 Preprint of 10.1109/TIM.2020.3041825

Dual Neural Extended Kalman Filtering Approach for Multirate Sensor Data Fusion

Jingyi Wang, Yousef Alipouri, *Member, IEEE,* and Biao Huang, *Fellow, IEEE*

***Abstract*—The dual neural extended Kalman filter is proposed in this paper to compensate for model inaccuracies and violations of noise assumption in the extended Kalman filter based multirate sensor fusion. The dual neural extended Kalman filter employs two neural networks to improve process state and output predictions through simultaneous state and parameter estimations using state vector augmentation. Taking advantage of frequent but less accurate measurements and infrequent but more accurate measurements provides a multirate parameter update strategy for neural networks training in the proposed appraoch. Two numerical examples and one industrial application demonstrate the effectiveness of the proposed method to compensate for inadequate process knowledge to improve multirate sensor fusion performance.**

***Index Terms*—Sensor fusion, extended Kalman filter, neural network, multirate measurement, colored noise.**

I. Introduction

The extended Kalman filter (EKF) has been widely applied to solving multisensor data fusion problems to improve state estimation accuracy from individual measurements. This algorithm uses the newly received measurements to correct the last predicted states while considering influence of process noises during online estimation [1]. In the EKF based sensor fusion process, fusion accuracy depends on the accuracy of state and output models that are used to describe system dynamics. The model inaccuracies need to be compensated during online operation to prevent inaccurate fusion results [2]. Furthermore, even though the EKF algorithm handles process randomness effectively, it assumes that in its state space formulation, process states and measurements are contaminated by white noises, and violations of this assumption may cause the EKF estimates to be inaccurate or even divergent [3], [4].

To address the above problems, much work has been conducted. In [5], the Kalman filter based robust state estimation was proposed to deal with deterministic errors caused by measuring instrument failures and sensor accuracy limitations. However, this method requires to derive error models in advance, and optimal state estimates are obtained only after an iterative process. Xu *et al.* put forward the idea of using a first-order Gauss-Markov process to compensate for inaccurately modeled accelerations in the Kalman filter based orbit determination [6]. However, this method is limited to solving specific orbit determination problems. In [3], the problem of colored noise instead of white noise was considered for sensor fusion using the Kalman filter. The adaptive fuzzy logic system (AFLS) is applied to prevent the Kalman filter estimates from diverging. However, this method is not applicable to solving model inaccuracy problems. Overall, these errors are difficult to capture using simple linear time-invariant models.

An effective online compensation technique for model inaccuracies and violations of noise assumption is desirable to ensure the fusion results to be accurate and reliable. In [7], a combination of the EKF algorithm and artificial neural network was proposed as the neural EKF (NEKF) technique for online model compensation and adjustment. In [8]–[12], this approach was used for online target tracking, improving state estimation accuracy, online sensor calibration, and online modeling. However, the NEKF approach only has either state or output measurement model compensated by an artificial neural network but not both, and is divided into state model compensation NEKF (SNEKF) and measurement model compensation NEKF (MNEKF). The neural networks utilized by them are named as state model compensation neural network (SNN) and measurement model compensation neural network (MNN) accordingly.

However, if both state and output measurement models have model-plant mismatches, the NEKF approach will be inadequate. Additionally, the above mentioned noise assumption compensation and model calibration approaches assume all sensors have the same sampling rate. In reality, online measurements that are frequently sampled and inexpensive to obtain, usually lack precision. Infrequent measurements, such as lab analysis, are commonly employed to provide more accurate measurements for quality variables.

Xie *et al.* put forward an adaptive soft sensor in the framework of the Kalman filter to synthesize frequent-sampling online data and high-accuracy infrequent lab data to provide more reliable estimation of quality variables [13]. Fusion in presence of slow-rate integrated measurement was analyzed in [14] to fuse the lab analysis results of gradually collected samples and fast-rate online measurements to improve state estimation accuracy of the process. However, these multirate sensor fusion approaches have not well investigated the model inaccuracies during fusion process.

When applying multirate sensor fusion, compensations for model deficiencies and violations of noise assumption are desirable as they achieve more robust and reliable real-time state estimation. To fulfill this objective, in this paper, the dual NEKF (DNKEF) approach is proposed. The proposed method uses multirate measurements to compensate for inadequate process knowledge through simultaneous state and

This work was supported in part by Natural Sciences and Engineering Research Council of Canada. *(Corresponding Author: Biao Huang.)*

J. Wang, Y. Alipouri, and B. Huang are with Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Alberta T6G 2V4, Canada (E-mail: wang9@ualberta.ca; alipouri@ualberta.ca; bhuang@ualberta.ca).

parameter estimations. In the primary separation vessel (PSV) system, different measurements are used to estimate the froth-middlings interface level. However, each of them is subject to estimation errors. Fusion of these measurements is desired to take advantage of all these sources to provide more accurate and robust interface level estimation.

The main contribution of this paper is to develop a multirate sensor data fusion approach, which is tolerant to model inaccuracies and violations of noise assumption. The weights and biases of the two employed neural networks are trained in a multirate strategy. By utilizing infrequent accurate measurements, this proposed approach provides more reliable state estimation and parameter training than the NEKF algorithm. In addition to the PSV system, the DNEKF method can be applied to general system that consists of multiple sensors and has high-accuracy but possibly slow-rate measurements to improve quality variable estimation performance.

The organization of this paper is as follows. Section II introduces the EKF based parallel sensor fusion algorithm. Section III explains problems caused by model inaccuracies and violations of noise assumption when applying the EKF based sensor fusion. Section IV provides a detail description of the DNEKF approach. In Section V, the DNEKF fusion technique is applied to two numerical examples as well as one industrial PSV interface level estimation problem to demonstrate its advantages over the traditional EKF based fusion and the NEKF algorithm. Finally, Section VI concludes this paper.

II. EKF based Parallel Sensor Fusion

Measurements from different sensors can be fused in parallel using the Kalman filter algorithm by augmenting the output vector to provide fusion results for quality variables [15], [16]. Two steps are involved in the Kalman filter algorithm, namely, prediction and correction. The EKF linearizes the nonlinear models around the current state to make the linear Kalman filter applicable [17]. Consider a nonlinear process:

$$x_t = f(x_{t-1}) + w_{t-1} \tag{1}$$

$$y_t = h(x_t) + v_t \tag{2}$$

where $f(\cdot)$ and $h(\cdot)$ represent the nonlinear state and output models, respectively, x_t is the state, and y_t denotes measurements from different sensors. Process and measurement noises are denoted as w_t and v_t, which are assumed to be zero-mean, uncorrelated, Gaussian white noises. Their corresponding covariance matrices are Q and R, respectively. Then the EKF based parallel fusion algorithm is performed as follows [18].

