James–Stein State Filtering Algorithms

Jonathan H. Manton, Vikram Krishnamurthy, and H. Vincent Poor, *Fellow, IEEE*

***Abstract*— In 1961, James and Stein discovered a remarkable estimator that dominates the maximum-likelihood estimate of the mean of a p-variate normal distribution, provided the dimension p is greater than two. This paper extends the James–Stein estimator and highlights benefits of applying these extensions to adaptive signal processing problems. The main contribution of this paper is the derivation of the James–Stein state filter (JSSF), which is a robust version of the Kalman filter. The JSSF is designed for situations where the parameters of the state-space evolution model are not known with any certainty. In deriving the JSSF, we derive several other results. We first derive a James–Stein estimator for estimating the regression parameter in a linear regression. A recursive implementation, which we call the James–Stein recursive least squares (JS-RLS) algorithm, is derived. The resulting estimate, although biased, has a smaller mean-square error than the traditional RLS algorithm. Finally, several heuristic algorithms are presented, including a James–Stein version of the Yule–Walker equations for AR parameter estimation.**

***Index Terms*— James–Stein estimation, Kalman filter, maximum-likelihood estimation, minimax estimation, recursive least squares, robust filtering, Yule–Walker equations.**

I. Introduction

CONSIDER the problem of estimating the mean of a p-dimensional random vector $\mathbf{X}$ having a multivariate normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^p$ and identity $p \times p$ covariance matrix, i.e., $\mathbf{X} \sim N(\boldsymbol{\mu}, I)$. Given the single realization $\mathbf{X}$, it is easily shown that the maximum likelihood estimate (MLE) of the mean is $\hat{\boldsymbol{\mu}} = \mathbf{X}$, and indeed, this is identical to the least squares estimate. Furthermore, it is readily shown that the risk of this MLE, i.e., the expected square error $J^{\mathrm{ML}}(\boldsymbol{\mu}) = \mathbf{E}[||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||^2]$, is p. Here, $||\cdot||$ denotes Euclidean length.

In 1961, James and Stein [11] proved the following remarkable result:[1] If the dimension p of $\mathbf{X}$ is greater than two, then the "James–Stein" estimator for $\boldsymbol{\mu}$

$$\hat{\boldsymbol{\mu}}^{\mathrm{JS}} = \left(1 - \frac{p-2}{||\mathbf{X}||^2}\right)\mathbf{X} \tag{1}$$

has a smaller risk (mean square error) $J^{\mathrm{JS}}(\boldsymbol{\mu}) = \mathbf{E}[||\hat{\boldsymbol{\mu}}^{\mathrm{JS}} - \boldsymbol{\mu}||^2]$ than the MLE *for all values* of $\boldsymbol{\mu}$ [i.e., $\forall\, \boldsymbol{\mu}$, $J^{\mathrm{JS}}(\boldsymbol{\mu}) < J^{\mathrm{ML}}(\boldsymbol{\mu})$]. It is important to note that the James–Stein estimator is a biased estimator, i.e., $\mathbf{E}[\hat{\boldsymbol{\mu}}^{\mathrm{JS}}] \neq \boldsymbol{\mu}$.

The James–Stein result has been considered by some to be paradoxical (see [6] for a popular article on this paradox). After all, for $p = 1, 2$, the MLE $\hat{\boldsymbol{\mu}}$ is admissible (that is, it cannot be beaten everywhere in the parameter space), and until the publication of [11], it was thought that the MLE $\hat{\boldsymbol{\mu}}$ was admissible for $p \geq 3$.

During the last 20 years, numerous papers have appeared in the statistical and econometrics literature that study applications and extensions of the James–Stein estimator [2], [5], [7], [9], [13], [24], [28]. Indeed, the James–Stein estimator is merely a special case of a "shrinkage estimator" [14]. Roughly speaking, this means that the factor $(p-2)/||\mathbf{X}||^2$ shrinks the MLE $\mathbf{X}$ to some centralized mean. Several shrinkage estimators have been studied in great detail in the mathematical statistics literature during the past 15 years.

Rather surprisingly, we have not come across any papers in statistical signal processing that consider the James–Stein estimator. In this paper, we extend the James–Stein estimator in several ways and consider some of its applications in statistical signal processing.

As detailed in [14, Sec. 4.6], the James–Stein estimator has a strong Bayesian motivation. A natural question that can be posed in the statistical signal processing context is: Does there exist a James–Stein version of the Kalman filter? *The main contribution of this paper is to derive the James–Stein state filter*. The James–Stein state filter, unlike the Kalman filter, provides sensible state estimates, regardless of how inaccurate the state-space evolution model is.

In deriving the James–Stein state filter, the contributions of this paper are threefold. First, we extend the James–Stein estimator to more general regression models. Then, we derive a James–Stein version of recursive least squares. These results lead onto our main result, which is the James–Stein state filter.

We now briefly describe these three contributions:

1) *Spherically Symmetric James–Stein Estimator:* The (spherically symmetric) James–Stein estimator for linear regression is introduced in Section II. The regression parameter estimate is proved to have a mean-square error no greater than that of the traditional (maximum-likelihood) estimate. Furthermore, the mean-square error

Manuscript received January 27, 1997; revised November 13, 1997. This work was supported by the Australian Telecommunication and Engineering Research Board, the Cooperative Research Centre for Sensor Signal and Information Processing, an Australian Research Council large grant, the Cooperative Research Centre for Sensor Signal and Information Processing, and the U.S. Office of Naval Research under Grant N00014-G4-1-0115. The associate editor coordinating the review of this paper and approving it for publication was Prof. Chi Chung Ko.

J. H. Manton and V. Krishnamurthy are with the Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Victoria, Australia (e-mail: jon@ee.mu.oz.au; vikram@ee.mu.oz.aa).

H. V. Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544-5263 USA (e-mail: poor@princeton.edu).

Publisher Item Identifier S 1053-587X(98)05957-1.

[1] In a recent published foreword [3] to James and Stein's paper, B. Efron states this result to be "the most striking theorem of post-war mathematical statistics."

can be further decreased if an *a priori* estimate of the true regression parameter exists.

2) *James–Stein Recursive Least Squares Algorithm:* While James–Stein estimation has been widely applied to linear regression in several fields (e.g., economics [7]), its absence from statistical signal processing may explain why recursive versions of the James–Stein estimator have not been developed before. In Section III, we develop a recursive implementation of the James–Stein estimator applied to least squares, which we call the James–Stein recursive least squares algorithm (JS-RLS). The JS-RLS algorithm yields parameter estimates of autoregressive with exogenous input (ARX) models. In particular, for an exogenous input model, the JS-RLS parameter estimates are guaranteed to have a mean-square error not exceeding that of the RLS. One application is the identification of the (finite) impulse response of a linear time-invariant system given both the input and the output signals.
3) *James–Stein State Filter:* The main result of this paper is the James–Stein state filter (JSSF), which is developed in Section IV. The signal model considered is a linear Gaussian state-space model—just as for the standard Kalman filter. (The JSSF is readily extended to nonlinear and non-Gaussian dynamics.) It is important to note that unlike the JS-RLS, which follows straightforwardly by shrinkage of the standard RLS, the JSSF cannot be derived as a straightforward application of shrinkage to the Kalman filter. (In fact, the differences between the Kalman filter and the JSSF make it misleading to call the JSSF the "James–Stein Kalman filter.")

   The JSSF derived in Section IV-C makes no assumptions concerning the accuracy of the state-space model. It is extremely robust in the sense that the mean-square error of the state estimate is guaranteed to be no larger than that of the MLE based on the observation model alone, regardless of how incorrect the state-space model is. (By comparison, even small perturbation errors in the state-space model can lead to the standard Kalman filter's mean-square error being much larger than that obtained if the observation model alone was used.) At the same time, the more accurate the state-space model is, the smaller the mean-square error will be.

   The JSSF has numerous potential applications. For example, in analysis of real-world data (economic, meteorological, etc.), the true system dynamics are often not known. Any approximation may be used for the system dynamics in the JSSF without fear of introducing a larger error than if no system dynamics were specified. In other words, the data are allowed to "speak for themselves."

   In Section IV-D, the James–Stein Kalman Filter with Hypothesis test (JSKF_{H}) is derived. This filter has the effect of implementing both the Kalman filter (KF) and the JSSF in parallel. At each time instant, a hypothesis test is used to determine if the system dynamics agree with the observations. If the system dynamics are in agreement with the observations, the KF state estimate is used. Otherwise, the JSSF state estimate is used.

   The JSKF_{H} has potential applications wherever the system dynamics are accurately known most of the time but, due to unexpected events, are inaccurate at certain instants in time. For example, the system dynamics of a target can be assumed to be those of straight-line motion. Although the target continues to move in a straight line, the KF state estimate is used. If the target suddenly maneuvers, the hypothesis test will detect this and use the JSSF state estimate instead.

   Both the JSSF and JSKF_{H} have a computational complexity of the same order of magnitude as the Kalman filter.

Applications: The algorithms derived in this paper can be applied to a wide range of problems. For example, the James–Stein versions of the Kalman filter (Section IV) can be applied to multidimensional imaging problems and multidimensional tracking problems [27] (see Section VI-A). In general, the JSSF and the JSKF_{H} can be applied directly to any system with more sensors than states.[2] The JSSF can also be used to filter observations (e.g., meteorological, econometrical) where the underlying model generating the data is not known with certainty. The James–Stein recursive least squares algorithm (Section III) can be used instead of the traditional RLS algorithm (see e.g., [10], [15], and [20] for typical applications of RLS). In problems such as estimating the (finite) impulse response of a linear time-invariant channel given both the input and the output signals (Section VI-B), the James–Stein RLS will give parameter estimates having a smaller mean-square error than the RLS algorithm's estimates. The James–Stein Yule–Walker algorithm (Section V-A) estimates the parameters of an autoregressive process and can therefore be used in such applications as linear predictive coding for speech processing [12].

Review of Other Robust Kalman Filters: Several robust Kalman filtering algorithms have been proposed in the adaptive signal processing and control literature. To put our JSSF in perspective, we give a brief overview of other robust Kalman filters. Many recent works (see [26] and references therein) assume the model parameters are subject to norm-bounded additive perturbations. Several discrete-time robust filters have been derived for such models. In particular, the infinite-horizon, time-invariant case is dealt with in [30], whereas the finite-horizon, time-varying case is considered in [26].

Other "robust Kalman filters" are robust against non-Gaussian noise (see [29] and references therein). Early approaches [21], [22] relied on the approximation of density functions. The early approaches were not without problems. In [18], an attractive approach was developed, based on score functions. Recently, a method for efficiently evaluating the necessary score functions has been given [29]. An alternative approach is to use a change of probability measure to transform the original noise into Gaussian noise [16], [17]. Finally, H^{∞} and risk-sensitive Kalman filters have been proposed in [8] and [23].

[2]More precisely, they can be applied directly to systems where the dimension of the observation vector is greater than or equal to the dimension of the state vector.

JSSF versus Other Robust Kalman Filters: In this paper, we use the term robustness in a "global" sense, stating that the JSSF is a globally robust state filter. More precisely, there is a lower bound on the JSSF's worst-case performance. Regardless of how inaccurate the state-space model is, the JSSF will always give sensible parameter estimates. This is very different from other robust Kalman filters in the literature, which are robust in a "local" sense, i.e., their performance is comparable with that of the Kalman filter even though there is some (small) error introduced into the model. On the one hand, the JSSF is expected to perform worse than a locally robust Kalman filter if the modeling errors are small. On the other hand, for sufficiently large modeling errors, the JSSF is expected to outperform any locally robust Kalman filter simply because the JSSF has a global upper bound on its mean-square error.

Limitations of James–Stein Estimators: Obviously, the James–Stein estimator is not a panacea to every estimation problem. To put our algorithms in perspective, it is important to stress their limitations:

1) The JSE (1) improves the overall risk and *not* the individual risk of each element of $\mathbf{X}$. This is important to note for two reasons. First, in certain applications, we may not be willing to trade a higher individual risk for a smaller overall risk. [Accurate location of an object in three dimensions is an example where a biased estimate that improves the overall risk (i.e., JSE) is preferable to the MLE.] Second, the fact that an individual element of $\mathbf{X}$ may have a larger risk than the MLE shows that the JSE, while being an extremely surprising result, stops short of being a paradox.
2) The JSE is a biased estimator; essentially, it trades bias for risk. Depending on the subsequent use of the estimate, this may or may not be a disadvantage. We note that in the derivation of the James–Stein state filter, the bias is used to our advantage.
3) The Kalman filter for state estimation of linear Gaussian systems is optimal (minimum mean-square error) if the model is accurately known. Therefore, in this case, the JSSF cannot have a smaller mean-square error. However, when one or more of the assumptions for the optimality of the KF do not hold (e.g., parameters accurately specified, linear dynamics), the JSSF can yield better state estimates (in terms of mean-square error) than the KF. The JSSF will be derived and discussed in detail in Section IV.
4) A key requirement in deriving the JSSF is that the state-space observation matrix has either the same number or more rows than columns. This precludes directly using JSSF for some applications. However, if the observation matrix has fewer rows than columns, the JSSF can be applied to an appropriately reduced state-space model (see Section IV-C). Note that models with observation matrices having more rows than columns occur frequently in multidimensional imaging systems [27]. In Section VI-A, we present an application of the JSSF to one such system.
5) As discussed above, the JSE (1) dominates the MLE. Is there an estimator that dominates the JSE? The answer is yes. In fact, an admissible estimator has been explicitly given in [25] (see [13] for an intuitive derivation). However, the correct amount of shrinkage requires numerical integration. It appears (see [19]) that the extra improvement in risk is small. Therefore, we will be content to use (1).

Some Definitions: To facilitate discussion of James–Stein estimators developed in this paper, the following definitions will be used in the sequel. These definitions are standard and can be found in [14].

The **risk** of an estimator is the mean-square error (MSE) of the estimator given the true parameter value, i.e., the risk of the estimator $\hat{\boldsymbol{\mu}}$ of the parameter $\boldsymbol{\mu}$ is $J(\boldsymbol{\mu}) = \mathbf{E}[\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2]$.[3] In the sequel, we use *risk* and *MSE* interchangeably.

An estimator is said to **dominate** another estimator if for every parameter value, the risk of the former is less than or equal to that of the latter, provided there exists at least one parameter value for which strict inequality holds.

A **minimax estimator** is an estimator with the property that its largest risk is no greater than that of any other estimator. Since the MLE of the mean of a multivariate normal distribution is minimax, any estimator that dominates the MLE must also be minimax. Conversely, since the risk of the MLE is independent of the true mean, the MLE cannot dominate any minimax estimator.

An **admissible estimator** is an estimator for which no other estimator exists that dominates it. James and Stein showed that the MLE of the mean of a multivariate normal distribution is not an admissible estimator if the dimension exceeds two.

A **James–Stein estimator (JSE)** is an estimator that provides an estimate of the mean of a multivariate normal distribution and dominates the classical maximum-likelihood estimate.

