Kalman Filter with Linear State Constraints

Madison McCarthy
Department of Electrical and Computer Engineering,
University of Western Ontario, London, ON, N6A 5B9, Canada
E-mail: mmccar25@uwo.ca

***Abstract*—The Kalman filter is an unbiased minimum-variance linear state estimator. Although powerful, the Kalman filter is unable to incorporate known state constraints which may allow for improved estimation performance. This paper aims to review four different methods for applying linear state constraints with the Kalman filter and one method to deal with nonlinear state constraints. Several simulations were used to evaluate the performance of the linear state constraint methods, with the results detailing the advantages and disadvantages of each method.**

***Keywords*—Kalman Filter; Minimum-Variance; State Constraints; Estimation**

I. Introduction

The Kalman filter is well supported in academia due to its excellent performance as an unbiased minimum-variance linear state estimator [1], and well supported in industry due to the relatively simple and complete algorithm ideal for real-time applications. Although powerful, the simplicity of the algorithm can also be considered as one of its weaknesses; often times a problem that can be solved with the Kalman filter may need to neglect information that cannot be readily captured within the confines of the algorithm. To overcome this weakness, many flavors of the Kalman filtering algorithm have been formulated, each allowing for additional system information to be leveraged in the state estimates.

One classification of information that cannot be readily captured within the original Kalman filtering algorithm is state constraints (these restrictions are commonly found in controls applications). Although the vanilla Kalman filter will converge to the minimum-variance state estimate if properly formulated, the incorporation of known state constraints could improve the filter's performance.

This paper aims to review four different variations of the Kalman filtering algorithm that are capable of incorporating linear state constraints into the state estimations. As well, a simple process to deal with nonlinear state constraints is also mentioned to provide the reader with a small taste for more advanced topics.

The rest of this paper is divided into five additional sections: following the introduction is Section II which covers some background related to Kalman filtering, state constraints, and 2-dimensional (2-D) tracking problems. Next is Section III which details four different linearly constrained Kalman filtering algorithms. Following is Section IV which briefly notes a technique to deal with nonlinear state constraints. Section V then discusses the results of several simulations performed with the four linearly constrained algorithms. Finally, Section VI completes the paper with a conclusion.

II. Background

Prior to diving into linearly constrained Kalman filtering methods, we will first review the Kalman Filtering algorithm, state constraints, and issues associated with 2-D tracking problems. This background will provide the nomenclature used to explain different Kalman filtering algorithms and setup the different scenarios that will be used for the simulations in Section V.

A. The Kalman Filter

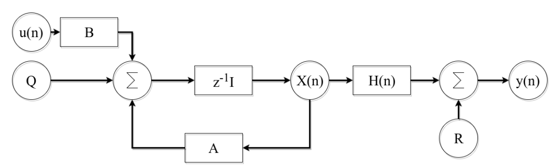

Fig. 1: System model with open loop control.

By examining the system model described by Fig. 1, the following two system equations (the process equation and the measurement equation) can be extracted:

$$x_{n+1} = Ax_n + Bu_n + Q, \tag{1}$$

$$z_n = Hx_n + R. \tag{2}$$

In this discrete model, n is the time-step, x is the states vector, A is the state transition matrix, B is the control matrix, u is the control vector, Q is the process noise, H is the measurement matrix, z is the measurement vector, and R is the measurement noise. Note that both Q and R are assumed to be Gaussian.

Using this model, the Kalman filter is derived to be

$$X_{predicted} = Ax_{n-1} + Bu_n, \tag{3}$$

$$P_{predicted} = AP_{n-1}A^T + Q, \tag{4}$$

$$y = z_n - Hx_{predicted}, \tag{5}$$

$$S = HP_{predicted}H^T + R, \tag{6}$$

$$K = P_{predicted} H^T S^{-1}, \quad (7)$$

$$X_n = X_{predicted} + Ky, \quad (8)$$

$$P_n = (I - KH) P_{predicted}, \quad (9)$$

where $P_{predicted}$ is the *a priori* covariance estimation error, P_n is the *a posteriori* covariance estimation error, y is the innovation, S is the covariance of the innovation, K is the Kalman gain, $X_{predicted}$ is the *a priori* state estimates, and X_n is the *a posteriori* state estimates.

