Report no. 07/18

Kalman Filtering with Equality and Inequality State Constraints

Nachi Gupta Raphael Hauser
Oxford University Computing Laboratory, Numerical Analysis Group,
Wolfson Building, Parks Road, Oxford OX1 3QD, U.K.

Both constrained and unconstrained optimization problems regularly appear in recursive tracking problems engineers currently address – however, constraints are rarely exploited for these applications. We define the Kalman Filter and discuss two different approaches to incorporating constraints. Each of these approaches are first applied to equality constraints and then extended to inequality constraints. We discuss methods for dealing with nonlinear constraints and for constraining the state prediction. Finally, some experiments are provided to indicate the usefulness of such methods.

Key words and phrases: Constrained optimization, Kalman filtering, Nonlinear filters, Optimization methods, Quadratic programming, State estimation

Oxford University Computing Laboratory
Numerical Analysis Group
Wolfson Building
Parks Road
Oxford, England OX1 3QD
E-mail: nachi@comlab.oxford.ac.uk

August, 2007

1 Introduction

Kalman Filtering [8] is a method to make real-time predictions for systems with some known dynamics. Traditionally, problems requiring Kalman Filtering have been complex and nonlinear. Many advances have been made in the direction of dealing with nonlinearities (e.g., Extended Kalman Filter [1], Unscented Kalman Filter [7]). These problems also tend to have inherent state space *equality* constraints (e.g., a fixed speed for a robotic arm) and state space *inequality* constraints (e.g., maximum attainable speed of a motor). In the past, less interest has been generated towards constrained Kalman Filtering, partly because constraints can be difficult to model. As a result, constraints are often neglected in standard Kalman Filtering applications.

The extension to Kalman Filtering with known equality constraints on the state space is discussed in [5, 12–14, 16]. In this paper, we discuss two distinct methods to incorporate constraints into a Kalman Filter. Initially, we discuss these in the framework of equality constraints. The first method, projecting the updated state estimate onto the constrained region, appears with some discussion in [5, 12]. We propose another method, which is to restrict the optimal Kalman Gain so the updated state estimate will not violate the constraint. With some algebraic manipulation, the second method is shown to be a special case of the first method.

We extend both of these concepts to Kalman Filtering with inequality constraints in the state space. This generalization for the first approach was discussed in [11].[1] Constraining the optimal Kalman Gain was briefly discussed in [10]. Further, we will also make the extension to incorporating state space constraints in Kalman Filter predictions.

Analogous to the way a Kalman Filter can be extended to solve problems containing non-linearities in the dynamics using an Extended Kalman Filter by linearizing locally (or by using an Unscented Kalman Filter), linear inequality constrained filtering can similarly be extended to problems with nonlinear constraints by linearizing locally (or by way of another scheme like an Unscented Kalman Filter). The accuracy achieved by methods dealing with nonlinear constraints will naturally depend on the structure and curvature of the nonlinear function itself. In the two experiments we provide, we look at incorporating inequality constraints to a tracking problem with nonlinear dynamics.

2 Kalman Filter

A discrete-time Kalman Filter [8] attempts to find the best running estimate for a recursive system governed by the following model[2]:

$$x_k = F_{k,k-1}x_{k-1} + u_{k,k-1}, \qquad u_{k,k-1} \sim \mathcal{N}\left(0, Q_{k,k-1}\right) \tag{2.1}$$

[1] The similar extension for the method of [16] was made in [6].

[2] The subscript k on a variable stands for the k-th time step, the mathematical notation $\mathcal{N}\left(\mu, \Sigma\right)$ denotes a normally distributed random vector with mean μ and covariance Σ, and all vectors in this paper are column vectors (unless we are explicitly taking the transpose of the vector).

$$z_k = H_k x_k + v_k, \qquad v_k \sim \mathcal{N}(0, R_k) \tag{2.2}$$

Here x_k is an n-vector that represents the true state of the underlying system and $F_{k,k-1}$ is an $n \times n$ matrix that describes the transition dynamics of the system from x_{k-1} to x_k. The measurement made by the observer is an m-vector z_k, and H_k is an $m \times n$ matrix that transforms a vector from the state space into the appropriate vector in the measurement space. The noise terms $u_{k,k-1}$ (an n-vector) and v_k (an m-vector) encompass known and unknown errors in $F_{k,k-1}$ and H_k and are normally distributed with mean 0 and covariances given by $n \times n$ matrix $Q_{k,k-1}$ and $m \times m$ matrix R_k, respectively. At each iteration, the Kalman Filter makes a state prediction for x_k, denoted $\hat{x}_{k|k-1}$. We use the notation $k|k-1$ since we will only use measurements provided until time-step $k-1$ in order to make the prediction at time-step k. The state prediction error $\tilde{x}_{k|k-1}$ is defined as the difference between the true state and the state prediction, as below.

$$\tilde{x}_{k|k-1} = x_k - \hat{x}_{k|k-1} \tag{2.3}$$

The covariance structure for the expected error on the state prediction is defined as the expectation of the outer product of the state prediction error. We call this covariance structure the error covariance prediction and denote it $P_{k|k-1}$.[3]

$$P_{k|k-1} = \mathbb{E}\left[\left(\tilde{x}_{k|k-1}\right)\left(\tilde{x}_{k|k-1}\right)'\right] \tag{2.4}$$

The filter will also provide an updated state estimate for x_k, given all the measurements provided up to and including time step k. We denote these estimates by $\hat{x}_{k|k}$. We similarly define the state estimate error $\tilde{x}_{k|k}$ as below.

$$\tilde{x}_{k|k} = x_k - \hat{x}_{k|k} \tag{2.5}$$

The expectation of the outer product of the state estimate error represents the covariance structure of the expected errors on the state estimate, which we call the updated error covariance and denote $P_{k|k}$.

$$P_{k|k} = \mathbb{E}\left[\left(\tilde{x}_{k|k}\right)\left(\tilde{x}_{k|k}\right)'\right] \tag{2.6}$$

At time-step k, we can make a prediction for the underlying state of the system by allowing the state to transition forward using our model for the dynamics and noting that $\mathbb{E}\left[u_{k,k-1}\right] = 0$. This serves as our state prediction.

$$\hat{x}_{k|k-1} = F_{k,k-1}\hat{x}_{k-1|k-1} \tag{2.7}$$

If we expand the expectation in Equation (2.4), we have the following equation for the error covariance prediction.

$$P_{k|k-1} = F_{k,k-1}P_{k-1|k-1}F'_{k,k-1} + Q_{k,k-1} \tag{2.8}$$

[3]We use the prime notation on a vector or a matrix to denote its transpose throughout this paper.

We can transform our state prediction into the measurement space, which is a prediction for the measurement we now expect to observe.

$$\hat{z}_{k|k-1} = H_k \hat{x}_{k|k-1} \tag{2.9}$$

The difference between the observed measurement and our predicted measurement is the measurement residual, which we are hoping to minimize in this algorithm.

$$\nu_k = z_k - \hat{z}_{k|k-1} \tag{2.10}$$

We can also calculate the associated covariance for the measurement residual, which is the expectation of the outer product of the measurement residual with itself, $\mathbb{E}\left[\nu_k \nu_k'\right]$. We call this the measurement residual covariance.

$$S_k = H_k P_{k|k-1} H_k' + R_k \tag{2.11}$$

We can now define our updated state estimate as our prediction plus some perturbation, which is given by a weighting factor times the measurement residual. The weighting factor, called the Kalman Gain, will be discussed below.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \nu_k \tag{2.12}$$

Naturally, we can also calculate the updated error covariance by expanding the outer product in Equation (2.6).[4]

$$P_{k|k} = \left(\mathrm{I} - K_k H_k\right) P_{k|k-1} \left(\mathrm{I} - K_k H_k\right)' + K_k R_k K_k' \tag{2.13}$$

Now we would like to find the Kalman Gain K_k, which minimizes the mean square state estimate error, $\mathbb{E}\left[\left|\tilde{x}_{k|k}\right|^2\right]$. This is the same as minimizing the trace of the updated error covariance matrix above.[5] After some calculus, we find the optimal gain that achieves this, written below.[6]

$$K_k = P_{k|k-1} H_k' S_k^{-1} \tag{2.14}$$

The covariance matrices in the Kalman Filter provide us with a measure for uncertainty in our predictions and updated state estimate. This is a very important feature for the various applications of filtering since we then know how much to trust our predictions and estimates. Also, since the method is recursive, we need to provide an initial covariance that is large enough to contain the initial state to ensure comprehensible performance. For a more detailed discussion of Kalman Filtering, we refer the reader to the following book [1].

[4]The I in Equation (2.13) represents the $n \times n$ identity matrix. Throughout this paper, we use I to denote the same matrix, except in Appendix A, where I is the appropriately sized identity matrix.

[5]Note that $v'v = \text{trace}\left[vv'\right]$ for some vector v.

