

**SOLVING STOCHASTIC PROGRAMMING  
PROBLEMS VIA KALMAN FILTER AND  
AFFINE SCALING**

**Sarat Puthenpura and Lakshman Sinha  
AT&T Bell Laboratories  
Murray Hill, NJ 07974**

**Shu-Cherng Fang  
North Carolina State University  
Raleigh, NC 27695-7913**

**Romesh Saigal  
The University of Michigan  
Ann Arbor, Michigan 48109-2117**

**Technical Report 94-24**

**October 1994**

# Solving Stochastic Programming Problems via Kalman Filter and Affine Scaling

*Sarat Puthenpura & Lakshman Sinha*

AT&T Bell Laboratories  
Murray Hill, New Jersey 07974

*Shu-Cherng Fang*

North Carolina State University  
Raleigh, NC

*Romesh Saigal*

University of Michigan  
Ann Arbor, MI

Kalman filtering theory [1] has been around for more than two decades and has been the backbone of modern applied stochastic estimation. It is used extensively in many areas including engineering and econometrics. A Kalman filter estimates the *states* of a system from observations made about the stochastic inputs and outputs of the system. The Kalman filter performs its operations in an attractive recursive fashion, and can be easily implemented on a computer.

The Karmarkar's algorithm [2] is a relatively recent invention, which is an interior point algorithm to solve linear programming problems. It has been shown to be very effective in solving large linear programs. There are several variants of this algorithm, and the *primal affine scaling algorithm* [3] is a popular one. This algorithm has also been extended to solve convex programs with linear constraints [4]. A comprehensive treatment of various interior point algorithms has been the main focus of a recent book by Fang and Puthenpura [5].

In this paper, we first establish a parallelism between the Kalman filter and the affine scaling algorithm. This is done by appropriate modeling of the affine scaling algorithm in a *state space form* and identifying the corresponding elements in a Kalman filter framework. In this case, the stochastic elements of the filter are suppressed to emulate a deterministic system. These ideas have immediate extension to quadratic programming as well.

Here several interesting concurrences (both conceptual and computational) between the affine scaling algorithm and the Kalman filter are established. This enables us to derive a unified treatment of both the estimation and the optimization problems.

When we activate the stochastic elements of the Kalman filtering theory, we can effectively deal with stochastic linear programming problems. We obtain some theoretical results and a new Karmarkar-based algorithm is developed for this problem. This new approach appears to be very promising for applications in stochastic optimization and control of systems.

## 1. THE STATE SPACE REPRESENTATION OF LINEAR SYSTEMS

In this section we review some basics of the state space representation of systems, within the framework of Kalman filtering. For the sake of brevity we consider only the discrete time representation of systems (as opposed to continuous time representation) with single input and single output (SISO). However, the results can easily be extended to multivariable systems in either discrete or continuous time.

Consider a SISO system, with a given input sequence  $\{u(t)\}$  and output sequence  $\{y(t)\}$ , where  $t$ , sometimes referred to as the *discrete time index*, represents the sampling time. We say that the system is of order  $n$  if the sequences satisfy an  $n$ -th order difference equation of the form,

$$y(t) + a_1y(t-1) + a_2y(t-2) + \dots + a_ny(t-n) = b_0u(t) + b_1u(t-1) + \dots + b_mu(t-m) \quad (1)$$

where  $m \leq n$ . The state space form transforms the above  $n$ -th order equation into  $n$  first-order difference equations by introducing dummy state variables. In equation (1), we can define variables:  $z_1(t) = y(t)$ ,  $z_2(t) = y(t-1) = z_1(t-1)$ ,  $\dots$ ,  $z_n(t) = y(t-n+1) = z_{n-1}(t-1)$ . Thus we have

$$\mathbf{z}(t) = \mathbf{Fz}(t-1) + \mathbf{Gu}(t) \quad (2a)$$

and

$$\mathbf{y}(t) = \mathbf{Hz}(t) \quad (2b)$$

where

$$\mathbf{z}(t) = [z_1(t), z_2(t), \dots, z_n(t)]^T,$$

$$\mathbf{F} = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_n \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

$$\mathbf{u}(t) = [u_1(t), u_2(t), \dots, u_m(t)]^T,$$

$$\mathbf{G} = [b_0, b_1, \dots, b_m], \text{ and}$$

$$\mathbf{H} = [1, 0, \dots, 0].$$

Equations (2a) and (2b) are usually called the state transition equation and the output equation respectively. Also, matrix  $\mathbf{F}$  is commonly referred to as the state transition matrix. The eigenvalues of  $\mathbf{F}$  represent the natural frequencies of the system. The choice of the state variables  $\{z_i\}$  is not unique. Actually it is possible to choose proper states so that  $\mathbf{F}$  follows a suitable canonical form [6]. Moreover, the state variables can be chosen so that the states represent a physical aspect of the system under consideration. For example, in the case of a DC motor, the system can be conveniently modeled as a second-order system with two states representing the angular displacement and velocity of the motor shaft.

The vector  $\mathbf{u}(t)$  can be considered as a forcing function of the difference equation (1), hence (2a). If  $\mathbf{u}(t) = \mathbf{0}$  for all  $t$ , the system is called an *autonomous system*.

## 2. THE FILTERING PROBLEM

If we have access to the states of a system, by virtue of the relationship (2a) and (2b), the output of the system can be expressed as a function of  $u(t)$ 's and  $z(t)$ 's. This aspect is discussed extensively in the literature of modern control theory, for example [6]. However, the control of the system becomes more challenging if the states are not readily available. We then have to estimate the states from some measurable aspects of the system. This means samples of input and output of the system are needed. A device which estimates the states of a system from input output samples is commonly referred to as an *observer*. If the data used by the observer is assumed to be noise free, the observer becomes deterministic. On the other hand, if the input and/or the output data is noisy, one has to estimate or filter-out the values of the states from the noisy data given some knowledge of the statistical properties of the noise. Such an observer is a stochastic observer and Kalman filter is a good example of it.