The initialization for the estimated states, X_0, can be set to an arbitrary value if the state's starting conditions are not known. The initial covariance estimation error P_0 should be initialized to

$$P_0 = \begin{bmatrix} G & 0 & 0 & 0 \\ 0 & G & 0 & 0 \\ 0 & 0 & G & 0 \\ 0 & 0 & 0 & G \end{bmatrix},$$

which is an identity matrix scaled by G. If the initial estimate for the states is arbitrarily chosen or known to have a large error, G should be large in magnitude to increase the speed of convergence. If the initial estimation is known to be accurate, G should be small in magnitude.

B. State Constraints

The system dynamics described by (1) and (2) could be subject to additional state constraints that are either linear or nonlinear. These constraints can then be further divided into four subcategories: time-invariant equality constraints, time-variant equality constraints, time-invariant inequality constraints, and time-variant inequality constraints.

Linear time-invariant equality constraints can be in the form

$$Dx_n = d, \quad (10)$$

while the linear time-invariant inequality constraint can be in the form

$$Dx_n \leq d. \quad (11)$$

In (10) and (11), D represents that state constraint matrix and d represents the linear state constraint vector. Linear time-variant equality constraints can take the form

$$D(n)x_n = d(n), \quad (12)$$

and linear time-variant inequality constraints can take the form

$$D(n)x_n \leq d(n). \quad (13)$$

Notice that linear state constraints can be represented as a set of linear equations within a matrix which eases integration with the Kalman filtering algorithm.

Contrasting the linear state constraints is the nonlinear state constraints. Nonlinear time-invariant equality constraints can be represented by

$$g(x_n) = h, \quad (14)$$

and nonlinear time-invariant inequality constraints can be represented by

$$g(x_n) \leq h. \quad (15)$$

For nonlinear constraints, $g(x_n)$ is a set of nonlinear functions and h is the nonlinear state constraint vector. Nonlinear time-variant equality constraints are represented by

$$g(n, x_n) = h(n), \quad (16)$$

and nonlinear time-variant inequality constraints are represented by

$$g(n, x_n) \leq h(n). \quad (17)$$

Since the nonlinear state constraints fail the superposition principle, implementing these constraints within the Kalman filtering algorithm pose a greater challenge than linear state constraints.

C. 2-Dimensional Tracking Problems

Similar to any Kalman filtering application, *a priori* information about the system's dynamics as well as any uncertainties are required for 2-D tracking problems. This information includes the state transition matrix, any control influencing the system, and any process noise modeled as a Gaussian distribution. Measurement information is also required, including a model for the measurement noise.

Typical 2-D tracking problems are represented with four different states: the X position, the X velocity, the Y position, and the Y velocity. The measurements taken for these four states can then apply to just the position of the object being tracked, just the velocities with an accurate initial position estimation, or some other combination of state measurements. Depending on the complexity of the problem, external forces acting upon the object can either be captured within the state transition matrix or through external control.

The previous description for a 2-D tracking problem is rather simplistic; additional complexities about the system can also be captured within the state-transition matrix or within the uncertainty models. Although most system dynamics can be incorporated within (1) and (2), some behaviors such as known state constraints cannot be so easily incorporated. If, for example, an object's motion was limit to travel within a known, confined region (something representable through an inequality constraint), how would one include this limitation within the state-transition matrix? If an object is assumed to be traveling on a predefined path (something representable through an equality constraint), is there a way to include this restriction within the measurement matrix?