Notation: Throughout, we use the notation $(x)^+ = \max(0, x)$, $\|x\|^2 = x'x$ (i.e., Euclidean norm) and $'$ to denote vector transpose. A (multivariate) normal distribution will be denoted by the standard $N(\boldsymbol{\mu}, \Sigma)$, with the covariance matrix Σ assumed positive definite. The trace of a matrix Ω is written as $\mathrm{tr}\{\Omega\}$ and the maximum eigenvalue as $\lambda_{\max}\{\Omega\}$. If $\mathbf{x}$ is a vector, $\mathbf{x}_j$ [or $(\mathbf{x}_k)_j$ since the vector $\mathbf{x}_k$ itself has a subscript] will be used to denote the jth element of $\mathbf{x}$, unless specifically stated to the contrary.

We will use J with an appropriate superscript to denote the risk of an estimator. For convenience, we may choose to omit the parameter [i.e., write J rather than $J(\boldsymbol{\mu})$], and furthermore, the inequality $J^{\mathrm{JS}} < J^{\mathrm{ML}}$ is an abbreviation of $\forall \boldsymbol{\mu} \in \mathbb{R}^p$, $J^{\mathrm{JS}}(\boldsymbol{\mu}) < J^{\mathrm{ML}}(\boldsymbol{\mu})$.

II. James–Stein Estimation for Linear Regression

This section introduces in more detail the James–Stein estimator and how it can be applied to linear regression problems. Although most of the results are known (e.g.,

[3] Note that in a Bayesian setting, which is not considered in this paper, MSE refers to the weighted average of the risk, namely, $\mathbf{E}[J(\boldsymbol{\mu})]$, the weighting function being the *a priori* probability density of the parameter.

see [7]), this section provides the basic results and notation required for our James–Stein versions of the RLS algorithm and the Kalman filter. In particular, Theorem 1 below extends the results in [7] to more general regression models.

A. Preliminaries

We first review the result in [7, Sec. 7.2], which shows that the JSE defined by (1) is readily extended to deal with a nonidentity covariance matrix. Unfortunately, depending on the covariance matrix, there may not exist a JSE. We explain intuitively why such a limitation exists. This leads to two important concepts we will subsequently use, namely, the "effective dimension" and "shifting the origin."

Let $\mathbf{X} \in \mathbb{R}^p$ denote a normally distributed random vector with mean $\boldsymbol{\mu}$ and positive definite covariance matrix Ω, i.e., $\mathbf{X} \sim N(\boldsymbol{\mu}, \Omega)$.

Define $P = \Omega^{-(1/2)}$ (since Ω is positive-definite, P exists). Because $P\mathbf{X} \sim N(P\boldsymbol{\mu}, I)$, (1) can be applied to $P\mathbf{X}$ to give

$$\hat{\boldsymbol{\mu}}^{\mathrm{JS}} = \left(1 - \frac{p-2}{\mathbf{X}'\Omega^{-1}\mathbf{X}}\right)\mathbf{X}. \tag{2}$$

This estimator is said to belong to the class of **spherically symmetric** estimators [2] since it is of the form $\hat{\boldsymbol{\mu}} = h(\mathbf{X}'\Omega^{-1}\mathbf{X})\mathbf{X}$, where h: $\mathbb{R} \to \mathbb{R}$ is any (Borel-measurable) function.

Bock [2] has shown that if $\mathrm{tr}\{\Omega\}/\lambda_{\max}\{\Omega\} \leq 2$, then no spherically symmetric estimator exists that dominates the MLE. For convenience, we will call $\mathrm{tr}\{\Omega\}/\lambda_{\max}\{\Omega\}$ the **effective dimension**. Note that the effective dimension is a real-valued quantity. A justification for the name effective dimension is given below.

To understand why such a restriction on Ω exists and to justify naming $\mathrm{tr}\{\Omega\}/\lambda_{\max}\{\Omega\}$ the effective dimension, it is necessary to view the problem from a different angle. Consider a diagonal covariance matrix, $\Omega = \mathrm{diag}\{\lambda_1, \cdots, \lambda_p\}$. The squared error may be written as $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2 = \sum_{j=1}^{p} \lambda_j((\widehat{P\boldsymbol{\mu}})_j - (P\boldsymbol{\mu})_j)^2$, where $P = \Omega^{-(1/2)}$. Since the JSE of $P\boldsymbol{\mu}$ in general does not have a smaller risk for every individual element, it is clear that if one of the λ_j is relatively large, it is no longer possible to compensate for the possibility of introducing a larger MSE into this particular element. In a sense, the "effective dimension" has been reduced since we are no longer able to safely shrink certain elements. Bock's theorem [2, Th. 2] then essentially states that the MLE is inadmissible if the effective dimension is greater than two.

Another important technique we will use subsequently is **shifting the origin**. It is well known [3] that the risk of the JSE (1) decreases as $\|\boldsymbol{\mu}\|^2 \to 0$. If it is known that $\boldsymbol{\mu}$ is near $\overline{\boldsymbol{\mu}}$ (i.e., $\|\boldsymbol{\mu} - \overline{\boldsymbol{\mu}}\|^2 < \|\boldsymbol{\mu}\|^2$), the risk of the JSE (1) is decreased by shifting the origin to $\overline{\boldsymbol{\mu}}$, i.e., by replacing $\boldsymbol{\mu}$ by $(\tilde{\boldsymbol{\mu}} + \overline{\boldsymbol{\mu}})$ and estimating $\tilde{\boldsymbol{\mu}}$ instead.

B. Spherically Symmetric James–Stein Estimators for Linear Regression Models

Both the RLS algorithm and the Kalman filter involve a linear regression. This subsection states the James–Stein estimator for the regression parameter, which we will use in the following sections.

Consider the problem of estimating the vector $\mathbf{x} \in \mathbb{R}^p$ given the observation vector $\mathbf{z} \in \mathbb{R}^n$ $(n \geq p)$ generated by the model

$$\mathbf{z} = C\mathbf{x} + D\mathbf{w}, \qquad \mathbf{w} \sim N(0, \sigma^2 I) \tag{3}$$

with $C \in \mathbb{R}^{n \times p}$ and $D \in \mathbb{R}^{n \times n}$ known matrices of full (column) rank and $\mathbf{w} \in \mathbb{R}^n$ an i.i.d. Gaussian noise vector with variance $\sigma^2 \in \mathbb{R}$. If $n > p$, σ^2 need not be known since it is possible to estimate it from the data $\mathbf{z}$. If $n = p$, we assume that σ^2 is known.

For notational convenience, define

$$R = (DD')^{-1}. \tag{4}$$

The MLE of $\mathbf{x}$ is well known [15] to be

$$\hat{\mathbf{x}}^{\mathrm{ML}} = (C'RC)^{-1}C'Rz \tag{5}$$

and furthermore, the MLE is normally distributed about the true parameter, i.e.,

$$\hat{\mathbf{x}}^{\mathrm{ML}} \sim N(\mathbf{x}, \sigma^2(C'RC)^{-1}). \tag{6}$$

This shows that linear regression is equivalent to estimating the mean $\mathbf{x}$ of the multivariate normal distribution (6) based on the single realization $\hat{\mathbf{x}}^{\mathrm{ML}}$. We refer to the covariance matrix of $\hat{\mathbf{x}}^{\mathrm{ML}}$, namely, $(C'RC)^{-1}$, as simply **the covariance matrix**.

We now state the spherically symmetric JSE for the regression parameter. It is based on the JSE presented in [7, Ch. 7]. The differences are that we have included the matrix D in the regression model (3) and included the term $(\min\{(p-2), 2(p^*-2)\})^+$ in (7).

Theorem 1: If σ^2 in (3) is known and $n \geq p$, the James–Stein estimator for $\mathbf{x}$ in the regression (3) is

$$\hat{\mathbf{x}}^{\mathrm{JS}} = \left(1 - \sigma^2 \frac{(\min\{(p-2), 2(p^*-2)\})^+}{\hat{\mathbf{x}}^{\mathrm{ML}\prime}(C'RC)\hat{\mathbf{x}}^{\mathrm{ML}}}\right)^+ \hat{\mathbf{x}}^{\mathrm{ML}} \tag{7}$$

where R is defined in (4), p is the dimension of $\mathbf{x}$, n the dimension of $\mathbf{z}$, $\hat{\mathbf{x}}^{\mathrm{ML}}$ is defined in (5), and the effective dimension (which is defined in Section II-A) is

$$p^* = \frac{\mathrm{tr}\{(C'RC)^{-1}\}}{\lambda_{\max}\{(C'RC)^{-1}\}}. \tag{8}$$

If $n > p$ and σ^2 is unknown, σ^2 is replaced in (7) by

$$\sigma^2 = \frac{\|D^{-1}(\mathbf{z} - C\hat{\mathbf{x}}^{\mathrm{ML}})\|^2}{n - p + 2}. \tag{9}$$

Furthermore, for the regression (3), the James–Stein estimator $\hat{\mathbf{x}}^{\mathrm{JS}}$ [which is defined by (7)] and the MLE $\hat{\mathbf{x}}^{\mathrm{ML}}$ [which is defined by (5)] have the following properties:

1) For all $\mathbf{x} \in \mathbb{R}^p$, $J^{\mathrm{JS}}(\mathbf{x}) \leq J^{\mathrm{ML}}(\mathbf{x})$, where $J^{\mathrm{ML}} = \mathbf{E}[\|\hat{\mathbf{x}}^{\mathrm{ML}} - \mathbf{x}\|^2]$ and $J^{\mathrm{JS}} = \mathbf{E}[\|\hat{\mathbf{x}}^{\mathrm{JS}} - \mathbf{x}\|^2]$ are the MSE's (risks) of $\hat{\mathbf{x}}^{\mathrm{ML}}$ (5) and $\hat{\mathbf{x}}^{\mathrm{JS}}$ (7), respectively.
2) If $p^* > 2$, $J^{\mathrm{JS}} < J^{\mathrm{ML}}$, with an upper bound on $J^{\mathrm{JS}}(\mathbf{x})$ decreasing as $\mathbf{x}$ approaches the origin (more precisely, as $\|\Omega^{-(1/2)}\mathbf{x}\|^2 \to 0$).
3) If $p^* \leq 2$, $\hat{\mathbf{x}}^{\mathrm{JS}} = \hat{\mathbf{x}}^{\mathrm{ML}}$.

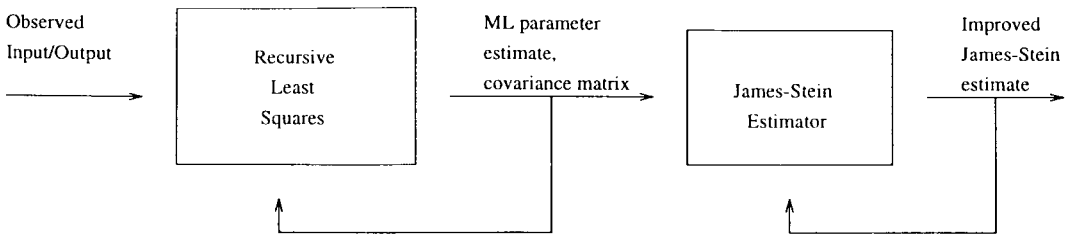

Fig. 1. James–Stein RLS algorithm.

Proof: Define $\boldsymbol{\mu} = \mathbf{x}$, $\Omega = \sigma^2(C'RC)^{-1}$, and $\mathbf{X} = \hat{\mathbf{x}}^{\mathrm{ML}}$. It is clear that $\mathbf{X} \sim N(\boldsymbol{\mu}, \Omega)$ with Ω positive-definite.

In [7, Sec. 7], it has been shown that the JSE

$$\hat{\boldsymbol{\mu}}^{\mathrm{JS}} = \left(1 - \frac{c}{\mathbf{X}'\Omega^{-1}\mathbf{X}}\right)^{+}\mathbf{X} \tag{10}$$

dominates the MLE for any positive constant c, provided $0 < c \leq 2(p^* - 2)$, where the effective dimension $p^* = \mathrm{tr}\{\Omega\}/\lambda_{\max}\{\Omega\}$.

Our estimator (7) is equivalent to (10) with

$$c = (\min\{(p-2), 2(p^*-2)\})^{+}. \tag{11}$$

Clearly, c satisfies the constraint $0 < c \leq 2(p^* - 2)$. Furthermore, the results remain valid when σ^2 is replaced by (9) [7].

Last, if $\Omega = I$, the risk of $\hat{\mathbf{x}}^{\mathrm{JS}}$ is a concave function of $||\mathbf{x}||^2$ (see [3]). For arbitrary (positive-definite) Ω, introduce the transform $P = \Omega^{-(1/2)}$ and note that

$$||P(\hat{\mathbf{x}}^{\mathrm{JS}} - \mathbf{x})||^2 = (\hat{\mathbf{x}}^{\mathrm{JS}} - \mathbf{x})'\Omega^{-1}(\hat{\mathbf{x}}^{\mathrm{JS}} - \mathbf{x}) \tag{12}$$

$$\geq \lambda_{\max}\{\Omega\}^{-1}||\hat{\mathbf{x}}^{\mathrm{JS}} - \mathbf{x}||^2. \tag{13}$$

Therefore, $\lambda_{\max}\{\Omega\}\mathbf{E}[||P(\hat{\mathbf{x}}^{\mathrm{JS}} - \mathbf{x})||^2]$ is a concave function of $||P\mathbf{x}||^2$ and is an upper bound of the risk $J^{\mathrm{JS}}(\mathbf{x}) = \mathbf{E}[||\hat{\mathbf{x}}^{\mathrm{JS}} - \mathbf{x}||^2]$. □

Remarks:

1) In [2] and [7], the JSE (10) has been derived in terms of the constant c. However, no specific equation for c is given. Our choice of (11), being somewhat tangential to the rest of this paper, is justified in Appendix A.
2) The properties of the JSE (7) given in Theorem 1 fail to hold if, in (3), C and/or D depend on $\mathbf{w}$ (or $\mathbf{z}$). The reason is because (6) will, in general, not hold if C and/or D are not (statistically) independent of $\mathbf{w}$.
3) It is important to note that there is no loss of generality in assuming that for the regression model (3) $(C'RC)^{-1}$ is diagonal. This can be explained as follows: Let P denote an invertible square matrix. Then, (3) is equivalent to $\mathbf{z} = (CP^{-1})(P\mathbf{x}) + D\mathbf{w}$. The conditional covariance matrix of $P\hat{\mathbf{x}}^{\mathrm{ML}}$ given $P\mathbf{x}$ becomes $P(C'RC)^{-1}P'$ and can be made diagonal by choosing P to be a suitable orthonormal $(P'P = I)$ square matrix. Most importantly, the choice of an *orthonormal* P ensures that the MSE of any estimator $\hat{\mathbf{x}}$ and of $\hat{\hat{\mathbf{x}}} = P^{-1}\hat{\mathbf{x}}$ are identical.

III. James–Stein Recursive Least Squares (JS-RLS)

In this section, we derive a James–Stein version of the recursive least squares (RLS) algorithm. We call the recursive algorithm "James–Stein recursive least squares" (JS-RLS). The schematic structure of the JS-RLS is shown in Fig. 1.

Because the JS-RLS is merely a recursive algorithm for implementing the JSE (7), the JS-RLS has identical properties to the JSE (7) (see Theorem 1). Theorem 1 is valid, provided (6) holds or, in other words provided in (3), the matrix C is independent of the observation vector z. Under this assumption, the JS-RLS will yield smaller MSE regression parameter estimates compared with the RLS.

