doi: 10.82124/CEAS-GNC-2026-095

CEAS-GNC-2026-095

EuroGNC Navigation Guidance Control CEAS

Madrid, Spain May 5th-7th 2026

uc3m Universidad Carlos III de Madrid

AIAA

SNN-Based Kernel Bandwidth Prediction for Adaptive MCUKF in GNSS/INS Integration

Minhwan Kim M.S. student, Konkuk University, Department of Aerospace Mobility Engineering, Seoul, South Korea. duru1211@konkuk.ac.kr

Sangkyung Sung Professor, Konkuk University, Department of Aerospace Mobility Engineering, Seoul, South Korea. sksung@konkuk.ac.kr

ABSTRACT

Reliable navigation in urban environments remains challenging due to non-Gaussian noise and time-varying error characteristics that degrade the performance of integrated Global Navigation Satellite System/Inertial Navigation System (GNSS/INS) solutions. This paper proposes a novel adaptive filtering method based on a Shallow Neural Network (SNN) incorporated into a Maximum Correntropy Unscented Kalman Filter (MCUKF) framework. The proposed SNN-MCUKF dynamically adjusts the robustness factor, known as the kernel bandwidth, by learning from the statistical history of measurement residuals. This adaptive mechanism enables the filter to maintain both robustness and statistical consistency even under rapidly changing environmental conditions. To achieve this, the SNN estimates the kernel bandwidth online using a self-supervised learning approach driven by the exponential moving average of the residual energy. This allows the filter to respond in real time to variations in measurement errors such as multipath and non-line-of-sight effects, which are common in dense urban areas. Experiments were conducted using a GNSS/INS platform operating in mixed urban environments with varying levels of signal obstruction. The results demonstrate that the proposed method achieves significantly improved estimation stability compared with conventional Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and fixed-bandwidth MCC-based filters. Notably, the SNN-MCUKF effectively suppresses long-term drift in attitude estimation and maintains consistent position and velocity accuracy even under severe multipath interference. These findings confirm that the SNN-MCUKF provides a practical and computationally efficient solution for integrated navigation in non-Gaussian and dynamic conditions, enhancing both the reliability and robustness of GNSS/INS fusion systems.

Keywords: GNSS/INS integration, robust filtering, maximum correntropy, shallow neural network

Nomenclature

x_k, z_k = state and measurement vector at time step k

w_{q_k}, w_{r_k} = Process and measurement noise

P_k = State covariance matrix

K_k = Kalman gain

$f(\cdot), h(\cdot)$ = Nonlinear state and measurement functions
e_k, r_k = Measurement error and residual norm
σ = Kernel bandwidth
σ^* = Adaptive kernel bandwidth predicted by SNN
σ^t = Target kernel bandwidth
W_1, W_2 = Weight matrices of the SNN
b_1, b_2 = Bias vectors of the SNN
C_k^x, C_k^y = Diagonal correntropy weighting matrices
S_p, S_r = Cholesky decompositions of covariance matrices

1 Introduction

Various unmanned systems such as autonomous vehicles and drones require global Position, Navigation, and Timing (PNT) performance, and the representative navigation system to provide this is the Global Navigation Satellite System (GNSS). Specifically, GNSS offers advantages in its wide operational range and long-term accuracy, providing absolute position information for moving objects [1]. However, GNSS cannot maintain a fast update rate necessary for the control of dynamic systems, and its navigation solution is highly sensitive to the distribution of visible satellites or the signal propagation environment in satellite signal-obstructed areas such as urban canyons [2]. Conversely, the Inertial Navigation System (INS) can perform fast and accurate position measurements within a short time, but it has a tendency for errors to accumulate over time. From a complementary perspective, the INS/GNSS integrated navigation system can overcome these mutual drawbacks by providing more accurate position and attitude information [3]. However, in environments densely populated with obstacles like tall buildings, the GNSS signal is blocked or reflected, inducing Multipath (MP) and Non-Line-of-Sight (NLOS) errors. This acts as a primary factor that severely degrades the position estimation performance of the integrated system [4, 5].

To address the state estimation problem, the Kalman Filter (KF) and its extended techniques (EKF, UKF) are widely used, but they assume Gaussian noise. When the system is exposed to a non-Gaussian environment, particularly heavy-tailed noise, the estimation performance is likely to degrade. Consequently, the Maximum Correntropy Criterion (MCC)-based Kalman filter has been proposed as a valid alternative. MCC-based filters effectively suppress the influence of outliers through Gaussian and various kernels applied to the residual [6-9].

However, conventional MCC-based filters face a structural limitation in that the kernel bandwidth is operated at a fixed value. Consequently, in dynamic environments where GNSS error characteristics are time-varying, the filter may not be able to actively adapt to the errors. Research efforts have thus been made to adaptively vary the kernel bandwidth [10, 11]. However, these techniques have the disadvantage of high computational burden.

This study proposes an adaptive kernel bandwidth prediction method based on a Shallow Neural Network (SNN) to overcome the limitations of this fixed robustness factor [12]. Distinct from conventional adaptive strategies that typically rely on simple residual-driven rules or heuristic thresholds, the proposed technique utilizes the residual statistical history to learn the optimal non-linear mapping for the kernel bandwidth online. This capability enables the filter to actively suppress outliers and minimize estimation drift by capturing complex error characteristics that pre-defined linear models fail to address. Consequently, the primary contributions of this work include the development of an effective SNN-based

adaptive mechanism and the empirical validation of its enhanced stability through rigorous comparisons with various conventional and robust filters—including EKF, UKF, and fixed-bandwidth MCUKF variants—particularly in suppressing long-term attitude drift under severe multipath conditions.

