ is a scalar multiplying factor. Hence the output performance of a general e^{Ft} subroutine implementation can ultimately be gauged by how close it comes to achieving the ideal exact solution (which is known for scaled versions of idempotent matrices as demonstrated next).

Using the representation of a matrix exponential, defined in terms of its Taylor series, but evaluated with a scaled idempotent matrix $A^{'} = \kappa A$, with A having the property of Eq. 1 and κ a scalar, being substituted along with time-step Δ; the expansion of $e^{A^{'}\Delta}$ now becomes

$$\begin{aligned} e^{A^{'}\Delta} &= e^{A\kappa\Delta} \\ &= \sum_{k=0}^{\infty} \frac{A^k}{k!}\kappa^k\Delta^k \\ &= I_{n\times n} + A\left(\frac{\Delta\,\kappa}{1!} + \frac{\Delta^2\,\kappa^2}{2!} + \frac{\Delta^3\,\kappa^3}{3!} + \cdots\right) \\ &= I_{n\times n} + A\left(1 + \frac{\Delta\,\kappa}{1!} + \frac{\Delta^2\,\kappa^2}{2!} + \cdots - 1\right) \\ &= I_{n\times n} + A(e^{\Delta\,\kappa} - 1)\,, \end{aligned} \tag{12}$$

as explained in [2, Sect. IV]. The above expression for the matrix exponential is thus reduced to a single scalar multiplication of a matrix (involving a scalar exponential) and a single Matrix addition (involving the identity matrix that merely adds one to each diagonal term). Thus, the resulting closed-form exact expression for the transition matrix corresponding to idempotent system matrices is as depicted in the last line of Eq. 12 as a finite two step

operation (relatively immune to computational truncation and roundoff error since so few terms and operations are invoked in the process as just one scalar-Matrix multiplication and one addition of unity to each diagonal element) as compared to an infinite series for e^{Ft} that must be truncated or terminated using a stopping rule usually based on the magnitude or norm of the discarded terms in the case of standard software implementations [2, Eq. 25 ff], [4, p. 197, No. 3] for more general matrices.

While an obvious alternative that comes to mind in seeking closed-form solutions is to work with *Jordan canonical forms* but, to do so, one would have to first find all eigenvalues, then all eigenvectors (with complications if eigenvalues are repeated), then inverses of the eigenvalue matrix for the necessary similarity transformations and a need to worry about (or determine) how any repeated eigenvalues encountered actually *break* (either clustered entirely together, in separate smaller groups, or distinctly separate [§]) to construct and place the appropriate super-diagonal 1's properly. One also has to calculate the inverse of the eigenvector matrix in order to perform the similarity transformation to reveal the Jordan form and to convert back when finished, so computational burden goes at least as n^3 and moreover this entire approach doesn't avail nice numbers in general [18] (so the comparison of rounded-off numbers is an issue in comparing computed results to closed-form analytic expressions).

Additionally, if repeated eigenvalues do occur, then derivatives need to be taken of the scalar function to determine the proper corresponding matrix function. In either case, the similarity transforms are to be applied in reverse to yield final answers as now weighted mixtures of displayed Jordan canonical form results which had previously been clear in revealing function evaluations occurring along the diagonals and its upper triangular derivatives (at eigenvalues) but now have ample opportunity for round-off to taint the final closed-form results that should serve as an unambiguous template to be compared to as ideal. One further wrinkle is that even if the arbitrary test matrices used for comparison have entries that are exclusively *real*, the underlying eigenvalues to be computed can be complex and so can the associated eigenvector matrix and its inverse so worries are compounded by the effects of round-off in both *real* and *imaginary* components of the underlying computations. The idempotent matrix S/W verification approach that we advocate doesn't require any complex arithmetic and so is simpler in that regard.

While its true that controllability and observability are sometimes easily confirmed or deduced from the associated Jordan canonical form but only if eigenvalues don't repeat does absence of all zero columns (rows) in the transformed input gain (observation) matrix mean that the underlying system under test is controllable (observable). While there *are* theoretical results (such as by J.J. O'Donnell) to amend the standard Jordan canonical approach to handle situations where eigenvalues are repeated [19], [20], the penalty is to incur even greater complexity in both additional clarifying computations and supporting analysis.

There is also no worry about stiffness or ill-conditioning of test matrices dragging out the calculations by requiring special *implicit integration* routines like *Gear's method* (which could impose further delay and hinder expedient output comparisons for algorithm verification) since all eigenvalues are in the same ball-park (viz., 0 or 1) when idempotent matrices are used for the tests. Since this limited range of likely eigenvalues is expedient for S/W testing, we ignore possible condition number issues here [¶] related to eigenvalue sensitivities affecting accuracy in the calculation of the matrix exponential since its not relevant to idempotent matrix-based S/W tests advocated here. However, this is

[§]If, for the sake of argument, test matrices were deliberately constructed by first dictating entries to appear on a diagonal matrix J then arbitrarily selecting the eigenvector matrices for convenience as say an orthogonal matrix E (so that when normalized, its transpose is its inverse), the resulting test matrix yielded by such a construction as $E^T J E$ would be symmetric. As mentioned above, the structure of *known solutions* as targeted for comparison to S/W outputs would still be altered from matrix pre- and post-multiplications in unwinding the diagonal function evaluations via the reverse similarity transform and so tainted by roundoff rather than be exact for unambiguous comparison to S/W outputs.

[¶]One worry is that the associated condition numbers for idempotent matrices is infinite since $\lambda_{max}/\lambda_{min} = 1/0$. However, several examples of correctly implemented independent commercial third party software (e.g., MatLab, MathCad, TK-MIP, Matrix-X, VisSim, ProtoSim) haven't had any trouble matching the expected template outputs that otherwise would be impossible if the cases were truly ill-conditioned as with, say, Hilbert matrices. Moreover, considerable evidence exists that the above ratio of max to min eigenvalue as a measure of possible ill-conditioning is misleading [21, pp. 71-72], for which the following example is used to illustrate: $C = diag(10^{100}, 1)$, with associated large condition number being 10^{100} yet exhibiting no difficulty in being able to accurately invert to solve Eq. 13 despite such a large condition number being present. Another exciting example of useful new computational results being obtained despite otherwise discouraging but evidently irrelevant indications of an infinite condition number being encountered as $\lambda_{max}/\lambda_{min} = 2/0$ also arises in [22] as novel inroads are made in image segmentation. Another structural observation about the idempotent matrices used here is that although eigenvalues can only be 0 and 1 and for matrix dimensions greater than two, eigenvalues are obviously repeated, the idempotent matrices here are all symmetric matrices and, as such, have associated eigenvectors that are orthogonal so that all these repeated

precisely the reason why output matching using this test approach is only a *necessary* test that should be achievable as representative of "best case" situations. The actual matrices encountered in a particular application may exhibit ill-conditioning, so actual internal S/W algorithm implementations may still need to be bolstered and made capable of handling more extreme situations even though outputs match for idempotent matrices (which is why successful matching is **not** sufficient S/W verification). So, again, only the proper S/W handling of cross-terms or off-diagonal terms is vindicated in passing test cases based on idempotent matrices.

The use of κ in the above is another scaling trick that increases the realm of test case possibilities of known closed-form analytic solutions available for testing against software. Correct answers could ostensibly be "hardwired" within candidate software under test, but appropriate scaling of the original test problems to be used as inputs can foil this possible stratagem of any unscrupulous software supplier/developer by swamping him with an infinite variety of possibilities for the selection of parameter κ and choice of idempotent A and variable dimensioning using the aggregation/augmentation technique offered here in Sec. 8.

4. OBTAINING IDEMPOTENT MATRICES

For clarity, motivation is now offered for how actual idempotent matrices may be obtained. Consider the problem of seeking to solve the following algebraic equation for an $(n \times 1)$ x, given an $(m \times 1)$ y and an $(m \times n)$ C:

$$\underline{y} = C\,\underline{x}. \tag{13}$$

Assuming that the rank of C is the same as the rank of the *augmented matrix* $[C|\underline{y}]$, it is reasonably well-known (see [23, Appendix A, Section A.1]) that a solution to Eq. 13 is of the form

$$\underline{x} = C^{\dagger}\underline{y} + (I_n - C^{\dagger}C)\underline{w} \tag{14}$$

for arbitrary $\underline{w}$ and that the **term within the parentheses in Eq. 14 is idempotent** (where $C^{\dagger}$ in Eq. 14 is the Moore-Penrose pseudoinverse). In forming two counterexamples in [23, Sects. II.B, II.D], the following two matrices and their respective pseudoinverses were obtained (as derived in [23, App. C]):

$$C_1 = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \quad ; \quad C_1^{\dagger} = \frac{1}{25}\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \tag{15}$$

and

$$C_2 = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad ; \quad C_2^{\dagger} = \begin{bmatrix} -\frac{1}{3} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 \\ \frac{2}{3} & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}. \tag{16}$$

Therefore via Eqs. 15 and 16, the following two matrices are idempotent:

$$A_1 \triangleq (I_{2\times 2} - C_1^{\dagger}C_1) = \begin{bmatrix} \frac{4}{5} & -\frac{2}{5} \\ -\frac{2}{5} & \frac{1}{5} \end{bmatrix},$$

$$A_2 \triangleq (I_{3\times 3} - C_2^{\dagger}C_2) = \begin{bmatrix} \frac{1}{3} & -\frac{1}{3} & \frac{1}{3} \\ -\frac{1}{3} & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} & \frac{1}{3} \end{bmatrix}, \tag{17}$$

both of which check as being idempotent by satisfying Eq. 1 as an identity. In considering the step-size to use in the evaluation of the final line of Eq. 12 for $\kappa = 1$, convenience in using just a scalar multiplying factor of one half times the matrix of Eq. 12 would dictate using $\Delta = 0.405465108$ since interpolation within Burlington's mathematical tables [24] yields:

$$(e^{\Delta} - 1) = (e^{0.405465108} - 1) = (1.50 - 1) = 0.50 \tag{18}$$

---

eigenvalues must "*break separately*" (as defined in [19]) since our idempotent matrices are all diagonalizable. This underlying structural observation justifies use of idempotent matrices but there's no requirement to actually perform a diagonalization in order to use them.

Therefore, the two evaluations corresponding to invoking Eq. 12 are:

$$e^{A_1\Delta} = I_{2\times 2} + \frac{1}{2}A_1 = \begin{bmatrix} 1.40 & -0.20 \\ -0.20 & 1.10 \end{bmatrix},$$

$$e^{A_2\Delta} = I_{3\times 3} + \frac{1}{2}A_2 = \begin{bmatrix} 1.16\overline{6} & -0.16\overline{6} & 0.16\overline{6} \\ -0.16\overline{6} & 1.16\overline{6} & -0.16\overline{6} \\ 0.16\overline{6} & -0.16\overline{6} & 1.16\overline{6} \end{bmatrix}. \tag{19}$$

The above two results are now the *known* closed-form exact solutions to an $e^{A\Delta}$ evaluation of the two matrices of Eq. 17 with $\Delta = 0.405465108$.

5. EXACT DISCRETE-TIME PROCESS NOISE COVARIANCE MATRIX FOR IDEMPOTENT MATRICES

Using the result of Eq. 12 for idempotent matrices within the general expression of Eq. 10 for Q_d allows this expression for the required discrete-time process noise covariance to be evaluated analytically in closed-form as:

$$\begin{aligned} Q_d &= \left[I_{n\times n} + A(e^{\Delta} - 1)\right] \int_0^{\Delta} \left[I_{n\times n} + A(e^{-\tau} - 1)\right] GQG^T \left[I_{n\times n} + A^T(e^{-\tau} - 1)\right] d\tau \\ &\quad \left[I_{n\times n} + A^T(e^{\Delta} - 1)\right] \\ &= \left[I_{n\times n} + A(e^{\Delta} - 1)\right] \\ &\quad \int_0^{\Delta} \left[GQG^T + (AGQG^T + GQG^TA^T)(e^{-\tau} - 1) + AGQG^TA^T(e^{-2\tau} - 2e^{-\tau} + 1)\right] d\tau \\ &\quad \left[I_{n\times n} + A^T(e^{\Delta} - 1)\right] \\ &= \left[I_{n\times n} + A(e^{\Delta} - 1)\right] \\ &\quad \left[GQG^T\Delta + (AGQG^T + GQG^TA^T)(1 - e^{-\Delta} - \Delta) + AGQG^TA^T(-\frac{3}{2} - \frac{1}{2}e^{-2\Delta} + 2e^{-\Delta} + \Delta)\right] \\ &\quad \left[I_{n\times n} + A^T(e^{\Delta} - 1)\right]. \end{aligned} \tag{20}$$

This is the **new** original result of [2] that is also useful to test against as a confirming check for general software implementations of Eq. 10 (by using an idempotent matrix as input). This result is utilized again within the context of two other applications in the next two sections in obtaining closed-form Lyapunov and LQ optimal control solutions, respectively.