One approach for solving these issues is through modifications of the Kalaman filtering algorithm which allow for known state constraints to be utilized. This approach will be discussed in Section III and Section IV for linear and nonlinear state constraints respectively. For now, several test cases will be described to illustrate several 2-D constrained tracking problems.

D. Test Scenario #1

Three test scenarios have been developed for the previously posed questions. These scenarios will be used during simulations for different linearly constrained Kalman filtering algorithms.

The first test scenario describes the dynamics of an object traveling with a fixed angle. Fig. 2 illustrates this first scenario which will be denoted as Test #1.

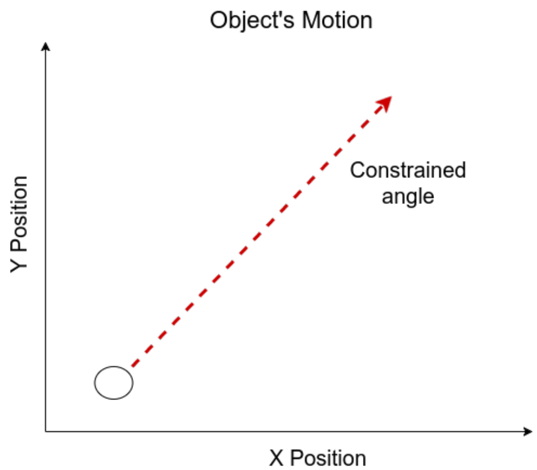

Fig. 2: Test #1: Object's motion for a 2-D tracking problem with a constrained angle.

To represent Test #1 as an unconstrained Kalman filter, the following parameters are set to

$$x_n = \begin{bmatrix} X \\ V_x \\ Y \\ V_y \end{bmatrix}, \quad A = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$u = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 90 & 0 \\ 0 & 90 \end{bmatrix},$$

$$Q = \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \quad P = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}.$$

Note that for all test scenarios, including Test #1, the only available measurements are of the X and Y position (as noted in matrix H). As well, the same, arbitrarily chosen process and measurement noise are applied to all scenarios. Also an identical, incorrect initial estimate was chosen for all scenarios which tend to a relatively large state covariance error P_0.

A time-invariant equality constraint is capable of capturing the limited motion of the object illustrated in Fig. 2. For this particular scenario, the equality constraint is described as

$$\begin{bmatrix} 0 & 1 & 0 & -\frac{15}{20} \end{bmatrix} x_n = 0. \tag{18}$$

E. Test Scenario #2

The second test scenario, denoted as Test #2, is for an object traveling along a constrained, predefined path. For simplicity, this path is a bend followed by a straight line. Fig. 3 illustrates this described motion.

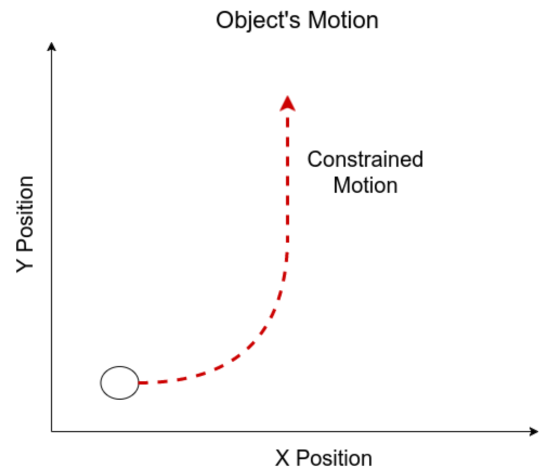

Fig. 3: Test #2: Object's motion for a 2-D tracking problem with a constrained path.

Only minor modifications need to be made to the previously detailed Kalman filter parameters in Test #1. These changes affect the control dynamics and can be seen below:

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad u = \begin{bmatrix} 0 & 0 & 0.5 * 4 * dt^2 & 4 * dt \end{bmatrix}.$$

The altered control dynamics are only applicable during the first half of the object's trajectory when the velocities are changing at a fixed rate. Then the object's motion is fixed to a constrained angle and the control matrix B is set to $\mathbf{0}$.