The state space model of the Kalman filter can be obtained by extending the model depicted by equations (2a) and (2b). For a multivariable system, these are given as follows:

$$\mathbf{z}(t+1) = \mathbf{F}\mathbf{z}(t) + \mathbf{G}\mathbf{u}(t) + \mathbf{v}(t) \quad (3a)$$

and

$$\mathbf{y}(t) = \mathbf{H}\mathbf{z}(t) + \mathbf{e}(t) \quad (3b)$$

where

$\mathbf{z}(t) \in R^n$ ,  $\mathbf{F} \in R^{n \times n}$ ,  $\mathbf{G} \in R^{n \times m}$ ,  $\mathbf{H} \in R^{l \times n}$ ,  $\mathbf{u}(t) \in R^m$ ,  $\mathbf{v}(t) \in R^n$ ,  $\mathbf{y}(t) \in R^l$ , and  $\mathbf{e}(t) \in R^l$ .

The parameter  $\mathbf{v}(t)$  is the vector of system noise (or plant noise vector) and  $\mathbf{e}(t)$  is known as the measurement noise vector. We assume that  $\mathbf{e}(t)$  and  $\mathbf{v}(t)$  are from two independent Gaussian distributions with

$$E[\mathbf{v}(t)] = \mathbf{0} \quad (4a)$$

$$E[\mathbf{e}(t)] = \mathbf{0} \quad (4b)$$

$$E[\mathbf{v}(t)\mathbf{v}^T(t)] = \mathbf{Q} \quad (4c)$$

$$E[\mathbf{e}(t)\mathbf{e}^T(t)] = \mathbf{R} \quad (4d)$$

and

$$E[\mathbf{e}(t)\mathbf{v}^T(t)] = \mathbf{0} \quad (4e)$$

where  $\mathbf{Q}$  and  $\mathbf{R}$  are covariance matrices (often diagonal) of orders  $n \times n$  and  $l \times l$  respectively.

The filtering problem is to estimate the state vector  $\mathbf{z}(t+1)$  from the sigma algebra  $Y^t$  of observations up to time  $t$ .

If one estimates  $\mathbf{z}(t+1)$  by minimizing its variance, it is equivalent to solving the conditional expectation problem given by

$$\hat{\mathbf{z}}(t+1) = E[\mathbf{z}(t+1) | Y^t] \quad (5)$$

where  $\hat{\mathbf{z}}(t+1)$  denotes the expectation of  $\mathbf{z}(t+1)$ . The geometrical interpretation of the above operation is the projection of the observations onto the hyperplane defined by  $\mathbf{z}(t)$ 's.

A Kalman filter solves the above problem in a recursive fashion.

### 3. THE KALMAN FILTER

The essential idea behind a Kalman filter is to propagate the expected value of the state vector and its covariance matrix in a recursive fashion. Without going into the detailed derivation [7], the essential mathematical equations that govern the Kalman filter of system (3a), (3b) are given as follows:

$$\begin{aligned}\hat{\mathbf{z}}(t | t) &= \hat{\mathbf{z}}(t | t-1) + \mathbf{P}(t | t-1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(t | t-1)\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y}(t) - \mathbf{H}\hat{\mathbf{z}}(t | t-1)) \\ &= \hat{\mathbf{z}}(t | t-1) + \mathbf{K}(t | t-1)(\mathbf{y}(t) - \mathbf{H}\hat{\mathbf{z}}(t | t-1))\end{aligned}\quad (6a)$$

$$\begin{aligned}\mathbf{P}(t | t) &= \mathbf{P}(t | t-1) - \mathbf{P}(t | t-1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(t | t-1)\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}(t | t-1) \\ &= \mathbf{P}(t | t-1) - \mathbf{K}(t | t-1)\mathbf{H}\mathbf{P}(t | t-1)\end{aligned}\quad (6b)$$

where

$$\mathbf{K}(t | t-1) = \mathbf{P}(t | t-1)\mathbf{H}^T(\mathbf{H}\mathbf{P}(t | t-1)\mathbf{H}^T + \mathbf{R})^{-1}\quad (6c)$$

and finally,

$$\hat{\mathbf{z}}(t+1 | t) = \mathbf{F}\hat{\mathbf{z}}(t | t) + \mathbf{G}\mathbf{u}(t)\quad (6d)$$

$$\mathbf{P}(t+1 | t) = \mathbf{F}\mathbf{P}(t | t)\mathbf{F}^T + \mathbf{Q}\quad (6e)$$

The matrix  $\mathbf{P}$  is called the *covariance matrix* and matrix  $\mathbf{K}$  the *Kalman gain matrix*.

As described before,  $\hat{\mathbf{z}}(t+1 | t)$  is the estimate of  $\mathbf{z}(t+1)$  at the instant  $t$  (based on all the measurements made up to and including at time  $t$ ), which is depicted by equation (5). The matrix  $\mathbf{P}(t+1 | t)$  is the covariance matrix of the above estimate of states. From this, we can select an arbitrary vector  $\hat{\mathbf{z}}(1 | 0)$  to start the Kalman filter so long as the covariance matrix  $\mathbf{P}(1 | 0)$  is set very large (theoretically infinitely large). In a Bayesian framework, this amounts to *non-informative prior* on  $\hat{\mathbf{z}}(1 | 0)$ . In practice, one can conveniently choose  $\hat{\mathbf{z}}(1 | 0) = \mathbf{0}$  and  $\mathbf{P}(1 | 0) = \lambda \mathbf{I}$  with  $\lambda \in \mathbb{R}^+$  and  $\lambda \gg 0$  and  $\mathbf{I}$  is the identity matrix. It is also important to note that the Kalman filter can be applied to time varying systems in which the matrices  $\mathbf{F}$ ,  $\mathbf{G}$  and  $\mathbf{H}$  vary in time.

