Aerospace Science and Technology 39 (2014) 465–471

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Multirate multisensor data fusion for linear systems using Kalman filters and a neural network

Sajjad Safari [a], Faridoon Shabani [a], Dan Simon [b,*]

[a] School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
[b] School of Electrical and Computer Engineering, Cleveland State University, Cleveland, USA

ARTICLE INFO

Article history:
Received 11 November 2013
Received in revised form 11 February 2014
Accepted 11 June 2014
Available online 1 July 2014

Keywords:
State estimation
Data fusion
Multirate
Multisensor
Neural network
Kalman filter

ABSTRACT

In this paper the data fusion problem for asynchronous, multirate, multisensor linear systems is studied. The linear system is observed by multiple sensor systems, each having a different sampling rate. Under the assumption that the state space model is known at the scale of the highest time resolution sensor system, and that there is a known mathematical relationship between the sampling rates, a comprehensive state space model that includes all sensor systems is presented. The state vector is estimated with a neural network that fuses the outputs of multiple Kalman filters, one filter for each sensor system. The state estimate is shown to perform better than other data fusion approaches due to the new neural network based sensor fusion approach.

1. Introduction

State estimation is the process of inferring the state of a system from indirect and uncertain observations [1]. Using multiple sensor systems instead of one single sensor system increases the performance of estimation due to the use of complementary information and increased reliability [24]. Data fusion is a process in which data from different sensor systems, observing the same system, are combined to obtain better estimation accuracy [8]. For example, in image processing, one scene may be captured by different cameras with different sampling rates [18]. In earlier data fusion work, the sensor systems observing the process had equal sampling rates, which led to fairly simple data fusion problem with limited applicability [13]. In reality, different sensor systems often use different sampling rates, and the sampling rates are often asynchronous [21].

Various methods have been presented to fuse data from multiple sensor systems. Among them, Carlson presents a method based on Kalman filtering to fuse data from sensor systems having the same sampling rate [3,4]. Kazerooni et al. developed a federated ensemble Kalman filter algorithm [12]. Other popular state estimators, such as particle filters [14] and H-infinity filters [10], have been used. Soft computing methods, such as fuzzy logic [17], genetic algorithms [15], and neural networks [6], have also been used. Wavelet methods have been developed to fuse data from different sensor systems with different sampling rates [5,23]. All of these works have some limitations on the sampling times of the sensor systems and on the relationship between the different sampling rates.

Yan et al.'s algorithm is applicable to systems in which the sensor system sample rates are asynchronous with any known integer sampling rate ratio [22]. In their work the limitation on the sampling rate ratio is relaxed relative to previous research; that is, the ratio between sampling rate ratios is assumed only to be a positive integer. The system model is known only at the finest sampling rate, and they extend the federated Kalman filter [3] to fuse data from the multiple sensor systems.

This paper uses Yan et al.'s method [22] to transform the multirate multisensor system into a single-rate multisensor model. The states of the new system are estimated with a standard Kalman filter, one filter for each sensor system. Then, instead of using a classical method such as a federated Kalman filter, a neural network fuses the estimated state vectors from each Kalman filter. The results are shown to be more accurate than the method of [22]. The results of this paper use the same sampling rate assumptions as in [22]; therefore, the improved results in this paper are due to the neural network based sensor fusion that is proposed here.

This paper is organized as follows. In Section 2, multirate system modeling is reviewed. In Section 3, data fusion and state estimation using the new combination of Kalman filters and a neural

* Corresponding author.
E-mail addresses: s.safari@gmail.com (S. Safari), shabani@shirazu.ac.ir (F. Shabani), d.j.simon@csuohio.edu (D. Simon).

http://dx.doi.org/10.1016/j.ast.2014.06.005
1270-9638/

network is presented. Section 4 presents simulation results, and Section 5 provides a conclusion.

2. Modeling a multirate system

A linear dynamic system with N sensor systems is described as follows [22]:

$$x(N,k+1) = A(N,k)x(N,k) + w(N,k) \tag{1}$$

$$z(i,k) = C(i,k)x(i,k) + v(i,k), \quad i = 1,2,\ldots,N \tag{2}$$

The state space model is valid at the highest sensor system sampling rate, which is denoted by N. Vector $x(N,k) \in R^{n\times 1}$ is the state variable at the k-th time step at time scale N, which is the same time scale as the highest time resolution sensor system. The system matrix $A(N,k) \in R^{n\times n}$. Vector $x(i,k) \in R^{n\times 1}$ is the state variable at the k-th time step at time scale i, and is generally different than $x(j,k)$ for $j \neq i$ because time scales i and j are different. Note that we do not have a system model for $x(i,k)$ for $i < N$. There are N sensor systems, with the i-th sensor system output at the k-th time step of the i-th time scale denoted by $z(i,k)$, and with $C(i,k) \in R^{q_i\times n}$. The system and measurement noises are independent, white, and zero-mean:

$$E\{w(N,k)w^T(N,l)\} = Q(N)\delta_{kl} \tag{3}$$

$$E\{v(i,k)v^T(j,l)\} = R(i)\delta_{ij}\delta_{kl} \tag{4}$$

Sensor system N has the highest sampling rate, and it is the only sensor system that uses uniform sampling. The other sensor systems have lower, and possibly non-uniform, sampling rates. The i-th sensor system sample rate is denoted S_i. The sampling rates of the sensor systems satisfy the following limitation:

$$S_i = S_{i+1}/n_i, \quad i \in [1, N-1] \tag{5}$$

where each n_i is a known positive integer. The n_i parameters are system-specific parameters that depend on the sensor system configuration. The determination of the n_i parameters are part of the system modeling problem, just as the determination of the model parameters in (1)–(4) are part of the system modeling problem.

This relationship given by (5) implies that the highest sensor system frequency is a fixed integer multiple of each of the other sensor system frequencies. There are many factors that affect sample rates in multisensor systems, and the assumption of (5) is restrictive. However, (5) is a reasonable model for many multisensor systems because data from multisensor systems are often fused in a central processor. The central processor often has a fixed rate at which it retrieves data from the multiple sensor systems. Therefore, (5) is often enforced by the architecture of the multisensor system. This is the case in many systems, including navigation systems [7], industrial systems [2], and transportation systems [11], and many others.

The initial state $x(N,0)$ is random with known mean x_0 and known covariance P_0, and is independent of the system and measurement noises.

An example of a multirate multisensor system is shown in Fig. 1. In this figure, there are three sensor systems ($N = 3$). The dynamic system is modeled at the rate S_3, which is the rate of the highest-rate sensor system, or the third sensor system. The second sensor system has sample rate $S_2 = S_3/2$. The first sensor system has sample rate $S_1 = S_2/3$.

The problem in this section is to find an approximate system model that applies to all sensor systems, under the assumption that the system presented in (1) is applicable only for the highest sampling rate N. In another words, our goal is to reformulate the multirate multisensor system as a single-rate multisensor system.

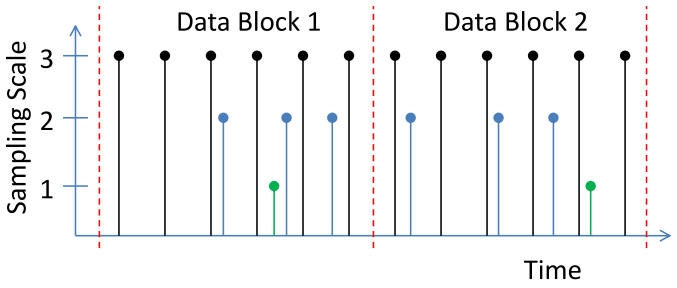

Fig. 1. Example of a multirate, multisensor system. The highest sample rate is sampling scale 3, which is uniform and includes 6 samples per data block. The two lower sample rates are asynchronous, but are constrained to be related to the highest sample rate by a known integer. This figure is adapted from [22].

Then, in the next section, we will deal with the multisensor data fusion problem by using one sampling rate for all sensors, which will be more tractable than the original problem. Based on [22] we approximate the state at time scale k as an average of the state at the highest time scale. That is,

$$x(i,k) \approx \frac{1}{\tilde{M}_i}\sum_{m=0}^{\tilde{M}_i-1} x(N, k\tilde{M}_i - m)$$

$$\tilde{M}_i = \prod_{j=1}^{N-1} n_j \tag{6}$$

for $i \in [1,\ N-1]$. An example will be given later in this section. This approximation gives the following state space model, which applies to all sensor system sampling rates:

$$X_N(k+1) = A_N(k)X_N(k) + W_N(k) \tag{7}$$

$$Z_i(k) = C_i(k)X_N(k) + V_i(k) \tag{8}$$

for $i \in [1, N-1]$, where

$$X_N(k) = \begin{bmatrix} x(N,(k-1)M+1) \\ x(N,(k-1)M+2) \\ \vdots \\ x(N,kM) \end{bmatrix} \tag{9}$$

$$A_N(k) = \begin{bmatrix} 0 & 0 & \cdots & A(N,kM) \\ 0 & 0 & \cdots & A(N,kM+1)A(N,kM) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \prod_{l=M-1}^{0} A(N,kM+l) \end{bmatrix} \tag{10}$$