[6]We could also minimize the mean square state estimate error in the N norm, where N is a positive definite and symmetric weighting matrix. In the N norm, the optimal gain would be $K_k^N = N^{\frac{1}{2}} K_k$.

3 Equality Constrained Kalman Filtering

A number of approaches have been proposed for solving the equality constrained Kalman Filtering problem [5, 12–14, 16]. In this paper, we show two different methods. The first method will restrict the state at each iteration to lie in the equality constrained space. The second method will start with a constrained prediction, and restrict the Kalman Gain so that the estimate will lie in the constrained space. Our equality constraints in this paper will be defined as below, where A is a $q \times n$ matrix, b a q-vector, and x_k, the state, is a n-vector.[7]

$$Ax_k = b \tag{3.1}$$

So we would like our updated state estimate to satisfy the constraint at each iteration, as below.

$$A\hat{x}_{k|k} = b \tag{3.2}$$

Similarly, we may also like the state prediction to be constrained, which would allow a better forecast for the system.

$$A\hat{x}_{k|k-1} = b \tag{3.3}$$

In the following subsections, we will discuss methods for constraining the updated state estimate. In Section 4, we will extend these concepts and formulations to the inequality constrained case, and in Section 6, we will address the problem of constraining the prediction, as well.

3.1 Projecting the state to lie in the constrained space

We can solve the following minimization problem for a given time-step k, where $\hat{x}^P_{k|k}$ is the constrained estimate, W_k is any positive definite symmetric weighting matrix, and $\hat{x}_{k|k}$ is the unconstrained Kalman Filter updated estimate.

$$\hat{x}^P_{k|k} = \underset{x \in \mathbb{R}^n}{\arg\min} \left\{ \left(x - \hat{x}_{k|k}\right)' W_k \left(x - \hat{x}_{k|k}\right) : Ax = b \right\} \tag{3.4}$$

The best constrained estimate is then given by

$$\hat{x}^P_{k|k} = \hat{x}_{k|k} - W_k^{-1} A' \left(AW_k^{-1}A'\right)^{-1} \left(A\hat{x}_{k|k} - b\right) \tag{3.5}$$

To find the updated error covariance matrix of the equality constrained filter, we first define the matrix Υ below.[8]

$$\Upsilon = W_k^{-1} A' \left(AW_k^{-1}A'\right)^{-1} \tag{3.6}$$

Equation (3.5) can then be re-written as following.

[7] A and b can be different for different k. We don't subscript each A and b to avoid confusion.

[8] Note that ΥA is a projection matrix, as is $(\mathrm{I} - \Upsilon A)$, by definition. If A is poorly conditioned, we can use a QR factorization to avoid squaring the condition number.

$$\hat{x}_{k|k}^{P} = \hat{x}_{k|k} - \Upsilon \left(A\hat{x}_{k|k} - b \right) \tag{3.7}$$

We can find a reduced form for $x_k - \hat{x}_{k|k}^{P}$ as below.

$$\begin{aligned} x_k - \hat{x}_{k|k}^{P} &= x_k - \hat{x}_{k|k} + \Upsilon \left(A\hat{x}_{k|k} - b - (Ax_k - b) \right) && \text{(3.8a)} \\ &= x_k - \hat{x}_{k|k} + \Upsilon \left(A\hat{x}_{k|k} - Ax_k \right) && \text{(3.8b)} \\ &= -\left(\mathrm{I} - \Upsilon A \right) \left(\hat{x}_{k|k} - x_k \right) && \text{(3.8c)} \end{aligned}$$

Using the definition of the error covariance matrix, we arrive at the following expression.

$$\begin{aligned} P_{k|k}^{P} &= \mathbb{E}\left[\left(x_k - \hat{x}_{k|k}^{P} \right) \left(x_k - \hat{x}_{k|k}^{P} \right)' \right] && \text{(3.9a)} \\ &= \mathbb{E}\left[\left(\mathrm{I} - \Upsilon A \right) \left(\hat{x}_{k|k} - x_k \right) \left(\hat{x}_{k|k} - x_k \right)' \left(\mathrm{I} - \Upsilon A \right)' \right] && \text{(3.9b)} \\ &= \left(\mathrm{I} - \Upsilon A \right) P_{k|k} \left(\mathrm{I} - \Upsilon A \right)' && \text{(3.9c)} \\ &= P_{k|k} - \Upsilon A P_{k|k} - P_{k|k} A' \Upsilon' + \Upsilon A P_{k|k} A' \Upsilon' && \text{(3.9d)} \\ &= P_{k|k} - \Upsilon A P_{k|k} && \text{(3.9e)} \\ &= \left(\mathrm{I} - \Upsilon A \right) P_{k|k} && \text{(3.9f)} \end{aligned}$$

It can be shown that choosing $W_k = P_{k|k}^{-1}$ results in the smallest updated error covariance. This also provides a measure of the information in the state at k.[9]

3.2 Restricting the optimal Kalman Gain so the updated state estimate lies in the constrained space

Alternatively, we can expand the updated state estimate term in Equation (3.2) using Equation (2.12).

$$A\left(\hat{x}_{k|k-1} + K_k \nu_k \right) = b \tag{3.10}$$

Then, we can choose a Kalman Gain K_k^R, that forces the updated state estimate to be in the constrained space. In the unconstrained case, we chose the optimal Kalman Gain K_k, by solving the minimization problem below which yields Equation (2.14).

$$K_k = \underset{K \in \mathbb{R}^{n \times m}}{\arg\min} \operatorname{trace}\left[\left(\mathrm{I} - KH_k \right) P_{k|k-1} \left(\mathrm{I} - KH_k \right)' + KR_k K' \right] \tag{3.11}$$

Now we seek the optimal K_k^R that satisfies the constrained optimization problem written below for a given time-step k.

[9] If M and N are covariance matrices, we say N is smaller than M if $M - N$ is positive semidefinite. Another formulation for incorporating equality constraints into a Kalman Filter is by observing the constraints as pseudo-measurements [14, 16]. When W_k is chosen to be $P_{k|k}^{-1}$, both of these methods are mathematically equivalent [5]. Also, a more numerically stable form of Equation (3.9) with discussion is provided in [5].

$$K_k^R = \underset{K\in\mathbb{R}^{n\times m}}{\arg\min} \text{trace}\left[(\mathrm{I}-KH_k)\,P_{k|k-1}\,(\mathrm{I}-KH_k)' + KR_kK'\right] \\ \text{s.t. } A\left(\hat{x}_{k|k-1} + K\nu_k\right) = b \tag{3.12}$$

We will solve this problem using the method of Lagrange Multipliers. First, we take the steps below, using the vec notation (column stacking matrices so they appear as long vectors, see Appendix A) to convert all appearances of K in Equation (4.8) into long vectors. Let us begin by expanding the following term.[10]

$$\begin{aligned} &\text{trace}\left[(\mathrm{I}-KH_k)\,P_{k|k-1}\,(\mathrm{I}-KH_k)' + KR_kK'\right] \\ =\ &\text{trace}\left[P_{k|k-1} - KH_kP_{k|k-1} - P_{k|k-1}H_k'K' + KH_kP_{k|k-1}H_k'K' + KR_kK'\right] \\ \overset{(2.11)}{=}\ &\text{trace}\left[P_{k|k-1} - KH_kP_{k|k-1} - P_{k|k-1}H_k'K' + KS_kK'\right] \\ =\ &\text{trace}\left[P_{k|k-1}\right] - \text{trace}\left[KH_kP_{k|k-1}\right] - \text{trace}\left[P_{k|k-1}H_k'K'\right] + \text{trace}\left[KS_kK'\right] \end{aligned} \tag{3.13a}$$

We now expand the last three terms in Equation (3.13a) one at a time.[11]

$$\begin{aligned} \text{trace}\left[KH_kP_{k|k-1}\right] &\overset{(A.9)}{=} \text{vec}\left[\left(H_kP_{k|k-1}\right)'\right]' \text{vec}\left[K\right] \\ &= \text{vec}\left[P_{k|k-1}H_k'\right]' \text{vec}\left[K\right] \end{aligned} \tag{3.14}$$

$$\text{trace}\left[P_{k|k-1}H_k'K'\right] \overset{(A.9)}{=} \text{vec}\left[K\right]' \text{vec}\left[P_{k|k-1}H_k'\right] \tag{3.15}$$

$$\begin{aligned} \text{trace}\left[KS_kK'\right] &\overset{(A.9)}{=} \text{vec}\left[K\right]' \text{vec}\left[KS_k\right] \\ &\overset{(A.7)}{=} \text{vec}\left[K\right]' \left(S\otimes \mathrm{I}\right) \text{vec}\left[K\right] \end{aligned} \tag{3.16}$$

Remembering that trace $\left[P_{k|k-1}\right]$ is constant, our objective function can be written as below.

$$\begin{aligned} &\text{vec}\left[K\right]' \left(\mathrm{I}\otimes S_k\right) \text{vec}\left[K'\right] - \text{vec}\left[P_{k|k-1}H_k'\right]' \text{vec}\left[K\right] \\ &\qquad - \text{vec}\left[K\right]' \text{vec}\left[P_{k|k-1}H_k'\right] \end{aligned} \tag{3.17}$$

Using Equation (A.8) on the equality constraints, our minimization problem is the following.

$$\begin{aligned} K_k^R = \underset{K\in\mathbb{R}^{n\times m}}{\arg\min}\ &\text{vec}\left[K\right]' \left(S_k\otimes \mathrm{I}\right) \text{vec}\left[K\right] \\ &- \text{vec}\left[P_{k|k-1}H_k'\right]' \text{vec}\left[K\right] \\ &- \text{vec}\left[K\right]' \text{vec}\left[P_{k|k-1}H_k'\right] \\ \text{s.t. }\ &\left(\nu_k'\otimes A\right) \text{vec}\left[K\right] = b - A\hat{x}_{k|k-1} \end{aligned} \tag{3.18}$$

[10] Throughout this paper, a number in parentheses above an equals sign means we made use of this equation number.