### 3.1 SIMPLIFIED FILTERING EQUATIONS FOR AUTONOMOUS SYSTEMS

In this section the Kalman filtering equations are simplified for autonomous systems, which will be directly applicable to our forthcoming discussion. To begin with, we introduce the

following notation.

$$\hat{z}(t+1 | t) = \hat{z}(t+1) ; \hat{z}(t | t-1) = \hat{z}(t) \quad \text{and} \quad P(t+1 | t) = P(t+1) ; P(t | t-1) = P(t).$$

Now we state our results as follows:

### Proposition 1

For an autonomous system, the Kalman filtering equations are,

$$\hat{z}(t+1) = F\hat{z}(t) + K(t)(y(t) - H\hat{z}(t)) \quad (7a)$$

$$K(t) = FP(t)H^T(HP(t)H^T + R)^{-1} \quad (7b)$$

and

$$P(t+1) = FP(t)F^T + Q - K(t)HP(t)F^T \quad (7c)$$

**Proof:**

Since for an autonomous system,  $u(t) = 0$  for all  $t$ , it follows from (6d) that  $\hat{z}(t+1) = F\hat{z}(t)$  for all  $t$ . This property, in conjunction with equations (6a) and (6b), and some simple algebraic simplifications yields the desired results.

□

Now let us focus on affine scaling algorithm for linear programming problems.

## 4. THE LINEAR PROGRAMMING PROBLEM AND THE AFFINE SCALING ALGORITHM

The linear programming problem can be stated as follows in the standard form:

$$\text{minimize } c^T x \quad (8a)$$

$$\text{such that } Ax = b \quad \text{and} \quad x \geq 0, \quad (8b)$$

where



- A** is an  $m \times n$  matrix called the constraint matrix,
- b** is an  $m$  vector called the right-hand-side,
- c** is an  $n$  vector called the cost vector,
- x** is an  $n$  vector of variables.

The affine scaling algorithm [3] is summarized below.

Assume  $\mathbf{x}^0$  is an interior feasible point to the problem (8a) and (8b). We let  $\mathbf{D}_x$  be the diagonal matrix containing the components of  $\mathbf{x}$ . Then the translation vector ( $\mathbf{c}_p$ ) from  $\mathbf{x}^0$  to a new feasible point  $\mathbf{x}^1$  is given by

$$\mathbf{c}_p = -\frac{\alpha}{\gamma} \mathbf{D}_x^2 \mathbf{r} \quad (9a)$$

where

$$\mathbf{r} = (\mathbf{c} - \mathbf{A}^T \mathbf{w}), \quad (9b)$$

$$\mathbf{w} = (\mathbf{A} \mathbf{D}_x^2 \mathbf{A}^T)^{-1} \mathbf{A} \mathbf{D}_x^2 \mathbf{c}, \quad (9c)$$

$$\gamma = \max_i (\mathbf{e}_i \cdot \mathbf{P}_x \mathbf{D}_x \mathbf{c}) \quad (9d)$$

here  $\mathbf{e}_i$  is  $i$ -th unit vector and  $\mathbf{P}_x$  is projection matrix given by

$$\mathbf{P}_x = \mathbf{I} - \mathbf{D}_x \mathbf{A}^T (\mathbf{A} \mathbf{D}_x^2 \mathbf{A}^T)^{-1} \mathbf{A} \mathbf{D}_x \quad (10)$$

The parameter  $\alpha$  is a scalar between 0 and 1. Vector  $\mathbf{r}$  is called the vector of *reduced costs*, and vector  $\mathbf{w}$  the vector of *dual variables*.

The new interior feasible point  $\mathbf{x}^1$  is obtained by

$$\mathbf{x}^1 = \mathbf{x}^0 - \frac{\alpha}{\gamma} \mathbf{D}_x^2 \mathbf{r} \quad (11)$$

The affine scaling algorithm essentially transforms the LP problem by performing the affine mapping (scaling of the affine space defined by equations (8a) and (8b) with respect to the current feasible solution  $\mathbf{x}^0$ )

$$\mathbf{x}' = \mathbf{D}_x^{-1} \mathbf{x} \quad (12)$$

and obtains the direction of descent by projecting the gradient of the transformed cost function ( $\mathbf{D}_{x_0}$ ) on to the null space of the transformed constraints set ( $\mathbf{AD}_x \mathbf{o}$ ). The new solution  $\mathbf{x}^1$  is then obtained by translating the current solution (in the transformed space) along this direction, and then mapping the result back into the original space.

Now an interesting observation on the affine scaling algorithm can be made.

### Proposition 2

The dual solution in equation (9c) is  $\arg \min_{\mathbf{w}} \|\mathbf{(AD}_x)^T \mathbf{w} - \mathbf{D}_x \mathbf{c}\|^2$ .

**Proof:**

Let  $V = \|\mathbf{(AD}_x)^T \mathbf{w} - \mathbf{D}_x \mathbf{c}\|^2$

Consequently,

$$V = [(\mathbf{AD}_x)^T \mathbf{w} - \mathbf{D}_x \mathbf{c}]^T [(\mathbf{AD}_x)^T \mathbf{w} - \mathbf{D}_x \mathbf{c}] = \mathbf{w}^T \mathbf{AD}_x^2 \mathbf{A}^T \mathbf{w} - 2\mathbf{w}^T \mathbf{AD}_x^2 \mathbf{c} + \mathbf{D}_x^2 \mathbf{c}^T \mathbf{c}$$

Partially differentiating  $V$  with respect to  $\mathbf{w}$  and setting it to zero yields the stated result. □

## 5. THE AFFINE SCALING METHOD IN A FILTERING FRAMEWORK

Based on the observation by Puthenpura and Sinha [9], we now derive the connection between the Kalman filter and the affine scaling algorithm, both conceptually and computationally.

**Theorem 1.**

The Kalman filter minimizes

$$E[\mathbf{e}'(t)^T \mathbf{e}'(t) \mid Y^t] \quad (13a)$$

where  $\mathbf{e}'(t)$  is called the *filtered state error vector* defined by

$$\mathbf{e}'(t) = \mathbf{z}(t) - \hat{\mathbf{z}}(t) \quad (13b)$$

and  $\hat{\mathbf{z}}(t)$  is the state estimated by the Kalman filter. Moreover, when the filter converges,

$$E[\boldsymbol{\eta}(t) | Y^t] = \mathbf{0} \quad (14a)$$

$$E[\boldsymbol{\eta}(t)\mathbf{y}^T(t)] = \mathbf{0} \quad (14b)$$

$$E[\mathbf{e}'(t)\boldsymbol{\eta}^T(t)] = \mathbf{0} \quad (14c)$$

$$E[\mathbf{e}'(t) | Y^t] = \mathbf{0} \quad (14d)$$

where  $\boldsymbol{\eta}(t) = \mathbf{y}(t) - \mathbf{H}\hat{\mathbf{z}}(t)$  is called the *innovations* or *residual* vector.

