JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 39, No. 11, November 2016

Engineering Notes

Interpolation Method for Update with Out-of-Sequence Measurements: The Augmented Fixed-Lag Smoother

Hyosang Yoon,* David C. Sternberg,† and Kerri Cahoy‡
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

DOI: 10.2514/1.G001800

Nomenclature

F	=	state transition matrix
H	=	matrix giving the ideal/noiseless connection between measurements and states
I	=	identity matrix
k	=	time index
L	=	noise sensitivity matrix in nonlinear systems
N	=	lag time constant integer
P	=	estimation-error covariance matrix
s	=	arbitrary time step used in inductive proof
x_k	=	state at time index k
$\hat{x}$	=	state estimate
$\hat{x}_k^-$	=	a priori state estimate
$\hat{x}_k^+$	=	a posteriori state estimate
y_k	=	measurement at time index k

I. Introduction

THE Kalman filter has been one of the most powerful tools in the field of estimation and tracking since it was published in 1960 [1]. Besides its optimality, its recursive formulation is another key benefit, especially for real-time applications. When considering applications to real-time systems, one must be aware of the timing differences that naturally occur between a sensor measurement, the state of interest, and the assumptions made by the selected Kalman filter or smoother. For real-time systems, many choose the state of interest to be the system state at the current time, though the estimator is very likely to have access to prior sensor data that were used in determining the current state estimate. The delay between measurement and estimate times may be larger if the sensor is part of another system that has digital processing to generate more accurate outputs as compared to simple analog sensors. For example, a star sensor, which is currently one of the most accurate attitude sensors for spacecraft, needs time to integrate starlight and process the image, to calculate the attitude, and to transfer the output to the computer that runs the Kalman filter. These effects are compounded by the need to compensate for time synchronization errors between commercial-off–the-shelf technologies (COTS). As COTS systems have become more commonly used in aerospace vehicles, such as CubeSats [2] or quadcopter drones [3], the time synchronization typically becomes less precise. For example, some COTS controllers are not typically sufficient for use in higher-performance systems because of their drift and lower speeds or their lack of compensation for thermal or pressure effects. The PIC16F627A/628A/648A, for example, is a COTS microprocessor used in small satellites and has an internal oscillator that, while typically 4 MHz, can range between 3.80 and 4.20 MHz under certain conditions [4]. This scenario leads to a configuration where it becomes more difficult to force or anticipate the exact measurement time.

To update the Kalman filter with the time-delayed sensor output, both the state and covariance estimate at the measurement time are needed. If the time delay can be synchronized with the Kalman filter update history, a smoothing algorithm can be used with only small changes to the measurement update equation, since the state estimate at the measurement time is available immediately. If not, to ensure optimality, the system must generate the estimates that are conditioned with all of the measurements. There have been several approaches to managing this problem. A suboptimal filtering method was described in [5,6], where Blackman and Popoli [5] described a process for simplified covariances associated with state retrodiction, and Hilton et al. [6] described a constrained minimum mean square error negative time update technique for highly varying states with limited data rates. They are suboptimal because they only partially take into account the process noise within the arbitrary delayed time. Bar-Shalom derived an exact solution when out-of-sequence measurements (OOSMs) were within the last sampling interval [7]. Though Bar-Shalom briefly mentioned the generalization of his method at the end of [7], it was not derived in the paper. Beyond the suboptimal solutions and Bar-Shalom's optimal solution for special cases, several general solutions that guaranteed optimality have been proposed. Zhang et al. proposed a method to save the innovation history for the state estimate and the covariance, and applied them to generate the state estimate and the covariance node at the time when OOSMs arrived [8]. For convenience, we will use the term "node" to represent the paired state estimate and covariance at a given time. Zhang and Bar-Shalom summarized the existing algorithms and proposed three methods for optimal update with OOSMs based on the complete in-sequence information (CISI) approach [9]. The concept of CISI is basically similar to [8], as it needs sequential updates from the OOSM time to the current time to obtain the OOSM node.

In this study, we propose a novel method to handle OOSMs in Kalman filtering. The proposed method, called the augmented fixed-lag smoother (AFLS), is based on the fixed-lag smoother (FLS) formulation, which has been shown to be optimal [10]. We generate the OOSM node from the two adjacent nodes, plug the generated estimations into the state vector and the covariance matrix, and update the filter with OOSMs using the FLS update equation. This approach gives a generalized solution that can handle any number of OOSMs. We also extend the AFLS algorithm to nonlinear system, called the extended AFLS (EAFLS), and give an application example on a satellite-tracking problem.

II. Conventional Kalman Filter Formulations

This section reviews two conventional Kalman filtering approaches (the discrete Kalman filter and the fixed-lag smoother FLS) to introduce the notation used in this study. The basic discrete

Presented as Paper 2016 at the AIAA Guidance, Navigation, and Control Conference, San Diego, CA, 4–8 January 2016; received 22 October 2015; revision received 9 April 2016; accepted for publication 15 July 2016; published online 14 September 2016.

*Doctor of Philosophy Candidate, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue. Student Member AIAA.

†Doctor of Science Candidate, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue. Student Member AIAA.

‡Assistant Professor, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue.

filter assumes that all of the input data arrive to the filter at the precise time step of the filter itself. The FLS for synchronized time delay measurement, however, relaxes this constraint to allow for input data arriving to the filter with a fixed time delay. In the next section, we describe the AFLS approach, which also relaxes the constraint on having a fixed time delay.

A. Discrete Kalman Filter

The Kalman filter can take many forms but, for implementation on a digital computer, discrete versions are used in order to take advantage of real time processing with inherently time-sequenced measurement updates. A discrete system can be written with the vectors w_k and v_k representing white, zero-mean, and uncorrelated random variables. Further details may be found in [11]:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ E(w_k w_j^T) &= Q_k\delta_{k-j}, \quad E(v_k v_j^T) = R_k\delta_{k-j}, \quad E(w_k v_j^T) = 0 \end{aligned} \tag{1}$$

The discrete-time Kalman filter can be implemented as a loop using the following steps:

1) Initialize the Kalman Filter with

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \end{aligned} \tag{2}$$

2) Iterate the time-step counter k as integers starting with $k = 1$:
Priori estimate:

$$\begin{aligned} \hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \\ P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \end{aligned} \tag{3}$$

Posteriori estimate:

$$\begin{aligned} K_k &= P_k^-H_k^T(H_kP_k^-H_k^T + R_k)^{-1} = P_k^+H_k^TR_k^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H\hat{x}_k^-) \\ P_k^+ &= (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T = (I - K_kH_k)P_k^- \end{aligned} \tag{4}$$

We are going to drop the input term $G_{k-1}u_{k-1}$ in the following sections for convenience.