6. WHAT ABOUT ITS USE AS SOLUTIONS TO MATRIX LYAPUNOV & RICCATI EQUATIONS?

An important theoretical aspect that directly affects the realm of applicability of idempotent matrices in validating linear systems software is the fact that idempotent matrices have eigenvalues that are exclusively zeroes and ones [25]. This interferes with using idempotent matrices as the system matrix F for validating solutions of the algebraic matrix Lyapunov equation:

$$0 = F\,P + P\,F^T + GQG^T \tag{21}$$

that must be solved for P, as the computed steady-state solution of the continuous-time matrix Lyapunov equation:

$$\frac{d}{dt}P(t) = F\,P(t) + P(t)\,F^T + GQG^T, \tag{22}$$

with initial condition $P(0) = P_o$ (since positive definite steady-state solutions only exist for stable system matrices and because of its unstable eigenvalues at zero and one, idempotent matrices clearly do not qualify in this role). Since the exact solution of Eq. 22 is of the form [7, Eq. 9.2-12]:

$$P(t) = \Phi(t,0)\,P_o\,\Phi^T(t,0) + \int_0^t \Phi(t,\tau)\,GQG^T\,\Phi^T(t,\tau)\,d\tau\,, \tag{23}$$

idempotent matrices can again be used to an advantage with $F = A$ and the entire transient solution is revealed (via the result of Eq. 20) to be of the form:

$$\begin{aligned} P(t) &= [I_{n\times n} + A(e^t - 1)] \\ &\left[P_o + GQG^T t + (AGQG^T + GQG^T A^T)(1 - e^{-t} - t) + AGQG^T A^T(-\frac{3}{2} - \frac{1}{2}e^{-2t} + 2e^{-t} + t)\right] \\ &[I_{n\times n} + A^T(e^t - 1)] \end{aligned} \tag{24}$$

that can then be compared to software output for this test case (notice the presence of P_o in Eq. 24 that distinguishes it from the result of Eq. 20). Moreover, idempotent matrices could be used as the system matrix in obtaining closed-form solutions for both the continuous-time and discrete-time Riccati equations since there is no restriction that system matrices be stable in these applications but other closed-form expressions [3], [4], [26], [27] are superior for software cross-checking in this specialized Riccati verification role (see [4, Table 1, App.] for, respectively, a recommended collection of tests and a way to avoid having to solve for the roots of an associated biquadratic polynomial otherwise encountered in seeking to utilize the result of [27] as a software test case availing a conveniently known answer both before and after a periodic measurement).

7. VERIFYING FINITE HORIZON LQ/LQG OPTIMAL CONTROL SOLUTIONS

For a linear system of the form of Eq. 3 with idempotent system matrix $F = A_1$ from Eq. 17 and $G = \underline{g} = [0\ 1]^T$, with initial condition $x(0) = [0,0]^T$, the present goal is to find the deterministic optimal control $u^*(t)$ that transfers the system of Eq. 3 to the specified final state

$$x(T) = x_1 \text{ (known)} \tag{25}$$

and simultaneously minimizes the following finite time horizon scalar quadratic cost function:

$$C[u(t)] = \int_0^T [u(t)]^2\, dt. \tag{26}$$

Since a feedback solution is not required here, we now obtain a closed-form open-loop solution for the above optimal control problem that can be used to validate software that provides numerically computed optimal controls of this same type.

One benefit of dealing with an idempotent system matrix, A, as occurs here, is that the Kalman "rank test for controllability" (as the standard regularity condition that must be satisfied before an optimal control can be sought) degenerates into a much more tractable expression for the *Controllability Grammian* as

$$\text{rank}\left[G\vdots AG\vdots\cdots\vdots A^{(n-1)}G\right] = \text{rank}\left[A\vdots AG\right]\ , \tag{27}$$

that one must check the rank of, where its being of rank n establishes that the system is *controllable*. However, in the case of the present 2-dimensional example, there is no actual reduction in complexity (but there would be if the second matrix of Eq. 17 were used instead) since the full blown test is no more involved than the simplified expression:

$$\text{rank}\left[\underline{g}\vdots A_1\underline{g}\right] = \text{rank}\begin{bmatrix} 0 & \vdots & -\frac{2}{5} \\ & \cdot & \\ 1 & \vdots & \frac{1}{5} \end{bmatrix} = 2\ , \tag{28}$$

and since the state size $n = 2$ for this system, Eq. 3 is in fact controllable for the above parameters specified in the first sentence of this section. By duality, tests of *observability* simplify in a manner analogous to Eq. 28 with the H' of Eq. 4 in the role of $G = \underline{g}$ above and A_1' in the role of A_1.

Recall from [2, pp. 97-100], that for a controllable system (as established above) of the form of Eq. 3, with zero initial condition, and final condition x_1, where the minimization of a simplified or abbreviated finite-horizon quadratic cost function of the form of Eq. 26 is sought, then the final form of the optimal control that solves this

problem (as worked out in [23, App. C] from a referenced theorem in Luenberger's 1969 textbook *Optimization by Vector Space Methods*) is:

$$u^*(t) = \underline{g}' e^{F'(T-t)} \left[\int_0^T e^{F(T-s)} \, \underline{g}\underline{g}' \, e^{F'(t-s)} \, ds \right]^{-1} x_1 \, . \tag{29}$$

There is no worry about the expression within the brackets being singular and non-invertible because it is recognized to be an alternative version of the controllability test [28, Theorem 6, p. 186] and is positive definite (nonsingular) iff the system is controllable (as already established here in Eq. 28). Please notice the similarity between the expression within brackets to be inverted in the above Eq. 29 and the structure that is routinely encountered in computing the *exact* discrete-time equivalent to a continuous-time white process noise covariance intensity matrix as Eqs. 10 and 20. The asserted similarity in form is seen to be identical when the two pre- and post-multiplying matrices in Eq. 10 are brought back under the integral sign as the following re-representation:

$$Q_d = \left[\int_0^\Delta e^{F(\Delta-\tau)} \, G \, Q \, G^T \, e^{F^T(\Delta-\tau)} \, d\tau \right] \delta_{kj} \, , \tag{30}$$

where Δ is used here in Eq. 30 in the role of T in Eq. 29, G serves in the role of $\underline{g}$, Q serves in the role of an identity matrix $I_{n\times n}$, and the dummy variable of integration τ is used here in the role of s in Eq. 29. From the simplification offered in [2, Eq. 42] as the closed-form evaluation of Eq. 10 when F is an idempotent matrix A, we have that the result of Eq. 20 again applies. The form of the optimal control solution in Eq. 29 is of the form:

$$u^*(t) = \underline{g}' e^{A_1'(T-t)} Q_d^{-1} x_1 \, , \tag{31}$$

where $Q_d \equiv M$ in [2] (using the evaluation result of Eq. 20), the apostrophe,', is now used to indicate matrix transpose to avoid confusion with T that is now used to represent the fixed known (but arbitrary) final time. The solution to this optimal problem (details being provided in [2]) is:

$$u^*(t) = \begin{bmatrix} \frac{8T\exp(-T-t)+2\exp(-t)-2\exp(-T-t)-8\exp(-T)+9\exp(-2T)-1}{2(-T+2-4\exp(-T)+T\exp(-2T)+2\exp(-2T))} \\ \frac{-2T\exp(-T-t)+2\exp(-t)-2\exp(-T-t)+2\exp(-T)-\exp(-2T)-1}{(-T+2-4\exp(-T)+T\exp(-2T)+2\exp(-2T))} \end{bmatrix}' x_1 . \tag{32}$$

This closed-form answer can be compared to the software computed solution for a variety of explicit values of t, T, and x_1 in the above (which allows the tester or user to alter these three last named variables enough to foil any possible attempts by software developers to "hard wire" solutions).

Actual LQG feedback control solutions to be implemented for practical applications would, hopefully, also use the additional step of invoking Loop Transfer Recovery (LTR). To otherwise proceed with an LQG solution without some such robustification of the solution would be extremely risky to pursue since it is now well-known (since at least 1978 [69], [70]) that pure LQG has zero phase margin and is therefore right on the cusp of being unstable (which can occur due to unmodeled dynamics being present but obscure, due to other environmental factors, or due to parameter changes associated with the aging of components).

8. AGGREGATION/AUGMENTATION OF LOWER ORDER RESULTS INTO HIGHER

A difficulty, as discussed in [26, Sect. I], is that most closed-form KF covariance solutions are of either dimension 1 or 2 (as in [8, pp. 138–142, pp. 243–244, p. 246, pp. 255–257, pp. 318–320]) or 3 (as in [27]). To circumvent this dimensional mismatch to higher dimensional real applications that may be hard-wired to a fixed dimension $\bar{n}$, we can achieve the dimension $\bar{n}$ sought by augmenting to obtain the requisite matrices and vectors as a concatenation of several lower dimensional test problems with solutions that are also already known. Use of only totally diagonal decoupled test problems is notorious for being too lenient and not taxing enough to uncover software implementation defects (when the problems exist in the portion of the code that handles cross-term effects). Augmenting either several low-dimensional 2-state problems or fewer 3-state problems is the way to proceed in order to easily obtain a general $\bar{n}$-state non-trivial non-diagonal test problem. A confirmation that this proposed augmentation is valid in general is provided next for a closed-form steady-state radar target tracking solution.

An initial worry in adjoining the same 3-state problem with itself to obtain a 6-state test case of known closed-form solution relates to whether "controllability and observability" are destroyed or preserved, while the 3-state problem

does itself possess the requisite "controllability and observability" properties. "Controllability and observability" conditions, or at least more relaxed but similar "stabilizability and detectability" conditions [10, pp. 62–64, pp. 76–78, pp. 462–465], need to be satisfied in order that the covariance of a KF be well-behaved [8, p. 70, p. 142], [10]. The following mathematical manipulations analytically establish that such an adjoining of two 3-state test problems does not destroy the "controllability and observability" of the resulting 6-state test problem.