**Proof:**

See [7] and [8] for a complete proof of these results.

□

One important note should be made here:

Conditions (14a) to (14d) imply that the filter converges when the estimates and the residuals become statistically orthogonal to (statistically independent of) the stochastics of the system. In this case, no component of the statistical information in the stochastics of the system contributes to the improvement of the estimates of the states.

Theorem 1 indicates that the states estimated by the Kalman filter,  $\hat{\mathbf{z}}(t)$ , is a *minimum variance estimator* of  $\mathbf{z}(t)$ . At the same time, Proposition 2 shows that the dual vector  $\mathbf{w}$  of the affine scaling algorithm is a least squares solution. Thus, the underlying computation for both cases is the minimization of deviations measured by the  $l_2$  norm. This is the key to the parallelism between the affine scaling algorithm and Kalman filter.

We now introduce a state space model to exam the affine scaling algorithm (depicted by equations (9a) through (11)) in a Kalman filtering framework.

**Theorem 2**

The state space equations for the affine scaling algorithm are

$$\mathbf{w}(t+1) = \mathbf{w}(t) \quad (15a)$$

and

$$\mathbf{c}_k(t) = \mathbf{A}_k(t)^T \mathbf{w}(t) + \mathbf{e}(t) \quad (15b)$$

where  $\mathbf{c}_k(t) = \mathbf{D}_k \mathbf{c}$  and  $\mathbf{A}_k(t) = \mathbf{A} \mathbf{D}_k$ ;  $\mathbf{D}_k$  being a diagonal matrix with diagonal elements being components of  $\mathbf{x}^k$ .

**Proof:**

At any step  $k$  of the affine scaling algorithm, from Equation (9c), we see the dual vector is unique provided the matrix  $\mathbf{A}$  has full row rank. Therefore, if we use the Kalman filter to estimate the dual corresponding to the current interior point, it has to converge to this unique dual vector  $\mathbf{w}^k$ . That is, at some sufficiently high value of  $t$ , we expect  $\mathbf{w}(t+1) = \mathbf{w}(t) = \mathbf{w}^k$ . This justifies (15a).

Now, from equation (11), we have

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \beta \mathbf{D}_k^2 (\mathbf{c} - \mathbf{A}^T \mathbf{w}^k) \quad (16a)$$

where  $\beta = \frac{\alpha}{\gamma}$ .

Rearranging terms results in

$$\mathbf{D}_k \mathbf{c} = \mathbf{D}_k \mathbf{A}^T \mathbf{w}^k - \frac{1}{\beta} \mathbf{D}_k^{-1} [\mathbf{x}^{k+1} - \mathbf{x}^k] \quad (16b)$$

Adding the discrete time index  $t$  to the above equation (to emulate the filtering environment),

we get Equation (15b).

□

Note that Equation (15b) depicts the  $t^{\text{th}}$  instant of the filtering iteration at the  $k^{\text{th}}$  affine scaling iteration.

The following result is a direct consequence of Theorem 2:

### Proposition 3

The affine scaling iteration is a special case of applying Kalman filter to an autonomous system with

$$\mathbf{F} = \mathbf{I}$$

$$\mathbf{H} = \mathbf{A}^T \mathbf{D}_k$$

$$\mathbf{y}(t) = \mathbf{D}_k \mathbf{c}$$

$$\mathbf{Q} = \mathbf{R} = \varepsilon \mathbf{I}$$

$$\varepsilon \rightarrow 0.$$

so that the state estimates  $\mathbf{z}(t)$  converges to the dual vector  $\mathbf{w}^k$  corresponding to the current interior point  $\mathbf{x}^k$ , when the filter converges.

Combining Propositions 1 and Proposition 4, one can immediately set up the Kalman filter equations corresponding to the affine scaling algorithm.

To render further clarity, a step-by-step implementation procedure on emulating affine scaling iteration via Kalman filter is given below.

1. Initialization of affine scaling iterations:

Find  $\mathbf{x}^0$  such that  $\mathbf{A}\mathbf{x}^0 = \mathbf{b}$ ,  $\mathbf{x}^0 > \mathbf{0}$ . Select  $0 < \alpha < 1$ .

(One may use a Phase 1 LP here, which itself could employ the Kalman filter approach)

2. Kalman filter iterations to obtain the dual vector:

(a) Set  $t = 0$ ;  $\mathbf{w}(0) = \mathbf{0}$ ;  $\mathbf{P}(0) = \lambda\mathbf{I}$  such that  $\lambda \gg 0$ . Also set

$\mathbf{H} = \mathbf{A}^T\mathbf{D}_k$ ;  $\mathbf{y}(t) = \mathbf{D}_k\mathbf{c}$  for all  $t$ ,  $\mathbf{Q} = \mathbf{R} = \epsilon\mathbf{I}$  such that  $\mathbf{D}_k$  is a diagonal matrix with diagonal elements being the components of  $\mathbf{x}^k$  and  $\epsilon \rightarrow 0$ .

(b) Perform the updates

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mathbf{K}(t)[\mathbf{y}(t) - \mathbf{H}\mathbf{w}(t)]$$

$$\mathbf{P}(t+1) = \mathbf{P}(t) + \mathbf{Q} - \mathbf{K}(t)\mathbf{H}\mathbf{P}(t)$$

where

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}^T[\mathbf{H}\mathbf{P}(t)\mathbf{H}^T + \mathbf{R}]^{-1}$$

and set

$$t \leftarrow t+1$$

until  $\|\mathbf{P}(t)\| \leq \phi$  where  $\phi$  is arbitrarily small, for some  $t = T$ .

(c) Set  $\mathbf{w}^k = \mathbf{w}(T)$ .