B. Fixed-Lag Smoother

This section briefly describes the concept of the FLS, which was also described in more detail in [11], to introduce the base structure of the AFLS that will be derived in the following sections. The FLS is based on obtaining a new state estimate at time k-N, where the time index k iterates forward in time but the delay time for measurement inclusion N is a constant. Let us consider an augmented system as the following:

$$\begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-N-1} \end{bmatrix} = \begin{bmatrix} F_{k-1} & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ x_{k-2} \\ \vdots \\ x_{k-N-2} \end{bmatrix} + \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} w_k \tag{5}$$

$$y_k = [H_k \quad 0 \quad \cdots \quad 0] \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-N-1} \end{bmatrix} + v_k \tag{6}$$

For convenience, we define $x_{k,m}$ as the state x_{k-m} propagated with an identity transition matrix and zero process noise to time k. With this definition, the augmented system can be reformulated as follows:

$$\begin{aligned} X_k &= A_{k-1}X_{k-1} + W_{k-1} \\ y_k &= C_kX_k + v_k \end{aligned} \tag{7}$$

where

$$\begin{aligned} X_k &= [x_k \;\; x_{k,1} \;\; \cdots \;\; x_{k,N+1}]^T, \;\; A_{k-1} = \begin{bmatrix} F_{k-1} & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix}, \\ C_k &= [H_k \;\; 0 \;\; \cdots \;\; 0] \end{aligned} \tag{8}$$

$$\begin{aligned} &E(W_kW_j^T) = \mathrm{diag}(Q_k\delta_{k-j}, 0, \ldots, 0), \quad E(v_kv_j^T) = R_k\delta_{k-j}, \\ &E(W_kv_j^T) = 0 \end{aligned} \tag{9}$$

Then, it is possible to apply the discrete Kalman filter [Eqs. (3) and (4)] to this augmented system to obtain an optimal estimate for $(x_k, \ldots, x_{k-N-1})$ conditioned with all the measurements up to y_k. The FLS is designed for estimating the past states within fixed lag time steps with all the available measurements but not for estimating the current state. However, we can take advantage of the optimal estimates of the past states and the covariances to update the current state, which is described in the following section.

III. Update with Out-of-Sequence Measurement

Figure 1 shows a time-delayed measurement in the Kalman filtering described in Sec. II.A. The Kalman filter estimates the states at t_a, t_b, and t_c with the measurements y_a, y_b, and y_c. Then, a time-delayed measurement, y_r measured at t_r, is delivered at t_c. This section deals with the time-delayed measurement y_r to obtain the optimal current state estimate $\hat{x}_c^+$ conditioned by all the measurements available: y_a, y_b, y_c, and y_r.

A. In-Sequence Processing

The easiest and simplest method to incorporate an OOSM is through the in-sequence processing (ISP) of all measurements. First, all the state estimate, covariance, and measurement histories from the ordinary Kalman filter described in Sec. II.A up to the current time can be stored in memory. When an OOSM arrives, the algorithm can go back to the time just before the OOSM, discard all the state estimates and covariance histories after the time of the OOSM, and process the Kalman filter through to the current state. Though this method guarantees the optimal estimation of the current state and covariance conditioned with all the measurements received, this approach is a computationally inefficient process because these OOSM measurements can occur frequently and result in excessive delays in providing updated state estimates.

B. FLS with Synchronized Time-Delayed Measurement

The FLS described in Sec. II.B and in [11] assumes the measurement without time delay y_k. However, the FLS can be easily modified to process a synchronized time-delayed measurement or in-sequence measurement (ISM), where the time delay is known and

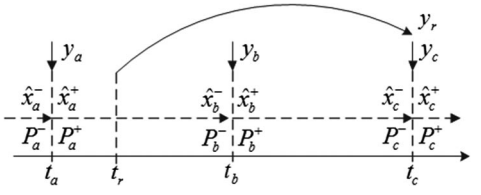

Fig. 1 Time-delayed measurement in Kalman filtering.

matches with the filter updates [10]. If a measurement is delivered at time k but is measured at time $k-l$ where $l < N$, then the output [Eqs. (7) and (8)] may be modified as the following:

$$y_{k,l} = C_{k,l}X_k + v_{k,l}$$
$$C_{k,l} = [0 \quad \cdots \quad 0 \quad H_{k-l} \quad 0 \quad \cdots \quad 0] \tag{10}$$

where $E(v_{k,l}v_{k,l}^T) = R_{k-l}$. Then, the same filtering equation of the Kalman filter can be applied. The update equation is derived as

$$\hat{x}_k^+ = \hat{x}_k^- + P_{k-l,k}^- H_{k-l}^T (H_{k-l}P_{k-l,k-l}^- H_{k-l}^T + R_{k-l})^{-1} \times (y_{k,l} - H_{k-l}\hat{x}_{k-l}^-) \tag{11}$$

where $P_{i,j} = E[(x_i - \hat{x}_i)(x_j - \hat{x}_j)^T]$, which is the covariance between the two state estimates at time i and j. This filtering method gives the optimal estimate, regardless of whether x_{k-l} has been updated before the time-delayed measurement or not.

C. Augmented Fixed-Lag Smoother

Unlike the ISM, the measurement time of an OOSM is assumed not to be synchronized with the previous filter updates, so the update method in Sec. III.B cannot be used directly. However, if it is possible to generate the state and covariance of the node at the OOSM time, we can apply Eq. (11) to update the state and covariance. The following steps summarize how the augmented fixed-lag smoother may be updated with any OOSM with a newly generated OOSM node. The OOSM node generation of the AFLS is derived in the following (Sec. III.D):

1) Run the general FLS Kalman filter.

2) If a time-delayed measurement is delivered, then proceed with one of the following:

a) If the measurement time is one of the FLS nodes, update the FLS using Eq. (11).

b) If the measurement time is not in the FLS nodes, i) generate a node using Eqs. (27–29); ii) plug the generated state estimate and the covariance into the FLS state vector and covariance matrix (Fig. 2); and iii) update the FLS using Eq. (11).

Figure 2 describes the "plug-in" process of the generated node into the FLS nodes where the subscript r means the OOSM time between a and b. Since the number of the state of estimates increases, the dimension of A_{k-1} in Eq. (8) should be increased with the dimension of the new covariance.

