

## Research Article

# Control Loop Sensor Calibration Using Neural Networks for Robotic Control

**Kathleen A. Kramer<sup>1</sup> and Stephen C. Stubberud<sup>2</sup>**

<sup>1</sup>Department of Engineering, University of San Diego, 5998 Alcalá Park, San Diego, CA 92110, USA

<sup>2</sup>Advanced Programs, Oakridge Technology, Del Mar, CA 92014, USA

Correspondence should be addressed to Kathleen A. Kramer, [kramer@sandiego.edu](mailto:kramer@sandiego.edu)

Received 15 July 2011; Accepted 8 November 2011

Academic Editor: Ivo Bukovsky

Copyright © 2011 K. A. Kramer and S. C. Stubberud. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Whether sensor model's inaccuracies are a result of poor initial modeling or from sensor damage or drift, the effects can be just as detrimental. Sensor modeling errors result in poor state estimation. This, in turn, can cause a control system relying upon the sensor's measurements to become unstable, such as in robotics where the control system is applied to allow autonomous navigation. A technique referred to as a neural extended Kalman filter (NEKF) is developed to provide both state estimation in a control loop and to learn the difference between the true sensor dynamics and the sensor model. The technique requires multiple sensors on the control system so that the properly operating and modeled sensors can be used as truth. The NEKF trains a neural network on-line using the same residuals as the state estimation. The resulting sensor model can then be reincorporated fully into the system to provide the added estimation capability and redundancy.

## 1. Introduction

In the target tracking applications, sensor systems are placed in various locations about the region of interest and provide measurements that are combined over time to provide an improved picture of the region. One of the significant problems with target tracking systems is that the sensors might not be properly calibrated, aligned, or modeled. This can wreak havoc with the performance of the involved tracking systems. When calibration and alignment are the main factors, the problem is referred to as sensor registration [1, 2]. Since the real-time activities of a tracking problem cannot be postponed, the correction of the registration or the mismodeling must be performed online. This problem can extend to other applications, including that of feedback-control systems.

While some control applications use a single sensor to provide measurements that are used in the feedback loop, many control systems use several sensors to provide the necessary measurements. Multiple sensors are used for a variety of reasons. Often times the sensors add redundancy, such as

with navigation systems where multiple inertial navigation systems (INSs) are employed or for safety considerations to provide back-up in case of failure [3]. Sometimes multiple sensors are employed because more accurate sensors cannot provide measurements at the necessary update rates. In such cases, less accurate sensors might be used to provide reports at the sampling time between the measurement reports of the more accurate system. Finally, additional sensors are added in that they provide measurements that contain directly observable state information. In mobile robotic systems, for example, sensors include position sensors to determine absolute position, such as a global position system (GPS), or relative position such as radars, sonars, and imagery equipment, as well as speedometers and possible accelerometers that provide the velocity states more directly than the indirectly observed values from position.

While additional sensors are often used to provide improved overall accuracy, errors can arise in the accuracy of the measurements provided to the control law by individual sensors. This occurs with tracking sensors and is well known with gyroscopes in an INS. While a sensor that drifts from its

calibration point in a nonrandom and identifiable manner can be recalibrated, there is no remedy other than redundancy or replacing a sensor that fails or breaks completely since it is unable to provide a meaningful measurement. For a sensor that can be recalibrated, online calibration is desired because taking the inaccurate sensor offline may not be possible in the short term or could have a deleterious effect on the performance of the controller. When a closed-loop control system relies upon a sensor with an error, the result is that the control signal varies from the value needed to achieve the desired response. These effects are evident, even in a linear system [4]. Sensor calibration is thus an important issue for the control design [5]. For nonlinear systems, these issues become even more pronounced and can lead to a more significant effect on stability, as the feedback may not fall within the Lyapunov stability criteria [6].

In this work, an online calibration technique is proposed for the case where the control law utilizes a state estimator [7]. The approach uses a technique referred to as a neural extended Kalman filter (NEKF) [8–10]. The NEKF has the benefit of being able to train a function approximation online and provide the state estimation simultaneously. The NEKF provides a unified approach using the same residual for both components. Techniques such as in [11], in contrast, can provide the desired sensor correction for sensors in a control loop but are not integrated to the state estimator and corrections are performed outside of the control loop. The technique applied to this sensor correction problem is based upon an approach that was developed for a multisensor tracking system where the sensor model was in error or a sensor registration was detected [12]. In that development, one sensor was considered to be local while the other was considered to be off-board. All corrections were made to the off-board reports relative to the local reference frame. Unlike the tracking problem where the estimator is open loop, control applications require the consideration of closed-loop issues including stability. The NEKF is used to recalibrate the sensor model based on online operational measurements while in the control loop. It also provides the state estimation for the control law. Thus, the proposed technique estimates the states and provides the training paradigm in a single algorithmic implementation [13, 14], unlike other neural network sensor modeling techniques [15–17]. The software correction to the actual sensor measurements can be applied to the case where the sensor is still operational but poorly calibrated.

The NEKF algorithm requires a truth measurement from which to calibrate the sensor. Using additional sensors on the dynamic system, a poorly calibrated sensor can be modeled such that its reporting errors are removed prior to their incorporation to the feedback loop.

In Section 2, the NEKF algorithm and its implementation in the control loop is developed. Section 3 provides the first example system and the performance results of the online calibration using the NEKF for a two-sensor moving platform that has a range-bearing sensor and a miscalibrated velocity sensor. A navigation problem where an intermittent use of a position sensor such as GPS is available is presented

in Section 4. These examples show the benefits of the NEKF calibration.