$$Z_i(k) = \begin{bmatrix} z(i,(k-1)M_i+1) \\ z(i,(k-1)M_i+2) \\ \vdots \\ z(i,kM_i) \end{bmatrix} \tag{11}$$

$$C_i(k) = \frac{1}{\tilde{M}_i}\operatorname{diag}\{C(i,(k-1)M_i+1)I_{\tilde{M}_i} \quad C(i,(k-1)M_i+2)I_{\tilde{M}_i} \quad \ldots \quad C(i,kM_i)I_{\tilde{M}_i}\} \tag{12}$$

$$M_i = \prod_{j=0}^{i-1} n_j$$

$$M = M_N$$

$$\tilde{M}_N = 1$$

$$n_0 = 1 \tag{13}$$

We use $I_{\tilde{M}_i}$ to denote the $n\tilde{M}_i \times n\tilde{M}_i$ identity matrix. Noise processes $W_N(k)$ and $V_i(k)$ are independent, white, and zero-mean, and are given as follows:

$$W_N(k) = B_N(k) \begin{bmatrix} w(N, kM) \\ w(N, kM+1) \\ \vdots \\ w(N, kM+M-1) \end{bmatrix} \tag{14}$$

$$V_i(k) = \begin{bmatrix} v(i, (k-1)M_i + 1) \\ v(i, (k-1)M_i + 2) \\ \vdots \\ v(i, kM_i) \end{bmatrix} \tag{15}$$

where $B_N(k)$ is given as follows:

$$B_N(k) = \begin{bmatrix} I & 0 & \cdots & 0 \\ A(N, km+1) & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \prod_{l=M-1}^{1} A(N, kM+l) & \prod_{l=M-1}^{2} A(N, kM+l) & \cdots & I \end{bmatrix} \tag{16}$$

$W_N(k)$ and $V_i(k)$ have covariance matrices $Q_N(k)$ and $R_i(k)$, which are given as follows:

$$\begin{aligned} Q_N(k) &= B_N(k)\,\mathrm{diag}\big\{ Q(N, kM) \quad Q(N, kM+1) \quad \cdots \\ &\qquad Q(N, kM+M-1) \big\} B_N^T(k) \\ R_i(k) &= \mathrm{diag}\big\{ R\big(i, (k-1)M_i + 1\big) \quad R\big(i, (k-1)M_i + 2\big) \quad \cdots \\ &\qquad R(i, kM_i) \big\} \end{aligned} \tag{17}$$

Example

Consider the multirate, multisensor system shown in Fig. 1. In this case, sensor system 2 is sampled twice as often as sensor system 1, and sensor system 3 is sampled three times as often as sensor system 2. Therefore, $n_1 = 3$ and $n_2 = 2$. From (13) we obtain $M_1 = 1$, $M_2 = 3$, and $M_3 = 6$. From (13) we also obtain $\tilde{M}_1 = 6$, $\tilde{M}_2 = 2$, and $\tilde{M}_3 = 1$. The new state vector is obtained from (9) as the following function of the original state vector:

$$X_N(k) = \begin{bmatrix} x(N, 6k-5) \\ x(N, 6k-4) \\ x(N, 6k-3) \\ x(N, 6k-2) \\ x(N, 6k-1) \\ x(N, 6k) \end{bmatrix} \tag{18}$$

The three sensor system measurements are obtained from (11) as follows:

$$\begin{aligned} Z_1(k) &= \big[z(1, k) \big] \\ Z_2(k) &= \begin{bmatrix} z(2, 3k-2) \\ z(2, 3k-1) \\ z(2, 3k) \end{bmatrix} \\ Z_3(k) &= \begin{bmatrix} z(3, 6k-5) \\ z(3, 6k-4) \\ z(3, 6k-3) \\ z(3, 6k-2) \\ z(3, 6k-1) \\ z(3, 6k) \end{bmatrix} \end{aligned} \tag{19}$$

At this point we can use a Kalman filter with the state space model and sensor system 1 to obtain a state estimate; we can use a second Kalman filter with the state space model and sensor system 2 to obtain another state estimate; and we can use a third Kalman filter with the state space model and sensor system 3 to obtain yet a third state estimate. The third Kalman filter will be optimal. The first two Kalman filters will be suboptimal because of the asynchronous sampling and the approximation of (6). However, even though the third Kalman filter is optimal, we can still improve on its state estimate by using the additional information from the other two sensor systems.

We can use a neural network to combine the three state estimates at each point in time, and obtain a better estimate than either sensor system could provide by itself. We discuss this data fusion approach in the next section.

3. State estimation and data fusion

In this section a hybrid method to fuse the N state estimates is presented. First, conventional Kalman filters are used to estimate the state at each of the sensor system time scales, using the state space model along with one Kalman filter for each sensor system as shown in (7) and (8). Then a neural network is used to fuse the state estimates and give the final estimate. In this section it is assumed that all measurements are available and that there are no missing measurements.

