Universidad de Málaga

Escuela Técnica Superior de Ingeniería de Telecomunicación

Programa de Doctorado en Ingeniería de Telecomunicación

TESIS DOCTORAL

Multi-Sensor Data Fusion for Underwater Positioning

Autor:

Nabil Shaukat

Director:

Pablo Otero Roth

Málaga, 2022

UNIVERSIDAD DE MÁLAGA

AUTOR: Nabil Shaukat

https://orcid.org/0000-0002-8363-349X

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

Nabil SHAUKAT

Estudiante del programa de doctorado en Ingeniería de Telecomunicación de la Universidad de Málaga, autor de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada "MULTI-SENSOR DATA FUSION FOR UNDERWATER POSITIONING",

Realizada bajo la tutorización y dirección de Pablo OTERO ROTH,

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 27 de noviembre de 2021,

Fdo.: Nabil SHAUKAT

VºBº Pablo OTERO

This page intentionally left blank

UNIVERSIDAD DE MÁLAGA

AUTORIZACIÓN DE LOS DIRECTORES DE TESIS DOCTORAL

El alumno del Programa de Doctorado en Ingeniería de Telecomunicación, NABIL SHAUKAT, con pasaporte nº PAK AL5195635, es primer autor de las siguientes publicaciones en revistas indexadas en los Journal Citation Reports (JCR) del Web of Science (WoS):

- **N. Shaukat,** M. Moinuddin, and P. Otero, "Underwater Vehicle Positioning by Correntropy-Based Fuzzy Multi-Sensor Fusion," Sensors, vol. 21, no. 18, p. 6165, Sep. 2021. DOI: https://doi.org/10.3390/s21186165
- **N. Shaukat**, A. Ali, M. Javed Iqbal, M. Moinuddin, and P. Otero, "Multi-Sensor Fusion for Underwater Vehicle Localization by Augmentation of RBF Neural Network and Error-State Kalman Filter," Sensors, vol. 21, no. 4, p. 1149, Feb. 2021. DOI: https://doi.org/10.3390/s21041149

Además, el Sr. Shaukat es el primer autor de las publicaciones de una conferencia internacional:

- **N. Shauka**t, A. Ali, M. Moinuddin and P. Otero, "Underwater Vehicle Localization by Hybridization of Indirect Kalman Filter and Neural Network," 2021 7th International Conference on Mechatronics and Robotics Engineering (ICMRE), 2021, pp. 111-115, DOI: 10.1109/ICMRE51691.2021.9384844
- **N. Shaukat** and P. Otero, "Underwater Vehicle Positioning by Fuzzy and Neural Adaptive Kalman Sensor Fusion," OCEANS 2021: San Diego – Porto, Sep. 2021, doi: 10.23919/oceans44145.2021.9705963.

Estas publicaciones avalan su tesis doctoral y ninguna otra tesis.

Por todo ello, su tutor y director de tesis Pablo Otero Roth autoriza al Sr. Shaukat a depositar su tesis doctoral ante las Autoridades académicas de la Universidad de Málaga.

En Málaga, a 27 de noviembre de 2021.

Fdo: Pablo Otero

This page intentionally left blank

Abstract

There are three fundamental questions to answer in solving underwater navigational problems: “where the underwater vehicle is?”, “where is the underwater vehicle is going?” and “how the underwater vehicle gets there?”. The thesis focuses on finding the answer to the question "where in the ocean underwater vehicle is?" which is also known as positioning or localization of underwater autonomous vehicles, an increasingly important question for the blue economy. A wide range of underwater tasks can now be accomplished with underwater vehicles. Several industries that rely on blue economies require detailed maps of the seafloor to conduct mining, drilling, and for other applications, which is possible through underwater vehicles for conducting operations safely and reliably. The unavailability of the Global Navigation Satellite System (GNSS) signals along with the complexity of the underwater environment makes underwater navigation and positioning particularly challenging compared to other unmanned autonomous systems. Specifically, this thesis focuses on the development of multi-sensor fusion methods for improving the position estimation of underwater vehicles based on the hybridization of neural networks, fuzzy and Bayesian filtering.

The Kalman filter variants are widely used in underwater multi-sensor fusion applications for positioning and navigation. Such multi-sensor integration approaches have a number of limitations, such as the process modeling dependencies, reliance on prior state knowledge, dependency on sensors accuracy and precision, and linearization error. In the case of Extended Kalman Filter (EKF), these filters use a first-order approximation for linearization, which results in decreased estimation accuracy under the conditions of high nonlinearity. A key part of this dissertation addresses the research question: how Artificial Neural Networks (ANNs) can help address the shortcomings of Kalman filtering in underwater environments. We present a novel multi-sensor fusion method for an underwater method that improves the accuracy of state estimation by the augmentation of the Radial Basis Function Neural Network (RBFNN) with Kalman filter. The equations of weights and centers of the RBFNN are derived by minimizing the estimated Mean Squared Error (MSE) using the steepest descent optimization approach. We compare our method with state-of-the-art methods under different sensor failure conditions. It performs well under noisy conditions, and only slight degradation occurs when sensor measurements are lost.

This thesis also contributes to improving the performance of underwater vehicle navigation in the presence of nonlinearity and non-Gaussian outliers by taking advantage of the correntropy strengths for improving the fuzzy Kalman filter. The objective is to enhance performance using fuzzy logic, which has the advantage of dealing with nonlinearity through expert knowledge, using correntropy to handle non-Gaussian outliers and Kalman filter for prediction of states with minimum error variance. This work also introduces new metrics based

on correntropy, which rely on high-order moments to improve covariance matching. The proposed method compares theoretical and actual covariance statistically with two new metrics, Degree of Similarity (DOS) and Degree of Convergence (DOC). Moreover, the fuzzy logic and correntropy-based similarity measures together provide greater robustness to the large outliers that are commonly present in underwater environments. Because of their particular benefits to the heavier tail underwater vehicle stochastic data, Correntropy provides a similarity metric based on kernels, notably the Gaussian and Versoria kernels. Through numerical simulation analysis, it is found that the proposed method is appreciably more robust and offers a higher level of performance by incorporating the novel correntropy-based metrics to mitigate the effects of outliers for underwater vehicle position estimation.

Finally, this PhD thesis has focused on developing a multi-sensor fusion architecture by exploiting the advantages of neural networks, fuzzy logic, and correntropy and combining them to improve the performance of an underwater navigation system.

Resumen

Sensores para posicionamiento submarino

El problema de la navegación de un vehículo submarino puede formarlizarse mediante tres preguntas: posición, destino y trayectoria, es decir,"¿en qué lugar del océano está?", "¿a dónde va?" y "¿cómo puede llegar hasta allí?". Esta tesis doctoral se centra en encontrar la respuesta a la primera de esas tres preguntas, lo que también se conoce como posicionamiento o localización de los vehículos autónomos submarinos, una cuestión cada vez más importante para la economía azul.

La existencia y disponibilidad de los sistemas GNSS (*Global Navigation Satellite System*), como GPS o Galileo, junto con los muy económicos receptores, pueden dar la impresión de que el problema de la navegación submarina está casi resuelto. Sin embargo, las señales GNSS son inaccesibles en escenarios submarinos debido a la casi imposible propagación de las ondas electromagnéticas de frecuencias usadas por esos sistemas, por lo que un reto importante para los robots submarinos es determinar su propia posición en relación con el marco de referencia.

Por otro lado, los sensores para vehículos submarinos disponibles en el mercado y que podrían usarse para su posicionamiento tienen sus propias deficiencias. Por regla general, los sensores más caros son los más precisos. Debido a estos factores, la determinación de la posición del vehículo bajo el agua es un proceso complicado y costoso. La fusión de datos de múltiples sensores ofrece una alternativa eficiente y eficaz para lograr este objetivo. Sin embargo, el desarrollo de nuevos métodos de fusión de datos multisensor y su aplicación en escenarios subacuáticos están aún por explorar, ya que los enfoques más recientes se han aplicado principalmente a entornos terrestres y aéreos. Se repasan a continuación los sensores que se han considerado en esta investigación.

Los sistemas de navegación inercial (*Inertial Navigation System*, INS) calculan la aceleración y la velocidad angular mediante una unidad de medición inercial (*Inertial Measurement Unit*, IMU). Una IMU de tres ejes se compone de tres acelerómetros, tres giroscopios y, generalmente, tres magnetómetros. La navegación por estima (*Dead Reckoning*, DR) mediante INS es un método relativamente sencillo, sin GNSS, que utiliza la velocidad y el tiempo de viaje para estimar la posición actual de un vehículo en comparación con una posición anterior. Este método se limita a la posición relativa con respecto a una posición inicial. En otras palabras, la DR puede utilizarse para responder a cuánto se ha movido un vehículo, pero no es una herramienta útil para localizar dónde está el vehículo. Un inconveniente importante de la navegación por estima es que, debido a la incertidumbre de las mediciones del sensor, los errores asociados a la estimación de la posición aumentan a medida que se incrementan el tiempo y la distancia. Esto provoca la integración de medidas inciertas de aceleración o de velocidad y da lugar a una deriva en la estimación de la posición. En la práctica, las lecturas de la IMU son excelentes para estimar los cambios de medición a corto plazo, pero la deriva las hace inexactas a medida que transcurre el tiempo. El uso de la fusión de datos para combinar la DR con los datos proporcionados con otros sensores o mediante otras técnicas es un área de investigación abierta para una navegación submarina absoluta, precisa y sin deriva.

El registro de velocidad Doppler (*Doppler Velocity Log*, DVL) es uno de los sensores de velocidad más utilizados en los vehículos submarinos. El efecto Doppler puede utilizarse para calcular

la velocidad relativa del vehículo. Para calcular la velocidad en las tres direcciones, el DVL utiliza múltiples haces de los transductores con diferentes ángulos. Transmite múltiples haces de ondas acústicas al fondo y detecta sus reflejos en el suelo marino, pero a veces, en las profundidades del océano, el DVL no consigue mantener el seguimiento del fondo, lo que perturba la precisión de la medición. Para evitar este problema, los vehículos submarinos utilizan un método de seguimiento y medición denominado modo relativo al agua (*water-track mode*) para determinar su posición en relación con la columna de agua. En entornos con fuertes corrientes submarinas, los vehículos que utilizan el seguimiento del agua son vulnerables a las derivas de velocidad y posición causadas por los movimientos no contabilizados de la columna de agua. Un problema adicional es que el DVL sólo mide las velocidades lineales cuando se sitúa en el centro de flotabilidad del vehículo, lo que no siempre es posible debido a las limitaciones mecánicas. Por lo tanto, el movimiento de rotación es detectado por el DVL como movimiento lineal, lo que provoca un error de medida. Este efecto recibe el nombre de alineación vehículo-sensor y es una de las fuentes de error más comunes de estos sensores. Debido a esto, los métodos de fusión de sensores DVL e INS son actualmente áreas de investigación abiertas en la navegación submarina.

Posicionamiento acústico submarino

El posicionamiento submarino se realiza mayormente por medio de sistemas acústicos, ya que el sonido se propaga en el agua mejor que ondas de otra naturaleza. Basándose en el tiempo de llegada de la señal acústica o en las mediciones de la diferencia de tiempo de llegada de la señal, estos sistemas calculan distancias o rumbos relativos a posiciones conocidas. Hay dos enfoques principales que utilizan balizas acústicas, que suelen instalarse en el fondo marino o en los buques de superficie. Los sistemas LBL (*Long Base Line*) se encuentran entre los sistemas de posicionamiento acústico submarino más utilizados. Las líneas de base son los segmentos entre balizas, que se considera larga en el caso de los LBL. Una baliza se fija en el fondo del mar con una ubicación conocida. Normalmente, los vehículos submarinos llevan transpondedores para interrogar a las balizas. Las distancias entre un vehículo y sus balizas se calculan midiendo el tiempo de viaje en ambos sentidos. Los sistemas LBL suelen tener prestaciones limitadas. Los alcances típicos son de unos pocos kilómetros. El tiempo de consulta suele ser largo y las actualizaciones tardan varios segundos. El rendimiento de LBL también se ve afectado negativamente por las imprecisiones en la calibración y tiene un alto coste de despliegue. Por otra parte, el perfil de la velocidad del sonido y los efectos de la propagación multicamino debida la reflexión de la onda acústica en la superficie del agua y el el fondo, en el caso de aguas someras, provocan que el comportamiento de los sistemas LBL sea no lineal. La estimación de la posición del vehículo submarino en tiempo real utilizando datos de medidas de instrumentos no lineales constituye también un reto de investigación.

El sistema USBL (*Ultra-Short Baseline*) es uno de los más utilizados para los vehículos submarinos por su escasa complejidad. No necesita que se desplieguen balizas en el fondo marino, lo que ahorra costes. El seguimiento se realiza con un conjunto de transpondedores acústicos compuesto por cuatro o más transpondedores acústicos montados en disposiciones geométricas determinadas, normalmente en la obra viva de un buque de superficie, junto con un interrogador (*pinger*) instalado en el vehículo submarino. Basándose en la dirección de llegada (*Direction of Arrival*, DoA) y el tiempo de llegada (*Time of Arrival*, ToA) en el array de recepción, pueden determinarse la dirección y la distancia

entre el transductor USBL y el *pinger* a bordo del vehículo submarino. Para determinar con precisión la posición absoluta del vehículo, es necesario conocer la posición y actitud reales del transductor USBL. La posición y actitud reales del transductor USBL se obtienen utilizando un INS y un receptor GNSS en el buque. La precisión del posicionamiento del sistema USBL está influida por el error del ángulo de instalación y la calibración. Cuando se rastrean varios transductores al mismo tiempo, es posible alcanzar un nivel aceptable de precisión de posición relativa. La complejidad de la trayectoria de los vehículos submarinos es el problema más difícil de la comunicación USBL, ya que se dan la propagación multicamino, fluctuaciones en la velocidad del sonido y alinealidades de las medidas.

En la actualidad se investiga para conseguir un sistema similar al GNSS que opere en el mundo submarino, aunque estas investigaciones se encuentrean en fase embrionaria. Los sistemas típicos de posicionamiento acústico subacuático cubren un área relativamente pequeña y no están bien escalados para soportar la navegación de múltiples vehículos. Además, consumen mucha energía y producen cantidades considerables de contaminación acústica. El sistema de la boya inteligente de posicionamiento (*GPS Intelligent Buoy's*, GIB) consiste en un conjunto de boyas de superficie con receptores GPS, instrumentos de posicionamiento acústico sumergidos y módems. Las boyas registran un pinger instalado a bordo del vehículo submarino, que se sincroniza con la hora del reloj atómico antes del despliegue del sistema, y se envía en tiempo real a través del enlace de radio a una unidad de control en el buque de apoyo, donde se procesan los datos y se calcula una posición fija. Como la información de posición sólo es accesible en la unidad de control, a diferencia del sistema LBL, el sistema puede emplearse para la navegación directamente [9]. A diferencia de la transmisión de ida y vuelta del LBL y el USBL, el GIB utiliza una transmisión de señales acústicas unidireccional desde los vehículos submarinos a las boyas, lo que lo hace menos susceptible a las perturbaciones.

La batimetría se utiliza para crear un perfil del fondo y, a continuación, se determina la estimación de la posición comparándola con una base de datos de cartas náuticas. Para que estos algoritmos de navegación funcionen correctamente, es necesario que haya una cierta variación del fondo. Los métodos de búsqueda de áreas y los métodos de gradiente son dos tipos principales de métodos de navegación por fondo. Los métodos de área de búsqueda realizan una búsqueda en la carta, o en partes de la carta, para encontrar el perfil que mejor se ajusta al perfil del terreno medido. Los métodos basados en el gradiente describen los cambios locales del fondo cerca de la posición inicial. Los métodos basados en el gradiente tienen la desventaja de que requieren una estimación mucho más precisa de la posición inicial. Cuando la incertidumbre en la posición inicial es alta, los métodos divergen. Los sistemas de posicionamiento basados en el fondo se enfrentan a dos retos principales, que son la carga computacional y los requisitos de memoria. Sin embargo, biólogos marinos siguen estudiando cómo las criaturas marinas navegan a través de entornos sin rasgos característicos con una precisión milimétrica, a menudo a través de largas distancias, utilizando capacidades de detección limitadas. Los ingenieros y científicos especializados en robótica intentan aplicar estos conceptos a los problemas de navegación de los vehículos submarinos autónomos.

Fusión de datos multisensor

Los seres humanos y los animales combinan múltiples informaciones cuando sus sentidos son estimulados por señales adecuadas. Por ejemplo, la identificación de una persona es más fácil cuando la voz se une a la información visual. Del concepto anterior se deriva la Fusión de Datos de Múltiples

Sensores (*Multi-Sensor Data Fusion*, MSDF) o, simplemente, Fusión de Múltiples Sensores (*Multi-Sensor Fusion*, MSF), que es la idea de combinar las mediciones de varios sensores de manera que se aproveche el efecto de la operación simultánea para para que las medidas proporcionada por el conjunto sean más exactas que la proporcionadas por cada uno de ellos individualmente. Desarrollado inicialmente para aplicaciones militares, el Grupo de Trabajo de Fusión de Datos de los Directores Conjuntos de Laboratorios ha elaborado un modelo funcional de fusión de datos multisensor, denominado JDL. En general, la MSF implica la integración de sensores, la estimación y el procesamiento de datos complejos para la toma de decisiones. El corazón de este proceso es una estimación precisa. El filtro de Kalman se considera una herramienta eficaz para la fusión multisensor en un entorno ruidoso. La clave de su éxito es su sencilla formulación de espacio de estados y su estructura recursiva de predicción-corrección. Además, estos algoritmos son fácilmente implementables en microcontroladores integrados en tiempo real. Estos algoritmos integran directamente todas las medidas proporcionadas por los sensores y calculan una estimación óptima de los estados. Para que el filtro de Kalman realice una estimación óptima, el sistema dinámico debe ser lineal y todas los ruidos, blancos gaussianos. Sin embargo, esto rara vez ocurre en la práctica. Para la estimación no lineal, el filtro de Kalman extendido (*Extended Kalman Filter*, EKF), el filtro de Kalman de estado de error (*Error State Kalman Filter*, ESKF), el filtro de Kalman-Uhlmann (*Unscented Kalman Filter*, UKF) y los filtros de partículas (*Particle Filters*) son los más utilizados para determinar la posición del vehículo submarino. El EKF no es apropiado para problemas que contengan una no linealidad significativa, ya que se basa en una aproximación lineal local. El EKF requiere que las funciones del modelo del proceso y de la medida sean diferenciables. Aparte de eso, algunas matrices jacobianas pueden ser difíciles de calcular, lo que hace que el uso de EKF sea complicado y potencialmente propenso a errores. El ESKF puede utilizarse para estimar el error de actitud, velocidad y posición, así como el sesgo de las medidas de la IMU y de los sistemas acústicos de posición. El ESKF proporciona una mejor estabilidad numérica en comparación con el EKF. Los filtros de Kalman-Uhlmann (UKF) estiman el resultado de aplicar una determinada transformación no lineal a una distribución de probabilidad que se caracteriza sólo en términos de un conjunto finito de estadísticos. Los UKF implican mayor complejidad de implementación y un aumento de la carga computacional. Además de las distintas variantes de los filtros de Kalman, en las aplicaciones MSF también se utilizan redes neuronales artificiales y lógica difusa.

Para que los vehículos submarinos funcionen de forma autónoma, deben percibir el entorno espacial y determinar su propia posición a partir de la medida del sensor o conjunto de sensores. Por ejemplo, la cartografía submarina del entorno es fundamental en las operaciones de prospección minera y de hidrocarburos. Dado que la creación de mapas depende en gran medida de las medidas de los sensores y de la localización de los vehículos submarinos, la determinación de la posición precisa del vehículo desempeña un papel crucial en la elaboración de mapas precisos y consistentes para el funcionamiento autónomo. Debido al ruido, la deriva y el sesgo que afectan negativamente a la medida de los sensores, los mapas y las operaciones autónomas se ven comprometidos. Además, no existe una representación matemática precisa de un vehículo en un entorno submarino. Dado que los entornos submarinos son complejos, muchos efectos quedan sin modelar; esto también se conoce como desajuste del modelo debido a la dinámica no modelada del vehículo y a las variaciones de los parámetros submarinos. Esta investigación está motivada principalmente por la necesidad de una solución MSF más precisa y fiable y aporta la idea de mejorar la estimación de la posición del vehículo submarino.

Combinación de redes neuronales y filtros Kalman

El primer trabajo de investigación que se presenta en esta tesis trata de llenar el vacío arriba mencionado mediante el desarrollo de una novedosa arquitectura de fusión multisensor que combina redes neuronales y filtros ESKF, para la aplicación de navegación submarina. Ambos algoritmos pueden combinarse para mejorar la capacidad de los vehículos submarinos para navegar en entornos sin señal GNSS. Un algoritmo de red neuronal se entrena utilizando la información del modelo de espacio de estado de un vehículo submarino, ya que esta información ayuda a mejorar la estimación. Este método elimina el inconveniente de sustituir todos los bloques del filtro Kalman por redes neuronales, ya que los sustituye todos por un solo conjunto de redes neuronales que actúan como una única y gran caja negra. La caja negra oculta por completo cómo la red neuronal combina las entradas de estado, las mediciones de los sensores y los estados anteriores.

Las modificaciones propuestas mejoran el rendimiento del ESKF y aprovechan las ventajas de la red neuronal de tipo RBF (Radial Basis Function, RBF). Laa redes RBF pueden aproximar cualquier función no lineal y también se conocen como aproximadores de funciones universales. En otras palabras, se basa en superposiciones de funciones de base localizadas que pueden aproximar la no linealidad. Aprenden considerablemente más rápido que las redes de retropropagación. Debido a la naturaleza de las unidades ocultas de las RBF, son menos propensas a tener dificultades con las señales ruidosas. La determinación del número de neuronas en la capa oculta de la red neuronal RBF es importante, ya que afecta a la complejidad y la generalización de la red. Si no hay suficientes neuronas en la capa oculta, la red RBF no puede aprender rápida y adecuadamente; si hay demasiadas neuronas, la generalización puede ser deficiente y puede producirse un sobreajuste. El centro RBF, su anchura y los pesos lineales de cada neurona de salida se modifican en cada iteración de un algoritmo de aprendizaje. Se ha completado la fase de entrenamiento cuando cada centro RBF está lo más cerca posible del vector de entrada y el error de salida de la red está dentro del límite objetivo. Por lo tanto, es posible expresar la aproximación de cualquier dependencia funcional entre variables como una combinación lineal del menor número posible de neuronas RBF con el peso y el centro adecuados. El diagrama de bloques de nivel superior de nuestro algoritmo de fusión propuesto se representa en la figura 1.

Como muestra la figura 1, el algoritmo toma como entrada el error de los sensores auxiliares y del INS. El algoritmo de fusión RBF-ESKF, tras el procesamiento, da una salida al INS para corregir el error y restablecer su posición. La red neuronal RBF utilizada para el procesamiento tiene una estructura básica de tres capas: capa de entrada, oculta y de salida. Además, en comparación con redes neuronales de tipo BP (*Back-Propagation Neural Network*), las variantes de RBF tienen menos carga computacional y un rápido aprendizaje en línea. Durante el entrenamiento en línea de la red RBF, los pesos y el centro se ajustan en tiempo real a medida que se dispone de datos secuenciales. La primera capa es la de entrada, que proporciona una interfaz entre los datos y la red neuronal. Los datos de la capa de entrada a la segunda capa oculta se transfieren de tal manera que el valor de salida de cada neurona oculta está inversamente relacionado con la distancia euclidiana del vector de entrada de esa neurona al centro de la neurona RBF. La tercera capa es la de salida, que tiene en cuenta los pesos y sesgos acumulados de todas las salidas de las neuronas RBF. En la literatura existen diversas variantes de RBF que dependen de la aplicación. Sin embargo, en este trabajo, utilizamos la función RBF de tipo

gaussiano. Los pesos de las neuronas en la capa oculta para la función RBF gaussiana indican el centro de la curva de distribución gaussiana simétrica. La novedad del algoritmo RBF-ESKF es que incluye la información del sistema dentro de las reglas de actualización de los pesos y del centro.

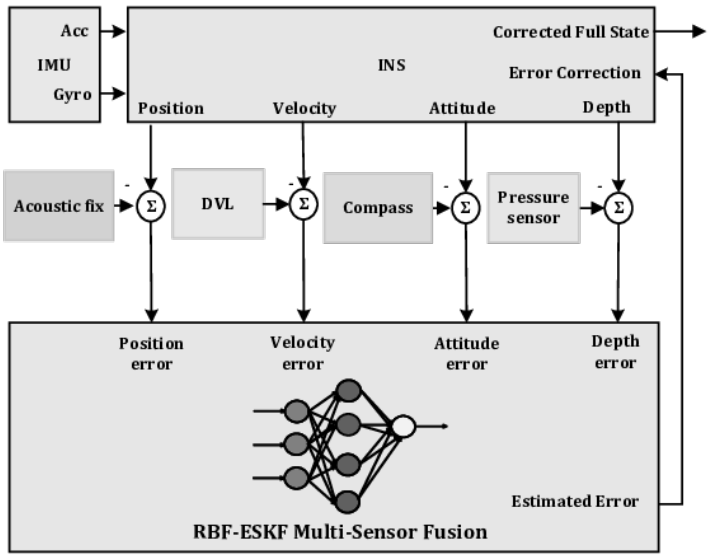

Figura 1. Diagrama de bloques de la arquitectura de navegación de fusión multisensor RBF-ESKF.

Para comparar su rendimiento, el algoritmo propuesto y el ESKF se simularon en tres escenarios realistas diferentes. Como la estructura de ESKF fue modificada por RBF en el algoritmo de fusión de datos propuesto, se escribieron funciones de bajo nivel para la simulación. Las especificaciones de ruido de los sensores utilizados en este trabajo son comparables a las de sus principales fabricantes. El objetivo fundamental de la simulación era comparar el error máximo y el error cuadrático medio (RMSE) de posición, velocidad y actitud. Se desarrollaron y simularon casos prácticos de pruebas de modo de fallo con pérdida de las medidas del DVL y del posicionamiento acústico de corta duración. Los transductores del sistema DVL emiten múltiples haces con varios ángulos. Las ondas acústicas se transmiten verticalmente y se detectan sus reflexiones en el fondo marino. Sin embargo, es posible que el DVL no mantenga el seguimiento del fondo, lo que introduce un error en la medida. Esta situación se produce al pasar por encima de zanjas o ser obstruido por la vida marina. Influye negativamente en la precisión de la posición. El fallo de comunicación del DVL puede deberse a varias razones, la más común es que el vehículo esté fuera de alcance. La segunda razón más común es que la fauna marina bloquee la comunicación. Se ha simulado esta situación para probar el algoritmo propuesto. Los resultados de la simulación se compararon con el ESKF convencional para evaluar su rendimiento. Además, para la simulación se asumió que los vehículos submarinos pueden moverse en cualquier dirección y con ángulos típicos de balance, cabezada y guiñada. Se generó una trayectoria de referencia del vehículo mediante las velocidades angulares y la aceleración. Para considerar los efectos de las variaciones aleatorias en la precisión de los algoritmos de fusión, se utilizó una simulación de Monte Carlo. La prueba consistió en numerosas ejecuciones. Dos filtros procesaron los mismos datos durante la prueba para garantizar una comparación justa. En los tres casos se utilizó la misma estructura de red

neuronal RBF para la simulación. Los pesos, centros y sigma de la RBF se inicializaron de forma aleatoria. En los tres casos, la estimación de posición, velocidad y actitud de RBF-ESKF es mejor que la de ESKF. Sin embargo, en los modos de fallo, el rendimiento de RBF-ESKF se degrada ligeramente.

El método propuesto mejoró la exactitud del posicionamiento y trató de llenar el vacío proponiendo una novedosa solución de fusión aumentada por ESKF y RBF, que se evaluó en tres casos submarinos diferentes. El rendimiento del ESKF depende en gran medida del conocimiento de los modelos del sistema y de las propiedades del ruido y se vio degradado por la no linealidad. Los errores del RBF-ESKF son menores que los del ESKF debido al aprendizaje recursivo del RBF. Además, el algoritmo de fusión basado en RBF-ESKF, con ayuda de los sensores auxiliares, fue capaz de corregir los problemas de deriva en el INS con mayor precisión.

Es conveniente señalar que el algoritmo propuesto no se limita al ámbito submarino, sino que también puede utilizarse en otras aplicaciones, como la mejora de la navegación y el seguimiento de aeronaves mediante el uso de sensores aéreos. Los vehículos terrestres autónomos son otro ámbito en el que puede emplearse este método. Además, al mejorar la respuesta del filtro de Kalman, la metodología propuesta también puede mejorar la precisión de la estimación de la actitud de los satélites.

El objetivo principal de este trabajo es aprovechar la red neuronal RBF para mejorar el rendimiento de la estimación de la ESKF convencional para la posición, la velocidad y la actitud de un vehículo submarino. Se contrastan los resultados con el ESKF en tres simulaciones distintas, mostrando que el RBF-ESKF se comporta mejor en la estimación de la posición, la velocidad y la actitud. El motivo por el que el ESKF estándar tiene un rendimiento inferior es que está diseñado empleando una aproximación lineal en la estimación de la matriz de covarianza del error, lo que resulta en una disminución de la precisión de la estimación bajo una alta no linealidad. Por lo tanto, se pierde información importante sobre la dinámica del agua subterránea debido a esta realización. Sin embargo, RBF-ESKF maneja eficazmente la no linealidad debido a su capacidad inherente de aproximación de funciones no lineales y a su capacidad de aprendizaje.

Durante la investigación también se comparó la consistencia del método propuesto en los casos en los que no hay información disponible sobre la posición a partir de la fijación acústica. Aquí, en relación con ESKF, RBF-ESKF demostró una mayor precisión. Cuando se dispone de información acústica fija, el algoritmo RBF-ESKF converge rápidamente. Además, cuando el DVL falla debido a duraciones cortas, RBF-ESKF también demuestra un menor error de estimación. En ambos casos, la red neuronal RBF se beneficia de su capacidad de aprendizaje y de su solidez frente al ruido para superar los modos de fallo.

Optimización del filtrado Kalman

La segunda contribución de la presente tesis doctoral se refiere a la mejora del filtro de Kalman adaptativo, ya que en el entorno submarino es difícil conocer el modelo del sistema y las características del ruido, que son requisitos previos para los algoritmos de fusión multisensor basados en el filtrado Kalman. Como es habitual en los modelos de procesos de sistemas, el componente determinista suele estar creado por principios cinemáticos, mientras que el elemento estocástico está representado por los ruidos resultantes de los errores de modelado y la no linealidad. Del mismo modo, las características y la fiabilidad de los sensores desempeñan un papel importante en la determinación de la parte estocástica

del modelo de medida. Las covarianzas del proceso del sistema y del ruido de medida no se conocen en un escenario práctico de navegación de vehículos submarinos. El uso de valores incorrectos compromete la estimación de la posición y la divergencia.

La lógica difusa, o borrosa, proporciona un método basado en sistemas expertos, adecuado para ajustar la covarianza del ruido. El enfoque más común para ajustar la covarianza teórica y la real ha sido utilizar la diferencia entre ambas, pero este enfoque no da resultados precisos cuando se trata de valores atípicos en la componente estocástica del modelo.

El aprendizaje teórico de la información (*Information-Theoretic Learning*, ITL) se ha utilizado con éxito para comprobar la similitud no lineal basada en la correntropía, especialmente con valores atípicos en entornos ruidosos. Como medida de similitud no lineal, la correntropía muestra la proximidad de dos variables aleatorias con un determinado tamaño de núcleo. Además, puede preservar las características no lineales, así como los momentos de alto orden. Sin embargo, los trabajos existentes basados en la correntropía carecen de las ventajas de utilizar la lógica difusa. Estas ventajas de la lógica difusa y la correntropía nos inspiró para proponer nuevos algoritmos y nos llevaron a responder a una importante pregunta de investigación: ¿Pueden utilizarse los puntos fuertes de la correntropía para mejorar la exactitud del posicionamiento de vehículos submarinos en presencia de no linealidad y valores atípicos?