## 2. Recalibration Approach

In this effort, a recalibration technique that can be used online is developed. The approach relies upon other sensors in the multisensor system operating properly which are used to provide a level of truth from which to calibrate. The sensor to be recalibrated may have some inaccuracy but cannot be irreparably damaged.

The dynamics of a nonlinear system can be modeled as a set of recursive difference equations:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{r}_k) + \boldsymbol{\nu}_k, \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\eta}_k, \end{aligned} \quad (1)$$

where  $\mathbf{x}_k$  is the state vector representing the system dynamics,  $\mathbf{f}(\cdot)$  is state-coupling model,  $\mathbf{r}_k$  is the reference input vector,  $\mathbf{z}_k$  is measurement vector, also considered the report from the sensor system,  $\mathbf{u}_k$  is an external input to the sensor system often simply the reference signal, and  $\mathbf{h}(\cdot)$  is the output-coupling function [18]. The vectors  $\boldsymbol{\nu}$  and  $\boldsymbol{\eta}$  represent noise on the system states and the measurement states, respectively.

If the reference input  $\mathbf{r}$  is assumed to be both the reference signal to the system and the external input to the sensor, then  $\mathbf{r}$  is considered to be the same as  $\mathbf{u}$ . Then, the state estimation model used in the control law would be rewritten as

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k), \\ \hat{\mathbf{z}}_k &= \hat{\mathbf{h}}(\mathbf{x}_k, \mathbf{u}_k). \end{aligned} \quad (2)$$

The accuracy of the models in the estimator determines the accuracies of state estimates. For this effort, it is assumed that the state-coupling function is very accurate while at least one of the sensor models has some inaccuracy. The error in this model is defined as

$$\boldsymbol{\varepsilon} = \mathbf{h}_{\text{true}}(\mathbf{x}, \mathbf{u}) - \mathbf{h}_{\text{model}}(\mathbf{x}, \mathbf{u}). \quad (3)$$

To reduce the measurement error, the function  $\boldsymbol{\varepsilon}$  needs to be identified. A neural network that satisfies the function approximation requirements of the Stone-Weierstrauss Theorem [9] is proposed to approximate the error function  $\boldsymbol{\varepsilon}$ . Such a neural network can be defined as

$$NN(\mathbf{x}, \mathbf{w}) = NN(\mathbf{x}, \boldsymbol{\omega}, \boldsymbol{\beta}) = \sum_j \beta_j g_j(\mathbf{x}, \boldsymbol{\omega}_j), \quad (4)$$

where  $\mathbf{w}$  are the weights of the neural network, which are decomposed into the set of input weights  $\boldsymbol{\omega}$  and the set of output weights  $\boldsymbol{\beta}$ . The hidden function  $g(\cdot)$  is a sigmoid function:

$$g(\mathbf{x}, \boldsymbol{\omega}) = \frac{1 - e^{-\boldsymbol{\omega}^T \mathbf{x}}}{1 + e^{-\boldsymbol{\omega}^T \mathbf{x}}} = \frac{1 - e^{-\sum_k \omega_j x_k}}{1 + e^{-\sum_k \omega_j x_k}}. \quad (5)$$

This creates a new measurement and more accurate measurement model:

$$e = \mathbf{h}_{\text{true}}(\mathbf{x}, \mathbf{u}) - \mathbf{h}_{\text{model}}(\mathbf{x}, \mathbf{u}) - NN(\mathbf{x}, \mathbf{u}, \mathbf{w}), \quad (6)$$

where  $\|e\| \ll \|\boldsymbol{\varepsilon}\|$ .

The extended Kalman filter (EKF) is a standard for nonlinear estimation in a control loop [4, 19]. An approach developed in [7] referred to as a neural extended Kalman filter provides both the state estimation of the system and the training of the weights [20] of a neural network using the same measurement estimates.

The NEKF is a coupled state-estimation and neural-network training algorithm and has similarities to parameter estimation approaches [21]. In such implementations, the state vector of the EKF contains both the system states and the parameters of the model being estimated. Parameter estimation, though, is based on a known model structure. The neural network of (4), in contrast, is a general function with its weights as the parameters. Therefore, the augmented state of this NEKF is the state estimate and the input and output weights of the neural network:

$$\bar{\mathbf{x}}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{w}_k \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\omega}_k \\ \boldsymbol{\beta}_k \end{bmatrix}. \quad (7)$$

Incorporating this new state and the neural network affects all of the Kalman filter equations. In the Kalman gain equation and the state error covariance update equation, (8) and (10), the Jacobian of the output-coupling function is properly augmented. In (9), the neural network is incorporated into the state update. Finally, the prediction equations, (11) and (12), must be properly augmented for the augmented state. Thus, the NEKF equations for the sensor modeling become