First consider the original 3-state test problem of [27] of the following form:

$$\overset{(3\times 1)}{x} = \begin{bmatrix} \text{position} \\ \text{velocity} \\ \text{acceleration} \end{bmatrix}, \tag{33}$$

with

$$\begin{aligned} \dot{x}(t) &= F_1 x(t) + G_1 w(t), \quad w(t) \sim \mathcal{N}(0, Q_1), \\ y(t) &= H_1 x(t) + v(t), \quad v(t) \sim \mathcal{N}(0, R_1), \end{aligned}$$

and assumed to be already satisfying Kalman's "controllability and observability" rank test criteria ([8, p. 70]), respectively, as

$$\begin{aligned} rank[G_1 \vdots F_1 G_1 \vdots F_1^2 G_1] &= n_1 = 3, \\ rank[H_1^T \vdots F_1^T H_1^T \vdots (F_1^T)^2 H_1^T] &= n_1 = 3. \end{aligned}$$

Now the new augmented 6-state system will be of the form:

$$x = \begin{bmatrix} \text{position} \\ \text{velocity} \\ \text{acceleration} \\ \cdots \\ \text{position} \\ \text{velocity} \\ \text{acceleration} \end{bmatrix}, \tag{34}$$

with

$$\dot{x}(t) = \begin{bmatrix} F_1 & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & F_1 \end{bmatrix} x(t) + \begin{bmatrix} G_1 & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & G_1 \end{bmatrix} \begin{bmatrix} w(t) \\ \cdots \\ w(t) \end{bmatrix} \tag{35}$$

$$y(t) = \begin{bmatrix} H_1 & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & H_1 \end{bmatrix} x(t) + \begin{bmatrix} 1 & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \cdots \\ v(t) \end{bmatrix} \tag{36}$$

has effective system, process noise gain, and observation matrices, respectively, of the form

$$\begin{aligned} F_2 &= \begin{bmatrix} F_1 & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & F_1 \end{bmatrix}, \; G_2 = \begin{bmatrix} G_1 & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & G_1 \end{bmatrix}, \\ H_2 &= \begin{bmatrix} H_1 & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & H_1 \end{bmatrix}. \end{aligned} \tag{37}$$

In testing for controllability of the augmented system, please consider the corresponding *Controllability Grammian* depicted below:

$$rank[G_2 \vdots F_2G_2 \vdots F_2^2G_2 \vdots F_2^3G_2 \vdots F_2^4G_2 \vdots F_2^5G_2] =$$

$$rank \begin{bmatrix} G_1 & \vdots & 0 & \vdots & F_1G_1 & \vdots & 0 & \vdots & F_1^2G_1 & \vdots & 0 & \vdots & \cdots \text{other stuff} \\ 0 & \vdots & G_1 & \vdots & 0 & \vdots & F_1G_1 & \vdots & 0 & \vdots & F_1^2G_1 & \vdots & \end{bmatrix} =$$

$$rank \begin{bmatrix} G_1 & \vdots & F_1G_1 & \vdots & F_1^2G_1 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots & \cdots \text{other stuff} \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & G_1 & \vdots & F_1G_1 & \vdots & F_1^2G_1 & \vdots & \end{bmatrix} \tag{38}$$

$$= n_1 + n_1 = 3 + 3 = 6 = n_2 \, .$$

In the next to the last line of Eq. 38, the columns of the Controllability Grammian are rearranged for convenience to provide the necessary insight. Permuting columns of a matrix doesn't alter its rank but can alter at-a-glance conclusions. Since we are able to show that the augmented system rank is 6 in Eq. 38, this system is confirmed to be controllable since its rank equals its state dimension. A similar conclusion (on the requisite observability being satisfied) can be obtained by identical steps using the duality that exists between controllability and observability results and the associated forms of arguments or proofs since identical matrix structures, such as are present here, are again encountered. The above described augmented system of Eqs. 35 and 36 can be used with

$$\begin{aligned} R_2 &= \begin{bmatrix} R_1 & \vdots & 0 \\ 0 & \vdots & R_1 \end{bmatrix}, \; Q_2 = \begin{bmatrix} Q_1 & \vdots & 0 \\ 0 & \vdots & Q_1 \end{bmatrix}, \\ P_2(0) &= \begin{bmatrix} P_1(0) & \vdots & 0 \\ 0 & \vdots & P_1(0) \end{bmatrix}, \end{aligned} \tag{39}$$

since now the augmented system has been demonstrated above to be both "observable and controllable" and the measurement noise covariance R_2 of Eq. 39 to be utilized is positive definite. This final observation allows us to use this 6-state augmented test problem without reservations to validate any Riccati equation-based covariance software.

By similarly adjoining/aggregating/augmenting systems of known closed-form solution, higher order optimal control solutions can also be analogously obtained. The corresponding cost function to use when adjoining the two state LQ test case of the previous section with itself is:

$$\begin{aligned} C[u(t)] &= \int_0^T [u_1(t), u_2(t)] \, I_{2\times 2} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} dt \\ &= \int_0^T \left[u_1^2(t) + u_2^2(t)\right] \, dt = \int_0^T u_1^2(t) \, dt + \int_0^T u_2^2(t) \, dt \\ &= C[u_1(t)] + C[u_2(t)] \end{aligned} \tag{40}$$

which is minimized only when each constituent integral on the right side of Eq. 40 (of the form of Eq. 26) is minimized, thus the minimizing solution of Eqs. 29, 30, 31 also apply for each $u_i^*(t)$, for $i = 1$ and 2 in the aggregated optimal control formulation of twice the original state size (and is seen to easily generalize in like manner to larger concatenations). Although the results of Sects. 4 to 8 have been for cross-checking linear system software modules, for many approaches to handling nonlinear systems (such as statistical linearization, approximate nonlinear estimation or Extended Kalman filtering, or describing function techniques) software modules for linear systems are present as necessary stepping stones that also need to be verified.

9. OTHER PRECEDENTS & EXTENSIONS

While the following result from [1, Corollary 4.4]:

$$e^{S(\omega)t} = I_{3\times 3} + \frac{\sin \omega_o t}{\omega_o} S(\omega) + \frac{1 - \cos \omega_o t}{\omega_o^2} S^2(\omega) \tag{41}$$

for $S(\omega)$ being a skew-symmetric rotation matrix with $S^2(\omega) = \omega\omega^T - \omega_o^2 I_{3\times 3}$ has tremendous practical significance in physical gyroscopes and as a consequence great impact in navigation, in guidance, and in robotics, we note a 1971 equivalent precedent for the result in [1] is [29, Eq. 3]:

$$e^{S(\omega)t} = \cos(\omega_o t) I_{3\times 3} + \frac{\sin \omega_o t}{\omega_o} S(\omega) + \frac{1 - \cos \omega_o t}{\omega_o^2} \omega\omega^T. \tag{42}$$

Bortz [29] offers a physically motivated derivation of the above expression but [29, p. 61] acknowledges that a more abstract version of this expression had in fact been derived in 1949 in [30].

The result of Eq. 41 also arises naturally, in what is by now a routine application of the Cayley-Hamilton technique [20, pp. 281–285] based on the Cayley-Hamilton theorem [19, Eq. 7.47] "that every $n \times n$ matrix F satisfies its own characteristic equation" and, as a consequence (known as Sylvester's theorem [20, pp. 276–379]), any function of a matrix defined in terms of its countably infinite term matrix Taylor series can be reexpressed exactly in terms of a finite matrix series having no more than n terms and involving powers of F no higher than $n-1$ of the form:

$$e^{F\,t} = \sum_{i=0}^{n-1} \alpha_i(t)\, F^i = \tag{43}$$

$$\alpha_o(t)\, I_{n\times n} + \alpha_1(t)\, F + \alpha_2(t)\, F^2 + \cdots + \alpha_{n-1}(t)\, F^{(n-1)}.$$

When Eq. 43 is evaluated for $F = S(\omega)$ above, the result is Eq. 41. Evaluation of the n unknown scalar coefficients $\alpha_i(t)$ in Eq. 43 for any square matrix F proceeds using the fact that the above equality is also satisfied by each of the n eigenvalues of F (not necessarily distinct) as:

$$e^{\lambda_j\, t} = \sum_{i=0}^{n-1} \alpha_i(t)\, \lambda_j^i \;\; (\text{for } j = 1, \cdots, n)\,. \tag{44}$$

If the eigenvalues of F are distinct, then the linear system of Eq. 44 consists of n linear equations in the n scalar unknowns $\alpha_i(t)$ and the equations are independent (relating back to the Wronskian associated with the differential equation of Eq. 3 being nonsingular) and therefore can be explicitly solved for these n unknowns that, once obtained, are then used in Eq. 43 to compute the exact transition matrix $e^{F\,t}$ using powers of F no higher than $n-1$ (e.g., [19, pp. 205–208], [20, Ex. 4.10-6]).

The above is a useful special case of the more general LaGrange-Sylvester polynomial evaluation procedure [31, pp. 95-98], where for any $n \times n$ square matrix A, with associated minimum polynomial $m(A)$ of degree $m \leq n$, *any* function of a complex variable $f(z)$ (analytic on the spectrum of A) can be evaluated exactly for its matrix counterpart using just $m-1$ terms via the following intermediate arguments. By use of the remainder theorem (and its generalization to functions of a matrix offered below), the goal of evaluating the arbitrary analytic function $f(A)$ is related to the matrix remainder polynomial $r(A)$ as $f(A) = r(A)$ and its maximum degree is at most $m-1$. The details are summarized here to justify this result using the vehicle of an interpolation polynomial $I(z_i)$:

$$\begin{aligned}
I(z_i) &= exp(z_i), \\
I(z) &= m(z)q(z) + r(z), \\
I(z_s) &= \exp(z_s) = m(z_s)q(z_s) + r(z_s) = r(z_s), \\
\exp(A) &= I(A) = m(A)q(A) + r(A) = r(A),
\end{aligned} \tag{45}$$

where in the first line immediately above, there is interpolation at arbitrary z_i; in the second line, there is division by the minimum polynomial of A; in the third, points z_s are eigenvalues of A, and $m(z_s) \equiv 0$; and in the fourth and last, an exact representation occurs of $\exp(A)$.

Although the above technique is undoubtedly useful for hand calculation of the *exact* transition matrix e^{Ft} for lower dimensions of order 2 or 3, we observe that the following four complications interfere with using the above described technique as a general approach for computing e^{Ft} for an arbitrary $n \times n$ matrix F:

1. the coefficients $\alpha_i(t)$ are themselves functions of time t (rather than just constants [‖]) so symbolic manipulation must be used in solving for these;
2. in case F has any repeated eigenvalues, in order to have a system of n independent equations to solve from Eq. 44, derivatives with respect to the repeated eigenvalues must be taken on both sides of Eq. 44 (where the order of the required number of additional equations needed as an outcome of performing differentiations corresponds exactly to the degree of multiplicity of the repeated eigenvalues) and is related to using the confluent form of Sylvester's theorem for repeated eigenvalues [20, pp. 279–281];
3. eigenvalues of F must be known explicitly prior to use of this approach and so must be calculated beforehand (a computational burden that goes as $O(n^3)$ [32, p. 235]) while eigenvalue calculation is unnecessary in other conventional approaches to transition matrix calculation (viz. [2, Sect. III]);
4. this whole approach has severe sensitivity to errors incurred in eigenvalue calculation (which is likely aggravated for higher dimensions of n) but can be quantified by standard eigenvalue perturbational techniques [32], [33].

The 1970 investigation of [34] looked in detail at use of this method as the basis for machine calculation of e^{Ft} but dismissed it as not being practical (at that time). With the advent of Maple, MacSyma, Mathematica, and other symbol manipulation programs of this same ilk (that don't necessarily start with the letter M like Derive, Reduce, etc.), the difficulties of the above items (1) and (2) are mitigated and the advent of MatLab, IMSL, etc. for accurate calculation of eigenvalues (repeated or otherwise) may have current implications in mitigating items (3) and (4) as well to make this a more attractive present day approach for practically evaluating e^{Ft} to capture an accurate representation of the transition matrix for time-invariant linear systems (e.g., expm3 in MatLab).

While standard Kalman filter implementation equations (as now used for GPS/INS navigation and radar target tracking) were originally derived by R. E. Kalman in 1959-60 using the *Projection Theorem* in a *Hilbert Space* context (with prescribed inner product related to expectations), there are now comparable Kalman filter results for systems described by partial differential equations (PDE's) involving a type of PDE Riccati equation [35, Sect. 3.2] (with proofs of existence of solutions of Riccati PDE's being provided later by Ruth Curtain [71]). The natural framework for such infinite dimensional PDE formulations is within a *Banach Space* (being norm-based) and there are generalizations of idempotent matrices similar to those offered herein for these spaces as well [36] that allow closed-form solutions for infinite dimensional linear systems to confirm PDE implementations in S/W code as in MatLab^{TM} Toolbox, PDEASE^{TM}, etc. Other closed-form test case extensions have been developed in [2] and [3] for S/W verification of multichannel maximum entropy power spectral estimation algorithms and of nonlinear/estimation implementations of Extended Kalman filtering, respectively. Finally, a recent theoretical revelation regarding the use of the Singular Value Decomposition (SVD) here as one approach to computing the pseudoinverse as an intermediate step in constructing idempotent test matrices, the alternative historical *Greville* algorithm (utilized by Kalman and Englar in this same role in their 1962 WPAFB report discussed in [2, p. 950, Ref. 19] and in their 1966 NASA report [37]) without associated left or right eigenvector calculation has recently been revealed to be more efficient and accurate in calculating pseudoinverses directly than via SVD [38] (but lacking a systollic array implementation that SVD does possess).