El método propuesto aprovecha las ventajas del uso de la correntropía y responde a la pregunta del párrafo anterior. El problema se aborda con el método propuesto, ya que no hay ningún método de fusión multisensor publicado anteriormente, hasta donde alcanza el conocimiento de los autores, que mejore los beneficios de la correntropía, el filtrado difuso y el de Kalman aplicados a la ingeniería de vehículos submarinos autónomos. Su objetivo es mejorar el rendimiento mediante el uso de la lógica difusa, que tiene la ventaja de manejar heurísticamente la no linealidad, la correntropía para el manejo robusto de un valor atípico no gaussiano y el filtro de Kalman para el procesamiento de la varianza de error mínimo en tiempo real. Llamamos a este algoritmo FC-MSF, donde FC significa Correntropía Difusa (*Fuzzy Correntropy*) y MSF significa Fusión de Datos Multi-Sensor (*Multi-Sensor data Fusion*). Otra contribución destacable de este trabajo es la introducción de nuevas métricas basadas en la correntropía, que utilizan momentos de alto orden para mejorar la adaptación de la coincidencia de covarianza. Proponemos estas métricas como Grado de Similitud (*Degree of Similarity*, DOS) y Grado de Convergencia (*Degree of Convergence*, DOC) que comparan estadísticamente la covarianza teórica y la real, lo que no ocurría en los anteriormente usados Grado de Desajuste (*Degree of Mismatch,* DOM) y Grado de Divergencia (*Degree of Divergence*, DOD). Además, las medidas de similitud basadas en la lógica difusa y en la correntropía proporcionan una mayor solidez frente a los grandes valores atípicos, que suelen estar presentes en las mediciones de posición acústica subacuática y en las mediciones de velocidad mediante DVL.

La correntropía proporciona una medida de similitud basada en *kernels*, concretamente los kernels gaussianos y de Versoria, debido a sus claras ventajas para los datos estocásticos de los vehículos submarinos que presentan distribuciones con colas más significativas. Un estudio de simulación ha demostrado el rendimiento superior del algoritmo FC-MSF propuesto y ha validado que la nueva métrica basada en la correntropía mejora la exactitud del posicionamiento submarino en presencia de valores atípicos. Es importante destacar que la técnica sugerida se evalúa inyectando el

valor atípico como un ruido impulsivo no gaussiano, que es un método utilizado por muchos investigadores para evaluar la robustez del filtro. Posteriormente, se realiza un análisis comparativo que permite a los autores evaluar la precisión del método sugerido. Las simulaciones de Monte Carlo indican que el enfoque es técnicamente viable y tiene el potencial de producir resultados positivos en un entorno submarino difícil.

Las modificaciones propuestas mejoran el rendimiento de la fusión multisensor basada en el filtro de Kalman utilizando los puntos fuertes de la correntropía y la lógica difusa. Se ha demostrado que la lógica difusa controla los procesos no lineales utilizando expresiones lingüísticas humanas y esta capacidad se combina con los filtros de Kalman para resolver los problemas de divergencia y mejorar la precisión. La coincidencia de covarianza ha sido una técnica muy utilizada en el filtrado Kalman difuso adaptativo en investigaciones anteriores; sin embargo, no hay una forma fiable de hacer coincidir la covarianza cuando los datos tienen una no linealidad significativa y grandes valores atípicos, como es el caso de un sistema de posicionamiento de vehículos submarinos. Por lo tanto, proponemos un emparejamiento de covarianza basado en la correntropía para la entrada del sistema difuso debido a su robustez frente a los valores atípicos y el ruido no gaussiano [136]. El diagrama de bloques de alto nivel de nuestro algoritmo de fusión multisensor propuesto se representa en la figura 2.

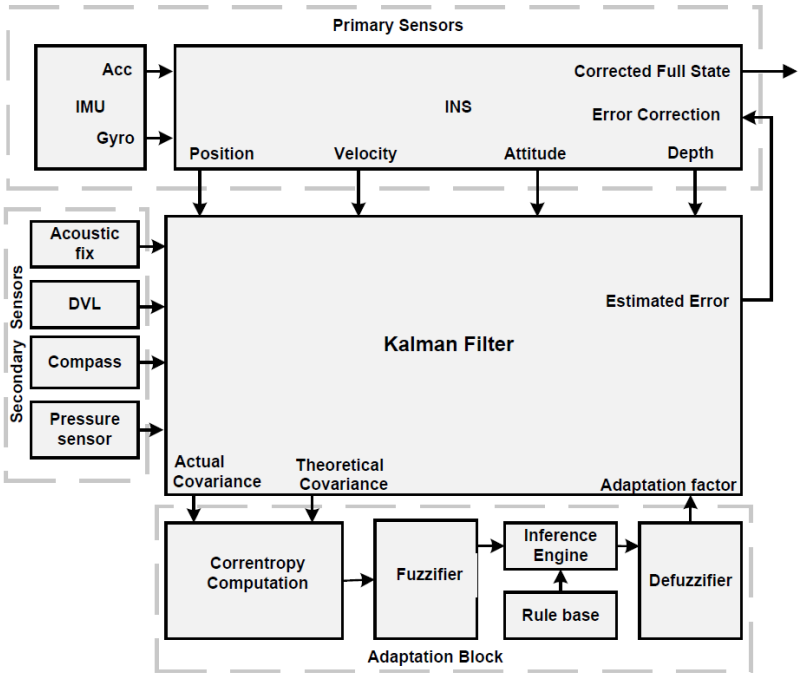

Figura 2. Diagrama de bloques de la arquitectura de navegación integrada propuesta para el vehículo

Como se muestra en el diagrama de bloques de la figura 2, los datos de los sensores primarios de la IMU y los sensores auxiliares, como el sensor de profundidad, el DVL, la posición acústica del USBL y la brújula, se introducen en el algoritmo de fusión basado en Kalman. El algoritmo de fusión trabaja sobre la dinámica de errores de los datos de entrada. El bloque de cálculo de correntropía es la

parte del bloque de adaptación que recibe la covarianza real y teórica como entrada del bloque de fusión. El bloque de correntropía proporciona una medida de similitud de 0 a 1, donde 1 significa la máxima similitud. El valor numérico del bloque de correntropía se utiliza para la "fuzzificación". Los términos lingüísticos "fuzzificados" pasan por un motor de inferencia que toma como base las reglas de expertos humanos. Para obtener el factor de adaptación, el resultado del motor de inferencia se "defuzzifica". La corrección se aplica a las matrices de covarianza que protegen la divergencia y mejoran la precisión del filtro.

La propagación multicamino es uno de los problemas más difíciles a los que se enfrentan los sistemas acústicos bajo el agua. En aguas poco profundas, la señal se propaga por las reflexiones de la superficie y el fondo, además del canal directo, lo que provoca el efecto multicamino. Por otro lado, en aguas profundas, el multicamino se crea principalmente como resultado de la flexión de los rayos debido a la refracción. Además, el perfil de la velocidad del sonido, que depende de la temperatura, la profundidad y la salinidad, es también un factor crítico que contribuye al multicamino. Además, las burbujas de aire y los animales marinos tienen su parte para los valores atípicos. La figura 4.3 de la Memoria de tesis muestra la trayectoria múltiple en aguas profundas y poco profundas.

La distribución con colas de valores representativos es una representación razonable para los datos que han sido corrompidos por los valores atípicos multitrayectoria, lo que comúnmente sucede en acústica submarina. La sustracción directa para el ajuste de la covarianza no reduce los efectos de los valores atípicos. La cola más pesada de estos valores atípicos significa que la probabilidad de resultados extremos es mayor en las colas que en la distribución normal. El núcleo de correntropía de Versoria es muy adecuado en un entorno submarino por su robustez frente a los valores atípicos. La cola de la función Versoria es más pesada que las distribuciones gaussiana y t de Student [142]. Además, el error del kernel Versoria converge más rápidamente que el kernel basado en la exponencial, aunque tiene una menor complejidad de cálculo en comparación con el kernel gaussiano.

El rendimiento del algoritmo multisensor basado en la correntropía difusa propuesto se ha comparado con la fusión multisensor basada en Kalman y con la fusión multisensor difusa mediante una simulación de Monte Carlo. Se eligió el la raíz del error cuadrático medio (*Root Mean Square Error*, RMSE) como métrica principal para la comparación porque compara el rendimiento general de la estimación del filtro durante un periodo de tiempo más largo. Para evaluar los efectos combinados de la posición y la velocidad norte, este y abajo, se calcula el RMSE medio.

La trayectoria del vehículo submarino se han simulado utilizando diferentes valores de aceleración y velocidades angulares. Todos los filtros se han probado en condiciones similares con los mismos datos de entrada para una comparación válida. Se ha supuesto que el vehículo está en el modo normal de funcionamiento, sin fallos de los sensores a bordo o fuera de él. Las velocidades iniciales de los tres ejes se han asumido nulas. Se han comparado los rendimientos de KF-MSF, Fuzzy (F-MSF) y fuzzy basado en la correntropía (FC-MSF) tras realizar simulaciones de Monte Carlo en presencia de ruido impulsivo no gaussiano. La estimación de nuestro algoritmo propuesto es significativamente mejor KF-MSF y F-MSF.

Los resultados que se presentan apoyan la hipótesis de que la fusión multisensorial difusa basada en la correntropía es menos susceptible a las perturbaciones que la F-MSF y la KF-MSF.

Además, la propiedad de robustez del kernel de correntropía permite que el FC-MSF funcione mejor cuando hay ruido impulsivo. Los resultados de estimación del FC-MSF son mucho mejores que los de posición y velocidad, ya que la correntropía tiene el potencial de capturar información de alto orden. Por el contrario, la corrección lógica difusa convencional y el KF-MSF sin correntropía tienen un impacto negativo en el rendimiento, dado que estas soluciones están expuestas al mismo ruido.

La segunda parte del estudio de investigación trata de llenar la laguna proporcionando un novedoso método de fusión adaptativa para el posicionamiento de vehículos submarinos aprovechando las ventajas de la correntropía y la difusa. El rendimiento del algoritmo propuesto se compara con las técnicas de fusión de sensores basadas en Kalman y fuzzy. Se encontró que tiene una mejor estimación de la posición y la velocidad bajo la influencia negativa del ruido impulsivo. El objetivo principal de este trabajo era aprovechar la correntropía y mejorar la técnica de coincidencia de covarianza utilizando dos nuevas métricas, el grado de similitud y el grado de convergencia. El propósito era mejorar la exactitud de la estimación de los métodos convencionales para el posicionamiento de vehículos submarinos. Las dos métricas propuestas ayudan a mejorar la exactitud de la estimación a través de un mejor emparejamiento de la covarianza teórica y la real. La técnica propuesta está diseñada para su uso en aplicaciones de cartografía del fondo marino para la exploración de los océanos.

Trabajos futuros

La futura ampliación de los métodos de posicionamiento propuestos consistiría en integrar los métodos sugeridos en un sistema de guiado, navegación y control submarino y probarlo en un entorno en tiempo real. Además, en nuestro trabajo, el tamaño de la ventana deslizante sobre la que se calcula la covarianza real de la innovación determina el rendimiento del MC-KF. Esta selección puede mejorarse determinando la influencia del parámetro en la estimación. Este estudio puede realizarse con más detalle. Otro trabajo potencial para el futuro es la selección de los parámetros críticos de la RBFNN, como el número de neuronas ocultas y la anchura del núcleo, que sólo se determinan empíricamente. La investigación puede ampliarse en el futuro para seleccionar automáticamente el número de neuronas ocultas y la anchura del núcleo durante el proceso de aprendizaje en función de la complejidad del problema. Este enfoque puede reducir el coste computacional del funcionamiento de las redes neuronales. En la segunda parte del trabajo, las funciones de membresía difusas se seleccionan para MC-KF mediante la regla general del esquema de adaptación. En el futuro podría ser posible optimizar los parámetros de la función de membresía de acuerdo con un conjunto de criterios de rendimiento para mejorar la precisión general del método de fusión multisensor. Aunque el método presentado en esta tesis fue diseñado para vehículos submarinos, en el futuro puede adaptarse fácilmente a vehículos autónomos, vehículos aéreos no tripulados y aplicaciones de minería del subsuelo. Por último, la hibridación de Kalman y la computación blanda es un nuevo enfoque para los algoritmos de navegación submarina; se necesita más investigación y desarrollo para acelerar una mayor adopción de la idea y desarrollar una arquitectura de fusión submarina artificialmente inteligente que pueda superar algunas de las limitaciones de los métodos tradicionales.

Nota: Las citas de este resumen se referencian en la bibliografía.

Acknowledgments

I would like to express my sincere appreciation and gratitude towards my thesis supervisor, Prof. Dr. Pablo Otero, for his expert guidance, support, and patience throughout the years. Dr. Otero has been an incredibly encouraging and motivating mentor throughout my research. For their support of my doctoral research, I am deeply thankful to the Institute of Oceanic Engineering Research, the University of Malaga, and the Spanish Government.

At this point, I must thank my mentor and colleague, Prof. Dr. Moinuddin, who always stood by me, guided, and helped me until the very end. Another important person, Dr. Muhammad Yousuf Irfan Zia, helped me throughout my journey; I am grateful for his kind help.

There are no words to express my deep gratitude to (late) Dr. Javier Poncela. Sadly, he passed away before my work was completed.

Most importantly, I would like to thank my parents and siblings for their support and prayers. Everything I am today, and everything I may become tomorrow, is because of my father and mother's efforts. I would like to express gratitude to my wonderful and loving wife, who never fails to motivate and encourage me.

Lastly, I am appreciative to my children for their understanding and their sacrifice of time; otherwise, I would not have accomplished my goals. I am grateful to my colleagues and friends for their help and support during my studies.

This page intentionally left blank

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ANN	Artificial Neural Network
ASV	Autonomes Surface Vehicle
AUV	Autonomous Underwater Vehicle
AKF	Adaptive Kalman Filter
BP	Backpropagation
BPNN	Backpropagation Neural Network
CRB	Cramer Rao Bound
DCM	Direction Cosine Matrix
DOF	Degree of Freedom
DR	Dead Reckoning
DVL	Doppler Velocity Log
EKF	Extended Kalman Filter
ESKF	Error State Kalman Filter
GAS	Globally Asymptotically Stable
GIB	GPS Intelligent Buoys
GLONASS	Global Orbital Navigation Satellite System
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISS	Input-to-State Stable
IVLB	Intrinsic Variance Lower Bound
KF	Kalman Filter
LBL	Long Baseline
LBL	Long Baseline
LS	Least Squares
LTI	Linear Time-Invariant.
LTV	Linear Time-Varying

MEKF	Multiplicative Extended Kalman Filter
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimator
MSF	Multi-Sensor Fusion
MSDF	Multi-Sensor Data Fusion
NED	North East Down Frame
PSD	Power Spectral Density.
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network
RD	Range Differences
RMS	Root Mean Square.
ROV	Remotely operated underwater vehicle
SBL	Short Baseline
SLAM	Simultaneous Localization and Mapping
TDOA	Time Differences of Arrival
TOA	Times of Arrival
UKF	Unscented Kalman Filter
USBL	Ultra-Short Baseline
UUV	Unmanned Underwater Vehicle
UW	Underwater
WGS-84	World Geodetic System of 1984

This page left intentionally blank

Chapter 1 : Introduction

Scientists have found that autonomous underwater robotics technology allows them to benefit from ocean resources in a way that had previously been impossible through traditional means. Ocean exploration and research have been greatly benefitted by remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) [1]. In the last few years, positioning-related technologies have seen a surge of interest and rapid development. A number of consumer products use Global Positioning System GPS receivers, compasses, and accelerometers to track their position. These developments, as well as the popularity of GPS receivers, may give the impression that the navigation problem is almost solved [2]. However, in underwater scenarios, GPS signals are inaccessible, so a major challenge for underwater vehicles is determining their own position in complex underwater environment in relation to the reference frame. In contrast to this, sensors for vehicles have their own set of shortcomings. As a general rule, more expensive sensors are the most accurate [3]. Due to these factors, determining underwater vehicle position is a complicated and expensive process. Multi-sensor data fusion provides an efficient and effective alternative way to accomplish this goal. Nevertheless, the development of new multi-sensor data fusion methods and the implementation of these methods in underwater scenarios are yet to be explored since newer approaches have mainly been applied to terrestrial and aerial environments.

Inertial navigation systems (INS) compute acceleration and angular rate using an inertial measurement unit (IMU). A three-axis IMU is made up of three accelerometers, three gyroscopes, and generally three magnetometers [3]. Dead Reckoning (DR) using INS is a relatively straightforward GPS-less method that uses velocity and travel time to estimate a vehicle's current position compared to a prior position [4]. This method is limited to relative position with respect to an initial position. In other words, DR can be used to answer how much a vehicle has moved but is not a useful tool for locating where the vehicle is. A major drawback of DR is that, because of the uncertainty in the sensor's measurements, errors associated with position estimation increase as time and distance increase. The

integration of uncertain measurement of acceleration or velocity generates drift. Using multi-sensor fusion to combine DR with other techniques is an open research area for drift-free precise absolute navigation. These multi-sensor fusion techniques have gained popularity in robotics researchers and hold great promise for precise exploration. Another method to reduce the number of errors associated with DR is to resurface periodically called Yo-Yo maneuvering in order to acquire GPS fixes; however, this method is time-consuming and energy-intensive for deep underwater missions [5].

The Doppler Velocity Log (DVL) is one of the most commonly used velocity sensors on underwater vehicles. Doppler effect can be used to calculate the relative speed of the vehicle [6]. To calculate velocity in all three directions, DVL makes use of multiple beams of the transducers with different angles. It transmits multiple beams of acoustic waves to the bottom and detects their reflections off the seafloor; however, sometimes in the deep ocean, the DVL fails to maintain bottom tracking, which disturbs the measurement accuracy. To avoid this problem, underwater vehicles use a tracking and measurement method to determine their position in relation to the water column called water-track mode [7]. In environments with strong underwater currents, vehicles that use water tracks are vulnerable to drifts in velocity and position caused by the unaccounted movements of the water column. An additional issue is that the DVL only measures linear velocities when it is placed at the center of the buoyancy of the vehicle, which is not always possible due to mechanical constraints; therefore, rotational motion is detected by the DVL as linear motion, which causes an offset in the measurement. This is also referred to as the vehicle-sensor alignment and is one of the most common sources of error in DR navigation using INS and DVL sensors. When integrated for a long period of time, even small attitude errors can lead to large position errors [8].

As sound propagates well in water, acoustic positioning is widely used for underwater positioning. Based on the measurement of the Times of Arrival (TOA) or Time Differences of Arrival (TDOA) of acoustic signals, these systems compute ranges or bearings relative to known positions. The main approaches use acoustic beacons that are usually installed on either a surface vessel or the seabed. Among the most popular and conventional underwater acoustic positioning systems are Long Baseline Systems (LBL). Distance between beacons is called a baseline, which is considered as long in the case of LBL. The beacons are fixed to the bottom of the sea with known locations. An underwater vehicle usually carries a transponder capable of interrogating beacons. In order to determine the distance between a vehicle and its beacons, the two-way travel time is used [9]. The shortcoming of the LBL system is that it can cover an area of just a few kilometers, depending on the specification. Moreover, the interrogation cycles are usually slow; most often, the updates take a number of seconds. Additionally, the performance of LBL can be affected directly by the calibration error and has a high cost of deployment.

An Ultra-Short Baseline (USBL) system position system is one of the most commonly used for underwater vehicles localization because of its low complexity [10]. It does not need a beacon to be

deployed on the seafloor, which saves cost. Tracking is accomplished with an acoustic transponder array comprised of four or more acoustic transponders mounted in convenient geometric arrangements along with a pinger installed on the vehicle. Based on the TOA and the TDOA in the receiving array, a direction and range can be determined between the USBL transducer and the Pinger on board the vehicle. To accurately determine the absolute position of the underwater vehicle, it is necessary to know the actual pose of the USBL transducer on the vessel. The actual position and attitude of the USBL transducer are obtained by using an INS and a GPS receiver. The positioning accuracy of the USBL system is influenced by the installation angle error and calibration [10]. When several pingers are tracked at the same time, it is possible to achieve an acceptable level of relative position accuracy.

There are still efforts underway to come up with a Global Satellite Navigation System (GNSS) for type systems for underwater vehicles. In contrast, typical acoustic underwater positioning systems cover a relatively small area and are not scaled well to support multiple vehicle navigation. Moreover, they are power-intensive and produce considerable amounts of acoustic pollution. The GPS Intelligent Buoy's (GIB) system consists of a set of surface buoys with GPS receivers, acoustic positioning instruments, and modems. On the support vessel, the buoys record pings that are synchronized with atomic clock time prior to deployment, and these data are sent in real-time to a control unit where a position fix is computed from the data. As position information is only accessible at the control unit, unlike with an LBL system, the system can be employed for navigation directly [11]. Whereas opposed to the round-trip transmission of LBL and USBL, GIB usually uses one-way acoustic signal transmission from underwater vehicles to buoys, which makes it less susceptible to disturbance.

Terrain measurements are used to create a terrain profile, and then the position estimate is determined by comparing it to a prior map database. A certain amount of terrain variation is required by terrain navigation algorithms to work correctly [12]. Search area methods and gradient methods are two major types of terrain navigation methods. Search area methods, perform map search, or parts of the map, to find the profile that best matches the measured terrain profile. Gradient-based methods describe the local changes in the terrain near the initial position. Gradient-based methods have the disadvantage that they require a much more accurate estimate of a position at the beginning. When the uncertainty in the starting position is too high, the methods will diverge [13]. Terrain-based positioning systems face two main challenges related to computation load and memory requirement. However, marine animal researchers are still investigating how sea creatures navigate through otherwise featureless environments with pinpoint accuracy, often over long distances. Robotics engineers and scientists are seeking to apply these concepts to autonomous underwater vehicles navigation problems [14].

Humans and animals combine multiple pieces of information when their senses are stimulated by appropriate signals. For instance, Identification of a person is made easier when a voice is coupled with visual information. Multi-Sensor Data Fusion (MSDF) or Multi-Sensor Fusion (MSF) is derived

from the above concept, which is the idea of combining the measurements from several sensors in a way that leverages the total effect to make them more effective than when they function individually [15]. Initially developed for military applications, the Joint Directors of Laboratories Data Fusion Working Group has developed a functional model of multi-sensor data fusion called JDL [16]. Overall, MSF involves the integration of sensors, estimation, and the processing of complex data for decision making. The heart of this process is an accurate estimation of states. The Bayesian techniques such as Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filters are widely used for navigation applications. The most common techniques used to determine the position of an underwater vehicle are Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). The key to success of Kalman Filter (KF) is its straightforward state-space formulation and recursive prediction-correction structure. Importantly, these algorithms are also easily implementable in real-time embedded microcontrollers. These algorithms integrate all of the measurements of these observables directly and calculate an optimal estimate of the state. Besides Kalman Filter variants, artificial neural networks and Fuzzy logic are also used in MSF applications [17].

1.1. Research motivation

In order for underwater vehicles to operate autonomously, they must perceive the spatial environment and determine their own position from the sensors' measurements. For instance, underwater mapping of the environment is critical in oil and mineral exploration operations. As the creation of maps is critically dependent upon sensor measurements, and underwater vehicle localization, determining the precise vehicle position plays a crucial role in producing accurate and consistent maps for autonomous operation. Sensor measurements are negatively affected by noise, drift, and bias, which compromise the accuracy of maps and autonomous operations. This research is mainly motivated by the need for a more accurate and reliable solution and brings the idea of improving the position estimation of the underwater vehicle.

Secondly, recent advances in soft computing techniques motivate us to develop multi-sensor fusion methods which augment their benefits to improve underwater vehicle navigation. In particular, we are interested in estimating states such as position to facilitate underwater vehicle navigation. The Kalman filter variants are most widely used in industries for estimation, but their accuracy suffers from the inaccurate process model, non-linearity, and process and inaccurate process and measurement covariance. This work mainly focuses on improving navigation filters in order to estimate the state, using benefits of neural networks, fuzzy logic, and information-theoretic learning in particular. The framework comprises existing design filters and integrates soft computing techniques. A nonlinear filtering framework serves as the basis for the framework.

Aside from that, this problem is rarely covered in the literature, which prompted us to investigate it.

1.2. Research gap

There is still much to be done when it comes to underwater navigation because there are no feasible GPS solutions. Neither the sensors nor the MSF algorithms are perfect; however, the quest to improve them continues. This work seeks to fill the gap by developing a novel multi-sensor fusion architecture that combines neural networks and error-state Kalman filters for underwater navigation. Both algorithms can be blended together to improve the ability of underwater vehicles to navigate in environments without GPS. A neural network algorithm is trained using state-space model information of an underwater vehicle since this information helps enhance estimation.

Moreover, the research utilizes correntropy strengths in order to address the current problem of underwater vehicle navigation under nonlinearities and non-Gaussian outliers. The method has aimed at improving the performance by using fuzzy logic, which has the benefit of handling nonlinearity based on expert knowledge, correntropy for robust handling of a non-Gaussian outlier, and Kalman filter for real-time processing

1.3. Research objectives

This research is designed in order to address the limitations and to fill the gap of underwater vehicle positioning as follows:

a) To explore and investigate current methods of localizing autonomous underwater vehicles and develop an underwater vehicle kinematics simulator.

b) To gain a theoretical understanding of the problem of MSF for underwater vehicle localization and to develop a framework for augmenting Kalman filter and neural networks in order to enhance the accuracy of the state estimate.

c) To propose an adaptive multi-sensor fusion method for underwater vehicle positioning with an information-theory approach that uses learning-based fuzzy rules to adapt Kalman filter covariances when outliers occur.

d) To make a comparison of the performance of the proposed MSF methods with the state-of-the-art method using its theoretical properties and simulation results.

1.4. Contributions

With the completion of this thesis, a number of contributions in the area of underwater navigation have been made. Specifically, these milestones were achieved while more detailed outlines are given in the subsequent sections:

e) **Underwater vehicles simulation environment:** The development of a simulation environment on Matlab. It is capable of modeling underwater vehicle motion and the associated sensor's measurement.

f) **Radial basis function (RBF) based neural network augmented error-state Kalman filter algorithm (ESKF):** RBF neural network augmented with error state Kalman filter for underwater positioning is mathematically derived, and it is tested with previous methods in normal mode and in failure modes.

g) **Backpropagation multi-layer perceptron neural network aided indirect Kalman filter algorithm**: An indirect Kalman filter for underwater positioning based on a hybridized backpropagation neural network is derived mathematically and tested compared to previous methods.

h) **Fuzzy and Neural network adaptive Kalman filter framework:** A framework is developed to combine Fuzzy and neural networks for handling non-linearity and incorrect noise covariances.

i) **Correntropy-Based Fuzzy Multi-Sensor Fusion:** Conventional Kalman-based and fuzzy approaches have difficulty handling non-Gaussian outliers. In order to resolve these issues, we developed an adaptive multi-sensor fusion method that uses correntropy-based fuzzy rules to adapt Kalman filter covariance.

1.5. Dissertation organization

In the dissertation, the chapters are organized as follows:

Chapter 1 gives an overview of underwater vehicle navigation systems. This is followed by a research gap and a description of the research objectives. Additionally, this chapter describes the dissertation's contribution and organization. The chapter concludes with a list of related publications.

Chapter 2 is mainly composed of state-of-the-art underwater positioning. It highlights frames of references as they are required to represent the system states, underwater vehicle kinematic parameters, and navigation information in multi-sensor fusion navigation systems discusses mathematical modeling of underwater vehicle motion models and mathematical models of sensors. Furthermore, it reviews currently existent sensor fusion methods and architectures based on Kalman filter variants. Moreover, it provides an overview of fundamentals and previous methods of fuzzy logic and neural networks. Additionally, it briefly overviews correntropy criteria.

Chapter 3 answers the question, “Can we incorporate the strengths of the neural networks to improve the underwater vehicle localization performance of Kalman filter?” Since Radial Basis Neural Networks (RBFNNs) and Backpropagation Multilayer Neural Networks (BPNNs) are well known for

their superior learning capabilities and nonlinear system identification abilities. This chapter provides a detailed explanation of the proposed neural network augmented multi-sensor underwater localization system. Mathematically, the choices made in the design of the system are explained, as well as the characteristics of each sensor. In the proposed algorithm, the RBF neural network is used to compensate for ESKF's lack of performance by reducing innovation errors. Utilizing steepest descent optimization, the RBF neural network's weights and centers are derived to minimize the estimation Mean Square Error (MSE). Under three different realistic scenarios, the proposed RBF-augmented ESKF multi-sensor fusion was compared to the conventional ESKF using Monte Carlo simulations.

Chapter 4 answers the question, "Can we use correntropy's strengths to improve underwater vehicle navigation performance in the presence of nonlinearity and outliers?" The changing underwater conditions, along with the incorrect process and measurement noise covariance, affect the accuracy of position estimation and can sometimes lead to divergence. The underwater multi-path effect and nonlinearity also contribute to outliers that have a significant impact on positional accuracy. In conventional Kalman-based methods and their fuzzy variants, non-Gaussian outliers are difficult to handle. The chapter discusses a new adaptive multi-sensor fusion method that uses knowledge-based fuzzy rules to adapt Kalman filter covariances when outliers are present. By utilizing correntropy Gaussian and Versoria kernels, new metrics for matching theoretical and actual covariances are developed. By combining correntropy-based metrics and fuzzy logic, the algorithm is robust against outliers in nonlinear dynamic underwater conditions. The proposed sensor fusion technique has been evaluated and compared using Monte-Carlo simulation, and significant improvements have been achieved in underwater position estimation.

Chapter 5 The last chapter summarizes the work done during the research and suggests some directions to be taken in future research.

1.6. Related publications

The following peer-review journals and international conference papers have been published as a result of this research:

[18] N. Shaukat, A. Ali, M. Javed Iqbal, M. Moinuddin, and P. Otero, "Multi-Sensor Fusion for Underwater Vehicle Localization by Augmentation of RBF Neural Network and Error-State Kalman Filter," Sensors, vol. 21, no. 4, p. 1149, Feb. 2021.

[19] N. Shaukat, A. Ali, M. Moinuddin and P. Otero, "Underwater Vehicle Localization by Hybridization of Indirect Kalman Filter and Neural Network," *2021 7th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, 2021, pp. 111-115, doi: 10.1109/ICMRE51691.2021.9384844.

[20] N. Shaukat, M. Moinuddin, and P. Otero, “Underwater Vehicle Positioning by Correntropy-Based Fuzzy Multi-Sensor Fusion,” Sensors, vol. 21, no. 18, p. 6165, Sep. 2021.

[21] N. Shaukat and P. Otero, "Underwater Vehicle Positioning by Fuzzy and Neural Adaptive Kalman Sensor Fusion," OCEANS 2021: San Diego – Porto, 2021, pp. 1-7, doi: 10.23919/OCEANS44145.2021.9705963.

Chapter 2 : Mathematical Modelling and State-of-the-Art

This chapter lays the groundwork for underwater navigation and sensor fusion that will be used in this thesis.

The first part describes a number of reference systems and their interpretations that are essential to underwater vehicle navigation and the study of underwater geodetic positioning. It describes the different coordinate frames and gives the equations for transforming between different frames of reference.

The second part of this chapter discusses sensors that are essential for building multi-sensor fusion algorithms. It presents mathematical models of onboard and off-board auxiliary sensors with error sources.

The third part examines the state-of -the-art of traditional sensor data fusion approaches applied to integrated navigation systems based on Kalman filtering algorithms. The Kalman filter is a powerful mathematical tool used in many engineering disciplines to estimate the state of dynamic systems accurately. A discussion of Kalman filtering's shortcomings is presented in this section. A short introduction is given to how the Fuzzy Kalman filter overcame some shortcomings of Kalman filtering. It also describes the fuzzy Kalman filter, along with an overview of fuzzification and defuzzification.

The fourth part provides a short introduction to neural networks and their benefits. Additionally, it provides a brief overview of information-theoretic learning.