2 Preliminary

2.1 Maximum Correntropy Criterion

Correntropy represents the local similarity between two random variables. Given two random variables, $X \in \mathbb{R}$ and $Y \in \mathbb{R}$, Correntropy is defined as in equation (1) [6].

$$V(X,Y) = E[\kappa(X,Y)] = \int \kappa(x,y) dF_{XY}(x,y) \tag{1}$$

where E denotes the expected value, $\kappa(\cdot,\cdot)$ is a shift-invariant Mercer kernel, and $F_{XY}(x,y)$ is the joint distribution function of the two random variables. The kernel typically refers to the Gaussian kernel, which is expressed as in equation (2).

$$\kappa(x,y) = G_\sigma(e) = \exp\left(-\frac{e^2}{2\sigma^2}\right) \tag{2}$$

where $e = x - y$ and $\sigma > 0$ is kernel bandwidth.

Estimation based on maximum correntropy criterion (MCC) involves solving an optimization problem as shown in equation (3).

$$\widehat{W} = \arg\max_W \frac{1}{N}\sum_{i=0}^{N} G_\sigma(e_i) \tag{3}$$

where $\widehat{W}$ is the optimal solution and W is the parameter vector of the model to be estimated.

2.2 Traditional Unscented Kalman Filter

Let us consider the nonlinear system described below.

$$x_k = f(x_{k-1}) + w_{q_{k-1}}, \tag{4}$$

$$z_k = h(x_k) + w_{r_k} \tag{5}$$

where $x_k \in \mathbb{R}^n$ is the state vector at the kth step, and $z_k \in \mathbb{R}^m$ is the measurement vector. The term f is the nonlinear system function and h is the nonlinear measurement function. Both functions are assumed

to be continuously differentiable. q_{k-1} is the process noise and r_k is the measurement noise. Both noise terms are assumed to be Gaussian white noise and independent.

Generally, UKF(Unscented Kalman Filter) contains the following two steps: time update and measurement update.

Time Update:

$2n+1$ samples, referred to as sigma points, are generated from the estimated state vector $\hat{x}_{k-1|k-1}$ and the covariance matrix $P_{k-1|k-1}$ at the previous time step $k-1$.

$$\begin{aligned} \chi^0_{k-1|k-1} &= \hat{x}_{k-1|k-1} \\ \chi^i_{k-1|k-1} &= \hat{x}_{k-1|k-1} + \left(\sqrt{(n+\lambda)P_{k-1|k-1}}\right) \text{ for } i = 0, \cdots, 2n_i \\ \chi^i_{k-1|k-1} &= \hat{x}_{k-1|k-1} - \left(\sqrt{(n+\lambda)P_{k-1|k-1}}\right)_i \text{ for } i = 0, \cdots, 2n \end{aligned} \tag{6}$$

where $\sqrt{(n+\lambda)P_{k-1|k-1}}$ is i-th column of the matrix square root of $(n+\lambda)P_{k-1|k-1}$ n denotes the dimension of the state and $\lambda = \alpha^2(n+\kappa) - n$.

The transformed points are calculated according to equation (7). Subsequently, the a priori state is estimated via equation (8) and the a priori covariance matrix is estimated via equation (9).

$$\chi^{i^*}_{k|k-1} = f\left(k-1, \chi^i_{k-1|k-1}\right) \text{ for } i = 0, \cdots, 2n \tag{7}$$

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2n} w^i_m \chi^{i^*}_{k|k-1} \tag{8}$$

$$P_{k|k-1} = Q_{k-1} + \sum_{i=0}^{2n} w^i_c \left[\chi^{i^*}_{k|k-1} - \hat{x}_{k|k-1}\right]\left[\chi^{i^*}_{k|k-1} - \hat{x}_{k|k-1}\right]^T \tag{9}$$

where the weights correspond to the state and the covariance matrix are given by the following:

$$\begin{aligned} w^0_m &= \frac{\lambda}{(n+\lambda)}, \\ w^0_c &= \frac{\lambda}{(n+\lambda)} + (1-\alpha^2+\beta), \\ w^i_m &= w^i_c = \frac{1}{2(n+\lambda)} \text{ for } i = 1, \cdots, 2n \end{aligned} \tag{10}$$

Measurement Update:

The measurements, $\hat{z}_k$, for the previously calculated sigma points are computed as shown in equation (11). Furthermore, the measurement covariance and the cross-covariance are calculated as shown in equation (12) and (13).

$$\hat{z}_k = \sum_{i=0}^{2n} w_m^i \cdot h\left(\chi_{k|k-1}^{i^*}\right) \tag{11}$$

$$P_{zz,k} = R + \sum_{i=0}^{2n} w_c^i \left[h\left(\chi_{k|k-1}^{i^*}\right) - \hat{z}_k\right]\left[h\left(\chi_{k|k-1}^{i^*}\right) - \hat{z}_k\right]^T \tag{12}$$

$$P_{xz,k} = \sum_{i=0}^{2n} w_c^i \left[\chi_{k|k-1}^{i^*} - \hat{x}_{k|k-1}\right]\left[h\left(\chi_{k|k-1}^{i^*}\right) - \hat{z}_k\right]^T \tag{13}$$

Based on the calculated covariances, the Kalman gain is computed as in equation (14) and the a posteriori covariance, and the state are finally calculated.

$$K_k = P_{xz,k}\left(P_{zz,k}\right)^{-1} \tag{14}$$

$$P_{k|k} = P_{k|k-1} - K_k P_{zz,k} K_k^T \tag{15}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z - \hat{z}_k) \tag{16}$$

3 Learning-Based Adaptive Kernel Bandwidth

Previous studies on the Maximum Correntropy Kalman Filter (MCKF) have relied on a fixed kernel bandwidth [8, 9], which limits the filter's ability to adapt to dynamic environmental changes. Conventional rule-based adaptive strategies, while offering some flexibility, often suffer from high sensitivity to tuning parameters and fail to track rapid, non-Gaussian error fluctuations in complex urban canyons. To overcome these limitations, this paper proposes an adaptive kernel bandwidth prediction method based on a Shallow Neural Network (SNN). Unlike rigid heuristic rules, the SNN-based approach learns the non-linear relationship between residual statistics and the optimal bandwidth, enabling robust tracking and reduced sensitivity to manual parameter tuning.