[11] We use the symmetry of $P_{k|k-1}$ in Equation (3.14) and the symmetry of S_k in Equation (3.16).

Further, we simplify this problem so the minimization problem has only one quadratic term. We complete the square as follows. We want to find the unknown variable μ which will cancel the linear term. Let the quadratic term appear as follows. Note that the non-"vec $[K]$" term is dropped as is is irrelevant for the minimization problem.

$$(\text{vec}\,[K] + \mu)' \,(S_k \otimes \text{I})\,(\text{vec}\,[K] + \mu) \tag{3.19}$$

The linear term in the expansion above is the following.

$$\text{vec}\,[K]' \,(S_k \otimes \text{I})\,\mu + \mu' \,(S_k \otimes \text{I})\,\text{vec}\,[K] \tag{3.20}$$

So we require that the two equations below hold.

$$\begin{aligned} (S_k \otimes \text{I})\,\mu &= -\text{vec}\left[P_{k|k-1}H_k'\right] \\ \mu' \,(S_k \otimes \text{I}) &= -\text{vec}\left[P_{k|k-1}H_k'\right]' \end{aligned} \tag{3.21}$$

This leads to the following value for μ.

$$\begin{aligned} \mu &\overset{\text{(A.3)}}{=} -\left(S_k^{-1} \otimes \text{I}\right)\text{vec}\left[P_{k|k-1}H_k'\right] \\ &\overset{\text{(A.8)}}{=} -\text{vec}\left[P_{k|k-1}H_k'S_k^{-1}\right] \\ &\overset{\text{(2.14)}}{=} -\text{vec}\left[K_k\right] \end{aligned} \tag{3.22}$$

Using Equation (A.6), our quadratic term in the minimization problem becomes the following.

$$(\text{vec}\,[K - K_k])' \,(S_k \otimes \text{I})\,(\text{vec}\,[K - K_k]) \tag{3.23}$$

Let $l = \text{vec}\,[K - K_k]$. Then our minimization problem becomes the following.

$$\begin{aligned} K_k^R = \underset{l \in \mathbb{R}^{mn}}{\arg\min}\ & l' \,(S_k \otimes \text{I})\, l \\ \text{s.t.}\ \ & (\nu_k' \otimes A)\,(l + \text{vec}\,[K_k]) = b - A\hat{x}_{k|k-1} \end{aligned} \tag{3.24}$$

We can then re-write the constraint taking the vec $[K_k]$ term to the other side as below.

$$\begin{aligned} (\nu_k' \otimes A)\, l &= b - A\hat{x}_{k|k-1} - (\nu_k' \otimes A)\,\text{vec}\,[K_k] \\ &\overset{\text{(A.8)}}{=} b - A\hat{x}_{k|k-1} - \text{vec}\,[AK_k\nu_k] \\ &= b - A\hat{x}_{k|k-1} - AK_k\nu_k \\ &\overset{\text{(2.12)}}{=} b - A\hat{x}_{k|k} \end{aligned} \tag{3.25}$$

This results in the following simplified form.

$$\begin{aligned} K_k^R = \underset{l \in \mathbb{R}^{mn}}{\arg\min}\ & l' \,(S_k \otimes \text{I})\, l \\ \text{s.t.}\ \ & (\nu_k' \otimes A)\, l = b - A\hat{x}_{k|k} \end{aligned} \tag{3.26}$$

We form the Lagrangian $\mathcal{L}$, where we introduce q Lagrange Multipliers in vector $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_q)'$

$$\mathcal{L} = l' \left(S_k \otimes \mathrm{I}\right) l - \lambda' \left[\left(\nu_k' \otimes A\right) l - b + A\hat{x}_{k|k}\right] \tag{3.27}$$

We take the partial derivative with respect to l.[12]

$$\frac{\partial \mathcal{L}}{\partial l} = 2l' \left(S_k \otimes \mathrm{I}\right) - \lambda' \left(\nu_k' \otimes A\right) \tag{3.28}$$

Similarly we can take the partial derivative with respect to the vector λ.

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \left(\nu_k' \otimes A\right) l - b + A\hat{x}_{k|k} \tag{3.29}$$

When both of these derivatives are set equal to the appropriate size zero vector, we have the solution to the system. Taking the transpose of Equation (3.28), we can write this system as $Mn = p$ with the following block definitions for M, n, and p.

$$M = \begin{bmatrix} 2S_k \otimes \mathrm{I} & \nu_k \otimes A' \\ \nu_k' \otimes A & 0_{[q \times q]} \end{bmatrix} \tag{3.30}$$

$$n = \begin{bmatrix} l \\ \lambda \end{bmatrix} \tag{3.31}$$

$$p = \begin{bmatrix} 0_{[mn \times 1]} \\ b - A\hat{x}_{k|k} \end{bmatrix} \tag{3.32}$$

We solve this system for vector n in Appendix C. The solution for l is pasted below.

$$\left(\left[S_k^{-1}\nu_k \left(\nu_k' S_k^{-1} \nu_k\right)^{-1}\right] \otimes \left[A' \left(AA'\right)^{-1}\right]\right) \left(b - A\hat{x}_{k|k}\right) \tag{3.33}$$

Bearing in mind that $b - A\hat{x}_{k|k} = \mathrm{vec}\left[b - A\hat{x}_{k|k}\right]$, we can use Equation (A.8) to re-write l as below.[13]

$$\mathrm{vec}\left[A' \left(AA'\right)^{-1} \left(b - A\hat{x}_{k|k}\right) \left(\nu_k' S_k^{-1} \nu_k\right)^{-1} \nu_k' S_k^{-1}\right] \tag{3.34}$$

The resulting matrix inside the vec operation is then an n by m matrix. Remembering the definition for l, we notice that $K - K_k$ results in an n by m matrix also. Since both of the components inside the vec operation result in matrices of the same size, we can safely remove the vec operation from both sides. This results in the following optimal constrained Kalman Gain K_k^R.

$$K_k - A' \left(AA'\right)^{-1} \left(A\hat{x}_{k|k} - b\right) \left(\nu_k' S_k^{-1} \nu_k\right)^{-1} \nu_k' S_k^{-1} \tag{3.35}$$

If we now substitute this Kalman Gain into Equation (2.12) to find the constrained updated state estimate, we end up with the following.

$$\hat{x}_{k|k}^R = \hat{x}_{k|k} - A' \left(AA'\right)^{-1} \left(A\hat{x}_{k|k} - b\right) \tag{3.36}$$

[12] We used the symmetry of $\left(S_k \otimes \mathrm{I}\right)$ here.

[13] Here we used the symmetry of S_k^{-1} and $\left(\nu_k' S_k^{-1} \nu_k\right)^{-1}$ (the latter of which is actually just a scalar).

This is of course equivalent to the result of Equation (3.5) with the weighting matrix W_k chosen as the identity matrix. The error covariance for this estimate is given by Equation (3.9).[14]

4 Adding Inequality Constraints

In the more general case of this problem, we may encounter equality and inequality constraints, as given below.[15]

$$\begin{aligned} Ax_k &= b \\ Cx_k &\leq d \end{aligned} \tag{4.1}$$

So we would like our updated state estimate to satisfy the constraint at each iteration, as below.

$$\begin{aligned} A\hat{x}_{k|k} &= b \\ C\hat{x}_{k|k} &\leq d \end{aligned} \tag{4.2}$$

Similarly, we may also like the state prediction to be constrained, which would allow a better forecast for the system.

$$\begin{aligned} A\hat{x}_{k|k-1} &= b \\ C\hat{x}_{k|k-1} &\leq d \end{aligned} \tag{4.3}$$

We will present two analogous methods to those presented for the equality constrained case. In the first method, we will run the unconstrained filter, and at each iteration constrain the updated state estimate to lie in the constrained space. In the second method, we will find a Kalman Gain $\check{K}_k^R$ such that the the updated state estimate will be forced to lie in the constrained space. In both methods, we will no longer be able to find an analytic solution as before. Instead, we use numerical methods.