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \bar{\mathbf{H}}^T (\bar{\mathbf{H}} \mathbf{P}_{k|k-1} \bar{\mathbf{H}}^T + \mathbf{R})^{-1}, \quad (8)$$

$$\begin{aligned} \bar{\mathbf{x}}_{k|k} &= \begin{bmatrix} \mathbf{x}_{k|k} \\ \mathbf{w}_{k|k} \end{bmatrix} = \bar{\mathbf{x}}_{k|k-1} \\ &+ \mathbf{K}_k (\mathbf{z}_k - (\mathbf{h}(\mathbf{x}_{k|k-1}, \mathbf{u}_k) + NN(\mathbf{x}_{k|k-1}, \mathbf{u}_k, \mathbf{w}_{k|k-1}))), \end{aligned} \quad (9)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \bar{\mathbf{H}} \mathbf{K}_k) \mathbf{P}_{k|k-1}, \quad (10)$$

$$\bar{\mathbf{x}}_{k+1|k} = \bar{\mathbf{f}}(\bar{\mathbf{x}}_{k|k}) = \begin{bmatrix} \mathbf{f}(\mathbf{x}_{k|k}) \\ \mathbf{w}_{k|k} \end{bmatrix}, \quad (11)$$

$$\begin{aligned} \mathbf{P}_{k+1|k} &= \frac{\partial \bar{\mathbf{f}}(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}} \mathbf{P}_{k|k} \frac{\partial \bar{\mathbf{f}}(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}}^T + \mathbf{Q}_k \\ &= \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{P}_{k|k} \begin{bmatrix} \mathbf{F}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} + \mathbf{Q}_k. \end{aligned} \quad (12)$$

The augmented Jacobian of the sensor function model,

$$\bar{\mathbf{h}}(\mathbf{x}, \mathbf{u}, \mathbf{w}) = \mathbf{h}(\mathbf{x}, \mathbf{u}) + NN(\mathbf{x}, \mathbf{u}, \mathbf{w}) \quad (13)$$

is therefore defined as

$$\begin{aligned} \bar{\mathbf{H}} &= \left[ \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} + NN(\mathbf{x}, \mathbf{u}, \mathbf{w}) \frac{\partial NN(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{w}} \right]_{\mathbf{x}=\mathbf{x}_{k|k-1}, \mathbf{w}_{k|k-1}} \\ &= \left[ \mathbf{H} + \frac{\partial NN(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{x}} \frac{\partial NN(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{w}} \right]_{\mathbf{x}=\mathbf{x}_{k|k-1}, \mathbf{w}_{k|k-1}}. \end{aligned} \quad (14)$$

The necessary coupling between the weights and the dynamic states occurs as a result of the output-coupling Jacobian. This coupling permits the weights to be completely observable to the measurements and to train off of the same residual as the system states. This also implies that the NEKF trains the neural network online without the need for additional off-line training sets.

The NEKF defined in (8)–(12) requires a truth measurement for the neural network to learn. For a target tracking application where a surveyed target can be used to provide ground truth, this implementation is appropriate [9]. Control applications are different in that such “truth” is unavailable. Instead, the state of the system dynamics must be generated using sensors in the control system that are more accurately modeled. This requires a modification of the NEKF design to utilize the existing state estimates and the measurements from the other sensors to correct the output-coupling function of the mismodeled sensor using the neural network.

For this implementation, the detection of a sensor failure is assumed to have occurred. Such detection techniques abound in tracking theory and similar techniques exist in dynamic system implementation [22]. The detection of the faulty sensor initiates two changes in the NEKF. First, the process noise matrix is modified. The ratio of the process noise  $\mathbf{Q}$  to the measurement noise  $\mathbf{R}$  is reduced to significantly favor the dynamic state estimate over the residual. This reduces the effect of the measurements from the poor sensor on the system estimate. The process noise of the weights is increased to favor the residual over the weight estimates so that the weights will train to learn the error function. A similar approach of reducing the ratio of  $\mathbf{Q}$  to  $\mathbf{R}$  could be used for the reduction of state-coupling error [7, 8]. The second modification is that, for the first  $n$  steps of the correction of the NEKF, the dynamic state estimates are decoupled from the control loop. The resulting modified NEKF equations are given as

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \bar{\mathbf{H}}^T (\bar{\mathbf{H}} \mathbf{P}_{k|k-1} \bar{\mathbf{H}}^T + \mathbf{R})^{-1}, \quad (15)$$

$$\begin{aligned} \bar{\mathbf{x}}_{k|k} &= \begin{bmatrix} \mathbf{x}_{k|k} \\ \mathbf{w}_{k|k} \end{bmatrix} = \bar{\mathbf{x}}_{k|k-1} \\ &+ \mathbf{K}_k (\mathbf{z}_k - (NN(\mathbf{x}_{k|k-1}, \mathbf{u}_k, \mathbf{w}_{k|k-1}) + \mathbf{h}(\mathbf{x}_{k|k-1}, \mathbf{u}_k))), \end{aligned} \quad (16)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \bar{\mathbf{H}} \mathbf{K}_k) \mathbf{P}_{k|k-1}, \quad (17)$$

$$\bar{\mathbf{x}}_{k+1|k} = \bar{\mathbf{f}}(\bar{\mathbf{x}}_{k|k}) = \begin{bmatrix} \mathbf{x}_{k+1|k}^{\text{acc}} \\ \mathbf{w}_{k|k} \end{bmatrix}, \quad (18)$$

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P}_{k+1|k}^{\text{acc}} & \mathbf{P}_{k|k}^{\text{Block1,2}} \\ \mathbf{P}_{k|k}^{\text{Block2,1}} & \mathbf{P}_{k|k}^{\text{Block2,2}} \end{bmatrix} + \mathbf{Q}_k, \quad (19)$$

where the superscript acc indicates the accurate covariance from the estimator using the other sensors. If more than one poor measurement is produced between accurate measurements, only the last measurement was used for the state estimator [7].