In the k-th measurement update, the measurement error vector $e_k \in \mathbb{R}^m$ is defined as the difference between the actual measurement z_k and estimated measurement $\hat{z}_k$, as shown in equation (17). The residual, expressed by equation (18), is obtained.

$$e_k = z_k - \hat{z}_k, \tag{17}$$

$$r_k \equiv \|e_k\|_2. \tag{18}$$

The input vector of the SNN, denoted as $u_k \in \mathbb{R}^{N+1}$, is constructed as shown in equation (19).

$$u_k = [E_k,\ q_k] \tag{19}$$

where $E_k = [r_{k-N+1},\ \cdots, r_k]$, which represents a buffer for the l_2 norm of the error and $q_k = \sqrt{\frac{1}{N}\sum_{i=1}^{N} r_{k-N+i}^2}$.

Furthermore, online regularization for learning is performed as shown in equation (20).

$$\bar{u}_k = \frac{u_k - \mu_k}{\Sigma_k} \tag{20}$$

where μ and Σ represents mean and standard deviation.

The regularization input undergoes forward propagation through the SNN containing a single hidden layer to obtain the output O_k. The hidden layer of the neural network employs the hyperbolic tangent activation function. The output of the hidden layer and the final linear output are as follows in equation (21)-(23).

$$z_k = W_1 \bar{u}_k + b_1 \tag{21}$$

$$h_k = \tanh(z_k) \tag{22}$$

$$O_k = W_2 h_k + b_2 \tag{23}$$

The obtained output is mapped via sigmoid function to ensure it remains within the physically meaningful boundaries $[\sigma_{\min}, \sigma_{\max}]$.

$$\sigma_k^* = \sigma_{min} + (\sigma_{max} - \sigma_{min}) \frac{1}{1 + \exp(-O_k)} \tag{24}$$

The kernel bandwidth predicted by neural network is used to weight the covariance during the measurement update process using a Gaussian kernel, similar to previous MCKF studies [6, 7]. This process is described below in equations (25)-(26).

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \bar{K}_k\left(z_k - H_k \hat{x}_{k|k-1}\right) \tag{25}$$

where

$\bar{K}_k = \bar{P}_{k|k-1} H_k^T \left(H_k \bar{P}_{k|k-1} H_k^T + \bar{R}_k\right)^{-1}$,

$\bar{P}_{k|k-1} = S_p [C_k^x]^{-1} S_p^T$,

$\bar{R}_k = S_r \left[C_k^y\right]^{-1} S_r^T$,

with S_p and S_r being the Cholesky decomposition of the covariance matrix, and $C_k^x = diag\left(G_{\sigma^*}(e_k^1), \cdots, G_{\sigma^*}(e_k^n)\right)$, $C_k^y = diag\left(G_{\sigma^*}(e_k^{n+1}), \cdots, G_{\sigma^*}(e_k^{n+m})\right)$

The a posteriori covariance is updated as follows.

$$P_{k|k} = (I - \bar{K}_k H_k) P_{k|k-1} (I - \bar{K}_k H_k)^T + \bar{K}_k R_k \bar{K}_k^T \tag{26}$$

The estimation of σ^* is learned online at every step, which is implemented using a self-supervised approach that defines the target value through the exponential moving average (EMA) of the root mean square (RMS).

$$\sigma_k^t = \alpha \sigma_{k-1}^t + (1 - \alpha) q_k \tag{27}$$

where $0 < \alpha < 1$.

The learning is optimized by minimizing a log-domain loss function.

$$\mathcal{L}_k = \frac{1}{2} (\log(\sigma_k^*) - \log(\sigma_k^t))^2 \tag{28}$$

Backpropagation is performed based on equation (28) to update the weights of the SNN.

The specific parameters used in this study are listed in Table 1.

Table 1 Parameters of SNN-MCUKF

Parameter	Value	Description
SNN Architecture		
Input Dimension	6	Buffer size
Hidden Neurons	8	Number of nodes in hidden layer
Activation Function	Tanh / Sigmoid	Hidden / Output mapping
Learning Settings		
Learning Rate	1e-4	SGD Learning rate
Optimizer	SGD	Stochastic Gradient Descent
Buffer Length	5	Sliding window size
Target Smoothing	0.9	EMA coefficient

4 Experiment configuration and Results

To validate the proposed SNN-MCUFK algorithm, experiments were conducted in an environment containing both appropriately tall and short buildings. The experimental path was divided into four distinct segments based on the environment. Segment 1 represents a typical urban environment, featuring low-rise buildings and a tall building at the end of the segment. Segment 2 is characterized by buildings on both sides, leading to a slightly more pronounced multipath error. Segment 3 is the environment with the most severe multipath error, where buildings are arranged along both sides of a narrow street. Finally, Segment 4 is the recovery segment, returning to the starting point from the severely multipath-affected environment.