Prediction:

$$\hat{x}_t^- = f(\hat{x}_{t-1}) \tag{3}$$

$$P_t^- = F_{t-1}P_{t-1}F_{t-1}^T + Q \tag{4}$$

where $\hat{x}_{t-1}$ and $\hat{x}_t^-$ are posterior and prior estimates of the state, and P_{t-1} and P_t^- represent corrected and prior state estimation error covariance matrices. The Jacobian matrix, F_{t-1}, is obtained through linearization of the nonlinear state model, $f(\cdot)$, at $\hat{x}_{t-1}$, namely, $F_{t-1} = \frac{\partial f}{\partial x}\big|_{\hat{x}_{t-1}}$.

Correction:

In the correction steps, the Kalman gain is computed as:

$$K_t = P_t^- H_t^T (H_t P_t^- H_t^T + R)^{-1} \tag{5}$$

where H_t is the Jacobian matrix representing linearization of nonlinear output model, $h(\cdot)$, at $\hat{x}_t^-$, and it is calculated as $H_t = \frac{\partial h}{\partial x}\big|_{\hat{x}_t^-}$. Then the posterior state estimate, $\hat{x}_t$, is obtained as:

$$\hat{x}_t = \hat{x}_t^- + K_t(y_t - \hat{y}_t) \tag{6}$$

where y_t represents actual measurements, and $\hat{y}_t$ denotes model predicted output values, which can be calculated as:

$$\hat{y}_t = h(\hat{x}_t^-) \tag{7}$$

Then the corrected state estimation error covariance is:

$$P_t = (I - K_t H_t) P_t^- \tag{8}$$

During multiple sensor fusion process, each sensor has its own output measurement model. If m sensors are utilized, then m output predictions can be obtained as:

$$\hat{y}_t = \begin{bmatrix} \hat{y}_{1,t} \\ \hat{y}_{2,t} \\ \vdots \\ \hat{y}_{m,t} \end{bmatrix} = \begin{bmatrix} h_1(\hat{x}_t^-) \\ h_2(\hat{x}_t^-) \\ \vdots \\ h_m(\hat{x}_t^-) \end{bmatrix} \tag{9}$$

The corresponding Jacobian matrix, H_t, can be calculated as:

$$H_t = \begin{bmatrix} \frac{\partial h_1}{\partial x}\big|_{\hat{x}_t^-} \\ \frac{\partial h_2}{\partial x}\big|_{\hat{x}_t^-} \\ \vdots \\ \frac{\partial h_m}{\partial x}\big|_{\hat{x}_t^-} \end{bmatrix} \tag{10}$$

Then the standard EKF procedure can be applied to perform sensor fusion through matrix augmentation [18].

III. Problem Statement

The EKF algorithm has been widely applied to finding the fusion criterion in solving multisensor data fusion problems. The inaccurate process models and noise assumption can degrade the accuracy of the EKF estimates. Consider a nonlinear system in (1) and (2), with slightly different model notations:

$$x_t = f_{true}(x_{t-1}) + w_{t-1} \tag{11}$$

$$y_t = h_{true}(x_t) + v_t \tag{12}$$

where $f_{true}(\cdot)$ and $h_{true}(\cdot)$ indicate the true process state and output models. Several reasons can lead to model mismatches. First, model built based on inadequate process knowledge may cause large deviations between the model and the actual process. Besides, process condition changes such as seasonal shifts can result in deviation of the current system dynamics from the previous. Moreover, the EKF algorithm is developed based on the white noise assumptions in the state space model. Colored state or measurement noise can lead to inaccurate state estimation or even divergence [3], [4].

Typically, two types of measurements are utilized in industrial processes. Comparing with frequent measurements, infrequent measurements normally have less availability but have higher accuracy. In Fig. 1, small ticks represent frequent sampling instants, t, and squares represent infrequent sampling instants, s. The infrequent measurement, y_s, is available at multiple of frequent sampling intervals. The objective is to design an online multirate sensor data fusion algorithm that is tolerant to model inaccuracies and violations of noise assumption.

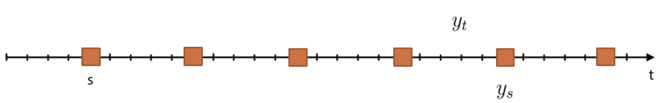

Fig. 1. Sampling time strategy for frequent and infrequent measurements.

IV. DNEKF Approach for Multirate Sensor Data Fusion

A. DNEKF Procedure

The proposed DNEKF approach is an extension of the NEKF approach, solving multirate sensor fusion problems. The DNEKF employs both SNN and MNN simultaneously to improve state estimation and output predictions using multirate measurements. The violations of noise assumption are also handled by the proposed approach. As mentioned in Section III, several reasons can lead to inaccuracies of system models. The relations between inaccurate predictions and actual values may be written as:

$$x_t = f(\hat{x}_{t-1}) + d_{t-1} = \hat{x}^-_{d,t} + d_{t-1} \tag{13}$$

$$y_t = h(\hat{x}^-_t) + l_t = \hat{y}_{d,t} + l_t \tag{14}$$

where $\hat{x}_{t-1}$ and $\hat{x}^-_t$ are posterior and prior state estimates from the EKF algorithm, $f(\cdot)$ and $h(\cdot)$ represent inaccurate state and output models, which result in the inaccurate predictions, $\hat{x}^-_{d,t}$ and $\hat{y}_{d,t}$, respectively. The deviations between inaccurate predictions of states and outputs and their corresponding true values, are denoted as d_t and l_t, respectively. These deviations can also include errors caused by violations of noise assumption. To compensate for the inaccuracies, neural networks, which do not require the pre-defined model structure, are utilized. After neural network compensations, the modified predictions become:

$$\hat{\bar{x}}^-_t = f(\hat{x}_{t-1}) + NN_x(\hat{x}_{t-1}, \hat{\phi}_{x,t-1}) \tag{15}$$

$$\hat{y}_t = h(\hat{x}^-_t) + NN_y(\hat{x}^-_t, \hat{\phi}^-_{y,t}) \tag{16}$$

where $\hat{x}^-_t$ and $\hat{y}_t$ are the modified state and output predictions; NN_x and NN_y are the outputs from SNN and MNN, with posterior state estimate, $\hat{x}_{t-1}$, and modified prior state estimate, $\hat{x}^-_t$, as their inputs, respectively; $\hat{\phi}_x$ and $\hat{\phi}_y$ are weights and biases of SNN and MNN, respectively. The relations between modified predictions and actual values are:

$$x_t = \hat{x}^-_t + \varepsilon_{t-1} \tag{17}$$

$$y_t = \hat{y}_t + \mu_t \tag{18}$$

where ε_t and μ_t are the prediction errors between modified predictions and true values after adding NN_x and NN_y compensations. Comparing with $\hat{x}^-_{d,t}$ and $\hat{y}_{d,t}$, the modified predictions, $\hat{x}^-_t$ and $\hat{y}_t$, are expected to have smaller differences from the actual states and measurements as they are compensated by the neural network models. The structure of SNN and MNN can be chosen as feed-forward neural networks with user-defined activation functions.

The DNEKF first uses a SNN to compensate for state prediction inaccuracy, then uses an MNN to improve frequent output predictions. To obtain accurate compensated results, proper weights and biases need to be trained for the two neural networks. The proposed approach estimates process states and neural network parameters simultaneously through state vector augmentation, and updates the weights and biases for the two neural networks through a multirate strategy.