Several heuristic modifications are made to the JS-RLS later in this paper (Section V).

A. The Model

The standard recursive least squares (RLS) algorithm (see [20]) is used to recursively estimate the parameter vector $\mathbf{x}_k = [a_1, \cdots, a_r, b_1, \cdots, b_q]' \in \mathbb{R}^{q+r}$ in the ARX model

$$z(k) = \sum_{t=1}^{r} a_t z(k-t) + \sum_{t=1}^{q} b_t u(k-t) + w(k) \tag{14}$$

where

$u(k) \in \mathbb{R}$	(known) exogenous input;
$z(k) \in \mathbb{R}$	observed output;
$w(k) \sim N(0, \sigma^2)$	additive white noise.

The subscript k denotes that the estimate of $\mathbf{x}_k$ is based on the k observations $\{z(1), z(2), \cdots, z(k)\}$. The noise variance σ^2 is assumed to be unknown. We define $s = \max(q, r)$ and denote the dimension of $\mathbf{x}_k$ by p $(p = q + r)$.

Remark: The application of James–Stein estimation to linear regression requires (6) to hold. If an AR model is present (i.e., $r \geq 1$) in (14), in general, (6) holds only asymptotically.

We write the estimation problem in matrix form to show its equivalence to the linear regression (3). At time instant k, we seek to estimate $\mathbf{x}_k$, given the observations $\{z(1), z(2), \cdots, z(k)\}$ and the regression relation

$$\mathbf{z}_k = C_k\mathbf{x}_k + \Omega_k^{-(1/2)}\mathbf{w}_k \tag{15}$$

where the mth element of $\mathbf{z}_k$ is $z(m+s)$, and the mth row of C_k is $[z(m+s-1), \cdots, z(m+s-r), u(m+s-1), \cdots, u(m+s-q)]$. $\Omega_k = \mathrm{diag}\{\lambda^{k-s-1}, \lambda^{k-s-2}, \cdots, 1\}$, where λ denotes the exponential forgetting factor (see [20]).

For convenience, we now state the standard RLS algorithm (e.g., see [15]):

Algorithm 1—Standard RLS: The standard RLS algorithm is

$$\begin{aligned}
&\mathbf{u}_k = [z(k-1), \cdots, z(k-r), u(k-1), \cdots, u(k-q)]' \\
&\mathbf{k}_k = P_k\mathbf{u}_k = \frac{P_{k-1}\mathbf{u}_k}{\lambda + \mathbf{u}_k' P_{k-1}\mathbf{u}_k} \\
&\hat{\mathbf{x}}_k = P_k\mathbf{d}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{k}_k(z(k) - \hat{\mathbf{x}}_{k-1}'\mathbf{u}_k) \\
&P_k = (C_k'\Omega_k C_k)^{-1} = \lambda^{-1}(P_{k-1} - \mathbf{k}_k\mathbf{u}_k' P_{k-1})
\end{aligned} \tag{16}$$

where λ denotes the forgetting factor $0 < \lambda \leq 1$. (Initialization can be performed according to the initialization procedure given in Algorithm 2 below.)

B. The James–Stein Recursive Least Squares Algorithm

We state the JS-RLS algorithm below and then devote the remainder of this section to an explanation of its derivation.

Algorithm 2—JS-RLS:

Initialization: Set $k = p$

$$P_k = (C_k'\Omega_k C_k)^{-1} \quad \hat{\mathbf{x}}_k = P_k\mathbf{d}_k \quad \mathbf{d}_k = C_k'\Omega_k\mathbf{z}_k \tag{17}$$

$$s_k = \mathbf{z}_k'\Omega_k\mathbf{z}_k \quad Q_k = (C_k'\Omega_k C_k) \quad k_k^{\text{eff}} = k \tag{18}$$

$$\overline{\mathbf{x}}_k = \begin{cases} \textit{a priori} \text{ estimate of } \mathbf{x}_k & \text{if available} \\ [0, \cdots, 0]' & \text{otherwise.} \end{cases} \tag{19}$$

$C_k \in \mathbb{R}^{(k-s)\times p}$ is a matrix with mth row $[z(m+s-1), \cdots, z(m+s-r), u(m+s-1), \cdots, u(m+s-q)]$. $\mathbf{z}_k \in \mathbb{R}^{(k-s)}$ is a vector with mth element $z(m+s)$. $\Omega_k \in \mathbb{R}^{(k-s)\times(k-s)}$ is the diagonal matrix $\text{diag}\{\lambda^{k-s-1}, \lambda^{k-s-2}, \cdots, 1\}$ with λ denoting the forgetting factor; $0 < \lambda \leq 1$.

Update Equations:

$$\mathbf{u}_k = [z(k-1), \cdots, z(k-r), u(k-1), \cdots, u(k-q)]' \tag{20}$$

$$\mathbf{k}_k = \frac{P_{k-1}\mathbf{u}_k}{\lambda + \mathbf{u}_k' P_{k-1}\mathbf{u}_k} \tag{21}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{k}_k(z(k) - \hat{\mathbf{x}}_{k-1}'\mathbf{u}_k) \tag{22}$$

$$P_k = \lambda^{-1}(P_{k-1} - \mathbf{k}_k\mathbf{u}_k' P_{k-1}) \tag{23}$$

$$\mathbf{d}_k = \lambda\mathbf{d}_{k-1} + z(k)\mathbf{u}_k \quad s_k = \lambda s_{k-1} + (z(k))^2 \tag{24}$$

$$Q_k = \lambda Q_{k-1} + \mathbf{u}_k\mathbf{u}_k' \quad k_k^{\text{eff}} = \lambda k_{k-1}^{\text{eff}} + 1. \tag{25}$$

Moreover, the JS-RLS estimate $\hat{\mathbf{x}}_k^{\text{JS}}$ is computed as

$$v = (\hat{\mathbf{x}}_k - \overline{\mathbf{x}}_{k-1})' Q_k (\hat{\mathbf{x}}_k - \overline{\mathbf{x}}_{k-1}) \quad p^* = \frac{\text{tr}\{P_k\}}{\lambda_{\max}\{P_k\}} \tag{26}$$

$$\hat{\mathbf{x}}_k^{\text{JS}} = \left(1 - \frac{s_k - 2(\hat{\mathbf{x}}_k - \overline{\mathbf{x}}_{k-1})'\mathbf{d}_k + v}{k_k^{\text{eff}} - p + 2} \cdot \frac{(\min\{(p-2), 2(p^*-2)\})^+}{v}\right)^+ \hat{\mathbf{x}}_k. \tag{27}$$

Risk: Define $J_k^{\text{RLS}} = \mathbf{E}[\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|^2]$ and $J_k^{\text{JS-RLS}} = \mathbf{E}[\|\hat{\mathbf{x}}_k^{\text{JS}} - \mathbf{x}_k\|^2]$ as the risks of the MLE $\hat{\mathbf{x}}_k$ (16) and the JSE $\hat{\mathbf{x}}_k^{\text{JS}}$ (27), respectively.

Remarks:

1) *Computational Complexity:* Excluding the computational cost of $\lambda_{\max}\{P_k\}$ required in (26), the JS-RLS and the RLS have the same order of computational cost $O(p^2)$ and the same order of memory requirements $O(p^2)$.

   Computing $\lambda_{\max}\{P_k\}$, the largest eigenvalue of a Toeplitz matrix also has a computational complexity $O(p^2)$. If computing $\lambda_{\max}\{P_k\}$ in (26) is undesirable, it may be avoided by

   - replacing $\lambda_{\max}\{P_k\}$ by any upper bound. The only effect is to reduce the difference in MSE between $\hat{\mathbf{x}}_j^{\text{JS}}$ and $\hat{\mathbf{x}}_k$. Note that $\hat{\mathbf{x}}_k^{\text{JS}}$ will still dominate $\hat{\mathbf{x}}_k$.
   - replacing $\lambda_{\max}\{P_k\}$ by its asymptotic value. Care must be taken since inaccurate (i.e., large MSE) estimates may result if $\lambda_{\max}\{P_k\}$ is larger than its asymptotic value.

2) If $r = 0$ in (14), i.e., no AR component is present, and a unity forgetting factor is used (i.e., $\lambda = 1$), the JS-RLS is guaranteed to have an MSE not exceeding that of the RLS, i.e., $J_k^{\text{JS-RLS}} \leq J_k^{\text{RLS}}$. [*Proof:* With initialization as given in Algorithm 2, the RLS (Algorithm 1) estimates are the maximum likelihood estimates of the linear regression (3), and the JS-RLS (Algorithm 2) estimates are the James–Stein estimates of Theorem 1.]
3) If $p^* \leq 2$, $\hat{\mathbf{x}}_k^{\text{JS}} = \hat{\mathbf{x}}_k$ (i.e., JS-RLS becomes ordinary RLS). A necessary condition for $p^* > 2$ is $p > 2$ since from (7), it follows that $p^* \leq p$.
4) A discussion of the JS-RLS applied to AR models (i.e., $r > 0$), along with several heuristic modifications, are presented in Section V.

Derivation of JS-RLS: The JS-RLS algorithm is derived in two steps. Initially, the case when $\lambda = 1$ is examined. The extension to $\lambda < 1$ is then given.

We assume that the data $\mathbf{z}_k$ have been generated by $\mathbf{z}_k = C_k\mathbf{x}_k + \mathbf{w}_k$.

1) *Unity Forgetting Factor:* If $\lambda = 1$, the standard RLS algorithm recursively calculates $\hat{\mathbf{x}}_k = (C_k'C_k)^{-1}C_k'\mathbf{z}_k$, which is the MLE of the linear regression $\mathbf{z}_k = C_k\mathbf{x}_k + \mathbf{w}_k$. Therefore, the JSE (7) may be used to improve on $\hat{\mathbf{x}}_k$.

   In this case, for any k (and $r = 0$), Theorem 1 proves that $J_k^{\text{JS-RLS}} \leq J_k^{\text{RLS}}$.
2) *General Forgetting Factor:* If $\lambda < 1$, the standard RLS recursively calculates $\hat{\mathbf{x}}_k = (C_k'\Omega_k C_k)^{-1}C_k'\Omega_k\mathbf{z}_k$, which is the MLE of the linear regression $\mathbf{z}_k = C_k\mathbf{x}_k + \Omega_k^{-(1/2)}\mathbf{w}_k$. However, the data was generated by $\mathbf{z}_k = C_k\mathbf{x}_k + \mathbf{w}_k$.

   There are two effects of the mismatched model. First, the estimation of the variance of $\mathbf{w}_k$ is no longer given by (9). Second, the covariance of $\hat{\mathbf{x}}_k$ is no longer $\sigma^2(C_k'\Omega_k C_k)^{-1}$, but in fact, $\sigma^2(C_k'\Omega_k C_k)^{-1}C_k'\Omega_k^2 C_k(C_k'\Omega_k C_k)^{-1}$. (This "mismatch" in variance prevents $\hat{\mathbf{x}}_k^{\text{JS}}$ from dominating $\hat{\mathbf{x}}_k$ if $\lambda < 1$.)

   The first effect is easily remedied; the second is ignored.[4] Rather than estimate σ^2 by (9), namely

$$\sigma^2 = \|\Omega_k^{1/2}(\mathbf{z}_k - C_k\mathbf{x}_k)\|/(k-p+2) \tag{28}$$

   we replace k in (28) by k_k^{eff}, where $k_k^{\text{eff}} = \lambda k_{k-1}^{\text{eff}} + 1$.

[4] After all, the inclusion of λ into RLS in the first place is heuristic.

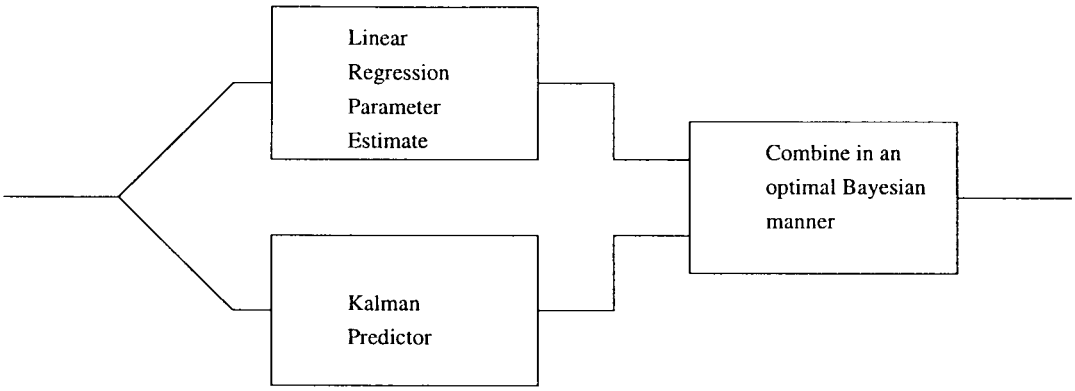

Fig. 2. Standard Kalman filter (feedback not shown).

The JS-RLS algorithm (Algorithm 2) can now be derived in the same way that the standard RLS algorithm (Algorithm 1) is derived; see, for example, [10]. In fact, (20)–(23) are identical to (16). Furthermore, (20)–(25) merely calculate (17) and (18) recursively (with the exception of k_k^{eff}). From the identities (17) and (18), it can then be verified that (27) is the JSE (7).

IV. James–Stein Versions of the Kalman Filter

This section derives two James–Stein versions of the Kalman filter (KF). The first version is the James–Stein state filter (JSSF), which was derived in Section IV-C. The JSSF places no constraints on the state-space model, i.e., the state-space model may be incorrectly specified, it may be nonlinear, it need not be Gaussian, etc. The observation model is a linear regression with Gaussian noise. The JSSF will always have a MSE less than the MSE obtainable from the maximum likelihood estimate of the state given only the observation model. The JSSF is then combined with the ordinary Kalman filter (KF) to give the James–Stein Kalman filter with hypothesis test (JSKF_{H}) algorithm. The JSKF_{H} derived in Section IV-D incorporates a hypothesis test to determine if the state-space model is correct.

A. Gaussian State Space Signal Model and Standard Kalman Filter (KF)

In this section, we describe our Gaussian state-space signal model and summarize the standard Kalman filter.

The Kalman filter [1] is the minimum MSE (i.e., optimal) filter for the linear Gaussian state-space model

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{e}_{k+1} \tag{29}$$

$$\mathbf{z}_k = C_k \mathbf{x}_k + D_k \mathbf{w}_k \tag{30}$$

where $\mathbf{x}_k \in \mathbb{R}^p$ is the state vector and $\mathbf{z}_k \in \mathbb{R}^n$ the observation vector. $\mathbf{e}_k \in \mathbb{R}^r$ and $\mathbf{w}_k \in \mathbb{R}^n$ are random noise vectors ($\mathbf{e}_k \sim$ i.i.d. $N(0, Q_k)$, $\mathbf{w}_k \sim$ i.i.d. $N(0, \sigma^2 I)$). $A_k \in \mathbb{R}^{p\times p}$, $B_k \in \mathbb{R}^{p\times r}$, $C_k \in \mathbb{R}^{n\times p}$, and $D_k \in \mathbb{R}^{n\times n}$ are (deterministic) matrices.