2.1. Frame of references

A reference system is a set of standards and norms for defining at any given time three-dimensional coordinate axes. Reference frames are based on definite points that are accessible either directly or indirectly. In other words, the various coordinate frames merely describe the same physical quantity in distinct ways. A physical quantity can exist in any coordinate frame. Reference frames can be used to represent physical quantities. In order to define the position of an underwater vehicle, some reference is required, which has an origin and a set of axes. Generally, the origin is selected, and the axes are chosen according to an orthogonal basis. This means that at least two coordinate frames are necessary for position determination - the vehicle frame and the reference frame. Figure 2.1 Frames of References: ECI, ECEF, NED and Body shows a frame of references used to develop navigation equations for underwater vehicles, with slight modification from [22].

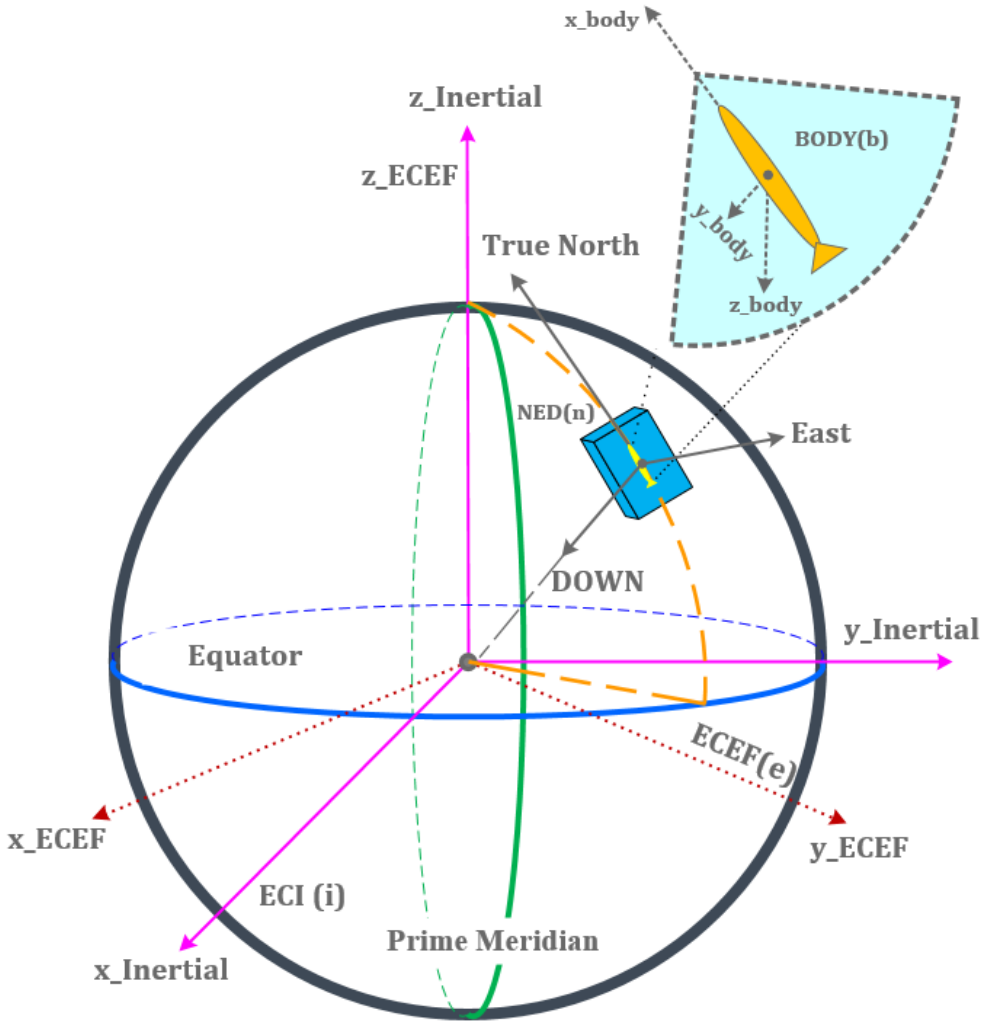

Figure 2.1 Frames of References: ECI, ECEF, NED and Body

2.1.1 Inertial frame (*i-frame*)

The inertial reference frame is considered to be the most fundamental of the reference frames. It is also called the Earth-centered inertial (ECI) frame, and Newton's laws of motion apply directly to it. At the center of the mass of the earth is the origin of the frame. Earth's rotational axis coincides with the z-axis of the frame. In the frame, the x-axis is located in the equatorial plane, pointing towards the intersection of the earth's equatorial plane and the earth's orbit around the Sun. A right-handed system is completed by the y-axis of the frame.

2.1.2 Body fixed frame (*b*-frame)

The body frame is usually attached to the origin of the center of gravity and oriented in the same way as the body of an underwater vehicle. This frame's axes remain fixed in relation to the body of the vehicle. Most commonly, the x-axis is oriented in a forward direction, then the z-axis is oriented downwards, and the y axis is oriented right, completing the orthogonal set. Choosing the center of gravity The origin of the body frame simplifies the vehicle kinematic equations[22].

2.1.3 Local geodetic or Navigation frame (*n*-frame)

Local geodetic or navigation frames are sometimes coincident with their corresponding underwater vehicle body frame. However, the axes are aligned based on the topography of northeast and down (NED), not the body of the underwater vehicle. North is located on the x-axis, east is located on the y-axis, and down is located along the z-axis. The purpose of the n-frame is to show velocity along with various directions, such as north, east, and down [23].

2.1.4 Earth Centered Earth Fixed (ECEF) frame (*e*-frame)

The origin of this frame is fixed at the center of the earth. The x-axis is located in the equatorial plane and corresponds to the mean Greenwich meridian. The y-axis will be defined to complete a right-handed system and will lie in the equatorial plane. The z-axis coincides with the axis of rotation of the earth.

The importance of frame of reference transformation for underwater navigation arises from the fact that sensors are mounted on the vehicle body. The origin of the body is defined as the center of the body frame (b). External position fixes are in a rotating, Earth-centered, Earth-fixed (ECEF) frame (e). Moreover, Newton's laws are applicable to the Earth-centered inertial (ECI) frame (i). However, the north-east-down (NED) frame or navigation frame (n) is the tangent plane to the Earth's surface at the location of the underwater vehicle, and its x-axis points toward the true north of the earth.

2.1.5 Mathematical notations

The mathematical notations used in this work are standard notations used to model underwater vehicle position, velocity, and attitude [22]. The position and velocity are three-dimensional vectors in Euclidean space. Rotations are represented by a quaternion. Subscripts and superscripts are used to represent the relationship between the frames. For example, ω_{ie}^{e} shows an angular velocity of frame (e) with respect to (i) represented in (e) frame. Vectors are represented by the bold small letters, scalars are represented by the normal small letters, and matrices by the capital bold letters. Table 2.1 Mathematical Notations used for Modelling shows a list of symbols used for modeling underwater vehicle navigation equations.

Table 2.1 Mathematical Notations used for Modelling

Notation	Description
$\mathbf{v}^b$ [u,v,w]	Linear velocity in the body frame (surge, sway, and heave)
$\mathbf{\Theta}_{nb}[\phi, \theta, \psi]$	Attitude in Euler angles from the body to NED frame
$\mathbf{q}_b^n[\eta, \epsilon]$	Attitude in quaternion from the body to NED frame
$\boldsymbol{\omega}^b$ [p,q,r]	Angular velocity in the body frame (roll, pitch, and yaw)
$\mathbf{p}^n$ $[n, e, d]$	Position in the NED frame (north, east, and down)
$\mathbf{v}^n$ $[v_N, v_E, v_D]$	Linear velocity in the NED frame (north, east, and down)
$\boldsymbol{\omega}^n$ $[\omega_N, \omega_E, \omega_D]$	Angular velocity in the NED frame
$\mathbf{p}^e$ $[x, y, z]$	Position in the rectangular ECEF frame
$\mathbf{v}^e$ $[v_{ex}, v_{ey}, v_{ez}]$	Linear velocity in the ECEF frame
$\mathbf{p}^e[\phi_{lat}, \lambda_{long}, h_d]$	Position in the ECEF geoid (latitude, longitude, and depth)
$\mathbf{R}_b^e$	Rotation matrix from the body to ECEF frame
$\mathbf{\Omega}$	Skew symmetric matrix of the angular velocity
$\mathbf{\Omega}^e$	Skew symmetric matrix in the ECEF frame
$\mathbf{\Omega}^b$	Skew symmetric matrix in the body frame
$\mathbf{g}^e$	Earth gravity vector in the ECEF frame
$\mathbf{g}^n$	Earth gravity vector in the NED frame
R_M	Radius of curvature of the prime vertical of earth
R_N	Radius of curvature of the meridian of earth
a	Semi-major axis of the ellipsoidal Earth model
e	Eccentricity of the ellipse approximation of earth

2.1.6 Geodesy coordinate frames

Geodesy uses spherical or elliptical coordinates for regional and global applications, but Cartesian coordinates for local applications.

2.1.6.1 Spherical coordinates

Considering that earth is almost like a sphere, it seems natural to use spherical coordinates. It simplifies the mathematics involved in relating a coordinate system grid with a model of the earth's shape. A sphere is distinctly defined by latitude and longitude. Latitude is defined relative to an equatorial plane, and longitude is the distance from the prime meridian to a point. The latitude ϕ_{lats}, longitude λ_{longs} and radial distance or altitude r_s , they are given as

$$r_s = \sqrt{x^2 + y^2 + z^2} \tag{2.1}$$

$$\lambda_{longs} = \tan^{-1}\left[\frac{y}{x}\right] \tag{2.2}$$

$$\phi_{lats} = \sin^{-1}\left[\frac{z}{r}\right] \tag{2.3}$$

2.1.6.2 Ellipsoid coordinates

The ellipsoid coordinates share the ECEF Cartesian coordinate system's origin as the center of mass and orientation as Earth spin axis. A reference ellipsoid is usually a flattened spheroid with two axes: the equatorial radius is the semi-major axis, and the polar radius is the semi-minor axis.

The latitude ϕ_{lat}, longitude λ_{long}, and depth h_d are computed are from rectangular ECEF points from the following equations

$$\phi_{lat} = \tan^{-1}\left[\frac{\frac{z}{\sqrt{x^2 + y^2}}}{1 - \frac{e^2 R_M}{(R_M + h_d)}}\right] \tag{2.4}$$

where $R_M = \frac{a}{\sqrt{1 - e^2 \sin\phi}}$

$$\lambda_{long} = \tan^{-1}\left[\frac{y}{x}\right] \tag{2.5}$$

$$h_d = \frac{\sqrt{x^2 + y^2}}{\cos\phi_{lat}} - R_M \tag{2.6}$$

In terms of north and east velocities, the relationship between latitude $\dot{\phi}_{lat}$ and longitude $\dot{\lambda}_{long}$ are represented by the following equations:

$$\dot{\phi}_{lat} = \frac{v_N}{R_M} + h_d \tag{2.7}$$

$$\dot{\lambda}_{long} = v_E/(R_N + h_d)\cos\phi_{lat} \tag{2.8}$$

where $R_N = R_M \frac{1-e^2}{\sqrt{1-e^2\sin\phi_{lat}}}$

The change in the height of the vehicle $\dot{h}_d$ is expressed in the form of the down velocity as

$$\dot{h}_d = -v_D \tag{2.9}$$

2.1.7 Rotation matrix

The attitude of undrwater vehicles is commonly associated with three angles, namely the roll (ϕ), pitch (θ), and yaw (ψ) angles. Rotation matrix $\boldsymbol{R}$ describes the rotation between two coordinate frames.

$$\boldsymbol{R} = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\phi\sin\theta\sin\psi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{2.10}$$

When pitch and roll angles are small, $\boldsymbol{R}$ *is* approximated as

$$\boldsymbol{R} \approx \begin{bmatrix} \cos\psi & -\sin\psi + \theta\phi\cos\psi & \phi\sin\psi + \theta\cos\psi \\ \sin\psi & \cos\psi + \theta\phi\sin\psi & -\phi\cos\psi + \theta\sin\psi \\ -\theta & \phi & 1 \end{bmatrix} \tag{2.11}$$

Rotation matrix $\boldsymbol{R}$ has the following properties

$$\boldsymbol{det(R) = 1, R^T R = RR^T = I, R^{-1} = R^T} \tag{2.12}$$

A rotation of a vector $\boldsymbol{p}^b$ from frame (b) to frame (n) is given as

$$\boldsymbol{p}^n = \mathbf{R}_b^n \boldsymbol{p}^b \tag{2.13}$$

2.1.8 Quaternion

Quaternions are mathematical representations of rotations that are singularity-free. A quaternion's normalization takes much less space and time than its matrix equivalent. Unlike matrix representation, unit quaternion notation is compact, and round-off errors are easier to handle.

A quaternion represents the attitude of an underwater vehicle by using a four-dimensional vector:

$$\boldsymbol{q} = \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \tag{2.14}$$

where η is scaler and ϵ is vector part.

The real part η can be written as by using a rotation angle ν

$$\eta = \cos(\nu/2) \tag{2.15}$$

The vector part or imaginary part can be written as by using a rotation angle ν and unit vector $\boldsymbol{n}$

$$\mathbf{q}_{vec} := \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} = [\boldsymbol{n} \sin(\nu/2)] \tag{2.16}$$

The rotation matrix is derived from quaternion from the following transformation

$$\boldsymbol{R}_a^b(\boldsymbol{q}) = \begin{bmatrix} 1-2(\epsilon_2^2+\epsilon_3^2) & 2(\epsilon_1\epsilon_2-\epsilon_3\eta) & 2(\epsilon_1\epsilon_3+\epsilon_2\eta) \\ 2(\epsilon_1\epsilon_2+\epsilon_3\eta) & 1-2(\epsilon_1^2+\epsilon_3^2) & 2(\epsilon_2\epsilon_3-\epsilon_1\eta) \\ 2(\epsilon_1\epsilon_3-\epsilon_2\eta) & 2(\epsilon_2\epsilon_3+\epsilon_1\eta) & 1-2(\epsilon_1^2+\epsilon_2^2) \end{bmatrix} \tag{2.17}$$

The inverse unit quaternion is given as

$$\mathbf{q}^{-1} = \begin{bmatrix} \eta \\ -\epsilon \end{bmatrix} \tag{2.18}$$

The quaternion product of two unit quaternions is also a unit quaternion

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \begin{bmatrix} \eta_1\eta_2 - \epsilon_1^T\epsilon_2 \\ \eta_1\epsilon_2 + \eta_2\epsilon_1 + \epsilon_1^\times\epsilon_2 \end{bmatrix} \tag{2.19}$$

where $\epsilon_1^\times$ is skew-symmetric matrix-vector part of $\boldsymbol{q}_1$.

2.1.9 Strapdown Inertial Navigation System

Strapdown Inertial navigation systems (SINS) or sometimes (INS) of underwater vehicles contain three orthogonal accelerometers that measure the specific force of a sensor in a *frame* (*b*) in relation to inertial frame (*i*). In addition, the INS contains a triadic arrangement of three gyroscopes that measure angular rates [22]. In this context, it is important to emphasize that the measurements are made in inertial space, which is why they are called inertial sensors. These sensors are part of the Inertial measurement unit (IMU). In some cases, an IMU includes a magnetometer or compass [3].

Accelerometers output:	$\mathbf{f}_i^b$	(specific force)
Gyroscope outputs:	$\boldsymbol{\omega}_{ib}^b$	(angular rates)

The simplified block diagram of IMU is shown in Figure 2.2 below

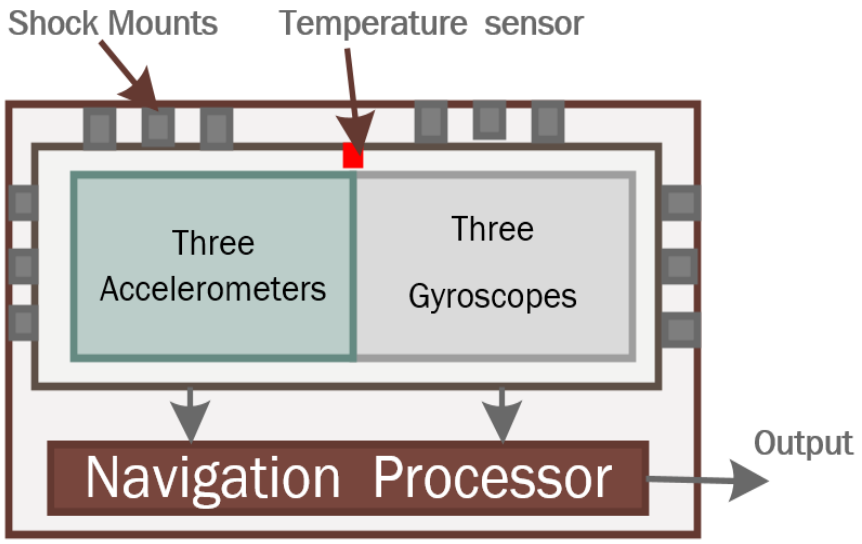

Figure 2.2 Inertial Measurement Unit (IMU)

2.1.10 INS navigation equations in e-frame

Navigation using an inertial navigation system is fundamentally based on the integration of inertially sensed acceleration with respect to time.

The rate of change in position $\dot{\mathbf{p}}^e$ and velocity $\mathbf{v}^e$ of vehicle in (e) frame is related by the following differential equation

$$\dot{\mathbf{p}}^e = \mathbf{v}^e \tag{2.20}$$

The rate of change in velocity $\dot{\mathbf{v}}^e$ of the underwater vehicle in (e) frame is dependent on accelerometer output $\mathbf{f}^b$, angular velocity $\boldsymbol{\omega}_{ie}^e$, and gravity vector $\mathbf{g}^e$ in ECEF and can be expressed by the following differential equation

$$\dot{\mathbf{v}}^e = \mathbf{R}_b^e \mathbf{f}^b - 2\boldsymbol{\Omega}_{ie}^e \mathbf{v}^e + \mathbf{g}^e \tag{2.21}$$

where $\boldsymbol{\Omega}_{ie}^e$ is a skew-symmetric matrix of angular velocity $\boldsymbol{\omega}_{ie}^e$ and $\mathbf{R}_b^e$ is the rotation matrix that transforms a specific force vector from the (b) frame to the (e) frame.

$$\boldsymbol{\Omega}_{ie}^e = [\boldsymbol{\omega}_{ie}^e \times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.22}$$

The rate of change of rotation matrix $\dot{\mathbf{R}}_b^e$ represented in the frame (e) is dependent on the angular velocity $\boldsymbol{\Omega}_{ib}^b$ of the body with respect to the frame (i), and angular velocity $\boldsymbol{\omega}_{ie}^b$ of the earth with respect to the frame (i) is expressed by the following equation:

$$\dot{\mathbf{R}}_b^e = \mathbf{R}_b^e(\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{ie}^b) \tag{2.23}$$

2.1.11 INS navigation equations in n-frame

The *n-frame* is fixed to the moving vehicle, which means the vehicle cannot move within it. In this case, the desired velocity is computed using an *e*-frame coordinate system. In other words $\dot{\mathbf{p}}^n \neq \mathbf{v}^n$. Velocity $\mathbf{v}^n$ can therefore be referred to by the e-frame when dealing with the navigation frame

$$\dot{\mathbf{v}}^n = \mathbf{R}_e^n \dot{\mathbf{v}}^e \tag{2.24}$$

The rate of change in velocity of the vehicle in the (n) frame $\dot{\mathbf{v}}_{eb}^n$ is expressed by the following differential equation:

$$\dot{\mathbf{v}}_{eb}^n = \mathbf{R}_b^n \mathbf{f}_{in}^b - (\boldsymbol{\Omega}_{en}^n + 2\boldsymbol{\Omega}_{ie}^n)\mathbf{v}_{eb}^n + \mathbf{g}_{eb}^n \tag{2.25}$$

where $\boldsymbol{\Omega}_{en}^n \mathbf{v}^n$ is the centripetal acceleration related to the motion of the (n) frame with respect to the (e) frame and the term $2\boldsymbol{\Omega}_{ie}^n \mathbf{v}^n$ is the Coriolis acceleration.

The local gravity vector $g_{eb}^{n} = [0 \quad 0 \quad g]^{T}$ depending on the latitude, longitude, radius of curvature of the meridian, and radius of curvature of the prime vertical is given by

$$\mathbf{g}_{eb}^{n} = \frac{g_0}{\left(1 + \frac{h_d}{\sqrt{R_N R_M}}\right)^2} \tag{2.26}$$

where $g_0 = 9.780318 \times (1 + 5.3024 \times 10^{-3}\sin^2\phi - 5.9 \times 10^{-6}\sin^2 2\phi)$ [Titterton and Weston(2004)]. The rate of change in the rotation matrix $\dot{\mathbf{R}}_b^n$ represented in the frame (n) is dependent on the angular velocity of the body with respect to the frame (i) and on the angular velocity of the frame (n) with respect to the frame (i):

$$\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n\left(\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{in}^b\right) \tag{2.27}$$

2.1.12 Quaternion kinematics

The attitude of the vehicle represented in quaternion $\dot{\mathbf{q}}_b^n$ is given as

$$\dot{\mathbf{q}}_b^n = \frac{1}{2}\begin{bmatrix} 0 \\ \omega_{nb,x}^n \\ \omega_{nb,y}^n \\ \omega_{nb,z}^n \end{bmatrix} \otimes \mathbf{q}_b^n = \frac{1}{2}\mathbf{q}_b^n \otimes \begin{bmatrix} 0 \\ \omega_{nb,x}^b \\ \omega_{nb,y}^b \\ \omega_{nb,z}^b \end{bmatrix} \tag{2.28}$$

The $\otimes$ sign represents a quaternion product. The alternate representation of $\dot{\mathbf{q}}_b^n$ is given as

$$\dot{\mathbf{q}}_b^n = \frac{1}{2}\mathbf{A}\mathbf{q}_b^n \tag{2.29}$$

where $\boldsymbol{\omega}_{nb}^b$ is given as

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{R}_n^b(\boldsymbol{\omega}_{en}^n + \mathbf{R}_{\boldsymbol{e}}^{\boldsymbol{n}}\boldsymbol{\omega}_{ie}^e) \tag{2.30}$$

and $\mathbf{A}$ is given as

$$\mathbf{A} = \begin{bmatrix} 0 & -\omega_{nb,x}^b & -\omega_{nb,y}^b & -\omega_{nb,z}^b \\ \omega_{nb,x}^b & 0 & \omega_{nb,z}^b & -\omega_{nb,y}^b \\ \omega_{nb,y}^b & -\omega_{nb,z}^b & 0 & \omega_{nb,x}^b \\ \omega_{nb,z}^b & \omega_{nb,y}^b & -\omega_{nb,x}^b & 0 \end{bmatrix}$$

2.1.13 INS navigation equations implementation

The purpose of this section is to discuss the implementation of navigation equations in a strapdown configuration, in which observations are obtained directly from the underwater vehicle's body frame. The gyroscope measurement and accelerometer measurement are associated with a timestamp t_k. The observation represents the average value over the interval $\delta t = t_k - t_{k-1}$ from the previous to the current timestamp. Quantities at the beginning and end of the update interval will be denoted by (-), while values at the end of the interval are denoted by (+).

The rotation quaternion is computed based on the measured angular rates in order to determine the attitude from the following equation

$$\mathbf{q}_b^n(+) = e^{\mathbf{A}^b \delta t/2}\mathbf{q}_b^n(-) \tag{2.31}$$

The observed specific force can now be transformed into n-frames by the following equation

$$\mathbf{f}^n = \overline{\mathbf{R}}_b^n \mathbf{f}^b \tag{2.32}$$

where $\overline{\mathbf{R}}_b^n$ is given as

$$\overline{\mathbf{R}}_b^n = \frac{\mathbf{R}_b^n(-) + \mathbf{R}_b^n(+)}{2} \tag{2.33}$$

The vehicle velocity is obtained by the following equation

$$\mathbf{v}^{\boldsymbol{n}}(+) = \mathbf{v}^{\boldsymbol{n}}(-) + [\mathbf{f}^{\boldsymbol{n}} - (\mathbf{2}\boldsymbol{\Omega}_{\boldsymbol{ie}}^{\boldsymbol{n}}(-) + \boldsymbol{\Omega}_{\boldsymbol{en}}^{\boldsymbol{n}}(-))\mathbf{v}^{\boldsymbol{n}}(-) + \mathbf{g}^{\boldsymbol{n}}(-)]\delta t \tag{2.34}$$

The vehicle position is obtained by numerical integration of the following equation

$$\mathbf{p}^{\boldsymbol{n}}(+) = \mathbf{p}^{\boldsymbol{n}}(-) + \dot{\mathbf{p}}^{\boldsymbol{n}}(-)\delta t \tag{2.35}$$

Figure 2.3 shows a flow diagram representation of the implementation process. In the next cycle, new position, velocity, and attitude information become old.

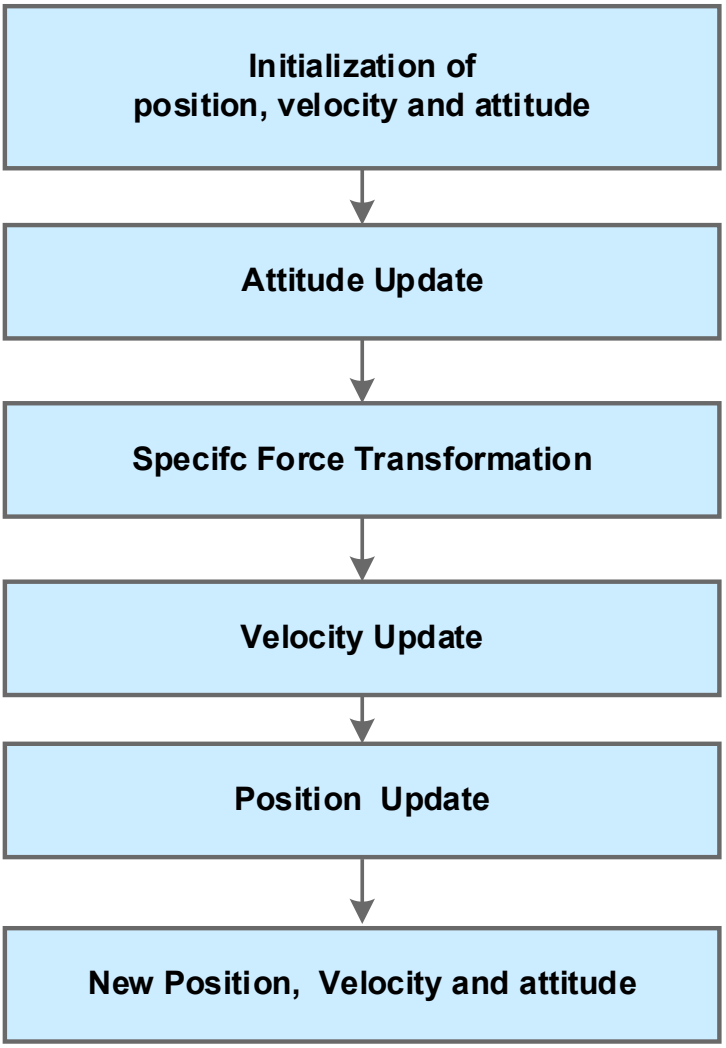

Figure 2.3: Flow chart representation of navigation equations implementation

2.1.14 Error dynamic model

The position $\mathbf{p}$, velocity $\mathbf{v}$, attitude $\mathbf{q}$, gyro bias $\mathbf{b}_g$, accelerometer bias $\mathbf{b}_{acc}$, acoustic fix bias $\mathbf{b}_h$ in full state vector $\mathbf{x}$ form are given as

$$\mathbf{x} = [\mathbf{p} \quad \mathbf{v} \quad \mathbf{q} \quad \mathbf{b}_g \quad \mathbf{b}_{acc} \quad \mathbf{b}_h \quad s_v]^T \tag{2.36}$$

where s_v is the sound velocity model underwater used to calculation underwater acoustic transmission.

The estimated state vector $\hat{\mathbf{x}}$ is written as

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{p}} & \hat{\mathbf{v}} & \hat{\mathbf{q}} & \hat{\mathbf{b}}_g & \hat{\mathbf{b}}_{acc} & \hat{\mathbf{b}}_h & \hat{s}_v \end{bmatrix}^T \tag{2.37}$$

The design used in this work estimate an error-state vector, which offers the main advantages of flexible sampling rate, robustness, and low computation burden [24], [25]. The error-state vector $\delta\dot{\mathbf{x}}$ is the difference between the true state and estimated state $\hat{\dot{\mathbf{x}}}$ of the model and is given as

$$\delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}} \tag{2.38}$$

The error-state vector can be represented as

$$\delta\mathbf{x} = [\delta\mathbf{p} \quad \delta\mathbf{v} \quad \delta\mathbf{q} \quad \delta\mathbf{b}_g \quad \delta\mathbf{b}_{acc} \quad \delta\mathbf{b}_h \quad \delta\mathbf{s_v}]^T \tag{2.39}$$

where the position, velocity, and attitude error equations are provided as under. For the complete derivation of this error model, readers can refer to [26], [27].

Assuming that the underwater vehicle travel at low speed and that the depth of operation is much less than the earth's radius, the rate of change in errors in longitude $\delta\dot{\lambda}_{long}$, latitude $\delta\dot{\phi}_{lat}$, and depth $\delta\dot{h}_d$ are given by

$$\begin{bmatrix} \delta\dot{\lambda}_{long} \\ \delta\dot{\phi}_{lat} \\ \delta\dot{h}_d \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dfrac{-v_n}{(R_N + h_d)^2} \\ \dfrac{v_e \sin(\lambda_{long})}{(R_N + h_d)\cos^2(\lambda_{long})} & 0 & \dfrac{-v_e}{(R_N + h_d)^2 \cos(\lambda_{long})} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\lambda_{long} \\ \delta\phi_{lat} \\ \delta h_d \end{bmatrix} + \begin{bmatrix} \dfrac{1}{R_M + h_d} & 0 & 0 \\ 0 & \dfrac{1}{(R_N + h)\cos(\lambda_{long})} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta v_n \\ \delta v_e \\ \delta v_d \end{bmatrix} \tag{2.40}$$

The velocity error $\delta\dot{\mathbf{v}}_{eb}^n$ is given by the following differential equation:

$$\delta\dot{\mathbf{v}}_{eb}^n = \delta\mathbf{g}_{eb}^n + \delta\mathbf{R}_b^n\mathbf{f}^b + \mathbf{R}_b^n\delta\mathbf{f}_{ib}^b - \left(2(\delta\boldsymbol{\Omega}_{ie}^n) + (\delta\boldsymbol{\Omega}_{en}^n)\right)\hat{\mathbf{v}}_{eb}^n - \left(2(\boldsymbol{\Omega}_{ie}^n) + (\boldsymbol{\Omega}_{en}^n)\right)\delta\mathbf{v}_{eb}^n \tag{2.41}$$

If the vehicle operates at low speed underwater, angler velocity ω_{ie}^n and $\delta\omega_{ie}^n$ can be neglected. Gravity error is also neglected because of the small operating area and accurate estimation [28],[29].

Thus, the velocity error $\delta\dot{\mathbf{v}}_{eb}^{n}$ a differential equation can be re-written as

$$\delta\dot{\mathbf{v}}_{eb}^{n} = \delta\mathbf{R}_{b}^{n}\mathbf{f}^{b} + \mathbf{R}_{b}^{n}\delta\mathbf{f}_{ib}^{b} \tag{2.42}$$

Assuming that the angular velocity of rotation of the earth with respect to the inertial frame ω_{ie} is accurately known, using the attitude error model in the form of the quaternion, it is given as

$$\dot{\delta}\mathbf{q} = \frac{1}{2}\delta\boldsymbol{\omega}_{ib}^{b} + \frac{1}{2}\left(\delta\boldsymbol{\Omega}_{ib}^{b}\right)\delta\mathbf{q} \tag{2.43}$$

The error of gyro bias $\delta\dot{\mathbf{b}}_{g}$ is given as

$$\delta\dot{\mathbf{b}}_{g} = \dot{\mathbf{b}}_{g} - \dot{\hat{\mathbf{b}}}_{g} \tag{2.44}$$

The error of accelerometer bias $\delta\mathbf{b}_{acc}$ is given as

$$\delta\mathbf{b}_{acc} = \dot{\mathbf{b}}_{acc} - \dot{\hat{\mathbf{b}}}_{acc} \tag{2.45}$$

The error of hydro-acoustic system bias $\delta\mathbf{b}_{acc}$ is given as

$$\delta\mathbf{b}_{h} = \dot{\mathbf{b}}_{h} - \dot{\hat{\mathbf{b}}}_{h} \tag{2.46}$$

2.2. Sensors on the vehicle

In the following subsections, we cover the sensors used in underwater with a mathematical formulation. In order to be as realistic as possible, the simulations in this work utilize several of these underwater sensors' specifications for multi-sensor fusion.