In the NEKF method, single sampling rate measurement residuals are used to provide neural network training references. However, using the frequent measurement residuals to train SNN will usually not be reliable and accurate. As aforementioned, the frequent measurements are of lower accuracy and output models can have significant mismatches. Reducing the frequent measurement residuals cannot ensure smaller state prediction errors. Without proper references, the parameters of SNN cannot converge to correct values, and its predictions will not be accurate.

In the proposed approach, a multirate parameter update strategy is proposed to update ϕ_x and ϕ_y using respective measurement residuals. Frequent measurement residuals are used to train MNN parameters, while the training of SNN is according to infrequent measurements. In the DNEKF, the state vector is augmented as $\bar{x} = \begin{bmatrix} x & \phi_x & \phi_y \end{bmatrix}^T$. During online implementation, ϕ_y is updated at a frequent sampling rate, while ϕ_x is updated only when the infrequent measurements arrive. The DNEKF procedure is illustrated in Fig. 2.

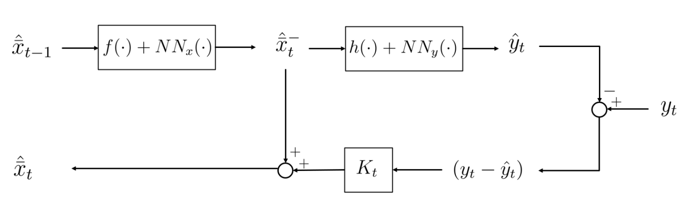

Fig. 2. Graphical illustration of the DNKEF approach.

The similar name convention as in [19], [20] is adopted in this paper, that is, *minor instants* mean the time instants when only frequent measurements are available, and *major instants* are the time instants when both types of measurements become available. In the DNEKF, the neural network parameters, ϕ_x and ϕ_y, are treated as states to be estimated, and remain unchanged in the prediction step. Then the state and output predictions are obtained as:

$$\hat{\bar{x}}_t^- = \begin{bmatrix} f(\hat{x}_{t-1}) + NN_x(\hat{x}_{t-1}, \hat{\phi}_{x,t-1}) \\ \hat{\phi}_{x,t-1} \\ \hat{\phi}_{y,t-1} \end{bmatrix} \tag{19}$$

$$\hat{y}_t = h(\hat{x}_t^-) + NN_y(\hat{x}_t^-, \hat{\phi}_{y,t}^-) \tag{20}$$

where $\hat{\bar{x}}_t^-$ is augmented prior state estimate. As neural network parameters are not updated in prediction step, $\hat{\phi}_t^- = \hat{\phi}_{t-1}$.

The state estimation error covariance in (4) becomes:

$$P_t^- = \bar{F}_{t-1} P_{t-1} \bar{F}_{t-1}^T + Q \tag{21}$$

where $\bar{F}_{t-1}$ is the augmented Jacobian matrix [21]. At *minor instants*, for an n-state system, $\bar{F}_{t-1}$ is calculated as:

$$\begin{aligned} \bar{F}_{t-1} &= \begin{bmatrix} \frac{\partial(f+NN_x)}{\partial x}\Big|_{\hat{x}_{t-1}} & \frac{\partial(f+NN_x)}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t-1}} & \frac{\partial(f+NN_x)}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t-1}} \\ \mathbf{0} & I_{\phi_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{\phi_y} \end{bmatrix} \\ &= \begin{bmatrix} F_{t-1} + \frac{\partial NN_x}{\partial x}\Big|_{\hat{x}_{t-1}} & \frac{\partial NN_x}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t-1}} & \frac{\partial NN_x}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t-1}} \\ \mathbf{0} & I_{\phi_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{\phi_y} \end{bmatrix} \\ &= \begin{bmatrix} F_{t-1} + \frac{\partial NN_x}{\partial x}\Big|_{\hat{x}_{t-1}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{\phi_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{\phi_y} \end{bmatrix}_{p\times p} \end{aligned} \tag{22}$$

where $p = \phi_x + \phi_y + n$, and $\frac{\partial NN_x}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t-1}} = \frac{\partial NN_x}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t-1}} = 0$, since weights and biases of SNN are not updated when infrequent accurate measurements are unavailable, and SNN does not contain ϕ_y values. Posterior state estimate, $\hat{x}_{t-1}$, is used as input to SNN to modify state predictions continuously.

The Kalman gain matrix in (5) is:

$$K_t = P_t^- \bar{H}_t^T (\bar{H}_t P_t^- \bar{H}_t^T + R)^{-1} \tag{23}$$

where $\bar{H}_t$ is the augmented Jacobian matrix. For a system with m frequent sensors, $\bar{H}_t$ is derived as:

$$\begin{aligned} \bar{H}_t &= \begin{bmatrix} \frac{\partial(h+NN_y)}{\partial x}\Big|_{\hat{x}_t^-} & \frac{\partial(h+NN_y)}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t}^-} & \frac{\partial(h+NN_y)}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t}^-} \end{bmatrix} \\ &= \begin{bmatrix} H_t + \frac{\partial NN_y}{\partial x}\Big|_{\hat{x}_t^-} & \frac{\partial NN_y}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t}^-} & \frac{\partial NN_y}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t}^-} \end{bmatrix} \\ &= \begin{bmatrix} H_t + \frac{\partial NN_y}{\partial x}\Big|_{\hat{x}_t^-} & \mathbf{0} & \frac{\partial NN_y}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t}^-} \end{bmatrix}_{m\times p} \end{aligned} \tag{24}$$

In (24), the derivative of NN_y with respect to the weights and biases of SNN can be computed as:

$$\begin{aligned} \frac{\partial NN_y}{\partial \phi_x} &= \frac{\partial NN_y}{\partial \hat{x}_t^-} \frac{\partial \hat{x}_t^-}{\partial \phi_x} \\ &= \frac{\partial NN_y}{\partial \hat{x}_t^-} \frac{\partial (f(\hat{x}_{t-1}) + NN_x)}{\partial \phi_x} \\ &= \frac{\partial NN_y}{\partial \hat{x}_t^-} \frac{\partial NN_x}{\partial \phi_x} \end{aligned} \tag{25}$$

In the correction step, state estimation and estimation error covariance are updated using following equations:

$$\hat{\bar{x}}_t = \hat{\bar{x}}_t^- + K_t(y_t - \hat{y}_t) \tag{26}$$

$$P_t = (I - K_t \bar{H}_t) P_t^- \tag{27}$$

where measurement residual, $y_t - \hat{y}_t$, is:

$$y_t - \hat{y}_t = \begin{bmatrix} y_{1,t} & -\Big(h_1(\hat{x}_t^-) + NN_{y,1}(\hat{x}_t^-, \hat{\phi}_{y,t}^-)\Big) \\ y_{2,t} & -\Big(h_2(\hat{x}_t^-) + NN_{y,2}(\hat{x}_t^-, \hat{\phi}_{y,t}^-)\Big) \\ \vdots & \\ y_{m,t} & -\Big(h_m(\hat{x}_t^-) + NN_{y,m}(\hat{x}_t^-, \hat{\phi}_{y,t}^-)\Big) \end{bmatrix} \tag{28}$$