Two approaches for decoupling have been considered. In the first, the measurement noise for the faulty sensor is kept artificially high throughout the experiment. This reduces the effect of the poor sensor even as the neural network trains. In the second approach, as the weights of the neural network settle, the measurement noise is reduced from its artificially high values to values closer to the calibrated sensor noise covariance. This allows the improved model of the sensor to have greater effect on the control loop.

### 3. Control Example I

A simulated control example is used to demonstrate the capability of the NEKF technique to model a sensor change while the system is in operation. A small motorized vehicle is operating in a two dimensional space, as seen in Figure 1.

The goal is for the vehicle to move from one point in the operational grid to a second point in the operational grid. The vehicle maintains its position via a sensor system that provides a range-bearing measurement to a surveyed location in the operational grid. The platform also has a speedometer and compass system that is used to provide the vehicle's heading relative to a local magnetic pole. This is similar to INS systems that are GPS denied.

The state-space model of the vehicle-dynamics is defined as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} + \mathbf{g}(\mathbf{u}), \quad (20)$$

$$\mathbf{g}(\mathbf{u}) = \begin{bmatrix} 0 & a_x & 0 & a_y \end{bmatrix}^T \\ = \begin{bmatrix} 0 & a \cdot \cos(\theta + \psi) & 0 & a \cdot \sin(\theta + \psi) \end{bmatrix}^T, \quad (21)$$

where  $a$  indicates acceleration,  $\theta$  denotes the heading angle, and  $\psi$  denotes the change in heading angle. For a sample rate of  $dt$ , the discretized representation becomes

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k+1} \\ y_k \\ y_{k+1} \end{bmatrix} + \mathbf{g}_k(\mathbf{u}_k). \quad (22)$$

The speed sensor and heading sensor have combined into a single sensor package modeled in (23). The package has an update rate of 0.2 seconds. The accuracy of the compass heading is 2.0 degrees while speedometer accuracy is 0.001 m/s. The position sensor, providing a range ( $r$ ), and bearing ( $\alpha$ ) relative to the surveyed location, is modeled in (24), also with an update rate of 0.2 seconds. The position sensor reports are interleaved with the velocity measurements, as shown in the time line of Figure 2. The

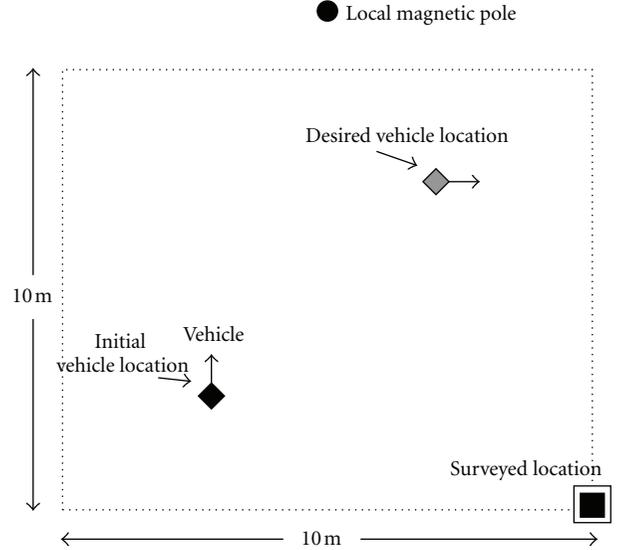


FIGURE 1: Example scenario of an autonomous vehicle using two sensors to achieve a specific end location.

accuracy of the position sensor is given as 1.0 degrees for bearing and 0.005 m for range:

$$\mathbf{z}_k^{\text{vel}} = \begin{bmatrix} s_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \sqrt{\dot{x}_k^2 + \dot{y}_k^2} \\ \arctan\left(\frac{\dot{y}_k}{\dot{x}_k}\right) \end{bmatrix}, \quad (23)$$

$$\mathbf{z}_k^{\text{pos}} = \begin{bmatrix} r_k \\ \alpha_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k - x_{\text{survey}})^2 + (y_k - y_{\text{survey}})^2} \\ \arctan\left(\frac{y_k - y_{\text{survey}}}{x_k - x_{\text{survey}}}\right) \end{bmatrix}. \quad (24)$$

For this effort, the control input has an input rate of 0.1 seconds to synchronize with the sensor reports. The initial location of the vehicle is defined at  $(-5 \text{ m}, -6 \text{ m})$  with a heading of 0 degrees, while the desired final point is  $(0 \text{ m}, 0 \text{ m})$  with a heading of 45.0 degrees. The range-bearing beacon is placed at  $(+3 \text{ m}, -8 \text{ m})$ , while the local north pole is defined at  $(-2 \text{ m}, +3 \text{ m})$ . The position sensor measurement uncertainty matrix had standard deviations of 0.005 m for the range and 1.0 radian for the bearing. For the velocity sensor, the measurement error covariance had standard deviations of 0.001 m/s for the speed and 2 radians for the heading. The broken velocity sensor reported speeds increased by a factor of 1.9 and added a 0.5-radian bias to the heading, in addition to the additive Gaussian noise.

For the NEKF, the process noise for the weights of the neural networks,  $\mathbf{Q}_w$ , and its initial error covariance,  $\mathbf{P}_{wts}$ , were increased to allow for changes based on the residuals using the broken sensor. The process noise for the input weights was set to 1.0 and for the output weights, it was set to 2.0. The initial state error covariance,  $\mathbf{P}_{\text{states}}$ , was set to 100.0.