2.2.1 Inertial Measurement Unit

The inertial measurement unit (IMU) provides a three-axis accelerometer and three-axis gyro outputs. These measurements are affected by various factors such as temperature, manufacturing process, scale factor, noise, and drift temperature, manufacturing process, scale factor, noise, and drift.

The SDI500 is a tactical-grade IMU designed for underwater use, with high performance based on Emcore's quartz MEMS inertial sensor technology. MEMS-based IMU technology demonstrates the true tactical grade performance with gyro bias and accelerometer bias stability and very low angle random walk [30]. Another industry standard for underwater applications is LN-200, which is a

compact, lightweight, highly reliable fiber-optic IMU from Northrop Grumman [31]. A compact package contains three fiber-optic gyros and three silicon microelectromechanical system accelerometers that measure the angle and velocity changes. The figure shows Emcore SD-1500 IMU [30].

Figure 2.4: Emcore SDI-1500 Inertial Measurement Unit [30]

Functional requirements for navigation may differ fundamentally between different underwater applications. IMU grading is mostly based on metrics that are related to bias performance. It is because different bias characteristics may have different cumulative impacts based on their characteristics. The following are the most important characteristics of IMU errors [32]:

Bias repeatability refers to the variance in bias offset across several runs. It is primarily caused by changes in sensor status and external influences across runs. In other words, it refers to the offset between the output value and the input value.

Scale factor error measures the ratio between the sensor's input and its output. The accuracy of the scale factor determines the system's ability to follow the rotation and translation of a vehicle.

Angle Random Walk (ARW) and **Angle Random Velocity** (ARV) are the integral of the white noise in sensors output.

Table 2.2 compares gyroscopes manufacturer's specifications used most commonly for underwater navigation

Table 2.2 IMU Gyro specifications

Gyro specifications	**Emcore SDI-1500**	**LN-200**
Bias Repeatability	1 °/hr.	1 °/hr.
Scale factor Accuracy	200 ppm	100 ppm
Random walk	0.02 °/√hr	0.07 °/√hr
Dynamic Range (max)	1000 °/sec	1,000 °/sec

Table 2.3 compares accelerometer manufacturer's specifications used widely used for underwater navigation

Table 2.3 IMU Accelermeter specifications

Accelerometer Specifications	**Emcore SDI-1500**	**LN-200**
Bias Repeatability	1 milli-g	300 micro-g
Scale factor Accuracy	200 ppm	300 ppm
Velocity Random Walk	100 µg/√Hz	35 µg/√Hz
Input Acceleration	50 g	40 g

The accelerometer measurement output vector $\mathbf{f}_{acc}^{b}$ in (b) frame is modeled as

$$\mathbf{f}_{acc}^{b} = \mathbf{f}_{ib}^{b} + \mathbf{b}_{acc} + \boldsymbol{\varrho}_{acc} \tag{2.47}$$

where $\boldsymbol{\varrho}_{acc}$ is white noise and $\mathbf{b}_{acc}$ is acceleration bias, which is modeled as a 1st-order Markov process.

The accelerometer bias $\mathbf{b}_{acc}$ is expressed by the random walk and the random constant shown as

$$\dot{\mathbf{b}}_{acc} = -\boldsymbol{\tau}_{acc}^{-1}\mathbf{b}_{acc} + \boldsymbol{\rho}_{acc} \tag{2.48}$$

where $\boldsymbol{\tau}_{acc}$ is the correlation time given by manufacturer. The correlation time gives information about the relationship between successive values of sensor bias instability, and the relationship between consecutive values of bias instability reduces when the sensor correlation time increases. A longer correlation time results in a longer convergence time. The value $\boldsymbol{\rho}_{acc}$ depends on standard deviation, and it is known as a process driving noise.

The accelerometer in IMU does not directly measure the true kinematic acceleration of vehicles due to the presence of the earth's gravitation. For that reason, the measurement from the accelerometer is known as the relative acceleration or specific force $\mathbf{f}_{ib}^{b}$ and it is related to the kinematics acceleration of the vehicle as follows:

$$\mathbf{f}_{ib}^{b} = \dot{\mathbf{R}}_{i}^{b}\ddot{\mathbf{p}}^{i} - \mathbf{g}^{b} \tag{2.49}$$

where $\ddot{\mathbf{p}}^{i}$ is acceleration represented in an inertial frame and $\mathbf{g}^{b}$ is gravitation sensed by the accelerometer in the body frame.

The actual output of the gyro $\boldsymbol{\omega}_{g}^{b}$ is influenced by noise and bias that is given by

$$\boldsymbol{\omega}_{g}^{b} = \boldsymbol{\omega}_{ib}^{b} + \mathbf{b}_{g} + \boldsymbol{\varrho}_{g} \tag{2.50}$$

where $\boldsymbol{\varrho}_{g}$ is the white noise and $\mathbf{b}_{g}$ is the gyro bias, which is modeled as a 1st-order Markov process.

The gyro bias $\mathbf{b}_{g}$ is represented by a random walk and a random constant given as follows:

$$\dot{\mathbf{b}}_{g} = -\boldsymbol{\tau}_{g}^{-1}\mathbf{b}_{g} + \boldsymbol{\rho}_{g} \tag{2.51}$$

where $\boldsymbol{\tau}_{g}$ is the gyro correlation time. Its value depends on the quality of the accelerometer sensors used in the IMU. The value $\boldsymbol{\rho}_{g}$ depends on the standard deviation, and it is called process driving noise. The IMU used for simulation is tactical grade Emcore SDI-1500. It uses high-precision micro-electro-mechanical systems (MEMS) quartz sensor technology with 1 °/h gyro bias and 1 mg accelerometer bias stability. The IMU offers the best cost to performance ratio compared to other technologies. It consists of three orthogonal accelerometer sensors that provide the measurement of specific forces. Three gyros provide the angular rates of the body with respect to the inertial frame of reference.

2.2.2 Underwater acoustic positing system

Underwater acoustic positioning systems measure the distance and direction of the vehicle from the reference positions [1]. HiPaP 502 is one of the leading acoustic positioning products in the industry. An operating range of 1 to 5000 m is provided by this system, which has a range detection accuracy of 0.2 m [33]. It can be connected to GPS for earth-related coordinates. Acoustic position estimates, however, are affected by factors such as GPS accuracy, system installation, ship attitude, sound velocity profile, and measurement noise.

The performance specification HiPaP502 are shown in Table 2.4 [33]

Table 2.4 Acoustic positioning system specifications

HiPAP 502 Specifications (single)	
S/N	20 dB rel. 1μPa
Operational coverage	$\pm 110°$
Angular accuracy (X & Y direction)	$0.06°$
Angular repeatability	$0.01°$
Cymbal range accuracy	0.02 m
Receiver beam	$10°$

Underwater decibels dB are measured in relation to a pressure of 1 microPascal (Pa), abbreviated as dB re 1 Pa. The above specification assumes no ray bending, a free line of sight between transducer and transponder, and an SNR of 12 dB. rel. 1μPa

Assuming the system is precisely calibrated, installation and ship attitude have negligible effects. The mathematical model of the system actual output $\hat{\mathbf{p}}$ is given as

$$\hat{\mathbf{p}}_h = \mathbf{p}_h + \mathbf{b}_h + \boldsymbol{\varrho}_h \tag{2.52}$$

where $\mathbf{p}_h$ is true output position, whereas $\mathbf{b}_h$ is the time-varying bias modeled as a 1st-order Markov process and depends on the sound velocity profile and ray bending effect. $\boldsymbol{\varrho}_h$ represents measurement white noise.

2.2.3 Doppler Velocity Log

Doppler velocity logs (DVL) measure the change in acoustic frequency to determine the speed of a vehicle with respect to the seabed [34]. DVL measures the speed of the water relative to the seabed in deep water when the seabed is not available. The DVL sends a known frequency signal to the seabed and receives the signal when it bounces back to the vehicle. The speed of the underwater vehicle is determined by the doppler effect.

Table 2.5 shows specifications of Nortek's DVL-500 [35] and Teledyne's WHN-300 [36] are widely used for underwater applications

Table 2.5 DVL specifications

DVL Specifications	DVL-500	WHN-300
Std dev at 3 m/s	±0.5cm/s	±0.4%±0.2cm/s
Long-term accuracy	0.1% / ±0.1 cm/s	±0.4%±0.2cm/s
Minimum altitude	0.3 m	1.0m
Maximum altitude	200m	200m
Maximum ping rate	8 Hz max	7Hz max
Maximum range	70 m	110m
Velocity resolution	0.01 mm/s	0.1cm/s

Considering that the attitude and installation errors are negligible, the actual output of DVL $\hat{\mathbf{v}}_{dvl}$ can be modeled for the true velocity measurement vector, noise, and bias given as

$$\hat{\mathbf{v}}_{dvl} = \mathbf{v}_{dvl} + \mathbf{b}_{dvl} + \boldsymbol{\varrho}_{dvl} \tag{2.53}$$

where $\mathbf{v}_{dvl}$ are the true output and random velocity error, $\mathbf{b}_{dvl}$ is modeled as the 1st-order Markov process, and $\boldsymbol{\varrho}_{dvl}$ is white noise. The Figure 2.5 Nortek DVL-500 [35]

Figure 2.5 Nortek DVL-500 [35]

2.2.4 Depth sensor

There is a direct relationship between depth and underwater pressure. As the vehicle goes deeper into the water, the pressure reading increases linearly. The figure illustrates the underwater depth sensor part number 8CDP700-I made by Paroscientific, Inc [37]. Figure 2.6 shows Paroscientific underwater depth sensor

Figure 2.6 Paroscientific underwater depth sensor [23]

The specifications of made by Paroscientific depth sensor used for underwater application are given in Table 2.6

Table 2.6 Depth sensor specifications

Depth Sensor Specifications	
Repeatability	≤± 0.005% Full Scale
Over Pressure	1.2 times
Resolution	1×10^{-8}
Accuracy	0.01%
Gauge Pressure	0-15 psig to 0-200 psig

The depth sensor actual output $\hat{h}_d$ is modeled by adding true depth h_d with noise:

$$\hat{h}_d = h_d + \varrho_d \tag{2.54}$$

where ϱ_d is measurement noises modeled as white noise.

2.2.5 e-Compass

An e-compass or magnetometer measures the strength and direction of the Earth's magnetic field. An example of a jewel instrument ECS-AC-RS232 e-compass is shown in Figure 2.7, which is equipped with a 3-axis magnetometer and a 2-axis tilt sensor [38]. e-compass or magnetometer readings are susceptible to the surrounding magnetic field. It requires a reliable calibration method to function properly. There are several major sources of magnetometer error, including declination angle, that is, the difference between true and sensor north; motors and ferromagnetic materials that cause soft and hard magnetic distortion; and sensor imperfections, misalignments, and noise. The majority of these errors in measurement can be eliminated by using proper calibration.

Figure 2.7 Jewel instrument e-compass [24]

Table 2.7 shows the specification of ECS-AC-RS232 e-compass. The Earth's magnetic field lines form an angle with the horizontal called the dip angle, or magnetic inclination. The tilt range is a range of inclination measurement in degrees.

Table 2.7 e-Compass specifications

e-Compass Specifications	
Accuracy (rms)	±0.5°
Repeatability	±0.3°
Response Time (msec)	36
Dip Angle Range	±0.3°
Tilt Range	±42°

The actual output $\mathbf{q}_m$ of the magnetometer is a combination of noise and true output $\mathbf{q}_m$ written as

$$\hat{\mathbf{q}}_m = \mathbf{q}_m \otimes \varrho_m \tag{2.55}$$

where ϱ_m is the sensor noise modeled as white noise.

2.3. Multi-sensor fusion using Kalman filtering state-of-the-art

In 1960, Rudolph Kalman developed a new method for minimum-mean-square error filtering. Kalman Filter is a stochastic observer, which means that it makes use of noise statistics to improve the estimation of the states by using a predictor-corrector structure [39]. This state is estimated using a model of the system dynamics and then corrected for measurements made by real-world measurment of sensors. The Kalman filter can be viewed as a naturlal direct data fuser, as it takes the measurements from sensors and merges them with the process model, making it very useful for combining multisensory data [40]. Multi-sensor data fusion can be integrated at underwater vechile kinematic kinematic level either by using centralised fusion, which uses a single KF, or by using decentralized approches, which use multiple KF.

Kalman filters possess the following properties [41]:

- The Kalman filter is unbiased and is the maximum likelihood estimator. Moreover, it is also the minimum mean-square error estimator.
- Kalman filter is the optimal state estimation algorithm for a linear system.

Kalman-filtering theory implies that the system model is described by a set of differential equations represented in state-space form as

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} + \mathbf{w} \tag{2.56}$$

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{2.57}$$

F is matrix of the dynamics of the system, **x** is a column state vector with the states of the system **u** is called the control vector, while **w** is a process noise. **H** is the measurement matrix, **z** is the measurement vector, and **v** is measurement noise. Noises $\mathbf{w}, \mathbf{v}$ assumed to be drawn from a zero mean normal distribution $\mathbf{w} \sim \mathcal{N}(0, \mathbf{Q})\ and\ \mathbf{v} \sim \mathcal{N}(0, \mathbf{R})$. The process noise covariance **Q** and measure noise covariance **R** is given as

$$\mathbf{Q} = E[\mathbf{w}\mathbf{w}^T] \tag{2.58}$$

$$\mathbf{R} = E[\mathbf{v}\mathbf{v}^T] \tag{2.59}$$

The discrete implementation of Kalman filter fundamental matrix **Φ** is calculated by system matrix as

$$\mathbf{\Phi}(t) = e^{\mathbf{F}t} \tag{2.60}$$

$$\mathbf{\Phi}_{\mathrm{k}} = \mathbf{\Phi}(\mathrm{T_s}) \tag{2.61}$$

Where $\mathrm{T_s}$ is sampling time.

Kalamn filter Implementation Steps

In following equations superscript minus $^-$ denotes the a priori state that occurs before. Superscript plus $^+$ denotes the posteriori state.

$\mathbf{x}_0^-$ = Initialization of the state variables.

$\mathbf{P}_0^-$ = Initialization of covariance matrix.

$\mathbf{Q}_0$ = Initialization of process noise covariance.

$\mathbf{R}_0$ = Initialization of measurement noise covariance.

State Estimation:

$$\mathbf{x}_{k+1}^- = \mathbf{\Phi}_k \mathbf{x}_k^- + \mathbf{w}_k \tag{2.62}$$

where $\mathbf{x}_{k+1}^-$ is the predicted error state and $\mathbf{\Phi}_k$ is the state transition matrix in discrete form.

Covariance Prediction:

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k^+ \mathbf{\Phi}_k^\top + \mathbf{Q}_k \tag{2.63}$$

The Kalman $\mathbf{K}_k$ gain is given as

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\top + \mathbf{R}_k\right)^{-1} \tag{2.64}$$

The Kalman gain can be large when there is a large variance in the state prediction and a small variance in the measurement. When the Kalman gain is small, the state prediction will have a small variance, but the measurement will have a large variance.

The updated estimate $\mathbf{x}_k^+$ is given as

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^-) \tag{2.65}$$

where actual measurement is $\mathbf{z}_k$ and the prediction of measurement $\mathbf{H}_k \mathbf{x}_k^-$.

The Updated covariance $\mathbf{P}_k^+$ is given as

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^- \tag{2.66}$$

Figure 2.8 shows flow chart representation of the Kalman filter [42]

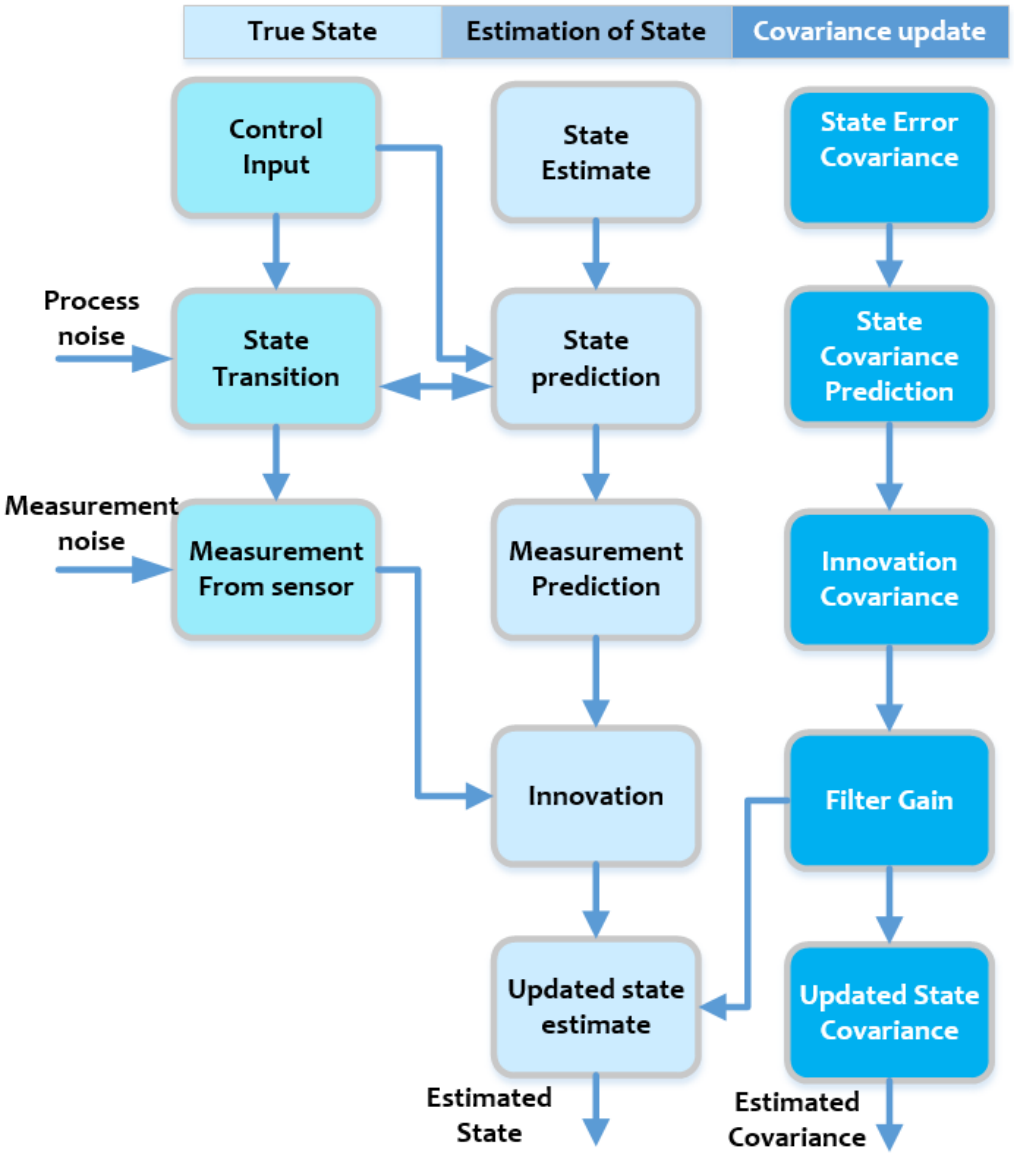

Figure 2.8 Flow Chart representation of Kalman Filter [42]

2.3.1 Extended Kalman Filter

Motions of underwater vehicles lead to nonlinear dynamics equations. Modeling the motion of an underwater vehicle in continuous time can be done by using

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \tag{2.67}$$

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}, t) \tag{2.68}$$

The Kalman filter is based on linear systems theory, and so it needs to be linearized in order to be used on nonlinear systems. This is done by using Taylor series approximation and keeping only 1st order term. Because we are dealing with matrices in the equations, The matrix **F** and **H** corresponds to the first-order derivative called Jacobian matrices.

$$\dot{\mathbf{x}} = \mathbf{F}(\hat{\mathrm{x}}, \mathbf{u}, t)\mathbf{x} \tag{2.69}$$

where $\mathbf{F}(\hat{\mathrm{x}}, \mathbf{u}, \mathrm{t}) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}$

$$\boldsymbol{z} = \mathbf{H}(\hat{\mathrm{x}}, t)\mathbf{x} \tag{2.70}$$

where $\mathbf{H}(\hat{\mathrm{x}}, \mathrm{t}) = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}$

The flow chart representation Extended Kalman filter is shown in Figure 2.9 [42]

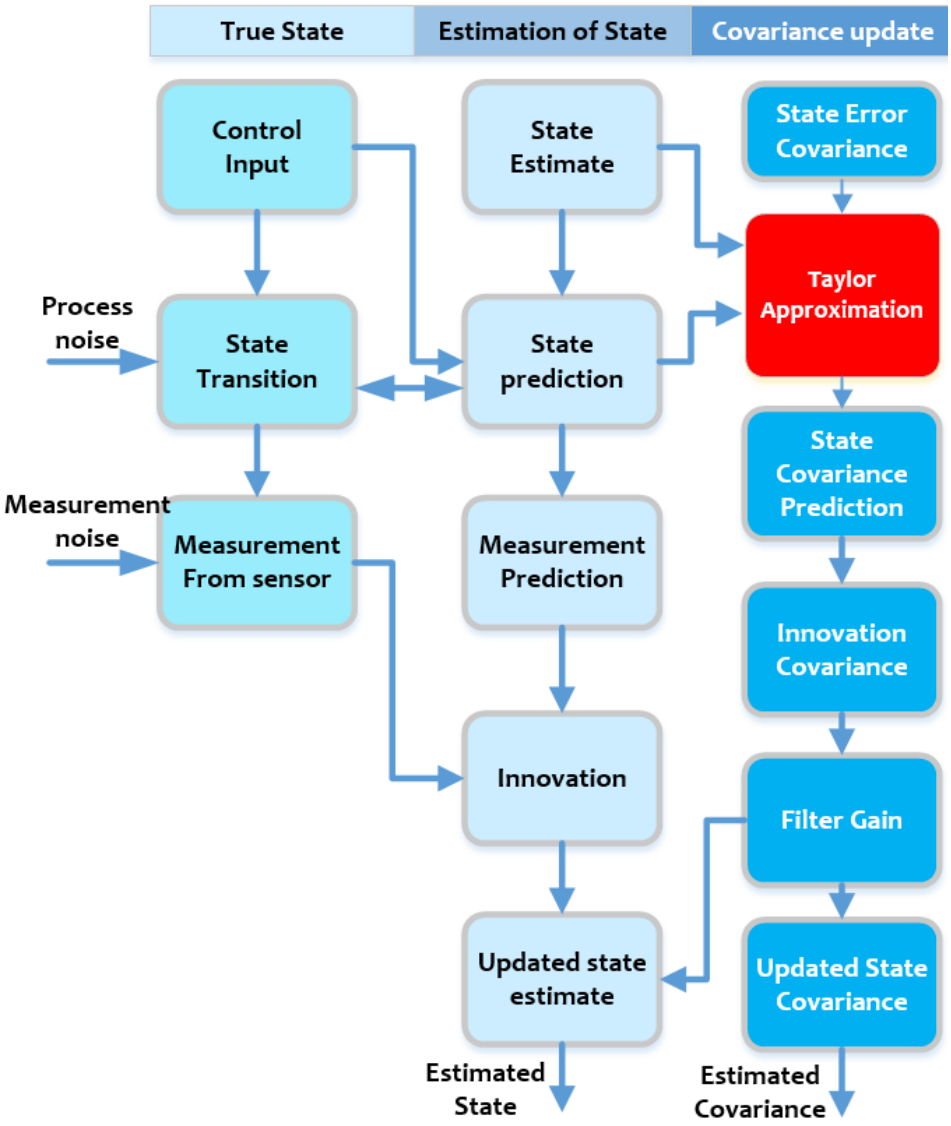

Figure 2.9 Flow Chart representation of Extended Kalman Filter

2.3.2 Error-state Kalman filter

In error-state models, Kalman Filters are used to express the difference between predictions and measurements. It has the advantage that if the error state is considered, the dynamic will be accurately represented by small parameters, far from possible parameters singularities, like gimbal locks. However, when true states are considered, significant nonlinear signals will be able to capture the dynamic of the system. Additionally, the equations for the Total State EKF conform to those for a closed-loop ESKF model [3]. Therefore, both approaches are expected to exhibit the same behavior. However, the implementation of the two approaches will differ.

Multi-sensor fusion uses the Error-State Kalman Filter, in which the measurement signal provided by the high rate INS is processed to identify the true navigation state [26]. During this step, the model imperfections, including errors and noise, are not considered. This will result in accumulating errors and drifting of the navigation solution over time. To provide an accurate solution, these errors must be estimated and corrected. The KF does this. Based on the measurements made by the low-rate sensors, it is possible to predict and correct errors in the INS. The predicted errors from the estimation are fed into the INS solution so that all states are corrected.

The following equation shows how ESKF estimates errors in position, velocity, attitude, and bias of INS for the underwater vehicle.

$$\delta \mathbf{x}_{k+1}^{-} = \mathbf{\Phi}_k \delta \mathbf{x}_k^{-} \tag{2.71}$$

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_k^{+} \mathbf{\Phi}_k^{\top} + \mathbf{Q}_k \tag{2.72}$$

$$\mathbf{K}_k = \mathbf{P}_k^{-} \mathbf{H}_k^{\top} \left(\mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^{\top} + \mathbf{R}_k \right)^{-1} \tag{2.73}$$

$$\delta \mathbf{z}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_k) \tag{2.74}$$

$$\delta \mathbf{x}_k^{+} = \delta \mathbf{x}_k^{-} + \mathbf{K}_k (\delta \mathbf{z}_k - \mathbf{H}_k \delta \mathbf{x}_k^{-}) \tag{2.75}$$

$$\mathbf{P}_k^{+} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^{-} \tag{2.76}$$

$$\hat{\mathbf{x}}_k^{+} = \hat{\mathbf{x}}_k^{-} + \delta \mathbf{x}_k^{+} \tag{2.77}$$

ESKF can be implemented in two ways: feedforward and close loop [22].

2.3.3 Shortcoming

There are several limitations of Kalman filtering; some of the major ones are listed below [43], [44], and [45].

- In the case of underwater application of ESKF filter, divergence occurs when the actual error of the system is not bounded in the prediction.
- The underwater vehicle process model is inaccurate, which affects the state estimation.
- Since underwater environments are highly non-linear, linear Taylor approximations to 1st order or 2nd order (or any type of approximation) can yield inaccurate results and bias in estimation.
- Noise statistic is non-Gausion and has outliers
- Inaccurate covariance matrices or noise statistics are always changing.
- The larger the error, the worse the performance of the filter.

2.3.4 Adaptive Kalman Filter

It is usually the underwater vehicles sensor specifications and system characteristics that determine the measurement noise covariance matrix R and the system noise covariance matrix Q in the initial design phase. Nevertheless, it may be difficult to determine R and Q with certainty, or the Kalman Filter tuning may change over time as a result of underwater operations, which affects sensor characteristics or vehicle dynamics profile. In either case, the noise covariance due to underwater operation can be corrected using an adaptive Kalman Filter.

Kalman filter adaptation techniques have been developed based on the many different adaptation algorithms [46], [47]. There are four main approaches for covariance adaptation such as,

- Covariance matching based adaptation
- Bayesian adaptation
- Maximum Likelihood estimation-based adaptation
- Correlation-based adaptation

Covariance matching is the most common adaptation approach, which aims to match theoretical covariance values with actual covariance values. Covariance matrices are adapted by adjusting both the process noise and measurement noise covariances at the same time or by adjusting either of those two covariances at a time if only one has been determined.

To compare, we use theoretical covariance. It can be expressed as

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k. \tag{2.78}$$

A moving windows average of measurement innovation is used to determine the actual covariance [46]. It is given as

$$\mathbf{C_k} = \frac{1}{\lambda} \sum_{\mathbf{i}=\mathbf{i}_0}^{\lambda} \mathbf{s_k} \mathbf{s_k^T}, \tag{2.79}$$

where λ is the size of the window and $\mathbf{s_k} = \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^-$.

Bayesian, Maximum Likelihood, and correlation techniques are out of the scope of this thesis.

2.3.5 Multi-sensor fusion using fuzzy logic state-of-the-art

A leading application of fuzzy logic in adaptive control systems is the modeling of errors in systems that have nonlinearities, for which heuristic knowledge can be used instead of mathematical modeling. The concept of a fuzzy set was introduced in 1965 by Lotfi Zadeh. A fuzzy set describes entities with a degree of membership, opposed to the strictly discrete sets used in classical logic states. If the system is incomplete or uncertain, fuzzy logic is used to make decisions based on estimated values. States are assessed based on a continuous range of membership between true (1)and false (0), and quantifies how close they are to a linguistic classification.

A fuzzy logic system can be implemented in three main steps:

- Fuzzification - the process of converting input values into fuzzy membership functions,
- Fuzzy logic rule base – based on the fuzzy inference
- Defuzzification - convert the fuzzy output to crisp.

The fuzzification involves mapping a numerical value, or a crisp input, into fuzzy sets. By interpreting these fuzzy inputs through a rule base, we are then able to meet fuzzy logic system (FLS) objectives. Mamdani's work is widely credited with being the first to apply fuzzy set theory to controller design [48]. Takagi-Sugeno is the second most popular fuzzy approach [49]. The output of a Mamdani fuzzy system is a unique fuzzy set, making it intuitive and ideally suited to integrate human intuition. Using fuzzy inputs and rule bases, the fuzzy outputs are then defuzzified to give numerical crisp output values for the system's control or adaptation parameters. Figure 2.10 shows the basic building blocks of FLS

2.3.6 Fuzzy adaptive Kalman filter architecture

Kalman filtering assumes that the noise is white noise, i.e., Gaussian noise with zero mean and finite variance. In practice, this assumption doesn't always hold. If non-gaussian noise is present, the system will diverge from its optimal solution. To address this problem, fuzzy adaptive Kalman filters are employed. The following sections will review fuzzy adaptive Kalman filtering.

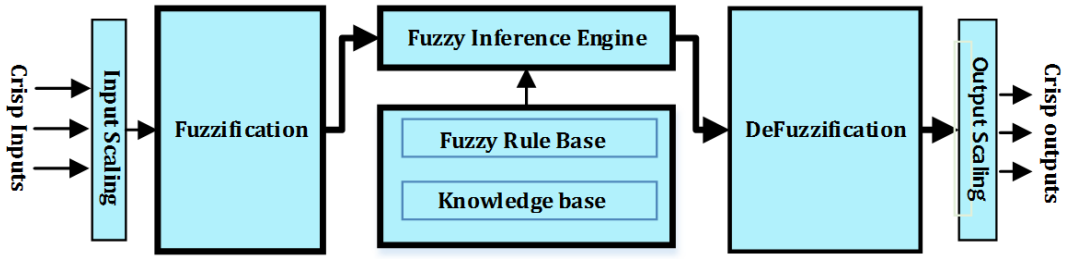

Figure 2.10 Fuzzy logic system building blocks

Fuzzy Adaptive Kalman Filters (FAKF) are Kalman filters augmented with a feedback adaptation control, which adjusts the covariance matrix Q and R by using fuzzy logic. The early research proposed using a Fuzzy Adaptive Extended Kalman Filter (FAEKF), which uses an exponential weights adjusted by a Fuzzy logic [50]. Figure 2.11 shows the architecture of Fuzzy Adaptive Kalman Filter (FAKF).