The restricted object's motion can be captured as a time-variant equality constraint described by

$$\begin{bmatrix} 0 & 1 & 0 & \frac{-15}{20+i*4-4} \end{bmatrix} x_n = 0 \tag{19}$$

for the first half of the trajectory, and

$$\begin{bmatrix} 0 & 1 & 0 & \frac{-15}{20+\frac{L}{2}*4-4} \end{bmatrix} x_n = 0 \tag{20}$$

for the second half of the trajectory, where i and L represents the current number and total number of time-steps for the simulation respectively.

F. Test Scenario #3

The final test scenario, denoted as Test #3, is for an object's motion that is confined to a travel within a positional bound on the 2-D plane. This restriction has been illustrated in Fig. 4.

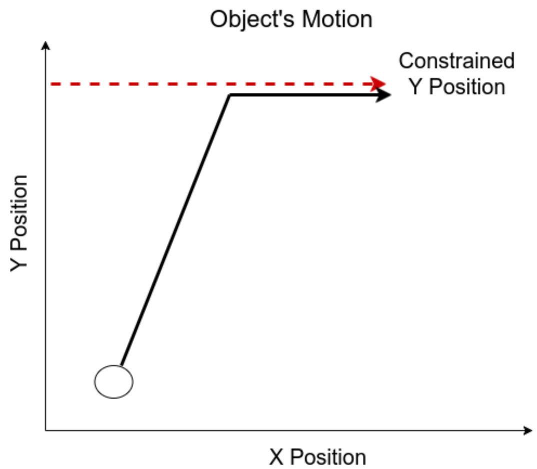

Fig. 4: Test #3: Object's motion for a 2-D tracking problem with a positional bound.

No additional modifications are required to the Kalman filter parameters detailed in Test #1 to capture this positional bound on the system's dynamics. The bound can however be represented through a time-invariant inequality constraint in the following form:

$$\begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix} x_n \leq 300. \tag{21}$$

III. Linear Constraint Methods

In this section, four different linearly constrained Kalman filter methods will be discussed. Each of these methods can incorporate linear constraints into the original, unconstrained Kalman filtering algorithm, creating an alternate constrained algorithm.

The first method that will be examined is the Perfect Measurement [1] method, with an explanation of the Model Reduction [1] method following. Next, the Estimation Projection [2] method will be shown. Last is the Probability Density Function (PDF) Truncation [3] method. Each of these methods was implemented with various MATLAB® simulations which will be discussed in each of the corresponding method's subsection.

A. Perfect Measurement

The first method, called the Perfect Measurement method, incorporates linear constraints into the Kalman filtering algorithm by treating the constraint as a pseudo-measurement. That is, we augment (2) with the linear constraint. In this paper, a pseudo-measurement describes a measurement that may not be directly observable about the system. The information in the constraint can however be treated as additional measurement in the Kalman filtering algorithm, thus the term pseudo.

By examining the measurement equation, it should be clear as to why this method is only applicable for equality constraints. Since the measurement equation is in the form of an equality, an inequality constraint cannot be adequately represented.

To apply this method, the measurement equation is augmented with a linear equality constraint,

$$\begin{bmatrix}y_n \\ d\end{bmatrix} = \begin{bmatrix}H \\ D\end{bmatrix} x_n + \begin{bmatrix}r_n \\ 0\end{bmatrix}. \tag{22}$$

Notice that in (22) there is no noise associated with the equality constraints, thus creating the perfect measurement. While in theory it should be idealized that the measurement is indeed perfect, numerical problems may ensue from the noiseless implementation. To deal with these numerical issues, a small scalar should be chosen for the noise.