Because of the AFLS's structure, the filtering process remains exactly the same, whether there are single or multiple OOSMs and regardless of whether the measurements arrive in sequential order. For implementation with real systems, one must consider the number of nodes that should be kept. The basic rule is to keep all nodes up to the expected maximum delay time, and not to keep the measurement histories. Keeping up-to-date estimates for all nodes is beneficial in reducing the response time of the estimator. This will be discussed in Sec. III.E in detail.

D. OOSM Node Generation

The OOSM node generation of the AFLS expands upon the FLS. The basic idea is the following: the time of the OOSM occurs between two adjacent nodes where the filter update is performed so that the state of the OOSM can be regarded as an additional node between the two adjacent nodes. This new OOSM node is created without a corresponding filter update. This process may be shown by an inductive proof, with the base case being simply the FLS algorithm, since the new method expands upon the initialization provided by the FLS.

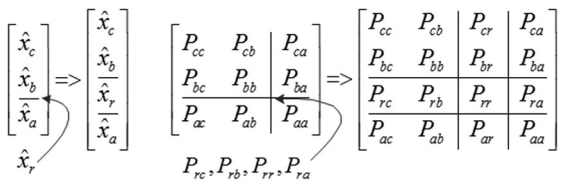

Fig. 2 Plug in the generated node.

The following represents the inductive step for a particular initial measurement step s with subsequent measurement steps $s+2$ and $s+3$, and a step $s+1$ for which the state is to be estimated with time delayed information. For this analysis, the letters a, b, c, and d represent s, $(s+1)$, $(s+2)$, and $(s+3)$, respectively. Also, $\hat{x}_{i,k}^{\pm}$ represents the conditioned state estimate of node i as $\hat{x}_{i,k}^- = E[x_i|y_1, \ldots, y_{k-1}]$ and $\hat{x}_{i,k}^+ = E[x_i|y_1, \ldots, y_k]$, where y_k is a measurement at t_k, and $P_{ij,k}^{\pm}$ represents the conditioned covariance between x_i and x_j as $P_{ij,k}^- = E[(x_i - \hat{x}_i)(x_j - \hat{x}_j)^T|y_1, \ldots, y_{k-1}]$ and $P_{ij,k}^+ = E[(x_i - \hat{x}_i)(x_j - \hat{x}_j)^T|y_1, \ldots, y_k]$, which means $\hat{x}_{i,k-1}^- = \hat{x}_{i,k}^+$ and $P_{ij,k-1}^- = P_{ij,k}^+$, essentially. The inductive step follows the initial base case of the initial Kalman smoother step, which has already been shown to generate state estimates optimally [12]. We assume no measurement at step b, which means $\hat{x}_{b,k}^+$ is a nonupdated node, and verify that the nonupdated node $(\hat{x}_{b,k}^+, P_{bb,k}^+, P_{bk,k}^+)$ can be generated by a linear combination of the two adjacent nodes. In the following equations, F_k, Q_k, R_k, and w_k follow the same definition as in Eq. (1).

For time step $a \to b$,

$$\begin{bmatrix} \hat{x}_{b,b}^- \\ \hat{x}_{a,b}^- \end{bmatrix} = \begin{bmatrix} F_a & 0 \\ I & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_{a,a}^+ \\ \hat{x}_{a-1,a}^+ \end{bmatrix} = \begin{bmatrix} F_a\hat{x}_{a,a}^+ \\ \hat{x}_{a,a}^+ \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} P_{bb,b}^- & P_{ba,b}^- \\ P_{ab,b}^- & P_{aa,b}^- \end{bmatrix} = \begin{bmatrix} F_aP_{aa,a}^+F_a^T + Q_a & F_aP_{aa,a}^+ \\ P_{aa,a}^+F_a^T & P_{aa,a}^+ \end{bmatrix} \tag{13}$$

Since there is no update at b, we can move on to the next time step.

For time step $b \to c$, similar to Eqs. (12) and (13), we can derive the posteriori estimates at c using Eq. (4) as

$$\hat{x}_{c,c}^+ = F_bF_a\hat{x}_{a,a}^+ + F_bF_aP_{aa,a}^+F_a^TF_b^Tm_c + F_bQ_aF_b^Tm_c + Q_bm_c \tag{14}$$

$$\hat{x}_{b,c}^+ = F_a\hat{x}_{a,a}^+ + F_aP_{aa,a}^+F_a^TF_b^Tm_c + Q_aF_b^Tm_c \tag{15}$$

$$\hat{x}_{a,c}^+ = \hat{x}_{a,a}^+ + P_{aa,a}^+F_a^TF_b^Tm_c \tag{16}$$

where

$$m_c \triangleq H_c^T(H_cP_{cc,c}^-H_c^T + R_c)^{-1}(y_c - H_c\hat{x}_{c,c}^-) \tag{17}$$

Combining Eqs. (14) and (16),

$$m_c = Q_{ca}^{-1}(\hat{x}_{c,c}^+ - F_bF_a\hat{x}_{a,c}) \tag{18}$$

where $Q_{ca} \triangleq F_bQ_aF_b^T + Q_b$ from $x_c = F_bF_ax_a + w_{ca}$, $w_{ca} \sim N(0, Q_{ca})$. From Eqs. (15), (16), and (18),

$$\hat{x}_{b,c}^+ = A\hat{x}_{a,c}^+ + B\hat{x}_{c,c}^+ \tag{19}$$

where

$$A \triangleq (I - BF_b)F_a, \qquad B \triangleq Q_aF_b^TQ_{ca}^{-1} \tag{20}$$

The covariance matrix can be calculated as follows (see Sec. A):

$$P_{bb,c}^+ = AP_{aa,c}^+A^T + BP_{cc,c}^+B^T + AP_{ac,c}^+B^T + BP_{ca,c}^+A^T + (I - BF_b)Q_a \tag{21}$$

$$P^+_{bj,c} = AP^+_{aj,c} + BP^+_{cj,c} \tag{22}$$

for all j, $j \neq b$.