Data acquisition was entirely performed on an MCU based on Cortex-M4, utilizing the ADIS16448 for the IMU and a single Ublox F9P as the GNSS receiver. The IMU data was collected at a frequency of 100 Hz and the GNSS data at 5 Hz, with both being logged to an SD card. To verify the navigation performance, a Novatel SPAN was used as the reference system. This system employed a DGPS receiver and a tactical grade IMU, the KVH1750, with data processing performed using Inertial Explorer. The data collection was performed using an Unmanned Ground Vehicle (UGV). The GNSS antennas were mounted on a pole attached to the UGV at a height of approximately 1.8 meters from the ground. For the antenna configuration, the reference system (Novatel SPAN) utilized a Harxon HX-CSX627A antenna, while the Ublox F9P receiver used for the proposed algorithm was connected to a Taoglas AA.200.151111 antenna.

The state model used in this study is given by equation (29).

$$\hat{x}_{k-1} = \begin{bmatrix} \hat{P}_{k-1} \\ \hat{V}_{k-1} \\ \hat{\Phi}_{k-1} \\ \hat{b}^{a}{}_{k-1} \\ \hat{b}^{g}{}_{k-1} \end{bmatrix} \tag{29}$$

where $\hat{P}_{k-1}$ denotes the position expressed in the navigation frame (n-frame), $\hat{V}_{k-1}$ is the velocity in the navigation frame, and $\hat{\Phi}_{k-1}$ represents the attitude expressed in terms of Euler angles. $\hat{b}^{a}{}_{k-1}$ and $\hat{b}^{g}{}_{k-1}$ denote the accelerometer and gyroscope biases, respectively.

The experimental trajectory is shown in Figure 1, and a comparison was conducted with MCKF-based filters, specifically against the MCUKF utilizing a Gaussian kernel, a Cauchy kernel, and a Gaussian-Cauchy mixed kernel, all with a fixed kernel bandwidth. The weight assigned to the Gaussian kernel in the mixed kernel was set to 0.7. The error results for the 3D position, velocity, and attitude can be confirmed in Figures 2-4.

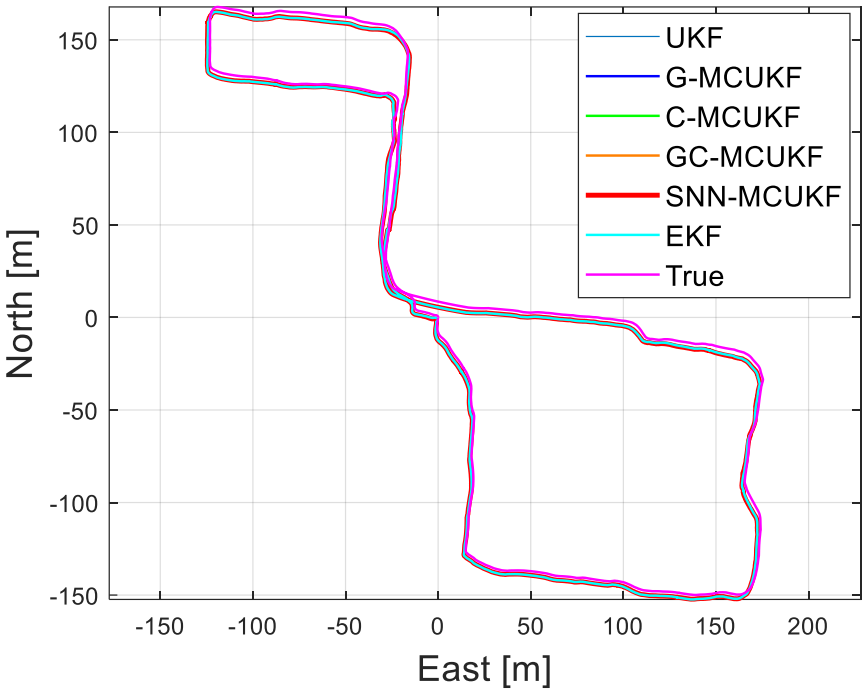

Fig. 1 Two-dimensional position results.