The perfect measurement method was implemented through MATLAB® simulations. Of the three scenarios previously described, Test #1 and Test #2 were chosen for implementation with this method (a time-invariant equality constraint and a time-variant equality constraint). The results of these simulations will be presented in Section V.

B. Model Reduction

The second Kalman filtering method that can capture linear constraints is the Model Reduction method. This method aims to reduce a constrained Kalman filtering problem into an unconstrained problem that encapsulates the known constraint. Depending on how the linear constraint is formulated, it is possible to reduce the number of states associated with the system model through various substitution.

Using Test #1 as an example, we can show that the four states can be reduced to a mere three states that will always produce a feasible solution. The process equations for Test #1 are

$$\dot{x_1} = x_1 + dt * x_2 + q_1, \tag{23}$$

$$\dot{x_2} = x_2 + q_2, \tag{24}$$

$$\dot{x_3} = x_3 + dt * x_4 + q_3, \tag{25}$$

$$\dot{x_4} = x_4 + q_4, \tag{26}$$

whereas the equality constraint is described as

$$x2 = \frac{15}{20} x4. \tag{27}$$

By substituting (27) into (23) and (24), the dependency upon state x_2 can be eliminated. The reduced set of states is then

$$x_n = \begin{bmatrix}X \\ Y \\ V_y\end{bmatrix}, \qquad A = \begin{bmatrix}1 & 0 & \frac{15}{20}dt \\ 0 & 1 & dt \\ 0 & 0 & 1\end{bmatrix}.$$

Again, it should be clear as to why this method is only applicable with equality constraints since the linear sets of equations being reduced are all of the equality form.

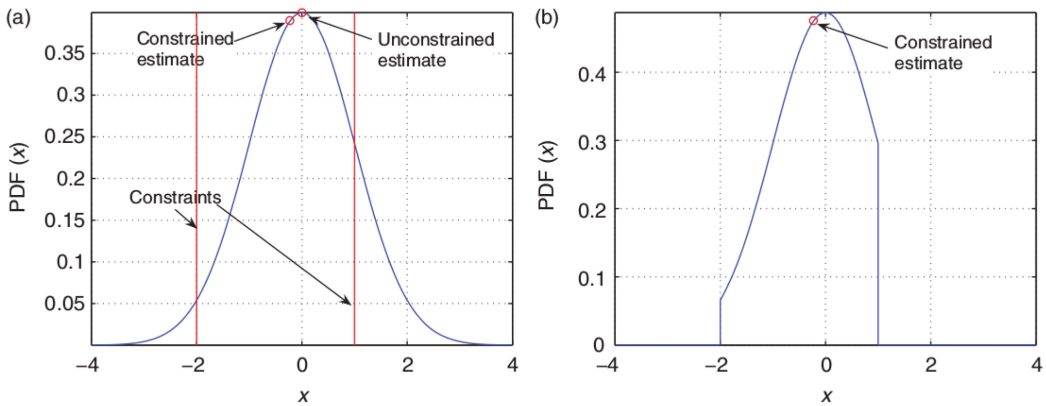

Fig. 5: Example of PDF Truncation. a) PDF prior to truncation. b) PDF after truncation. [3]

The Model Reduction method holds an advantage over typical flavors of the Kalman filtering algorithm as it actually decreases the computational overhead of the algorithm rather than increasing it. This key advantage comes at a heavy price however, as the algorithm may no longer be estimating several states of interest. Depending on the form and number of equality constraints used during the reduction process, the state-transition matrix can also lose meaning, putting the Model Reduction method at a further disadvantage.

Similar to the Perfect Measurement method, Test #1 and Test #2 were chosen as the scenarios for the implementation.

C. Estimation Projection

The third method for Kalman filtering with linear constraints is the Estimation Projection method. This method operates as an additional step in the unconstrained Kalman filtering algorithm. Since the estimated states provided by the unconstrained algorithm may not be feasible (they may not satisfy the linear constraints), these estimations need to be projected onto the feasible region define by the constraints. After the projection, the resultant state estimates represent the constrained solution.