In this paper a multilayer perceptron (MLP) is used as the neural network. An MLP is a network of simple neurons called perceptrons. Each perceptron computes a single output from multiple inputs by forming a weighted linear combination of the inputs, and then passing the result through a nonlinear activation function:

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + a \right) = \varphi\big(w^T x + a \big) \tag{20}$$

where w denotes the vector of weights, x denotes the n-element input vector to the neuron, a denotes the bias input to the neuron, and φ denotes the nonlinear activation function. The original Rosenblatt perceptron used a Heaviside step function as the activation function φ [19]. More recently, and especially in multilayer networks, the activation function is often chosen to be the logistic sigmoid $1/(1+e^{-x})$ or the hyperbolic tangent $\tanh(x)$ [20]. They are related as follows:

$$\frac{\tanh(x) + 1}{2} = \frac{1}{1 + e^{-2x}}$$

These functions are used because they are mathematically convenient, and because they are close to linear near $x = 0$ while saturating rather quickly as x becomes greater than or less than 0. However, a single perceptron is not very useful because of its limited mapping ability [16]. Therefore, a typical MLP network consists of a set of source nodes forming the input layer, one or more hidden layers of computational nodes, and a layer of output nodes. The input signal propagates through the network one layer at a time. The computations performed by such a feedforward network with a single hidden layer containing m neurons, with a nonlinear activation function at the hidden layer, and a linear output layer, can be written as follows:

$$y = f(x) = B^T \varphi\big(W^T x + a \big) + b \tag{21}$$

where x is the n-element vector of inputs, W is the $n \times m$ weight matrix from the input layer to the hidden layer, and a is the m-element bias vector of the hidden layer. The function φ denotes an element-wise nonlinearity. B is the $m \times p$ weight matrix of the second layer, b is the p-element bias vector of the output layer, and y is the p-element output vector.

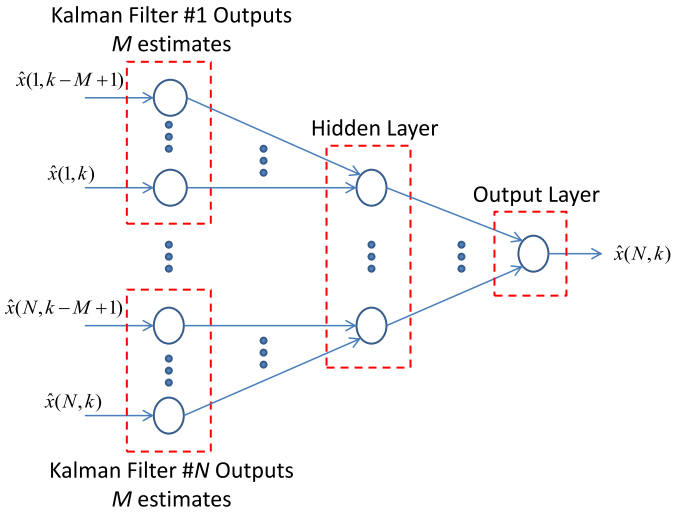

Fig. 2. Multilayer perceptron for fusing Kalman filter estimates. There are N sensor systems in this figure, and thus N sets of Kalman filter estimates. Each set of Kalman filter estimates includes M estimates as shown in (9). This MLP includes MN inputs, a user-determined number of hidden neurons, and 1 output.

In this paper an MLP with one hidden layer, described by (21), is used to fuse the results of multiple Kalman filters. The MLP is shown in Fig. 2. There are N Kalman filters, one for each sampling rate. Each Kalman filter outputs M state estimates, as shown in (9). These MN estimates are passed through the MLP to a given number of hidden neurons, where the number of hidden neurons are chosen by the designer as a tuning parameter. The outputs of the hidden neurons are passed to the output neuron, and the output of the MLP is the fused state estimate based on the highest sample rate. Two examples are presented in the following section.

4. Simulation results

In both examples, 78% of the input data are used for training (700 out of 900 points), and the rest of the data are used as test data. In this section two examples are studied.

4.1. Example 1

The first example is the scalar system studied in [22]. The system model is given by (1) and (2), and there are 3 sensor systems; that is, $N = 3$. The relationships between the sampling rates are $S_3/S_2 = 2$ and $S_2/S_1 = 3$. Note that this is the relationship depicted in Fig. 1 and discussed at the end of Section 2. The system matrices are given as follows: $A(3,k) = 0.906$, and $C(i,k) = 1$ for $i = 1, 2, 3$. Initial conditions are $x(3,0) = 10$, and the initial variance of the three Kalman filter state estimates is 10. System and measurement noise variances are $Q(3) = 1$, $R(3) = 1$, $R(2) = 0.1$, and $R(1) = 0.01$. The simulations were run for 900 time steps, with 700 data points used for training data and 200 data points used for test data. The neural network was trained with backpropagation until the training error decreased to 0.1 [9].