At *major instants*, infrequent but accurate measurements are available, and the DNEKF steps are the same as those at *minor instants*. However, dimensions and elements for some matrices are different. The state prediction equation is the same as (19). While with availability of infrequent measurements, the output predictions are further augmented as:

$$\hat{y}_t = \begin{bmatrix} \hat{y}_{1,t} \\ \vdots \\ \hat{y}_{m,t} \\ \hat{y}_{1,s} \\ \vdots \\ \hat{y}_{n,s} \end{bmatrix} = \begin{bmatrix} h_1(\hat{x}_t^-) + NN_{y,1}(\hat{x}_t^-, \hat{\phi}_{y,t}^-) \\ \vdots \\ h_m(\hat{x}_t^-) + NN_{y,m}(\hat{x}_t^-, \hat{\phi}_{y,t}^-) \\ \hat{x}_{1,s}^- \\ \vdots \\ \hat{x}_{n,s}^- \end{bmatrix} \tag{29}$$

where $\hat{y}_s$ is infrequent measurement, and since it is considered as accurate measurement, $\hat{y}_s = \hat{x}_s^-$. The Jacobian matrix, $\bar{F}_{t-1}$, in (21) becomes:

$$\bar{F}_{t-1} = \begin{bmatrix} F_{t-1} + \frac{\partial NN_x}{\partial x}\Big|_{\hat{x}_{t-1}} & \frac{\partial NN_x}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t-1}} & \mathbf{0} \\ \mathbf{0} & I_{\phi_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{\phi_y} \end{bmatrix}_{p\times p} \tag{30}$$

Furthermore, the Jacobian matrix, $\bar{H}_t$, in (23) is:

$$\begin{aligned} \bar{H}_t &= \begin{bmatrix} \frac{\partial(h+NN_y)}{\partial x}\Big|_{\hat{x}_t^-} & \frac{\partial(h+NN_y)}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t}^-} & \frac{\partial(h+NN_y)}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t}^-} \\ I_n & \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} H_t + \frac{\partial NN_y}{\partial x}\Big|_{\hat{x}_t^-} & \frac{\partial NN_y}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t}^-} & \frac{\partial NN_y}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t}^-} \\ I_n & \mathbf{0} & \mathbf{0} \end{bmatrix}_{(m+n)\times p} \end{aligned} \tag{31}$$

In the correction step, the measurement residual is augmented into:

$$y_t - \hat{y}_t = \begin{bmatrix} y_{1,t} - \hat{y}_{1,t} \\ \vdots \\ y_{m,t} - \hat{y}_{m,t} \\ y_{1,s} - \hat{x}_{1,s}^- \\ \vdots \\ y_{n,s} - \hat{x}_{n,s}^- \end{bmatrix} \tag{32}$$

where $\hat{x}_s^- = f(\hat{x}_{s-1}) + NN_x(\hat{x}_{s-1}, \hat{\phi}_{x,s-1})$. The differences between the infrequent accurate measurement, y_s, and the model predicted state, $f(\hat{x}_{s-1})$, provide the more accurate references for NN_x, and hence SNN can be trained effectively using infrequent measurements in the DNEKF procedure.

B. Model Compensation Neural Network Structure and Output Computation

Compensation neural networks can be selected as single-hidden-layer, feed-forward neural networks. Sample neural network structures for SNN and MNN are illustrated in Fig. 3. Assign n_x and n_y neurons in the hidden layers of the two neural networks. The inputs to SNN are the posterior state estimates from the last time instant, and inputs to MNN are the modified prior state estimates at current time instant.

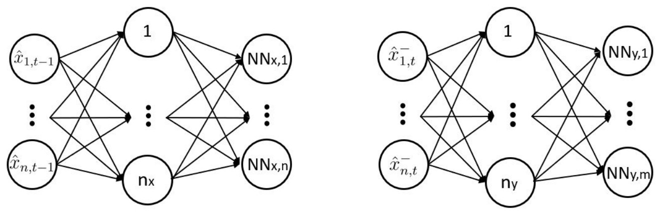

Fig. 3. Sample model compensation neural network structures.

Choosing $tanh$ as the activation function of each neuron, then SNN output value, NN_x, can be calculated as [22]:

$$NN_x = tanh\left(W_x^{(3)} \cdot tanh\left(W_x^{(2)}\hat{x}_{t-1} + b_x^{(2)}\right) + b_x^{(3)}\right) \quad (33)$$

where:

$$W_x^{(2)} = \begin{bmatrix} w_{1,1}^{(2)} & \cdots & w_{1,n}^{(2)} \\ \vdots & \ddots & \vdots \\ w_{n_x,1}^{(2)} & \cdots & w_{n_x,n}^{(2)} \end{bmatrix}, W_x^{(3)} = \begin{bmatrix} w_{1,1}^{(3)} & \cdots & w_{1,n_x}^{(3)} \\ \vdots & \ddots & \vdots \\ w_{n,1}^{(3)} & \cdots & w_{n,n_x}^{(3)} \end{bmatrix},$$

$$\hat{x}_{t-1} = \begin{bmatrix} \hat{x}_{1,t-1} \\ \vdots \\ \hat{x}_{n,t-1} \end{bmatrix}, b_x^{(2)} = \begin{bmatrix} b_1^{(2)} \\ \vdots \\ b_{n_x}^{(2)} \end{bmatrix}, b_x^{(3)} = \begin{bmatrix} b_1^{(3)} \\ \vdots \\ b_n^{(3)} \end{bmatrix}$$

The superscript of weights, w, represents the destination layer number. The first subscript represents the destination neuron number, while the second subscript indicates the departure neuron number [23]. As an example, $w_{1,2}^{(3)}$ means this weight is from the second neuron of the second layer towards the first neuron of the third layer. The superscript of biases, b, represents its location layer, and the subscript indicates the number of neuron where this bias locates.

Similarly, NN_y can be computed as:

$$NN_y = tanh\left(W_y^{(3)} \cdot tanh\left(W_y^{(2)}\hat{x}_t^- + b_y^{(2)}\right) + b_y^{(3)}\right) \quad (34)$$

where:

$$W_y^{(2)} = \begin{bmatrix} w_{1,1}^{(2)} & \cdots & w_{1,n}^{(2)} \\ \vdots & \ddots & \vdots \\ w_{n_y,1}^{(2)} & \cdots & w_{n_y,n}^{(2)} \end{bmatrix}, W_y^{(3)} = \begin{bmatrix} w_{1,1}^{(3)} & \cdots & w_{1,n_y}^{(3)} \\ \vdots & \ddots & \vdots \\ w_{m,1}^{(3)} & \cdots & w_{m,n_y}^{(3)} \end{bmatrix},$$

$$\hat{x}_t^- = \begin{bmatrix} \hat{x}_{1,t}^- \\ \vdots \\ \hat{x}_{n,t}^- \end{bmatrix}, b_y^{(2)} = \begin{bmatrix} b_1^{(2)} \\ \vdots \\ b_{n_y}^{(2)} \end{bmatrix}, b_y^{(3)} = \begin{bmatrix} b_1^{(3)} \\ \vdots \\ b_m^{(3)} \end{bmatrix}$$

V. Case Studies

In this section, the proposed DNEKF approach is evaluated through two numerical examples and one industrial application. In the first numerical example, both state and output models are inaccurate and are compensated using the DNEKF approach. In the second numerical example, colored process and measurement noises are adopted to check compensation capacity of the proposed method. In the industrial application, linear time-invariant models are utilized to describe the process, and the DNEKF approach is applied for online model inaccuracy compensation.