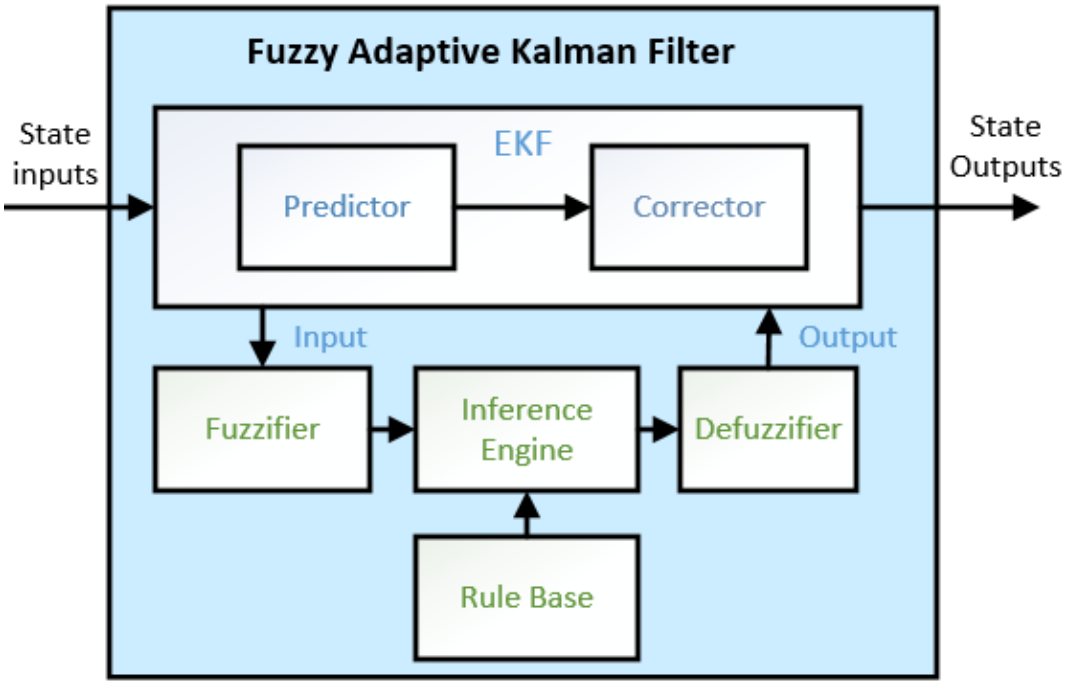

Figure 2.11 Fuzzy Adaptive Kalman Filter (FAKF)

To implement FL with KL, three steps can be taken. Starting with choosing the state and membership functions that will be used by the FL, establishing FL's basic rules, and how to adjust the Kalman filter based on the FL output.

A fuzzy set can be visually represented by membership functions. Moreover, The fuzzy sets describe how much an input value corresponds to the fuzzy logic criteria. The membership functions are used to map inputs into a universe of discourse from zero to one, taking different shapes into account in a manner appropriate to the distribution of data [51].

The membership function of a fuzzy set J can be represented by

$$J = \{(l, \mu_J(l)), \quad \text{suchthat } l \in L\}. \tag{2.80}$$

The membership value of the element l in fuzzy subset J is denoted as $\mu_J(l)$. The universe L contains the crisp variable l.

The triangular function is defined by the following equations:

$$\mu_j(l) = \begin{cases} 0, & l \le a \\ \dfrac{l-a}{m-a}, & a < l \le m \\ \dfrac{b-l}{b-m}, & m < l < b \\ 0, & l \ge b \end{cases} \tag{2.81}$$

where the lower limit is defined by a, an upper limit b. *m represent a place where membership function is one.* where $a < m < b$.

Figure 2.12 shows the fuzzy triangular function

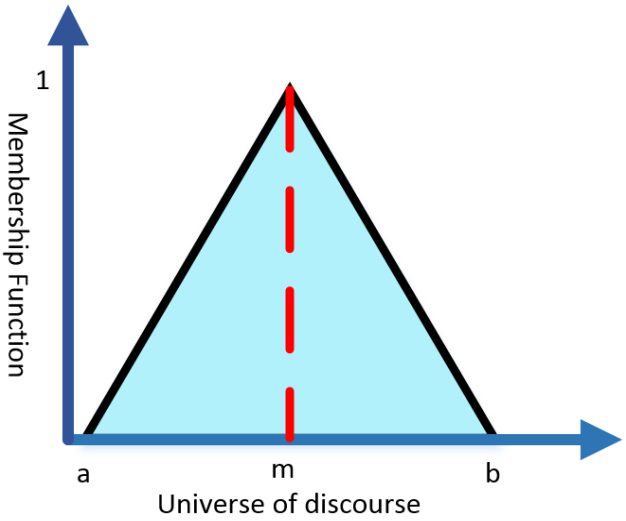

Figure 2.12 Fuzzy triangular membership function

The trapezoidal functions are used on the extreme left and right. The right trapezoidal function is defined as

$$\mu_J(l) = \begin{cases} 0, & l > d \\ \dfrac{d-l}{d-c}, & c \le l \le d \\ 1, & l < c \end{cases} \tag{2.82}$$

The left trapezoidal function is defined as

$$\mu_J(l) = \begin{cases} 0, & l < a \\ \dfrac{l-a}{b-a}, & a \le l \le b \\ 1, & l > b \end{cases} \tag{2.83}$$

The Figure 2.13 Trapezoidal membership functions show trapezoidal membership funcitons

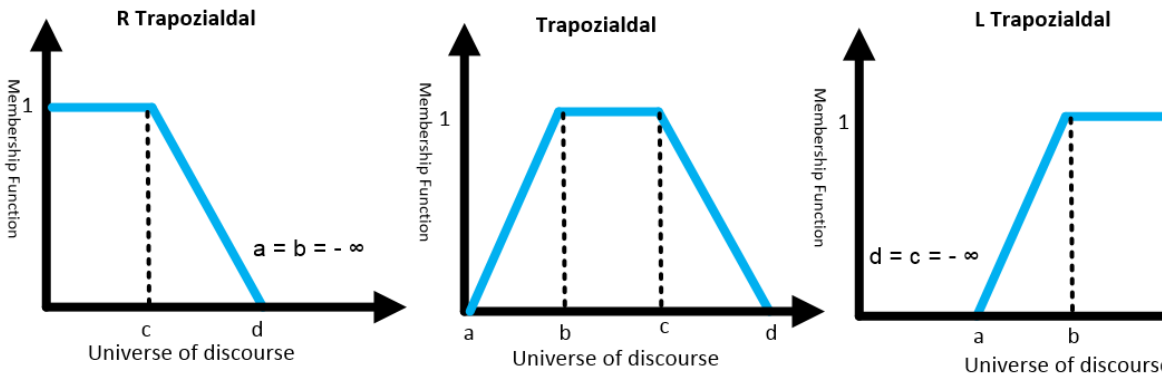

Figure 2.13 Trapezoidal membership functions

The Fuzzy based rule uses a deductive form described as the IF-THEN rule base. It expresses an inference such that if one fact is known, it is possible to infer a conclusion. There is usually more than one premise needed to draw an overall conclusion. A combination of conjunctive or disjunctive rules can be used for aggregation. The AND logical conjunction connects the rules in conjunctive systems, whereas disjunctive systems connect their rules by an "OR" logical conjunction.

The general fuzzy rules is given as

$IF\ premise$ (Temperature is High), $THEN\ conclusion$ (Cooling ON)

Once the inputs have been mapped to fuzzy sets, followed by defining rule bases that describe the logical relations that should be maintained by the FLS, the inference is then employed to assess the likelihood of an input corresponding to an output. Based on Mamdani inference, the output membership functions are fuzzy sets. Fuzzy "and" operation computes Zadeh-min, taking the minimum of the two membership values.

Defuzzification is the process of transforming a fuzzy output, which cannot be used directly in a distinct crisp value. There are many defuzzification methods most popular are the weighted average method, the center of sums, the centroid method, and the Center of gravity (COG) method.

Among these methods, the most accurate method for estimating crisp values from fuzzy input is using the COG method. The (COG) method [52] is used to get a crisp value of α_k, given by the following equation

$$\alpha_k = \frac{\sum_{i=1}^{n} \Delta_i \times e_i}{\sum_{i=1}^{n} \Delta_i}, \tag{2.84}$$

where n depend on the partition of linguistic rules. Δ_i represent the area under the membership function (i), and e_i is the ith centroid.

For fuzzy adaptive Kalman filters, the most commonly used criteria are by using a Degree of Mismatch ($\mathbf{DOM}$) and Degree of Divergence (DOD) , which are mathematically expressed as:

$$DOD = Tr(\mathbf{S}_k) - Tr(\mathbf{C}_k) \tag{2.85}$$

$$\mathbf{DOM} = \mathbf{S}_k(j,j) - \mathbf{C}_k(j,j) \tag{2.86}$$

Where Tr is trace operator and DOD is scaler and $\mathbf{DOM}$ is vector**.**

When theoretical and actual covariances are matched perfectly, DOD and DOM are almost equal. Additionally, positive or negative values of these metrics indicate the direction of tuning for covariance matrices.

2.4 Artificial Neural Networks

Artificial neural networks try to mimic the human brain. By using a training set, a computer is taught to do certain tasks. This can be referred to as machine learning. In recent years, neural networks have been the cutting-edge technology for many applications. Signal processing, pattern recognition, non-linear modeling, nonlinear controller design, and navigation fall under this category. The nodes of neural networks called perceptrons are usually organized into multilayers. Each layer of the network has an input, output, and weight vector, respectively. An output layer is simultaneously an input of the next layer. In a feedforward neural network, data moves from input to output layers through hidden layers in one direction.

A number of factors have motivated huge research interest in the use of neural networks for underwater navigation, especially as alternatives to conventional methods, particularly:

- Any linear or nonlinear functions can be learned by a neural network. Artificial neural networks are able to learn on their own, eliminating the need for complex and complicated mathematical implementation.
- The main navigation algorithm based on the Kalman filtering method is optimal only on linear systems. Multilayered neural networks include nonlinear activation functions in the hidden neurons that can be used to map highly nonlinear problems for which traditional navigation methods are ineffective.
- A Kalman filtering technique requires mathematical modeling of underwater vehicles and environments, which cannot be accurate. Neural networks are capable of self-learning, so they do not require large amounts of information.
- Parallel processing of neural networks provides very fast multiprocessing capability

In a neural network, forward propagation refers to the process by which intermediate variables and outputs are calculated by perceptron. A perceptron is a basic unit that consists of many inputs and a single output. The initial hidden layer receives the input data forward, processes it, and passes the results to the subsequent layers.

In Figure 2.14, the inputs from the states are connected to the simple perceptron [53] [54].

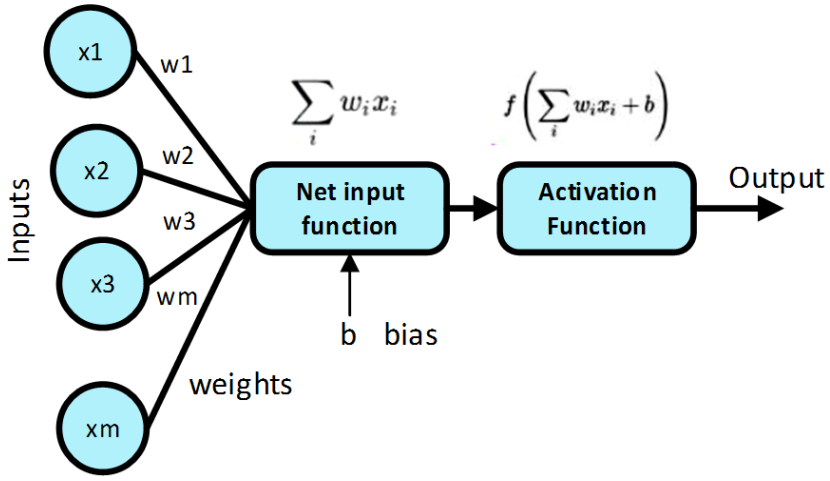

Figure 2.14 Design of a standard Perceptron

The inputs are multiplied by a weight, then added together. A non-linear operation may be performed by the activation function on the sum.

Multilayer Perceptron (MLP) networks can be implemented easily with a fully connected layer architecture. The layer's inputs are all the outputs of its previous layer. Perceptrons in an MLP are arranged in layers like in Figure 2.15

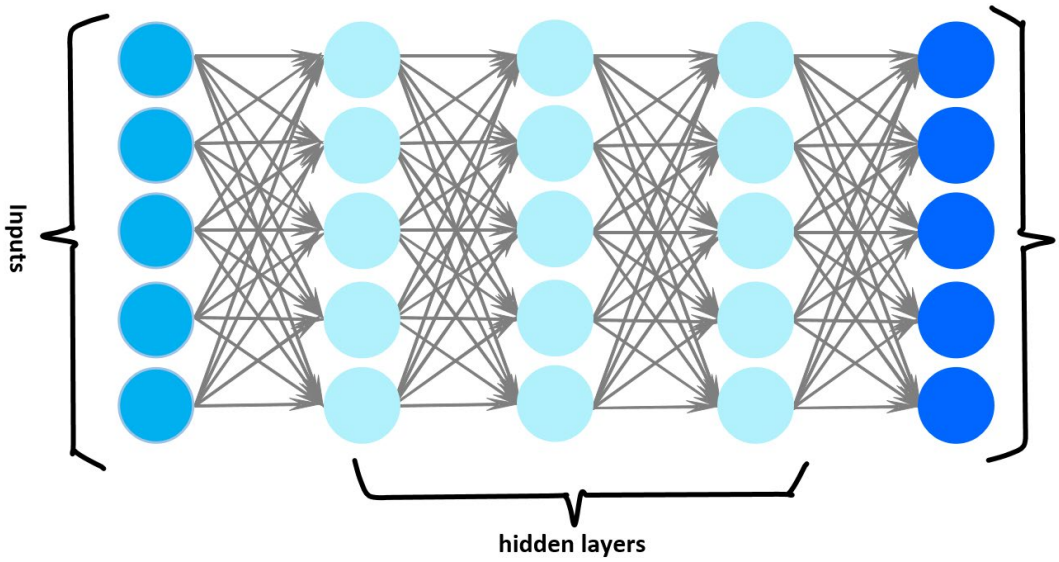

Figure 2.15 Multi-layer perceptron neural network

The MLP can be mathematically represented as

$$\mathbf{a}^l = \mathbf{f}^l\left(\mathbf{W}^l\mathbf{a}^{l-1} + \mathbf{b}^l\right) \quad \text{for} \quad l = 0,1,\dots,L \tag{2.87}$$

Where weight matrix in neural network is denoted by $\mathbf{W}^l$ where superscript (l) denotes layer number. The $\mathbf{a}$ denotes input to layer and $\mathbf{a}^{l-1}$ comes from external data. The $\mathbf{f}^l$ denotes transfer function of the layer and $\mathbf{b}^l$ denotes bias of the layer l. The performance index backpropagation algorithm uses the Mean Square Error (MSE). An MSE metric is a simple metric that can be computed quickly and has properties such as convexity, symmetry, differentiability. The MSE equation can be written as

$$\hat{F}(\mathbf{x}) = (\mathbf{t}_k - \mathbf{a}_k)^T(\mathbf{t}_k - \mathbf{a}_k) \tag{2.88}$$

where $\mathbf{x}$ is a vector containing weights and $\mathbf{t}$ shows target output. Where k shows the iteration number.

The error $\mathbf{e}$ is represented by subtracting $\mathbf{a}_k$ from target output $\mathbf{t}_k$ can be written as

$$\mathbf{e} = (\mathbf{t}_k - \mathbf{a}_k) \tag{2.89}$$

So the MSE can be written as

$$\hat{F}(\mathbf{x}) = \mathbf{e}_k^T\mathbf{e}_k \tag{2.90}$$

To train an MLP neural network, each perceptron's weights must be tuned. One of the most common approaches is to use stochastic gradient descent. Optimization algorithms based on stochastic gradient descent minimize a value of a loss function. This method is called Back Propagation (BP). It reflects how the errors in the loss function propagate backward into the weights so that updates to the weights are made. To control the magnitude of weight changes for each update, learning rate and gradient are multiplied.

From gradient descent algorithm weight $w^l_{ji(k+1)}$ and bias $b^l_{j(k+1)}$ of the network can be calculated as

$$w^l_{ji,(k+1)} = w^l_{ji,k} - \alpha_w \frac{\partial \hat{F}}{\partial w^l_{ji,k}} \tag{2.91}$$

where α_w is the learning rate of weight update and subscript i and j show row and column of weight, respectively

$$b^l_{j,(k+1)} = b^l_{j,k} - \alpha_b \frac{\partial \hat{F}}{\partial b^l_j} \tag{2.92}$$

The sensitivity term shows how much weight and bias should be adjusted to get the desired output. It is calculated by taking gradient of $\hat{F}$ with respect to net layer $\mathbf{n}^l$ and written as

$$\mathbf{s}^l = \frac{\partial \hat{F}}{\partial \mathbf{n}^l} = \left(\frac{\partial \mathbf{n}^{l+1}}{\partial \mathbf{n}^l}\right)^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{l+1}} \tag{2.93}$$

2.4.1 Radial Basis Functions Neural Networks

Radial Basis Neural Functions Networks (RBFNN) was introduced in the '80s, and they have attracted recent attention for their generalizability and a simple network structure that avoids unnecessary and lengthy calculations in comparison to feed-forward networks with multiple layers. Furthermore, the ability to learn online fast, easy design, good generalization, and tolerating input noise are also important features of RBFNN. Unfortunately, neural network models can be biased by adversarial noise and make incorrect predictions. This is not true for RBF networks due to their nonlinear nature.

Three layers are present in RBFNN: the input layer of neurons, the hidden layer, and the output layer. The task of this layer is to link input data with the hidden layers. This layer is made up of as many nodes as the vector dimension. Radial basis functions activate the neurons at the hidden layer. This layer is composed of hidden nodes that contain computing units. Hidden neurons have their own centers. This function computes the distance between the center of each input and the input itself and returns some nonlinear function based on that distance. A radially symmetric activation function determines the output of each node in the hidden layer. Usually, the largest output is obtained when inputs are near the node's center. The output layer computes the linear combination of the values passed from the hidden layer. Its dimension is determined by the output vector. The RBFNN is shown in Figure 2.16

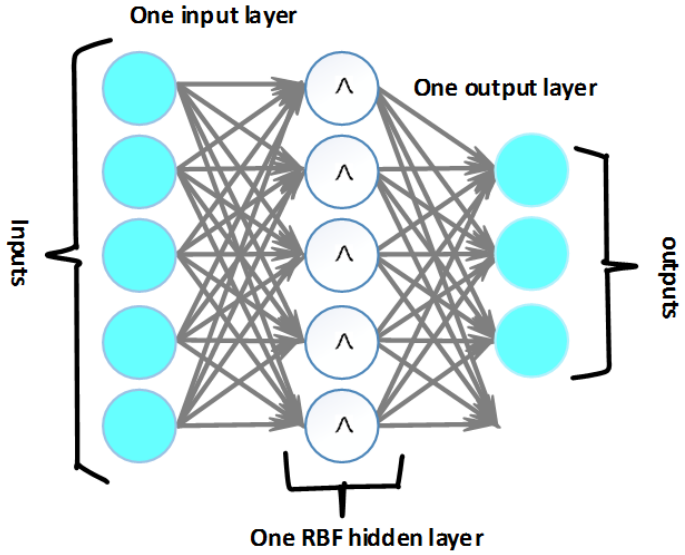

Figure 2.16 RBFNN

Normally, there are many types of activation functions. One widely used is a Gaussian function. The activation of the ith node is given as follows:

$$\mathbf{y}_k(i) = \exp\left(-\frac{\|\mathbf{x}_k^- - \mathbf{c}_i\|^2}{2\boldsymbol{\sigma}_i^2}\right), \quad i = 1,2,\dots,N_c \tag{2.94}$$

where $\mathbf{c}_i$ is the neuron center and $\boldsymbol{\sigma}_i$ is the width of the Gaussian RBF function. The weights of the neurons in matrix form $\mathbf{w}_k$ are represented as

$$\mathbf{W}_k = \begin{pmatrix} w_{1k}(1) & w_{1k}(2) & \cdots & w_{1k}(N_c) \\ w_{2k}(1) & w_{2k}(2) & \cdots & w_{2k}(N_c) \\ \vdots & \vdots & \ddots & \vdots \\ w_{Mk}(1) & w_{Mk}(2) & \cdots & w_{Mk}(N_c) \end{pmatrix} \tag{2.95}$$

where M donates the size of the state vector and N_c represents the number of neuron centers. The size of $\mathbf{W}_k$ is $M \times Nc$. The size of $\mathbf{W_k y}_k$ is $M \times 1$, and the size of $\mathbf{y}_k$ is $N_c \times 1$.

For the weight matrix $\mathbf{W}_k$, we only consider weight associated with the mth-specific output. Thus, the weight vector of the mth output of RBF can be denoted by $\mathbf{w}_{mk}$. Therefore, the weight update rule of the specific mth output is given as

$$\mathrm{w}_{mk+1} = \mathrm{w}_{mk} - \eta_w \frac{1}{2}\frac{\partial \mathbf{J}}{\partial \mathbf{w}_{mk}} \tag{2.96}$$

where η_w is the learning rate, and its value is selected by experimentation. Too large value of η_w can cause unstablity and divergance, and too low value can make the response sluggish.

The center update rule for the jth element of the mth neuron center is given as

$$\mathbf{c}_{i+1}(j) = \mathbf{c}_i(j) - \eta_c \frac{1}{2}\frac{\partial \mathbf{J}}{\partial \mathbf{c}_i(j)} \tag{2.97}$$

where η_c is the learning rate of center update.

The width of each Gaussian kernel is assigned as: the wider the width, the greater the separation between the centers, whereas the smaller the width, the closer the centers. Another way to decide the widths is to set the widths to a constant across all radial basis functions. The constant is calculated by dividing the maximum distance between any two neurons by the square root of the two times of a number of centroids.

2.4.2 Information Theoretic Learning and Correntropy State of the art

The Computational Neuro-Engineering Laboratory (CNEL), led by Prof. José Carlos Principe, established Information Theoretic Learning (ITL) at the University of Florida. Using information-theoretic descriptors instead of conventional statistics, the ITL was designed as a framework to implement non-parametrically adaptive systems. As a matter of fact, in machine learning and signal processing applications, the learning process is traditionally based on mean square error (MSE). The MSE for the two random variables A and B is given by the following formula:

$$\mathrm{MSE}(A,B) = E[(A-B)^2] \tag{2.98}$$

According to the least-squares criterion with N observation, also called the Minimum Mean Square Error (MMSE) criterion, the following error e is minimized as

$$\frac{1}{N}\sum_{i=1}^{N}(a_i - b_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 \tag{2.99}$$

From the above equation, it is obvious by moving away from the a is equal to b line, the MSE quadratically increases. A sample that is far from the mean will have a greater impact. In other words, the MMSE is optimal for Gaussian distributions.

For the same two variables A and B, the Correntropy function is:

$$v(A,B) = E[\kappa_\sigma(A-B)] \tag{2.100}$$

where κ_σ is the kernel function and E is expectation operator.

Correntropy can be calculated directly from samples using Information Theoretic Learning principles

$$\hat{v}(A,Y) = \frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(a_i,b_i) \tag{2.101}$$

Correntropy is also a similarity metric, but unlike MSE, which is global, it can be attributed as a local metric. The Gaussian kernel bandwidth corresponds to local properties, and it practically does not consider the samples that are outside the local region, defined by the kernel bandwidth. In other

words, Global indicates that all samples in a joint space contribute significantly to the similarity measure's value, whereas local indicates that its value is primarily determined by the kernel function. Correntropy also has the property that, as the kernel bandwidth tends to infinity, the estimator becomes equivalent to the MMSE (least squares) estimator. It can be said that the Correntropy function connected entropy cost functions to Huber's robust statistics [55]. Correntropy has the advantage of being a local similarity criterion, and it is likely to perform particularly well when noise is nonzero mean, non-Gaussian, or contains significant outliers [56]. Figure 2.17 shows Mean Square Error (MSE), and Figure 2.18 shows correntropy in join spaces A and B [57].

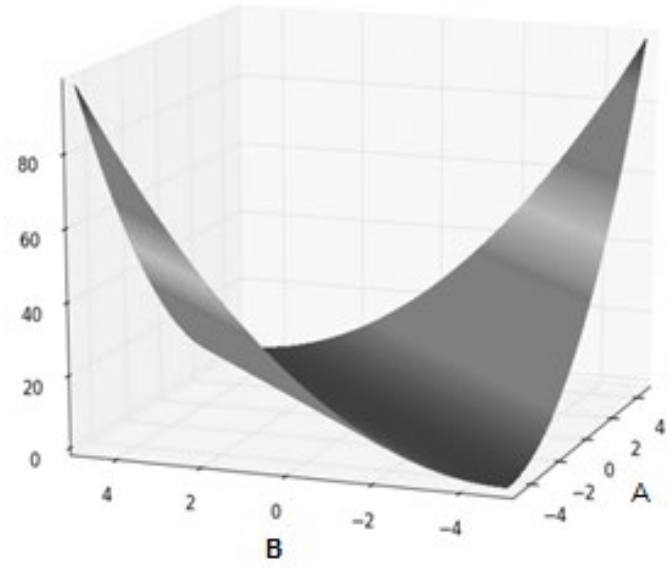

Figure 2.17 Mean Square Error in joint space (A-B) [38]

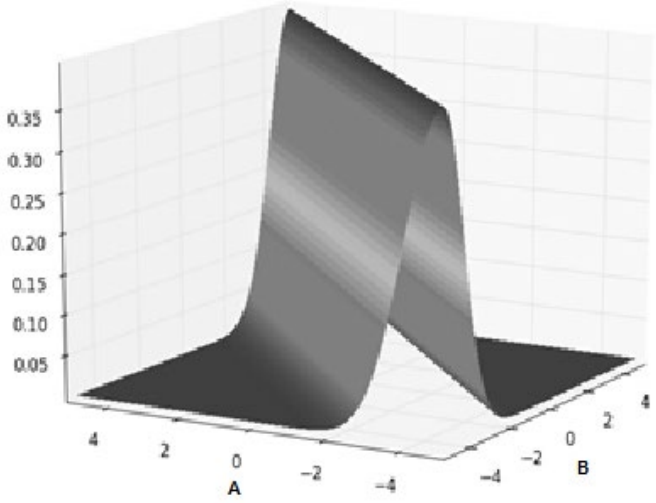

Figure 2.18 Correntropy in joint space (A-B) [20]

2.5 Conclusion

This chapter establishes the foundation for underwater navigation and sensor fusion, which is utilized throughout the thesis. The first section of this text covers a variety of reference systems and their interpretations that are critical for underwater vehicle navigation and underwater geodetic positioning research. It offers the equations for changing between multiple frames of reference and defines the various coordinate frames. The second section of this chapter goes through the sensors that are required to create multi-sensor fusion algorithms. Mathematical models of onboard and offboard auxiliary sensors with error sources are presented. The third part examines the state of the art of traditional sensor data fusion approaches to integrated navigation based on Kalman filtering algorithms with shortcomings. Here, we also introduce how the Fuzzy Kalman filter overcomes some of the limitations of Kalman filtering and provides the designer with the ability to incorporate human intuition and expert knowledge in the rules of adaptation. A brief overview of neural networks is provided in the fourth part. Further, it gives a brief overview of information-theoretic learning.

Chapter 3 MSF for Underwater Vehicle Localization by augmentation of ANN and ESKF

This chapter is based on our work of Multi-Sensor Fusion (MSF) for improving underwater vehicle localization by augmentation of Radial Basis Function Neural Network (RBFNN) and Error State Kalman Filter (ESKF) [18]. In this chapter, the key research question is how Artificial Neural Networks (ANNs) can assist with Kalman filtering in underwater environments. We propose a novel multi-sensor fusion method for underwater state estimation that takes advantage of Radial Basis Neural Networks (RBFNN) to enhance accuracy. The equations of weights and centers of the RBFNN are derived by minimizing the estimation Mean Square Error (MSE). Our method is compared with state-of-the-art methods under different sensor failure conditions. It works well under noisy conditions and is only slightly worse when the sensor fails.

The mathematical modeling of underwater vehicle kinematics and sensors used in the chapter is discussed in the previous chapter. The first and second sections of this chapter give a brief introduction and state of the art of multi-sensor fusion. Thrid part discusses the novelty of the research. The structure and mathematical formulation of the multi-sensor fusion algorithm are discussed in the fourth part. The fifth part discusses the complexity of the algorithm. The sixth part shows test results of the proposed multi-sensor fusion filter in three different conditions. In addition, acoustic communication lost and on-board sensor malfunctioning is tested and analyzed. Moreover, the performance of the RBF-ESKF is compared with ESKF under different scenarios. A comparative analysis is performed, showing that the proposed algorithm has promising results. The last part of this chapter is based on the conclusions.

3.1. Introduction

The ocean floor has billions of dollars of natural resources in the form of precious elements and medicinal herbs. To take advantage of ocean resources, seabed mapping is the ultimate tool that depends on precise sensors and robust navigation fusion algorithms for Autonomous Underwater Vehicles (AUVs) and Remotely Operated Underwater vehicles (ROVs). Navigational accuracy is a key requirement for complex seabed mapping tasks [58]. However, primary sensors, three-axis gyros, and accelerometers have biases and drifts, which vary with time and are affected by noise. In contrast, most commonly used fusion algorithms based on Extended Kalman filter (EKF) and its variant error-state Kalman (ESKF) suffer from divergence and degraded Mean Square Error (MSE) performance in the nonlinear underwater condition because of linear approximation [59]. Thus, the higher the nonlinearity present in the system, the greater the error of the EKF state prediction, and it can also induce filter divergence.

Backpropagation multi-layer Neural Networks (BPNN) and Radial Basis Neural Networks (RBFNN) have excellent learning abilities and are well-known for their nonlinear system identification [60],[61],[62]. In comparison, the RBFNN has much greater accuracy of prediction and versatility in their choice of base functions [63]. Furthermore, they have fast convergence and less computation load compared to BPNN. These advantages of RBFNN lead to a major research question: Can we incorporate the strengths of the RBFNN to improve the underwater vehicle localization performance of ESKF?

.

3.2. Review of state-of-the-art

The basic form of an on-board navigation system on any underwater vehicle comprises an Inertial Measurement Unit (IMU) that can determine positions by integrating three-axis acceleration and angular velocities [64],[4]. This basic form of navigation suffers from drift, typically 1.8 km per day to 1.5 km per hour based on the grade of IMU [65], [64], [66], which makes them practically impossible to use for long missions. On top of that, most common off-board positioning by Global Positing System (GPS) satellites does not work underwater because of radio frequency attenuation [67]. Alternate communication means based on acoustic positioning are widely used underwater, which suffers from communication uncertainly and delays [2], [68]. On-board aiding sensors, Doppler velocity log, pressure sensor, and magnetometers can also help to reduce the effect of IMU drift, but all these sensors are affected by noise. To improve the navigation accuracy and to minimize disturbance because of noise, EKF-based algorithms are the most commonly used in underwater navigation and localization [69], [70], [71], [72],[73], [74].

On the other hand, the EKF algorithm has its shortcoming in that the accuracy of estimation is reduced under high nonlinear system dynamics. However, many variants of EKF have been proposed in academic research to cater to this problem [75], [76]. Most of the EKF and neural network estimation

algorithms are designed for land-based vehicles. In these approaches, when GPS data is present and valid, the neural network is trained and, when GPS information is not available, the neural network output improves the EKF prediction. For instance, a detailed study [77] proposed a hybrid offline trained RBFNN with time series prediction for measurement update during GPS outage. Another study [78] combined extreme learning machine neural network (ELM) and EKF to bridge the GPS outage. They claimed to have a better real-time performance by improving the computation load. In addition, some authors [79], [80] also suggested using machine learning techniques to improve localization via intelligent communication networks, but their research is limited to land-based applications. Recently, an intelligent methodology was proposed that uses deep learning neural networks with EKF [81]. Moreover, they claimed that, by using Recurrent Neural Networks (RNNs), state estimation can be improved and their model can also work well with low-cost sensors. Nevertheless, they did not incorporate oceanic parameters for state prediction besides the fact that RNNs have a high computational cost.

Another researchers group, in [82], used underwater model-aided dead reckoning to improve EKF response. They calculated aided velocity using an identified surge dynamic model. This work design reached a position accuracy of 92% during external position fix outage. However, using a model in this design makes the system difficult to tune under different sensors and working conditions because models depend on various factors such as size, the weight of the AUV, and the physical characteristics of the sensors. The authors of [78] proposed to embed underwater vehicle dynamic equations in the EKF and estimated error in the navigation. This work claimed to have less computation load and better accuracy compared to a full model integration. However, it is also dependent on the physical parameters of the vehicle and has more implementation complexity.