4.1 By Projecting the Unconstrained Estimate

Given the best unconstrained estimate, we could solve the following minimization problem for a given time-step k, where $\check{x}_{k|k}^P$ is the inequality constrained estimate and W_k is any positive definite symmetric weighting matrix.

[14] We can use the unconstrained or constrained Kalman Gain to find this error covariance matrix. Since the constrained Kalman Gain is suboptimal for the unconstrained problem, before projecting onto the constrained space, the constrained covariance will be different from the unconstrained covariance. However, the difference lies exactly in the space orthogonal to which the covariance is projected onto by Equation (3.9). The proof is omitted for brevity.

[15] C and d can be different for different k. We don't subscript each C and d to simplify notation.

$$\begin{aligned} \check{x}^P_{k|k} = \arg\min_x \ & \left(x - \hat{x}_{k|k}\right)' W_k \left(x - \hat{x}_{k|k}\right) \\ \text{s.t. } & Ax = b \\ & Cx \leq d \end{aligned} \tag{4.4}$$

For solving this inequality constrained optimization problem, we can use a variety of standard methods, or even an out-of-the-box solver, like `fmincon` in Matlab. Here we use an active set method [4]. This is a common method for dealing with inequality constraints, where we treat a subset of the constraints (called the active set) as additional equality constraints. We ignore any inactive constraints when solving our optimization problem. After solving the problem, we check if our solution lies in the space given by the inequality constraints. If it doesn't, we start from the solution in our previous iteration and move in the direction of the new solution until we hit a set of constraints. For each iteration, the active set is made up of those inequality constraints with non-zero Lagrange Multipliers.

We first find the best estimate (using Equation (3.5) for the equality constrained problem with the equality constraints given in Equation (4.1) plus the active set of inequality constraints. Let us call the solution to this $\check{x}^{P*}_{k|k,j}$ since we have not yet checked if the solution lies in the inequality constrained space.[16] In order to check this, we find the vector that we moved along to reach $\check{x}^{P*}_{k|k,j}$. This is given by the following.

$$s = \check{x}^{P*}_{k|k,j} - \check{x}^P_{k|k,j-1} \tag{4.5}$$

We now iterate through each of our inequality constraints, to check if they are satisfied. If they are all satisfied, we choose $\tau_{\max} = 1$. If they are not, we choose the largest value of $\tau_{\max}$ such that $\hat{x}_{k|k,j-1} + \tau_{\max} s$ lies in the inequality constrained space. We choose our estimate to be

$$\check{x}^P_{k|k,j} = \check{x}^P_{k|k,j-1} + \tau_{\max} s \tag{4.6}$$

If we find the solution has converged within a pre-specified error, or we have reached a pre-specified maximum number of iterations, we choose this as the updated state estimate to our inequality constrained problem, denoted $\check{x}^P_{k|k}$. If we would like to take a further iteration on j, we check the Lagrange Multipliers at this new solution to determine the new active set.[17] We then repeat by finding the best estimate for the equality constrained problem including the new active set as additional equality constraints. Since this is a Quadratic Programming problem, each step of j guarantees the same estimate or a better estimate.

When calculating the error covariance matrix for this estimate, we can also add on the safety term below.

$$\left(\check{x}^P_{k|k,j} - \check{x}^P_{k|k,j-1}\right)\left(\check{x}^P_{k|k,j} - \check{x}^P_{k|k,j-1}\right)' \tag{4.7}$$

[16] For the inequality constrained filter, we allow multiple iterations within each step. The j subscript indexes these further iterations.

[17] The previous active set is not relevant.

This is a measure of our convergence error and should typically be small relative to the unconstrained error covariance. We can then use Equation (3.9) to project the covariance matrix onto the constrained subspace, but we only use the defined equality constraints. We do not incorporate any constraints in the active set when computing Equation (3.9) since these still represent inequality constraints on the state. Ideally we would project the error covariance matrix into the inequality constrained subspace, but this projection is not trivial.

4.2 By Restricting the Optimal Kalman Gain

We could solve this problem by restricting the optimal Kalman gain also, as we did for equality constraints previously. We seek the optimal K_k that satisfies the constrained optimization problem written below for a given time-step k.

$$\begin{aligned} \check{K}_k^R = \underset{K \in \mathbb{R}^{n \times m}}{\arg\min} & \operatorname{trace}\left[(\mathrm{I} - KH_k) P_{k|k-1} (\mathrm{I} - KH_k)' + KR_kK' \right] \\ \text{s.t. } & A\left(\hat{x}_{k|k-1} + K_k\nu_k\right) = b \\ & C\left(\hat{x}_{k|k-1} + K_k\nu_k\right) \leq d \end{aligned} \tag{4.8}$$

Again, we can solve this problem using any inequality constrained optimization method (e.g., `fmincon` in Matlab or the active set method used previously). Here we solved the optimization problem using SDPT3, a Matlab package for solving semidefinite programming problems [15]. When calculating the covariance matrix for the inequality constrained estimate, we use the restricted Kalman Gain. Again, we can add on the safety term for the convergence error, by taking the outer product of the difference between the updated state estimates calculated by the restricted Kalman Gain for the last two iterations of SDPT3. This covariance matrix is then projected onto the subspace as in Equation (3.9) using the equality constraints only.

5 Dealing with Nonlinearities

Thus far, in the Kalman Filter we have dealt with linear models and constraints. A number of methods have been proposed to handle nonlinear models (e.g., Extended Kalman Filter [1], Unscented Kalman Filter [7]). In this paper, we will focus on the most widely used of these, the Extended Kalman Filter. Let's re-write the discrete unconstrained Kalman Filtering problem from Equations (2.1) and (2.2) below, incorporating nonlinear models.

$$x_k = f_{k,k-1}\left(x_{k-1}\right) + u_{k,k-1}, \qquad u_{k,k-1} \sim \mathcal{N}\left(0, Q_{k,k-1}\right) \tag{5.1}$$

$$z_k = h_k\left(x_k\right) + v_k, \qquad v_k \sim \mathcal{N}\left(0, R_k\right) \tag{5.2}$$

In the above equations, we see that the transition matrix $F_{k,k-1}$ has been replaced by the nonlinear vector-valued function $f_{k,k-1}\left(\cdot\right)$, and similarly, the matrix H_k, which transforms a vector from the state space into the measurement space, has been replaced by the nonlinear vector-valued function $h_k\left(\cdot\right)$. The method proposed by the Extended Kalman Filter is to

linearize the nonlinearities about the current state prediction (or estimate). That is, we choose $F_{k,k-1}$ as the Jacobian of $f_{k,k-1}$ evaluated at $\hat{x}_{k-1|k-1}$, and H_k as the Jacobian of h_k evaluated at $\hat{x}_{k|k-1}$ and proceed as in the linear Kalman Filter of Section 2.[18] Numerical accuracy of these methods tends to depend heavily on the nonlinear functions. If we have linear constraints but a nonlinear $f_{k,k-1}(\cdot)$ and $h_k(\cdot)$, we can adapt the Extended Kalman Filter to fit into the framework of the methods described thus far.

5.1 Nonlinear Equality and Inequality Constraints

Since equality and inequality constraints we model are often times nonlinear, it is important to make the extension to nonlinear equality and inequality constrained Kalman Filtering for the methods discussed thus far. Without loss of generality, our discussion here will pertain only to nonlinear inequality constraints. We can follow the same steps for equality constraints.[19] We replace the linear inequality constraint on the state space by the following nonlinear inequality constraint $c(x_k) = d$, where $c(\cdot)$ is a vector-valued function. We can then linearize our constraint, $c(x_k) = d$, about the current state prediction $\hat{x}_{k|k-1}$, which gives us the following.[20]

$$c\left(\hat{x}_{k|k-1}\right) + C\left(x_k - \hat{x}_{k|k-1}\right) \lessapprox d \tag{5.3}$$

Here C is defined as the Jacobian of c evaluated at $\hat{x}_{k|k-1}$. This indicates then, that the nonlinear constraint we would like to model can be approximated by the following linear constraint

$$Cx_k \lessapprox d + C\hat{x}_{k|k-1} - c\left(\hat{x}_{k|k-1}\right) \tag{5.4}$$

This constraint can be written as $\tilde{C}x_k \leq \tilde{d}$, which is an approximation to the nonlinear inequality constraint. It is now in a form that can be used by the methods described thus far.

The nonlinearities in both the constraints and the models, $f_{k,k-1}(\cdot)$ and $h_k(\cdot)$, could have been linearized using a number of different methods (e.g., a derivative-free method, a higher order Taylor approximation). Also an iterative method could be used as in the Iterated Extended Kalman Filter [1].