10. STATUS OF 2-D KALMAN FILTERING

10.1. History of 2-D KF for Image Restoration

Generalizations of standard 1-D random process evolving in time or indexed on a single time variable (isomorphic to the real line so that it is *totally ordered* for simply distinguishing past from present from future [i.e., for any t_1

[‖]In computing just $e^{F'}$ without either time t or a time step in the exponent, the resulting coefficients α_i *are* merely constants, but having just $e^{F'}$ is useless for continuous-time or discrete-time computations unless the F' that is used has already absorbed the time step as $F' = F\Delta$. Moreover, the rich internal structure of Eqs. 43 and 44 is not revealed when the α_i's are just used as constant numbers.

and t_2, either $t_1 < t_2$, or $t_1 = t_2$, or $t_1 > t_2$] and having a standard unique definition of causality) have already been extended to 2-D [61] for Input/Output realizations. Early approaches to 2-D modeling usually invoked non-symmetric half-plane (NSHP) type causality merely for simplicity and convenience [53], [56].

The following representative milestones are recounted in briefly summarizing the generalization of Kalman filter formulations from 1-D to 2-D:

- Although Eugene Wong [40] alerts the reader in the mid 1970's and raises their level of consciousness to appreciate the difficulty of this problem (since the 2-D planar index of a random field can't be *totally ordered* for a clear unambiguous delineation of what's past, present, and future as can be done for the real line [as occurs for the time index of a random process]; however, the 2-D plane can be *partially ordered* but partial orderings are not unique and are also not wholly satisfying since there are several viable candidates that are reasonable to use but all have ambiguous "past", "present" (being a set rather than being a mere point, as occurs with a random process), and "future" defined, depending on which partial ordering convention is invoked). While [40] originally doesn't extend much hope for immediate resolution, a few years later he reports substantial progress in this area [41], [42].
- In the 1980's, Howard Kaufman along with his students and colleagues blazed an impressive development trail in further generalizations of 2-D Kalman filters specifically for image restoration applications [43]–[47]. In particular:
  - Quoting [43]: "it is established that for typical autoregressive signal models with nonsymmetric half-plane support, the dimension of the state size to be used within the Kalman filter is approximately equal to the product of the image model order and the pixel width of the image."
  - Quoting [46]: " a parallel identification and restoration procedure is described for images with symmetric noncausal blurs. It is shown that the identification problem can be recast as a parallel set of one dimensional ARMA identification problems. By expressing the ARMA models as equivalent infinite-order AR models **(sic) [the present author takes issue with this limiting claim and clarifies why in the first bullet in [39, Sect. 2.3] as a minor improvement]**, an entirely linear estimation procedure can be followed."
  - Quoting [47]: "it is established that an EKF for on-line parameter Identification was found to be unsuitable for blur parameter identification **(sic) [the present author takes issue with this limiting claim and clarifies why in the second bullet in [39, Sect. 2.3] as a minor improvement]** because of the presence of significant process noise terms that caused large deviations between the predicted pixel estimates and the true pixel intensities."
  - Quoting [45]: "model-based segmentation and restoration of images is performed. It was assumed that space-variant blur can be adequately represented by a collection of L distinct point-spread functions, where L is a predefined integer. (*The 'Multiple Model of Magill' (MMM)*) bank of parallel Kalman filters was applied to this problem but the Interactive Multiple Model (IMM) approach could be similarly applied as a more lucrative up-to-date algorithmic alternative."
  - Quoting [47]: "it is revealed that image restoration based upon unrealistic homogeneous image and blur models can result in highly inaccurate estimates with excessive ringing. Thus it is important at each pixel location to restore the image using the particular image and blur parameters characteristic of the immediate local neighborhood."

Also see more recent results and important extensions [63], [64].

10.2. Our Version of 2-D KF

The equation for 2-D optimal linear estimation of a scalar partial differential equation (PDE) system $\Psi(x, y)$ described over the x, y plane with boundary over the interval $[0, y_f]$ by:

$$\frac{\partial \Psi(x,y)}{\partial x} = \frac{\partial^2 \Psi(x,y)}{\partial y^2} + w(x,y) \,, \tag{46}$$

with boundary condition:

$$\alpha\Psi(x,y) = \frac{\partial\Psi(x,y)}{\partial y} \text{ at } y = 0 \text{ and at } y = y_f \tag{47}$$

for scalar $\alpha > 0$ and with $w(x,y)$ being additive Gaussian white process noise in the plane of positive semi-definite intensity $Q(x,y)$ and scalar sensor measurements:

$$z(x,y) = H\,\Psi(x,y) + v(x,y)\,, \tag{48}$$

with $v(x,y)$ being additive Gaussian white measurement noise in the plane of positive definite intensity $R(x,y)$, then the associated Riccati PDE to be solved as part of optimal linear KF estimation of linear systems described by Eq. 46 is

$$\frac{\partial P(x,y,t)}{\partial x} = \frac{\partial^2 P(x,y,t)}{\partial y^2} + \frac{\partial^2 P(x,y,t)}{\partial t^2} - \int_y P(x,t,x')HR^{-1}HP(x,y,x')\,dx' + Q(y,t)\,, \tag{49}$$

with corresponding boundary conditions:

$$\alpha P(x,y,t) = \frac{\partial P(x,y,t)}{\partial y} \text{ at } y = 0 \text{ and at } y = y_f\,, \tag{50}$$

$$\alpha P(x,t,y) = \frac{\partial P(x,t,y)}{\partial t} \text{ at } y = 0 \text{ and at } y = y_f\,, \tag{51}$$

and

$$P(x_o,t,y) = S(t,y) \text{ an initial condition.} \tag{52}$$

These can be solved by using rectangular discretization over the 2-D plane or by using the, by now, well-known Finite Element technique of Fix and Strang for specifying meshes and utilizing *PDEase* code or **MatLab**'s ** PDE code (both available on a PC). We generally follow the results pioneered by Kaufman for a single sensor but for real-time use we advocate synchronizing parallel processing using one processor for each local sensor filter and a final one for the Unification Collating Filter. Compare our equations to perspectives offered in [51] and [52, Ch. 7]. Our proposed Image Combining Rule is offered in [35]. Simulations to date have only been with a single processor to merely demonstrate proof of concept and images to date have been *Lena* at different resolutions with different levels of additive Gaussian White Noise superimposed to corrupt the image. We welcome a real data test case!

11. TRADING-OFF BENEFITS OF BLS FOR EWR TARGET TRACKING

The Batch Least Squares Maximum Likelihood algorithm [112]–[114], [115] is present at the core of many estimation approaches such as within the Prony method of power spectral estimation [109 , pp. 261–265], or within the method utilized by [110, Appendix , pp. 155–156] in Local Area Augmentation Systems (LAAS) for refined GPS navigation, or within input probing for improved parameter identification [111]. The BLS present in all situations exhibit the same fundamental characteristics in common.

Although not explicitly mentioned in [72] but merely implied or aluded to: the more accurate (non-optimistic) on-line prediction of 1-sigma BLS bounds helps properly constrain the region that the interceptor needs to search for target acquisition. Use of an optimistic bound in this role would result in improperly limiting search to too

One technical problem that we did encounter was with **MatLab's new capability to isolate level-crossing instant of either constant or specified time-varying thresholds with almost infinite precision. This **MatLab** capability actually exists only for completely deterministic situations since the underlying algorithms are predictor/corrector-based which are stymied when noise [albeit pseudo-random noise (PRN)] is introduced in the simulation. The presence of noise has been the bane of all but the coarsest and simplest of integration methodologies since the earliest days of digital simulation. However, engineering applications where threshold comparisons are crucial usually include the presence of noise, as in detection (i.e., is the desired signal present or just noise) in radar or communications, in Kalman filter-based failure detection or maneuver detection, or in peak picking as it arises in sonar processing and in image processing [62]. Other problems with calculation of matrix functions using matrix Signum functions, as occurs in some MatLab routines, are elucidated in **(1)** Barnett, S., "Comments on 'The Matrix Sign Function and Computation in Systems'," *Applied Mathematics and Computation*, Vol. 4, pp. 277–279, 1978; **(2)** Barrand, A. Y., "Comments on 'The Numerical Solution of $A^TQ + QA = -C$'," *IEEE Trans. on Automatic Control*, Vol. AC-24, No. 4, pp. 671–672, Aug. 1977. Also **(3)** Petkov, P. H., Christov, N. D., Konstantinov, M. M., "On the Numerical Properties of the Schur Approach for Solving the Matrix Riccati Equation, *System Control Letters*, Vol. 9, No. 3, pp. 197–201, 1987 for weakness in using the Schur method.

small a volume of space and therefore risk missing the target altogether, although supporting theoretical numerical calculations may falsely assure success (because they expect the available 1- sigma to be trustworthy, which it is not, in general, for the sub-optimal covariances provided from an EKF).

The EKF immediately avails outputted estimates in a more timely fashion and will follow any measurement data that it is provided with. The EKF is appropriate to use with the data association algorithm for multi-target tracking (MTT) because it is a fixed lesser CPU burden. On the other hand, the BLS algorithm provides more accurate estimates with a higher fidelity (more trust-worthy) on-line computed covariance accompanying its estimates for the same data segment length. Estimation errors are predicted from the on- line computation of 1-sigma bounds while the actual value was observed to be at the level 8 times higher for the representative scenarios that were investigated. The Batch algorithm on-line calculation predicts error bounds of a similar magnitude, however, BLS pays off by actually realizing estimation errors which are in the same vicinity.

However, BLS use incurs a larger computational burden and more associated senescence (computational delay time that is not fixed) than exhibited or needed by an in-place EKF (which has a delay time for computed output that is fixed and known [9, Sec. 7.6]). The known but not fixed BLS total processing demands always grow with the amount of measurement data collected for the particular track ID.

As stated above, BLS provides greater accuracy and a better (more trustworthy) on-line 1-sigma gauge of what the true accuracy actually is but is more sensitive and may not converge at all (i.e., may not produce any useable output if LMAX is exceeded because no ballistic curve could be fit to it) when the measurement data record is tainted.

The worrisome tainting may be because of standard model mis-match either caused by target data mis-associations, or by failure to correctly prune away the boosting segment, or by later presence of thrusting if it constitutes a significant portion of the data record, or by the effects of ionospheric scintillation. BLS expects data which it assumes matches its internal model of a pure ballistic target; otherwise, it may fail to converge. This is less of a problem the longer the measurement data segment is that BLS is provided with to operate on, as long as the dominant regime it represents is ballistic (and ionospheric scintillation errors have been approximately compensated for, as planned, or either are not dominant). The good news is that this better situation for BLS use will occur naturally if BLS is only applied to more mature tracks of interest (which corresponds to those with a longer data collection record).

There is still room for improvement of the EFK itself either via inclusion of more terms in the approximating Taylor series of the measurements, known as the Hessian; or by including a few additional iterations (2 or 3) of the measurement linearization [74]. Both these strategies should improve the accuracy of the measurement linearization with but a slight increase in the CPU burden. Another approach involving the Hessian for handling measurement structures like this involving a direct measurement of range (Widnall, 1973) is actually adaptive. Other options are to use different degrees of decoupling in the initial covariance (and Kalman gain as a conesquence) or to pursue exquisite analytic variations of EKF and its creative generalizations offered in [98]. Other more challenging AOT filter issues arise [77] when escort jammers accompany RV's.

A parallelized version of BLS could be even faster but that is a different architecture. Only sequential Von Neumann implementations were considered here. A parallelized version of the Householder transformation, at the heart of BLS, may be found in [95]. Lincoln Laboratory investigated this systolic array version of a parallel Householder algorithm in the late 1980's for radar signal processing but later converted to implement a cordic algorithm based on Givens rotations.

A parallel processing Fortran reference [89] indicates that interpolation benefits from being implemented on parallel processors by greatly speeding up the attainment of the goal. However, the clean lines of the original BLS algorithm were altered in [85] to replace interpolation with more frequent integrations between the available measurements that are logged. This version of BLS, written up for the s/w spec, presumes only a sequential Von Neumann implementation since it is impossible to accurately guess how a multi-threaded version would be partitioned across processors and performance also depends on target platform capabilities, its OS, and on final compiler settings.

Another possible variation on the original BLS is to use SVD in place of the Householder's transformation when solving the fundamental system of linear equations at the heart of the BLS algorithm. When possible, a "Householder transformation" should be used over SVD in well-behaved situations since Householder constitutes a lesser computational burden (being a voltage or squareroot method) rather than that of SVD (which is a power

method, as revealed in [86, App. B]). However, the vagaries of real data in some applications may warrant use of SVD for robustness.