3. Affine scaling update to obtain the primal vector:

Set

$$\mathbf{r}^{k+1} = \mathbf{c} - \mathbf{A}^T\mathbf{w}^k; \quad \mathbf{d}^k = -\mathbf{D}_k\mathbf{r}^k$$

$$\gamma^k = \min_i \left[ \frac{\alpha}{-d_i^k} \mid d_i^k < 0 \right] \quad \mathbf{x}^{k+1} = \mathbf{x}^k + \gamma^k\mathbf{D}_k\mathbf{d}^k$$

4. Test for optimality of  $\mathbf{x}^{k+1}$  (KKT conditions). If optimality conditions are not satisfied, set  $k \leftarrow k+1$  and go back to Step 2.

Note that the main computational burden at each affine scaling iteration is the evaluation of the matrix inverse  $(\mathbf{A}\mathbf{D}_x^2\mathbf{A}^T)^{-1}$  which is an  $O(m^3)$  operation given  $\mathbf{A}$  is a  $m \times n$  matrix. Now if we invoke KFS to compute the dual vector (see Step 2 of the above step-by-step implementation procedure), then the bulk of the computation is to evaluate the Kalman gain

matrix  $\mathbf{K}(t)$ , specifically the computation of  $[\mathbf{HP}(t)\mathbf{H}^T + \mathbf{R}]^{-1}$  which is an  $O(n^3)$  operation. Note that this inverse operation is to be carried out several times within an AFS iteration. Moreover, the matrix  $\mathbf{P}(t)$  is dense so that sparse matrix techniques cannot be employed to reduce the computational burden. Therefore, for deterministic case, invoking KFS tends to be a rather expensive way to obtain the dual vector. However, as will be seen later, the real usefulness of this approach will be for the stochastic case where the dual vector will be filtered out by the KFS from noisy (stochastic) environment. Before that, let us look at the correspondence between Kalman filter and the affine scaling algorithm in a detailed manner, which will be useful in subsequent discussions.

## 6. CORRESPONDENCE BETWEEN KALMAN FILTER AND AFFINE SCALING ALGORITHM

The conceptual and computational parallelism between the the Kalman filter and affine scaling are summarized in this section. For brevity we denote the Kalman filter as KFS and the affine scaling algorithm as AFS.

### 6.1 ENVIRONMENT OF OPERATION

KFS operates in a stochastic environment, while AFS works essentially in a deterministic framework.

### 6.2 MEASUREMENTS (OBSERVATIONS)

KFS estimates the states from the measurements of the noisy output of the system i.e.,  $\mathbf{y}(t)$ . By virtue of Proposition 3, AFS derives the information from the gradient of the cost in the transformed space, i.e.,  $\mathbf{D}_x \mathbf{c}$ . This is quite true, as  $\mathbf{c}$  is a constant the only variable is  $\mathbf{D}_x$  or  $\mathbf{x}$  itself, which essentially contains the entire information.

### 6.3 OPTIMAL ESTIMATES

KFS estimates the states  $z$ , of a system in the sense of minimum variance from the observations. AFS basically computes the dual vector  $w$ , in the sense of least squares, from the observation of the current interior feasible point.

### 6.4 SOURCES OF INFORMATION

Both KFS and AFS can be considered as a predictor-corrector mechanism. That is, start at some point and move towards optimality in steps by predicting the next point from the information that is currently available. As far as KFS is concerned, the next estimate of states is based on the "residuals (innovations)" defined as

$$s(t) = y(t) - H\hat{z}(t) \quad (17a)$$

For AFS, the next direction of move is based on "reduced cost vector" ( $r_k(t)$ ), that is defined in the transformed space by the equation

$$r_k(t) = c_k(t) - A_k^T(t)w(t) \quad (17b)$$

where  $c_k(t)$  and  $A_k(t)$  are defined in Equation (15b).

### 6.5 OPTIMALITY CRITERIA

For KFS, the optimality conditions are given by the zero mean of residuals (equation (14a)) and the statistical orthogonality (statistical independence) between the residuals vector and the observation vector (equation (14b)).

Likewise, for AFS optimality occurs when the reduced cost are non-negative (dual feasibility) and the reduced costs vector and the current interior point vector becomes geometrically orthogonal (complementary slackness).



By now we see that the affine scaling algorithm is equivalent to the Kalman filter working in a deterministic environment. In other words, one can emulate the affine scaling algorithm by the Kalman filter by shutting off stochastic elements. Now the question is, why can't we make use of the inherent power of the Kalman filter to deal with stochastics, to solve stochastic programming problems. After all, the Kalman filter has been designed to deal with stochastic systems. This idea leads to our accomplishment of developing a Karmarkar-based algorithm for stochastic linear programs.

## 7. KARMARKAR-BASED ALGORITHM FOR STOCHASTIC LINEAR PROGRAMS

In this section we introduce stochastic parameters into the linear program and focus on two classes of problems. The first consists stochastic cost vector and deterministic constraints, and the second class of has both the cost vector and the constraint matrix stochastic.

### 7.1 STOCHASTIC PROGRAMMING WITH DETERMINISTIC CONSTRAINTS AND STOCHASTIC COST VECTOR

The problem can be stated mathematically as follows:

$$\text{minimize } [c_0 + \delta c(t)]^T x \quad (18a)$$

$$\text{such that } Ax = b \quad \text{and} \quad x \geq 0, \quad (18b)$$

where  $\delta c(t)$  is assumed to have zero mean Gaussian noise with a covariance matrix  $\Sigma_c$  and  $c_0$  is the unknown "true" value of the cost vector.

Note that in this case, the projection matrix which projects on to the null space of the constraint set is deterministic, as  $A$  and  $b$  are deterministic. However the projected gradient of the cost and hence the dual estimates at each step of the iteration are stochastic.

**Proposition 4**

The state space equations at the  $k^{\text{th}}$  affine scaling iteration, corresponding to the case with only the cost vector stochastic, are:

$$\mathbf{w}(t+1) = \mathbf{w}(t) \tag{19a}$$

and

$$\mathbf{c}_k(t) = \mathbf{A}_k(t)^T \mathbf{w}(t) + \mathbf{e}(t) \tag{19b}$$

where  $\mathbf{c}_k(t) = \mathbf{D}_k \mathbf{c}(t)$ ,  $\mathbf{A}_k(t) = \mathbf{A} \mathbf{D}_k$ , and  $\mathbf{D}_k$  is diagonal matrix whose diagonal elements are components of  $\mathbf{x}^k$ .