In this example, the neural network has 18 inputs, 1 hidden layer, and 1 output. The 18 inputs are the state estimates from the 3 Kalman filters. Each of the 3 Kalman filters outputs the state estimate for 6 time steps as shown in the state vector description in (18).

The true state, along with the estimated state and the estimation errors, are shown in Figs. 3–6. Fig. 3 depicts the true state and the estimated state using the algorithm of [22]. Fig. 4 depicts the true state and the estimated state using the algorithm proposed in this paper. Figs. 5 and 6 show the estimation errors of the two algorithms.

Figs. 3–6 lead to the conclusion that the algorithm proposed in this paper can be relatively effective at fusing data from multiple asynchronous sensor systems.

Note that [22] claims to derive a minimum covariance estimator. That claim is true, but only if the approximation in (6) is error-free. If (6) is only approximately true, as with the asynchronous sensor systems used in this example that are illustrated in Fig. 1, then the estimator in [22] is only approximately optimal. The results of this example confirm that nonlinear estimators can provide better performance.

4.2. Example 2

The second example is a simple radar tracking system and is also taken from [22]:

$$x(t+1) = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} x(t) + w(t) \tag{22}$$

$$z(i,t) = C(i,t)x(i,t) + v(i,t), \quad i = 1, 2, 3 \tag{23}$$

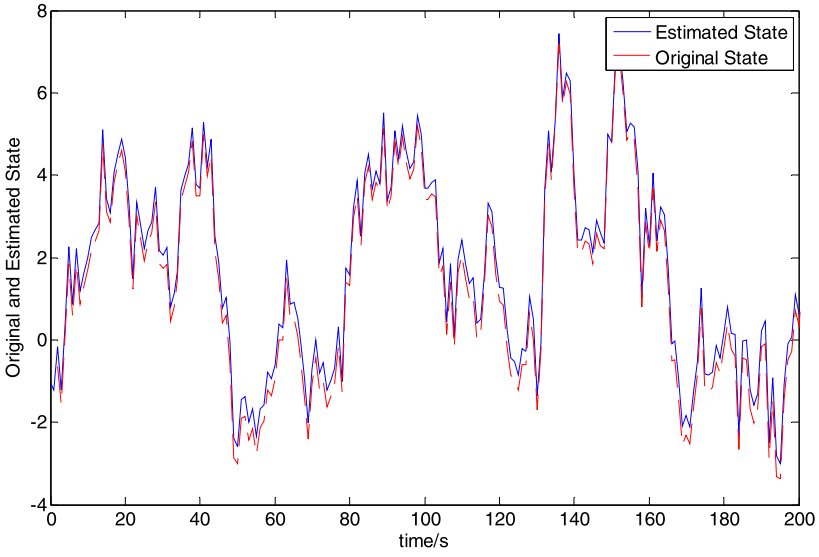

Fig. 3. Example 1 – True state and estimated state using the algorithm of [22].

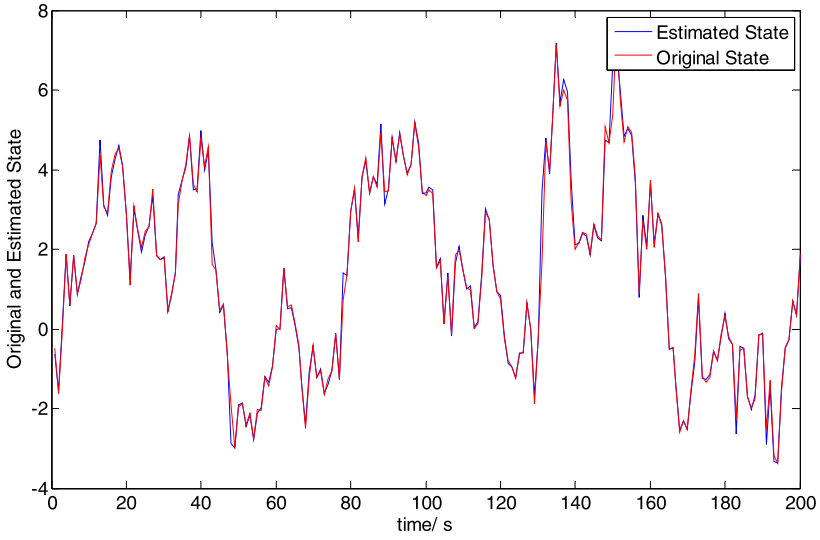

Fig. 4. Example 1 – True state and estimated state using the algorithm proposed in this paper.