Estimation Projection is implemented by projecting the estimated states of the unconstrained solution onto the constrained subspace,

$$x_{proj} = x_n - PD'(DPD')^{-1}(Dx_n - d). \tag{28}$$

Since the Estimation Projection method is executed after the initial unconstrained estimate, the constrained subspace can vary with each iteration's projection. The advantage of this flexibility is that an inequality constraint could be incorporated into the algorithm when the constraint is active, and removed when the constraint is inactive. This algorithm is therefore capable of including both equality constraints and inequality constraints on the condition that *a priori* information is known regarding when the inequality constraint is active.

To evaluate the performance of the Estimation Projection method Test #1 and Test #3 were chosen as the scenarios. Test #1 was chosen for a common grounds comparison between all of the algorithms and Test #3 was selected to demonstrate an inequality constraint.

D. Probability Density Function Truncation

The final constrained method implemented was PDF truncation. This method leverages the fact that the original unconstrained Kalman filtering algorithm relies upon Gaussian PDFs for each state estimate. These PDFs can be truncated by each state constraint to produce new PDFs which only incorporates the feasible regions.

To better explain this method, Fig. 5 illustrates the truncation of a PDF [3]. Notice how, in this example, the unconstrained and the constrained estimate differ even though the inequality constraint that truncates the PDF is inactive. This is a result of a bias that is introduced by the PDF Truncation method. Since a bias is being introduced into the state estimates, it should be expected that this method cannot produced the optimal solution unless if this bias is insignificantly small.

To implement this method, several complex steps need to be iterated through for each constraint after the initial unconstrained state estimate. For a more detailed explanation, refer to [3]. The simplified version of this algorithm boils down to 6 steps:

1) Transform the first constraint into a normalized, scalar constraint using Jordan canonical decomposition and Gram-Schmidt orthogonalization.
2) Remove the part of the Gaussian PDF that is outside of the constrained solution.
3) Normalize the truncated PDF.
4) Determine the mean μ and covariance σ^2 of the transformed state estimate after the constraint enforcement.
5) Determine the mean X_n and covariance P_n of the state estimates after the enforcement of the constraint.
6) Repeat step 1-5 for each of the following constraints using the new mean and covariance.

Although the mathematics to implement this method are computationally expensive, a counteracting advantage is the ability to include inequality constraints without *a priori* knowledge regarding when/if the constraint becomes active.

Test #1 and Test #3 were the chosen scenarios for the PDF Truncation method with similar reasons as the Estimation Projection method.

IV. Nonlinear Constraint Methods

The four previously discussed state constraint methods are only applicable for linear constraints as they rely upon minor variations of the original unconstrained Kalman filter. Since the original Kalman filter deals with linear equations in state space, nonlinearities are incompatible. There are however methods to incorporate nonlinearities within the Kalman filter through approximation with the Taylor Series Expansion. The Extended Kalman filter (EKF) is one method to approximate the nonlinearities within the measurement and process equation through the mentioned Taylor Series Expansion.

The EKF is not the highlight of this section but rather the inspiration for a method to deal with nonlinear state constraints. Similar to linearizing the process and measurement equation with the Taylor Series Expansion, state constraints too can be linearized to produce an approximation of the constraint's behavior in a compatible format with linear Kalman filtering algorithms.

A. State Constraint Linearization

In the simplest form, state constraints can be linearized by computing the Taylor Series Expansion about some point, $\hat{x}_n$, and neglecting the second and higher order terms as described by

$$g(x) \approx \sum_{n=0}^{1} \frac{g^n(0)}{n!} = g(0) + g'(0)x. \tag{29}$$

Although $\hat{x}_n$ could be chosen to be the initial estimated states for the Kalman filter and fixed at that approximation, a better approach is to linearize the state constraint during each iteration of the Kalman filter about the previously estimated states.