Let $Z_k = \{\mathbf{z}_0, \cdots, \mathbf{z}_k\}$ denote the observations up to time k. The objective is to compute the filtered state estimate based on the observations Z_k, i.e., compute $\hat{\mathbf{x}}_{k|k} = \mathbf{E}[\mathbf{x}_k | Z_k]$. $\hat{\mathbf{x}}_{k+1|k}$ will similarly be used to denote $\mathbf{E}[\mathbf{x}_{k+1}|Z_k]$, which is the predicted state estimate. $\hat{\mathbf{x}}_k^{\text{ML}}$ is the MLE of $\mathbf{x}_k$ given $\mathbf{z}_k$ in (30), i.e.,

$$\hat{\mathbf{x}}_k^{\text{ML}} = (C_k'(D_k D_k')^{-1} C_k)^{-1} C_k'(D_k D_k')^{-1} \mathbf{z}_k. \tag{31}$$

The Kalman filter equations are [1]

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - C_k \hat{\mathbf{x}}_{k|k-1}) \tag{32}$$

$$K_k = P_{k|k-1} C_k' [C_k P_{k|k-1} C_k' + \sigma^2 D_k D_k']^{-1} \tag{33}$$

$$\hat{\mathbf{x}}_{k+1|k} = A_k \hat{\mathbf{x}}_{k|k} \tag{34}$$

$$P_{k+1|k} = A_k [I - K_k C_k] P_{k|k-1} A_k' + B_k Q_{k+1} B_k'. \tag{35}$$

K_k is the Kalman gain, and

$$P_{k|k-1} = \mathbf{E}[(x_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})' | Z_{k-1}] \tag{36}$$

the covariance of $\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}$.

The Kalman filter is shown in block diagram form in Fig. 2 (with feedback of $\hat{\mathbf{x}}_{k|k}$ and $P_{k|k-1}$ omitted for clarity). The Kalman predictor computes $\hat{\mathbf{x}}_{k|k-1}$ (34) along with its covariance $P_{k|k-1}$. The linear regression parameter estimate computes $\hat{\mathbf{x}}_k^{\text{ML}}$, which is the MLE of $\mathbf{x}_k$ given $\mathbf{z}_k$ in (30). Finally, $\hat{\mathbf{x}}_k^{\text{ML}}$ and $\hat{\mathbf{x}}_{k|k-1}$ are combined in a Bayesian manner [cf., (32)] to give $\hat{\mathbf{x}}_{k|k}$.

Risk: Define $J_k^{\text{KF}} = \mathbf{E}[\|\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k\|^2]$ and $J_k^{\text{ML}} = \mathbf{E}[\|\hat{\mathbf{x}}_k^{\text{ML}} - \mathbf{x}_k\|^2]$, which are the risks of the Kalman filter (32) and of the MLE $\hat{\mathbf{x}}_k^{\text{ML}}$ (31), respectively.

B. Outline of Approach

Confusion can arise by comparing the JSSF with the KF too closely. Therefore, this section sketches an alternative derivation of the JSSF given in the next section.

Consider a sequence $\{\mathbf{x}_0, \mathbf{x}_1, \cdots\}$ of (in general, dependent) random variables. Each random variable in the sequence is observed via the linear regression (30), namely, $\mathbf{z}_k = C_k \mathbf{x}_k + D_k \mathbf{w}_k$, where $\mathbf{w}_k \sim$ i.i.d. $N(0, \sigma^2 I)$.

If nothing is known about the probability space from which the sequence of random variables from which $\{\mathbf{x}_0, \mathbf{x}_1, \cdots\}$ was generated, only the single measurement $\mathbf{z}_k$ can be used to estimate $\mathbf{x}_k$.

For this estimation problem (a linear regression), $\mathbf{x}_k$ can be estimated by the MLE (31). It can also be estimated by the JSE (7) with the origin shifted (see Section II-A) to any point. Remember that regardless of which shifted JSE is used, the

MSE of the estimate will be less than the MSE of the MLE (31). In other words, any shifted JSE can be used without affecting the "worst-case" MSE of the estimate.

Assume now that we believe that $\mathbf{x}_k$ is approximately equal to some function of $\mathbf{x}_{k-1}$, say, $f(\mathbf{x}_{k-1})$. How can we incorporate this belief into an estimate of $\mathbf{x}_k$ while still requiring the worst-case MSE of the estimator to be no worse than the MLE (31)? The answer is to use the JSE (7) with the origin shifted to $f(\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1})$, where $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}$ is the estimate of $\mathbf{x}_{k-1}$.

It is seen then that the main difference between the JSSF and the KF is that the JSSF assumes that the state sequence $\{\mathbf{x}_0, \mathbf{x}_1, \cdots\}$ comes from a completely unknown probability space, whereas the KF assumes that $\{\mathbf{x}_0, \mathbf{x}_1, \cdots\}$ was generated by the linear state-space model (29).

C. James–Stein State Filter (JSSF)

The JSSF is based on the linear Gaussian state-space model (29) and (30) with the following differences in assumptions.

1) Require $n \geq p > 2$ so that the JSE (7) can be applied. (For the same reason, if σ^2 is unknown, we require $n > p$, that is, we require the observation matrix C_k to have equal or more rows than columns.)
2) Require C_k and D_k to be accurately known and have full (column) rank so that the appropriate inverses exist.
3) Require $\mathbf{e}_{k+1}$ to be independent of $\mathbf{w}_k$. Otherwise, it is conceivable that a certain combination of $\mathbf{e}_{k+1}$ and $\mathbf{w}_k$ can increase the MSE beyond that of the MLE based on (30) alone.
4) Other than independence of $\mathbf{e}_{k+1}$, no further assumptions are necessary for (29). $\mathbf{e}_{k+1}$ need not be Gaussian, and A_k, B_k need not be known. In fact, $\mathbf{x}_{k+1}$ can depend on $\mathbf{x}_k$ in a nonlinear fashion.

Remark: Assumption 1 (i.e., $n \geq p > 2$) and Assumption 2 ensure that $(C_k'(D_kD_k')^{-1}C_k)^{-1}$ in (40) and (41) exists.

Discussion of Assumption 1: Requiring the observation matrix C_k to have equal or more rows than columns may appear restrictive. However, there are several multidimensional output systems such as multidimensional tracking and imaging systems (see Section VI-A) that have $n \geq p > 2$. Moreover, Assumption 1 is *not* restrictive in the sense that any system with $n < p$ contains a $(p-n)$-dimensional subspace of the state-space $\mathbb{R}^p$, which is unobservable[5] if the state-space model is unknown, i.e., the observations $\mathbf{z}_k$ contain no information about these unobservable states, and therefore, no sensible estimator exists for certain (linear combinations of) states. The JSSF can, however, be applied to the remaining (observable) n-dimensional subspace of the state space.

In particular, if[6] $2 < n < p$, then by appropriate use of (the equivalent of) pseudo inverses, the estimates of certain linear combinations of states (i.e., observable states) can be obtained from the JSSF. These state estimates still dominate the MLE. We explain this procedure below:

We assume rank $C_k = n$ (cf., Assumption 2). Introduce the $p \times p$ (real) orthonormal matrix S_k (i.e., $S_kS_k' = S_k'S_k = I$) such that its first n rows span the row space of C_k. Therefore, $C_kS_k' = [\tilde{C}_k|0]$, where $\tilde{C}_k \in \mathbb{R}^{n\times n}$ is invertible. Let $\tilde{\mathbf{x}}_k$ denote the first n elements of $S_k\mathbf{x}_k$. Since $C_k = [\tilde{C}_k|0]S_k$, the observation equation (30) can be written as

$$\mathbf{z}_k = \tilde{C}_k\tilde{\mathbf{x}}_k + D_k\mathbf{w}_k. \tag{37}$$

Since $\tilde{C}_k$ is square, $\tilde{\mathbf{x}}_k$ can be estimated as in the JSSF. Finally, we can equate $\mathbf{x}_k$ in the state-space model (29) to $\mathbf{x}_k = S_k^{-1}[\tilde{\mathbf{x}}_k'|0]' = S_k'[\tilde{\mathbf{x}}_k'|0]'$, where we have simply set the unobservable entries (more precisely, the unobservable linear combinations) of $\mathbf{x}_k$ to zero.

Derivation of JSSF: In the Kalman filter, the state-space model (29) is used to compute $\hat{\mathbf{x}}_{k|k-1}$, which is the state prediction based on past observations. The JSSF also uses the state-space model for prediction. If $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k}$ is the current state estimate, $\hat{\mathbf{x}}^{\mathrm{JS}}_{k+1|k} = A_k\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k}$ is the prediction of $\mathbf{x}_{k+1}$ [cf., (34)].

Since we do not know how accurate the state-space model (29) is, $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}$ may be very inaccurate. Therefore, our strategy is to estimate the state vector $\mathbf{x}_k$ in (29) with the JSE (7) based on the regression (30), with the origin shifted to $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}$. We recall from Theorem 1 that regardless of how inaccurate $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}$ is, the resulting estimate of $\mathbf{x}_k$ will always have a MSE no greater than that of $\hat{\mathbf{x}}^{\mathrm{ML}}_k$ (31).

Since the JSE will have significantly smaller MSE if the true parameter $\mathbf{x}_k$ is near the origin, choosing $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}$ to be the origin has the effect of combining information contained in $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}$ and in $\hat{\mathbf{x}}^{\mathrm{ML}}_k$ together in a robust manner. The more accurate $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}$ is, the more accurate the resulting estimate of $\mathbf{x}_k$ will be, yet at the same time, our estimate of $\mathbf{x}_k$ is guaranteed to be no worse than $\hat{\mathbf{x}}^{\mathrm{ML}}_k$.

The block diagram of the JSSF is shown in Fig. 3 (with feedback of $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k}$ omitted for clarity). Although the KF combines $\hat{\mathbf{x}}^{\mathrm{ML}}_k$ and $\hat{\mathbf{x}}_{k|k-1}$ in a Bayesian manner (Fig. 2), the JSSF combines $\hat{\mathbf{x}}^{\mathrm{ML}}_k$ and $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}$ by shrinking $\hat{\mathbf{x}}^{\mathrm{ML}}_k$ toward $\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}$ (Fig. 3).

The James–Stein state filter algorithm is summarized below.

Algorithm 3. JSSF:

Initialization: $\hat{\mathbf{x}}^{\mathrm{JS}}_{1|0} = \mathbf{E}[\mathbf{x}_1]$

Recursive Filter:

$$\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k} = \hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1} + \left(1 - \sigma^2\,\frac{(\min\{(p-2),\,2(p^*-2)\})^+}{\|D_k^{-1}C_k(\hat{\mathbf{x}}^{\mathrm{ML}}_k - \hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1})\|^2}\right)^+ \cdot(\hat{\mathbf{x}}^{\mathrm{ML}}_k - \hat{\mathbf{x}}^{\mathrm{JS}}_{k|k-1}) \tag{38}$$

$$\hat{\mathbf{x}}^{\mathrm{JS}}_{k+1|k} = A_k\hat{\mathbf{x}}^{\mathrm{JS}}_{k|k} \tag{39}$$

where

$$p^* = \frac{\mathrm{tr}\{(C_k'(D_kD_k')^{-1}C_k)^{-1}\}}{\lambda_{\max}\{(C_k'(D_kD_k')^{-1}C_k)^{-1}\}} \tag{40}$$

$$\hat{\mathbf{x}}^{\mathrm{ML}}_k = (C_k'(D_kD_k')^{-1}C_k)^{-1}C_k'(D_kD_k')^{-1}\mathbf{z}_k. \tag{41}$$

[5] As in the Kalman filter, knowledge of the state-space model (29) allows estimates of all the states to be possible. The consequence of assuming no knowledge of the state-space model (Assumption 4) is that certain (linear combinations of) states become unobservable.

[6] If $p \leq 2$, the JSSF can still be applied, but it reduces to the MLE $\hat{\mathbf{x}}^{\mathrm{ML}}_k$ (31).

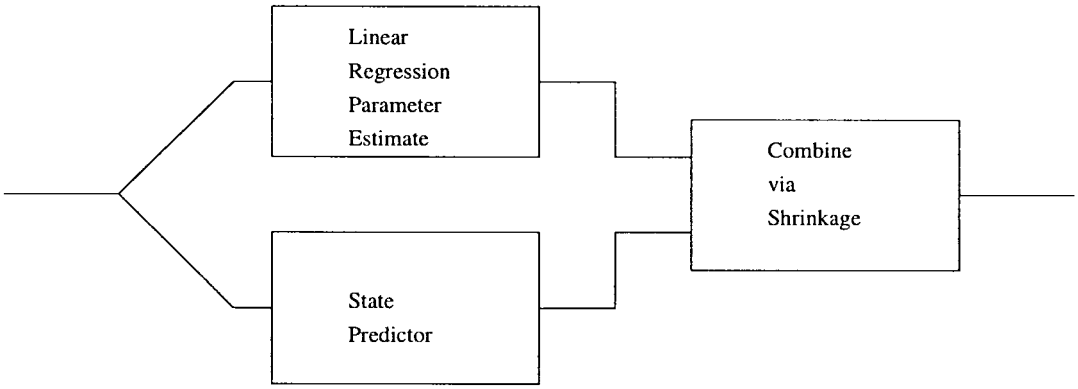

Fig. 3. James–Stein state filter (feedback not shown).

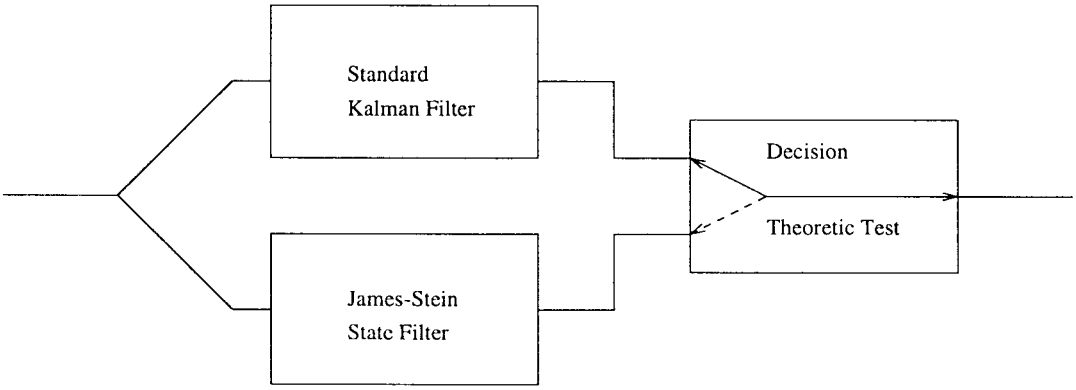

Fig. 4. James–Stein Kalman filter with hypothesis test (feedback not shown).

Remark: If σ^2 is unknown, it can be replaced (providing $n > p$) in (38) by [see (9)]

$$\sigma^2 = \frac{||D_k^{-1}(\mathbf{z}_k - C_k\hat{\mathbf{x}}_k^{\mathrm{ML}})||^2}{n-p+2}. \tag{42}$$

Note that this is *not* an estimate of σ^2 as such. In [7], (9) is chosen such that the JSE "works," i.e., the JSE continues to dominate the MLE. Naively replacing σ^2 by an estimate of it will, in general, cause the JSE to no longer dominate the MLE.