BLS processes all the available measurements en masse and requires several iterations (albeit a small number, nominally 2) to do so. (The number of BLS iterations previously encountered for the interpolated version ranged as high as 8 but was nominally 3.) Intermediate arrays of transition matrices as well as regression arrays as well as the fundamental (9 by m) Aggregate Batch Measurement Matrix B must be accounted for in 8 byte double precision for these variable arrays for each track ID in process, where m is the number of measurements available for each track ID. As m increases, so must these intermediate arrays increase within the BLS algorithm. Memory for all but the above (9 x m) BLS B-matrix array can be released afterwards and re-allocated where needed. The actual BLS program code itself remains as a known fixed size.

Regarding sensitivity to the radar data that it is operating on, an EKF will follow any radar measurements it receives while the BLS tries to fit all the received data en masse to its internal model. The BLS internal model assumes that all the data be exclusively from a ballistic regime.

Timing is frequently performed for situations that were already known to converge. The real world is not so accommodating as to allow us to know the situation beforehand. Conservatism would dictate use of the worst-case number of iterations, LMAX, that can be incurred as a multiplier of these per measurement timing estimates. While the in-place EKF processes measurements one-at-a-time so CPU-time-per-a-measurement is a valid criteria for an EKF, the BLS does not process measurements one-at-a-time so timing estimates portrayed only this way are, at best, an intermediate approximation for BLS. For BLS, it is all or nothing in terms of measurement processing so CPU time should be interpreted for the whole measurement set that is being processed to be reasonable. BLS uses "all the measurements available to it all the time" when it is invoked.

The appropriate final CPU timing number corresponds to when BLS is merely called once. There is an upper bound worse case for this situation. In nominally benign situations (the prevalent case), the timing expression holds with 2 replacing LMAX as the appropriate pre-multiplying factor. Two expressions with these two alternate prefixes, can be used to bracket the actual CPU time from above and below. Use of both bounds together in this way is more conservative for performing predictions because the user does not have to know what the actual situation is beforehand regarding number of iterations to be incurred for BLS to converge for an upcoming trial.

The per measurement normalization for BLS is appropriate and consistent with numerical analysis theory for this main potential bottle-neck apparently facing BLS. The main problems being solved at the heart of each BLS iteration is the solution of a system of linear equations (the array of regression equations). Recall that this is the crux or fundamental kernel and the Householder transformation is used to solve it (as the algorithm of least computational complexity, which accomplishes the task at hand). Parallel implementations should be no slower than these estimates for a Von Neumann machine and parallel multi-threaded implementations may be considerably faster.

Operations Counts are available for a perfectly implemented sequential version of the Householder transformation from page 148 of [32]. The operations count incurred in applying back-substitution as Householder's transformation is being applied to solve the linear problem [ignoring for the moment any considerations related to obtaining the BLS covariance]), is: O(mn) flops, where n = 6 and m is the total number of measurements. For this aspect, averaging by dividing the previous expression by m to obtain a per measurement normalization yields the constant n= 6. The CPU burden is merely linear in the number of measurements and consistent with the above criteria selected of "per measurement evaluations". Since we also need the explicit upper triangular matrix in order to calculate the BLS covariance matrix (but don't need an explicit representation of the matrix transformation that gets us there), the numerical complexity of a Householder transformation in this case is greater for this aspect but still only linear in the measurements and still consistent with the criteria selected for conveying CPU time. Also from page 148 of [32], the expression for the Househholder operations counts in this case of providing an explicit U matrix is: $(n^2m - n^3/3)$ flops, where n = 6 and m is the total number of measurements. This operations count goes as m (the dominant power) and again just grows linearly with m. Averaging by dividing the previous expression by m to obtain an expected per measurement normalization yields a constant based on this numerical analysis theory. A similar invocation of a Householder transformation per a measurement depicted on page 252 of [87] also obtained a constant that is a cubic in the remaining fixed variable, being n^3. All evaluations hare are consistent.

12. NOT ART VERSUS SCIENCE BUT, RATHER, ART AUGMENTING SCIENCE

The people who may benefit from the following elaboration to lay out explicitly what the potential problems are likely to be in estimator or tracker implementation are the system engineers who may need to be made aware of possible interactions and cross-effects between Extended Kalman Filter (EKF) tracking behavior as it interacts with Multi-target Data Association algorithms (such as Multiple Hypothesis Testing (MHT), Jonker-Volgenant-Castanon, Munkres, generalized Hungarian) for solving the "assignment problem" of associating resources to tasks (like returns-to-targets spawned or sensor-to-sensor correlation for multi-sensor data fusion of same target seen). Such interaction may likewise occur if Interactive Multiple Model (IMM) filters are used in the architecture. IMM's are typically used to track targets exhibiting radically different modes of behavior (e.g., between the boost and ballistic regime for RV's).

For the ideal case of a LINEAR possibly time-varying system with additive Gaussian white process and measurement noises of known covariance intensities, with initial condition independent of the aforementioned noises and of specified mean and initial covariance, and satisfying certain technical regularity conditions (of being *Observable* and *Controllable* or less restrictive technical conditions of being merely *Detectable* and *Stabilizable*), the following seven properties below are associated with a correctly designed and implemented IDEAL Kalman filter:

1. the Kalman filter is an optimal LINEAR estimator and is the OPTIMAL estimator (according to five different criteria of goodness or Measures of Effectiveness) for tracking the state of the LINEAR system;
2. the estimation problem is completely solved using just the conditional mean and variance available on-line in real-time from the Kalman filter estimate and its associated Riccati equation solution, respectively. (Conditial refers to being conditioned on the sensor measurements received);
3. there is an analytically provable guarantee that the Kalman filter is stable and will converge to the true state (even if the underlying system being tracked is unstable) as obtained using Lyapunov functions;
4. the Kalman filter will converge exponentially asymptotically fast (this is darn quick) to the true state;
5. even if the initializing estimate x_0 and P_0 are way off (incorrect) but P_0 is still positive definite, THEN the Kalman filter STILL will CONVERGE quickly at the same exponential rate to the right answer (independent of how bad the initial guess or starting values were);
6. the on-line computed covariance (from the Joseph's form of the Riccati equation) is an excellent gauge or measure of how well estimation is proceeding and is even better (more accurate) in fact than statistics computed from any finite number of Monte-Carlo simulations or mission time records (remember that this statement is being made ONLY for the linear case);
7. the IMM is only an approximate estimator for even a purely linear system (incurring two significant levels of approximation) and it additionally has accompanying probabilities calculated as an on-line indication of which of the IMM models is instantaneously more favored (where the structure of such probability calculation equations depends even more on the structure of having additive GWN and an underlying linear structure and so can be a source of sensitivity beyond merely invoking approximating EKF's, as is typically done to handle nonlinear situations).

For NONLINEAR systems, all seven of the above bets are off! All are violated in general!

Strategic target tracking typically employes a system model that is nonlinear with inverse square gravity along with its second zonal harmonic in the describing Ordinary Differential Equations (ODE) and is also nonlinear in the observation equation as well to compensate for range-Doppler ambiguity in the plane of the antenna face. EKF's are typically used for tracking the state in a ballistic trajectory.

To explicitly draw the distinction between what to expect for a tracker for the above postulated IDEAL linear case and the more realistic nonlinear case encountered in practice, I now discuss the situation of approximate NONLINEAR filtering paralleling the presentation offered above for the linear case:

1. the Optimal nonlinear filter is infinite dimensional, in general, and therefore not practical to attempt to compute (taking possibly an infinite amount of time to do so) while a reasonable engineering approximation is to, instead, employ is an Extended Kalman Filter as a best LINEAR estimator (but not expected to be an OPTIMAL estimator but, hopefully, adequate for tracking the state of the NONLINEAR system);

2. the estimation problem is NOT completely solved using just the conditional mean and variance available on-line in real-time from the Extended Kalman filter estimate and its associated Riccati solution, respectively. Hopefully, the estimate will be adequate but its intermediary variance usually is not. Unlike the situation for the linear case where everything is completely characterized by just the estimator mean and variance, the actual optimal estimator needs all higher moments specified as well (or, equivalently, specification of the conditional pdf). The on-line variance can be *optimistic* (smaller than actual) or *pessimistic* (larger than actual) and may criss-cross several times over a tracking time interval between being one or the other. The focus is on the adequacy of just the state estimate as the primary consideration.

3. there is NO LONGER any analytically provable guarantee that the EKF is stable and will converge to the true state. Unfortunately, EKF sometimes diverge;

4. the EKF doesn't converge exponentially asymptotically fast to the true state. We are happy if it gets there fast enough to be useful;

5. when the initializing estimate x_0 and P_0 are way off (incorrect) but P_0 is still positive definite, THEN the EKF may DIVERGE away from the right answer at an exponential rate. (EKF performance can be highly dependent of how good or bad the initial guess or starting values were);

6. the on-line computed covariance (from the Joseph's form of the Riccati eqn) is a LOUSY gauge or measure of how well estimation is proceeding and is NEVER better (or AS accurate) as the off-line statistics computed from an adequately large finite number of Monte-Carlo simulations or mission time records. (Employing a 97% histogram-based Spherical Error Probable [SEP] from as much as 250 Monte-Carlo run evaluations is not atypical in some applications.)

7. the IMM is even more suspect since it is only an approximate estimator for even a purely linear system (incurring two significant levels of approximation) and it additionally has accompanying probabilities calculated as an on-line indication of which of the IMM models is instantaneously more favored (where the structure of such probability calculation equations still depends on the underlying structure of having additive GWN and an underlying linear structure preserves Gaussianity but Nonlinear systems do not preserve Gaussianity and so can be a source of sensitivity beyond merely invoking approximating EKF's, as is typically done by tracking specialist to handle nonlinear situations).

The chasm between the above two Items 7 as spelled out for the linear and nonlinear cases is our biggest current concern.

On the topic of IMM versus earlier architectures such as Magil Multiple Model (MMM) bank-of-filter approach, recall the older (1965) MMM technique. MMM has output that picks the best candidate, IMM blends them together (but IMM is potentially more responsive to changes and willing to entertain the alternative filter model candidates that have been hypothesized and enunciated by virtue of the presence of the sojourn time and the finite state Markov chain transition probabilities that keeps the alternative filter models viable and active). Bar-Shalom, himself, warns that some people don't like the answers from IMM because, instead of yielding black or white selections or decisions as to the appropriate model, it instead yields "shades of gray". I neglected to mention that the concerns that I expressed above about the two levels of severe approximations incurred by IMM, even for the totally linear case (my Item 7 above), were not just my baseless allegations but were actually explicitly admitted on page 32 of Y. Bar-Shalom and X. Rong Li, Multitarget-Multisensor Tracking: Principles and Techniques, YBS, Storrs, CT, 1995. However, they, perhaps, "guilded the lilly" by phrasing it as, hey, IMM is really a GPB1 algorithm and its performance is about as good as a GPB2. In reality, GPB2 can be lousy as well. This is like saying "using the InterNet to purchase items using your credit card or trading stock is about as safe as, say, on the U.S. DOD computers on the Net". That's the line that *Charles Schwab* uses. (In reality, however, security is actually horrible for both!) The previous statement doesn't really endorse use of IMM per se but is totally truthful as it stands.

There was the ideal (which is too large and impractical to calculate since it is infinite dimensional), then GPBn (one level of approximation down but finite dimensional yet can be large, depending on the size of n specifying the depth of hypotheses retained in making current decisions), then there is IMM (yet another extreme level of approximation down) being identically GPB1. (GPB stands for Generalized Pseudo-Bayesian Estimator and is of Computational complexity $O(r^n)$ for a GPBn algorithm where r is the number of candidate filter calculations to be performed and n is the depth of prior hypothesis data retained in making current decisions with the algorithm.