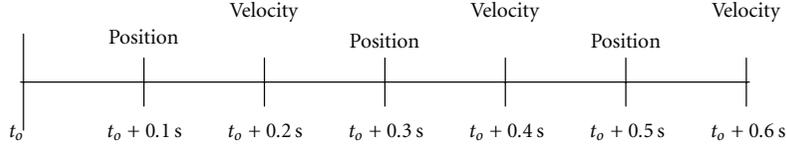


FIGURE 2: Sensor reports from different sensors are interleaved to provide uniform reporting.

The process noise,  $\mathbf{Q}$ , was set to the integrated white noise model [15]:

$$\mathbf{Q} = q^2 \begin{bmatrix} \frac{dt^3}{3} & \frac{dt^2}{2} & 0 & 0 \\ \frac{dt^2}{2} & dt & 0 & 0 \\ 0 & 0 & \frac{dt^3}{3} & \frac{dt^2}{2} \\ 0 & 0 & \frac{dt^2}{2} & dt \end{bmatrix}, \quad (25)$$

with a factor  $q$  of 0.0017. For the NEKF, the initial value for  $\mathbf{P}_{\text{wts}}$  was set to 1000. For comparison purposes, an EKF using the same values for the dynamic state components was generated as well.

Four separate cases, two with the EKF and two with the NEKF were implemented. The four cases are (1) an EKF with an inflation factor of 100 throughout the scenario, (2) an EKF with an inflation factor of 10,000,000 throughout the scenario, (3) an NEKF with an inflation factor of 100 throughout the scenario, and (4) an NEKF with an initial inflation factor of 1000 held for 80 iterations and then decayed by 15% for later iterations down to a minimum value of 1.

Inflating the uncertainty matrix,  $\mathbf{R}$ , of the velocity sensor by a given inflation factor allows the state update component of EKF and of the NEKF to be less affected by these poor measurements. For the NEKF, a 4-node hidden layer was used. The results are shown in Figures 3(a)–3(d).

Figure 3(a) indicates that the EKF is unable to remain stable with the poor measurements having the reduced, but not completely insignificant, effect on the state estimate. By overcompensating with a significant increase in the measurement covariance, the EKF basically eliminates all of the poor sensor reports. This is seen in Figure 3(b). However, the vehicle never settles in at the location. The NEKF results are shown in Figures 3(c) and 3(d) where it learns sensor errors while on line to provide clearly improved control performance. Even with the same measurement covariance as the EKF in first case, as in Figure 3(c), the NEKF implementation remains stable. In both Figures 3(c) and 3(d), the results are quite similar with more changes after the measurement noise decays.

With multiple sensor systems, the NEKF is able to provide a fault correcting mechanism for sensors that are still providing information, although that information needs correction. The research of this effort has also shown that the slower the decay rate is on the inflation factor, the lower the initial inflation factor can be.

## 4. Control Example II

The small motorized vehicle operating in a two-dimensional space shown in Figure 1 is also the basis for the second control example. In this case, the vehicle maintains its position via a two-sensor system. One sensor provides a range-bearing measurement to a surveyed location in the operational grid and another, slower reporting, position sensor provides a linear position report. This would be similar to a GPS system updating an INS.

The state-space model of the vehicle-dynamics is defined as in (20)–(22). The range-bearing error is described as in (23). The position sensor has an update rate of 0.6 seconds and provides a linear report as seen in (26). The accuracy of the position sensor is assumed to be  $\pm 0.01$  m in both directions. The other position sensor, providing a range ( $r$ ), and bearing ( $\alpha$ ) relative to the surveyed location, is modeled in (24) and has an update rate of 0.2 seconds. The accuracy of the faster, range-bearing position sensor is given as 1.0 degree of bearing accuracy and 0.005 m of range accuracy:

$$\mathbf{z}_k^{\text{pos}} = \begin{bmatrix} x_k \\ y_k \end{bmatrix}. \quad (26)$$

The sensor reports of positions are interleaved as shown in the time line of Figure 4. For this effort, the control input has an input rate of 0.2 seconds to synchronize with the range-bearing sensor report.

The initial location of the vehicle is defined at (−5 m, −6 m) with a heading of 0 degrees, while the desired final point is (0 m, 0 m) with a heading of 45.0 degrees. The range-bearing beacon is placed at (+3 m, −8 m). The position sensor providing linear reports provides measurements with a bias of (0.5 m, −0.25 m) added to the coordinates and is considered to be miscalibrated.

For the NEKF, the process noise for the weights of the neural networks,  $\mathbf{Q}_w$ , and its initial error covariance,  $\mathbf{P}_{\text{wts}}$ , were increased to allow for changes based on the residuals using the broken sensor. The process noise for the input weights was set to 1.0 and for the output weights, it was set to 2.0. The initial state error covariance,  $\mathbf{P}_{\text{states}}$ , was set to 10.0. The process noise,  $\mathbf{Q}$ , was set to the integrated white noise model [14] of (25) with a factor  $q$  of 0.0017. For the NEKF, the initial value for  $\mathbf{P}_{\text{wts}}$  was set to 0.1. The inflation factor for the broken sensor in both the NEKF and the EKF tests was set to 100. For comparison purposes, the EKF that was generated used the same values for the dynamic state components.