6 Constraining the State Prediction

We haven't yet discussed whether the state prediction (Equation (2.7)) also should be constrained. Forcing the constraints should provide a better prediction (which is used for fore-

[18] We can also do a midpoint approximation to find $F_{k,k-1}$ by evaluating the Jacobian at $\left(\hat{x}_{k-1|k-1} + \hat{x}_{k|k-1}\right)/2$. This should be a much closer approximation to the nonlinear function. We use this approximation for the Extended Kalman Filter experiments later.

[19] We replace the '$\leq$' sign with an '$=$' sign and the '$\lessapprox$' with an '$\approx$' sign.

[20] This method is how the Extended Kalman Filter linearizes nonlinear functions for $f_{k,k-1}(\cdot)$ and $h_k(\cdot)$. Here $\hat{x}_{k|k-1}$ can be the state prediction of any of the constrained filters presented thus far and does not necessarily relate to the unconstrained state prediction.

casting in the Kalman Filter). Ideally, the transition matrix $F_{k,k-1}$ will take an updated state estimate satisfying the constraints at time $k-1$ and make a prediction that will satisfy the constraints at time k. Of course this may not be the case. In fact, the constraints may depend on the updated state estimate, which would be the case for nonlinear constraints. On the downside, constraining the state prediction increases computational cost per iteration.

We propose three methods for dealing with the problem of constraining the state prediction. The first method is to project the matrix $F_{k,k-1}$ onto the constrained space. This is only possible for the equality constraints, as there is no trivial way to project $F_{k,k-1}$ to an inequality constrained space. We can use the same projector as in Equation (3.9f) so we have the following.[21]

$$F^P_{k,k-1} = \left(\mathrm{I} - \Upsilon A\right) F_{k,k-1} \tag{6.1}$$

Under the assumption that we have constrained our updated state estimate, this new transition matrix will make a prediction that will keep the estimate in the equality constrained space. Alternatively, if we weaken this assumption, i.e., we are not constraining the updated state estimate, we could solve the minimization problem below (analogous to Equation (3.4)). We can also incorporate inequality constraints now.

$$\begin{aligned} \check{x}^P_{k|k-1} = \arg\min_x \ & \left(x - \hat{x}_{k|k-1}\right)' W_k \left(x - \hat{x}_{k|k-1}\right) \\ \text{s.t. } & Ax = b \\ & Cx \leq d \end{aligned} \tag{6.2}$$

We can constrain the covariance matrix here also, in a similar fashion to the method described in Section 4.1. The third method is to add to the constrained problem the additional constraints below, which ensure that the chosen estimate will produce a prediction at the next iteration that is also constrained.

$$\begin{aligned} A_{k+1} F_{k+1,k} x_k &= b_{k+1} \\ C_{k+1} F_{k+1,k} x_k &\leq d_{k+1} \end{aligned} \tag{6.3}$$

If $A_{k+1}, b_{k+1}, C_{k+1}$ or d_{k+1} depend on the estimate (e.g., if we are linearizing nonlinear functions $a\left(\cdot\right)$ or $b\left(\cdot\right)$, we can use an iterative method, which would resolve A_{k+1} and b_{k+1} using the current best updated state estimate (or prediction), re-calculate the best estimate using A_{k+1} and b_{k+1}, and so forth until we are satisfied with the convergence. This method would be preferred since it looks ahead one time-step to choose a better estimate for the current iteration.[22] However, it can be far more expensive computationally.

[21] In these three methods, the symmetric weighting matrix W_k can be different. The resulting Υ can consequently also be different.

[22] Further, we can add constraints for some arbitrary n time-steps ahead.

7 Experiments

We provide two related experiments here. We have a car driving along a straight road with thickness 2 meters. The driver of the car traces a noisy sine curve (with the noise lying only in the frequency domain). The car is tagged with a device that transmits the location within some known error. We would like to track the position of the car. In the first experiment, we filter over the noisy data with the knowledge that the underlying function is a noisy sine curve. The inequality constrained methods will constrain the estimates to only take values in the interval $[-1, 1]$. In the second experiment, we do not use the knowledge that the underlying curve is a sine curve. Instead we attempt to recover the true data using an autoregressive model of order 6 [3]. We do, however, assume our unknown function only takes values in the interval $[-1, 1]$, and we can again enforce these constraints when using the inequality constrained filter.

The driver's path is generated using the nonlinear stochastic process given by Equation (5.1). We start with the following initial point.

$$x_0 = \begin{bmatrix} 0 \text{ m} \\ 0 \text{ m} \end{bmatrix} \tag{7.1}$$

Our vector-valued transition function will depend on a discretization parameter T and can be expressed as below. Here, we choose T to be $\pi/10$, and we run the experiment from an initial time of 0 to a final time of 10π.

$$f_{k,k-1} = \begin{bmatrix} (x_{k-1})_1 + T \\ \sin\left((x_{k-1})_1 + T\right) \end{bmatrix} \tag{7.2}$$

And for the process noise we choose the following.

$$Q_{k,k-1} = \begin{bmatrix} 0.1 \text{ m}^2 & 0 \\ 0 & 0 \text{ m}^2 \end{bmatrix} \tag{7.3}$$

The driver's path is drawn out by the second element of the vector x_k – the first element acts as an underlying state to generate the second element, which also allows a natural method to add noise in the frequency domain of the sine curve while keeping the process recursively generated.

7.1 First Experiment

To create the measurements, we use the model from Equation (2.2), where H_k is the square identity matrix of dimension 2. We choose R_k as below to noise the data. This considerably masks the true underlying data as can be seen in Fig. 1.[23]

$$R_k = \begin{bmatrix} 10 \text{ m}^2 & 0 \\ 0 & 10 \text{ m}^2 \end{bmatrix} \tag{7.4}$$

[23]The figure only shows the noisy sine curve, which is the second element of the measurement vector. The first element, which is a noisy straight line, isn't plotted.

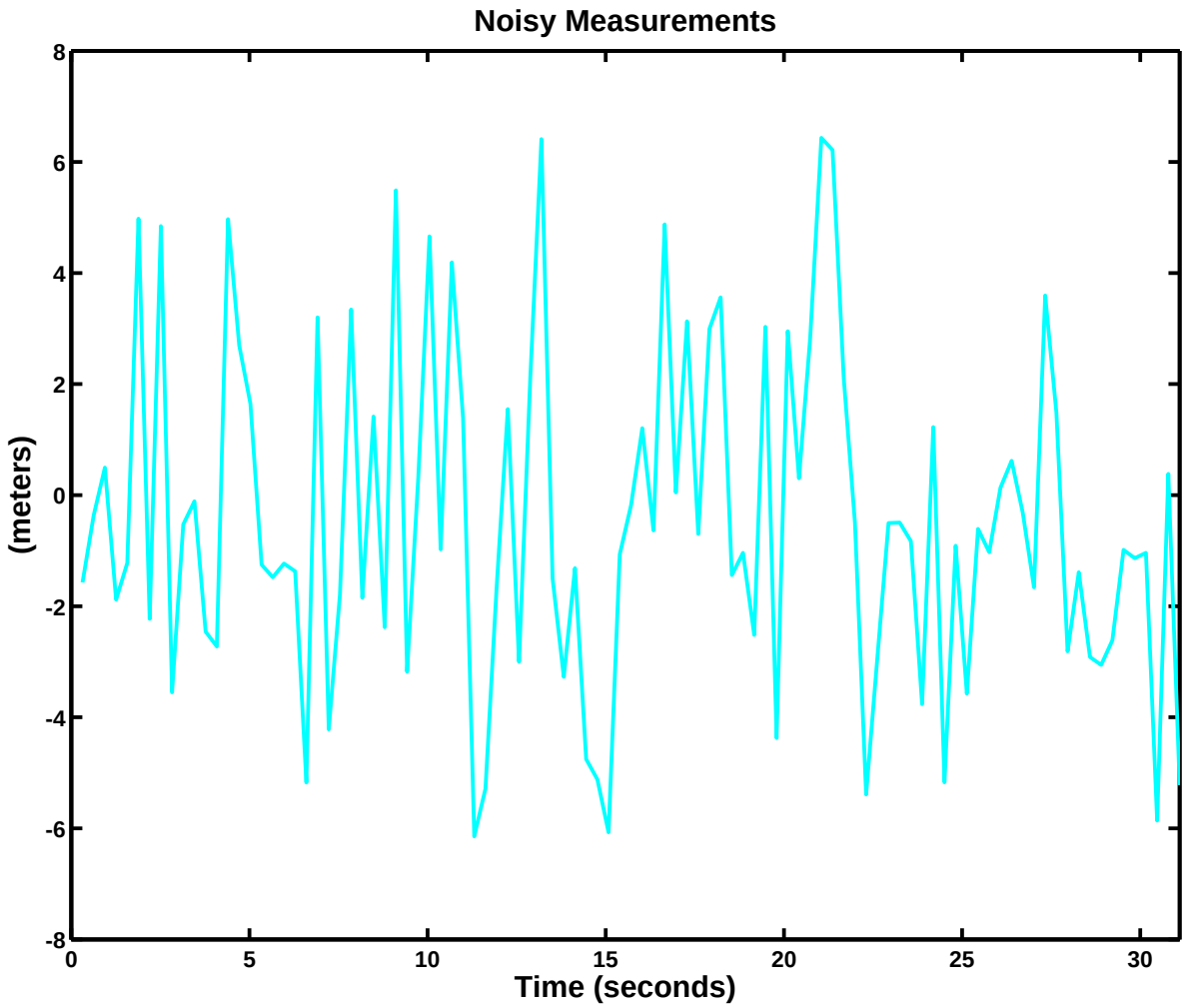

Figure 1: *We take our sine curve, which is already noisy in the frequency domain (due to process noise), and add measurement noise. The underlying sine curve is significantly masked.*