Likewise, a study [83] compared underwater EKF and its statistical linearization variant, also known as Unscented Kalman Filter (UKF) [84] in their project. They found that the statistical form provides better accuracy in highly nonlinear conditions. Similar results were found in [85]. Nonetheless, the UKF drawbacks include implementation complexity, high computational time and cost, and round-off error [86],[87].

Traditionally, the Kalman filter is used to train RBFNN [88],[89] or in offline training of the radial basis function (RBF) [90]. These methods underperform in uncertain conditions with unmodeled dynamics. The authors of [91] introduced a forgetting factor, which is based on RFBNN, to improve the performance of the Central Difference Kalman Filter (CDKF) for attitude-of-the-satellite estimation. They proposed the range of forgetting at 0.2 to 2 as a multiplier to the Kalman gain, but a limited description of the selection of the forgetting factor was provided. Recently, the RBFNN-aided Kalman Filter was proposed to improve the state estimation accuracy for spacecraft navigation [92]. Moreover, they did not use the multi-sensor fusion of high-rate and low-rate sensors.

3.3. Contributions and novelty of research

The proposed work fills the gap by proposing a novel multi-sensor fusion architecture based on the strengths of the RBFNN and error-state Kalman filter for underwater navigation, which has not been proposed to date to the authors' knowledge. We named this algorithm RBF-ESKF. The augmentation of both algorithms improves the navigation of underwater vehicles in GPS-less environments. Our major contribution is the derivation of a multi-sensor fusion algorithm that improves the accuracy of underwater localization by taking advantage of a Radial Basis Function (RBF) neural network that has the capability of nonlinear universal approximation via recursive learning [93]. Moreover, a simple structure of the RBF network can be trained online with less computation cost compared to the backpropagation neural network (BPNN) [94].

3.4. RBF-ESKF multi-sensor fusion

The proposed modifications improve ESKF performance and make use of the advantages of the RBF neural network. The RBF neural network can approximate any nonlinear function, and they are also known as universal function approximators [95]. The RBF center, its width, and the linear weights for each output neuron are altered at every iteration of a learning algorithm. When each RBF center is as close to the input vector as possible, and the network output error is within the target limit, the training phase is completed. Therefore, it is possible to express the approximation of any functional dependency between variables as a linear combination of the best possible number of RBF neurons with appropriate weight and center. The top-level block diagram of our proposed fusion algorithm is depicted in Figure 3.1

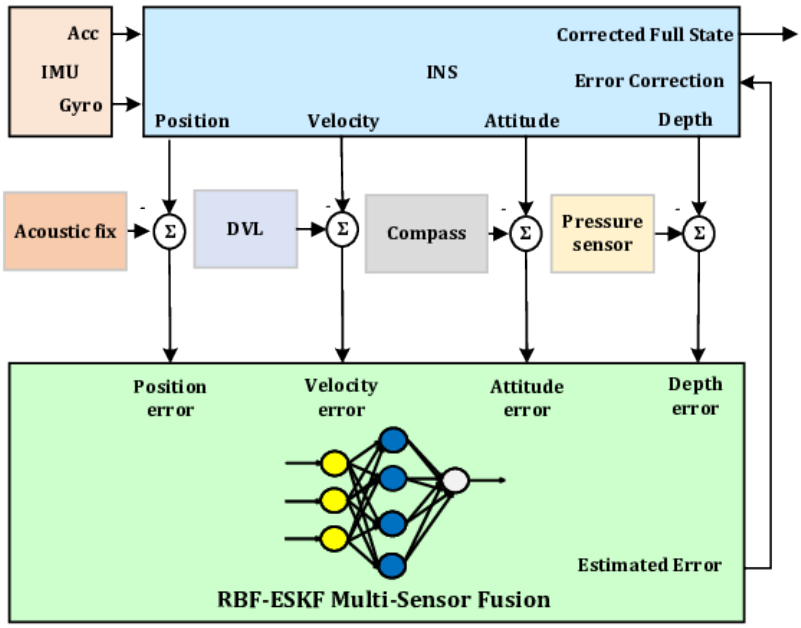

Figure 3.1 Top level diagram of the RBF-ESKF multi-sensor fusion navigation architecture.

As shown by Figure above, the algorithm takes the error of the aiding sensors and INS as input. The RBF-ESKF fusion algorithm, after processing, gives an output to INS for error correction and reset. The RBF neural network used for processing has a basic three-layer structure; input, hidden, and output layer. Furthermore, compared to BPNN, the RBF variants have less computational load and fast online learning. The first layer is the input layer, which provides an interface between the data and the neural network. The data from the input layer to the second hidden layer is transferred in such a manner that the output value of every hidden neuron is inversely related to the Euclidean distance from that neuron's input vector to the RBF neuron's center. The third layer is the output layer, which takes into account the cumulative weights and biases of all RBF neuron outputs.

Several variants of RBF exist in the literature that depends on the application [96]. However, in this work, we used the Gaussian-type RBF function. The weights of neurons in the hidden layer for the Gaussian RBF function indicate the center of the symmetrical Gaussian distribution curve. The novelty of the RBF-ESKF algorithm is that it includes system information within the weight and center learning update rules.

3.4.1 RBF-ESKF mathematical formulation

The underwater vehicle navigation system has nonlinear dynamics and measurement characteristics. The Kalman filter [39], in its basic form, is based on linear system and measurement models, which in reality might not be the case. In our underwater vehicle, navigation equations are used to construct the mathematical model and are not linear with respect to the state variables. This is done by linearization of any predicted trajectory, leading to an error-state model as discussed in Chapter 2. This linear approximation was proven to be incomplete and requires special consideration in underwater navigation, which leads to the derivation of the RBF-ESKF algorithm.

For the mathematical formulation of RBF-ESKF, we start with ESKF implementation, which utilizes definitions of measurement error and state dynamics error, as described by the ESKF Equations [97]. The ESKF algorithm has three major components. The first component is an initialization, in which state, covariance, process, and measurement covariance are initialized randomly[98].

- $\mathbf{x}_0^- =$ Initialization of the state variables.
- $\mathbf{P}_0^- =$ Initialization of covariance matrix.
- $\mathbf{Q}_0 =$ Initialization of process noise covariance.
- $\mathbf{R}_0 =$ Initialization of measurement noise covariance.

where superscript minus $^-$ denotes the a priori state that occurs before innovation is updated. Superscript plus $^+$ denotes the posteriori state after innovation calculation.

The second component is time update, in which the error state and error-state covariance are updated, given by

$$\delta\mathbf{x}_{k+1}^- = \mathbf{\Phi}_k \delta\mathbf{x}_k^- \tag{3.1}$$

where $\delta\mathbf{x}_{k+1}^-$ is the predicted error state and $\mathbf{\Phi}_k$ is the state transition matrix in discrete form.

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_k^{+} \mathbf{\Phi}_k^{\top} + \mathbf{Q}_k \tag{3.2}$$

where $\mathbf{P}_{k+1}^{-}$ is the predicted error covariance and $\mathbf{Q}_k$ is the process noise covariance v_k.

The third component is measurement update, in which the residual of measurement is updated. The residual of measurement $\delta\mathbf{z}_k$ is given by the difference between the actual measurement $\mathbf{z}_k$ and the prediction of measurement $h(\hat{\mathbf{x}}_k)$

$$\delta\mathbf{z}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_k) \tag{3.3}$$

The Kalman $\mathbf{K}_k$ gain is given as

$$\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k^{\top}\left(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^{\top} + \mathbf{R}_k\right)^{-1} \tag{3.4}$$

The posteriori error estimate $\delta\mathbf{x}_k^{+}$ is given as

$$\delta\mathbf{x}_k^{+} = \delta\mathbf{x}_k^{-} + \mathbf{K}_k(\delta\mathbf{z}_k - \mathbf{H}_k\delta\mathbf{x}_k^{-}) \tag{3.5}$$

The expression $(\delta\mathbf{z}_k - \mathbf{H}_k\delta\mathbf{x}_k^{-})$ is referred to as innovation. It is the difference between the error of observation and its expected error, represented by $\mathbf{s}_k$ as

$$\mathbf{s}_k = \delta\mathbf{z}_k - \mathbf{H}_k\delta\mathbf{x}_k^{-} \tag{3.6}$$

The posteriori error-state covariance $\mathbf{P}_k^{+}$ is given as

$$\mathbf{P}_k^{+} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{-} \tag{3.7}$$

The complete and corrected navigation state $\hat{\mathbf{x}}_k^{+}$ can be written as the sum of the error estimate and prior full state estimate $\hat{\mathbf{x}}_k^{-}$ as

$$\hat{\mathbf{x}}_k^{+} = \hat{\mathbf{x}}_k^{-} + \delta\mathbf{x}_k^{+} \tag{3.8}$$

The major assumption for obtaining proper results from ESKF is that the time interval should be short for error calculation and nonlinearity should not be dominant in the calculation of the innovation term. To compensate for the effect of nonlinearity in the innovation equation, we propose to modify it by incorporating an RBFNN. The modified innovation term $\tilde{\mathbf{s}}_\mathbf{k}$ is given as

$$\tilde{\mathbf{s}}_\mathbf{k} = \mathbf{s}_k - \mathbf{W}_\mathbf{k}\mathbf{y}_k \tag{3.9}$$

where the term $\mathbf{W}_\mathbf{k}\mathbf{y}_k$ is the output of the RBF neural network. The term $\mathbf{y}_k$ is the output of the hidden layer of the RBF neural network and $\mathbf{W}_\mathbf{k}$ is the weight matrix that provides the link between output and hidden layers of RBF neural network. These weights can be designed by minimizing the mean square error (MSE) cost function $\mathbf{J}$, defined by

$$\mathbf{J} = \| \mathbf{s}_k - \mathbf{W}_\mathbf{k}\mathbf{y}_k \|^2 \tag{3.10}$$

To improve the estimation of multi-sensor fusion under nonlinear conditions, the Gaussian RBF function utilizes an a priori error-state estimate, which is given as

$$\mathbf{y}_k(i) = \exp\left(-\frac{\|\delta\mathbf{x}_k^- - \mathbf{c}_{ik}\|^2}{2\boldsymbol{\sigma}_i^2}\right), \quad i = 1,2,\dots,N_c \tag{3.11}$$

where $\mathbf{c}_{ik}$ is the neuron center and $\boldsymbol{\sigma}_i$ is the width of the Gaussian RBF function. The weights of the neurons in matrix form $\mathbf{w}_k$ are represented as

$$\mathbf{W}_k = \begin{pmatrix} w_{1k}(1) & w_{1k}(2) & \cdots & w_{1k}(N_c) \\ w_{2k}(1) & w_{2k}(2) & \cdots & w_{2k}(N_c) \\ \vdots & \vdots & \ddots & \vdots \\ w_{Mk}(1) & w_{Mk}(2) & \cdots & w_{Mk}(N_c) \end{pmatrix}$$

where M donates the size of the state vector and N_c represents the number of neuron centers. The size of $\mathbf{W}_k$ is $M \times Nc$. The size of $\mathbf{W_k y}_k$ is $M \times 1$, and the size of $\mathbf{y}_k$ is $N_c \times 1$.

3.4.2 Derivation of weight update of RBF-ESKF

For the weight matrix $\mathbf{W}_k$, we only consider weight associated with the mth-specific output. Thus, the weight vector of the mth output of RBF can be denoted by $\mathbf{w}_{mk}$. Therefore, the weight update rule of the specific mth output is given as

$$\mathbf{w}_{mk+1} = \mathbf{w}_{mk} - \eta_w \frac{1}{2}\frac{\partial \mathbf{J}}{\partial \mathbf{w}_{mk}} \tag{3.12}$$

where η_w is the learning rate, and its value is selected by experimentation. A gradient descent method, namely the steepest descent, is used to minimize the cost function **J** relative to the RBF neural network weight., the gradient of the cost function can be written as

$$\frac{\partial \mathbf{J}}{\partial \mathbf{w}_{mk}} = \frac{\partial \|\mathbf{s_k} - \mathbf{W_k y_k}\|^2}{\partial \mathbf{w}_{mk}} \tag{3.13}$$

The result of the derivative is given as

$$\frac{\partial \mathbf{J}}{\partial \mathbf{w}_{mk}} = -2(\delta \mathbf{z}_k(m) - \mathbf{H}_k(m)\delta\mathbf{x}_k^-(m) - \mathbf{w}_{mk}\mathbf{y}_k)\mathbf{y}_k^T \tag{3.14}$$

Thus, after doing matrix calculas, we get the complete weight update equation:

$$\mathbf{w}_{mk+1} = \mathbf{w}_{mk} + \eta_w(\delta \mathbf{z}_k(m) - \mathbf{H}_k(m)\delta\mathbf{x}_k^-(m) - \mathbf{w}_{mk}\mathbf{y}_k)\mathbf{y}_k^T \tag{3.15}$$

3.4.3 Derivation of center update of RBF-ESKF

The center update rule for the jth element of the mth neuron center is given as

$$\mathbf{c}_{ik+1}(j) = \mathbf{c}_{ik}(j) - \eta_c \frac{1}{2}\frac{\partial \mathbf{J}}{\partial \mathbf{c}_{ik}(j)} \tag{3.16}$$

where η_c is the learning rate of center update. It is determined experimentally. By applying the steepest descent method, we minimized cost function **J** with respect to weight. The result of the derivative is shown in the following equation:

$$\frac{\partial \mathbf{J}}{\partial \mathbf{c}_{ik}(j)} = -2\sum_{i=1}^{M}\left(s_k(m) + \mathbf{w}_{mk}\mathbf{y}_k\right)\cdot\left(\frac{\mathbf{w}_{mk}(i)\mathbf{y}_k(i)(\delta\mathbf{x}_k^{-}(j)-\mathbf{c}_{ik}(j))}{\boldsymbol{\sigma}_i^2}\right) \quad (3.17)$$

Thus, after doing matrix calculas, the new center update equation becomes

$$\mathbf{c}_{ik+1}(j) = \mathbf{c}_{ik}(j) + \eta_c\sum_{i=1}^{M}\left(s_k(m) + \mathbf{w}_{mk}\mathbf{y}_k\right)\cdot\left(\frac{\mathbf{w}_{mk}(i)\mathbf{y}_k(i)(\delta\mathbf{x}_k^{-}(j)-\mathbf{c}_{ik}(j))}{\boldsymbol{\sigma}_i^2}\right) \quad (3.18)$$

It is evident from the weight update equation and center update equation that our algorithm uses system information to train the RBF neural network to overcome the drawbacks of ESKF.

The complementary form block diagram representation of the RBF-ESKF fusion algorithm is shown by Figure 3.2

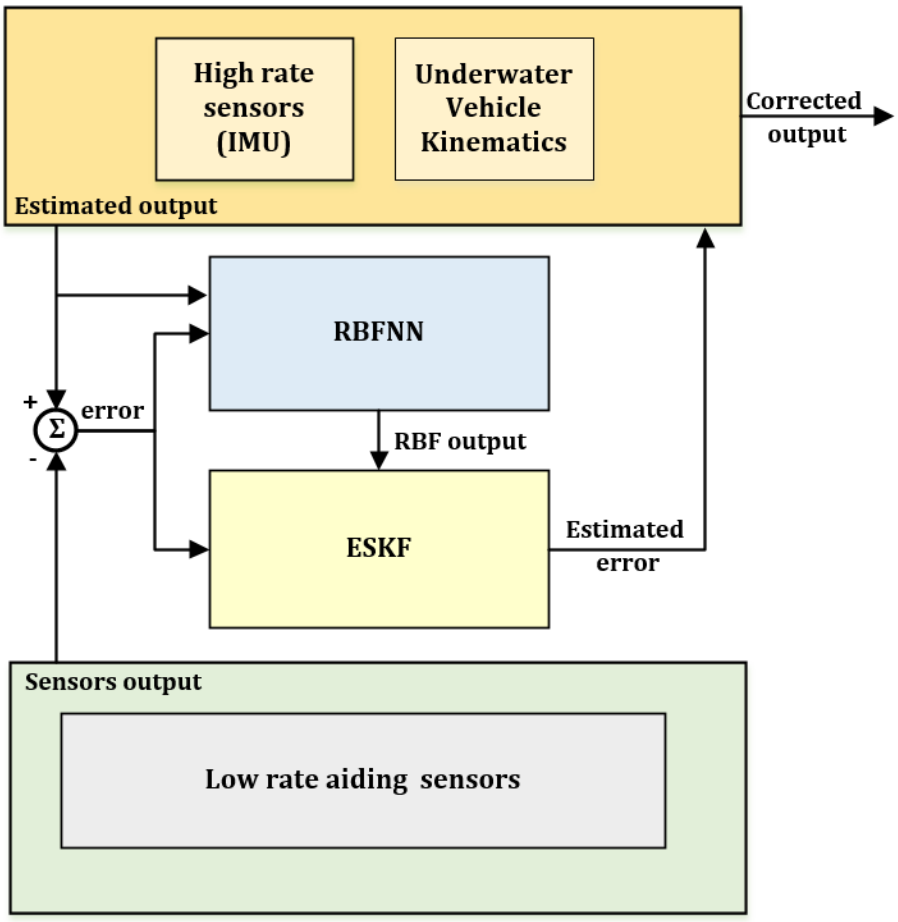

Figure 3.2 Complementary form representation of RBF-ESKF

The vehicle kinematics combined with IMU high-rate sensors of IMU provide the estimated output. This estimated output is then subtracted from low-rate sensors and fed into RBF-ESKF. RBF and ESKF work together to find the best error estimate, which is then added into the underwater vehicle kinematics in a feed-forward fashion to obtain the total state.

Algorithm 3.1

Algorithm 1 shows the iterative steps of the proposed method. The term $\tilde{\mathbf{s}}_\mathbf{k}$ is RBF modified innovation term used by RBF-ESKF to improve the accuracy of filter in highly non-linear conditions

Initialization:

Initialize ESKF variables $\mathbf{x}_0^-, \mathbf{P}_0^-, \mathbf{Q}_0, \mathbf{R}_0$

Initialize RBF variables $\mathbf{w}_0' \mathbf{c}_0, \boldsymbol{\sigma}_0, \eta_w, \eta_c$

Kalman gain update: calculate Kalman gain

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\top + \mathbf{R}_k\right)^{-1}$$

RBF Gaussian function update: Learning non-linearity of the error state vector

$$\mathbf{y}_k(i) = \exp\left(-\frac{\left\|\delta \mathbf{x}_k^- - \mathbf{c}_{ik}\right\|^2}{2\boldsymbol{\sigma}_i^2}\right), \quad i = 1,2,\ldots,N_c$$

Innovation update: Non-linearity influence is minimized by using the output of RBF neural network in innovation term

$$\tilde{\mathbf{s}}_\mathbf{k} = \delta \mathbf{z}_k - (\mathbf{H}_k \delta \mathbf{x}_k^- + \mathbf{W}_\mathbf{k} \mathbf{y}_k)$$

Measurement update: Estimate error state by using innovation term and Kalman gain

$$\delta \mathbf{x}_k^+ = \delta \mathbf{x}_k^- + \mathbf{K}_k \tilde{\mathbf{s}}_\mathbf{k}$$

Error state covariance update.

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

Full State correction Full state is corrected by error adding error estimate.

$$\mathbf{x} = \hat{\mathbf{x}} + \delta \mathbf{x}_k^+$$

RBF Neural Network Weight and Center update: RBF weight update

$$\mathbf{w}_{mk+1} = \mathbf{w}_{mk} + \eta_w (\delta \mathbf{z}_k(m) - \mathbf{H}_k(m) \delta \mathbf{x}_k^-(m) - \mathbf{w}_{mk} \mathbf{y}_k) \mathbf{y}_k^T$$

RBF center update

$$\mathbf{c}_{mk+1}(j) = \mathbf{c}_{ik}(j) + \eta_c \left(\frac{\mathbf{w}_{ik}(j) \mathbf{y}_k(j) \mathbf{c}_{ik}(j)}{\boldsymbol{\sigma}_i^2}\right)$$
$$. \sum_{i=1}^{M} \left(\delta \mathbf{z}_k(i) - (\mathbf{H}_k(i) \delta \mathbf{x}_k^-(i) + \mathbf{w}_{ik} \mathbf{y}_k)\right)$$

Time propagation: Time propagation of error state and covariance

$$\delta \mathbf{x}_{k+1}^- = \boldsymbol{\Phi}_k \delta \mathbf{x}_k^+$$

$$\mathbf{P}_{k+1}^- = \boldsymbol{\Phi}_k \mathbf{P}_k^+ \boldsymbol{\Phi}_k^\top + \mathbf{Q}_k$$

Next iteration (posterior becomes prior)

3.5. Complexity of RBF-ESKF

The proposed algorithm uses RBFNN to enhance the performance of underwater vehicle localization with a slight increase in time and space complexity compared to ESKF due to matrix multiplications. Compared to BPNN and deep learning neural networks, RBFNN has less complexity because of its simple three-layer structure because the time and memory space complexity of the neural networks is directly related to the structure and number of layers. However, faster matrix multiplication algorithms such as the Strassen algorithm [99] can be used to decrease execution time. Table 3.1 shows the structure of the RBFNN used in this work.

Table 3.1 RBF neural network structure with a Gaussian activation function.

RBFNN	Parameters
Input layer neurons	20
Hidden layer neurons	50
output layer neurons	20
Learning rate of weights	0.001
Learning rate of centers	0.001

3.6. Results and discussion

In order to compare performances, the proposed algorithm and ESKF were simulated in three different realistic scenarios. As the ESKF structure was modified by RBF in our fusion algorithm, low-level functions were written for simulation. The noise specifications of the sensors used in this work are comparable to their datasheets. The main purpose of the simulation was to compare the maximum error (max), and root means square error (RMSE) of position, velocity, and attitude. Practical failure mode tests cases were developed and simulated with DVL and acoustic positioning loss of measurements for a short duration. The simulation results were compared with conventional ESKF for performance evaluation. Furthermore, for simulation, the assumption was made that underwater vehicles can move in any direction and with any roll, pitch, and yaw angle. A reference trajectory of the vehicle was generated by angular velocities and acceleration.

To consider the effects of random variations in the accuracy of the fusion algorithms, a Monte Carlo simulation was used. The test consisted of 100 runs. Two filters processed the same data during the test to ensure a fair comparison. For all three cases, the same RBF neural network structure was used, as listed in Table 3.1. For the simulation, the RBF weights, centers, and sigma were initialized randomly.

3.6.1 Test Case 1: Normal working condition

In the first case, the vehicle was considered to be working in a normal operating mode without any onboard and offboard sensor failure. To simulate a real situation, the noise and drift characteristics of the sensors used for simulation were almost the same as listed in the manufacturer's documentation stated in Chapter 2. Both ESKF and RBF-ESKF were tested on the same operating conditions. The performances of ESKF and RBF-ESKF are compared by running 100 Monte Carlo simulations side by side in Table 3.2.

Table 3.2 ESKF and RBF-ESKF results with all sensors working in normal condition

	ESKF	RBF-ESKF
North Position Max error	1.1509	0.32578
East Position Max error	0.8218	0.45685
Down Position Max error	0.0081786	0.0071222
North Position RMSE	0.446124	0.1464
East Position RMSE	0.3122	0.1844
Down Position RMSE	0.0040719	0.002709
Sum Position RMSE	0.76239	0.333509
North Velocity Max error	0.038185	0.01844
East Velocity Max error	0.0045943	0.0037403
Down Velocity Max error	0.0032507	0.0022476
North Velocity RMSE	0.039455	0.011407
East Velocity RMSE	0.033652	0.012919
Down Velocity RMSE	0.0035526	0.020227
Sum Velocity RMSE	0.0766596	0.0263487
Roll Max error	0.13268	0.060874
Pitch Max error	0.1809	0.16171
Yaw Max error	0.46624	0.36001
Roll RMSE	0.00039554	0.00019122
Pitch RMSE	0.00055597	0.00036354
Yaw RMSE	0.00049874	0.00035049
Sum Attitude RMSE	0.00145025	0.00122578

From the results of the north position prediction displayed in Table 3.2, it can be observed that ESKF has an almost three times higher maximum error than that of the RBF-ESKF. Furthermore, for the east position, the ESKF maximum error was twice as high as the RBF-ESKF. In the case of a maximum error in the prediction of depth, the performance of the two filters is approximately identical. The RBF-ESKF RMSE for position estimation was even better, almost twice as high. Significant improvement was seen in the north, where it was three times higher than the one achieved by the ESKF.

Overall, the RMSE sum for the RBF-ESKF position estimate was approximately two times better than that of the ESKF.

The estimation of velocity showed considerable improvement compared to the ESKF. The overall northern velocity error was almost double that of the ESKF compared to the RBF-ESKF. Small but noticeable improvements were seen in the maximum error for east and depth velocities. The RMSE of the RBF-ESKF was roughly two times better than the ESKF for the north, east, and down velocities. Overall, with our proposed fusion algorithm, the sum of all RMSE states was almost three times better than ESKF.

Overall, in terms of attitude, relative to ESKF, the sum of all state RMSEs improved significantly by about double as much.

Figure 3.3 and Figure 3.4 show the 2D and 3D trajectories. It can be seen that the trajectory is not linear. The position, velocity, and attitude errors are estimated on this trajectory. The star symbol in the 2D trajectory shows an acoustic fix from the external source. The performances of these ESKF and ESKF-RBF were evaluated by how close the estimated path was to the actual value.

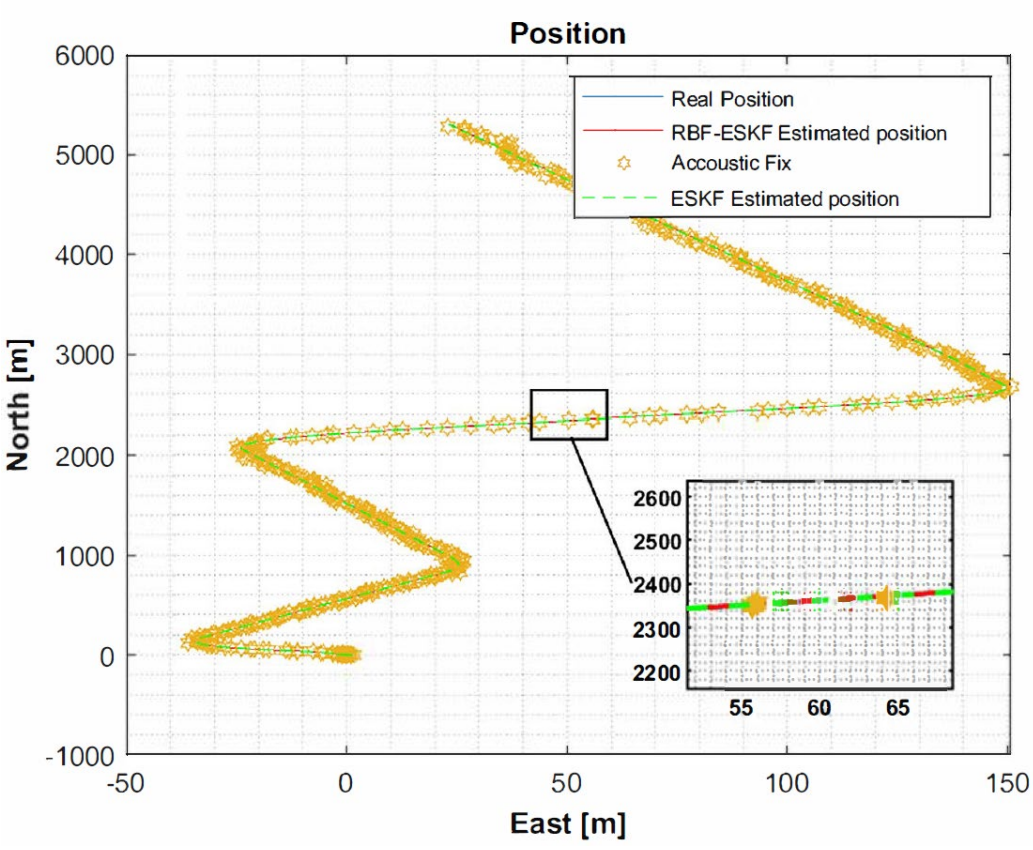

Figure 3.3 Comparison of the 2D trajectory of an underwater vehicle in the east and north directions

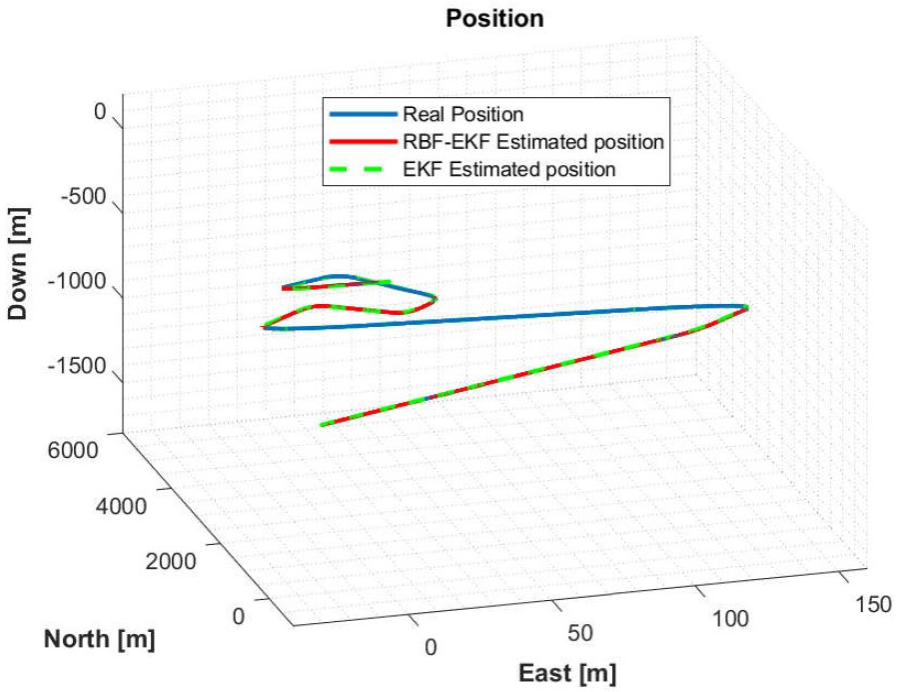

Figure 3.4 ESKF-RBF and ESKF comparison of the 3D trajectory of an underwater vehicle

Figure 3.5 to Figure 3.10 show simulation results of ESKF and RBF-ESKF for time 0–2500 s (x-axis)