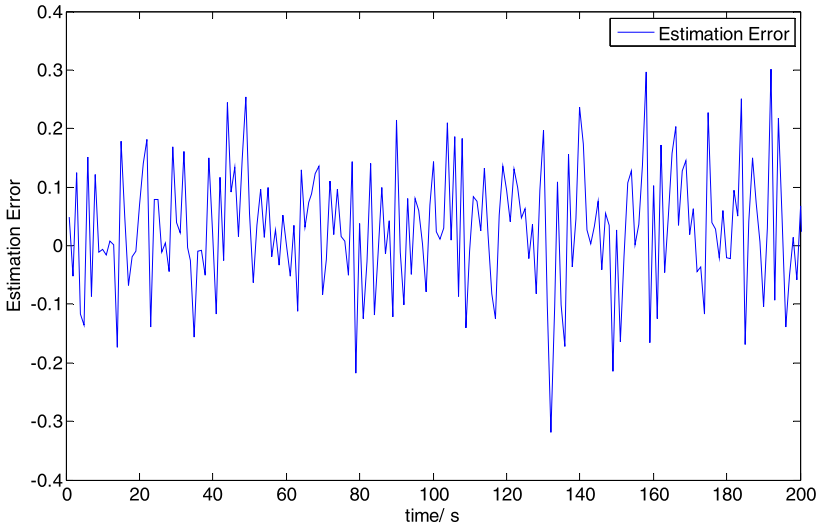

Fig. 5. Example 1 – Estimation error using the algorithm of [22].

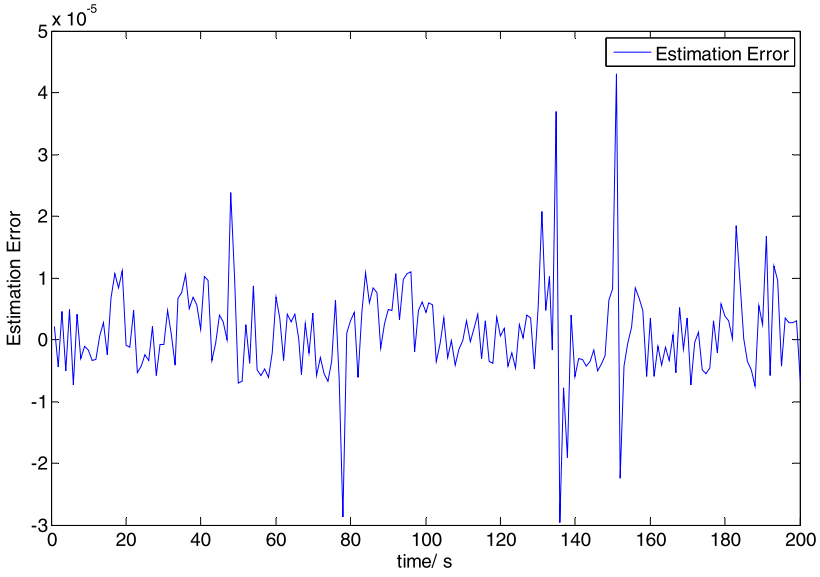

Fig. 6. Example 1 – Estimation error using the algorithm proposed in this paper.

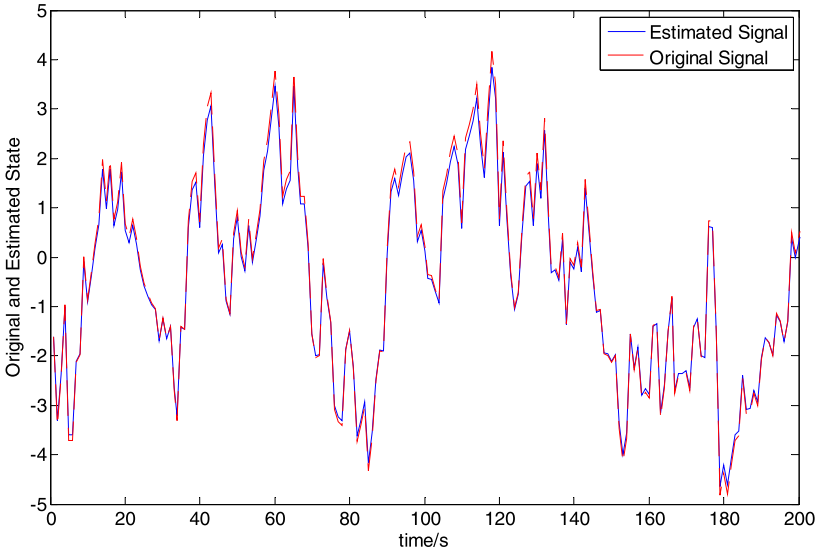

Fig. 7. Example 2 – True position and estimated position using the algorithm of [22]. RMS error = 0.1928.