For the initial point of our filters, we choose the following point, which is different from the true initial point given in Equation (7.1).

$$\hat{x}_{0|0} = \begin{bmatrix} 0 \text{ m} \\ 1 \text{ m} \end{bmatrix} \tag{7.5}$$

Our initial covariance is given as below.[24].

$$P_{0|0} = \begin{bmatrix} 1 \text{ m}^2 & 0.1 \\ 0.1 & 1 \text{ m}^2 \end{bmatrix} \tag{7.6}$$

[24]Nonzero off-diagonal elements in the initial covariance matrix often help the filter converge more quickly

In the filtering, we use the information that the underlying function is a sine curve, and our transition function $f_{k,k-1}$ changes to reflect a recursion in the second element of x_k – now we will add on discretized pieces of a sine curve to our previous estimate. The function is given explicitly below.

$$f_{k,k-1} = \begin{bmatrix} (x_{k-1})_1 + T \\ (x_{k-1})_1 + \sin\left((x_{k-1})_1 + T\right) - \sin\left((x_{k-1})_1\right) \end{bmatrix} \tag{7.7}$$

For the Extended Kalman Filter formulation, we will also require the Jacobian of this matrix denoted $F_{k,k-1}$, which is given below.

$$F_{k,k-1} = \begin{bmatrix} 1 & 0 \\ \cos\left((x_{k-1})_1 + T\right) - \cos\left((x_{k-1})_1\right) & 1 \end{bmatrix} \tag{7.8}$$

The process noise $Q_{k,k-1}$, given below, is chosen similar to the noise used in generating the simulation, but is slightly larger to encompass both the noise in our above model and to prevent divergence due to numerical roundoff errors. The measurement noise R_k is chosen the same as in Equation (7.4).

$$Q_{k,k-1} = \begin{bmatrix} 0.1 \text{ m}^2 & 0 \\ 0 & 0.1 \text{ m}^2 \end{bmatrix} \tag{7.9}$$

The inequality constraints we enforce can be expressed using the notation throughout the chapter, with C and d as given below.

$$C = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \tag{7.10}$$

$$d = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{7.11}$$

These constraints force the second element of the estimate $x_{k|k}$ (the sine portion) to lie in the interval $[-1, 1]$. We do not have any equality constraints in this experiment. We run the unconstrained Kalman Filter and both of the constrained methods discussed previously. A plot of the true position and estimates is given in Fig. 2. Notice that both constrained methods force the estimate to lie within the constrained space, while the unconstrained method can violate the constraints.

7.2 Second Experiment

In the previous experiment, we used the knowledge that the underlying function was a noisy sine curve. If this is not known, we face a significantly harder estimation problem. Let us assume nothing about the underlying function except that it must take values in the interval $[-1, 1]$. A good model for estimating such an unknown function could be an autoregressive model. We can compare the unconstrained filter to the two constrained methods again using these assumption and an autoregressive model of order 6, or AR(6) as it is more commonly referred to.

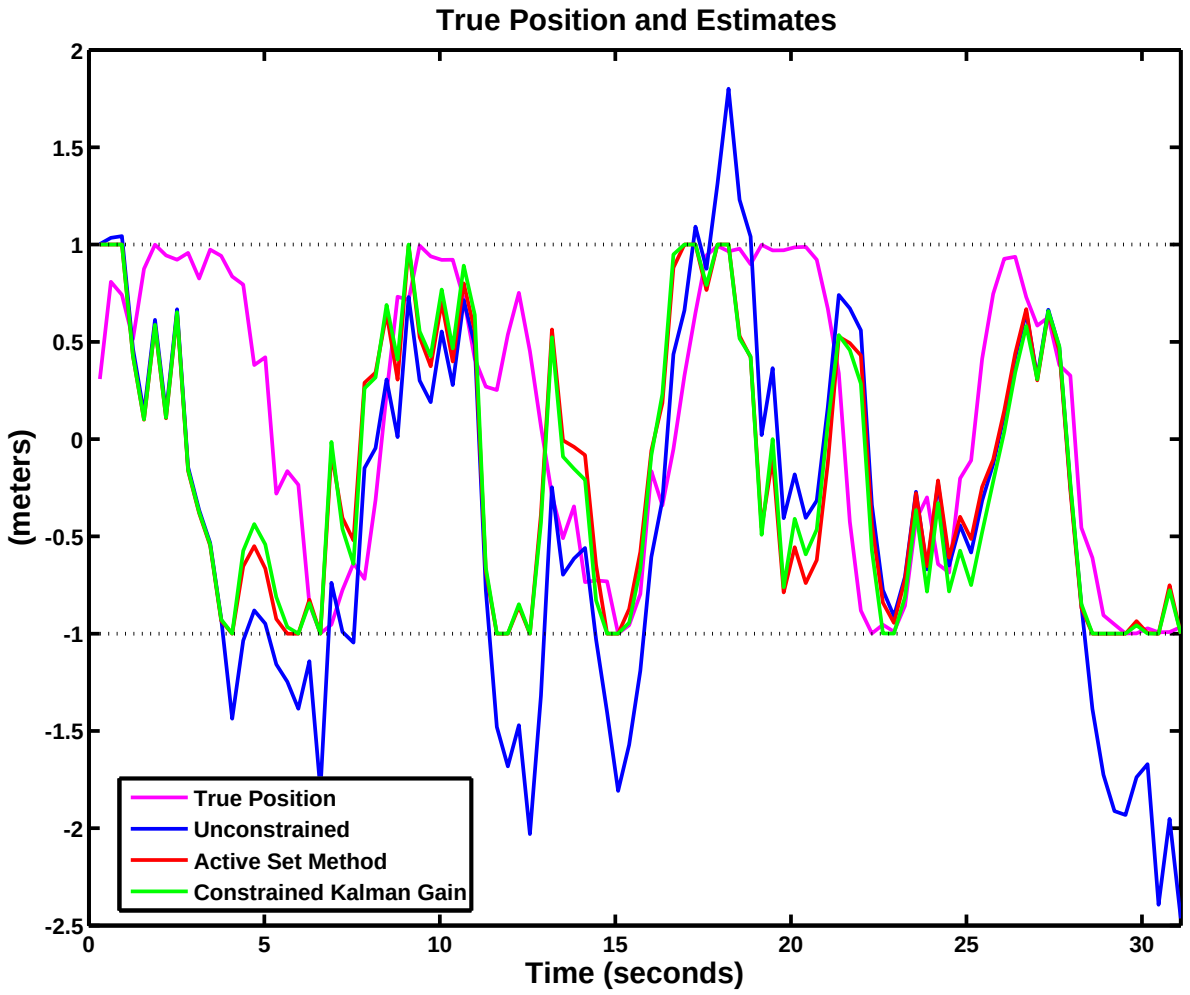

Figure 2: *We show our true underlying state, which is a sine curve noised in the frequency domain, along with the estimates from the unconstrained Kalman Filter, and both of our inequality constrained modifications. We also plotted dotted horizontal lines at the values -1 and 1. Both inequality constrained methods do not allow the estimate to leave the constrained space.*

In the previous example, we used a large measurement noise R_k to emphasize the gain achieved by using the constraint information. Such a large R_k is probably not very realistic, and when using an autoregressive model, it will be hard to track such a noisy signal. To generate the measurements, we again use Equation (2.2), this time with H_k and R_k as given below.

$$H_k = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{7.12}$$

$$R_k = \begin{bmatrix} 0.5 \text{ m}^2 \end{bmatrix} \tag{7.13}$$

Our state will now be defined using the following 13-vector, in which the first element is the current estimate, the next five elements are lags, the six elements afterwards are coefficients on the current estimate and the lags, and the last element is a constant term.

$$\hat{x}_{k|k} = \begin{bmatrix} y_k & y_{k-1} & \cdots & y_{k-5} & \alpha_1 & \alpha_2 & \cdots & \alpha_7 \end{bmatrix}' \tag{7.14}$$