Time Step $c \to d$: Similar to Step $b \to c$, we can derive the following:

$$\begin{aligned}
\hat{x}^+_{c,d} &= \hat{x}^+_{c,c} + P^+_{cc,c} n_d \\
\hat{x}^+_{b,d} &= \hat{x}^+_{b,c} + F_a P^+_{ac,c} n_d + Q_a F_b^T Q_{ca}^{-1} (P^+_{cc,c} - F_b F_a P^+_{ac,c}) n_d \\
\hat{x}^+_{a,d} &= \hat{x}^+_{a,c} + P^+_{ac,c} n_d
\end{aligned} \tag{23}$$

where

$$n_d \triangleq F_d^T H_d^T (H_d P^-_{dd,d} H^T + R_d)^{-1} (y_d - H_d \hat{x}^-_{d,d}) \tag{24}$$

Try the same form of (19) with (23) as follows:

$$\begin{aligned}
&F_a \hat{x}^+_{a,d} + Q_a F_b^T Q_{ca}^{-1} (\hat{x}^+_{c,d} - F_b F_a \hat{x}^+_{a,d}) \\
&\quad = F_a \hat{x}^+_{a,c} + Q_a F_b^T Q_{ca}^{-1} (\hat{x}^+_{c,c} - F_b F_a \hat{x}^+_{a,c}) + F_a P_{ac,c} n_d \\
&\qquad + Q_a F_b^T Q_{ca}^{-1} (P^+_{cc,c} - F_b F_a P^+_{ac,c}) n_d \\
&\quad = \hat{x}^+_{b,c} + F_a P_{ac,c} n_d + Q_a F_b^T Q_{ca}^{-1} (P^+_{cc,c} - F_b F_a P^+_{ac,c}) n_d \\
&\quad = \hat{x}^+_{b,d}
\end{aligned} \tag{25}$$

Similar to (21–22), the covariance is given as the following (see Appendix A3):

$$\begin{aligned}
P^+_{bb,d} &= AP^+_{aa,d} A^T + BP^+_{cc,d} B^T + AP^+_{ac,d} B^T + BP^+_{ca,d} A^T + (I - BF_b) Q_a \\
P^+_{bj,d} &= AP^+_{aj,d} + BP^+_{cj,d}
\end{aligned} \tag{26}$$

for all j, $j \neq b$, where A and B are given by Eq. (20).

Since $d = c + 1$, it is shown by mathematical induction that we can generate the nonupdated node point b with the linear combination of two adjacent nodes, a and c, for the state estimation and covariance as follows:

$$\hat{x}^+_{b,k} = A\hat{x}^+_{a,k} + B\hat{x}^+_{c,k} \tag{27}$$

$$\begin{aligned}
P^+_{bb,k} &= AP^+_{aa,k} A^T + BP^+_{cc,k} B^T + AP^+_{ac,k} B^T \\
&\quad + BP^+_{ca,k} A^T + (I - BF_b) Q_a
\end{aligned} \tag{28}$$

$$P^+_{bj,k} = AP^+_{aj,k} + BP^+_{cj,k} \tag{29}$$

for all j, $j \neq b$, and for $k = c, c+1, c+2, \ldots$, where A and B are given by Eq. (20). Note that there are no assumptions made about b, except that b occurs after the first filter update that generates updated node a. Therefore, this induction shows that the proposed method is the general solution.

E. Comparison to Existing Algorithms

The existing methods for optimal updating with OOSM were well summarized by Zhang and Bar-Shalom [9]. Among the several algorithms introduced in [9], the complete in-sequence information-based fixed-interval smoothing (CISI-fixed-point smoother (FPS)) was claimed as the most efficient one among the algorithms that guaranteed the optimal estimation result and was verified quantitatively by numerical simulations. The concept of CISI-FPS was that, if an OOSM arrived, the CISI-FPS went back to the last step before the OOSM and updated the state estimation and covariance all the way to the current step. Another comparable algorithm was proposed as ALG-I in [8]. In this reference, ALG-I was compared to ALG-S, which was proposed in [10] and capable of handling the ISM only. Zhang et al. [8] noted that the FLS approach was conceptually clearer, simpler, and more elegant, but criticized the required memory and the computation because the FLS algorithm needed to increase the sampling rate to generate and save the possible nodes for expected OOSMs. However, with Eqs. (27) and (28), the AFLS was able to generate the node for OOSMs when it arrived, so the sampling rate could remain the same as the others. It was true that the FLS or the AFLS required more throughput to update all the nodes in the FLS compared to the sequential-update-based algorithms, but the AFLS could be more efficient in terms of response time because it did not need to renew all the stored nodes to update the current state with it. Once the OOSM node was generated, the AFLS could give the real-time estimation at the current time first and update the other nodes later when the other time-critical processes were complete. Since most of real-time estimation was a part of a control loop, this could be effective to minimize the response delay in the overall control system.

The required memory is another issue for applications. The AFLS needs more memory than ALG-I or CISI-FPS because the AFLS requires $sp + s^2(p \times p)$, whereas ALG-I needs $(s-1)p + 2s(p \times p)$, where p is the dimension of the state x_k and s is the number of the maximum delay [8]. This is a typical tradeoff problem between processing speed and memory usage, so it is hard to conclude which one is globally the most efficient.

Though Bar-Shalom did not explicitly derive his approach with FLS in [7], it could be directly used with the FLS in the same manner as the AFLS. Combined with the FLS, Bar-Shalom's algorithm (BSA) guaranteed optimal state estimation, as did the AFLS.

The difference between BSA and the AFLS is how to generate the OOSM node. BSA explicitly estimates the process noise and applies it when calculating the OOSM node, whereas the AFLS interpolates the two adjacent nodes. Despite the conceptual difference, either BSA with the FLS or the AFLS will require similar throughput for updates. In terms of the required memory, the AFLS is slightly more efficient than BSA because BSA needs to store the previous measurements to calculate the process noise, whereas the AFLS does not.

F. Extension to Nonlinear Systems

Since most dynamic systems in the aerospace field are nonlinear, the AFLS needs to be extended to nonlinear systems for actual applications. Consider a discrete-time nonlinear system for which the system and measurement equations are given as follows:

$$\begin{aligned}
x_{k+1} &= f_k(x_k, w_k) \\
y_k &= h_k(x_k, v_k) \\
w_k &\sim N(0, Q_k) \\
v_k &\sim N(0, R_k)
\end{aligned} \tag{30}$$

By taking the same approach as the extended Kalman filter, the covariance of a node can be generated using Eqs. (28) and (29) for this system with slight modification:

$$\begin{aligned}
P^+_{bb,k} &= AP^+_{aa,k} A^T + BP^+_{cc,k} B^T + AP^+_{ac,k} B^T \\
&\quad + BP^+_{ca,k} A^T + (I - BF_b) Q_a
\end{aligned} \tag{31}$$

$$P^+_{bj,k} = AP^+_{aj,k} + BP^+_{cj,k} \tag{32}$$

where

$$A = (I - BF_b) F_a, \qquad B = Q_{la} F_b^T Q_{lca}^{-1} \tag{33}$$

$$F_k = \left.\frac{\partial f_k}{\partial x}\right|_{\hat{x}^+_k}, \qquad L_k = \left.\frac{\partial f_k}{\partial w}\right|_{\hat{x}^+_k} \tag{34}$$

$$\begin{aligned}
Q_{la} &= L_a Q_a L_a^T \\
Q_{lca} &= F_b L_a Q_a L_a^T F_b^T + L_b Q_b L_b^T
\end{aligned} \tag{35}$$