This approximation for the nonlinear constraint may not be as accurate as one attained through the use of higher order terms in the Taylor Series Expansion or more advanced nonlinear state constrained algorithms. Instead this method's advantage is with minimal additional computational overhead added to the Kalman filtering algorithm.

With the resultant approximation in (29), we are able to derive the linear constraint parameters, D and d. These two parameters are

$$D = g'(\hat{x}_n), \tag{30}$$

$$d = h - g(\hat{x}_n) + g'(\hat{x}_n)\hat{x}_n. \tag{31}$$

Using (30) and (31), the linear equality or inequality constraint can then be formulated and any of the previously discussed linearly constrained Kalman filter algorithms can be applied.

V. Results

To evaluate the performance of the different linearly constrained methods, each method was simulated in the corresponding test scenario detailed in Section III. The original unconstrained Kalman filtering algorithm was also implemented for each test scenario to act as the baseline. Two metrics for evaluation were chosen: the first is the mean-squared error (MSE) of the estimated state errors which is the objective function for the Kalman filtering algorithm. The second metric was computation time per algorithm iteration. Since software and hardware used for measuring performance can vary drastically, ratios are used to compare the speed of the algorithms, with the unconstrained Kalman filter being the baseline ratio. The second metric was only used to evaluate Test #1 and it is assumed that these ratios will roughly hold true across different simulations. Each of the test scenarios ran for 100 iterations to better represent the ensemble averaged performance. The MATLAB® code for these simulations is available at https://github.com/McCarthyMadison/Kalman-Filter-Constrained.

A. Test Scenario #1

The first scenario is a linear time-invariant equality constraint. All four constrained algorithms and the unconstrained algorithm were simulated. A sample of the simulation for the 5 algorithms estimating the location of the object can be seen in Fig. 6. Take note that all of the constrained Kalman filtering algorithms converge to the optimal solution quicker than the unconstrained solution. As well, the estimations for the Perfect Measurement method and the Model Reduction method overlap exactly which implies that they are mathematically identical in this simulation.

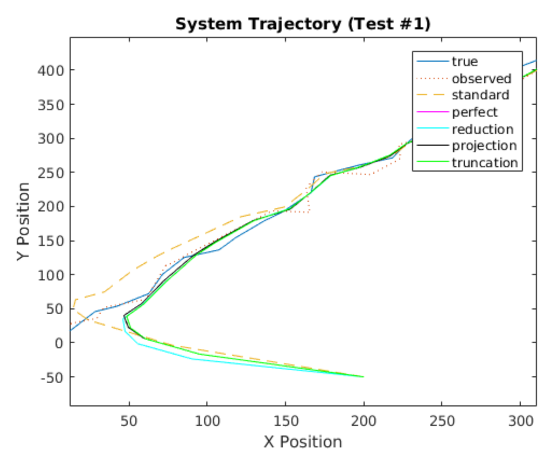

Fig. 6: Test #1: Tracking of the system's trajectory.

To discern the fine details about the different algorithm's performances, Table I shows the ensemble-averaged performance. Notice that all of the constrained methods perform better than the unconstrained method with near identical MSE values. As expected, the Model Reduction method reduces the algorithms computation time, while the PDF Truncation method drastically increased the computation time.

	Normal	Perfect	Reduction	Projection	Truncation
MSE (dB)	5.64	5.48	5.48	5.47	5.48
Estimation Time (ratio)	1.00	1.01	0.91	1.28	4.11

TABLE I: Test #1: Performance of the algorithms.

B. Test Scenario #2

The second test scenario dealt with a linear time-variant equality constraint. Only the unconstrained algorithm, the Perfect Measurement method, and the Model Reduction method were implemented with this test. Fig. 7 shows these three algorithms estimating the trajectory for this simulation. Again, notice that the constrained methods converge quicker than the unconstrained method. A key difference in this test when compared to the previous is that the Perfect Measurement method and the Model Reduction method no longer operate with identical performance.