Risk: Define $J_k^{\mathrm{JSSF}} = \mathbf{E}[||\hat{\mathbf{x}}_{k|k}^{\mathrm{JS}} - \mathbf{x}_k||^2]$, which is the risk of the JSSF state estimate $\hat{\mathbf{x}}_{k|k}^{\mathrm{JS}}$ (38).

Computational Complexity: The computational complexity of the JSSF algorithm above is of the same order as the standard KF algorithm (32)–(35).

Discussion: Let us explain why the above JSSF algorithm is robust.

1) The JSSF achieves something that, at first glance, does not seem possible. Regardless of how incorrect the system dynamics are, $J_k^{\mathrm{JSSF}} < J_k^{\mathrm{ML}}$. That is, the JSSF always performs better (i.e., smaller MSE) than ignoring the system dynamics (29) and using only the observations $\mathbf{z}_k$ (30) to estimate the state $\mathbf{x}_k$ (30) by the traditional MLE, i.e., $\hat{\mathbf{x}}_k^{\mathrm{ML}}$ (41).
2) The more accurate the system dynamics (29) are, the smaller J_k^{JSSF} is. That is, the closer $\hat{\mathbf{x}}_{k|k-1}^{\mathrm{JS}}$ (39) is to $\mathbf{x}_k$ (29), the smaller the MSE of $\hat{\mathbf{x}}_{k|k}^{\mathrm{JS}}$ (38).
3) The filter is robust to perturbations in A_k, B_k, and even nonnormality of the noise $\mathbf{e}_k$. (See point 1 above.)
4) Equation (39) can be generalized to $\hat{\mathbf{x}}_{k+1|k}^{\mathrm{JS}} = f(\hat{\mathbf{x}}_{k|k}^{\mathrm{JS}})$, where f: $\mathbb{R} \to \mathbb{R}$ is any (e.g., nonlinear) prediction of $\mathbf{x}_{k+1}$ based on $\mathbf{x}_k$.

D. James–Stein Kalman Filter with Hypothesis Test (JSKF$_{\mathrm{H}}$)

The JSSF assumes complete ignorance about the accuracy of the state-space model (29). This section assumes that at each time instant, the state-space model parameters are either correct or incorrect. We propose a test statistic to decide at each time instant whether or not the state-space model is correct. If the state-space model is believed to be correct, the KF is used to estimate the state at that time instant. Otherwise, the JSSF is used. The resulting algorithm, which is termed the JSKF$_{\mathrm{H}}$, is illustrated in Fig. 4 (feedback paths between the standard KF and JSSF have been omitted for clarity).

An example application is to track a target subject to maneuvers. While not maneuvering, the target moves with known dynamics. When a maneuver occurs, the state-space model is inaccurate. The ordinary Kalman filter, if it is designed to track the target accurately during nonmaneuvering periods, may be slow in responding to the new change of coordinates induced by the maneuver.

Model: We assume the linear Gaussian state-space model of Section IV-A, with the following changes.

1) Require $n \geq p > 2$.
2) Require C_k and D_k to be accurately known and have full (column) rank. (We also assume A_k, B_k, and σ^2 to be accurately known.)

3) Assume $\mathbf{e}_k$ consists of Gaussian noise plus occasional outliers and is independent of $\mathbf{w}_k$. Mathematically, at any time instant k, either the state-space model (29) is accurate, and

$$\mathbf{e}_k \sim N(0, Q_k) \tag{43}$$

or the state-space model is sufficiently inaccurate, such that

$$\|\mathbf{e}_k\|^2 \gg \mathrm{tr}\{Q_k\}. \tag{44}$$

Remark: Equation (44) expresses the criterion that $\mathbf{e}_k$ be much larger than its average value under (43). This criterion can be met either by $\mathbf{e}_k$ being large or by A_k and/or B_k being sufficiently inaccurate.

Derivation of JSKF$_{\mathrm{H}}$: There are two steps in the derivation below. A decision theoretic test is derived to decide whether or not the KF is functioning correctly. The computation of the James–Stein equivalent of the Kalman covariance $P_{k|k-1}$ is then derived.

Consider the hypothesis test at each time instant k

$H0$: accurate state-space model, i.e., $\mathbf{e}_k \sim N(0, Q_k)$.

$H1$: inaccurate state-space model, i.e., $\|\mathbf{e}_k\|^2 \gg \mathrm{tr}\{Q_k\}$.

The presence of a large $\|\mathbf{e}_k\|^2$ can be detected by examining $\|\mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1}\|^2$, which is the (squared) distance between the actual observation and the predicted observation. A small norm suggests $H0$; a large norm suggests $H1$. More precisely, under $H0$,

$$\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \sim N(0, P_{k|k-1}) \tag{45}$$

$$\mathbf{z}_k - C_k\mathbf{x}_k \sim N(0, \sigma^2 D_k D_k') \tag{46}$$

$$\therefore \mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1} \sim N(0, C_k P_{k|k-1} C_k' + \sigma^2 D_k D_k'). \tag{47}$$

Define the test statistic

$$T = (\mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1})'(C_k P_{k|k-1} C_k' + \sigma^2 D_k D_k')^{-1} \cdot (\mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1}). \tag{48}$$

From (47), it follows that under $H0$, $T \sim \chi_n^2$ (the Chi-squared distribution with n degrees of freedom). Therefore, we propose the following standard decision theoretic test [19] to choose between $H0$ and $H1$

$$T \underset{H1}{\overset{H0}{\lessgtr}} T_c \tag{49}$$

where T_c is the threshold, or cut-off, constant. The value of T_c can be chosen to give a fixed probability of false alarm based on (48).

At each time instant, the KF computes the covariance of the prediction error $P_{k|k-1}$ (36). The empirical Bayesian viewpoint of a James–Stein estimator [14] suggests that in a sense, the James–Stein estimator implicitly estimates $P_{k|k-1}$. (It can be shown along similar lines that the estimated $P_{k|k-1}$ is typically much larger than the actual $P_{k|k-1}$. More precisely, it is a biased estimate, erring on the side of "caution," or larger $P_{k|k-1}$.)

To compute the final form of our filter, let us compare the KF and JSSF.

Under $H0$, the optimal state estimate is given by the Kalman filter (32)

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1}). \tag{50}$$

Under $H1$, we use the JSE (38) instead, namely

$$\hat{\mathbf{x}}_{k|k}^{\mathrm{JS}} = \hat{\mathbf{x}}_{k|k-1}^{\mathrm{JS}} + s(\cdot)(\hat{\mathbf{x}}_k^{\mathrm{ML}} - \hat{\mathbf{x}}_{k|k-1}^{\mathrm{JS}}) \tag{51}$$

$$= \hat{\mathbf{x}}_{k|k-1}^{\mathrm{JS}} + s(\cdot)(C_k'(D_kD_k')^{-1}C_k)^{-1}C_k'(D_kD_k')^{-1} \cdot (\mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1}^{\mathrm{JS}}) \tag{52}$$

where

$$s(\cdot) = \left(1 - \sigma^2 \frac{(\min\{(p-2), 2(p^*-2)\})^+}{\|D_k^{-1}C_k(\hat{\mathbf{x}}_k^{\mathrm{ML}} - \hat{\mathbf{x}}_{k|k-1}^{\mathrm{JS}})\|^2}\right)^+. \tag{53}$$

Both estimators now have the form $\hat{\mathbf{x}}_{k|k}^{\mathrm{H}} = \hat{\mathbf{x}}_{k|k-1}^{\mathrm{H}} + \rho(\cdot)(\mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1}^{\mathrm{H}})$. Comparing (50) and (52) shows that the JSSF simply replaces the Kalman gain K_k with the new gain $K_k^{\mathrm{JS}} = s(\cdot)(C_k'(D_kD_k')^{-1}C_k)^{-1}C_k'(D_kD_k')^{-1}$. Furthermore, inverting (33) gives

$$P_{k|k-1} = \sigma^2(I - K_kC_k)^{-1}K_kD_kD_k'C_k(C_k'C_k)^{-1}. \tag{54}$$

Replacing K_k by K_k^{JS} in (54) yields the James–Stein equivalent of the Kalman covariance $P_{k|k-1}$

$$P_{k|k-1}^{\mathrm{JS}} = \sigma^2 \frac{s(\cdot)}{1 - s(\cdot)}(C_k'(D_kD_k')^{-1}C_k)^{-1}. \tag{55}$$

Collecting these ideas together leads to the following JSKF$_{\mathrm{H}}$ algorithm.

Algorithm 4—James–Stein Kalman Filter with Hypothesis Test:

Initialization:

- Choose an appropriate threshold T_c [cf., (48) and (49)].
- Initialize KF parameters

$$\hat{\mathbf{x}}_{1|0}^H = \mathbf{E}[\mathbf{x}_1] \tag{56}$$

$$P_{1|0}^H = \mathbf{E}[(\hat{\mathbf{x}}_{1|0} - \mathbf{x}_1)(\hat{\mathbf{x}}_{1|0} - \mathbf{x}_1)']. \tag{57}$$

Recursive Filter:

1) At time instant k, compute the test statistic

$$T = (\mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1}^H)'(C_kP_{k|k-1}^HC_k' + \sigma^2D_kD_k')^{-1} \cdot (\mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1}^H). \tag{58}$$

If $T \le T_c$ ($H0$ holds), use the standard KF in Step 3. Otherwise ($H1$ holds), use the JSSF by executing Steps 2 and then 3.

2) Calculate the James–Stein equivalent of $P_{k|k-1}$

$$P_{k|k-1}^{\mathrm{JS}} = \sigma^2 \frac{s(\cdot)}{1 - s(\cdot)}(C_k'(D_kD_k')^{-1}C_k)^{-1} \tag{59}$$

where

$$s(\cdot)=\left(1-\sigma^2\frac{(\min\{(p-2),\,2(p^*-2)\})^+}{\|D_k^{-1}C_k(\hat{\mathbf{x}}_k^{\mathrm{ML}}-\hat{\mathbf{x}}_{k|k-1}^H)\|^2}\right)^+ \quad (60)$$

$$\hat{\mathbf{x}}_k^{\mathrm{ML}}=(C_k'(D_kD_k')^{-1}C_k)^{-1}C_k'(D_kD_k')^{-1}\mathbf{z}_k \quad (61)$$

$$p^*=\frac{\mathrm{tr}\{(C_k'(D_kD_k')^{-1}C_k)^{-1}\}}{\lambda_{\max}\{(C_k'(D_kD_k')^{-1}C_k)^{-1}\}}. \quad (62)$$

3) For the standard Kalman filter

$$K_k^H=P_{k|k-1}C_k'[C_kP_{k|k-1}C_k'+\sigma^2D_kD_k']^{-1} \quad (63)$$

$$\hat{\mathbf{x}}_{k|k}^H=\hat{\mathbf{x}}_{k|k-1}^H+K_k^H(\mathbf{z}_k-C_k\hat{\mathbf{x}}_{k|k-1}^H) \quad (64)$$

$$\hat{\mathbf{x}}_{k+1|k}^H=A_k\hat{\mathbf{x}}_{k|k}^H \quad (65)$$

$$P_{k+1|k}^H=A_k[I-K_k^HC_k]P_{k|k-1}A_k'+B_kQ_{k+1}B_k' \quad (66)$$

where

$$P_{k|k-1}=\begin{cases}P_{k|k-1}^H & \text{under } H0\\ P_{k|k-1}^{\mathrm{JS}} & \text{under } H1.\end{cases} \quad (67)$$

4) Increment k, and return to Step 1.

Risk: Define $J_k^{\mathrm{JSKF_H}}=\mathbf{E}[\|\hat{\mathbf{x}}_{k|k}^{\mathrm{H}}-\mathbf{x}_k\|^2]$ as the risk of the $\mathrm{JSKF_H}$ state estimate $\hat{\mathbf{x}}_{k|k}^{\mathrm{H}}$ (32).

Remark: As T_c in (49) approaches 0, the filter becomes the JSSF presented in Algorithm 3. As $T_c\to\infty$, the filter becomes the ordinary Kalman filter.

V. Extensions: Asymptotic and Heuristic Algorithms

Up to this point, the results in this paper have been based on the JSE (7) applied to the linear regression (3), resulting in the JS-RLS and JSSF algorithms. The results thus far have been rigorous.

The estimators presented in this section may have a larger MSE than their corresponding traditional estimators. However, we include these estimators not only for the sake of completeness but also because simulation results show that in a number of cases, the James–Stein type estimators in this section can significantly reduce the MSE. This illustrates the potential for further research into extending the JSE to general distributions that approach (6) asymptotically.

A. James–Stein Version of the Yule–Walker Equations (JSYW)

Many statistical signal processing applications require estimating the parameters of an autoregressive (AR) process. The least squares estimate of the AR parameters is obtained by solving the Yule–Walker equations [20]. We now use the JSE (7) of Theorem 1 to present a James–Stein version of the Yule–Walker equations. Unfortunately, Theorem 1 is not strictly applicable since (6) holds only asymptotically.

Consider the real-valued observations $\{y(1),\cdots,y(K)\}$ from the AR(p) process

$$y(k)=\sum_{m=1}^{p}\alpha_m y(k-m)+w(k)$$
$$w(k)\sim\text{i.i.d. }N(0,\,\phi^2) \quad (68)$$

where $\alpha_m\in\mathbb{R}$, $m=1,\cdots,p$ are the AR parameters to be estimated. The variance ϕ^2 is an arbitrary positive constant.

The MLE[7] of the AR parameters (which is equivalent to the least squares estimator) is obtained by solving the Yule–Walker equations for $\mathbf{x}$

$$C\mathbf{x}=\mathbf{z} \quad (69)$$

where

$$\mathbf{x}=[\alpha_1,\cdots,\alpha_p]',\quad \mathbf{z}=[\gamma(1),\cdots,\gamma(p)]'$$
$$\gamma(\tau)=\frac{1}{K-p}\sum_{k=1}^{K-p}y(k)y(k+\tau) \quad (70)$$

and C is a Toeplitz symmetric $p\times p$ matrix with first row $\{\gamma(0),\gamma(1),\cdots,\gamma(p-1)\}$, i.e.,

$$(C)_{ij}=\gamma(|i-j|). \quad (71)$$

It is straightforward to verify that asymptotically, $\mathbf{z}=C\mathbf{x}+\mathbf{w}$, where $\mathbf{w}\sim N(0,\,\sigma^2C)$, and $\sigma^2=1/(K-p)$. Applying the JSE of Theorem 1 to the linear regression $\mathbf{z}=C\mathbf{x}+\mathbf{w}$ yields the following algorithm.

Algorithm 5—James–Stein Version of Yule–Walker Equations: Given K observations $\{y(1),\cdots,y(K)\}$ assumed to come from the pth-order AR process (68), the James–Stein estimate of the AR parameters can be calculated as follows.