For generating the state estimate, Eq. (27) is also useful for solving a nonlinear system. Equation (27) can be rewritten as

$$\begin{aligned}\hat{x}^+_{b,k} &= F_a\hat{x}^+_{a,k} + Q_{la}F_b^TQ^{-1}_{lca}F_b(F_b^{-1}\hat{x}^+_{c,k} - F_a\hat{x}^+_{a,k})\\ &= \hat{x}_b|_{\hat{x}^+_{a,k}} + Q_{la}F_b^TQ^{-1}_{lca}F_b(\hat{x}_b|_{\hat{x}^+_{c,k}} - \hat{x}_b|_{\hat{x}^+_{a,k}})\end{aligned} \tag{36}$$

In a nonlinear system given as Eq. (30), $\hat{x}_b|_{\hat{x}^+_{c,k}}$ and $\hat{x}_b|_{\hat{x}^+_{a,k}}$ can be calculated:

$$\begin{aligned}\hat{x}_b|_{\hat{x}^+_{a,k}} &= f_a(\hat{x}^+_{a,k}, 0)\\ \hat{x}_b|_{\hat{x}^+_{c,k}} &= f_b^{-1}(\hat{x}^+_{c,k}, 0)\end{aligned} \tag{37}$$

where $f_k^{-1}(\cdots)$ denotes the backward propagation from $k+1$ to k:

$$x_{k+1} = f_k(x_k, w_k) \Leftrightarrow x_k = f_k^{-1}(x_{k+1}, w_k) \tag{38}$$

To summarize, the state estimate and covariance of a node can be generated using the two adjacent nodes:

$$\begin{aligned}&\hat{x}^+_{b,k} = f_a(\hat{x}^+_{a,k},0) + BF_b(f_b^{-1}(\hat{x}^+_{c,k},0) - f_a(\hat{x}^+_{a,k},0))\\ &P^+_{bb,k} = AP^+_{aa,k}A^T + BP^+_{cc,k}B^T + AP^+_{ac,k}B^T + BP^+_{ca,k}A^T + (I - BF_b)Q_a\\ &P^+_{bj,k} = AP^+_{aj,k} + BP^+_{cj,k} \text{ for } j \neq b\end{aligned} \tag{39}$$

where A and B are given by Eqs. (33–35). We will call this the extended AFLS for the rest of this study.

IV. Numerical Example for the EAFLS Application

Real-time satellite tracking for optical communication is a good example of the utility of the EAFLS. Optical communication has been highlighted as a key next-generation technology with advantages in power efficiency compared with radio frequency (RF) communication systems [13]. The higher gain from having a narrow optical communication beam width is a key enabling factor, which correspondingly demands a lower pointing error. For modern low-Earth-orbit satellites, there are three measurements that are generally available: range between the satellite and ground station (GS), range rate from the Doppler shift, and Global Navigation Satellite System (GNSS) data from an onboard GNSS receiver. Since the first two are measured at the ground station, we assume for this example that the time delay for the ground-based measurements is small compared to the GNSS data that measured on the satellite and must be downlinked through an RF link. So, the GNSS data have some time delay, and the delay can be arbitrary, depending on the RF link status. Our example uses one of the simplest formulations for orbit estimation with the intention of showing the application of the EAFLS to an aerospace system.

From two-body orbital dynamics, the differential equation of position $\boldsymbol{r} = [x, y, z]^T$ and velocity $\boldsymbol{v} = [u, v, w]^T$ are given as

$$\dot{\boldsymbol{r}} = \boldsymbol{v}, \qquad \dot{\boldsymbol{v}} = -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{\eta} \tag{40}$$

where $r = |\boldsymbol{r}|$ and $\boldsymbol{\eta}$ are random accelerations (process noise) assumed to be white noise as $E[\boldsymbol{\eta}\boldsymbol{\eta}^T] = \text{diag}(q_1, q_2, q_3)$. The measurement models for range ρ, range rate $\dot{\rho}$, and satellite position from the GNSS output $\boldsymbol{r}_G$ are given as the following:

$$\begin{aligned}\rho &= \sqrt{(x - x_{gs})^2 + (y - y_{gs})^2 + (z - z_{gs})^2} + n_\rho\\ \dot{\rho} &= \frac{1}{\rho}\{(x - x_{gs})(u - u_{gs}) + (y - y_{gs})(v - v_{gs})\\ &\quad + (z - z_{gs})(w - w_{gs})\} + n_{\dot{\rho}}\\ \boldsymbol{r}_G &= \boldsymbol{r} + \boldsymbol{n}_G\end{aligned} \tag{41}$$

where (x_{gs}, y_{gs}, z_{gs}) and (u_{gs}, v_{gs}, w_{gs}) are the position and velocity of the GS in an inertial frame such as J2000, which is assumed to be known; and n_ρ, $n_{\dot{\rho}}$, and $\boldsymbol{n}_G$ are measurement noise for each measurement. Also, n_ρ and $n_{\dot{\rho}}$ are usually assumed to be exponentially correlated in time, and $\boldsymbol{n}_G$ is modeled as a more complicated noise process [14]. However, here, we assume they are white noise processes for which the covariance is given as $E[n_\rho^2] = r_\rho$, $E[n_{\dot{\rho}}^2] = r_{\dot{\rho}}$, and $E[\boldsymbol{n}_G\boldsymbol{n}_G^T] = \text{diag}(r_x, r_y, r_z)$, respectively, to simplify the problem.