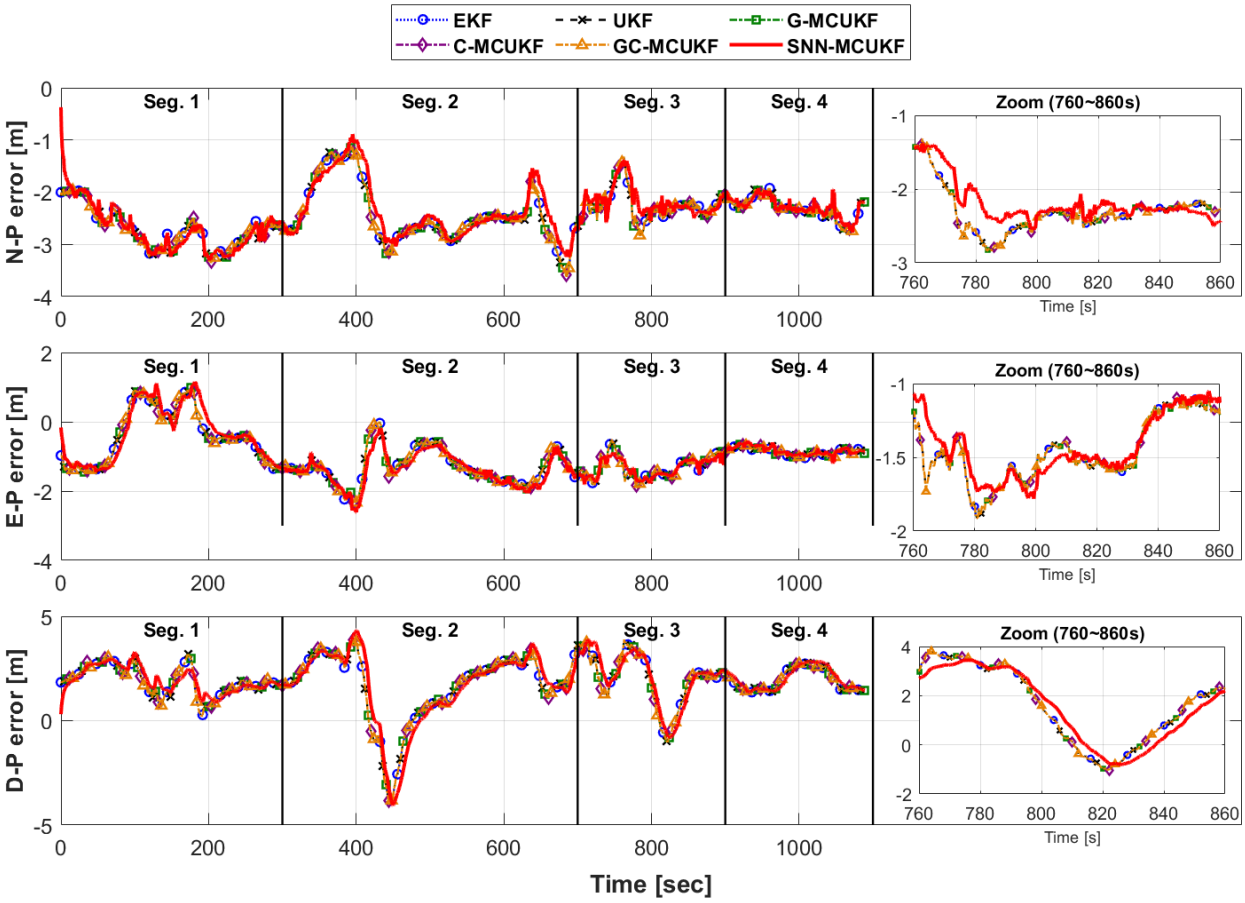

Fig. 2 Three-dimensional position error.

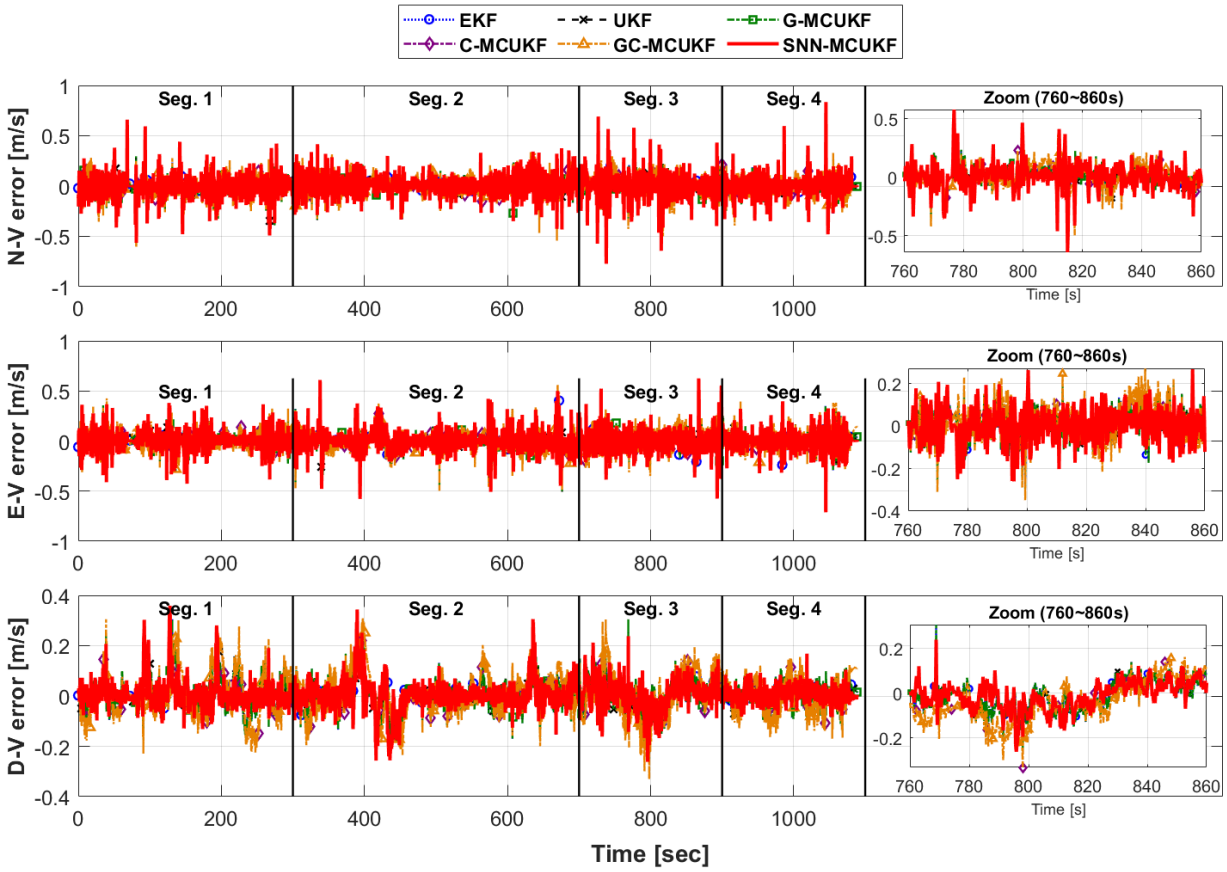

Fig. 3 Three-dimensional velocity error.