**Proof:**

When  $\mathbf{A}$  and  $\mathbf{b}$  are deterministic, the projection operator onto the hyperplane containing the rows of  $\mathbf{A} \mathbf{D}_k$  is unique. This implies (19a), provided  $\mathbf{A}$  is of full rank. Also note that  $\mathbf{c}_k(t)$  is the stochastic measurement (as  $\mathbf{c}$  is stochastic) which implies  $\mathbf{e}(t)$  is the noise to be filtered using the output equation (19b).

□

Now assume that we are at the  $k^{\text{th}}$  affine scaling iteration. We Set

(a)  $\mathbf{P}(0) = \lambda \mathbf{I}$  where  $\lambda \gg 0$ ,

(b)  $\mathbf{w}(0)$  arbitrary (a good choice would be the dual vector of the previous iteration).

We set

(c)  $\mathbf{Q} = \epsilon \mathbf{I}$ , where  $\epsilon \rightarrow 0$ , and

(d)  $\mathbf{R} = \Sigma_c$

in the Kalman filtering framework as described in Proposition 1. Then we run the filter until the optimality criteria of Theorem 1 are satisfied.

Assume that  $w(T)$  and  $P(T)$  are the estimate of the dual vector and its covariance matrix respectively, when the filter converges. We take  $w(T)$  as the value of the dual vector for the  $k^{\text{th}}$  affine scaling iteration, and translate  $\mathbf{x}^k$  to  $\mathbf{x}^{k+1}$  via Equation (11). While doing so, one could use an estimate of the expected value of  $c$  in Equation (9b). A good choice is to use the mean value of the known samples, i.e.,

$$\bar{c}(M) = \bar{c}(M-1) - \frac{1}{M}[\bar{c}(M-1) - c(M)] \quad (20)$$

where  $\bar{c}(M)$  is the mean up to the  $M^{\text{th}}$  sample and  $M$  is the number of samples obtained up to time  $t$ .

## 7.2 STOCHASTIC PROGRAMMING WITH STOCHASTIC CONSTRAINT MATRIX AND STOCHASTIC COST VECTOR

This problem can be stated mathematically as follows:

$$\text{minimize } [\mathbf{c}_0 + \delta\mathbf{c}(t)]^T \mathbf{x} \quad (21a)$$

$$\text{such that } [\mathbf{A}_0 + \delta\mathbf{A}(t)]\mathbf{x} = \mathbf{b} \quad \text{and} \quad \mathbf{x} \geq \mathbf{0}, \quad (21b)$$

where  $\delta\mathbf{c}(t)$  is Gaussian noise with zero mean and covariance matrix  $\Sigma_c$ , and  $\delta\mathbf{A}(t)$  is Gaussian with zero mean and covariance matrix  $\Sigma_A$ . We also assume that  $\mathbf{b}$  is deterministic and that the system (21b) is feasible for all outcomes  $\delta\mathbf{A}(t)$ . We show later how the infeasibility of (21b) can be handled.

### Proposition 5

The state space equations at the  $k^{\text{th}}$  affine scaling iteration, in the case that both the cost vector

and the constraint matrix are stochastic, are:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mathbf{v}(t) \quad (22a)$$

and

$$\mathbf{c}_k(t) = \mathbf{A}_k(t)^T \mathbf{w}(t) + \mathbf{e}(t) \quad (22b)$$

where  $\mathbf{c}_k(t) = \mathbf{D}_k \mathbf{c}(t)$  and  $\mathbf{A}_k(t) = \mathbf{A}(t) \mathbf{D}_k$ .

**Proof:**

When  $\mathbf{A}$  is stochastic, the projection onto to the space containing the rows of  $\mathbf{A}(t) \mathbf{D}_k$  is not unique. Hence, the dual vector  $\mathbf{w}$  has some inherent uncertainty or system noise and it is impossible to completely filter out this noise from the dual estimates. As a matter of fact, the covariance matrix of the dual estimate does not vanish, but asymptotically reaches the Cramer-Rao lower bound [12]. Correspondingly, we can model the dynamics of the system corresponding to the dual vector as described in (22a). Equation (22b) can be obtained in the same manner as before.

□

In this case we can set up the Kalman filter as follows:

At  $k^{\text{th}}$  affine scaling iteration, we set

(a)  $\mathbf{P}(0) = \lambda \mathbf{I}$  where  $\lambda \gg 0$ ,

(b)  $\mathbf{w}(0)$  arbitrary [a good choice is the dual vector at the previous affine scaling iteration ( $k-1$ )],

(c)  $\mathbf{Q} = \Sigma_A$

(Since  $\mathbf{A}(t)$  contributes to  $\mathbf{v}(t)$ )

$$(d) \mathbf{R} = \Sigma_A \tau + \Sigma_c$$

(due to the fact that  $\mathbf{A}(t)$  and  $\mathbf{c}(t)$  contribute to  $\mathbf{e}(t)$ )

in the Kalman filtering framework as described in Proposition 1. Then we run the filter and stop it when the optimality criteria of Theorem 1 are satisfied. Assume that  $\mathbf{w}(T)$  and  $\mathbf{P}(T)$  is the value for the dual vector estimate and its covariance matrix respectively, when the filter converges. We shall take  $\mathbf{w}(T)$  as the value of the dual vector for the  $k^{th}$  affine scaling iteration.

### 7.3 INFEASIBILITY OF (21b)

In order to avoid the infeasibility of (21b), we only consider those samples of  $\mathbf{A}$  which makes the set

$$\{\mathbf{x} \mid [\mathbf{A}^0 + \delta\mathbf{A}(t)]\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

feasible. A simple Phase-1 step can be quickly detect the case that a sample of  $\mathbf{A}$  leads to an infeasible problem. By discarding such samples in the Kalman filter, we have the minimization problem (21a) conditioned on feasible samples of  $\mathbf{A}$ .

## 8. ERROR ANALYSIS

In this section we look into an analysis of the error propagation properties of the proposed algorithm, and assume that (21b) is feasible for all values of  $\mathbf{A}$  in the sample space.