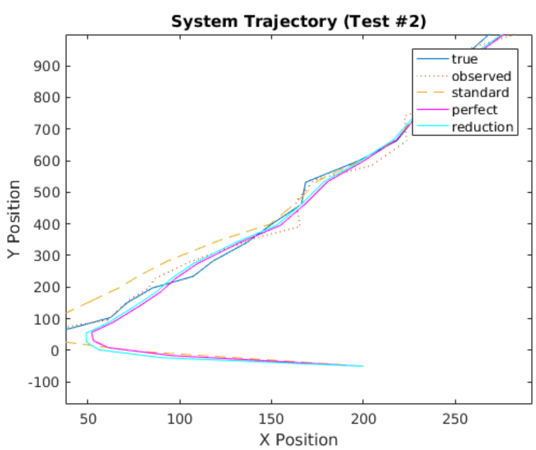

Fig. 7: Test #2: Tracking of the system's trajectory.

The results for the performance of these three algorithms against the first metrics can be seen in Table II. In this scenario, the Model Reduction method had the best performance overall.

	Normal	Perfect	Reduction
MSE (dB)	5.69	5.49	5.40

TABLE II: Test #2: Performance of the algorithms.

C. Test Scenario #3

The final scenario involved a time-invariant inequality constraint. The unconstrained algorithm, the Estimation Projection method, and the PDF Truncation method were simulated for this scenario. The tracking of the system trajectory can be seen in Fig. 8. Notice how all three algorithms have identical performance prior to the inequality constraint becoming active.

A particular detail worth noting in Fig. 8 is the PDF Truncation methods bias. This bias forces an offset in the estimated Y position to always be less than the inequality bound, preventing the PDF truncation method from performing as well as alternate constrained methods for inequality constraints. This result is shown in Table III.

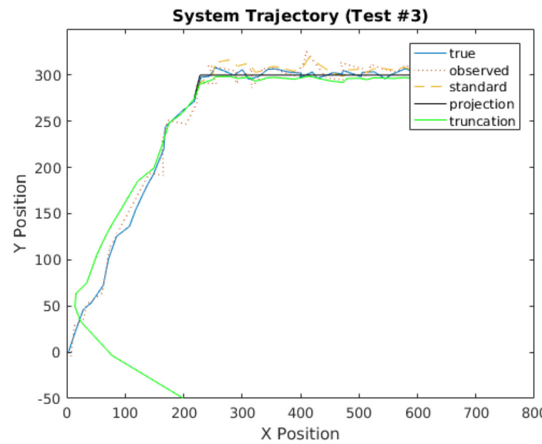

Fig. 8: Test #3: Tracking of the system's trajectory.

	Normal	Projection	Truncation
MSE (dB)	5.88	5.58	5.66

TABLE III: Test #3: Performance of the algorithms.

VI. Conclusion

This paper has demonstrated four different Kalman filtering methods that can incorporate linear state constraints. Each of these methods improved the performance of the unconstrained Kalman filter by reducing the MSE for the predicted states. The improved performance by each method is situational dependent however, and the disadvantages of each method should be considered prior to implementation. Nonlinear constraints can also be incorporated in the Kalman filter algorithm by leveraging the Taylor Series Expansion. Although quick, linearization of the constraints is only an approximation which may introduce additional estimation errors. These additional induced errors can be avoided through more advanced nonlinear state constraint methods if needed.

References

[1] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *IET Control Theory Applications*, vol. 4, no. 8, pp. 1303–1318, August 2010.

[2] D. Simon and Tien Li Chia, "Kalman filtering with state equality constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 128–136, Jan 2002.

[3] Dan Simon and Donald L. Simon, "Constrained kalman filtering via density function truncation for turbofan engine health estimation," *International Journal of Systems Science*, vol. 41, no. 2, pp. 159–171, 2010.