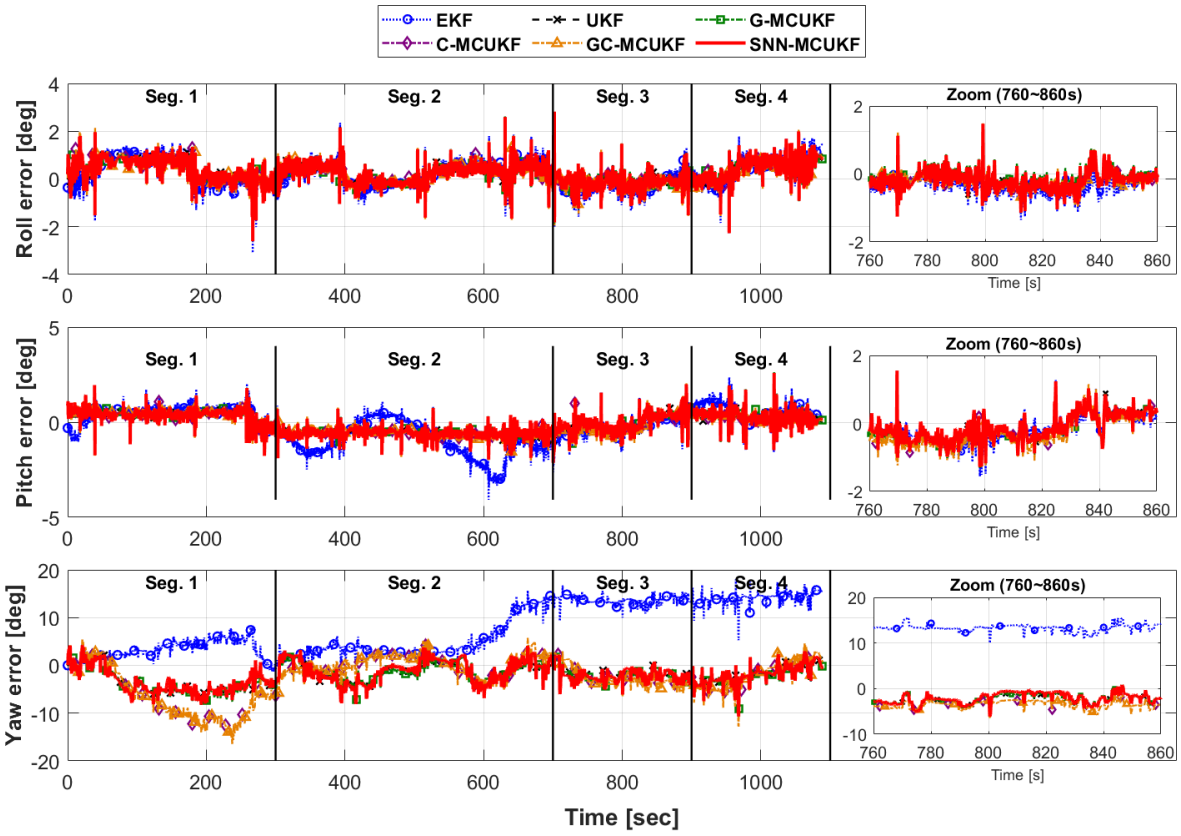

Fig. 4 Three-dimensional attitude error.

Figures 2 through 4 present the estimation errors for position, velocity, and attitude. In the benign environments of Seg. 1 and Seg. 2 (0–700 s), all filters showed comparable performance, validating the nominal accuracy of the SNN-MCUKF. However, in the severe multipath environment of Seg. 3 (700–900 s), the EKF and UKF exhibited significant degradation, with the EKF suffering a catastrophic yaw divergence of 13.58 degrees (RMSE). While fixed-bandwidth MCUKFs reduced this drift to approximately 3.09 degrees, the SNN-MCUKF demonstrated superior robustness, limiting yaw drift to 2.46 degrees. This improvement is attributed to the SNN's ability to dynamically adapt the kernel bandwidth to time-varying error characteristics, effectively suppressing outliers.

The Root Mean Square Error (RMSE) for the position, velocity, and attitude in each segment is tabulated in Table 2 below.

Table 2 RMSE for each segment

	Method	Position RMSE [m]			Velocity RMSE [m/s]			Attitude RMSE [deg]		
		North	East	Down	North	East	Down	Roll	Pitch	Yaw
Seg1	EKF	2.75	0.88	2.03	0.10	0.09	0.06	0.69	0.66	4.32
	UKF	2.75	0.88	2.02	0.08	0.07	0.06	0.61	0.50	4.42
	G-MCUKF	2.75	0.88	2.02	0.08	0.07	0.06	0.61	0.50	4.43
	C-MCUKF	2.75	0.88	2.02	0.09	0.09	0.09	0.62	0.51	8.28
	GC-MCUKF	2.75	0.88	2.02	0.08	0.09	0.09	0.62	0.51	8.28
	SNN-MCUKF	2.74	0.86	2.02	0.09	0.07	0.05	0.61	0.49	3.81
Seg2	EKF	2.50	1.45	2.40	0.09	0.10	0.06	0.53	1.34	6.08
	UKF	2.50	1.40	2.36	0.08	0.08	0.06	0.48	0.63	2.30
	G-MCUKF	2.50	1.40	2.36	0.08	0.08	0.06	0.48	0.63	2.27
	C-MCUKF	2.50	1.40	2.36	0.09	0.09	0.07	0.48	0.66	2.25
	GC-MCUKF	2.50	1.40	2.36	0.09	0.09	0.07	0.48	0.66	2.25
	SNN-MCUKF	2.42	1.40	2.36	0.07	0.08	0.06	0.48	0.63	2.23
Seg3	EKF	2.27	1.43	2.49	0.13	0.10	0.06	0.50	0.59	13.58
	UKF	2.27	1.43	2.43	0.10	0.09	0.06	0.40	0.47	2.76
	G-MCUKF	2.27	1.43	2.43	0.10	0.09	0.06	0.40	0.47	2.76
	C-MCUKF	2.27	1.43	2.43	0.10	0.10	0.10	0.40	0.53	3.09
	GC-MCUKF	2.27	1.43	2.43	0.10	0.10	0.10	0.40	0.53	3.09
	SNN-MCUKF	2.16	1.43	2.43	0.09	0.08	0.06	0.39	0.44	2.46
Seg4	EKF	2.31	0.91	2.10	0.10	0.10	0.04	0.63	0.66	14.19
	UKF	2.31	0.91	2.04	0.07	0.09	0.04	0.54	0.47	2.97
	G-MCUKF	2.31	0.91	2.04	0.07	0.09	0.04	0.54	0.47	2.97
	C-MCUKF	2.31	0.91	2.04	0.09	0.09	0.07	0.54	0.47	3.52
	GC-MCUKF	2.31	0.91	2.04	0.09	0.09	0.07	0.53	0.46	3.52
	SNN-MCUKF	2.29	0.90	2.04	0.07	0.08	0.03	0.53	0.46	2.50