In this example case, when the system had two fully functional sensors, the vehicle took 3.6 seconds to reach its desired endpoint with a 0.001 m total distance error.

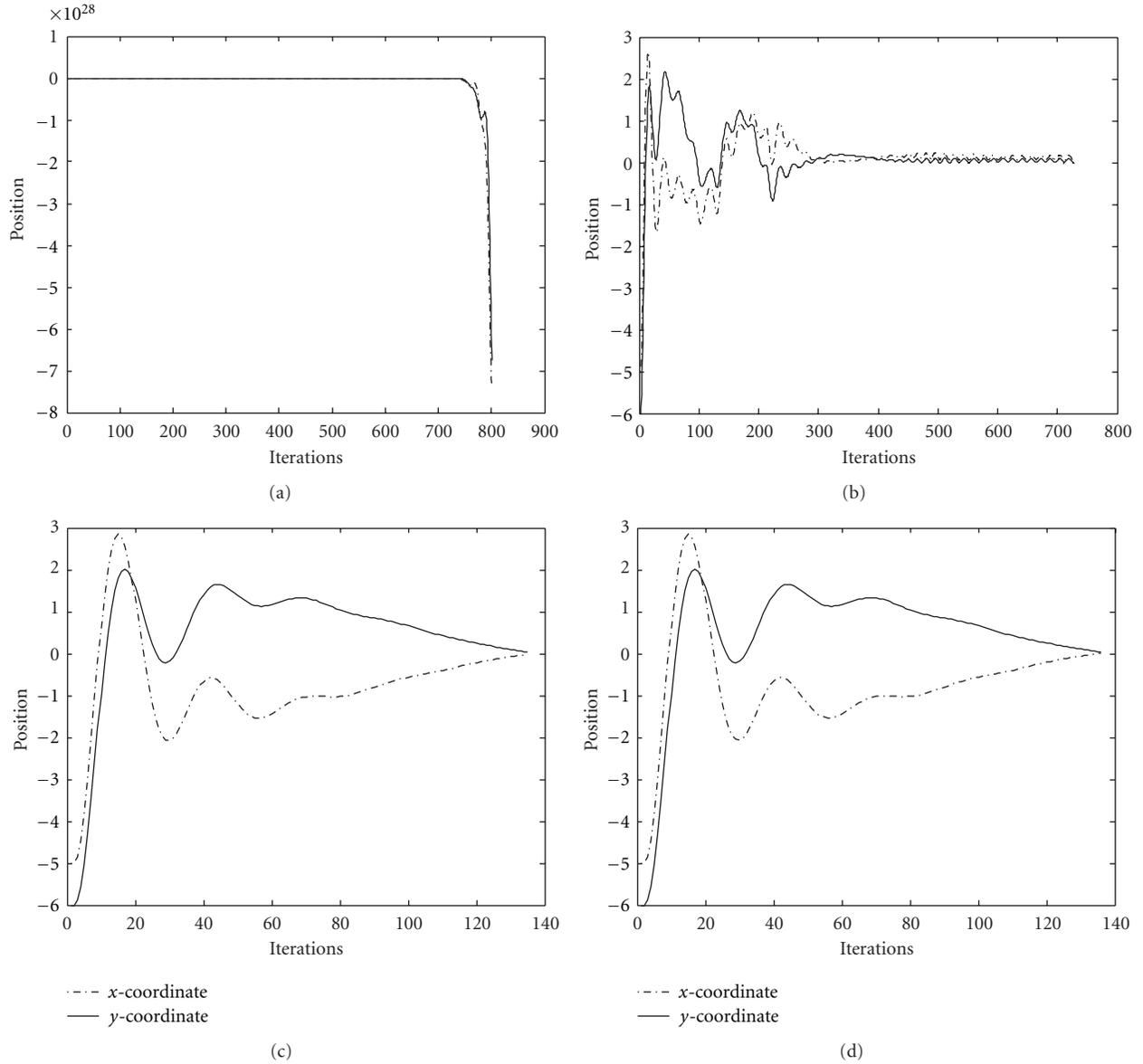


FIGURE 3: (a) Position coordinates using the EKF and a low inflation rate. (b) Position coordinates using the EKF and a high inflation rate. (c) Position coordinates using the NEKF and a low inflation rate. (d) Position coordinates using the NEKF and a higher inflation rate that is allowed to decay after 80 iterations.

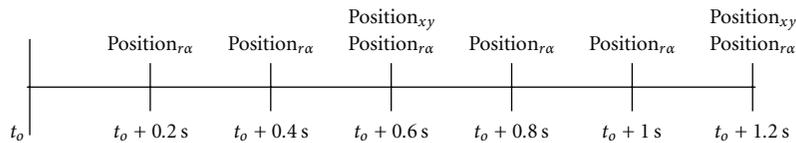


FIGURE 4: Sensor reports from different sensors are interleaved to provide uniform reporting.

A comparison between the EKF with the bad sensor and the NEKF with the bad sensor was generated. Two different comparisons were run. In the first comparison, an inflation factor of 10 for measurement error covariance matrix of the broken sensor was used. Inflating the uncertainty matrix,  $\mathbf{R}$ , of the velocity sensor by a given inflation factor allows the state update component of EKF and of the NEKF to be less

affected by these poor measurements. For the NEKF, a 4-node hidden layer was used. In this case, both the NEKF and the EKF went unstable.