### Theorem 3

Assume that we are at an interior feasible point  $\mathbf{x}^k$ , and the algorithm generates a new interior feasible point  $\mathbf{x}^{k+1}$  for the interior feasible point. If  $\mathbf{A}$  is contaminated by a Gaussian noise with zero mean and covariance matrix  $\Sigma_A$  and  $\mathbf{c}$  is contaminated by a Gaussian noise with zero mean and covariance matrix  $\Sigma_c$ , then we have

$$E[\mathbf{x}^{k+1} | Y^t] = \mathbf{x}_0^{k+1} \quad (23)$$

and

$$E[(\tilde{\mathbf{x}}^{k+1})(\tilde{\mathbf{x}}^{k+1})^T | Y^t] \leq \beta^2 \mathbf{D}_k^2 \left[ \Sigma_c + \mathbf{A}_0^T \mathbf{P}_\infty \mathbf{A}_0 + \sigma \Sigma_A \right] \mathbf{D}_k^2 \quad (24)$$

where  $\mathbf{x}_0^{k+1}$  is the true solution vector,  $Y^t$  is the sigma algebra of observations,  $\tilde{\mathbf{x}}^{k+1} = \mathbf{x}^{k+1} - \mathbf{x}_0^{k+1}$ ,  $\mathbf{A}_0$  is the true value of  $\mathbf{A}$  ( which in practice could be replaced by  $E[\mathbf{A}(t)]$  ),  $\mathbf{P}_\infty$  is the steady state covariance matrix of the dual estimate, and  $0 \leq \sigma < \infty$ . The scalar  $\beta$  is a positive constant which assures that  $\mathbf{x}^{k+1}$  will lie strictly within the polyhedral set defined by the constraints.

**Proof:**

Let  $\mathbf{w}(t)$  be the estimate of the dual obtained by the filtering process,  $\mathbf{A}_0$  and  $\mathbf{c}_0$  be the "true" values of  $\mathbf{A}$  and  $\mathbf{c}$ . Given the history  $Y^t$  of observations, we define the "true" interior (primal) solution as

$$\mathbf{x}_0^{k+1} = \mathbf{x}^k - \beta \mathbf{D}_k^2 \left[ \mathbf{c}_0 - (\mathbf{A}_0)^T \mathbf{w}_0^k \right] \quad (25)$$

where  $\mathbf{w}_0^k$  is the 'true' dual solution at the interior point  $\mathbf{x}^k$ . Also, the "stochastic" interior point is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \beta \mathbf{D}_k^2 \left[ \mathbf{c}(t) - (\mathbf{A}(t))^T \mathbf{w}(t) \right] \quad (26)$$

Subtracting (24) from (25) and substituting  $\tilde{\mathbf{x}}(k+1)$ ,  $\tilde{\mathbf{c}}(t)$ ,  $\tilde{\mathbf{A}}(t)$  and  $\tilde{\mathbf{w}}(t)$  for  $\mathbf{x}(k+1) - \mathbf{x}_0^{k+1}$ ,  $\mathbf{c}(t) - \mathbf{c}_0$ ,  $\mathbf{A}(t) - \mathbf{A}_0$ , and  $\mathbf{w}(t) - \mathbf{w}_0^k$  respectively, we obtain

$$\tilde{\mathbf{x}}^{k+1} = -\beta \mathbf{D}_k^2 \left[ \tilde{\mathbf{c}}(t) - (\mathbf{A}_0)^T \tilde{\mathbf{w}}(t) - (\tilde{\mathbf{A}}(t))^T \mathbf{w}_0^k - (\tilde{\mathbf{A}}(t))^T \tilde{\mathbf{w}}(t) \right] \quad (27)$$

It can be readily confirmed that  $E[\tilde{\mathbf{x}}(t+1) | Y^t] = \mathbf{0}$ . (Note that the conditioning on  $Y^t$  renders  $\mathbf{x}(t)$  a constant). Hence the first part of the theorem follows. Also

$$\begin{aligned} E \left[ (\tilde{\mathbf{x}}^{k+1} (\tilde{\mathbf{x}}^{k+1})^T \mid Y^t \right] &= \beta^2 \mathbf{D}_k^2 \left[ E \left[ \tilde{\mathbf{c}}(t) (\tilde{\mathbf{c}}(t))^T \mid Y^t \right] + E \left[ \mathbf{A}_0^T \tilde{\mathbf{w}}(t) (\tilde{\mathbf{w}}(t))^T \mathbf{A}_0 \mid Y^t \right] \right] \mathbf{D}_k^2 + \\ &\quad \beta^2 \mathbf{D}_k^2 \left[ E \left[ (\tilde{\mathbf{A}}(t))^T \mathbf{w}_0^k (\mathbf{w}_0^k)^T \tilde{\mathbf{A}}(t) \mid Y^t \right] \right] \mathbf{D}_k^2 \end{aligned} \quad (28)$$

Since  $\mathbf{w}(t)$  is the dual estimate obtained by Kalman filter, and the filtering process has stabilized, all the stochastics of the system become "statistically orthogonal" to the error in estimate. Hence the other terms like  $\tilde{\mathbf{c}}(t) \tilde{\mathbf{w}}(t)^T \tilde{\mathbf{A}}(t)$ ,  $\tilde{\mathbf{c}}(t) (\mathbf{w}_0^k)^T \tilde{\mathbf{A}}(t)$  etc. in the calculation of  $E[\tilde{\mathbf{x}}^{k+1} (\tilde{\mathbf{x}}^{k+1})^T]$  from (27), have zero expectations. Also, the Kalman filter gives  $E[\tilde{\mathbf{w}}(t) (\tilde{\mathbf{w}}(t))^T]$  as the matrix  $\mathbf{P}_\infty$ . Now,

$$E \left[ (\tilde{\mathbf{A}}(t))^T \mathbf{w}_0^k (\mathbf{w}_0^k)^T \tilde{\mathbf{A}}(t) \mid Y^t \right] \leq \sigma E \left[ (\tilde{\mathbf{A}}(t))^T \tilde{\mathbf{A}}(t) \mid Y^t \right] = \sigma \Sigma_A \quad (28a)$$

where  $\sigma = \sup \left\{ \rho \left[ \mathbf{w}_0^k (\mathbf{w}_0^k)^T \right] \right\}$ ;  $\rho(\cdot)$  denotes the matrix spectral radius. Clearly  $\sigma \geq 0$ . Now, given the primal LP is feasible, the duals are bounded, hence  $\sigma < \infty$ . Thus we have (24).