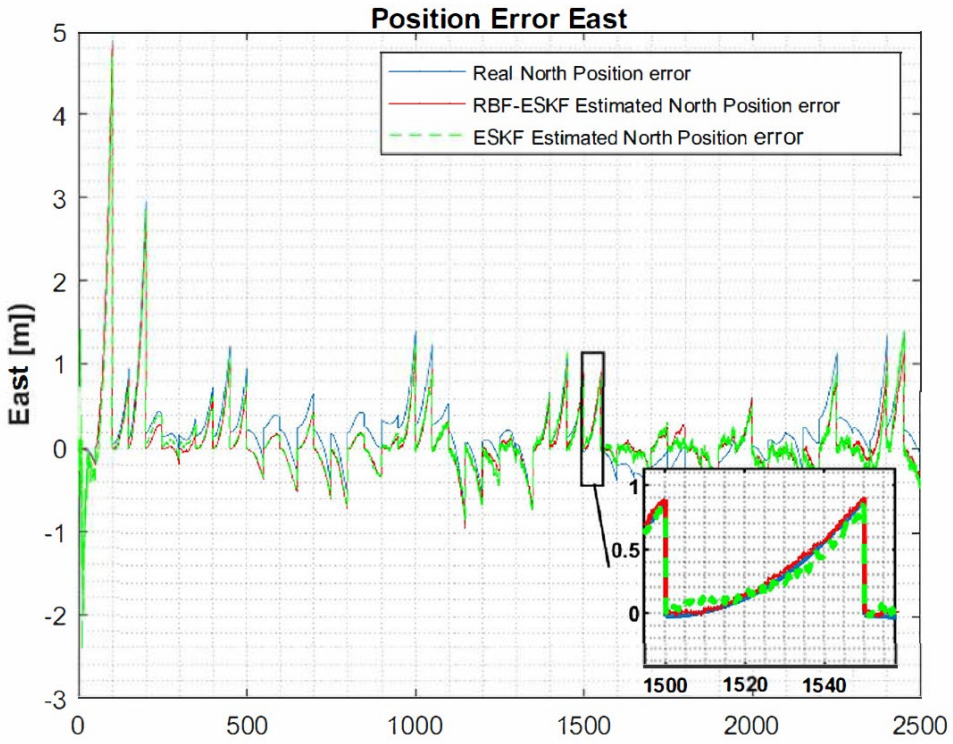

Figure 3.5 ESKF-RBF and ESKF comparison for east position error

*

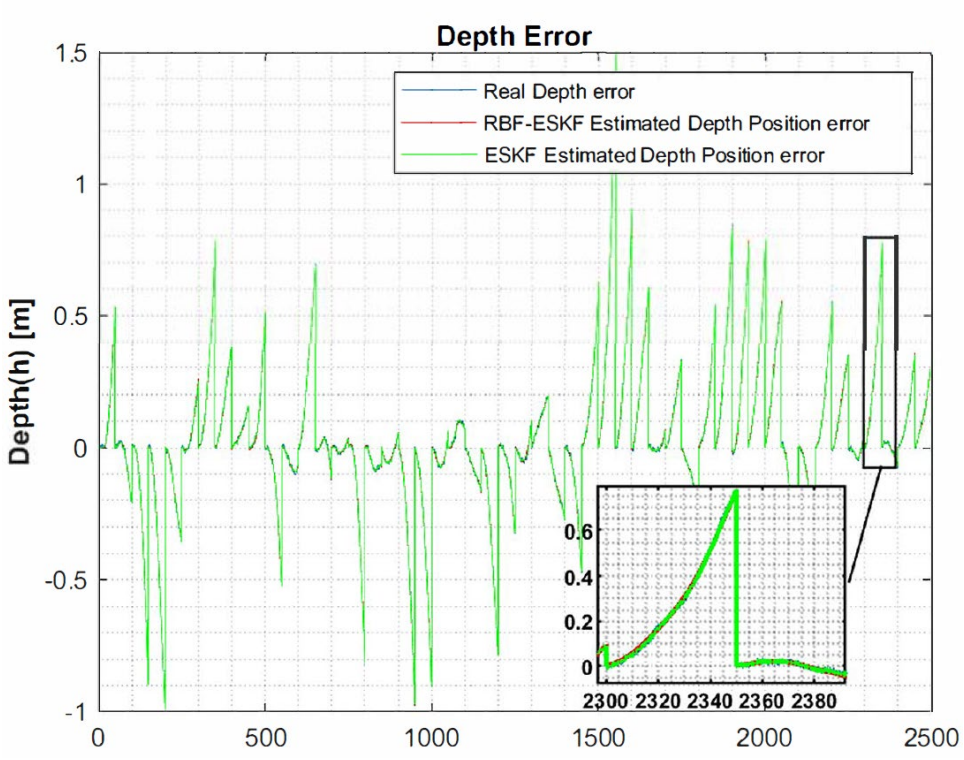

Figure 3.6 ESKF-RBF and ESKF comparison for depth position error

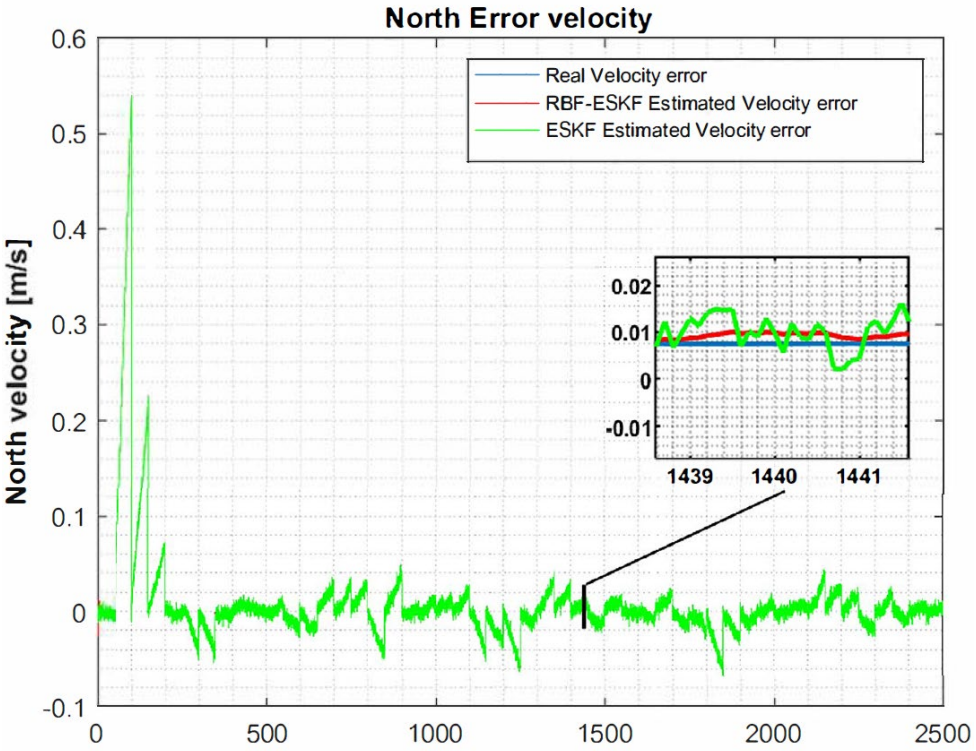

Figure 3.7 RBF-ESKF and ESKF estimation for the north velocity error

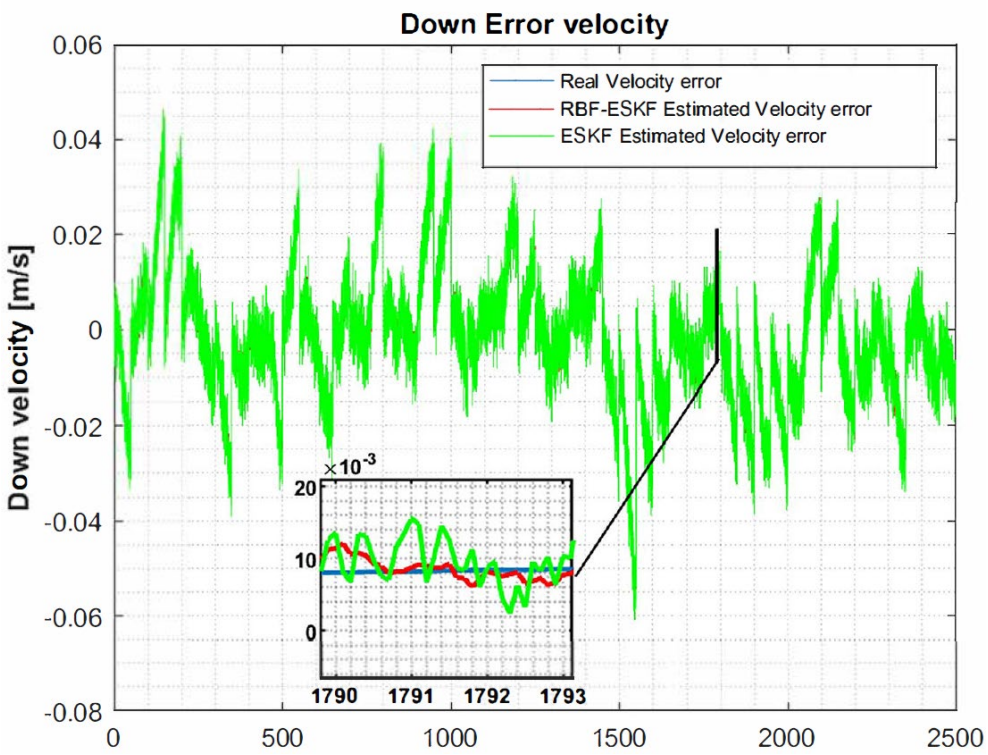

Figure 3.8 ESKF-RBF and ESKF comparison for down error velocity

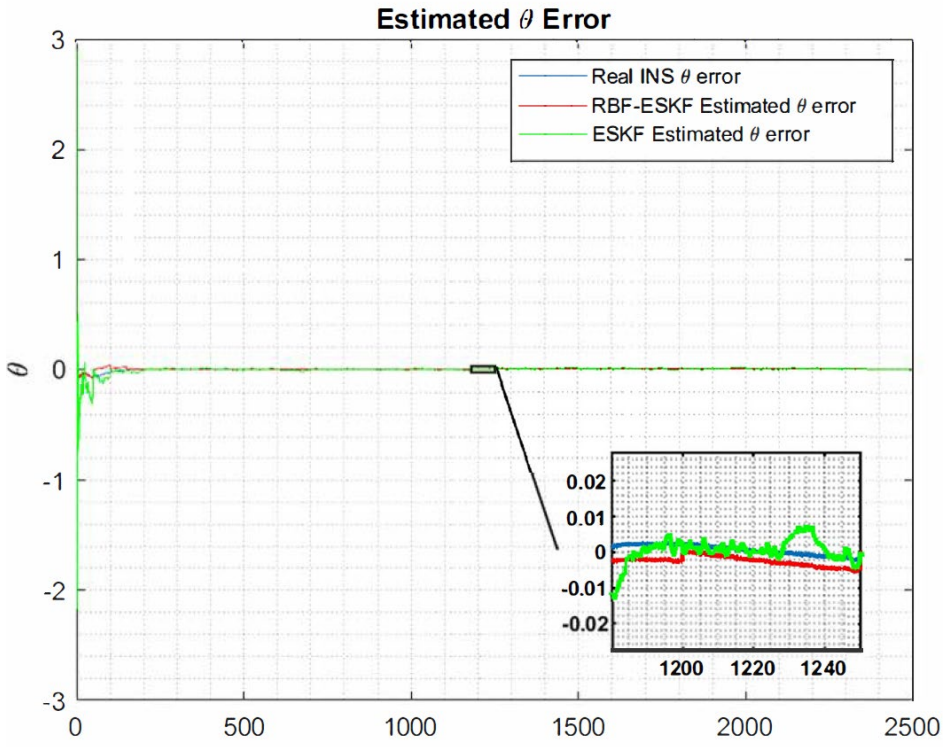

Figure 3.9 RBF-ESKF and ESKF estimation for the pitch error

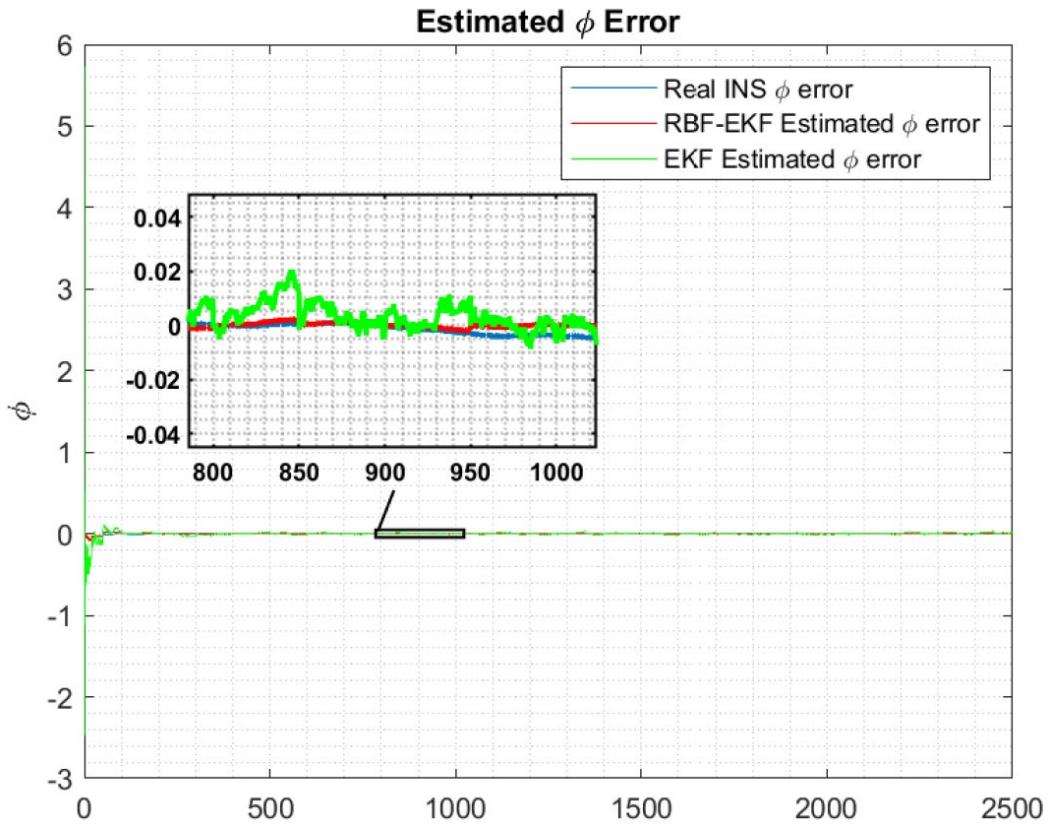

Figure 3.10 RBF-ESKF and ESKF comparison for the roll error

For positions, as shown by Figure 3.5 and Figure 3.6, RBF-ESKF estimation for the east positions and depth error is close to the actual error value. Figure 3.7 and Figure 3.8 show error velocity for down and north has minimum error as compare to ESKF. Euler angles errors for roll and pitch are shown in Figure 3.9 and Figure 3.10, it is evident from the figers that ESKF has an oscillatory response that contributes to compromised accuracy compared to RBF-ESKF.

3.6.2 Test Case 2: Acoustic fix not available

In this case, the robustness of the multi-sensor fusion algorithm was tested by simulating the loss of the underwater acoustic fix for a short period. The unavailability of the position information from acoustic fix mostly influenced the position estimate of both filters, predominantly ESKF. In this case, the position estimate was only available from the integration of the DVL velocity, which compensated the acoustic fix loss effect and reduced the drift error. The detailed comparison is shown for loss of acosutic fix for short period by running 100 Monte Carlo simulations.in Table 3.3.

Table 3.3 Performance Comparison of ESKF and RBF-ESKF with loss of acoustic fix

	ESKF	RBF-ESKF
North Position Max error	5.2115	1.3849
East Position Max error	1.4275	0.4663
Down Position Max error	0.033167	0.016192
North Position RMSE	0.94036	0.50629
East Position RMSE	0.66511	0.40109
Down Position RMSE	0.005613	0.0039411
Sum Position RMSE	1.611083	0.9113211
North Velocity Max error	0.040376	0.029489
East Velocity Max error	0.0036513	0.0030517
Down Velocity Max error	0.0061216	0.0036592
North Velocity RMSE	0.0439324	0.21818
East Velocity RMSE	0.065722	0.035722
Down Velocity RMSE	0.0065451	0.0039131
Sum Velocity RMSE	0.1161995	0.0614531
Roll Max error	0.2567	0.1231
Pitch Max error	0.41582	0.19178
Yaw Max error	0.5494	0.46001
Roll RMSE	0.0006135	0.00020485
Pitch RMSE	0.00052358	0.00011917
Yaw RMSE	0.00051363	0.0003413
Sum Attitude RMSE	0.00169071	0.00117532

It can be noted from *Table 3.3* that the maximum error of RBF-ESKF was almost five times better for the north and east directions as compared to ESKF. The down position estimate was also two times better than that of ESKF. The overall RMSE in all three directions is considerably improved by almost one and a half times that of ESKF. Overall, compared to normal working conditions, the position accuracy had a detrimental effect, but RBF-ESKF has proven to be more robust.

In contrast to standard operating conditions, the velocity estimation was marginally influenced by acoustic fix unavailability. The maximum ESKF error was worse than that of RBF-ESKF for velocity in all directions. For RBF-ESKF, the RMSE of velocity was significantly better by almost two times that for ESKF. Overall, with our proposed method of fusion, the sum of all RMSE velocity states was almost two times better than ESKF.

For RBF-ESKF, the RMSE of the roll, pitch, and yaw angles were substantially better by about one and a half times that of ESKF. The maximum roll, pitch, and yaw angle errors were two times better for RBF-ESKF. Overall, the sum of all states for RBF-ESKF was better than that for ESKF.

shows Simulation results of ESKF and RBF-ESKF for time 0–3000 s (x-axis) in case 2.

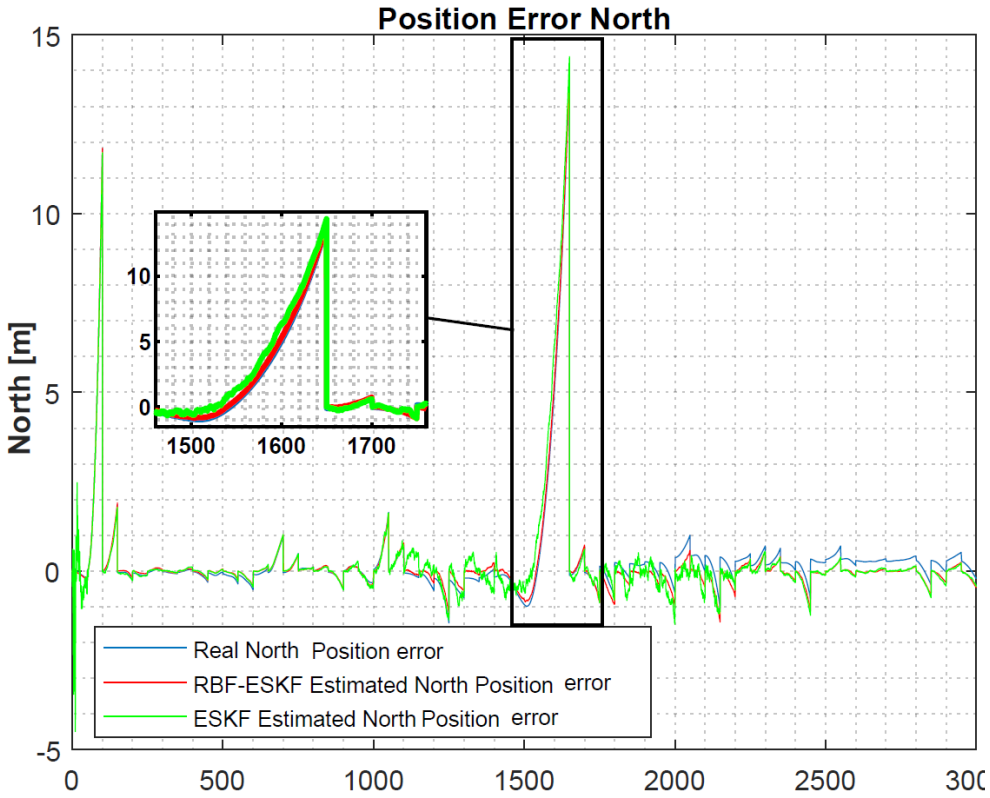

Figure 3.11 North position error with the loss of acoustic fix

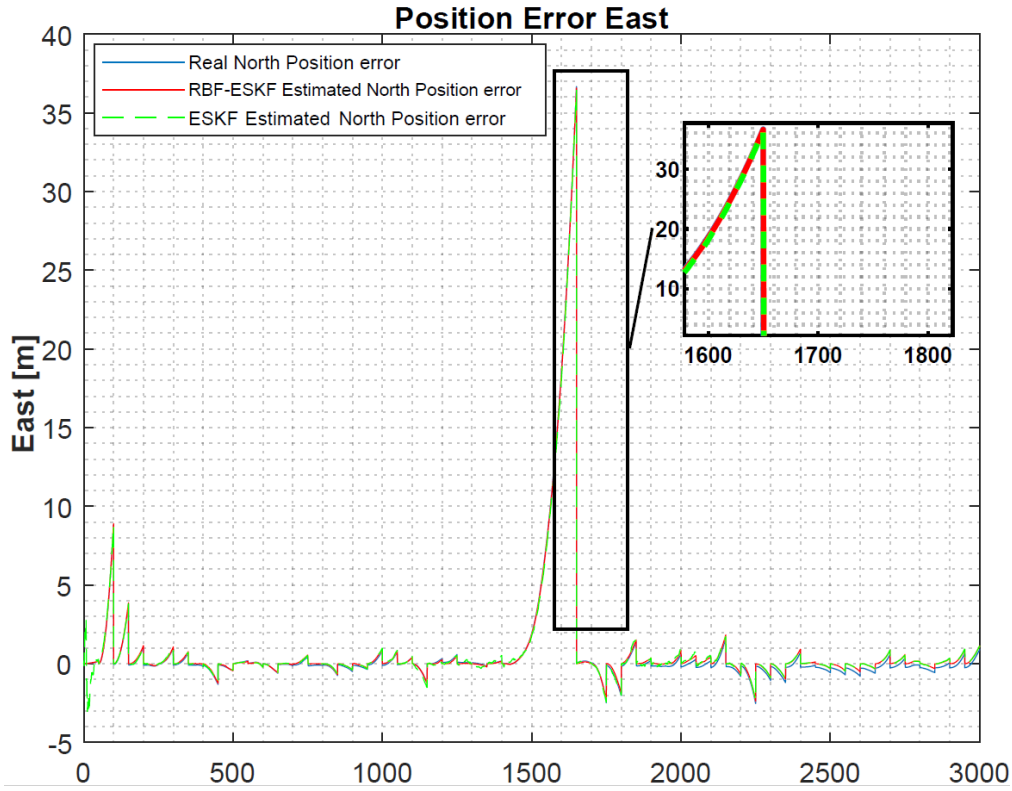

Figure 3.12 East position error with the loss of acoustic fix

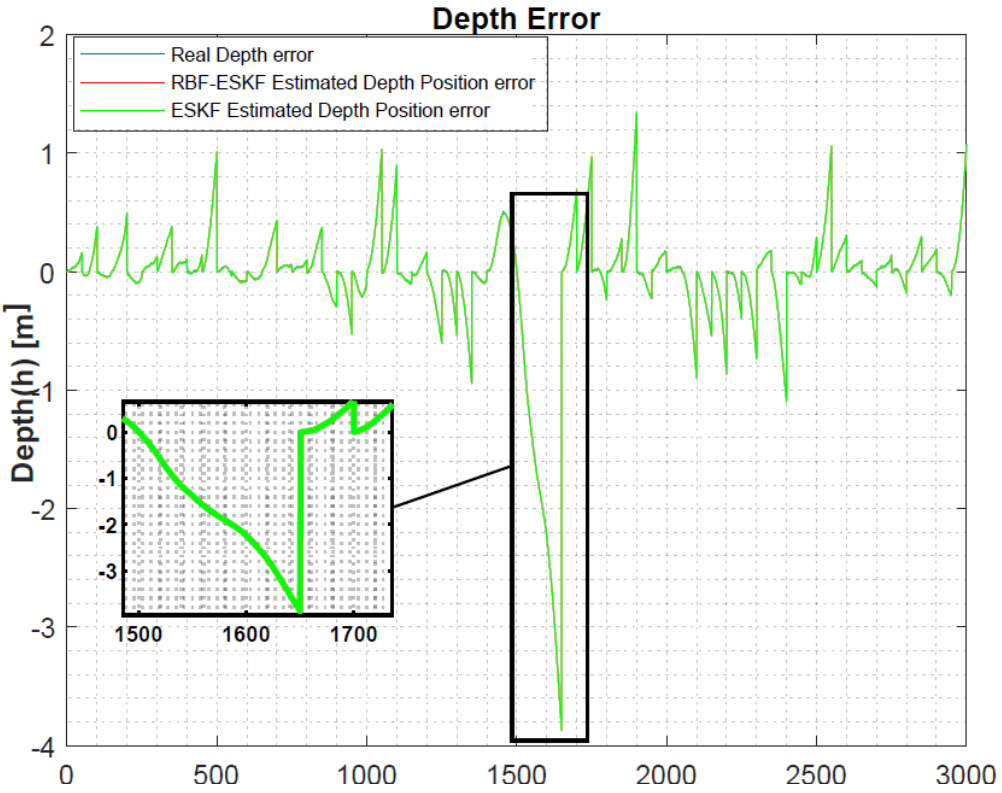

Figure 3.13 Depth error with the loss of acoustic fix

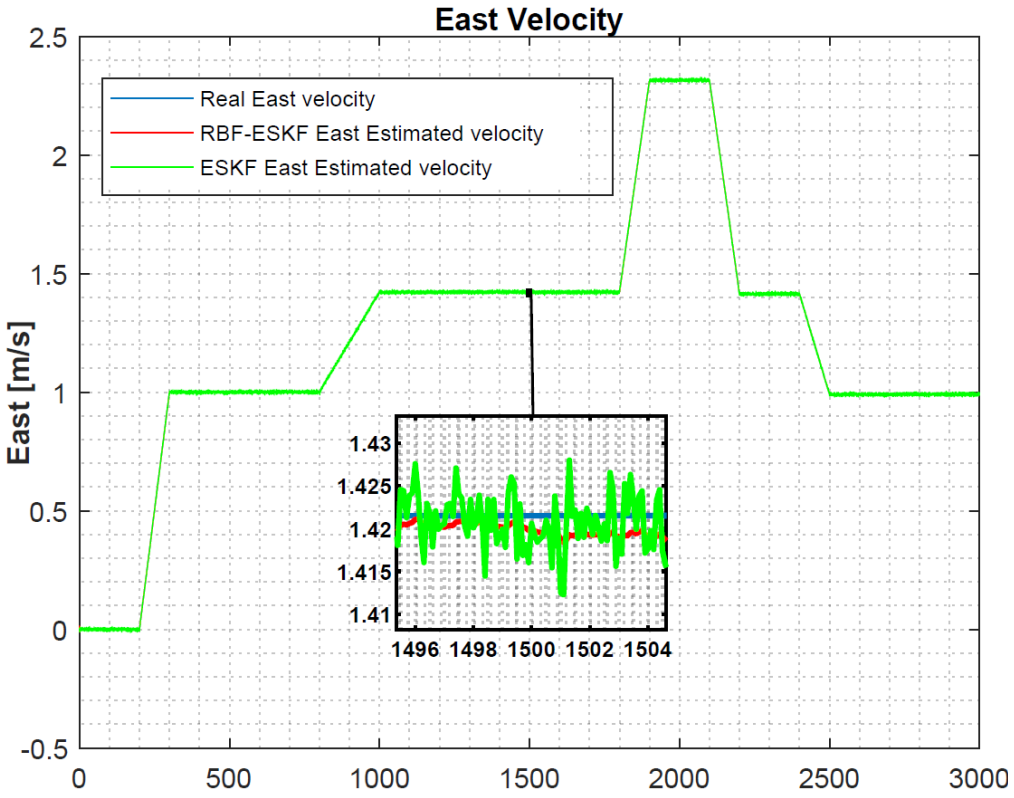

Figure 3.14 East velocity with the loss of acoustic fix

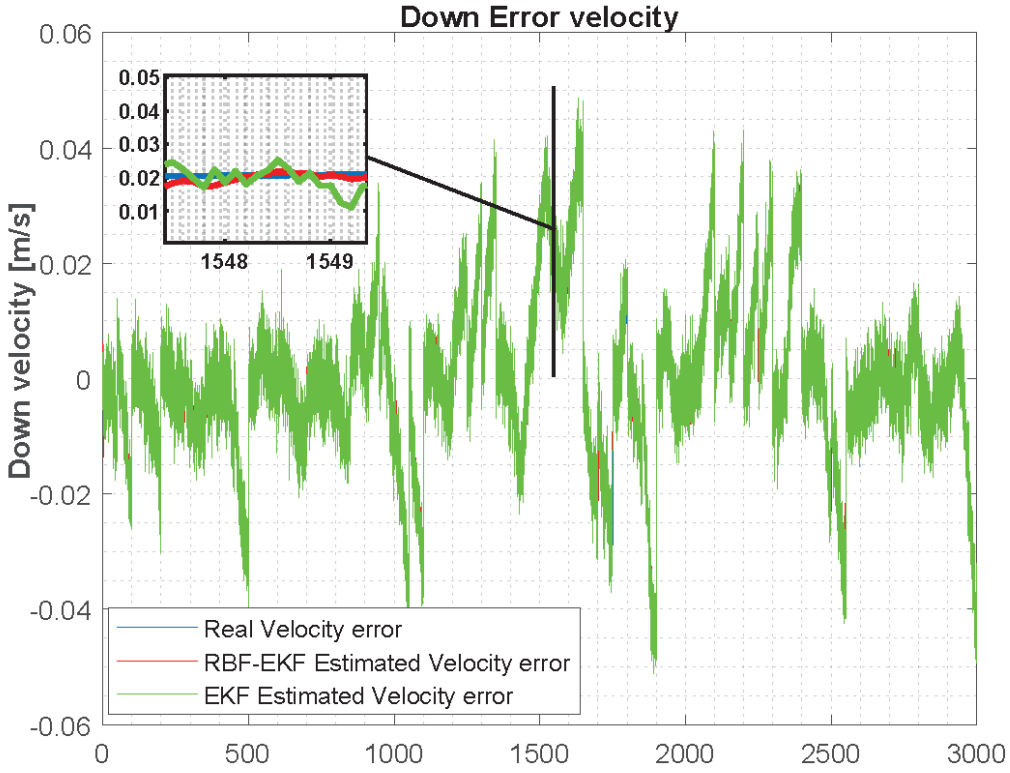

Figure 3.15 Down error velocity with the loss of acoustic fix

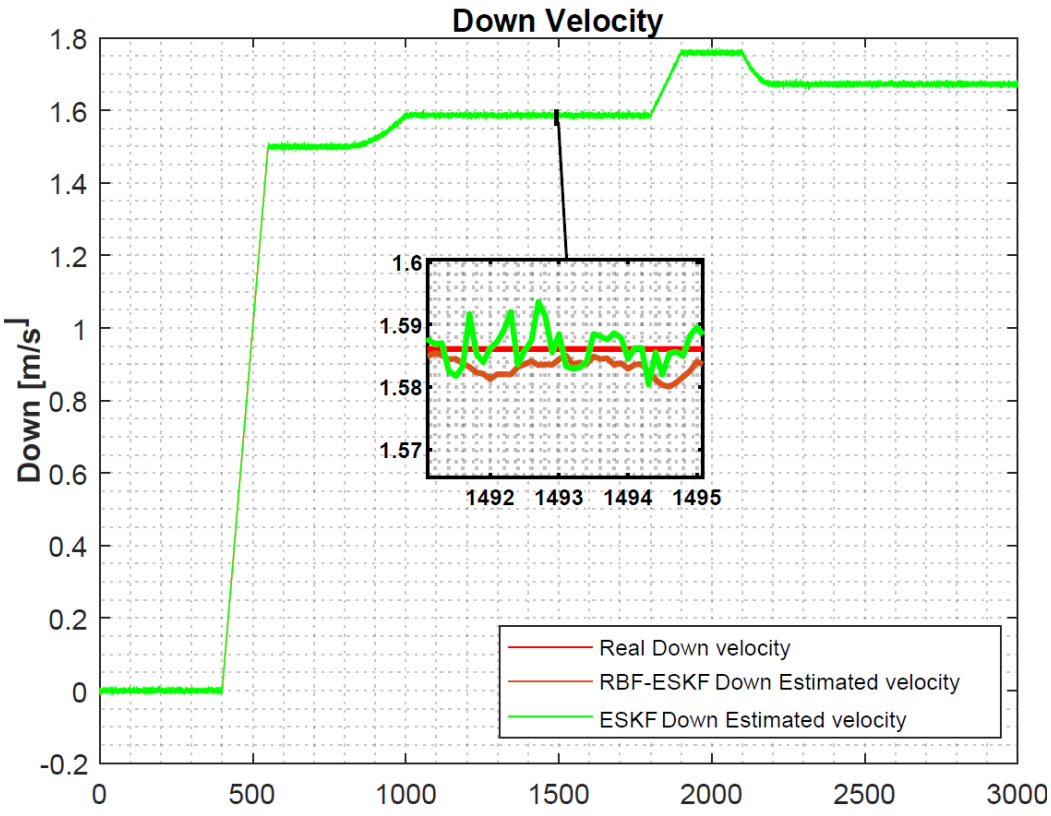

Figure 3.16 Down velocity with the loss of acoustic fix

As seen from the above figures for case 2, the position error significantly grows with a loss of acoustic fix; however, for the proposed algorithm, estimation is close to the actual error. Furthermore, it can be noted from the figures that RBF-EKF has a minimal effect of acoustic fix loss on velocity estimation, and the response is much smoother than that of ESKF.