Our matrix H_k in the filter is a row vector with the first element 1, and all the rest as 0, so $y_{k|k-1}$ is actually our prediction $\hat{z}_{k|k-1}$ in the filter, describing where we believe the expected value of the next point in the time-series to lie. For the initial state, we choose a vector of all zeros, except the first and seventh element, which we choose as 1. This choice for the initial conditions leads to the first prediction on the time series being 1, which is incorrect as the true underlying state has expectation 0. For the initial covariance, we choose $\mathrm{I}_{[13\times 13]}$ and add 0.1 to all the off-diagonal elements.[25] The transition function $f_{k,k-1}$ for the AR(6) model is given below.

$$\begin{bmatrix} \min\left(1, \max\left(-1, \alpha_1 y_{k-1} + \cdots + \alpha_6 y_{k-6} + \alpha_7\right)\right) \\ \min\left(1, \max\left(-1, y_{k-1}\right)\right) \\ \min\left(1, \max\left(-1, y_{k-2}\right)\right) \\ \min\left(1, \max\left(-1, y_{k-3}\right)\right) \\ \min\left(1, \max\left(-1, y_{k-4}\right)\right) \\ \min\left(1, \max\left(-1, y_{k-5}\right)\right) \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_6 \\ \alpha_7 \end{bmatrix} \tag{7.15}$$

Putting this into recursive notation, we have the following.

$$\begin{bmatrix} \min\left(1, \max\left(-1, \left(x_{k-1}\right)_7 \left(x_{k-1}\right)_1 + \cdots + \left(x_{k-1}\right)_{13}\right)\right) \\ \min\left(1, \max\left(-1, \left(x_{k-1}\right)_1\right)\right) \\ \min\left(1, \max\left(-1, \left(x_{k-1}\right)_2\right)\right) \\ \min\left(1, \max\left(-1, \left(x_{k-1}\right)_3\right)\right) \\ \min\left(1, \max\left(-1, \left(x_{k-1}\right)_4\right)\right) \\ \min\left(1, \max\left(-1, \left(x_{k-1}\right)_5\right)\right) \\ \left(x_{k-1}\right)_7 \\ \left(x_{k-1}\right)_8 \\ \vdots \\ \left(x_{k-1}\right)_{12} \\ \left(x_{k-1}\right)_{13} \end{bmatrix} \tag{7.16}$$

The Jacobian of $f_{k,k-1}$ is given below. We ignore the $\min(\cdot)$ and $\max(\cdot)$ operators since the derivative is not continuous across them, and we can reach the bounds by numerical error.

[25] The bracket subscript notation is used through the remainder of this paper to indicate the size of zero matrices and identity matrices.

Further, when enforced, the derivative would be 0, so by ignoring them, we are allowing our covariance matrix to be larger than necessary as well as more numerically stable.

$$\left[\begin{array}{c:c:c} (x_{k-1})_7 \cdots & (x_{k-1})_{12} & (x_{k-1})_1 \cdots (x_{k-1})_6 \ 1 \\ \hdashline \mathrm{I}_{[5\times5]} & 0_{[5\times1]} & 0_{[5\times7]} \\ \hdashline 0_{[7\times6]} & & \mathrm{I}_{[7\times7]} \end{array}\right] \tag{7.17}$$

For the process noise, we choose $Q_{k,k-1}$ to be a diagonal matrix with the first entry as 0.1 and all remaining entries as 10^{-6} since we know the prediction phase of the autoregressive model very well. The inequality constraints we enforce can be expressed using the notation throughout the chapter, with C as given below and d as a 12-vector of ones.

$$C = \left[\begin{array}{c:c} \mathrm{I}_{[6\times6]} & 0_{[12\times7]} \\ \hdashline -\mathrm{I}_{[6\times6]} & \end{array}\right] \tag{7.18}$$

These constraints force the current estimate and all of the lags to take values in the range $[-1, 1]$. As an added feature of this filter, we are also estimating the lags at each iteration using more information although we don't use it – this is a fixed interval smoothing. In Fig. 3, we plot the noisy measurements, true underlying state, and the filter estimates. Notice again that the constrained methods keep the estimates in the constrained space. Visually, we can see the improvement particularly near the edges of the constrained space.

8 Conclusions

We've provided two different formulations for including constraints into a Kalman Filter. In the equality constrained framework, these formulations have analytic formulas, one of which is a special case of the other. In the inequality constrained case, we've shown two numerical methods for constraining the estimate. We also discussed how to constrain the state prediction and how to handle nonlinearities. Our two examples show that these methods ensure the estimate lies in the constrained space, which provides a better estimate structure.

9 Acknowledgements

The first author would like to thank the Clarendon Bursary for financial support.

Appendix A Kron and Vec

In this appendix, we provide some definitions used earlier in the chapter. Given matrix $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, we can define the right Kronecker product as below.[26]

[26]The indices m, n, p, and q and all matrix definitions are independent of any used earlier. Also, the subscript notation $a_{1,n}$ denotes the element in the first row and n-th column of A, and so forth.

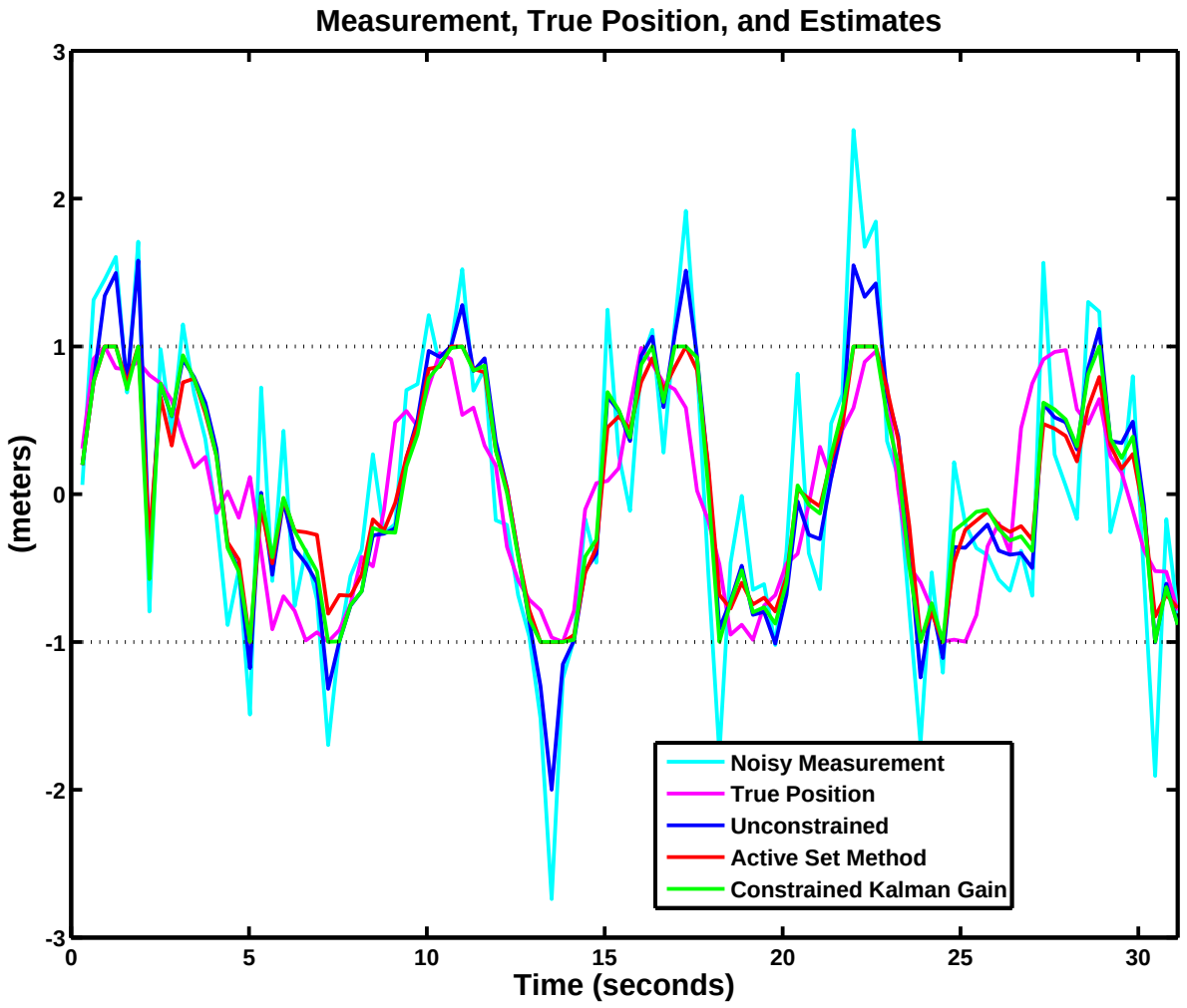

Figure 3: *We show our true underlying state, which is a sine curve noised in the frequency domain, the noised measurements, and the estimates from the unconstrained and both inequality constrained filters. We also plotted dotted horizontal lines at the values -1 and 1. Both inequality constrained methods do not allow the estimate to leave the constrained space.*

$$(A \otimes B) = \begin{bmatrix} a_{1,1}B & \cdots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \cdots & a_{m,n}B \end{bmatrix} \tag{A.1}$$