A. Numerical Examples

1) *Model Inaccuracies Compensation*: The key objective of the DNEKF approach is to compensate for model inaccuracies. In this example, the true models are unknown, and inaccurate models are utilized for predictions to test model inaccuracies compensation ability of the DNEKF. The true state and output models are given as follows:

$$\begin{aligned} x_t &= f_{true}(x_{t-1}) + w_{t-1} \\ &= 3sin^2(x_{t-1}) - 5cos(x_{t-1}) + w_{t-1} \end{aligned} \quad (35)$$

and

$$y_t = \begin{bmatrix} h_{1,true}(x_t) + v_{1,t} \\ h_{2,true}(x_t) + v_{2,t} \end{bmatrix} = \begin{bmatrix} x_t^2 + v_{1,t} \\ 3cos(x_t) + v_{2,t} \end{bmatrix} \quad (36)$$

where $w_t \sim N\left(0, 4 \times 10^{-4}\right)$, and $v_t \sim N\left(0, R\right)$ with

$$R = \begin{cases} \text{diag}\left\{1.6 \times 10^{-3}, 1.6 \times 10^{-3}, 1 \times 10^{-4}\right\}, & \text{major instants} \\ \text{diag}\left\{1.6 \times 10^{-3}, 1.6 \times 10^{-3}\right\}, & \text{minor instants} \end{cases}$$

The inaccurate state and output models are:

$$\begin{aligned} \hat{x}_t^- &= f(\hat{x}_{t-1}) \\ &= 2sin^2(\hat{x}_{t-1}) - 3cos(\hat{x}_{t-1}) \end{aligned} \quad (37)$$

and

$$\hat{y}_t = \begin{bmatrix} h_1(\hat{x}_t^-) \\ h_2(\hat{x}_t^-) \end{bmatrix} = \begin{bmatrix} 2\hat{x}_t^- \\ 5cos(\hat{x}_t^-) \end{bmatrix} \quad (38)$$

Applying DNEKF algorithm, the state predictions are:

$$\hat{\bar{x}}_t^- = \begin{bmatrix} f(\hat{x}_{t-1}) + NN_x(\hat{x}_{t-1}, \hat{\phi}_{x,t-1}) \\ \hat{\phi}_{x,t-1} \\ \hat{\phi}_{y,t-1} \end{bmatrix} \quad (39)$$

The output predictions are:

$$\hat{y}_t = \begin{bmatrix} h_1(\hat{x}_t^-) + NN_{y,1}(\hat{x}_t^-, \hat{\phi}_{y,t}^-) \\ h_2(\hat{x}_t^-) + NN_{y,2}(\hat{x}_t^-, \hat{\phi}_{y,t}^-) \end{bmatrix} \quad (40)$$

In this case study, $tanh$ is used as the activation function. For this single-dimension quality variable system with two frequent measurements, 1-3-1 and 1-3-2 are selected as neural network structures. The numbers of elements of ϕ_x and ϕ_y are 10 and 14, respectively. Then total number of elements of augmented state vector, $\bar{x}$, is 25. At *major instants*, the Jacobian matrices, $\bar{F}_{t-1}$ and $\bar{H}_t$, are:

$$\bar{F}_{t-1} = \begin{bmatrix} \frac{\partial f(\hat{x}_{t-1})}{\partial x}\Big|_{\hat{x}_{t-1}} + \frac{\partial NN_x}{\partial x}\Big|_{\hat{x}_{t-1}} & \frac{\partial NN_x}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t-1}} & \mathbf{0} \\ \mathbf{0} & I_{\phi_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{\phi_y} \end{bmatrix}_{25\times 25} \quad (41)$$

where:

$$\frac{\partial f(\hat{x}_{t-1})}{\partial x}\Big|_{\hat{x}_{t-1}} = 4sin(\hat{x}_{t-1})cos(\hat{x}_{t-1}) + 3sin(\hat{x}_{t-1}) \tag{42}$$

and

$$\bar{H}_t = \begin{bmatrix} \frac{\partial h_1(\hat{x}_t^-)}{\partial x}\Big|_{\hat{x}_t^-} + \frac{\partial NN_{y,1}}{\partial x}\Big|_{\hat{x}_t^-} & \frac{\partial NN_{y,1}}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t}^-} & \frac{\partial NN_{y,1}}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t}^-} \\ \frac{\partial h_2(\hat{x}_t^-)}{\partial x}\Big|_{\hat{x}_t^-} + \frac{\partial NN_{y,2}}{\partial x}\Big|_{\hat{x}_t^-} & \frac{\partial NN_{y,2}}{\partial \phi_x}\Big|_{\hat{\phi}_{x,t}^-} & \frac{\partial NN_{y,2}}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t}^-} \\ 1 & \mathbf{0} & \mathbf{0} \end{bmatrix}_{3\times 25} \tag{43}$$

where:

$$\frac{\partial h_1(\hat{x}_t^-)}{\partial x}\Big|_{\hat{x}_t^-} = 2 \tag{44}$$

$$\frac{\partial h_2(\hat{x}_t^-)}{\partial x}\Big|_{\hat{x}_t^-} = -5sin(\hat{x}_t^-) \tag{45}$$

At *minor instants*, only frequent measurements are available, and the corresponding Jacobian matrices are:

$$\bar{F}_{t-1} = \begin{bmatrix} \frac{\partial f(\hat{x}_{t-1})}{\partial x}\Big|_{\hat{x}_{t-1}} + \frac{\partial NN_x}{\partial x}\Big|_{\hat{x}_{t-1}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{\phi_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{\phi_y} \end{bmatrix}_{25\times 25} \tag{46}$$

and

$$\bar{H}_t = \begin{bmatrix} \frac{\partial h_1(\hat{x}_t^-)}{\partial x}\Big|_{\hat{x}_t^-} + \frac{\partial NN_{y,1}}{\partial x}\Big|_{\hat{x}_t^-} & \mathbf{0} & \frac{\partial NN_{y,1}}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t}^-} \\ \frac{\partial h_2(\hat{x}_t^-)}{\partial x}\Big|_{\hat{x}_t^-} + \frac{\partial NN_{y,2}}{\partial x}\Big|_{\hat{x}_t^-} & \mathbf{0} & \frac{\partial NN_{y,2}}{\partial \phi_y}\Big|_{\hat{\phi}_{y,t}^-} \end{bmatrix}_{2\times 25} \tag{47}$$

To compare performance of the DNEKF fusion and the standard EKF fusion, the Monte Carlo performance evaluations are conducted using 10000 simulation runs over 200 frequent sampling intervals, assuming infrequent accurate measurements are available at every 40 frequent sampling intervals. Resulted averaged root squared errors of 10000 simulation runs for both methods are plotted in Fig. 4. At transient time, inaccurate models and randomly assigned neural network parameters lead to larger estimation errors for both the standard EKF algorithm and the proposed DNEKF algorithm. With increase of time, estimation errors from both methods decrease, and the proposed approach demonstrates better performance, in terms of averaged root squared estimation error, with compensation of model inaccuracies.