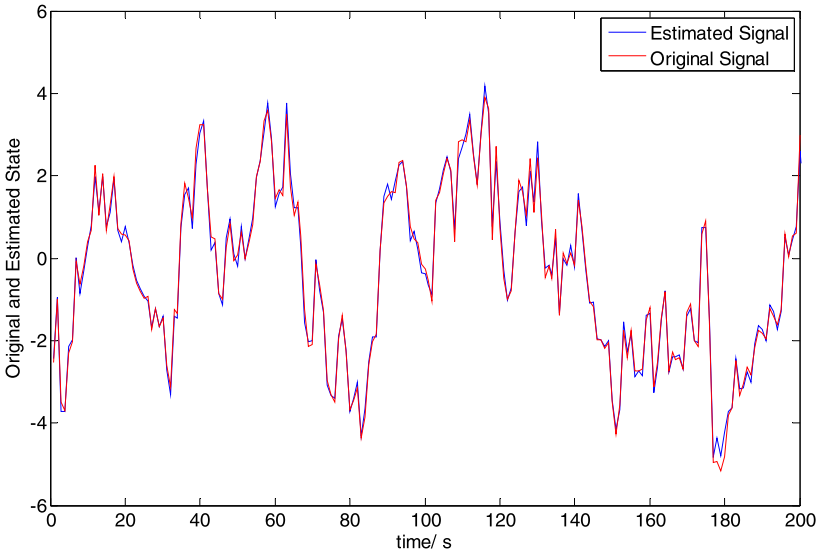

Fig. 8. Example 2 – True position and estimated position using the algorithm proposed in this paper. RMS Error = 0.008.

where $T = 0.01$ s is the discretization step size, and the system is observed by 3 sensors. This system has 3 states, which are position, speed, and acceleration. $z(i,t)$ are the measurements of 3 sensors with measurement matrices $C(1) = [0\ 0\ 1]$, $C(2) = [0\ 1\ 0]$ and $C(3) = [1\ 0\ 0]$ respectively. $w(t)$ and $v(i,t)$ are zero-mean white independent noise processes with variances $Q = 1$, $R(3) = 20$, $R(2) = 15$, and $R(1) = 8$. The initial state is $x(0) = 0$, and the initial estimation variance for all 3 Kalman filters is $P_0 = 0.1I_3$. The relationship between the sensor system sampling rates is the same as in the previous example: $S_3/S_2 = 2$ and $S_2/S_1 = 3$ (see Fig. 1).

Our goal in this example is to estimate position. Fig. 7 shows the position estimate using the approach of [22], while Fig. 8 shows the position estimate of the algorithm proposed in this paper. It can be seen that the estimate based on the algorithm proposed in this paper is more accurate. Fig. 7 shows that the estimation error using the approach of [22] is especially high when the velocity changes sign. The mean square error of the estimate using the approach of [22], and that using the algorithm proposed in this paper, is 0.1928 and 8×10^{-3}, respectively. This difference in estimation error indicates the effectiveness of the proposed algorithm.

5. Conclusion

In this paper a multirate multisensor data fusion problem for linear systems is studied. The highest sampling rate is uniform, but the lower sampling rates are different and asynchronous. Each sensor system observes the state independently. Assuming that the state dynamics at the highest sampling rate are known, and that the sampling rate ratios between sensor systems are known positive integers, an approximate single-rate model for the original system is developed. Parallel Kalman filters are used for state estimation at each sampling rate, and a neural network is used to fuse the state estimates. Simulation results show better performance than a previously-published method based on federated Kalman filtering, due to the neural network based sensor fusion approach in this paper.

Future work could include extending the proposed algorithm to the cases where the sampling rate ratios are non-integer, or where the sampling rate ratios vary stochastically, or where the highest sampling rate ratio is non-uniform. Other future work could include the use of nonlinear networks other than neural networks

for data fusion of the multiple Kalman filter outputs (for example, fuzzy systems).

Conflict of interest statement

The authors confirm that there are no known conflicts of interest associated with this publication and that there has been no significant financial support for this work that could have influenced its outcome.

References

[1] Y. Bar-Shalom, X.R. Li, T. Kirubarajam, Estimation with Application to Tracking and Navigation, John Wiley and Sons, New York, 2001.