New results but still ONLY for linear models of the dynamics (ODE) but sometimes nonlinear in the measurements appear in Yaakov Bar-Shalom and William Dale Blair (Editors) book entitled 'Multitarget-Multisensor Tracking: Applications and Advances," Vol. III, Artech House Inc., Boston, 2000, which offers the following perspectives: Although each successive installment of this IMM series is advertised as becoming more practical, every chapter development (written by contributing authors) still invokes ONLY a linear dynamics model for the evolution of the state in applying IMM. (Of course, in order to track strategic targets, it is well know that we need to use a nonlinear model consisting of inverse square gravity and at least the second harmonic of gravity, J_2 but we don't see such models in any book yet for IMM. The other standard consideration is compensating for range-Doppler r-rdot ambiguity in the coordinates of the antenna face plane in *sine space.*) The newly available unclassified open system MatLab code for TMD uses the Modified J-V-C, denoted as MJV by leaving off the important final contribution due to Prof. David Castenon (Boston Univ.) that greatly improved the efficiency of the original J-V-C in solving the "assignment problem" of *Operations Research* in situations where sparse matrices prevail as being dominant (as present in both the TMD and NMD applications). As of 6/30/01, Prof. Castenon suspects that full J-V-C for TMD could be achievable with more effective use of MatLab's standard *sparse matrix features* within this TMD version of the algorithm. Other concerns about the current status of advertised MatLab processing and compilation options are expressed in [84]. One can confirm the invoking of only linear IMM models for the system dynamics in the above cited book from this exhaustive list: p. 31, eq. 1.56; p. 83, eq. 2.2; p. 132, eq. 2.85; p. 164, eq. 3.1; p. 214,eq. 4.21; p. 237, eq. 5.2 looks completely general and nonlinear but their only examples are linear as p. 237, eq. 6.5; and p. 240, eq. 5.9; p. 269; eq. 6.13 uses an Extended Kalman Filter but again only the measurement equation is nonlinear; p. 326, eq. 7.1 has the same situation as mentioned immediately above; (Blair and Keel have a nice concise overview of radar system considerations for tracking in Chapter 7 but their system dynamics model is again linear, as cited just above). Also notable is a nice section (Chapt. 8) on Countermeasure considerations and how they specifically affect tracking simulations and implementations. This and the previously cited radar system considerations motivated my purchase of this book (especially the ECM and ECCM of Tables 8.1, 8.2, 8.3). Resuming the list of more linear-ONLY dynamics models on: p. 470, eq. 9.20; p. 501, eqs. 10.3 & 10.6; p. 509, Table 10.1; p. 520, eq. 10.23; eq. 10.24; p. 540, eq. 10.41. All other previous papers and textbooks on IMM theory and applications have, to date, invoked only linear models for the system dynamics although some had nonlinear measurement models present. Rong Li's 1993 and '95 versions of DynaEst don't have *sojourn times* included within the s/w although his theoretical IMM discussions of the last 10 years do.

13. CONCLUSIONS

We explained why our proposed new idempotent matrix approach is superior in S/W verification to the well-known conventional approach of invoking specific similarity transformations to obtain closed-form solutions via conversion to a diagonally (and, perhaps, super-diagonally) banded *Jordan canonical form* by **(1)** being a lesser computational burden; **(2)** incurring less roundoff; and **(3)** exhibiting no ambiguity when repeated eigenvalues are encountered (unlike the Jordan approach, where encountering repeated eigenvalues is more challenging and more computationally burdensome to correctly decipher by requiring a *confluent* form to be invoked which is **seldom** straight forward). Other numerical analysis issues associated with use of these examples (based on idempotent matrices) were investigated such as indications of ill-conditioning conveyed by somewhat controversial condition numbers and demonstrated such concerns are insignificance for these idempotent test cases.

Other useful extensions were also offered here including hints from [36] (using [65]–[67], as will be made more explicit by me in SPIE small targets 2001) on how to appropriately generalize this same *idempotent matrix-based* methodology for software verification of ordinary differential equation (ODE) implementation to also handle S/W verification of infinite dimensional partial differential equation (PDE) implementations (as can be mechanized with the MatLab^{TM} PDE Toolbox or as in PDEASE^{TM}).

Ideas for handling S/W verification of nonlinear estimation (also potentially infinite dimensional) have already been worked out [4] (also see other claims along these same lines in a paper entitled "Range-Angle Coupling in Target Tracking" by Robert J. Fitzgerald (Raytheon) at SPIE Small Targets in December 1999).

ACKNOWLEDGMENTS

We became alerted to many of these aspects from practical experience and from realistic simulations[76] using **TK-MIP 2.0**, a product of **TeK Associates**, available commercially for performing Kalman filter analysis and simulation (and even actual on-line implementation via use of Data Acquisition Cards, or serial port input, and/or PCI) from an easy to use Graphical User Interface (GUI). On-line tutorials and extensive application examples are also available for **TK-MIP** including an on-line self-contained professional level textbook and short course complete with lectures, tests, corresponding answers, and a guest lecturer. This software runs on 80386 or later Personal Computer (PC) processor with hardware math co-processor chip under *Microsoft Windows 95/98/NT/ME/2000* (32-bit) operating systems.

TK-MIP is a software product for the PC (without MatLab) that we recently developed for teaching others about the theory and practice of KF simulation [72]–[85] and for actually implementing KF technology and its many variations on-line for linear and nonlinear estimation and tracking.

APPENDIX A. NEW CONSIDERATIONS REGARDING BIERMAN'S AND CARLSON'S UPDATED SQUAREROOT FILTERING

Squareroot filtering is a convenient and practical contrivance used to obtain an effective double precision implementation of a Kalman filter without having to actually resort to explicit implementation in double precision but merely by use of an alternate implementation (in single precision) of the factors of the covariance matrix being propagated forward in time. The so-called Bierman's form or $U-D-U^T$ form of squareroot filtering [87] (which propagates U and a diagonal D) had historically proved to be the best formulation up until the late 1990's in that it is a numerically stable implementation (an important consideration for long duration on-line run times) and has the smallest number of required operations and does not call for the implementation of explicit scalar square-roots (as earlier squareroot implementations did). Earlier versions of squareroot filtering constituted much larger computational burdens than conventional Kalman filtering; however, the Bierman implementation was no worst a computational burden than the standard KF implementation but offers an effective doubling in the precision availed (and with guarantees of numerical stability [which a standard Kalman Filter implementation lacks]). For more detail, see Chap. 7 (and, in particular, the comparisons of Table 7.1) on p. 403 of [9]. Other important considerations in $U-D-U^T$ squareroot filltering are addressed in [99], which shows how to rigorously handle vector measurements in $U-D-U^T$ filters. Ref. [100] discusses how to capitalize on certain efficiencies that arise in Navigation applications involving GPS as an external navaid for an Inertial Navigation System (INS). Ref. [101] demonstrates how to incorporate handling of manual updates (known by military pilots and navigators as "MARK ON TOP" as a procedure that uses landmarks of known location to update position in an aircraft's internal INS filter at the more or less "precise moment" that the aircraft flies over the landmark). Ref. [102] was one of the first designs that clearly demonstrated the details of how to accommodate a $U-D-U^T$ filter formulation within a NAVSTAR GPS application.

Bierman's $U-D-U^T$ Squareroot filter formulation was preceded by or evolved from other efforts at deriving squareroot filter formulations by J. Potter (1964), J. F. Bellantoni and K. W. Dodge (1967), A. Andrews (1968), P. Dyer and S. McReynolds (1969), P. G. Kaminski and A. E. Bryson and S. I. Schmidt (1971), W. S. Agee and R. H. Turner (1972), and N. Carlson (1973). Bierman's formulation (1974, '75, '77) had originally proved to be the best of the lot for embedded architectures where explicit scalar square root extraction was a much more time consuming algorithm until computer architectures surfaced in the late 1990's where this was no longer the active constraint in force and now scalar squareroot calculation is about the same as a floating point multiply thus now favoring Carlson's [103] over Bierman's formulation. There are also other recent contenders and logical extensions [104]–[107].

For situations where the discrete-time dynamic state variable system model is of the form:

$$x(k+1) = Ax(k) + Fw(k) + Bu(k), \text{ with initial condition : } x(0) = x_o \tag{53}$$

with (optional) deterministic control input (i.e., exogenous input) being present and the discrete-time sensor data measurement observation model is of the form:

$$z(k) = Cx(k) + Gv(k), \tag{54}$$

where $w(k)$ and $v(k)$ are independent Gaussian white noises (GWN) with intensity variances of Q and R, respectively. For the purpose of further reducing the adverse effect of round-off error accumulation and to avoid explicit calculation of the matrix inverse within the Kalman filter by using a degenerate scalar form that is, instead, only a division, it is frequently desired to update squareroot filters using only one-scalar- measurement-component-at-a-time, but the standard procedure for doing so is only valid if R is diagonal (corresponding to uncorrelated measurement noise) and G is the identity matrix. In

the more general case where both of these conditions fail to be met, yet the user still wants to update the filter one scalar measurement component at a time, the following simple (possibly time-varying) transformation can be applied to achieve the desired structual objective for single-component-at-a-time updating. Merely form $[G(k)R(k)G^T(k)]$ and decompose it via a Choleski decomposition into $[G(k)R(k)G^T(k)] = W(k)W^T(k)$, where $W(k)$ is lower triangular, then just pre-multiply the entire measurement equation above to obtain

$$z_1(k) \triangleq W^{-1}(k)z(k) = [W^{-1}(k)C(k)]x(k) + [W^{-1}(k)G(k)]v(k), \tag{55}$$

and we have that

$$[W^{-1}(k)G(k)R(k)G^TW^{-1}(k)] = W^{-1}(k)W(k)W^T(k)W^T(k) = I_{mxm}, \tag{56}$$

where I_{mxm} is the mxm identity matix. The original Kalman filter, described recursively by the following eqn., driven by the measurement $z(k)$ and control as:

$$\hat{x}(k+1|k) = \Phi(k+1,k)[I - K(k)C]\hat{x}(k|k-1) + \Phi(k+1,k)K(k)z(k) + Bu(k) \tag{57}$$

and with on-line propagate covariance of estimation error equation being:

$$P(k|k) = [I - K(k)C]P(k|k-1)[I - K(k)C]^T + K(k)GR_3G^TK(k)^T \tag{58}$$

and with on-line update covariance of estimation error equation being:

$$P(k|k-1) = \Phi(k,k-1)P(k-1|k-1)\Phi^T(k,k-1) + FQ_3(k)F^T, \text{ (then } k = k+1) \tag{59}$$

with the standard discrete-time Kalman gain being:

$$K(k) = P(k)C^T[CP(k|k-1)C^T + GR_3G^T] - 1. \tag{60}$$

The above four equations are now modified for one-component-at-a-time filtering as the following equivalent Kalman filter driven by the transformed measurement $z_1(k)$ and same deterministic control $u(k)$, respectively, as:

$$\hat{x}(k+1|k) = \Phi(k+1,k)[I - K'(k)W^{-1}C]\hat{x}(k|k-1) + \Phi(k+1,k)K'(k)z_1(k) + Bu(k) \tag{61}$$

and

$$P(k|k) = [I - K'(k)W^{-1}C]P(k|k-1)[I - K'(k)W^{-1}C]^T + K'(k)K'^T(k) \tag{62}$$

and

$$P(k|k-1) = \Phi(k,k-1)P(k-1|k-1)\Phi^T(k,k-1) + FQ_3(k)FT, (then k = k+1) \tag{63}$$

with the new discrete-time Kalman gain being:

$$K'(k) = P(k)C^TW^{-T}[W^{-1}CP(k|k-1)C^TW^{-T} + I]^{-1}. \tag{64}$$

Hint: If noises $w(k)$ and $v(k)$ are present but matrices F and/or G are not apparent in the defining system and measurement models; then, obviously, $F = I_{nxn}$ and $G = I_{mxm}$.

Again, the new wrinkle of the late 1990's in Square Root Filtering is that within new processor chips, the algorithm for performing explicit scalar squareroots is no longer an iterative mechanization but now is just as fast as multiplication or addition operations. The prior motivation to use a particular version of Square Root Filtering, based on operation counts that penalized explicit computation of scalar squareroots is no longer viable for implementation processors that calculate the squareroot this new way. Motivation still exists to use a Square Root Filtering structure for real-time implementations with long run times because these squareroot formulations are still the only numerically stable implementation of a Kalman filter. Its use avoids such contrivances as inserting stops to prevent any main diagonal terms of the covariance matrix from eventually becoming negative as a consequence of adverse effects of accumulated round-off errors within the more straight forward implementation of the easy-to-read conventional Kalman filter (although possibly "Stabilized" by adding the transposed covariance to itself and dividing by 2). Every navigation application should be using a Square Root Filter formulation in the case of long run times. However, target tracking for strategic missiles may not require such stringent mechanizations because main targets of interest (RVs) don't persist for long enough time intervals to warrant Square Root Filter use. New targets trigger new filter starts. Implementation needs for radar tracking of persistent cooperative FAA targets can be a different story. The longer "Control Segment" tracking intervals for GPS satellite ephimerii drift definitely do use Square Root Filter formulations. Evidently, whether or not to use Bierman's $U - D - U^T$ or Carlson's squareroot filter formulation in a particular application should be decided on a case-by-case basis.