For this system, we can define the state vector to be estimated as $\boldsymbol{x} = [x, y, z, u, v, w]^T$. Then, the partial derivative of the differential equation of the state is given as [15]

$$\frac{\partial\dot{\boldsymbol{x}}}{\partial\boldsymbol{x}} = F = \begin{bmatrix} 0 & I \\ F' & 0 \end{bmatrix} \tag{42}$$

where

$$F' = -\frac{\mu}{r^3}I + 3\frac{\mu}{r^5}\begin{bmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{bmatrix} \tag{43}$$

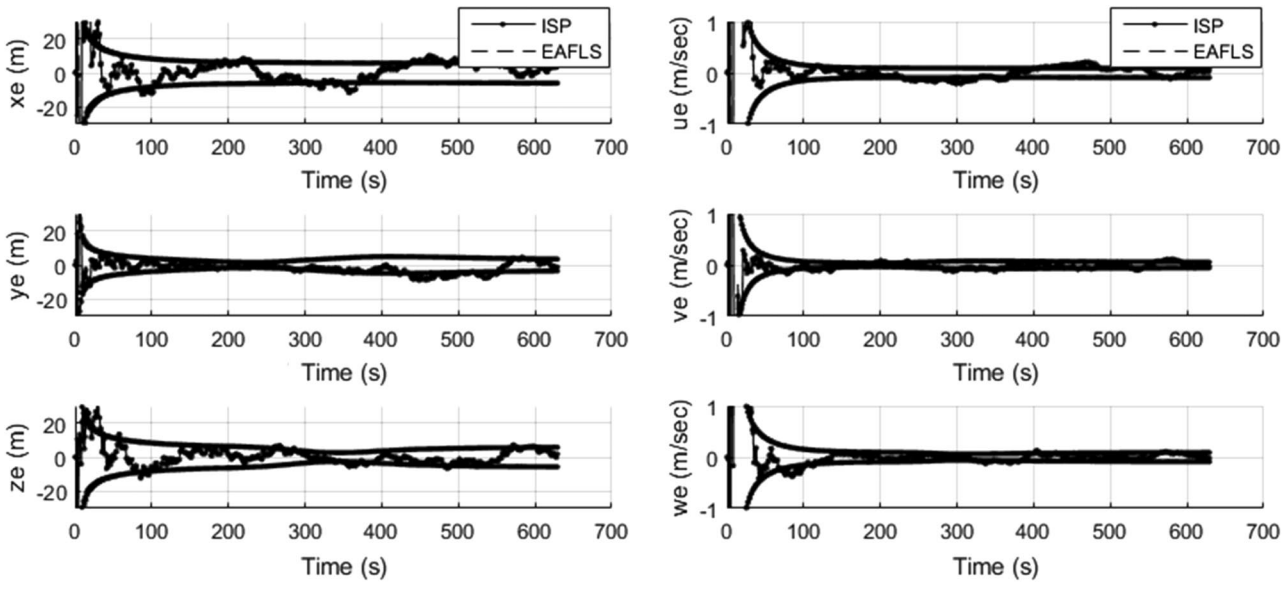

Fig. 3 Satellite-tracking error and 1σ boundary, which shows the estimates of two methods are very similar.

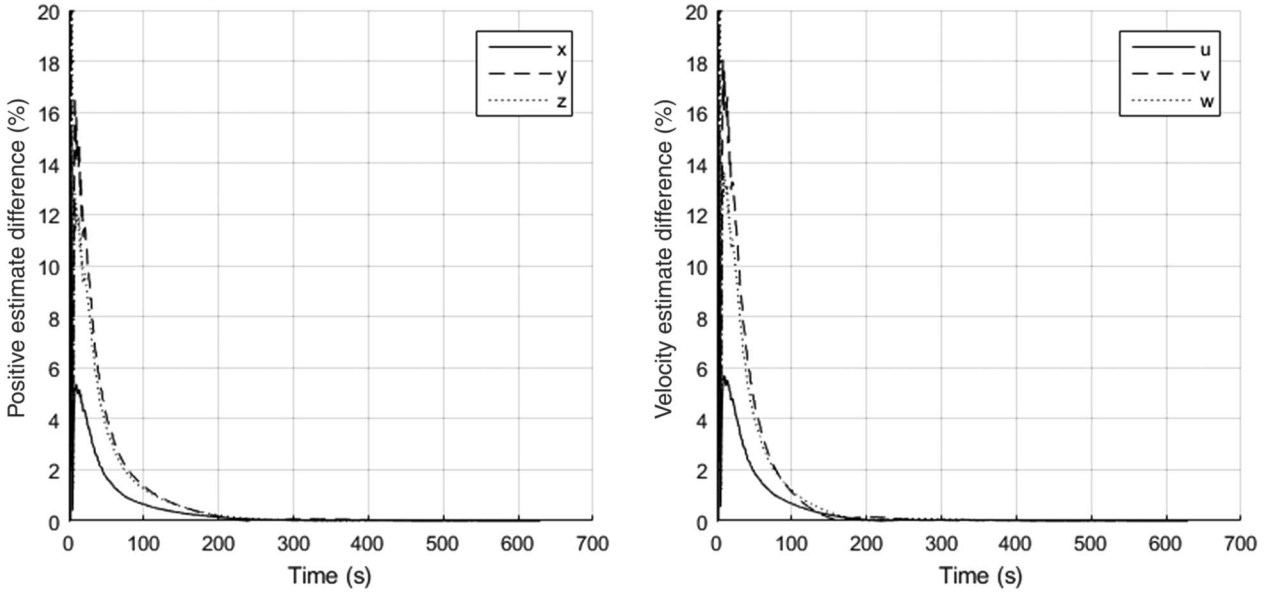

Fig. 4 Difference between the ISP and the EAFLS estimates ratio to the standard deviation.

Given the state vector, it is straightforward to derive the partial derivatives of the measurements. Further details may be found in [14,15].

In this example, the International Space Station orbit is used (400 km altitude and 51.65 deg inclination), and the ground station location is assumed to be at $(\text{Lat}, \text{Lon}) = (42.3584488, -71.0912204)^\circ$, which is the geographical location of the Massachusetts Institute of Technology. The simulation period is set as a contact opportunity of 630 s at 8 November 2015 10:22:14.876 hrs. For orbit propagation, two-body dynamics as given in Eq. (40) are numerically integrated by the leapfrog integration method. For the noise parameters, $q_1 = q_2 = q_3 = (0.01\ \text{m}/\sqrt{\text{sec}^3})^2$, $r_\rho = (1\ \text{m})^2$, $r_{\dot{\rho}} = (0.1\ \text{m}/\text{sec})^2$, and $r_x = r_y = r_z = (30\ \text{m})^2$ are used. The process noise is given in the continuous domain, so it is converted to a discrete random variable by multiplying with the square root of the step time and added into the numerical integrator. As mentioned before, the range and range rate measurement are assumed to be measured and available instantly, but the GNSS measurements are assumed to be out of sequence, each with a time delay that is assumed to be randomly distributed between 2 and 3 s. The filter is updated at 1 Hz. The initial guess of the covariance is set to be

$$P_0 = \text{diag}((10\ \text{km})^2, (10\ \text{km})^2, (10\ \text{km})^2, (100\ \text{m}/\text{sec})^2, (100\ \text{m}/\text{sec})^2, (100\ \text{m}/\text{sec})^2)$$