Given appropriately sized matrices A, B, C, and D such that all operations below are well-defined, we have the following equalities.

$$(A \otimes B)' = (A' \otimes B') \tag{A.2}$$

$$(A \otimes B)^{-1} = \left(A^{-1} \otimes B^{-1}\right) \tag{A.3}$$

$$(A \otimes B)(C \otimes D) = (AC \otimes BD) \tag{A.4}$$

We can also define the vectorization of an $[m \times n]$ matrix A, which is a linear transformation on a matrix that stacks the columns iteratively to form a long vector of size $[mn \times 1]$, as below.

$$\text{vec}[A] = \begin{bmatrix} a_{1,1} \\ \vdots \\ a_{m,1} \\ a_{1,2} \\ \vdots \\ a_{m,2} \\ \vdots \\ a_{1,n} \\ \vdots \\ a_{m,n} \end{bmatrix} \tag{A.5}$$

Using the vec operator, we can state the trivial definition below.

$$\text{vec}[A + B] = \text{vec}[A] + \text{vec}[B] \tag{A.6}$$

Combining the vec operator with the Kronecker product, we have the following.

$$\text{vec}[AB] = (B' \otimes \text{I})\,\text{vec}[A] \tag{A.7}$$

$$\text{vec}[ABC] = (C' \otimes A)\,\text{vec}[B] \tag{A.8}$$

We can express the trace of a product of matrices as below.

$$\text{trace}[AB] = \text{vec}[B']'\,\text{vec}[A] \tag{A.9}$$

$$\begin{aligned} \text{trace}[ABC] &= \text{vec}[B]'\,(\text{I} \otimes C)\,\text{vec}[A] && \text{(A.10a)} \\ &= \text{vec}[A]'\,(\text{I} \otimes B)\,\text{vec}[C] && \text{(A.10b)} \\ &= \text{vec}[A]'\,(C \otimes \text{I})\,\text{vec}[B] && \text{(A.10c)} \end{aligned}$$

For more information, please see [9].

Appendix B Analytic Block Representation for the inverse of a Saddle Point Matrix

M_S is a saddle point matrix if it has the block form below.[27]

$$M_S = \begin{bmatrix} A_S & B_S' \\ B_S & -C_S \end{bmatrix} \tag{B.1}$$

In the case that A_S is nonsingular and the Schur complement $J_S = -\left(C_S + B_S A_S^{-1} B_S'\right)$ is also nonsingular in the above equation, it is known that the inverse of this saddle point matrix can be expressed analytically by the following equation (see e.g., [2]).

$$M_S^{-1} = \begin{bmatrix} A_S^{-1} + A_S^{-1} B_S' J_S^{-1} B_S A_S^{-1} & -A_S^{-1} B_S' J_S^{-1} \\ -J_S^{-1} B_S A_S^{-1} & J_S^{-1} \end{bmatrix} \tag{B.2}$$

Appendix C Solution to the system $Mn = p$

Here we solve the system $Mn = p$ from Equations (3.30), (3.31), and (3.32), re-stated below, for vector n.

$$\begin{bmatrix} 2S_k \otimes \mathrm{I} & \nu_k \otimes A' \\ \nu_k' \otimes A & 0_{[q \times q]} \end{bmatrix} \begin{bmatrix} l \\ \lambda \end{bmatrix} = \begin{bmatrix} 0_{[mn \times 1]} \\ b - A\hat{x}_{k|k} \end{bmatrix} \tag{C.1}$$

M is a saddle point matrix with the following equations to fit the block structure of Equation (B.1).[28]

$$A_S = 2S_k \otimes \mathrm{I} \tag{C.2}$$
$$B_S = \nu_k' \otimes A \tag{C.3}$$
$$C_S = 0_{[q \times q]} \tag{C.4}$$

We can calculate the term $A_S^{-1} B_S'$.

$$A_S^{-1} B_S' = \left[2\left(S_k \otimes \mathrm{I}\right)\right]^{-1} \left(\nu_k' \otimes A\right)' \tag{C.5a}$$
$$\overset{\text{(A.2)(A.3)}}{=} \frac{1}{2}\left(S_k^{-1} \otimes \mathrm{I}\right)\left(\nu_k \otimes A'\right) \tag{C.5b}$$
$$\overset{\text{(A.4)}}{=} \frac{1}{2}\left(S_k^{-1} \nu_k\right) \otimes A' \tag{C.5c}$$

[27] The subscript S notation is used to differentiate these matrices from any matrices defined earlier.
[28] We use Equation (A.2) with B_S' to arrive at the same term for B_s in Equation (C.1).

And as a result we have the following for J_S.

$$J_S = -\frac{1}{2}\left(\nu_k' \otimes A\right)\left[\left(S_k^{-1}\nu_k\right) \otimes A'\right] \tag{C.6a}$$

$$\overset{(A.4)}{=} -\frac{1}{2}\left(\nu_k' S_k^{-1}\nu_k\right) \otimes \left(AA'\right) \tag{C.6b}$$

J_S^{-1} is then, as below.

$$J_S^{-1} = -2\left[\left(\nu_k' S_k^{-1}\nu_k\right) \otimes \left(AA'\right)\right]^{-1} \tag{C.7a}$$

$$\overset{(A.3)}{=} -2\left(\nu_k' S_k^{-1}\nu_k\right)^{-1} \otimes \left(AA'\right)^{-1} \tag{C.7b}$$

For the upper right block of M^{-1}, we then have the following expression.

$$A_S^{-1}B_S' J_S^{-1} = \left[\left(S_k^{-1}\nu_k\right) \otimes A'\right]\left[\left(\nu_k' S_k^{-1}\nu_k\right)^{-1} \otimes \left(AA'\right)^{-1}\right] \tag{C.8a}$$

$$\overset{(A.4)}{=} \left[S_k^{-1}\nu_k\left(\nu_k' S_k^{-1}\nu_k\right)^{-1}\right] \otimes \left[A'\left(AA'\right)^{-1}\right] \tag{C.8b}$$

Since the first block element of p is a vector of zeros, we can solve for n to arrive at the following solution for l.

$$\left(\left[S_k^{-1}\nu_k\left(\nu_k' S_k^{-1}\nu_k\right)^{-1}\right] \otimes \left[A'\left(AA'\right)^{-1}\right]\right)\left(b - A\hat{x}_{k|k}\right) \tag{C.9}$$

The vector of Lagrange Multipliers λ is given below.

$$-2\left[\left(\nu_k' S_k^{-1}\nu_k\right)^{-1} \otimes \left(AA'\right)^{-1}\right]\left(b - A\hat{x}_{k|k}\right) \tag{C.10}$$

References

[1] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, Inc., 2001.

[2] Michele Benzi, Gene H. Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[3] G.E.P. Box and G.M. Jenkins. Time Series Analysis. Forecasting and Control (Revised Edition). *Oakland: Holden-Day*, 1976.

[4] R. Fletcher. *Practical methods of optimization. Vol. 2: Constrained Optimization*. John Wiley & Sons Ltd., Chichester, 1981. Constrained optimization, A Wiley-Interscience Publication.

[5] Nachi Gupta. Kalman filtering in the presence of state space equality constraints. In *IEEE Proceedings of the 26th Chinese Control Conference*, July 2007, arXiv:physics.ao-ph/0705.4563, Oxford na-tr:07/14.

[6] Nachi Gupta, Raphael Hauser, and Neil F. Johnson. Forecasting financial time-series using an artificial market model. In *Proceedings of the 10th Annual Workshop on Economic Heterogeneous Interacting Agents*, June 2005, Oxford na-tr:05/09.

[7] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, volume 3, pages 182–193, 1997.

[8] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[9] Peter Lancaster and Miron Tismenetsky. *The Theory of Matrices: With Applications*. Academic Press Canada, 1985.

[10] A. G. Qureshi. Constrained kalman filtering for image restoration. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1405 – 1408, 1989.

[11] D. Simon and D.L. Simon. Aircraft turbofan engine health estimation using constrained kalman filtering. *Journal of Engineering for Gas Turbines and Power*, 127:323, 2005.

[12] Dan Simon and Tien Li Chia. Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1):128–136, January 2002.

[13] Taek L. Song, Jo Young Ahn, and Chanbin Park. Suboptimal filter design with pseudomeasurements for target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 24(1):28–39, January 1988.

[14] Minjea Tahk and Jason L. Speyer. Target tracking problems subject to kinematic constraints. *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 1058–1059, 1988.

[15] K. C. Toh, M. J. Todd, and R. Tutuncu. SDPT3 — a Matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

[16] L. S. Wang, Y. T. Chiang, and F. R. Chang. Filtering method for nonlinear systems with constraints. *IEE Proceedings - Control Theory and Applications*, 149(6):525–531, November 2002.