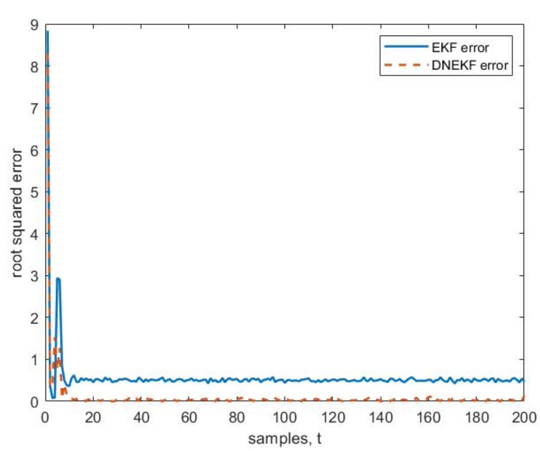

Fig. 4. Averaged root squared estimation errors from the standard EKF fusion and the DNEKF fusion in presence of model inaccuracies.

2) *Colored Noises Compensation*: In this example, the actual models are available and colored noise are applied to both process state and measurements, but white noises are assumed in the nominal models. Red noise, which has zero mean, constant variance, and is correlated along time is generated for simulation using the following equations [24]:

$$\begin{aligned} p_1 &= w_1 \\ p_t &= rp_{t-1} + (1-r^2)^{1/2} w_t \end{aligned} \tag{48}$$

where w_t is Gaussian white noise, r is the correlation between red noise, p_{t+1} and p_t, and $r = 0.6$ in this simulation. The state and output predictions are:

$$\hat{\bar{x}}_t^- = \begin{bmatrix} f_{true}(\hat{x}_{t-1}) + NN_x(\hat{x}_{t-1}, \hat{\phi}_{x,t-1}) \\ \hat{\phi}_{x,t-1} \\ \hat{\phi}_{y,t-1} \end{bmatrix} \tag{49}$$

and

$$\hat{y}_t = \begin{bmatrix} h_{1,true}(\hat{x}_t^-) + NN_{y,1}(\hat{x}_t^-, \hat{\phi}_{y,t}^-) \\ h_{2,true}(\hat{x}_t^-) + NN_{y,2}(\hat{x}_t^-, \hat{\phi}_{y,t}^-) \end{bmatrix} \tag{50}$$

In this example, infrequent accurate measurements are available at every 50 frequent sampling intervals, and 10000 Monte Carlo experiments are performed to obtain the averaged root squared estimation errors, with results illustrated in Fig. 5. This figure shows that the DNEKF approach yields lower estimation error than the standard EKF approach, and proves that even if accurate models are available, violations of noise assumption will also decrease the EKF fusion performance. However, the DNEKF approach is able to compensate for violations of noise assumption and provide more accurate state estimation.

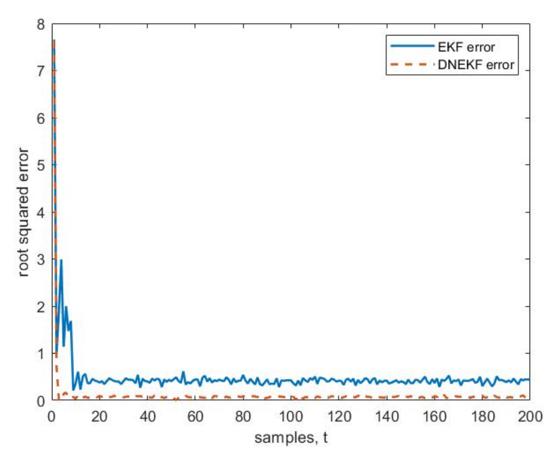

Fig. 5. Averaged root squared estimation errors from the standard EKF fusion and the DNEKF fusion in presence of violations of noise assumption.

B. Industrial Application: PSV Interface Level Estimation

The PSV is used to separate bitumen from oil sands slurry through a water-based gravity separation process [25]. The typical cross-section view of PSV is shown in Fig. 6. The oil sands slurry in PSV is separated into three layers: the froth layer, middlings layer, and tailings layer. The exact location of froth-middlings interface is critical to bitumen recovery rate control [26]. Common measurement instruments of interface level are nuclear density profiler, differential pressure (DP) cell, and an integrated sensor, with same sampling rate as others.

The profiler uses nuclear technique to detect distinct density changes in the froth and middlings layers. This measurement assumes that middlings densities are measured in a continuous block. In this case, if one of the upper density segments reads a density with error, the error will be accumulated, and lead to biases in interface level estimation. DP cell detects the interface level through pressure measurements in these two layers. The integrated sensor combines fixed point density measurements with DP cell pressure measurements to provide interface level estimation. Both DP cell and integrated sensor measurements assume fixed densities at top and bottom of the two layers, which may not be consistent with real situations, and will lead to estimation errors. In this analysis, these three measurements are considered as frequent but less accurate measurements at the same sampling rate.

The image processing based computer vision system, which relies on a camera mounted on the PSV sight glasses to provide vision frames to estimate the interface level, is considered to be the most accurate [27], [28]. However, it suffers from problems of occasional low-quality sight glasses visions. When sight glasses visions are unclear or the interface crosses transition regions between sight glasses, the computer vision system is not able to provide interface level estimation and will freeze itself by holding on to the last reliable estimate. As a result, the computer vision results are considered as more accurate but infrequent measurements in the PSV system.

Overall, the current measurements used to estimate PSV froth-middlings interface level are all subject to inaccuracies due to their intrinsic problems. Fusion of these measurements, however, takes advantage of all these measurements to increase the robustness and accuracy of froth-middlings interface level estimation. In this case study, 500 frequent sampling intervals, during which the sight glasses visions are clear and computer vision results are of high accuracy, are used to test fusion performance of different approaches. Over this period, high-accuracy computer vision results are assumed to be available at slower sampling rates. All the analyzed data are collected from real site operation through open protocol communication (OPC), and the industrial data have been normalized for proprietary reason.