Table 3 Comparison of peak and 95% errors

Method	Pos Peak [m]	Pos 95% [m]	Vel Peak [m/s]	Vel 95% [m/s]	Att Peak [deg]	Att 95% [deg]
EKF	4.29	4.06	0.47	0.24	0.14	0.13
UKF	4.28	4.06	0.47	0.24	0.15	0.13
G-MCUKF	4.29	4.06	0.47	0.24	0.15	0.13
C-MCUKF	4.29	4.06	0.48	0.27	0.29	0.24
GC-MCUKF	4.29	4.06	0.47	0.24	0.15	0.13
SNN-MCUKF	**3.93**	**3.78**	**0.47**	**0.24**	**0.14**	**0.11**

Table 2 provides a detailed quantitative comparison of the RMSE for position, velocity, and attitude across the four distinct segments. In the relatively benign environments of Segments 1 and 2, all filters exhibited comparable performance, indicating that the proposed SNN-MCUKF maintains nominal accuracy without degradation in standard conditions. However, a marked disparity is observed in Segment 3, where severe multipath signals were induced. The conventional EKF showed catastrophic divergence with a Yaw RMSE of 13.58 degrees. Meanwhile, the robust filters with fixed kernel bandwidths (C-MCUKF and GC-MCUKF) exhibited moderate performance degradation with a Yaw RMSE of 3.09 degrees. In contrast, the proposed SNN-MCUKF maintained a Yaw RMSE of 2.46 degrees. While this does not imply a complete elimination of errors, it demonstrates a reasonable improvement over the fixed-bandwidth approaches and confirms that the SNN-based adaptive mechanism effectively contributes to maintaining estimation stability within a manageable range compared to the standard EKF.

The stability of the proposed method is further evidenced by the peak and 95th percentile errors for the entire trajectory, as presented in Table 3. The proposed SNN-MCUKF achieved a peak position error of 3.93 m and a 95th percentile error of 3.78 m. This 95th percentile metric indicates that the position error remained below 3.78 m for 95% of the total operation time, demonstrating consistency compared to conventional robust filters, which exhibited peak errors exceeding 4.28 m. Furthermore, in attitude estimation, the SNN-MCUKF minimized the peak error to 0.14 degrees and the 95th percentile error to 0.11 degrees, effectively minimizing long-term yaw drift. In terms of computational efficiency (MATLAB 2023b, Intel i5-10600KF), the processing time was approximately 1.7 seconds, nearly identical to the conventional MCUKF. This indicates that the SNN-based adaptive kernel adjustment improves robustness without increasing the computational burden.

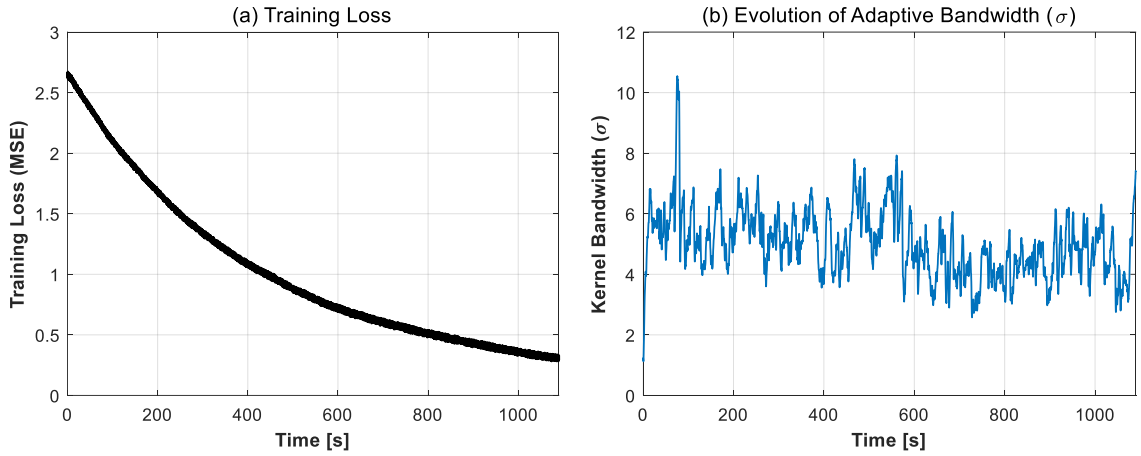

Fig. 5 Online learning performance: (a) training loss and (b) adaptive kernel bandwidth.