1) Set $\overline{\mathbf{x}}$ to the *a priori* estimate of the true AR parameter $\mathbf{x}$. (If an *a priori* estimate is unavailable, set $\overline{\mathbf{x}}$ to the zero vector.)
2) Compute C from (70) and (71). Compute $\mathbf{z}=[\gamma_1,\cdots,\gamma_p]'$ from (70).
3) Solve the standard Yule–Walker equation (69), i.e.,

$$\hat{\mathbf{x}}^{\mathrm{ML}}=C^{-1}\mathbf{z}. \quad (72)$$

4) Apply the JSE to $\hat{\mathbf{x}}^{\mathrm{ML}}$ as [cf., (7) and (8)]

$$\sigma^2=\frac{1}{K-p} \quad (73)$$

$$p^*=\frac{\mathrm{tr}\{C^{-1}\}}{\lambda_{\max}\{C^{-1}\}} \quad (74)$$

$$\hat{\mathbf{x}}^{\mathrm{JS}}=\overline{\mathbf{x}}+\left(1-\sigma^2\frac{(\min\{(p-2),\,2(p^*-2)\})^+}{(\hat{\mathbf{x}}^{\mathrm{ML}}-\overline{\mathbf{x}})'C(\hat{\mathbf{x}}^{\mathrm{ML}}-\overline{\mathbf{x}})}\right)^+\cdot(\hat{\mathbf{x}}^{\mathrm{ML}}-\overline{\mathbf{x}}). \quad (75)$$

Risk: Define $J_k^{\mathrm{YW}}=\mathbf{E}[\|\hat{\mathbf{x}}^{\mathrm{ML}}-\mathbf{x}\|^2]$ and $J_k^{\mathrm{JSYW}}=\mathbf{E}[\|\hat{\mathbf{x}}^{\mathrm{JS}}-\mathbf{x}\|^2]$ as the risks of the Yule–Walker estimate $\hat{\mathbf{x}}^{\mathrm{ML}}$ (72) and of the James–Stein Yule–Walker estimate $\hat{\mathbf{x}}^{\mathrm{JS}}$ (75), respectively.

Remarks:

1) Note that $\hat{\mathbf{x}}^{\mathrm{JS}}=\hat{\mathbf{x}}^{\mathrm{ML}}$ if the effective dimension $p^*\le 2$. The effective dimension depends on the correlation matrix C, which, in turn, depends on the actual AR parameters. (A necessary condition for $p^*>2$ is $p\ge 3$, i.e., a James–Stein estimator requires at least three dimensions before it can dominate the MLE.)

[7] More precisely, the approximate (but asymptotically correct) MLE.

2) For a reduction in MSE, the origin $\overline{\mathbf{x}}$ (which is defined in Step 1 of Algorithm 5) should be close to the true AR parameter $\mathbf{x}$. [This would be a consequence of Theorem 1 if (6) held. Although we do not attempt to prove it, simulations suggest that for $\overline{\mathbf{x}}$ sufficiently close to $\mathbf{x}$, the JSE (75) will have a smaller MSE than the MLE (72).]

B. Modified JS-RLS

The JS-RLS algorithm (Algorithm 2) is given for the general ARX model (see Section III-A), although its guarantee of a smaller MSE compared with the RLS algorithm is only valid for the special case of an exogenous input model.

Simulations suggest that provided our *a priori* estimate $\overline{\mathbf{x}}$ [which is defined in (19)] is sufficiently close to the true parameter $\mathbf{x}_k$, the JS-RLS will have a smaller MSE than the RLS, even for the general ARX model (14). Since (6) holds asymptotically, we also expect the JS-RLS to have a smaller MSE for sufficiently large k.

Throughout this paper, we have shifted the origin (see Section II-A) to a suitable *a priori* estimate of the true parameter. Due to the convex risk function (see Theorem 1) of the JSE, we claim that the reduction in MSE caused by using the JSE rather than the MLE is significant if the *a priori* estimate of the true parameter value is accurate. This is the key idea in the modified JS-RLS algorithm below.

Motivation: $\hat{\mathbf{x}}_{k-1}^{\mathrm{JS}}$ (27) represents the "best *a priori* guess" of the parameter vector $\mathbf{x}_k$ (14) at time instant k. Since significant reduction in MSE occurs if $\overline{\mathbf{x}}_k$ (19) is close to $\mathbf{x}_k$, it is tempting to set $\overline{\mathbf{x}} = \hat{\mathbf{x}}_{k-1}^{\mathrm{JS}}$. Unfortunately, $\hat{\mathbf{x}}_{k-1}^{\mathrm{JS}}$ is correlated with $\hat{\mathbf{x}}_k$ (22), and it is feasible that $\hat{\mathbf{x}}_{k-1}^{\mathrm{JS}}$ is correlated in such a way as to make the MSE worse rather than better.

Therefore, in an attempt to reduce the correlation between $\hat{\mathbf{x}}_{k-1}^{\mathrm{JS}}$ and $\hat{\mathbf{x}}_k$, $\overline{\mathbf{x}}_k$ is updated by $\overline{\mathbf{x}}_k = \alpha\hat{\mathbf{x}}_k^{\mathrm{JS}} + (1-\alpha)\overline{\mathbf{x}}_{k-1}$. For small α, $\overline{\mathbf{x}}_k$ will (hopefully) creep toward the true origin and be sufficiently uncorrelated with $\hat{\mathbf{x}}_k$ to allow (6) to hold approximately.

Algorithm: The modified JS-RLS algorithm is identical to Algorithm 2 with the following update performed at the end of each iteration

$$\overline{\mathbf{x}}_k = \alpha\hat{\mathbf{x}}_k^{\mathrm{JS}} + (1-\alpha)\overline{\mathbf{x}}_{k-1} \tag{76}$$

where $0 \le \alpha \le 1$.

Remark: Simulations verify the following intuitive ideas. For $\lambda = 1$, $\hat{\mathbf{x}}_k$ contains as much information as possible about the true parameter; hence, $\alpha > 0$ is attempting to "reuse the data" and leads to a larger MSE of the parameter estimate. For $\lambda < 1$, $\hat{\mathbf{x}}_k$ "loses" (or "forgets") past information about the true parameter, and hence, a nonzero α can (but not always) have a significant improvement.

VI. Simulation Results

This section presents computer simulation results of the James–Stein versions of the Kalman filter detailed in Section IV, the James–Stein Recursive Least Squares algorithm of Section III, and the James–Stein Yule–Walker equations of Section V-A.

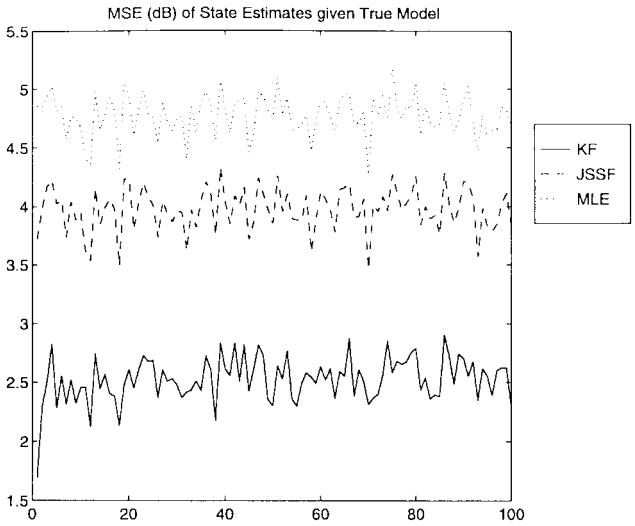

Fig. 5. Simulated transient response of the Kalman filter, the James–Stein state filter, and the maximum-likelihood estimator under the true model (see Section IV). Here, the horizontal axis represents time k, whereas the vertical axis is the risk in decibels.

A. James–Stein State Filters (JSSF and JSKF$_{\mathrm{H}}$)

James–Stein State Filter (JSSF): The model (29) and (30) was used to generate 500 data points with parameters

$$A_k = \begin{bmatrix} 1.0 & -0.1 & -0.1 \\ 0.2 & 0.9 & -0.1 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$$

$$B_k = C_k = D_k = Q_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \sigma^2 = 1. \tag{77}$$

Three models were used to filter the data: the correct model, which was a perturbed model where A_k and B_k had their elements corrupted by $N(0, 0.0625)$ noise, and a totally incorrect model, where

$$A_k = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad B_k = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}. \tag{78}$$

The risks J_k^{ML}, J_k^{KF}, and J_k^{JSSF} for $k = 1, \cdots, 500$ were computed for the three models by averaging 500 independent runs of the KF (32)–(35) and the JSSF (Algorithm 3). Figs. 5–7 display these risks for the first 100 data points (i.e., $k = 1, \cdots, 100$). Table I presents the average MSE (i.e., $\sum_{k=1}^{500} J_k/500$) of the MLE, KF, and JSSF state estimates.

We make the following observations.

- The JSSF always gives state estimates with smaller MSE than the MLE regardless of how incorrect the model is.
- Even small perturbations in the model parameters cause the KF to have a larger MSE than the MLE.
- As the model parameters become more accurate, the MSE of the JSSF state estimates decrease.

These results are in agreement with our theory, showing that the JSSF, unlike the KF, can never perform worse than the MLE.

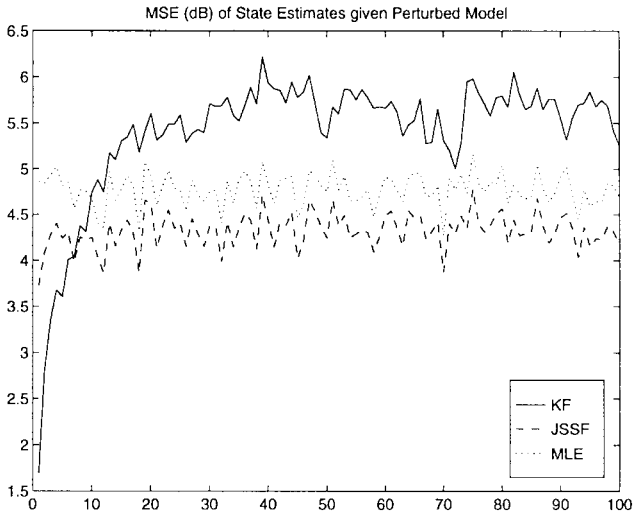

Fig. 6. Simulated transient response of the Kalman filter, the James–Stein state filter, and the maximum-likelihood estimator under a perturbed model (see Section IV). Here, the horizontal axis represents time k, whereas the vertical axis is the risk in decibels.

Fig. 7. Simulated transient response of the Kalman filter, the James–Stein state filter, and the maximum-likelihood estimator under a totally incorrect model (see Section IV). Here, the horizontal axis represents time k, whereas the vertical axis is the risk in decibels. Note that the JSSF estimate and the MLE are almost identical.

TABLE I
PERFORMANCE OF JSSF. UNDER "ABSOLUTE MSE," EACH ENTRY IS $10\log_{10}(\sum_{k=1}^{500} J_k/500)$, WHERE J_k IS J_k^{ML}, J_k^{KF}, OR J_k^{JSSF}

	Absolute MSE			Improvement to MLE	
Model	MLE (dB)	KF (dB)	JSSF (dB)	KF (dB)	JSSF (dB)
Correct	4.771	2.519	3.976	2.252	0.795
Perturbed	4.771	5.595	4.331	-0.824	0.440
Incorrect	4.771	7.336	4.759	-2.565	0.012

Two-Dimensional (2-D) Target Tracking Example (JSSF): A 2-D maneuvering target tracking example is given here. In this case, the observation matrix C_k (30) has equal or more rows than columns (i.e., $n \geq p$), and hence, the JSSF can be used to estimate the (amplitudes of the) intensities of the targets from noisy measurements. Note, in general, that since n is related to the number of sensors used to observe the current state, systems that use many sensors will satisfy the JSSF requirement of $n \geq p$.

TABLE II
PERFORMANCE OF JSSF IN 2-D TRACKING. EACH ENTRY IS $10\log_{10}(J_k^{\mathrm{ML}}/J_k)$ EVALUATED AT $k=10$, WHERE J_k IS J_k^{KF} OR J_k^{JSSF}

α	0.1	0.2	0.5	0.8	0.9	0.98	1
KF (dB)	-2.483	-2.160	-0.5537	2.168	2.913	3.148	3.145
JSSF (dB)	0.6651	0.7598	1.134	1.480	1.474	1.403	1.374

In [27], an estimation procedure was derived to track multiple targets with time-varying amplitudes based on 2-D optical observations. We concern ourselves with the subproblem of determining the (amplitude of the) intensities of the targets assuming we know their locations. The observation model of is an example of when $n > p$ [i.e., C_k in (30) has more rows than columns]. In particular, the following simulation example uses $n = 16$ sensors and $p = 4$ states. Each of the $n = 16$ sensors measures the light intensity in one cell of a 4×4 grid. There are $p = 4$ stationary light sources (targets).

Number the cells in a 4×4 observation grid (arbitrarily) from 1 to 16. The ith element of $\mathbf{z}_k \in \mathbb{R}^{16}$ is the light intensity observed in cell i at time instant k. We assume there are four stationary light sources (targets) with $\mathbf{x}_k \in \mathbb{R}^4$ denoting their intensities. The observation model is (30), where C_k represents the 2-D point spread function [we used a Gaussian point spread function in our simulations with the resulting C_k given in (79)]. The time-varying source intensities were generated from (29) with $A_k = 0.98I$ and $B_k = I$. A frame ($k = 1, \cdots, 10$) of observations was generated from (30) with

$$C_k = \begin{bmatrix} 0.0862 & 0.0000 & 0.0002 & 0.0000 \\ 0.0117 & 0.0000 & 0.0862 & 0.0000 \\ 0.0000 & 0.0000 & 0.6366 & 0.0000 \\ 0.0000 & 0.0000 & 0.0862 & 0.0002 \\ 0.6366 & 0.0000 & 0.0000 & 0.0000 \\ 0.0862 & 0.0002 & 0.0117 & 0.0000 \\ 0.0002 & 0.0000 & 0.0862 & 0.0117 \\ 0.0000 & 0.0000 & 0.0117 & 0.0862 \\ 0.0862 & 0.0117 & 0.0000 & 0.0000 \\ 0.0117 & 0.0862 & 0.0000 & 0.0002 \\ 0.0000 & 0.0117 & 0.0002 & 0.0862 \\ 0.0000 & 0.0000 & 0.0000 & 0.6366 \\ 0.0002 & 0.0862 & 0.0000 & 0.0000 \\ 0.0000 & 0.6366 & 0.0000 & 0.0000 \\ 0.0000 & 0.0862 & 0.0000 & 0.0117 \\ 0.0000 & 0.0002 & 0.0000 & 0.0862 \end{bmatrix}, \; D_k = I. \quad (79)$$

Both the KF and the JSSF (Algorithm 3) were then used to filter the data but with $A_k = \alpha I$, where $0 < \alpha \leq 1$ is the autoregressive coefficient used to model the source intensities.

Table II presents the MSE of the state estimate at the end of the frame (i.e., $k = 10$) relative to the MLE state estimate based on (30) alone (i.e., $J_k^{\mathrm{ML}}/J_k^{\mathrm{KF}}$ and $J_k^{\mathrm{ML}}/J_k^{\mathrm{JSSF}}$). (The MSE was estimated by averaging the results of 1000 independent runs.)

We make the following observations from Table II.