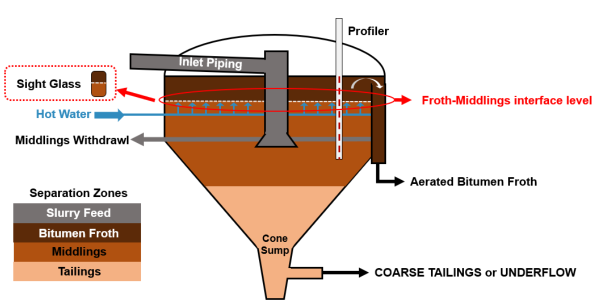

Fig. 6. Sample cross-section view of PSV.

In the DNEKF application, 1-3-1 and 1-4-3 structures are utilized for SNN and MNN, respectively. Linear time-invariant state and output models, F and H, are used to describe the process. The number of elements of ϕ_x is 10 and that of ϕ_y is 23. After adding one state, total elements of $\bar{x}$ is 34. The initial values of ϕ_x and ϕ_y are random numbers that are close to zero. The output model is:

$$H = \begin{cases} \begin{bmatrix} H_{DP} & H_{IS} & H_P & H_C \end{bmatrix}^T, & \text{major instants} \\ \begin{bmatrix} H_{DP} & H_{IS} & H_P \end{bmatrix}^T, & \text{minor instants} \end{cases}$$

where DP, IS, P, and C represent DP cell, integrated sensor, profiler, and computer vision system, respectively, and $H_C = 1$. The corresponding measurement noise covariance matrix is:

$$R = \begin{cases} \text{diag}\left\{R_{DP}, R_{IS}, R_P, R_C\right\}, & \text{major instants} \\ \text{diag}\left\{R_{DP}, R_{IS}, R_P\right\}, & \text{minor instants} \end{cases}$$

For comparison, mean squared error (MSE) values from individual measurements and different fusion approaches are calculated using high-accuracy computer vision results as references. The MSE is computed as:

$$MSE = \frac{1}{n} \sum_{t=1}^{n} (x_t - \hat{x}_t)^2 \tag{51}$$

where n is number of frequent sampling intervals. Performance of individual measurements and different fusion approaches in terms of MSE values are presented in Table I. All the analyzed fusion methods yield lower estimation errors than individual measurements. For the SNEKF, as aforementioned, without a reliable reference, its neural network training is not reliable, hence outputs from the NN_x are not able to improve the fusion performance. The MSE from this fusion method is higher than that from the EKF fusion. The neural network training process in the MNEKF is more reliable, and its fusion accuracy is improved from the EKF method. In terms of the DNEKF approach, with reliable training process for both neural networks, its performance outperforms that of the other three methods, with respect to MSE values. In addition, with increased availability of infrequent reference measurements, the DNKEF fusion performance is further improved.

TABLE I
PERFORMANCE COMPARISON FOR INDIVIDUAL MEASUREMENTS AND DIFFERENT FUSION METHODS WITH RESPECT TO MSE VALUES.

Individual measurements	Infrequent sampling intervals	MSE
Profiler DP cell Integrated sensor	N/A	2.0007 1.9018 1.0862
Fusion methods		
EKF SNEKF MNEKF	N/A	0.48129 0.51000 0.46327
DNEKF	250 125 50	0.42302 0.37454 0.34494

Fig. 8 shows results for the DNKEF fusion with infrequent measurements available at every 20 frequent sampling intervals, as well as the weights and biases for SNN and MNN. As can be seen from the middle subplot, the weights and biases for SNN change every 20 frequent sampling intervals, consistent with accurate computer vision results availability. As shown in the bottom subplot, the weights and biases for MNN change at a frequent sampling rate.

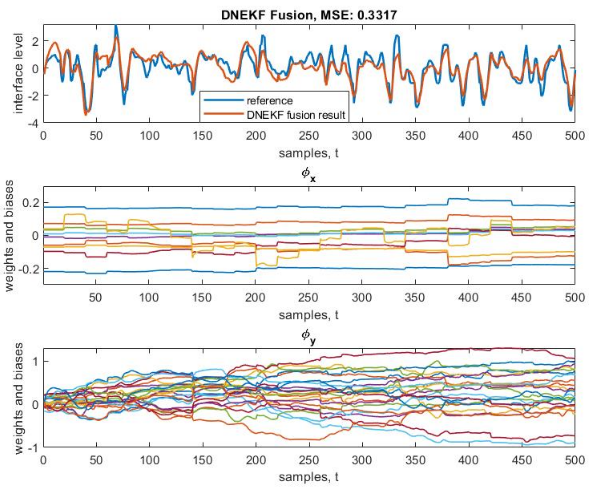

Fig. 7. The DNEKF fusion with infrequent measurements available at every 20 frequent sampling intervals, and the weights and biases for SNN and MNN.

Additionally, the proposed DNEKF approach not only is a standalone algorithm but also can be integrated with the computer vision system to improve the existing measurement technique. As shown in Fig. 8, when the interface level increases from the bottom to the middle sight glass, the computer vision result is stuck at the lower boundary of the middle sight glass since the camera cannot capture a clear interface level movement, and warnings are issued. The proposed sensor fusion approach then shifts from *major instants* mode into *minor instants* mode, and provides dynamic interface level estimation continuously, resulting in the sensor fusion and computer vision integrated system.

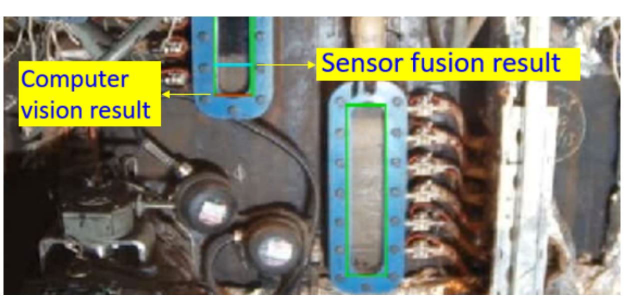

Fig. 8. Illustration of sensor fusion and computer vision integrated system when interface level crosses transition region between sight glasses.

VI. Conclusions

In this paper, the DNEKF approach for multirate sensor data fusion is proposed. The DNEKF estimates process states and neural network parameters simultaneously through state vector augmentation. While compensating for inadequate process knowledge, the DNEKF takes advantage of both frequent and infrequent measurements to update the weights and biases for MNN and SNN through a multirate strategy to improve the fusion robustness and accuracy.

The effectiveness of the proposed approach is demonstrated through two numerical examples and one industrial application example. The simulation and industrial application results demonstrate that the proposed DNEKF approach is robust to model inaccuracies and violations of noise assumption. In addition, the DNEKF not only is a standalone algorithm but also is integrated with the existing measurement technique to improve measurement performance. Besides the PSV system, the proposed approach can be applied to more general processes, in which infrequent but more accurate measurements, such as lab analysis, are available.

References

[1] S. L. Sun and Z. L. Deng, “Multi-sensor optimal information fusion Kalman filter,” *Automatica*, vol. 40, no. 6, pp. 1017–1023, 2004, doi: 10.1016/j.automatica.2004.01.014.

[2] C. B. Chang and K. P. Dunn, “Kalman filter compensation for a special class of systems,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-13, no. 6, pp. 700–706, 1977, doi: 10.1109/TAES.1977.308511.