The usual benefits touted for the use of a U-D is that (1) it is numerically stable (it mitigates the build-up of round-off errors), while other formulations devoid of square root filtering are not numerically stable; (2) it exposes problematic effects early on as evidenced by algorithmically examining the principal diagonals of the propagated matrix D at each time step, and

(3) it effecttively doubles the effective precision of the on-line computed covariances and the associated estimator's gain as a consequence (although this last reason is usually the primary reason for using a U-D formulation, it may not be necessary in some applications where the register size is already adequate). The CPU burden for a U-D-U square root filter is no greater than that of an ordinary Kalman filter but cross-checking proper implementation is a little more challenging because it's output estimates and covariances are usually compared to those of a non-square root filter for the short term (when both should be identical) but only the U-D square root formulation will be adequate for the long term of longer missions and frequent measurements being obtained. Finally, there is even a parallel processing implementation of it available [108].

REFERENCES

1. Bernstein, D. S., and So, W., "Some explicit formulas for the matrix exponential," *IEEE Transactions on Automatic Control,* Vol. AC-38, No. 8, pp. 1228-1232, August 1993.
2. Kerr, T. H., "A constructive use of idempotent matrices to validate linear systems analysis software," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. AES-26, No. 6, pp. 935-952, Nov. 1990 (corrections in Vol. AES-27, pp. 951-2, '91 & Vol. AES-33, No. 2, pp. 862-3, Apr. '95).
3. Kerr, T. H., "Rationale for Monte-Carlo simulator design to support multichannel spectral estimation and/or Kalman filter performance testing and software validation/verification using closed form test cases," MIT Lincoln Laboratory Project Report No. PA-512, Lexington, MA, 22 December 1989.
4. Kerr, T. H., "Numerical approximations and other structural issues in practical implementations of Kalman filtering," in *Approximate Kalman Filtering,* edited by Guanrong Chen, World Scientific, London, pp. 193-222, 1993.
5. Moler, C., and Van Loan, C., "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review,* Vol. 20, No. 4, pp. 801-836, October 1978.
6. Ward, R. C., "Numerical Computation of the Matrix Exponential with Accuracy Estimate," *SIAM J. of Numerical Analysis,* Vol. 14, No. 4, pp. 600-610, September 1977.
7. Sage, A. P., and White, C. C., *Optimum Systems Control,* Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1977.
8. Gelb, A. (ed.), *Applied Optimal Estimation,* MIT Press, Cambridge, MA, 1974.
9. Maybeck, P. S., *Stochastic Models, Estimation, and Control,* Vol. 1, Academic Press, N.Y., 1979.
10. Kwakernaak, H., and Sivan, R., *Applied Optimal Estimation,* Wiley-Interscience, John Wiley & Sons, NY, 1972.
11. Rimer, M., and Fredrick, D. K., "Solutions of the Grumman F-14 benchmark control problem," *IEEE Control Systems Magazine,* Vol. 7, pp. 36-40, August 1987.
12. Huang, C., and Fredrick, D., "Solutions to the third benchmark control problem," *Proceedings of 1991 Automatic Control Conference,* Vol. 1, pp. 976-977, Boston, MA, 26-28 June 1991.
13. Blum, M., and Wasserman, H., "Program Result-Checking: A Theory of Testing Meets a Test of Theory," *Proceedings of* 35^{th} *Annual IEEE Symposium on Foundations of Computer Science,* Vol. 35, pp. 382-393, Santa Fe, NM, Nov. 1994.
14. Bierman, G. J., "A Remark Concerning Discrete Approximation to White Noise and Covariance Calculation," *IEEE Transactions on Automatic Control,* Vol. AC-14, No. 4, pp. 432-433, 1969.
15. B. C. Kuo and D. W. Peterson, "Optimal discretization of continuous-data control system," *Automatica,* Vol. 19, No. 1, pp. 125-129, January 1973.
16. K. B. Janiszowski, "A modification and the Tustin approximation," *IEEE Transactions on Automatic Control,* Vol. AC-38, No. 8, pp. 1313-1316, August 1993.
17. S. C. Gupta, "Phased-locked loops," *Proceedings of the IEEE,* Vol. 68, No. 2, February 1975.
18. Galvin, W. P., "Matrices with 'Custom-Built' Eigenspaces," *SIAM Mathematical Monthly,* pp. 308-309, May 1984.
19. Gupta, S. C., *Transform and State Variable Methods in Linear Systems,* John Wiley & Sons, NY, 1966.
20. DeRusso, P. M., Roy, R. J., and Close, C. M., *State Variables for Engineers,* John Wiley & Sons, NY, 1967 (2nd Edition, 1998).
21. Daum, F. E., and Fitzgerald, R. J., "Decoupled Kalman Filters for Phased Array Radar Tracking," *IEEE Trans. on Automatic Control,* Vol. AC-28, pp. 269-283, Mar. 1977.
22. Shi, J., and Malik, J.' "Normalized Cuts of Image Segmentation," *IEEE Conf. on Computer Vision and Pattern Recognition,* Puerto Rico, June 1997.
23. Kerr, T. H., "Computational techniques for the matrix pseudoinverse in minimum variance reduced-order filtering and control," in *Control and Dynamic Systems-Advances in Theory and Applications,* Vol. XXVIII: Advances in Algorithms and computational Techniques for Dynamic Control Systems, Part 1 of 3, C. T. Leondes (Ed.), Academic Press, N.Y., pp. 57-107, 1988.
24. Burlington, R. S., *Handbook of Mathematical Tables and Formulas,* McGraw-Hill, NY, 1962.
25. Healy, M. J. R., *Matrices for Statistics,* Clarendon Press, Oxford, England, 1986.
26. Kerr, T. H., "An analytic example of a Schweppe likelihood ratio detector," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. AES-25, No. 4, pp. 545-558, July 1989.
27. Ramachandra, K. V., "Optimum steady state position, velocity, and acceleration estimation using noisy sampled position data," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. AES-23, No. 5, pp. 705-708, September 1987 (a correction appears in 1989 issue).
28. Lee, E. B., and Markus, L., *Foundations of Optimal Control Theory,* John Wiley & Sons, NY, 1967.

29. Bortz, J. E., "A new mathematical formulation for strapdown inertial navigation," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. AES-7, No. 1, pp. 61-66, January 1971.
30. Laning, J. H., "The vector analysis of finite rotations and angles," Massachusetts Institute of Technology/Instrumentation Laboratory, Cambridge, MA, MIT/IL Special Report No. 6398-S-3, 1949.
31. Gantmacher, F. R., *Matrix Theory,* Vol. 1, Chelsea, NY, 1959.
32. Golub, G. E., and Van Loan, C. F., *Matrix Computations,* Johns Hopkins University Press, Baltimore, MD, 1983.
33. Stewart, G. W., *Introduction to Matrix Computations,* Academic Press, NY, 1973.
34. Vidyasagar, M., "A novel method of evaluating e^{At} in closed form," *IEEE Transactions on Automatic Control,* Vol. AC-15, No. 5, pp. 600-601, October 1970.
35. Kerr, T. H., "Extending Decentralized Kalman Filtering (KF) to 2-D for Real-Time Multisensor Image Fusion and/or Restoration: Optimality of Some Decentralized KF Architectures," *Proceedings of the International Conference on Signal Processing Applications & Technology,* Boston, MA, pp. 155-170, 7-10 Oct. 1996.
36. McCullough, S. A., Rodman, L., "Hereditary Classes of Operators and Matrices," *The American Mathematical Monthly,* Vol. 104, No. 5, pp. 415-430, May 1997.
37. Kalman, R.E., and Englar, T. S., *A User's Manual for the Automatic Synthesis Program (Program C),* NASA Contractor Report No. NASA CR-475, June 1966.
38. Caspary, O., and Nus, P., "Implementation of the Greville Algorithm on a Motorola DSP96002 Application to Least-Squares Problems," *Proceedings of the International Conference on Signal Processing Applications & Technology,* Boston, MA, pp. 142-145, 7-10 Oct. 1996.
39. Kerr, T. H., "Extending Decentralized Kalman Filtering (KF) to 2-D for Real-Time Multisensor Image Fusion and/or Restoration," *Signal Processing, Sensor Fusion, and Target Recognition V,* Proceedings of SPIE Conference, Vol. 2755, Orlando, FL, pp. 548-564, 8-10 April 1996.
40. Wong, E., "Recursive Filtering for Two Dimensional Random Fields," *IEEE Trans. on Information Theory,* pp. 84-86, January 1975.
41. Wong, E., Tsui, E. T., "One-Sided Recursive Filters for Two-Dimensional Random Fields," *IEEE Trans. on Information Theory,* pp. 596-599, Sept. 1977.
42. Wong, E., "Recursive Causal Linear Filtering for Two-Dimensional Random Fields," *IEEE Trans. on Information Theory,* pp. 50-59, January 1978.
43. Angwin, D., Kaufman, H., "Image Restoration Using a Reduced Order Kalman filter," *Proceedings of ICASSP '88,* NY, pp. 1000-1003, 11-14 April 1988.
44. Kaufman, H., Tekalp, A. M., "Survey of Estimation Techniques in Image Restoration," *IEEE Control Systems Magazine,* Vol. 11, No. 1, pp. 16-24, January 1991.
45. Tekalp, A. M., Kaufman, H., and Woods, J., "Edge-Adaptive Kalman Filtering for Image Restoration with Ringing Suppression," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. ASSP-37, No. 6, pp. 892-899, June 1989.
46. Biemond, J., Kaufman, H., "Identification and Restoration Using Parallel Kalman Filters," *Proceedings of* 26^{th} *IEEE Conf. on Decision and Control,* Los Angeles, CA, pp. 1198-1203, 9-11 December 1987.
47. Koch, S., Kaufman, H., Biemond, J., "Restoration of Spatially Varying Blurred Images Using Multiple Model-Based Extended Kalman Filters," *IEEE Trans. on Image Processing,* Vol. 4, No. 4, pp. 520-523, April 1995.
48. Iyengar, S. S., Presard, L., Min, H. (eds.), *Advances in Distributed Sensor Integration: Application and Theory,* Prentice-Hall PTR, Upper Saddle River, NJ, 1995.
49. Saha, R. K., "Effect of Common Process Noise on Two-Sensor Track Fusion," *AIAA Journal of Guidance, Control, and Dynamics,* Vol. 19, No. 4, pp. 829-835, July-August 1996.
50. Pao, L. Y., "Measurement Reconstruction Approach for Distributed Multisensor Fusion," *AIAA Journal of Guidance, Control, and Dynamics,* Vol. 19, No. 4, pp. 842-847, July-August 1996.
51. Sebek, M., "Polynomial Solution of 2-D Kalman Bucy Filtering Problem," *IEEE Transactions on Automatic Control,* Vol. AC-37, No. 10, pp. 1530-1532, October 1992.
52. Sage, A., *Optimum Systems Control,* Prentice-Hall, Englewood Cliffs, NJ, 1968.
53. Andrews, H. C., and Hunt, B. R., *Digital Image Restoration,* Prentice-Hall, Englewood Cliffs, NJ, 1977.
54. Dudgeon, D. E., and Mersereau, R. M., *Multidimensional Digital Signal Processing,* Prentice-Hall, Englewood Cliffs, NJ, 1984.
55. Bose, N. K., *Multidimensional Systems Theory,* D. Reidel Publishing Co., Boston, 1985.
56. Pratt, W. K., *Digital Image Processing,* 2^{nd} Edition, John Wiley & Sons, Inc., NY, 1991 (originally 1978).
57. Baxes, G. A., *Digital Image Processing: Principles and Applications,* John Wiley & Sons, Inc., NY, 1994.
58. Salt, E., and Yuam, Y.-X., "On the Accuracy of Approximations in the Error Analysis of Registration Systems," *IEEE Trans. on AES,* Vol. AES-27, No. 6, pp. 922-924, Nov. 1991.
59. Covino, J. M., and Griffiths, B. E., "A New Estimation Architecture for Multisensor Data Fusion," *PROCEEDINGS of SPIE (Society of Photo-Optical Instrumentation Engineers),* J. Wade and A. Tuchman (Eds.), Volume 1478, pp. 114-125, Orlando, FL, 2-3 April 1991.
60. Longo, M., Lookabaugh, T. D., and Gray, R. M., "Quantization for Decentralized Hypothesis Testing Under Communication Constraints," *IEEE Transactions on Information Theory,* Vol. IT-36, No. 2, pp. 241-255, March 1990.