- For $\alpha \leq 0.5$, the KF state estimate is worse than the MLE.
- However, the JSSF state estimate is always better than the MLE.

TABLE III
PERFORMANCE OF THE JSKF_H ALGORITHM. EACH ENTRY IS $10\log_{10}(\sum_{k=1}^{1000} J_k/1000)$, WHERE J_k IS J_k^{ML}, J_k^{KF}, OR J_k^{JSSF}

	ML (dB)	KF (dB)	JSKF_H (dB)							
false alarm	-	-	0.01%	0.05%	0.1%	0.5%	1%	5%	10%	20%
$P_e = 0.02$	4.772	3.635	2.785	2.780	2.780	2.817	2.859	3.129	3.349	3.620
$P_e = 0.1$	4.776	3.451	3.048	3.016	3.005	3.008	3.032	3.239	3.425	3.672

Thus, in the range $\alpha \leq 0.5$, the JSSF yields superior estimates of the target intensities compared with the standard Kalman filter algorithm.

James–Stein Kalman Filter with Hypothesis Test (JSKF_H): The JSKF_H (Algorithm 4) was used to filter 1000 data points generated by the model [cf., (29) and (30)]

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{e}_{k+1} & \text{with probability } 1 - P_e \\ 0 \in \mathbb{R}^3 & \text{with probability } P_e \end{cases} \tag{80}$$

$$\mathbf{z}_k = \mathbf{x}_k + \mathbf{w}_k \tag{81}$$

where $\mathbf{x}_k \in \mathbb{R}^3$. In addition, $\mathbf{w}_k \in \mathbb{R}^3$ and $\mathbf{z}_k \in \mathbb{R}^3$ are $N(0, I)$ uncorrelated white noise processes, and $0 < P_e < 1$ is the probability that the state $\mathbf{x}_k$ will be reset to zero. The average MSE (i.e., $\sum_{k=1}^{1000} J_k/1000$) of the MLE, the KF, and the JSKF_H are presented in Table III. (The MSE was estimated by averaging together the results of 500 independent runs.)

The following observation can be made based on Table III:

- For the threshold T_c [cf., (49)] set for $\leq$10% false alarm rate, the JSKF_H state estimate had a smaller MSE than that of the KF state estimate.

We conclude that the JSKF_H significantly outperforms the KF for the model under consideration. It is pleasing to note that in this example, the correct choice of T_c in the JSKF_H is not critical.

B. James–Stein Recursive Least Squares (JS-RLS)

The JS-RLS algorithm (Algorithm 2) is used to estimate the parameters of an finite impulse response (FIR) channel. The model (14) with $r = 0$, $q = 5$ is used to generate 1000 data points, with the exogenous input $u(k)$ a white noise process [i.e., $u(k) \sim$ i.i.d. $N(0, 1)$]. Two different sets of parameters were used, namely, $\mathbf{x}_k = [0.4, 0.1, 0.2, 0.3, 0.4]'$ and $\mathbf{x}_k = [1, 2, 3, 4, 5]'$. In both cases, no *a priori* estimate of the true parameter $\mathbf{x}_k$ was used, i.e., $\overline{\mathbf{x}}_k = 0$ initially. The results of 750 independent runs of the JS-RLS algorithm were averaged to estimate the initial (82), final (83), and total (84) improvement in the MSE of the parameter estimates. These estimates are presented in Tables IV and V. The entries in the tables are defined by

$$\text{Initial} = 10\log_{10}\left(\sum_{k=15}^{45} J_k^{\text{RLS}} \Big/ \sum_{k=15}^{45} J_k^{\text{JS-RLS}}\right) \tag{82}$$

$$\text{Final} = 10\log_{10}\left(\sum_{k=970}^{1000} J_k^{\text{RLS}} \Big/ \sum_{k=970}^{1000} J_k^{\text{JS-RLS}}\right) \tag{83}$$

$$\text{Total} = 10\log_{10}\left(\sum_{k=15}^{1000} J_k^{\text{RLS}} \Big/ \sum_{k=15}^{1000} J_k^{\text{JS-RLS}}\right). \tag{84}$$

We make the following observations about the results in Tables IV and V for $\lambda = 1$ and $\alpha = 0$.

TABLE IV
PERFORMANCE OF JS-RLS FOR FIR CHANNEL IDENTIFICATION. TRUE PARAMETER $\mathbf{x}_k = [0.4,\ 0.1,\ 0.2,\ 0.3,\ 0.4]'$. INITIAL, FINAL, AND TOTAL ARE DEFINED BY (82)–(84)

	$\lambda = 0.95$								$\lambda = 1$
α	0	0.01	0.02	0.05	0.1	0.2	0.5	1	0
Initial (dB)	0.709	0.792	0.890	1.08	1.02	0.555	-0.280	-0.966	0.562
Final (dB)	0.0129	8.73	9.46	8.39	7.29	6.33	5.22	4.09	0.0114
Total (dB)	0.104	4.10	5.14	5.46	5.07	4.48	3.71	2.93	0.299

TABLE V
PERFORMANCE OF JS-RLS FOR FIR CHANNEL IDENTIFICATION. TRUE PARAMETER $\mathbf{x}_k = [1,\ 2,\ 3,\ 4,\ 5]'$. INITIAL, FINAL, AND TOTAL ARE DEFINED BY (82)–(84)

	$\lambda = 0.95$								$\lambda = 1$
α	0	0.01	0.02	0.05	0.1	0.2	0.5	1	0
Initial (dB)	0.0037	-0.0532	-0.780	-6.58	-9.04	-9.13	-7.44	-4.17	0.0045
Final (dB)	-0.0043	-12.9	-11.9	-10.6	-9.60	-8.44	-6.49	-5.02	0.0008
Total (dB)	0.0004	-12.9	-12.6	-11.8	-11.0	-10.0	-8.08	-6.21	0.0025

- As guaranteed by Theorem 1, the JS-RLS estimates have smaller MSE than the RLS estimates.
- As indicated by Theorem 1, the improvement in MSE is more significant if $\overline{\mathbf{x}}_k$ is close to $\mathbf{x}_k$.
- The savings in MSE are greatest for small k. This is because for small k, the RLS estimate is (relatively) inaccurate; therefore, shrinking the estimate toward the origin leads to a noticeable reduction in MSE.

We make the following observations about the results in Table IV for $\lambda = 0.95$.

- For small k, the optimal α (i.e., the one that gives the greatest savings) is slightly higher than the optimal α for large k.
- The asymptotic MSE (i.e., final) savings can exceed 9 dB. Therefore, α can be used to compensate for $\lambda < 1$. (As λ is decreased, the asymptotic MSE increases.)

We make the following observations about the results in Table V for $\lambda = 0.95$.

- For $\lambda < 1$, the negative entry in final for $\alpha = 0$ shows that the MSE of the JS-RLS estimate need not be smaller than that of the RLS estimate due to the mismatch in variance (see Derivation of JS-RLS in Section III-B).
- We cannot find an $\alpha > 0$ to give a smaller MSE than for $\alpha = 0$.

The vast difference between Tables IV and V is attributed to the true parameter in Table V being relatively far away from the initial $\overline{\mathbf{x}}_k = 0$. It shows that the heuristic idea of updating $\overline{\mathbf{x}}_k$ by (76) only works well if $\overline{\mathbf{x}}_k$ is originally close to $\mathbf{x}_k$. However, for $\alpha = 0$ and $\lambda = 1$, the JS-RLS parameter estimates always have smaller MSE's than the RLS parameter estimates.

C. AR Parameter Estimation (JS-RLS and JSYW)

James–Stein Recursive Least Squares (JS-RLS): The averaged results of 500 independent runs of the JS-RLS algorithm

TABLE VI
PERFORMANCE OF JS-RLS FOR AR(3) MODEL. TRUE PARAMETER $\mathbf{x}_k = [0.1, -0.1, -0.2]'$. k IS THE NUMBER OF DATA POINTS, MSE (IN DECIBELS) IS $10\log_{10}(J^{\text{JS-RLS}})$, AND IMPROVEMENT (IN DECIBELS) IS $10\log_{10}(\mathbb{R}^r/J^{\text{JS-RLS}})$, THE IMPROVEMENT IN DECIBELS OF JS-RLS RELATIVE TO RLS

	k	10	20	50	100	200	500	1000
$\overline{\mathbf{x}}_k = [0,0,0]'$	MSE (dB)	27.8	19.6	14.7	11.9	8.91	4.72	1.77
	Improvement (dB)	0.1128	0.4448	0.3746	0.0649	-0.0306	-0.0285	0.0076
$\overline{\mathbf{x}}_k = 0.95\mathbf{x}_k$	MSE (dB)	27.8	19.3	13.8	10.4	7.11	2.68	-0.177
	Improvement (dB)	0.1349	0.7549	1.327	1.545	1.764	2.006	1.957

TABLE VII
PERFORMANCE OF JS-RLS FOR AR(3) MODEL. TRUE PARAMETER $\mathbf{x}_k = [0.2, 0.2, -0.5]'$. k IS THE NUMBER OF DATA POINTS, MSE (IN DECIBELS) IS $10\log_{10}(J_k^{\text{JS-RLS}})$, AND IMPROVEMENT (IN DECIBELS) IS $10\log_{10}(\mathbb{R}^r/J_k^{\text{JS-RLS}})$, THE IMPROVEMENT IN DECIBELS OF JS-RLS RELATIVE TO RLS

	k	10	20	50	100	200	500	1000
$\overline{\mathbf{x}}_k = [0,0,0]'$	MSE (dB)	28.3	20.3	15.0	11.5	8.02	3.73	0.750
	Improvement (dB)	0.0446	-0.0615	-0.2216	-0.1170	-0.0654	-0.0439	-0.0190
$\overline{\mathbf{x}}_k = 0.95\mathbf{x}_k$	MSE (dB)	28.2	19.5	13.6	9.71	5.87	1.32	-1.31
	Improvement (dB)	0.1323	0.7064	1.268	1.661	2.087	2.367	2.039

TABLE VIII
PERFORMANCE OF JS-RLS FOR AR(3) MODEL. TRUE PARAMETER $\mathbf{x}_k = [0, 0, 0.9]'$. k IS THE NUMBER OF DATA POINTS, MSE (IN DECIBELS) IS $10\log_{10}(J_k^{\text{JS-RLS}})$, AND IMPROVEMENT (IN DECIBELS) IS $10\log_{10}(\mathbb{R}^r/J_k^{\text{JS-RLS}})$, THE IMPROVEMENT IN DECIBELS OF JS-RLS RELATIVE TO RLS

	k	10	20	50	100	200	500	1000
$\overline{\mathbf{x}}_k = [0,0,0]'$	MSE (dB)	28.0	19.8	12.1	7.82	3.72	-1.02	-4.65
	Improvement (dB)	-0.0246	-0.1488	-0.0911	-0.0729	-0.0448	-0.0260	-0.0171
$\overline{\mathbf{x}}_k = 0.95\mathbf{x}_k$	MSE (dB)	27.8	18.9	10.4	6.05	2.18	-1.21	-3.48
	Improvement (dB)	0.1369	0.7072	1.551	1.693	1.499	0.1642	-1.185

TABLE IX
PERFORMANCE OF JSYW EQUATIONS. TRUE PARAMETER $\mathbf{x}_k = [0.1, -0.1, -0.2]'$. k IS THE NUMBER OF DATA POINTS, MSE (IN DECIBELS) IS $10\log_{10}(J_k^{\text{JSYW}})$, AND IMPROVEMENT (IN DECIBELS) IS $10\log_{10}(J_k^{\text{YW}}/J_k^{\text{JSYW}})$, THE IMPROVEMENT IN MSE OF THE JSYW ESTIMATE OVER THE YW ESTIMATE

	k	10	20	50	100	200	500	1000
$\overline{\mathbf{x}} = [0,0,0]'$	MSE (dB)	41.6	19.2	14.4	11.6	8.91	4.85	1.93
	Improvement (dB)	0.0071	0.5502	0.3203	0.0720	0.0162	-0.0008	0.0096
$\overline{\mathbf{x}} = 0.95\mathbf{x}$	MSE (dB)	41.6	18.8	13.2	10.1	7.28	3.02	0.122
	Improvement (dB)	0.0088	0.8960	1.439	1.593	1.643	1.828	1.822

(Algorithm 2 with $\lambda = 1$ and $\alpha = 0$) applied to data generated by an AR(3) [see (14) with $r = 3$, $q = 0$] for three different parameters are presented in Tables VI–VIII.

We make the following observations (by "improvement" below, we mean the difference between the MSE of the RLS estimate and the MSE of the JS-RLS estimate).

- If $\overline{\mathbf{x}}_k = 0$ (i.e., $\overline{\mathbf{x}}_k$ not close to $\mathbf{x}_k$), the improvement in MSE is never less than -0.25 dB and asymptotically approaches 0 dB.
- If $\overline{\mathbf{x}}_k = 0.95\mathbf{x}_k$ (i.e., $\overline{\mathbf{x}}_k$ close to $\mathbf{x}_k$), two different behaviors were observed. For Tables VI and VII, the improvement in MSE rose sharply to an asymptotic value of around 2 dB. For Table VIII, however, the improvement rose sharply to around 2 dB and then fell to below -1 dB.

We conclude that the JS-RLS algorithm should be used with caution if an AR model is present. However, given a good *a priori* estimate, the JS-RLS can outperform the RLS algorithm.

James–Stein Yule–Walker Equations (JSYW): The averaged results of 500 independent runs of the JSYW equations applied to 1000 data points generated by an AR(3) process (68) for three different parameters are presented in Tables IX–XI. We make the following observations (by "improvement" below, we mean the difference between the MSE of the Yule–Walker estimate and the MSE of the James–Stein Yule-Walker estimate):

- If $\overline{\mathbf{x}} = 0$ (i.e., no *a priori* estimate available), the JSYW estimate in general has a larger MSE than the YW estimate. (Table IX is an exception, but observe that in this case, $\mathbf{x}$ is close to 0 as well.) More specifically, the improvement in Tables X and XI initially decreases and then increases toward 0 dB. Table IX shows the improvement to be initially positive and increasing, but after $k = 20$, it decreases and perhaps oscillates after $k = 500$.
- If $\overline{\mathbf{x}} = 0.95\mathbf{x}$ (i.e., accurate *a priori* estimate available), the JSYW estimate has a smaller MSE than the YW estimate. More specifically, the improvement in Tables IX–XI increases and then decreases as the data length k is increased.
- The JSYW estimate reduces to the YW estimate if the effective dimension (74) is less than or equal to two. We observed that for an arbitrary parameter vector $\mathbf{x}$, it is quite likely that the effective dimension drops below two. Therefore, the simulation results we present have parameters $\mathbf{x}$ chosen such that the effective dimension is above two.