Filtering results from the EAFLS and the ISP using the extended Kalman filter (EKF), described in Sec. III.A, are compared. Figure 3 shows the estimation error and the covariance of each method. The results are similar, but they show certain differences. Figure 4 shows the state estimate difference percentage between those two method to the estimated 1σ. Unlike a linear system case, the EKF and the EAFLS calculate the partial derivative matrix [Eq. (34)] for each step, which means the matrix will be different for each unique state estimate. This process creates the difference between the two results shown in Fig. 4. This kind of difference is inevitable for the extended approaches to nonlinear systems because the partial derivatives are just an approximation of nonlinear system. Regardless, the difference percentage converges to less than 10% within 30 s, which is generally sufficient for practical applications.

V. Conclusions

In this study, a new approach is presented to update the Kalman filter with out-of-sequence measurements for real-time estimation in general form. By generating the estimate at the out-of-sequence measurement time through the creation of a new node and using the information from the two adjacent nodes, the Kalman filter can be updated with the existing fixed-lag smoother formulation. The mathematical formulation of the proposed method, the augmented fixed-lag smoother, is derived from the fixed-lag smoother with an induction proof. The augmented fixed-lag smoother (AFLS) is extended to a nonlinear system as the extended AFLS. To demonstrate the extended AFLS (EAFLS) numerically, a satellite-tracking problem is described with simplified noise models. The simulation results show that the difference between the estimates of the EAFLS and the conventional extended Kalman filter is small. This is because the extended approaches approximate the nonlinear system as a linear system for a given state. The results also show that the small difference between the EAFLS and the extended Kalman filter converges less than 10% within 30 s and goes to zero as the filter converges to a steady state.

Appendix A: Derivations

A1. $P^+_{bb,c}$ Derivation

From Eqs. (1) and (19), the covariance matrix $P^+_{bb,c}$ can be expanded as

$$\begin{aligned} P^+_{bb,c} &= E[(x_b - \hat{x}^+_{b,c})(x_b - \hat{x}^+_{b,c})^T] \\ &= AP^+_{aa,c}A^T + BP^+_{cc,c}B^T + AP^+_{ac,c}B^T + BP^+_{ca,c}A^T \\ &\quad + (I - BF_b)Q_a(I - BF_b)^T + BQ_bB^T \\ &\quad + E[A(x_a - \hat{x}^+_{a,c})((I - BF_b)w_a - Bw_b)^T \\ &\quad + B(x_c - \hat{x}^+_{c,c})((I - BF_b)w_a - Bw_b)^T] \\ &\quad + E[A(x_a - \hat{x}^+_{a,c})((I - BF_b)w_a - Bw_b)^T \\ &\quad + B(x_c - \hat{x}^+_{c,c})((I - BF_b)w_a - Bw_b)^T]^T \end{aligned} \tag{A1}$$

where A and B are are given by Eq. (20). It is straightforward to show that the fifth and sixth terms of Eq. (A1) can be expanded as

$$(I - BF_b)Q_a(I - BF_b)^T + BQ_bB^T = (I - BF_b)Q_a \tag{A2}$$

Considering the third line of Eq. (A1), it is clear that

$$\begin{aligned} &E[(x_a - \hat{x}^+_{a,c})w_a^T] \neq 0, \quad E[(x_a - \hat{x}^+_{a,c})w_b^T] \neq 0 \\ &E[(x_c - \hat{x}^+_{c,c})w_a^T] \neq 0, \quad E[(x_c - \hat{x}^+_{c,c})w_b^T] \neq 0 \end{aligned} \tag{A3}$$

Before expanding the third line, let us consider the following. From Eq. (17),

$$m_c = H_c^T(H_cP_{cc,c}^-H_c^T + R_c)^{-1}(y_c - H_c\hat{x}_{c,c}^-) = H_c^T(\cdots)^{-1}(H_cF_bF_a(x_a - \hat{x}_{a,a}^+) + H_cF_bw_a + H_cw_b + v_c) \tag{A4}$$

Since $E[(x_a - \hat{x}_{a,a}^+)w_a] = 0$ and $E[(x_a - \hat{x}_{a,a}^+)w_b] = 0$,

$$E[m_cw_a^T] = M_cF_bQ_a, E[m_cw_b^T] = M_cQ_b \tag{A5}$$

where $M_c \triangleq H_c^T(H_cP_{cc,c}^-H_c^T + R_c)^{-1}H_c$. Using Eq. (A5), it is straightforward to prove that

$$E[m_c((I - BF_b)w_a - Bw_b)^T] = 0 \tag{A6}$$

More specifically,

$$E[(y_c - H_c\hat{x}_{c,c}^-)((I - BF_b)w_a - Bw_b)^T] = 0 \tag{A7}$$

Equation (A7) is a powerful condition that means that the innovation $(y_c - H_c\hat{x}_{c,c}^-)$ is orthogonal to the residual process noise $((I - BF_b)w_a - Bw_b)$. From Eqs. (16) and (14) with Eq. (A6), we can obtain

$$E[A(x_a - \hat{x}_{a,c}^+)((I - BF_b)w_a - Bw_b)^T] = 0 \tag{A8}$$

$$E[B(x_c - \hat{x}_{c,c}^+)((I - BF_b)w_a - Bw_b)^T] = 0 \tag{A9}$$

Finally, from Eqs. (A1), (A2), (A8), and (A9),

$$P_{bb,c}^+ = AP_{aa,c}^+A^T + BP_{cc,c}^+B^T + AP_{ac,c}^+B^T + BP_{ca,c}^+A^T + (I - BF_b)Q_a \tag{A10}$$

A2. $P_{bj,c}^+$ Derivation

Similar to Eq. (A1),

$$P_{bj,c}^+ = E[(x_b - \hat{x}_{b,c}^+)(x_j - \hat{x}_{j,c}^+)^T] = AP_{aj,c}^+ + BP_{cj,c}^+ + E[(x_j - \hat{x}_{j,c}^+)((I - BF_b)w_a - Bw_b)^T]^T \tag{A11}$$