61. Bisiacco, M., Fornasini, E., and Zampieri, S., "Realization and Partial Realization of 2-D Input/Output Maps," *New Trends in Systems Theory*, Proceedings of Joint Conference at Universita di Genova-Ohio State University, G. Conte, A. M. Perdon, B. Wyman (eds.), pp. 120-127, Birkhauser, 9-11 July 1990.
62. Zakhor, A., Oppenheim, A. V., "Reconstruction of Two-Dimensional Signals from Level Crossings," *Proceedings of the IEEE*, Vol. 78, No. 1, pp. 31-55, Jan. 1990.
63. Azimi-Sadjadi, M. R., "New Results in Strip Kalman Filtering," *IEEE Trans. on Circuits and Systems*, Vol. 36, No. 6, pp. 893-897, June 1989.
64. Azimi-Sadjadi, M. R., "Estimation and Identification for 2-D Block Kalman Filtering," *IEEE Trans. on Signal Processing*, Vol. 39, No. 8, pp. 1885-1889, Aug. 1991.
65. Moon, P., Spencer, D. E., *Field Theory for Engineers*, D. Van Nostrand Co. Inc., Princeton, NJ, 1960.
66. Philip, J. R., Knight, J. H., "Table Errata for Field Theory Handbook, 2nd Ed. by P. Moon and D. E. Spencer" *Mathematics of Computation*, pp. 651-653, Oct. 1987.
67. Miller, K. S., *Some Eclectic Matrix Theory*, Robert E. Krieger Publishing Company, Malabar, FL, 1987.
68. Kerr, T. H., "On Duality Between Failure Detection and Radar/Optical Maneuver Detection," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-25, No. 4, pp. 581-583, July 1989.
69. Doyle, J. C., "Guaranteed Margins for LQG Regulators?," *IEEE Trans. on Automatic Control*, Vol. 23, No. 4, pp. 756-757, Aug. 1978.
70. Astrom, K. J., Hagguend, T., "Automatic Tuning of Simple Regulators with Specification on Phase and Amplitude Margins," *Automatica*, Vol. 20, No. 5, pp. 645-651, 1984.
71. Curtain, R. F., Pritchard, A. J., *Infinite Dimensional Linear Systems Theory*, Springer-Verlag, NY, 1978.
72. Kerr, T. H., "Status of CR-Like Lower bounds for Nonlinear Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-25, No. 5, pp. 590-601, September 1989 (Author's reply in Vol. AES-26, No. 5, pp. 896-898, September 1990).
73. Kerr, T. H., "Fallacies in Computational Testing of Matrix Positive Definiteness/Semidefiniteness," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-26, No. 2, pp. 415-421, March 1990.
74. Kerr, T. H., "Streamlining Measurement Iteration for EKF Target Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-27, No. 2, March 1991.
75. Kerr, T. H., "Emulating Random Process Target Statistics (using MSF)," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-30, No. 2, pp. 556-577, April 1994.
76. T.H. Kerr, "Use of GPS/INS in the Design of Airborne Multisensor Data Collection Missions (for tuning NN-based ATR algorithms)," *Proceedings of GPS-94*, pp. 1173-1188, Salt Lake City, UT, 21-23 Sept. 1994.
77. Kerr, T. H., "Assessing and Improving the Status of Existing Angle-Only Tracking (AOT) Results," *Proceedings of the International Conference on Signal Processing Applications & Technology (ICSPAT)*, Boston, MA, pp. 1574-1587, 24-26 Oct. 1995
78. Kerr, T. H., and Satz, H., S., "Applications of Some Explicit Formulas for the Matrix Exponential in Linear Systems Software Validation," *Proceedings of 16th Digital Avionics System Conference*, Vol. I, pp. 1.4-9 to 1.4-20, Irvine, CA, 26-30 Oct. 1997.
79. Kerr, T. H., "Critique of Some Neural Network Architectures and Claims for Control and Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-34, No. 2, pp. 406-419, April 1998.
80. Kerr, T. H., "Developing Cramer-Rao Lower Bounds to Gauge the Effectiveness of UEWR Target Tracking Filters," *Proceedings of AIAA/BMDO Technology Readiness Conference and Exhibit*, Colorado Springs, CO, 3-7 August 1998.
81. Kerr, T. H., "Perceived Rudimentary Vulnerabilities in Recent Adaptive Antenna processing (and STAP) to Statistically Non-Stationary Jammer Threats," *Proceedings of SPIE: Signal and Data Processing of Small Targets*, San Diego, CA, 29 July–3 August 2001.
82. Kerr, T. H., "New lamps for Old: A Shell Game for Generalized Likelihood Ratio use in Radar? Or this isn't you father's GLR!," *Proceedings of SPIE: Signal and Data Processing of Small Targets*, San Diego, CA, 29 July–3 August 2001.
83. Kerr, T. H., "Sensor Scheduling in Kalman Filters: Evaluating a Procedure for Varying Submarine Navaids," *Proceedings of Institute of Navigation 57th Annual Meeting*, Albuquerque, NM, 11-13 June 2001.
84. Kerr, T. H., "Further Critical Perspectives on Certain Aspects of GPS Development and Use," *Proceedings of Institute of Navigation 57th Annual Meeting*, Albuquerque, NM, 11-13 June 2001.
85. Satz, H. S., Kerr, T. H., *Comparison of Batch and Kalman Filtering for Radar Target Tracking* (Unclassified) AIAA/BMDO Conference as paper No. 9-2, Williamsburg, VA, 25 July 2001.
86. Grewal, M. S., and Andrews, A. P., *Kalman Filtering: Theory and Practice*, Prentice-Hall, Upper Saddle River, NJ, 1993.
87. Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, 1977.
88. Fitzgerald, R. J., "Effects of Range-Doppler Coupling on Chirp Radar Tracking Accuracy," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-10, No. 4, pp. 528-532, July 1974.
89. Press, W. H., Teukolsky, S. A., et al, *Numerical Recipes in Fortran 90: The Art of Parallel Scientific Computing*, Vol. 2, 2nd Edition, Cambridge University Press, NY, 1999.
90. Kerr, T. H., "Multichannel AR Modeling for the Active Decoy (U)," MIT Lincoln Laboratory Report No. PA-499, Lexington, MA, March 1987 (Secret).

91. Kerr, T. H., "Multichannel Shaping Filter Formulations for Vector Random Process Modeling Using Matrix Spectral Factorization," MIT Lincoln Laboratory Report No. PA-500, Lexington, MA, 27 March 1989 (BMO limited distribution)
92. Kerr, T. H., "A Critical Perspective on Some Aspects of GPS Development and Use," *Proceedings of 16th Digital Avionics System Conference*, Vol. II, pp. 9.4-9 to 9.4-20, Irvine, CA, 26-30 Oct. 1997.
93. Kerr, T. H., "Assessing and Improving the Status of Existing Angle-Only Tracking (AOT) Results," *Proceedings of the International Conf. on Signal Processing Applications &Technology (ICSPAT)*, Boston, MA, pp. 1574-1587, 24-26 Oct. 1995.
94. Kerr, T. H., "Rationale for Monte-Carlo Simulator Design to Support Multichannel Spectral Estimation and/or Kalman Filter Performance Testing and Software Validation & Verification Using Closed-Form Test Cases," MIT Lincoln Laboratory Report No. PA- 512, Lexington, MA, 22 December 1989 (BSD [previously BMO] limited distribution).
95. Rader, C. M., Steinhardt, A. O., "Hyperbolic Householder Transformation," *IEEE Trans. on Signal Processing*, Vol. 34, No. 6, pp. 1589-1502, Dec. 1986.
96. Danis, N. J., "Space-Based Tactical Ballistic Missile Launch Parameter Estimation," *IEEE Trans. On Aerospace and Electronic Systems*, Vol. AES-29, No. 2, April 1993.
97. Yeddanapudi, M., Bar-Shalom, Y., "Trajectory Prediction for Ballistic Missiles Based on Boost Phase LOS Measurements," *Proceedings of SPIE: Signal and Data Processing of Small Targets 1997*, O. E. Drummond (Ed.), Vol. 3163, pp. 316-328, San Diego, CA, 29- 31 July 1997.
98. Gura, I. A., "Extension of Linear Estimation Techniques to Nonlinear Problems," *The Journal of Astronautical Sciences*, Vol. XV, No. 4, pp. 194-205, July/August 1968.
99. Brown, A., and Bowles, W. M., "Measurement Updating Using the U-D Factorization Method When the Measurement Noise is Correlated," *Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON)*, Dayton, OH, pp. 344-348, 17-19 May 1983.
100. Lupash, L. O., "Comments on 'Efficient Time Propagation of U-D Covariance Factors'," *IEEE Trans. on Automatic Control*, Vol. AC-28, No. 11, pp. 1061-062, June 1983.
101. Lupash, L. O., "Case of Updating the Factorized Covariance Matrix," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 1, pp. 221-222, Jan.-Feb. 1994.
102. Upadhyay, T. N., and Damonlakis, J. N., "Sequential Piecewise Recursive Filter for GPS Low-Dynamics Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-16, No. 4, pp. 481-491, July 1980.
103. Carlson, N. A., "Fast Triangular Formulation of the Squareroot Filter," *AIAA Journal*, Vol. 11, No. 9, pp. 1259-1265, 1973.
104. Boncelet, C. G., Dickinson, B. W., "An Extension to the Kalman Filter," *IEEE Trans. on Automatic Control*, Vol. 32, No. 2, pp. 176-179, Feb. 1987.
105. Oshman, Y., "Gain-free Information Filtering using the Spectral Decomposition," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 12, pp. 681-690, 1989.
106. Park, P., Kailath, T., "New Square-Root Algorithms for Kalman Filtering," *IEEE Trans. on Automatic Control*, Vol. 40, No. 5, pp. 895-900, May 1995.
107. Campbell, L. A., TRACE Trajectory Analysis and Orbit Determination Program, Vol. XIII: Square Root Information Filtering and Smoothing, USAF Space Division Report No. SSD-TR-91-07, AD-A234957, 15 March 1991.
108. Itzkowitz, H. R., Baheti, R. S., "Demonstration of Square Root Kalman Filter on WARP Parallel Computer," *Proceedings of American Control Conference*, Pittsburgh, PA, June 1989.
109. Kay, S. M.., *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall Signal Processing Series, Alan V. Oppenheim, series editor, Simon and Schuster Co., Upper Saddle River, NJ, 1993.
110. Kelly, R. J.., "Comparison of LAAS B-Values with Linear Model Optimum B-Values, *Proceedings of Navigation: The Journal of the Institute of Navigation*, Vol. 47, No. 2, Summer 2000.
111. Kerr, T. H., "A New Multivariate Cramer-Rao Inequality for Parameter Estimation (Application: Input Probing Specification)," *Proceedings of IEEE Conference on Decision and Control*, Phoenix, AZ, pp. 97-103 Dec. 1994.
112. Fang, B. T., "A Nonlinear Counterexample for Batch and Extended Sequential Estimation Algorithms," *IEEE Transactions on Automatic Control*, Vol. 21, No. 1, pp. 138-139, Feb. 1976.
113. Fang, B. T., "Iterated batch least-squares and extended sequential estimation: convergence to different estimates," *IEEE TAC*, pp. 138-139, Feb. 1976.
114. Fang, B. T., "Linear weighted least-squares estimator: proof of minimum variance property," *IEEE TAC*, pp. 765-766, Dec. 1969.
115. Vetter, W. J., "Linear estimation with a priori information: minimum variance and least-squares criteria," *IEEE TAC*, pp. 265-266, June 1971.
116. Kerr, T. H., "Considerations in whether to use Marquardt Nonlinear Least Squares vs. Lambert Algorithm for NMD Cue Track Initiation (TI) calculations," TeK Associates Report, Lexington, MA, (for Raytheon, Sudbury, MA), 27 Sept. 2000.