□

### Corollary 1

For the case where only  $\mathbf{c}$  is stochastic, we have

$$E[\tilde{\mathbf{x}}^{k+1} \mid Y^t] = \mathbf{x}_0^{k+1}$$

and

$$E[(\tilde{\mathbf{x}}^{k+1} (\tilde{\mathbf{x}}^{k+1})^T) \mid Y^t] = \beta^2 \left[ \mathbf{D}_k^2 \Sigma_c \mathbf{D}_k^2 \right] \quad (29)$$

**Proof:**

Since only  $\mathbf{c}$  is stochastic, clearly  $\Sigma_A = \mathbf{0}$  and the error in the state equation  $\mathbf{v}(t) = \mathbf{0}$ , and from a standard result,  $\mathbf{P}_\infty = \mathbf{0}$ . Hence (29) follows from (24).

□

Notice that Theorem 3 and Corollary 1 indicate that the proposed algorithm generates an

unbiased estimator with controlled variance.

The key thing to note here, that the step length  $\beta$  can be adjusted to control the variance. This is intuitively true. In a stochastic environment, taking small step lengths (that is, making "cautious translations") is expected to reduce the uncertainty in estimates. However, the price we pay here is the slow convergence (that is, sacrificing efficiency). More studies need to be done in striking a trade off between the reduction of variance and improving the efficiency of the algorithm.

When the algorithm is converging to the solution (i.e.  $x$  is converging to the solution), by considering the limiting behavior, the trajectory, starting at  $x^k$  under the stochastic case, and the trajectory starting at  $x^k$  but determined by the true values  $A$  and/or  $c$ , are the closest in the minimum variance sense, dictated by the Cramer-Rao lower bound. Thus, the estimate of the conditional variance given by (24) is a reasonably good estimate of the variance of the solution generated, at least for the stochastic cost vector only case. In the other case, only a bound on the variance can be established. Our preliminary numerical experiments also support the above claim. However, we caution that the conditional variance ( given by (24) ) must not be used to predict the variance of the solution in the earlier iterations. Further research work needs to be done in estimating the variance of  $x^k$  also recursively in a computationally efficient manner.

## 9. CONCLUSIONS

We have shown how the Karmarkar algorithm in conjunction with Kalman filter can be used to solve stochastic programming problems. As it stands now, with the proposed approach, one can solve stochastic linear programming problems with stochastic cost vector and/or stochastic constraint matrix. To our knowledge, this is the first interior-point method that solves linear



programs in a stochastic environment. The proposed approach actually differs from the conventional approaches studied in [10] to [13], since these approaches rely on the knowledge of the distribution functions of the stochastic elements while our approach only depends upon a finite number of available samples. This distinction renders us the capability to perform "on-line" analysis or "on-line" monitoring of stochastic linear programs. For the problem with non-Gaussian noise, the idea of "robust Kalman filtering" [14] and "bootstrap algorithm" [15] can be incorporated in our approach.

More work needs to be done to extend these results to the cases involving stochastic right hand side vector along with stochastic constraint matrix as well as stochastic cost vector. Besides, means have to be explored to enhance the computational efficiency of the proposed method. These aspects are under investigation by the authors.

#### **Acknowledgement**

We thank the anonymous reviewer, whose comments were very useful to improve the original version of this paper.

**REFERENCES**

- [1] Kalman, R. E., "A new Approach to Linear Filtering and Prediction problems", Trans. ASME J. Basic Engineering, Vol. 82, pp.35-45,1960.
- [2] Karmarkar, N. K., "A New Polynomial Time Algorithm for Linear Programming", *Combinatorica*, Vol.4, No.4, pp.373-391, 1984.
- [3] Vanderbei, R., Meketon, M.S., and Freedman, B.A., "A Modification of the Karmarkar Linear Programming g Algorithm", *Algorithmica*, pp.395-407, 1986.
- [4] Freedman, B.A., Puthenpura, S.C., and Sinha, L.P., "A New Karmarkar-Based Algorithm for Optimizing Convex, Non-Linear Cost Functions with Linear Constraints", 13-th International Math. Symp., Tokyo, Japan, Aug.29-Sept.2, 1988.
- [5] S.C. Fang and S. Puthenpura, "Linear Optimization and Extensions: Theory and Algorithms", Prentice Hall, 1993.
- [6] Kailath, T., "Linear Systems", Prentice Hall, 1980.
- [7] Haykin, S., "Adaptive Filtering Theory", Prentice Hall, 1986.
- [8] Anderson, B.D.O, and Moore, J.B., "Linear Optimal Control", Prentice Hall, 1979.
- [9] Puthenpura,S., and Sinha, L.P., "A Joint Observer-Controller Design Based on the Variants of the Karmarkar Algorithm and the Kalman Filter", 14-th Mathematical Programming Symposium, Aug. 1991, Amsterdam.
- [10] Tintner, G., "A Note on Stochastic programming", *Econometrica*, Vol. 28, pp. 490-495.
- [11] Tintner,G., and Sengupta, J.K., "Stochastic Economics: Stochastic Processes, Control, and Programming", Academic Press, 1972.

- [12] Van Trees, H.L., "Detection, Estimation, and Modulation Theory", Part I, Wiley, 1968.
- [13] Stancu-Minasian, I.M., "Stochastic Programming with Multiple Objective Functions", D. Reidel Publishing Company, 1984.
- [14] Masreliez, C.J. and Martin, R.D., "Robust-resistant Spectrum Estimation", IEEE Transaction on Automatic Control, Vol.22, pp. 361-371, 1978.
- [15] Puthenpura S. and Sinha, N.K., "Robust Bootstrap Method for Joint Estimation of States and Parameters of a Linear System", Journal of Dynamic Systems, Measurement, and Control, Vol.108, pp.255-263, 1986.