[3] J. Sasiadek and P. Hartana, “Sensor data fusion using kalman filter,” in *Proc. Int. Conf. Inf. Fusion*, vol. 2, Aug. 2000, pp. WED5/19 – WED5/25, doi: 10.1109/IFIC.2000.859866.

[4] M. Mombeini and H. Khaloozadeh, “State and parameter estimation of the lorenz system in existence of colored noise,” Available at https://arxiv.org/abs/1211.4228v1 (2020/03/10).

[5] M. A. Akram, P. Liu, M. O. Tahir, W. Ali, and Y. Wang, “A state optimization model based on Kalman filtering and robust estimation theory for fusion of multi-source information in highly non-linear systems,” *Sensors*, vol. 19, no. 7, p. 1687, 2019, doi: 10.3390/s19071687.

[6] T. H. Xu, Y. P. Chen, and Q. X. Wang, “Kalman filter-based orbit determination with dynamic model compensation for a maneuvered geo satellite,” *J. Adv. Mater. Res.*, vol. 383, pp. 5626–5631, 2012, doi: 10.4028/www.scientific.net/AMR.383-390.5626.

[7] S. C. Stubberud, R. N. Lobbia, and M. Owen, “An adaptive extended Kalman filter using artificial neural networks,” in *Proc. IEEE Int. Conf. Decis. Control*, vol. 2, 1995, pp. 1852–1856, doi: 10.1109/CDC.1995.480611.

[8] M. Choi, R. Sakthivel, and W. K. Chung, “Neural network-aided extended Kalman filter for SLAM problem,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1686–1690, doi: 10.1109/ROBOT.2007.363565.

[9] M. W. Owen and A. R. Stubberud, “A neural extended Kalman filter multiple model tracker,” in *Oceans*, vol. 4, 2003, pp. 2111–2119 Vol.4, doi: 10.1109/OCEANS.2003.178229.

[10] S. C. Stubberud and K. A. Kramer, “A 2-d intercept problem using the neural extended Kalman filter for tracking and linear predictions,” in *Proc. IEEE Int. SE. Symp. Syst. (SSST)*, 2005, pp. 367–372, doi: 10.1109/SSST.2005.1460938.

[11] K. A. Kramer, S. C. Stubberud, and J. A. Geremia, “Sensor calibration using the neural extended Kalman filter in a control loop,” in *Proc. IEEE Int. Conf. Comput. Intell. Meas. Syst. Appl.*, 2007, pp. 19–24, doi: 10.1109/CIMSA.2007.43625314.

[12] S. C. Stubberud, K. A. Kramer, and J. A. Geremia, “Online sensor modeling using a neural Kalman filter,” *IEEE Trans. Instrum. Meas.*, vol. 56, no. 4, pp. 1451–1458, 2007, doi: 10.1109/TIM.2007.900125.

[13] L. Xie, Y. Zhu, B. Huang, and Y. Zheng, “Kalman filtering approach to multirate information fusion for soft sensor development,” in *Int. Conf. Inf. Fusion*, 2012, pp. 641–648.

[14] A. Fatehi and B. Huang, “State estimation and fusion in the presence of integrated measurement,” *IEEE Trans. Instrum. Meas.*, vol. 66, no. 9, pp. 2490–2499, 2017, doi: 10.1109/TIM.2017.2701143.

[15] M. K. Kalandros, L. Trailovic, L. Y. Pao, and Y. Bar-Shalom, “Tutorial on multisensor management and fusion algorithms for target tracking,” in *Proc. Am. Control Conf. (ACC)*, vol. 5, 2004, pp. 4734–4748 vol.5, doi: 10.23919/ACC.2004.1384060.

[16] S. Canan, R. Akkaya, and S. Ergintav, “Extended Kalman filter sensor fusion and application to mobile robot,” in *Proc. IEEE Signal Process. Commun. Appl. Conf.*, 2004, pp. 771–774, doi: 10.1109/SIU.2004.1338645.

[17] D. Simon, *Optimal state estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sonss, 2006, doi: 10.1002/0470045345.

[18] N. Shivashankarappa, S. Adiga, R. A. Avinash, and H. R. Janardhan, “Kalman filter based multiple sensor data fusion in systems with time delayed state,” in *Int. Conf. Signal Process. Integr. Netw. (SPIN)*, 2016, pp. 375–382, doi: 10.1109/SPIN.2016.7566723.

[19] A. Gopalakrishnan, N. Kaisare, and S. Narasimhan, “Incorporating delayed and infrequent measurements in extended Kalman filter based nonlinear state estimation,” *J. Process Contr.*, vol. 21, pp. 119–129, 01 2011, doi: 10.1016/j.jprocont.2010.10.013.

[20] Y. Guo, Y. Zhao, and B. Huang, “Development of soft sensor by incorporating the delayed infrequent and irregular measurements,” *J. Process Contr.*, vol. 24, no. 11, pp. 1733–1739, 2014, doi: 10.1016/j.jprocont.2014.09.006.

[21] B. M. Wilamowski, N. J. Cotton, O. Kaynak, and G. Dundar, “Computing gradient vector and jacobian matrix in arbitrarily connected neural networks,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 10, pp. 3784–3790, 2008, doi: 10.1109/TIE.2008.2003319.

[22] I. Aizenberg and C. Moraga, “Multi-layered neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm,” in *Proc. IEEE Int. Conf. Neural Netw.*, 07 2004, doi: 10.17877/DE290R-8012.

[23] M. Nielsen, “Neural networks and deep learning,” Available at http://neuralnetworksanddeeplearning.com/ (2019/08/22).

[24] J. Tetazoo, “Atm s 552 notes,” Available at http://atmos.washington.edu/~dennis/552_Notes_ftp.html (2005/06/12).

[25] Y. Tu, J. O’Carroll, L. Kotlyar, B. Sparks, S. Ng, K. Chung, and G. Cuddy, “Recovery of bitumen from oilsands: gelation of ultra-fine clay in the primary separation vessel,” *Fuel*, vol. 84, no. 6, pp. 653–660, 2005, doi: 10.1016/j.fuel.2004.03.020.

[26] B. Li, F. Xu, Z. Ren, and A. Espejo, “Extended abstract: Primary separation vessel interface control,” in *Int. Symp. Adv. Control of Ind. Process. (ADCONIP)*, 2011, pp. 262–264.

[27] P. Jampana, S. L. Shah, and R. Kadali, “Computer vision based interface level control in separation cells,” *Control Eng. Pract.*, vol. 18, no. 4, pp. 349–357, 2010, doi: 10.1016/j.conengprac.2009.12.004.

[28] A. Vicente, R. Raveendran, B. Huang, S. Sedghi, A. Narang, H. Jiang, and W. Mitchell, “Computer vision system for froth-middlings interface level detection in the primary separation vessels,” *Comput. Chem. Eng*, vol. 123, pp. 357–370, 04 2019, doi: 10.1016/j.compchemeng.2019.01.017.