Figure 5 illustrates the online learning stability and the adaptive capability of the proposed method. As shown in Fig. 5(a), the training loss decreases monotonically and converges near zero, confirming that the SNN parameters are stably updated without divergence throughout the operation. Consequently, the learned kernel bandwidth in Fig. 5(b) exhibits dynamic fluctuations. This indicates that the algorithm adjusts the kernel bandwidth in real-time to mitigate the effects of non-Gaussian noise and outliers, thereby maintaining estimation robustness.

5 Conclusion

This paper proposed a robust MCUKF framework integrated with a Shallow Neural Network (SNN) for online adaptive kernel bandwidth estimation, addressing the limitations of fixed-bandwidth approaches in non-Gaussian urban navigation. Unlike conventional adaptive methods that rely on heuristic rules or simple residual-based thresholds, the proposed SNN-based approach learns the non-linear mapping between the innovation sequence and the optimal kernel bandwidth. This data-driven mechanism allows the filter to autonomously adjust the bandwidth without heavy reliance on manual parameter tuning, thereby significantly reducing sensitivity to empirical thresholds and enabling rapid response to sudden environmental changes. Experimental results in real-world urban environments demonstrated that the SNN-MCUKF outperforms standard EKF, UKF, and conventional MCUKF-series filters in terms of tracking stability and outlier suppression, particularly effectively reducing long-term drift in Yaw estimation. The computational load was confirmed to be comparable to that of the conventional MCUKF, verifying its feasibility for real-time applications. Future work will extend this framework to Deep Neural Networks (DNN) or time-series models to capture temporal dependencies in error characteristics for enhanced adaptability.

Acknowledgments

This paper was supported by the Korean National Research Fund (NRF-2022R1A2C1005237) and Unmanned Vehicles Core Technology Research and Development Program (NRF-2020M3C1C1A01086408).

This research was supported by Korea Research Institute for defense Technology planning and advancement (KRIT) grant funded by the Korea government (DAPA (Defense Acquisition Program Administration)) (No. KRIT-CT-22-030, Reusable Unmanned Space Vehicle Research Center, 2025) and Specialized Workforce Development Program for and expert-to-be building up in the Urban Air Mobility (Project No. P0020647).

Declaration of Use of Artificial Intelligence

We used AI only for translation. Specifically, we employed [Chat-GPT-4] to translate portions of the manuscript between Korean and English. All machine-translated text was carefully reviewed and edited by the authors to ensure accuracy, clarity, and scientific integrity. No generative AI was used to draft, summarize, or expand the scientific content, and the authors remain fully responsible for all content.

References

[1] C. Xue, P. A. Psimoulis, and X. Meng, “Feasibility analysis of the performance of low-cost GNSS receivers in monitoring dynamic motion,” Measurement, vol. 202, 2022, Art. no. 111819. doi: 10.1016/j.measurement.2022.111819

[2] H. Yuan et al., “An extended robust estimation method considering the multipath effects in GNSS real-time kinematic positioning,” IEEE Trans. Instrum. Meas., vol. 71, 2022, pp. 1–9. doi: 10.1109/TIM.2022.3193967

[3] N. El-Sheimy, K.-W. Chiang, and A. Noureldin, "The utilization of artificial neural networks for multisensor system integration in navigation and positioning instruments," IEEE Trans. Instrum. Meas., vol. 55, no. 5, pp. 1606–1615, 2006. doi: 10.1109/TIM.2006.881033

[4] W. Jiang, Y. Li, and C. Rizos, "A multisensor navigation system based on an adaptive fault-tolerant GOF algorithm," IEEE Trans. Intell. Transp. Syst., vol. 18, no. 1, pp. 103–113, 2016. doi: 10.1109/TITS.2016.2562700

[5] M. Peretic et al., "Statistical analysis of GNSS multipath errors in urban canyons," in Proc. IEEE/ION Position, Location and Navigation Symp. (PLANS), Salt Lake City, UT, USA, 2025, pp. 1216–1225. doi: 10.1109/PLANS61210.2025.11028411

[6] B. Chen et al., "Maximum correntropy Kalman filter," Automatica, vol. 76, pp. 70–77, 2017. doi: 10.1016/j.automatica.2016.10.004

[7] X. Liu et al., "Maximum correntropy unscented filter," Int. J. Syst. Sci., vol. 48, no. 8, pp. 1607–1615, 2017. doi: 10.1080/00207721.2016.1277407

[8] J. Wang et al., "Cauchy kernel-based maximum correntropy Kalman filter," Int. J. Syst. Sci., vol. 51, no. 16, pp. 3523–3538, 2020. doi: 10.1080/00207721.2020.1817614

[9] Q. Ge, X. Bai, and P. Zeng, "Gaussian-Cauchy mixture kernel function based maximum correntropy criterion Kalman filter for linear non-Gaussian systems," IEEE Trans. Signal Process., vol. 73, pp. 158–172, 2025. doi: 10.1109/TSP.2024.3479723

[10] K. Feng et al., "Robust cubature Kalman filter for SINS/GPS integrated navigation systems with unknown noise statistics," IEEE Access, vol. 9, pp. 9101–9116, 2020. doi: 10.1109/ACCESS.2020.3036423

[11] Y. Xu et al., "Dual free-size LS-SVM assisted maximum correntropy Kalman filtering for seamless INS-based integrated drone localization," IEEE Trans. Ind. Electron., vol. 71, no. 8, pp. 9845–9854, 2023. doi: 10.1109/TIE.2023.3323737

[12] N. H. Singh and K. Thongam, "Mobile robot navigation using MLP-BP approaches in dynamic environments," Arab. J. Sci. Eng., vol. 43, no. 12, pp. 8013–8028, 2018. doi: 10.1007/s13369-018-3267-2