[2] E. Bustamante, E. Guijarro, F.-J. García-Diego, S. Balasch, A. Hospitaler, A. Torres, Multisensor system for isotemporal measurements to assess indoor climatic conditions in poultry farms, Sensors 2012 (12) (2012) 5752–5774.

[3] N.A. Carlson, Federated square root filter for decentralized parallel processors, IEEE Trans. Aerosp. Electron. Syst. 3 (1990) 517–525.

[4] N.A. Carlson, Federated Kalman filter simulation results, J. Inst. Navig. 41 (3) (1994) 297–321.

[5] K.C. Chou, A.S. Willsky, A. Benveneste, Multiscale recursive estimation, data fusion and regularization, IEEE Trans. Autom. Control 39 (3) (March 1994) 464–478.

[6] D.W. Fincher, D.F. Max, Multi-sensor data fusion using neural networks, in: IEEE International Conference on Systems, Man and Cybernetics, Los Angeles, USA, 1990, pp. 835–838.

[7] M. Grewal, A. Andrews, C. Bartone, Global Navigation Satellite Systems, Inertial Navigation, and Integration, 3rd edition, Wiley-Interscience, 2013.

[8] D.C. Hall, J. Llinas, An introduction to multisensor data fusion, Proc. IEEE 85 (January 1997) 6–23.

[9] R. Hecht-Nielsen, Theory of the backpropagation neural network, in: International Joint Conference on Neural Networks, Washington, DC, June 1989, pp. 593–605.

[10] J. Hou, Z. Jing, T. Gao, Y. Yang, Adaptive phased-array tracking in the stand-off jammer using H infinity filter, in: International Conference on Intelligent Control and Information Processing, Beijing, China, June 2013, pp. 649–654.

[11] M. Kastek, R. Dulski, M. Zyczkowski, M. Szustakowski, P. Trzaskawka, et al., Multisensor system for the protection of critical infrastructure of a seaport, in: Proc. SPIE 8388, Unattended Ground, Sea, and Air Sensor Technologies and Applications XIV, 83880M, May 2012.

[12] M. Kazerooni, F. Shabaninia, M. Vaziri, S. Vadhva, Federated ensemble Kalman filter in no reset mode design, in: IEEE International Conference on Information Reuse and Integration, San Fransisco, USA, Aug. 14–16, 2013.

[13] B. Khaleghi, A. Khamis, F.O. Karray, S.N. Razavi, Multisensor data fusion: a review of the state-of-the-art, Inf. Fusion 14 (1) (January 2013) 28–44.

[14] T. Khan, P. Ramuhalli, S. Dass, Particle-filter-based multisensor fusion for solving low-frequency electromagnetic NDE inverse problems, IEEE Trans. Instrum. Meas. 60 (6) (March 2011) 2142–2153.

[15] F. Liu, X. Cai, B. Shi, Y. Wang, Multisensor data fusion based on genetic algorithm, in: Proc. SPIE 2898, Electronic Imaging and Multimedia Systems, vol. 43, September 1996.

[16] M. Minsky, S. Papert, Perceptrons, MIT Press, 1969.

[17] W. Naeem, R. Sutton, T. Xu, An integrated multi-sensor data fusion algorithm and autopilot implementation in an uninhabited surface craft, Ocean Eng. 39 (January 2012) 43–52.

[18] Z. Ni, S. Sunderrajan, A. Rahimi, B. Manjunath, Distributed particle filter tracking with online multiple instance learning in a camera sensor network, in: IEEE International Conference on Image Processing, Hong Kong, September 2010, pp. 37–40.

[19] F. Rosenblatt, Perceptron simulation experiments, Proc. IRE 48 (3) (March 1960) 301–309.

[20] T. Sorsa, H. Koivo, H. Koivisto, Neural networks in process fault diagnosis, IEEE Trans. Syst. Man Cybern. 21 (4) (July 1991) 815–825.

[21] C.-Y. Xiao, J. Ma, S.-L. Sun, Design of information fusion filter for a class of multi-sensor asynchronous sampling systems, in: Chinese Control and Decision Conference, Mianyang, China, May 2011, pp. 1081–1084.

[22] L.P. Yan, B.S. Liu, D.H. Zhou, The modeling and estimation of asynchronous multirate multisensor dynamic systems, Aerosp. Sci. Technol. 10 (2006) 63–71.

[23] L. Zhang, X.L. Wu, Q. Pan, H.C. Zhang, Multiresolution modeling and estimation of multisensor data, IEEE Trans. Signal Process. 52 (11) (2004) 3170–3182.

[24] Z. Zhou, Y. Li, J. Liu, G. Li, Kalman-filter-based heterogeneous multi-sensor navigation, IEEE Trans. Aerosp. Electron. Syst. 49 (4) (October 2013) 2146–2157.