The characteristic of the improvement is that it is either greatest (i.e., most positive) or worst (i.e., most negative) for medium data lengths (k), depending on whether or not $\overline{\mathbf{x}}$ is close to $\mathbf{x}$. In other words, for medium data lengths, the JSYW equations rely heavily on (the accuracy of) $\overline{\mathbf{x}}$. A likely explanation is that by using the approximation (6), too much

TABLE X
PERFORMANCE OF JSYW EQUATIONS. TRUE PARAMETER $\mathbf{x}_k = [0.2,\ 0.2,\ -0.5]'$. k IS THE NUMBER OF DATA POINTS, MSE (IN DECIBELS) IS $10\log_{10}(J_k^{\mathrm{JSYW}})$, AND IMPROVEMENT (IN DECIBELS) IS $10\log_{10}(J_k^{\mathrm{YW}}/J_k^{\mathrm{JSYW}})$, THE IMPROVEMENT IN MSE OF THE JSYW ESTIMATE OVER THE YW ESTIMATE

	k	10	20	50	100	200	500	1000
$\bar{\mathbf{x}} = [0,0,0]'$	MSE (dB)	39.9	20.0	14.4	11.2	8.09	3.82	0.853
	Improvement (dB)	0.0053	-0.1678	-0.267	-0.1317	-0.0659	-0.0327	-0.0154
$\bar{\mathbf{x}} = 0.95\mathbf{x}$	MSE (dB)	39.9	19.0	12.9	9.63	6.42	1.89	-0.772
	Improvement (dB)	0.0171	0.8185	1.272	1.409	1.611	1.893	1.609

TABLE XI
PERFORMANCE OF JSYW EQUATIONS. TRUE PARAMETER $\mathbf{x}_k = [0,\ 0,\ 0.9]'$. k IS THE NUMBER OF DATA POINTS, MSE (IN DECIBELS) IS $10\log_{10}(J_k^{\mathrm{JSYW}})$, AND IMPROVEMENT (IN DECIBELS) IS $10\log_{10}(J_k^{\mathrm{YW}}/J_k^{\mathrm{JSYW}})$, THE IMPROVEMENT IN MSE OF THE JSYW ESTIMATE OVER THE YW ESTIMATE

	k	10	20	50	100	200	500	1000
$\bar{\mathbf{x}} = [0,0,0]'$	MSE (dB)	50.0	30.5	12.3	7.92	3.59	-1.56	-4.71
	Improvement (dB)	-0.0006	-0.0225	-0.1202	-0.0881	-0.0546	-0.0274	-0.0183
$\bar{\mathbf{x}} = 0.95\mathbf{x}$	MSE (dB)	50.0	30.5	11.3	6.84	2.68	-1.94	-4.65
	Improvement (dB)	0.0010	0.0443	0.8427	0.9919	0.8496	0.3497	-0.0701

shrinkage occurs. This is most noticable for medium k since we have the following.

- For small k, the YW estimate has such a large MSE that shrinking toward the origin does little harm (i.e., the MSE of the JSYW estimate is comparable to the MSE of the YW estimate).
- For large k, the YW estimate will on average be closer than $\bar{\mathbf{x}}$ is to $\mathbf{x}$. Therefore, the JSYW estimate will rely more on the YW estimate rather than $\bar{\mathbf{x}}$.

We conclude that the JSYW can, but not always does, give AR parameter estimates that are better than the standard YW estimates.

VII. CONCLUSION AND FUTURE RESEARCH

This paper contains three main contributions. The first is the James–Stein estimator (7) for the linear regression (3), which has a MSE (risk) that never exceeds the MSE (risk) of the traditional MLE (5) (see Theorem 1). The second contribution is the James–Stein recursive least squares estimator (Algorithm 2), which recursively estimates the parameters of the ARX model (14) and, in certain (quite general) circumstances, provides a smaller MSE parameter estimate compared with the traditional RLS algorithm. The third and main contribution is the James–Stein state filter. The JSSF (Algorithm 3) is a robust filter. It gives state estimates with MSE less than the MSE of the traditional MLE applied to the observation equation (30) alone, regardless of how inaccurate the state-space model (29) is. The $\mathrm{JSKF_H}$ (Algorithm 4) implements the KF and the JSSF in parallel using a hypothesis test to determine which state estimate to use. We note that the computational complexity of the James–Stein algorithms are of the same order as their traditional counterparts.

Future Research: The $\mathrm{JSKF_H}$ (Algorithm 4) essentially switches between the KF (32)–(35) and the JSSF (Algorithm 3). A natural extension is to replace this "hard decision" switching with a "soft decision" (i.e., continuous) approach. The key idea is in the calculation of the covariance matrix of $\mathbf{x}_k$ (29). The Kalman filter calculates the covariance matrix $P_{k|k-1}$ (66). The James–Stein state filter estimates the covariance matrix by $P^{\mathrm{JS}}_{k|k-1}$ (59), which we would expect to be typically much larger than $P_{k|k-1}$. The larger the covariance matrix, the less emphasis is placed on the *a priori* distribution of $\mathbf{x}_k$ [which is determined by the state-space model (29)].

Consider forming the covariance matrix $P(t) = tP_{k|k-1} + (1-t)P^{\mathrm{JS}}_{k|k-1}$. Using $P(1)$ in the Kalman filter equation corresponds to the ordinary Kalman filter; using $P(0)$ corresponds to the James–Stein state filter. The former expresses 100% confidence in the accuracy of the state-space model, and the latter expresses 0%.

Heuristically, in situations where the state-space model may vary over time, sometimes being very accurate while at other times inaccurate, the modified Kalman filter with covariance matrix $P(t)$ may be used. The determination of t is expected to be based on $\|\hat{\mathbf{x}}^{\mathrm{ML}}_k - \hat{\mathbf{x}}_{k|k-1}\|^2$ (equivalently, on $\|\mathbf{z}_k - C_k\hat{\mathbf{x}}_{k|k-1}\|^2$) as well as any external information that may be available. Clearly, the $\mathrm{JSKF_H}$ of Section IV-D is a special case of this filter, where t is restricted to take values zero or one only.

APPENDIX
JUSTIFICATION OF (11)

This section justifies our choice of (11) for c in the JSE (10), where $\mathbf{X} \sim N(\boldsymbol{\mu}, \Omega)$.

Without loss of generality (Remark 3 following Theorem 1), let $\Omega = \mathrm{diag}\{\lambda_1, \cdots, \lambda_p\}$ and $P = \Omega^{-(1/2)}$. We showed in Section II-A that the risk of the estimate $\hat{\boldsymbol{\mu}}$ of the mean $\boldsymbol{\mu}$ given $\mathbf{X}$ [where $\mathbf{X} \sim N(\boldsymbol{\mu}, \Omega)$] can be written as $\mathbf{E}[\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2] = \sum_{j=1}^{p} \lambda_j E_j$, where $E_j = \mathbf{E}[\{(P\hat{\boldsymbol{\mu}})_j - (P\boldsymbol{\mu})_j\}^2]$ is the risk of the jth element of $P\hat{\boldsymbol{\mu}}$. Since $P\hat{\boldsymbol{\mu}} \sim N(P\boldsymbol{\mu}, I)$, E_j is the risk of the jth element of the mean of a multivariate normal distribution *with identity covariance matrix.*

Using the James–Stein estimate [of which (1) is a special case with $c = p - 2$]

$$\widehat{P\boldsymbol{\mu}} = \left(1 - \frac{c}{\mathbf{X}'P'P\mathbf{X}}\right) P\mathbf{X} \tag{85}$$

for $P\boldsymbol{\mu}$, it is not possible to obtain an analytic expression for E_j. However, expressions for $\sum_{j=1}^{p} E_j$ are easily computed [7], [24]. In particular, for any $\boldsymbol{\mu}$ on the circle of radius r (i.e., $\|\boldsymbol{\mu}\| = r$), $\sum_{j=1}^{p} E_j$ is a quadratic in c, with its minimum at $c = p-2$. Therefore, the choice (11) corresponds to minimizing $\sum_{j=1}^{p} E_j$ subject to the constraint that $c \leq 2(p^*-2)$. This constraint is a necessary and sufficient condition for the JSE (10) to dominate the MLE (see the proof of Theorem 1).

To see the relation between $\sum_{j=1}^{p} E_j$ and the true risk $\mathbf{E}[||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||^2] = \sum_{j=1}^{p} \lambda_j E_j$, we define the "normalized" risk of the JSE (10) as

$$\frac{\sum_{j=1}^{p} \lambda_j E_j}{\sum_{j=1}^{p} \lambda_j} \tag{86}$$

where the denominator $\sum_{j=1}^{p} \lambda_j$ is the risk of the MLE $\hat{\boldsymbol{\mu}} = \mathbf{X}$. Note that the normalized risk (86) is a convex combination of the E_j's. Around the ellipse[8] $||P\boldsymbol{\mu}|| = r$ for some constant r, we already mentioned that $\sum_{j=1}^{p} E_j$ is a quadratic in c. On the ellipse, the minimum and maximum of (86) lie below and above $(1/p)\sum_{j=1}^{p} E_j$, respectively. Our choice of c can therefore be viewed as minimizing some "central" risk $(1/p)\sum_{j=1}^{p} E_j$ subject to the constraint that the maximum risk never exceeds that of the MLE.

We remark that while it may be preferable to choose c to minimize the maximum risk (86) around any ellipse $||P\boldsymbol{\mu}|| = r$, the lack of analytic expressions for the E_j's makes the determination of such a c exceedingly difficult.

REFERENCES

[1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering.* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[2] M. E. Bock, "Minimax estimators of the mean of a multivariate normal distribution," *Ann. Stat.*, vol. 3, no. 1, pp. 209–218, 1975.

[3] B. Efron, *Introduction to James and Stein (1961) Estimation with Quadratic Loss.* New York: Springer-Verlag, 1992, vol. 1, pp. 437–442.

[4] B. Efron and C. Morris, "Stein's estimation rule and its competitors—An empirical Bayes approach," *J. Amer. Stat. Assoc.*, vol. 68, pp. 117–130, Mar. 1973.

[5] ———, "Data analysis using Stein's estimator and its generalizations," *J. Amer. Stat. Assoc.*, vol. 70, pp. 311–319, June 1975.

[6] ———, "Stein's paradox in statistics," *Sci. Amer.*, vol. 236, pp. 119–127, 1977.

[7] E. Greenberg and C. E. Webster, *Advanced Econometrics: A Bridge to the Literature.* New York: Wiley, 1983.

[8] M. J. Grimble, "Robust filter design for uncertain systems defined by both hard and soft bounds," *IEEE Trans. Signal Processing*, vol. 44, pp. 1063–1071, May 1996.

[9] Y. Y. Guo and N. Pal, "A sequence of improvements over the James–Stein estimator," *J. Multivariate Anal.*, vol. 42, pp. 302–317, 1992.

[10] S. Haykin. *Adaptive Signal Processing*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[11] W. James and C. M. Stein, "Estimation with quadratic loss," in *Proc. 4th Berkeley Symp. Math. Statist. Prob.*, 1961, vol. 1, pp. 311–319.

[12] N. S. Jayant and P. Noll, *Digital Coding of Waveforms.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[13] T. Kubokawa, "An approach to improving the James–Stein estimator," *J. Multivariate Anal.*, vol. 36, pp. 121–126, 1991.

[14] E. L. Lehmann, *Theory of Point Estimation.* New York: Wiley, 1983.

[15] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[16] A. M. Makowski, "Results on the filtering problem for linear systems with non-Gaussian initial conditions," in *Proc. 21st IEEE Conf. Decision Contr.*, Dec. 1982.

[17] A. M. Makowski, W. S. Levine, and M. Asher, "The nonlinear MMSE filter for partially observed systems driven by non-Gaussian white noise, with application to failure estimation," in *Proc. 23rd IEEE Conf. Decision Contr.*, Dec. 1984.

[18] C. J. Masreliez, "Approximate non-Gaussian filtering with linear state and observation relations," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 107–110, 1975.

[8] More precisely, the shape is only an ellipse if we assume P to be fixed.

[19] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York: Springer-Verlag, 1994.

[20] T. Soderstrom and P. Stoica, *System Identification.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[21] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, pp. 465–479, 1971.

[22] H. W. Sorenson and A. R. Stubberud, "Non-linear filtering by approximation of the *a posteriori* density," *Int. J. Contr.*, vol. 18, pp. 33–51, 1968.

[23] J. L. Speyer, C. Fan, and R. N. Banavar, "Optimal stochastic estimation with exponential cost criteria," in *Proc. 31st IEEE Conf. Decision Contr.*, Dec. 1992, pp. 2293–2298.

[24] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. Stat.*, vol. 9, no. 6, pp. 1135–1151, 1981.

[25] W. E. Strawderman, "Proper Bayes minimax estimators of the multivariate normal mean," *Ann. Math. Stat.*, vol. 42, pp. 385–388, 1971.

[26] Y. Theodor and U. Shaked, "Robust discrete-time minimum-variance filtering," *IEEE Trans. Signal Processing*, vol. 44, pp. 181–189, Feb. 1996.

[27] S. M. Tonissen and A. Logothetis, "Estimation of multiple target trajectories with time varying amplitudes," in *Proc. 8th IEEE Signal Process. Workshop Statist. Signal Array Process.*, June 1996, pp. 32–35.

[28] A. Ullah, "On the sampling distribution of improved estimators for coefficients in linear regression," *J. Econometr.*, vol. 2, pp. 143–150, 1974.

[29] W. R. Wu and A. Kundu, "Recursive filtering with non-Gaussian noises," *IEEE Trans. Signal Processing*, vol. 44, pp. 1454–1468, June 1996.

[30] L. Xie, Y. C. Soe, and C. E. de Souza, "Robust Kalman filtering for uncertain discrete-time systems," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1310–1314, 1994.

Jonathan H. Manton was born in Australia in 1973. He received the B.Sc. degree in mathematics and the B.Eng. degree in electrical engineering from the University of Melbourne, Parkville, Australia, in 1994. He recently submitted his Ph.D. dissertation in electrical engineering at the University of Melbourne.

He is currently a Research Fellow with the Department of Electrical Engineering, University of Melbourne. His research interests include estimation theory, stochastic filtering theory, consistency of estimators, stochastic convergence, and channel identification.

Vikram Krishnamurthy was born in India in 1966. He received the B.S. degree in electrical engineering from the University of Auckland, Auckland, New Zealand, in 1988 and the Ph.D. degree from the Department of Systems Engineering, Australian National University, Canberra, in 1992.

He is currently an Associate Professor with the Department of Electrical Engineering, University of Melbourne, Parkville, Australia. His research interests are in time-series analysis, hidden Markov models, stochastic filtering theory, and statistical signal processing.

H. Vincent Poor (F'87) received the Ph.D. degree in electrical engneering and computer science from Princeton University, Princeton, NJ, in 1977.

From 1977 until he joined the Princeton faculty in 1990, he was a Faculty Member at the University of Illinois, Urbana-Champaign. He also held visiting and summer appointments at several universities and research organizations in the United States, Britain, and Australia. His research interests are primarily in the area of statistical signal processing and its applications. His publications in this area include the graduate textbook *An Introduction to Signal Detection and Estimation* (New York: Springer-Verlag, 1988 and 1994).

Dr. Poor is a Fellow of the Acoustical Society of America and of the American Association for the Advancement for Science. He has been involved in a number of IEEE activities, including service as President of the IEEE Information Theory Society in 1990 and as a member of the IEEE Board of Directors in 1991 and 1992. In 1992, he received the Terman Award from the American Society for Engineering Education, and in 1994, he received the Distinguished Member Award from the IEEE Control Syetms Society.