In the second comparison, an inflation factor of 1000 was used. The results of the four states,  $x$ -position,  $x$ -velocity,  $y$ -position, and  $y$ -velocity are shown for the EKF in Figure 5. While the individual elements are hard to discern,

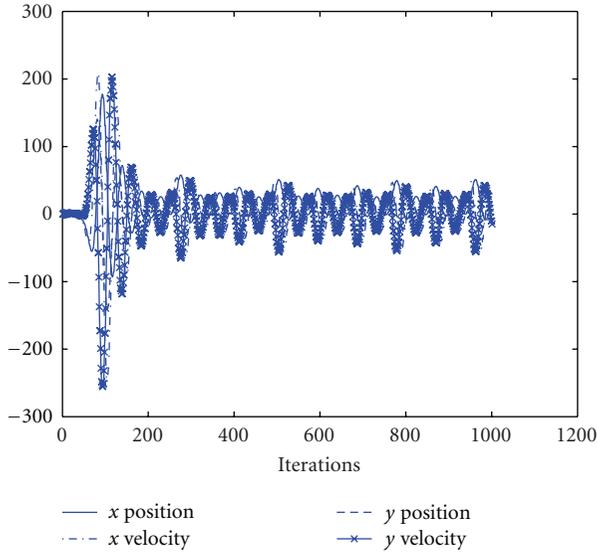


FIGURE 5: EKF state responses.

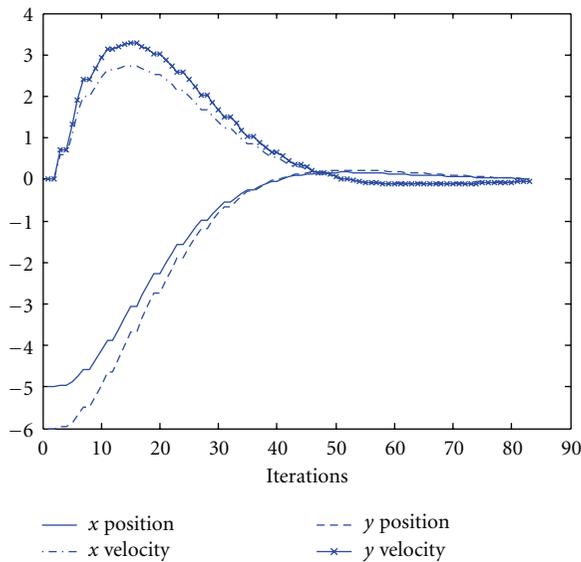


FIGURE 6: NEKF state responses.

the important fact is that none of the values converge. This causes the vehicle to leave the region of interest. Figure 6 depicts the NEKF results for the four states. As is clearly seen, both techniques achieve a result, but the NEKF trajectory is much smoother and the states are less erratic and require significantly fewer iterations to converge to the desired result (84 iterations versus 498 iterations).

If the position sensor was miscalibrated and not corrected but only had a reduced effect on the navigation estimator, as would be the case when using the EKF only and assumed it were inaccurate, the vehicle position becomes unstable causing it to leave the grid area within 10 seconds of beginning the test. When the NEKF was used, the vehicle was able to reach its desired endpoint in 38.4 seconds with 0.45 m total distance error and 0.3 m/s velocity error.

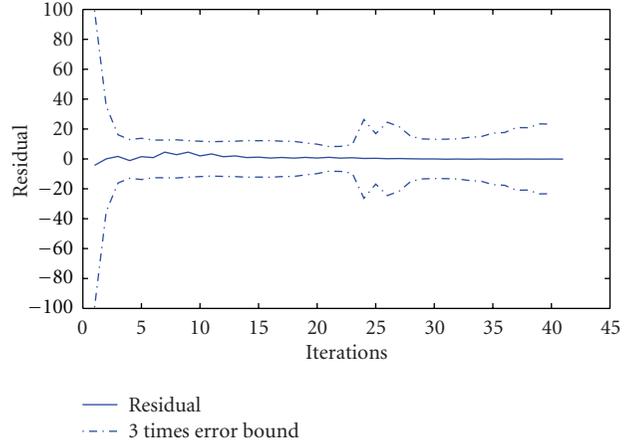


FIGURE 7: Comparison of EKF residual and the error covariance weighting.

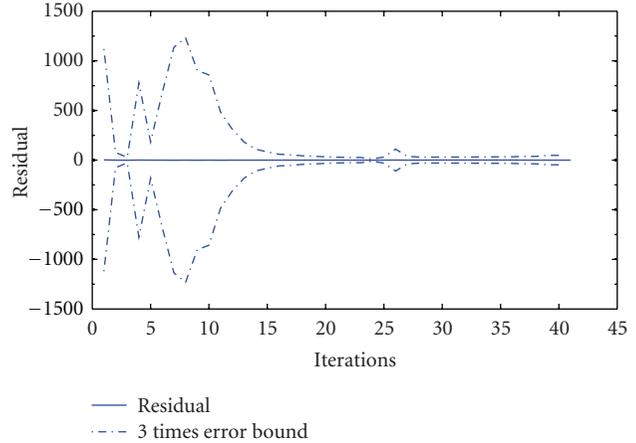


FIGURE 8: Comparison of NEKF residual and the error covariance weighting.