The third term of Eq. (A11) will become zero for all j, $j \neq b$. For the case of $j > b$, which means $j = c$, it is shown in Eq. (A9). When $j < b$, similar to Eqs. (14–17), it is straightforward to show that

$$\hat{x}_{j,c}^+ = \hat{x}_{j,a}^+ + P_{jc,c}^-m_c \tag{A12}$$

From Eqs. (A6) and (A12),

$$E[(x_j - \hat{x}_{j,c}^+)((I - BF_b)w_a - Bw_b)^T] = 0 \tag{A13}$$

Therefore, from Eqs. (A11) and (A13),

$$P_{bj,c}^+ = AP_{aj,c}^+ + BP_{cj,c}^+ \tag{A14}$$

A3. $P_{bb,d}^+$ and $P_{bj,d}^+$ Derivation

Let us consider m_d similar to Eq. (A4) as

$$m_d = H_d^T(H_dP_{dd,d}^-H_d^T + R_d)^{-1}(y_d - H_d\hat{x}_{d,d}^-) = H_d^T(\cdots)^{-1}(H_dF_c(x_c - \hat{x}_{c,c}^+) + H_dw_c + v_d) \tag{A15}$$

By Eq. (A9),

$$E[m_d((I - BF_b)w_a - Bw_b)^T] = 0 \tag{A16}$$

Similar to Eq. (A1),

$$\begin{aligned} P_{bb,d}^+ = {} & AP_{aa,d}^+A^T + BP_{cc,d}^+B^T + AP_{ac,d}^+B^T + BP_{ca,d}^+A^T \\ & + (I - BF_b)Q_a(I - BF_b)^T + BQ_bB^T \\ & + E[A(x_a - \hat{x}_{a,d}^+)((I - BF_b)w_a - Bw_b)^T \\ & + B(x_c - \hat{x}_{c,d}^+)((I - BF_b)w_a - Bw_b)^T] \\ & + E[A(x_a - \hat{x}_{a,d}^+)((I - BF_b)w_a - Bw_b)^T \\ & + B(x_c - \hat{x}_{c,d}^+)((I - BF_b)w_a - Bw_b)^T]^T \end{aligned} \tag{A17}$$

Since $\hat{x}_{a,d}^+ = \hat{x}_{a,c}^+ + (\cdots)m_d$ and $\hat{x}_{c,d}^+ = \hat{x}_{c,c}^+ + (\cdots)m_d$,

$$E[(x_a - \hat{x}_{a,d}^+)((I - BF_b)w_a - Bw_b)^T] = 0$$
$$E[(x_c - \hat{x}_{c,d}^+)((I - BF_b)w_a - Bw_b)^T] = 0 \tag{A18}$$

By Eqs. (A2), (A17), and (A18),

$$P_{bb,d}^+ = AP_{aa,d}^+A^T + BP_{cc,d}^+B^T + AP_{ac,d}^+B^T + BP_{ca,d}^+A^T + (I - BF_b)Q_a \tag{A19}$$

Also, similar to Eq. (A11),

$$P_{bj,d}^+ = AP_{aj,d}^+ + BP_{cj,d}^+ + E[(x_j - \hat{x}_{j,d}^+)((I - BF_b)w_a - Bw_b)^T]^T \tag{A20}$$

And, it is straightforward to show that $E[(x_j - \hat{x}_{j,d}^+)((I - BF_b)w_a - Bw_b)^T] = 0$ with the same approach from Eq. (A9). Finally,

$$P_{bj,d}^+ = AP_{aj,d}^+ + BP_{cj,d}^+ \tag{A21}$$

References

[1] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problem," *Journal of Basic Engineering*, Vo. 82, No. 1, March 1960, pp. 35–45. doi:10.1115/1.3662552

[2] Hutputtanasin, A., and Toorin, A., "CubeSat Design Specification (CDS) Rev 9," *Cal Poly*, California Polytechnic State Univ., San Luis Obispo, CA, May 2015, http://org.ntnu.no/studsat/docs/proposal_1/A8%20-%20Cubesat%20Design%20Specification.pdf [retrieved 31 May 2015].

[3] Marris, E., "Drones in Science: Fly, and Bring Me Data," *Nature*, Vol. 498, June 2013, pp. 156–158.

[4] "PIC16F627A/628A/648A Data Sheet: Flash-Based, 8-Bit CMOS Microcontrollers with Nanowatt Technology," Microchip Technologies, Microchip Document DS40044G, Chandler, AZ, 2009.

[5] Blackman, S. S., and Popoli, R., *Design and Analysis of Modern Tracking Systems*, Artech House, Dedham, MA, 1999, Chap. 10.

[6] Hilton, R. D., Martin, D. A., and Blair, W. D., "Tracking with Time-Delayed Data in Multisensor Systems," Rept. NSWCDD/TR-93/351, Naval Surface Warfare Center, Dahlgren, VA, Aug. 1993.

[7] Bar-Shalom, Y., "Update with Out-of-Sequence Measurements in Tracking: Exact Solution," *IEEE Transactions on Aerospace Electronic Systems*, Vol. 38, No. 3, 2002, pp. 769–777. doi:10.1109/TAES.2002.1039398

[8] Zhang, K., Li, X. R., and Zhu, Y., "Optimal Update With Out-of-Sequence Measurements," *IEEE Transactions on Signal Processing*, Vol. 53, No. 6, 2005, pp. 1992–2004. doi:10.1109/TSP.2005.847830

[9] Zhang, S., and Bar-Shalom, Y., "Optimal Update with Multiple Out-of-Sequence Measurements with Arbitrary Arriving Order," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, No. 4, 2012, pp. 3116–3132. doi:10.1109/TAES.2012.6324681

[10] Challa, S., Evans, R., and Wang, X., "A Bayesian Solution to OOSM Problem and Its Approximations," *Information Fusion*, Vol. 4, No. 3, 2003, pp. 185–199. doi:10.1016/S1566-2535(03)00037-X

[11] Simon, D., *Optimal State Estimation: Kalman, H [Infinity] and Nonlinear Approaches*, Wiley, Hoboken, NJ, 2006, Chaps. 9, 13.
[12] Anderson, B. D. O., and Moore, J. B., *Optimal Filtering*, Dover, New York, 2005, Chap. 9.
[13] Kingsbury, R. W., "Optical Communications for Small Satellites," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, Sept. 2015.
[14] Jordan, J. F., "Orbit Determination for Powered Flight Space Vehicles on Deep Space Missions," *Journal of Spacecraft and Rockets*, Vol. 6, No. 5, 1969, pp. 545–550.
doi:10.2514/3.29611
[15] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, 3rd ed., Space Technology Library, Microcosm Press, Hawthorne, CA, 2007, Chap. 10.