One of the interesting comparisons is that of the residual errors for heading in this example compared to a 3-sigma error based upon the system and sensor covariances. Figure 7 shows the results for the EKF, and Figure 8 shows the results for the NEKF. In both cases, the residuals are within the bounds, but the NEKF has a more accurate bound in the steady-state portion allowing for the Kalman gain to permit greater residual effect on the state update for the corrected sensor. Thus, without the NEKF, the system would go unstable with deleterious effects almost immediately. With the NEKF, the vehicle remains in the operational area and slowly converges to the desired point.

### 5. Conclusion

A new neural extended Kalman filter algorithm has been developed that can improve sensor modeling for a poorly modeled sensor in the control loop. The technique expanded on an approach originally developed for target tracking

that relied upon that availability of truth. In the algorithm, properly modeled, operating sensors were used to provide truth used to recalibrate the sensor that is poorly modeled. This NEKF approach decouples its state estimates from the accurate estimates using the measurement error covariance.

The NEKF was shown to have improved performance over that of the EKF in its application to estimate the states of the control loop of an autonomous vehicle for two example cases. Performance of the algorithm will continue to be evaluated in other applications and with different training parameters.

## References

- [1] R. Lobbia, E. Frangione, and M. Owen, "Noncooperative target sensor registration in a multiple-hypothesis tracking architecture," in *Signal and Data Processing of Small Targets*, vol. 3163 of *Proceedings of SPIE*, San Diego, Calif, USA, July 1997.
- [2] S. Blackman, *Multiple-Target Tracking with Radar Applications*, Artech House, Norwood, Mass, USA, 1986.
- [3] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, Institute of Electrical Engineers, London, UK, 1997.
- [4] W. S. Levine, Ed., *The Control Handbook*, CRC Press, Boca Raton, Fla, USA, 1996.
- [5] A. Martinelli, "State estimation on the concept of continuous symmetry and observability analysis: the case of calibration," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 239–255, 2011.
- [6] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.
- [7] S. C. Stubberud and K. A. Kramer, "Control loop sensor calibration using neural networks," in *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC '08)*, pp. 472–477, Victoria, Canada, May 2008.
- [8] S. C. Stubberud, R. N. Lobbia, and M. Owen, "Adaptive extended Kalman filter using artificial neural networks," *International Journal of Smart Engineering System Design*, vol. 1, no. 3, pp. 207–221, 1998.
- [9] A. R. Stubberud, "A validation of the neural extended kalman filter," in *Proceedings of the International Conference on Systems Engineering*, pp. 3–8, Coventry, UK, September 2006.
- [10] S. C. Stubberud, R. N. Lobbia, and M. Owen, "Adaptive extended Kalman filter using artificial neural networks," in *Proceedings of the 34th IEEE Conference on Decision and Control*, pp. 1852–1856, New Orleans, La, USA, December 1995.
- [11] D. Klimánek and B. Šulc, "Evolutionary detection of sensor discredibility in control loops," in *Proceedings of the 31st Annual Conference of the IEEE Industrial Electronics Society (IECON '05)*, vol. 2005, pp. 136–141, Raleigh, NC, USA, November 2005.
- [12] S. C. Stubberud, K. A. Kramer, and J. A. Geremia, "On-line sensor modeling using a neural Kalman filter," in *Proceedings of the 23rd IEEE Instrumentation and Measurement Technology Conference (IMTC '06)*, pp. 969–974, Sorrento, Italy, April 2006.
- [13] K. A. Kramer, S. C. Stubberud, and J. A. Geremia, "Target registration correction using the neural extended kalman filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 7, Article ID 5280378, pp. 1964–1971, 2010.
- [14] S. C. Stubberud, K. Kramer, and A. R. Stubberud, "Parameter estimation using a novel nonlinear constrained sequential state estimator," in *Proceedings of the UKACC International Conference on Control*, pp. 1031–1036, Coventry, UK, September 2010.
- [15] M. J. Gao, J. W. Tian, and K. Li, "The study of soft sensor modeling method based on wavelet neural network for sewage treatment," in *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR '07)*, vol. 2, pp. 721–726, November 2007.
- [16] Enderle, G. K. Kraetzschmar, Sablatnoeg, and Palm, "One sensor learning from another," in *Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN '99)*, vol. 2, no. 470, pp. 755–760, September 1999.
- [17] J. W. M. van Dam, B. J. A. Krose, and F. C. A. Groen, "Adaptive sensor models," in *Proceedings of the IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 705–712, December 1996.
- [18] M. S. Santina, A. R. Stubberud, and G. H. Hostetter, *Digital Control System Design*, Saunders College Publishing, Fort Worth, Tex, USA, 2nd edition, 1994.
- [19] A. K. Mahalanabis, "Introduction to random signal analysis and kalman filtering. Robert G. Brown," *Automatica*, vol. 22, no. 3, pp. 387–388, 1986.
- [20] S. Singhal and L. Wu, "Training multilayer perceptrons with the extended kalman algorithm," in *Advances in Neural Processing Systems I*, D. S. Touretsky, Ed., pp. 133–140, Morgan Kaufmann, 1989.
- [21] S. C. Iglehart and C. T. Leondes, "Estimation of a dispersion parameter in discrete kalman filtering," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 262–263, 1974.
- [22] Brotherton, Johnson, and Chadderdon, "Classification and novelty detection using linear models and a class dependent—elliptical basis function neural network," in *Proceedings of the IEEE World Congress on Computational Intelligence (IJCNN '98)*, vol. 2, pp. 876–879, Anchorage, Alaska, USA, 1998.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