3.6.2 Test Case 3: DVL unavailable

In this case, the robustness of both filters was tested when DVL was not available for a short duration. The velocity estimate had a larger influence than the position estimate because, when DVL was not available, it was calculated by taking the derivative of the position measurement, which suffers from noise amplification from the differentiation process. Moreover, acoustic fix measurement bias was negatively influenced by DVL measurements that contribute to increasing the position error. Table 3.4 compares the performance robustness multi-sensor fusion algorithm with DVL failure by running 100 Monte Carlo simulations.

.

Table 3.4 Performance comparison of ESKF and RBF-ESKF with the Doppler velocity log (DVL) measurement unavailable for a short duration

	ESKF	RBF-ESKF
North Position Max error	1.9455	0.90707
East Position Max error	0.91649	0.63511
Down Position Max error	0.0084106	0.0072865
North Position RMS error	0.669324	0.41818
East Position RMS error	0.53722	0.30722
Down Position RMS error	0.0049455	0.0035131
Sum Position RMS error	1.2114895	0.7289131
North Velocity Max error	0.105751	0.051603
East Velocity Max error	0.099018	0.061305
Down Velocity Max error	0.0178454	0.00897
North Velocity RMS error	0.191296	0.09809
East Velocity RMS error	0.15337	0.079982
Down Velocity RMS error	0.0092455	0.005387
Sum Velocity RMS error	0.3539115	0.183459
Roll Max error	0.29935	0.171758
Pitch Max error	0.51906	0.210442
Yaw Max error	0.66041	0.42246
Roll RMSE	0.00066481	0.000450075
Pitch RMSE	0.00060017	0.00042717
Yaw RMSE	0.00062205	0.000422102
Sum Attitude RMSE	0.00188703	0.001299347

It can be observed from the above data that position estimate in this case was better than case 2 but slightly less accurate than normal working conditions. However, in the north direction, ESKF maximum error was almost two times worse and almost one and a half times worse for east and depth. Furthermore, the RBF-ESKF RMSE for the position was notably better than that for ESKF by around one and a half times in all directions. Thus, the position of estimation for the overall mission was improved in all directions by the RBF-ESKF algorithm.

The velocity estimation without DVL was less accurate compared to case 1 and case 2. However, the overall results of our method are much better than those of ESKF, which were almost doubly improved with respect to the maximal error and RMSE. In addition, the sum of all estimated velocity states from RBF-ESKF in all directions was approximately two times better than that of ESKF. Hence, the RBF-ESKF velocity estimation was more robust than that of ESKF. The RBF-ESKF attitude estimation of the roll, pitch, and yaw angles had a lower maximum estimation error. Moreover, the RMSE for RBF-ESKF showed considerable improvement. Overall, the sum of all estimated attitude states was one and half times better than that of ESKF.

Figure 3.17 to Figure 3.22 simulation results of ESKF and RBF-ESKF for case 3. In the following figures of RBF-ESKF and ESKF are compared when the DVL measurement was not available for a short duration from 1900 to 2000 s. The performances of ESKF and ESKF-RBF were evaluated on a similar trajectory as test case 1.

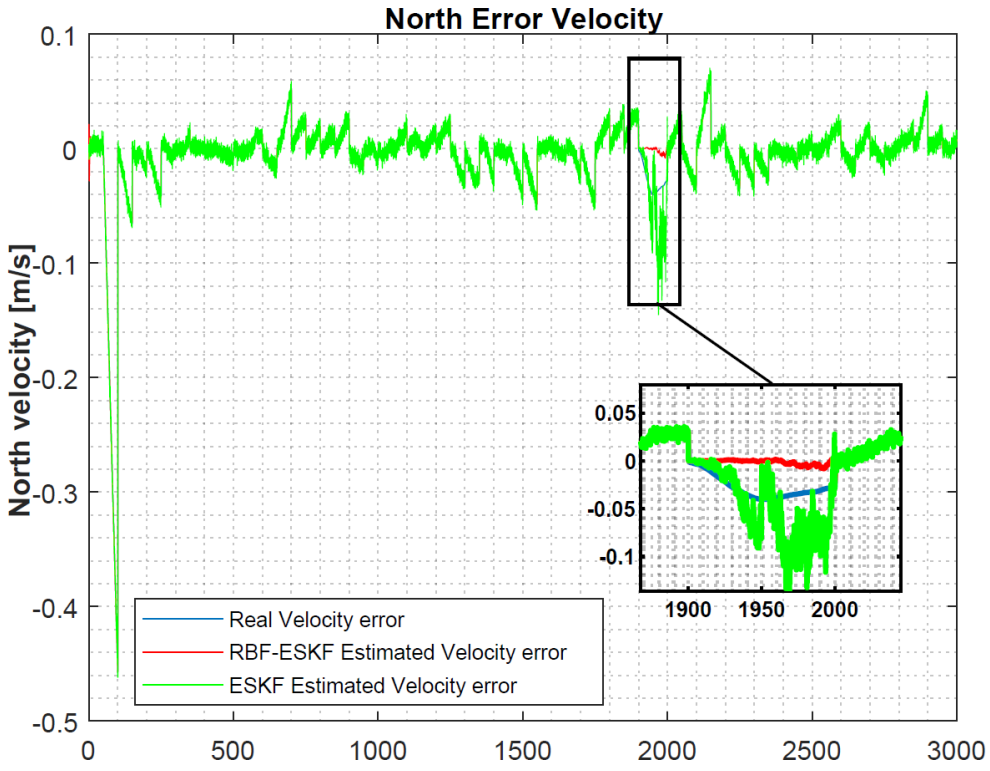

Figure 3.17 North velocity error with loss of DVL measurements

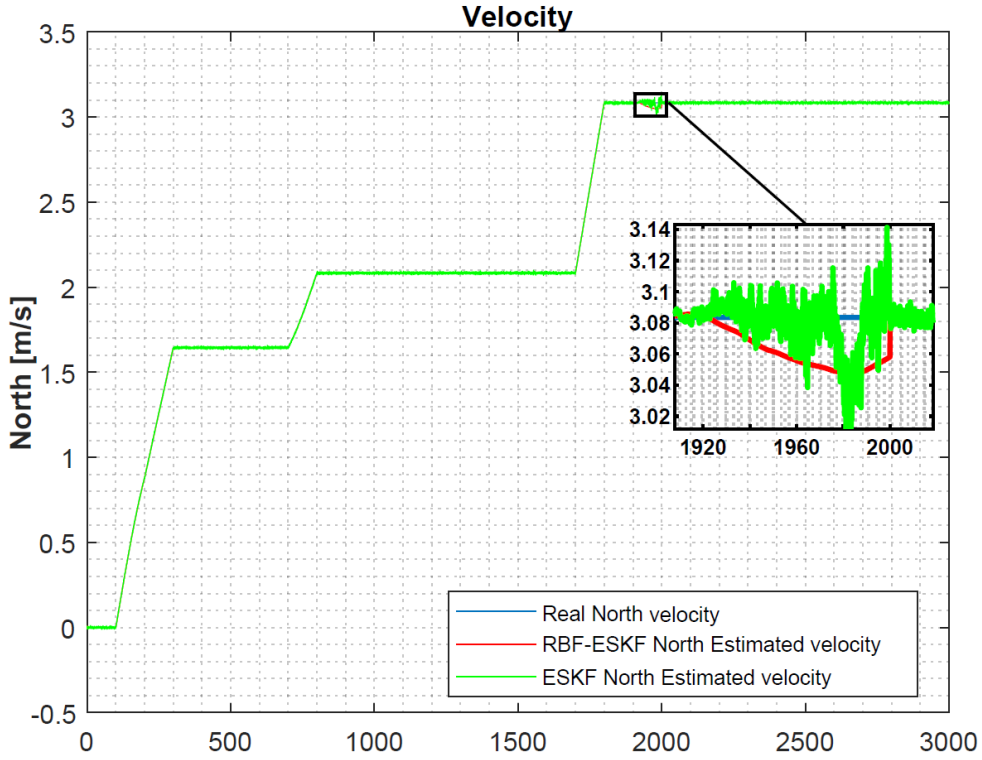

Figure 3.18 North velocity with loss of DVL measurements

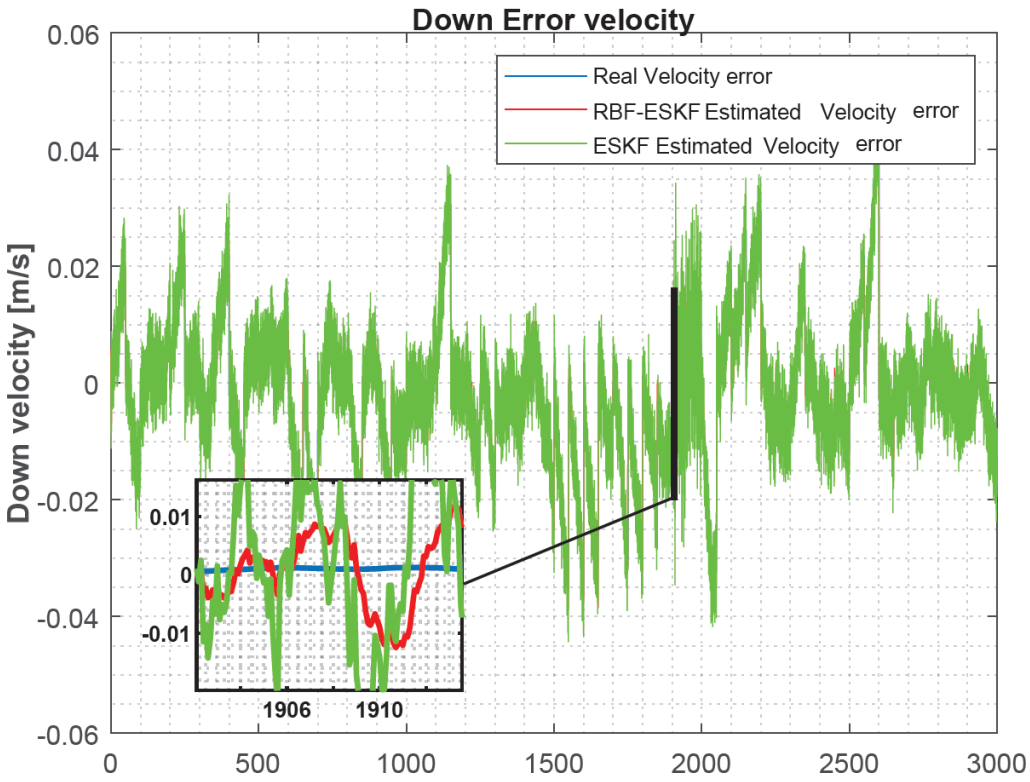

Figure 3.19 Down velocity error with loss of DVL measurements

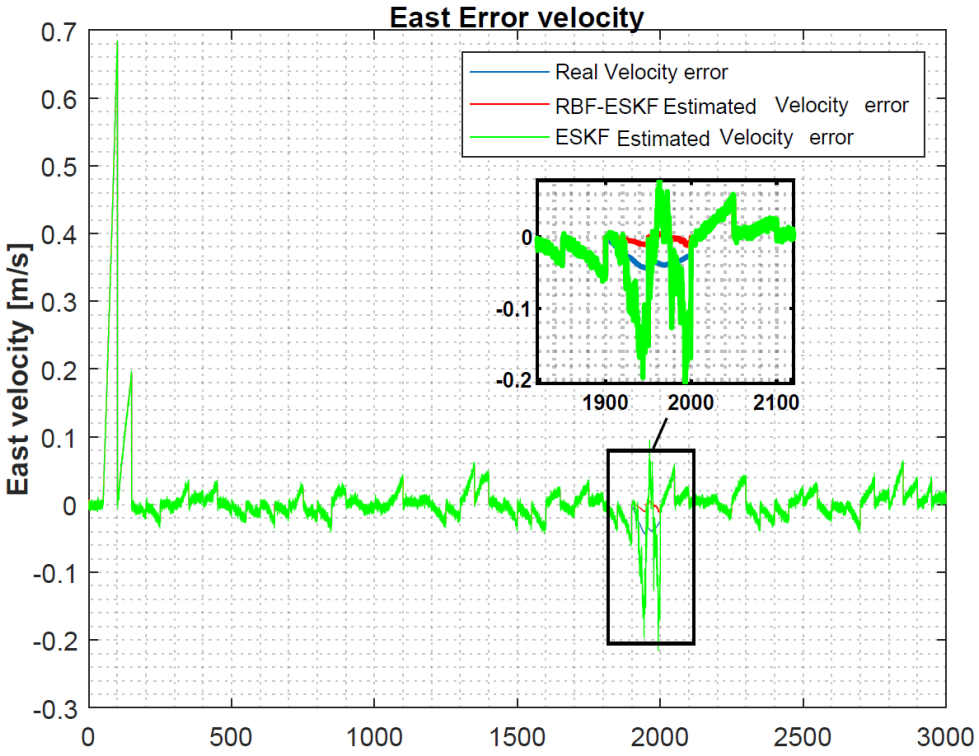

Figure 3.20 East velocity error with loss of DVL measurements

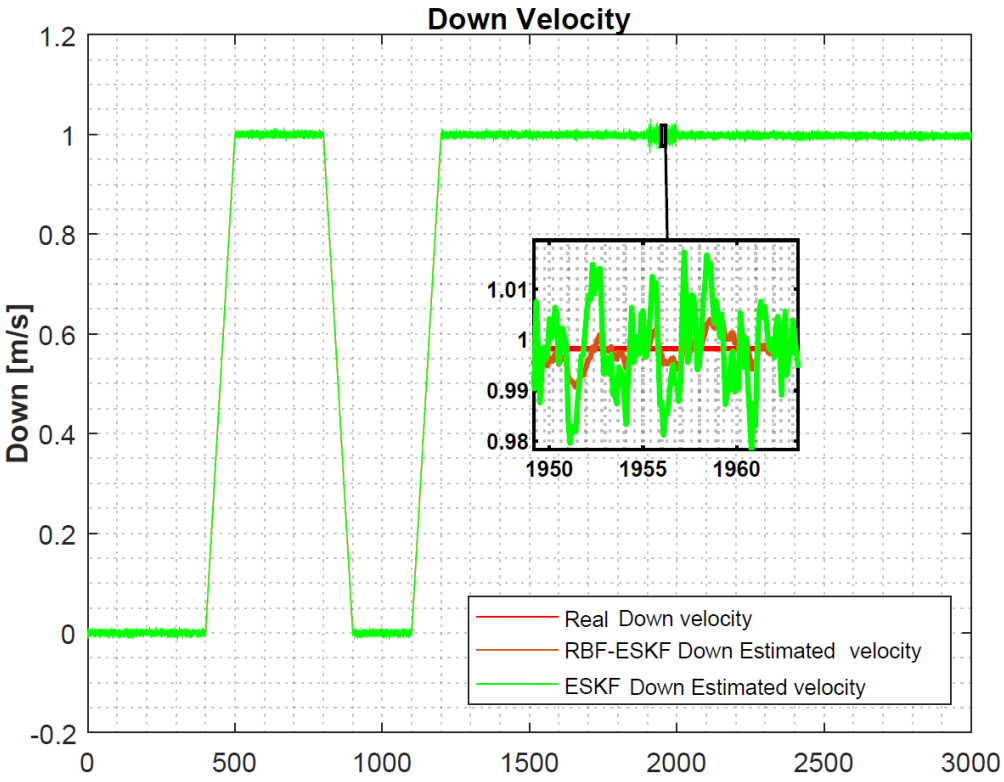

Figure 3.21 Down velocity with loss of DVL measurements

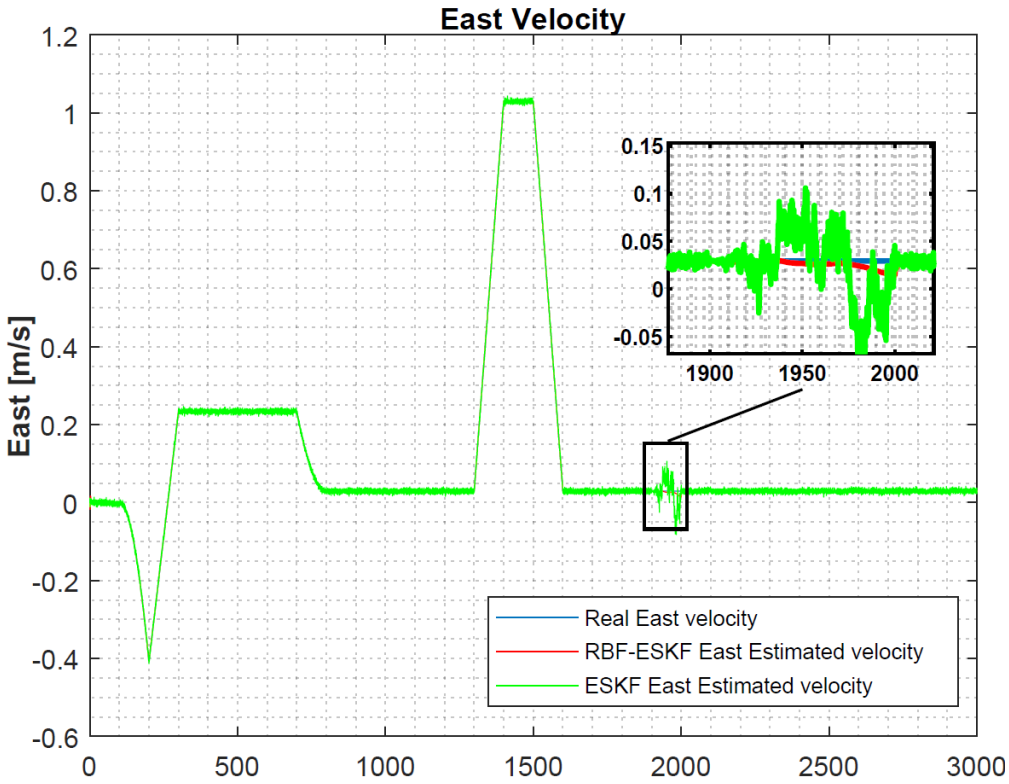

Figure 3.22 East velocity with loss of DVL measurements

As shown from above figures the error increases, nonlinearity in error also increases. The velocity error for the north, east, and downward directions increase significantly with DVL loss. The proposed velocity error estimate of the algorithm is similar to real error. RBF-ESKF has a smoother response and faster convergence of the full state velocity estimation.

The time complexity of ESKF and ESKF-RBF was tested by running the algorithms on an Intel I7 CPU with 4 GB RAM without any GPU. The testing software used was Matlab 2020b on Windows 7 platform. The timing comparison of both methods is given in Table 3.5

Table 3.5 Execution time (seconds)

ESKF	ESKF-RBF
0.0028	0.0039

On a high-speed microcontroller or field-programmable gate array FPGA, the execution time difference will be further reduced. Furthermore, the speed of surveying the type of underwater vehicle is in the range of 2 km/h to 10 km/h; this difference has no significant effect on navigation and localization.

The work tried to fill the gap by proposing a novel ESKF and RBF-augmented fusion solution, which was assessed in three different underwater cases. The ESKF performance strongly relies on the knowledge of the system models and noise properties, which was degraded by nonlinearity. The errors in the RBF-ESKF are smaller than the errors in the ESKF because of the recurcive learning of RBF. Moreover, the fusion algorithm based on RBF-ESKF with help from the aiding sensors was able to correct the drift problems in the INS with better accuracy.

Nevertheless, the proposed algorithm is not limited to underwater; it can also be used in other applications [100],[101] such as improving aircraft navigation and tracking by using aerial sensors. Moreover, autonomous ground vehicles are another area where this method can be employed. Furthermore, by improving the Kalman filer response, our methodology can also enhance accuracy satellite attitude estimation [102].

3.7. Conclusions

This chapter discussed the performance of the proposed algorithm RBF-ESKF for underwater vehicle localization. The primary aim of this work is to take advantage of the RBF neural network to improve the estimation performance of the conventional ESKF for the position, velocity, and attitude of an underwater vehicle. It contrasts outcomes with ESKF in three separate simulations, showing that RBF-ESKF performs better in estimating position, velocity, and attitude. Why the standard ESKF has an inferior performance is because it is designed by employing first-order Taylor series approximation in the error covariance matrix estimation, which results in a decrease in estimation accuracy under high nonlinearity. Thus, important information about the dynamic of underwater is lost because of this realization. However, RBF-ESKF efficiently handles nonlinearity due to its inherent capability for nonlinear function approximation and learning ability.

The research also compared robustness in cases when there is no available position information from the acoustic fix. Here, relative to ESKF, RBF-ESKF demonstrated better accuracy. When acoustic fix becomes available, RBF-ESKF converges quickly. In addition, when DVL fails due to short durations, RBF-ESKF also demonstrates less estimation error.

Chapter 4 : Underwater Vehicle Positioning by Correntropy-Based FMSF

This chapter is based on our work on underwater vechile positioning using correntroppy based Fuzzy Multi-Sensor Fusion (FMSF) [20]. A successful mission depends on the underwater vehicle's ability to determine its precise position. Underwater vehicle positioning is commonly based on Kalman filtering, which requires knowledge of process and measurement noise covariances. The changing conditions underwater affect the accuracy of position estimation and sometimes cause divergence. Moreover, the underwater multi-path effect and nonlinearities cause outliers that have a significant impact on positional accuracy. The conventional Kalman-based method cannot handle these non-Gaussian outliers. In response to these issues, this chapter presents an improved adaptive multi-sensor fusion method that adapts Kalman filter covariance to outliers by utilizing information-theoretic, learning-based fuzzy rules. The performance of the proposed sensor fusion technique is compared and evaluated using Monte-Carlo simulations, and substantial improvements in underwater position estimation are obtained.

This chapter starts with an introduction which is Section 4.1. Section 4.2 discusses state of the art in the field. Section 4.3 highlights novelty of the proposed research. Section 4.4 provides a brief overview of the underwater navigation system. The mathematical details are given in Chapter 2 of this thesis. These details act as the foundation of the proposed multi-sensor fusion. Moreover, this chapter highlights the major shortcomings of Kalman filtering in Section 4.5. In Section 4.6, a fuzzy correntropy-based multi-sensor fusion algorithm is proposed for the adaptation of unknown process noise covariances. Section 4.7 describes a method by using correntropy-based robust adaptation of Measurement noise covariance using versoria Kernel. Section 4.8 discusses and compares the test results of the proposed correntropy-based fuzzy multi-sensor fusion with previous methods.

4.1. Introduction

Precise seabed mapping is the ultimate requirement for extracting minerals and other natural resources from the ocean. Underwater vehicles play important roles in mapping and exploration, but their precision is highly affected by the noise conditions in the ocean environment. Moreover, the main navigation sensors of the underwater vehicle, such as gyros and accelerometers, suffer from drift and bias. As the worldwide satellite-based positioning system that uses radiofrequency cannot be accessed underwater, an alternate means of communication-based on acoustic positioning systems is usually employed, making precise location determination of vehicles considerably more difficult than it is for land vehicles. Ray bending, reflection, and the multi-path effect are all serious barriers in determining a vehicle's underwater position [103].

On the other hand, multi-sensor fusion algorithms based on Kalman filtering require complete knowledge of system model and noise characteristics, which is difficult to obtain in underwater environments. Typically, for the system process model, the deterministic component of the underwater vehicle is often derived using kinematic principles, whereas the stochastic element of the model is represented by noises, which are mostly influenced by modeling errors and nonlinearity. Likewise, the stochastic portion of the measurement model is heavily impacted by sensor characteristics and reliability. In a practical underwater vehicle navigation scenario, the system process and measurement noise covariance are unknown, and incorrect values cause compromised position estimation and divergence. The best estimation of underwater vehicle position is only possible with prior knowledge of noise covariance [46]. Thus, correct adaptation to the underwater noise environment is a key requirement for getting a precise position to mapping the seafloor.

Fuzzy logic has provided a simple solution for adaptation of noise covariance using expert knowledge [104],[105],[106]. It has the ability to define complex nonlinear equations with a simple linguistic rule base. It is possible for computer programs to use human experience, understanding, and rationale with the help of fuzzy set theory. Since the invention of fuzzy logic and its accompanying mathematics, particularly the fuzzy Kalman filter, fuzzy systems have been applied to control and estimation in a variety of controller design research and state estimation navigation filters.Most previous research studies [107],[108],[109] have been based on matching theoretical and actual covariance based on the difference, but this method does not give accurate results in the presence of heavy outliers in the stochastic part of the model.

Information-theoretic learning (ITL) has been successfully utilized to test nonlinear similarity based on correntropy, particularly for noisy outlier environments [110]. As a nonlinear similarity measure, correntropy shows the closeness of two random variables with the given kernel size. Furthermore, it can preserve nonlinear features, as well as high-order moments [111]. However, the existing correntropy-based works are lacking the advantages of using fuzzy logic. These advantages of

fuzzy logic and correntropy motivated us to propose new algorithms and drived us to answer a major research question: Can we use correntropy's strengths to improve underwater vehicle navigation performance in the presence of nonlinearity and outliers?

4.2. Review of previous work

Different types of adaptive Kalman filtering for underwater navigation applications have been developed and used since the advent of the Kalman filter [2],[112]. Initially, Mehra laid the foundation by proposing four state-of-the-art approaches intended to address a situation in which the system and measurement noise covariance matrices cannot be known during the design phase or to correct for scenarios in which both covariance matrices change over time [46]. These four techniques are Bayesian, correlation, Maximum Likelihood Estimation (MLE), and covariance matching. These techniques have been applied in various land, air, and space applications [113],[114],[115],[116],[117].

Many research investigations are being done on underwater vehicle positioning, and navigation evolves during off-shore resources exploration. A study proposed modifications of multi-model Kalman filters for underwater navigation by using the probabilistic data association theory and claimed to improve the navigation accuracy [118]. Overall, the method is computation- and memory-intensive due to the use of multiple Kalman filters and requires previous steps of states for autocorrelation calculation. Through probability calculations, they dynamically determine the most efficient navigation routes. Another piece of research conducted recently used the MLE and RTS smoother for process and measurement noise adaptation to eliminate range error [119]. According to them, using only one acoustic beacon in this solution makes it more cost-effective. However, the smoother RTS can significantly increase the computational time and memory requirement. In addition, the main problem of the MLE method is high sensitivity to outlier auxiliary data.

The improved Sage-Husa adaptive Kalman filter was claimed to enhance the underwater navigation accuracy of a tightly coupled, Strapped down Inertial Navigation system (SINS) and Doppler Velocity Log (DVL)-based system [120]. The method employed the forgetting factor for memory optimization and variable sliding window for decreasing computational time. Another study proposed a two-stage adaptive information filter that used an Ultra-Short Baseline (USBL) with DVL for estimation of an unknown sea current [121]. Their design is based on two information filters one is a standard information filter for estimation of states, and the other is based on the sequential least squares algorithm for estimating the velocity of unknown sea currents. Furthermore, they introduced the forgetting factor for the fast processing of new data. However, running two-stage filters can add computational load, which is not discussed in the study.

Recently, improvement in the Variational Bayesian approximation-based Adaptive Kalman Filter (VB-AKF) was proposed by using asynchronous auxiliary sensor measurements [122]. They claimed that this technique significantly reduces RMSE for position estimation. In recent years, there

has been growing interest in neural network-based underwater navigation. Various approaches have been proposed, and more recently, a research study used a deep recurrent neural network involving sequential learning with Long Short-Term Memory (LSTM)[123]. They claimed that their method outperformed Kalman-based solutions in terms of accuracy. However, they did not mention the training time requirement, processing load, and the total number of neurons used in the network. Another current study proposed an end-to-end navigation solution based on deep hybrid recurrent neural networks and used raw sensors data directly to estimate the location of underwater vehicles [124]. An investigation that was conducted recently took advantage of Reinforcement Learning (RL) and incorporated the deep deterministic policy gradient for tuning the process noise covariance matrix online from low-cost navigational sensors [125]. Their method used the positioning error as a reward function for training RL. However, the performance of RL neural network-based algorithms is directly proportional to the training period and previous data storage in the memory.

Several authors have also suggested the use of terrain-based underwater navigation, which aims to solve the long-distance underwater navigation problem [126],[12]. These terrain-based navigation solutions utilize bathymetric data, underwater topographical features, and underwater earth gravitational and magnetic profiles [65]. The concept behind terrain navigation is to construct a terrain profile map from sensor measurements, then compare it to a previous map database to get the best location estimate. For instance, a detailed study targeting underwater vehicles used multi-model adaptive estimation (MMAE) for terrain-based navigation. They utilized Principal Component Analysis (PCA) with MMAE for underwater terrain matching [127]. In recent research, neural networks were combined with terrain-based navigation [128]. They used the Rao-Blackwellized particle filter and offline trained neural network with terrain maps for pattern recognition from time-series data. However, terrain-based techniques are only useful when previous map databases are available, and sensors are accurate. Moreover, carrying these map databases on underwater vehicles increase memory requirements, processing power, and computational load.

The application of the fuzzy set theory enables human experience, understanding, and rationale to be used by computer programs. Fuzzy systems have been documented in numerous control theories and state estimation filters since the invention of fuzzy logic and the accompanying mathematics, notably the fuzzy Kalman Filter. Because they do not require any mathematical model of the system, adaptive fuzzy filters are particularly effective in dealing with nonlinearity and modeling inaccuracies. Researchers seminal contribution by employing fuzzy logic with a Kalman filter for adaptation of noise covariance. The research work utilized nine rules, with covariance residuals and the mean of residuals as input, for designing the exponential weighting factor [129]. A large number of existing research studies in the broader literature have used fuzzy logic for designing an addition or multiplication factors to the noise covariances [106],[130],[131]. However, there are wider choices of fuzzy inputs available in the literature depending on the application. The popular choices of input are actual covariance,

Degree of Divergence (DOD), Degree of Mismatch (DOM), the difference between the theoretical and actual covariance, and mean value of the residual[132],[133]. More recently, adaptive fuzzy Kalman was proposed for spacecraft navigation based on DOM and DOD. They claimed that fuzzy-based methods require fewer computing resources than the MLE-based method [134]. Despite this, the performance of this method has not been tested with data affected by outliers, which are common in underwater environments [33],[135].

4.3. Novelty and contributions of the proposed research

The ultimate goal of this research is to take the benefits of correntropy's strengths and to address the current need for improving underwater vehicle navigation performance in the presence of nonlinearity and non-Gaussian outliers. There is no previous underwater multi-sensor-fusion method, to the best knowledge of the authors, that augments the benefits of correntropy, fuzzy, and Kalman filtering applied to the improvement of autonomous underwater vehicle navigation. It is aimed at improving the performance by using fuzzy logic, which has the benefit of handling nonlinearity based on expert knowledge, correntropy for robust handling of a non-Gaussian outlier, and Kalman filter for real-time minimum error variance processing. We named this algorithm FC-MSF, where FC stands for Fuzzy Correntropy and MSF for Multi-Sensor data Fusion. Another noteworthy contribution of this work is the introduction of new metrics based on correntropy, which uses high-order moments for improving covariance matching adaptation. We proposed these metrics as a Degree of Similarity (DOS) and Degree of Convergence (DOC) that compare theoretical and actual covariance statistically, which was not the case in previous DOM and DOD. Moreover, the fuzzy logic and correntropy-based similarity measures together provide more robustness to the large outliers, which are commonly present in underwater acoustic position measurements and velocity measurements by DVL. Correntropy provides a similarity measure based on kernels, specifically the Gaussian and Versoria kernels, because of their distinct advantages to the heavier tail underwater vehicle stochastic data. A simulation study has demonstrated the superior performance of the proposed FC-MSF algorithm and validated that the novel correntropy-based metrics improved outlier-influenced underwater navigation in the absence of global satellite-based positioning systems. Importantly, the suggested technique is evaluated by injecting the outlier as a shot non-Gaussian noise, which is a prominent method used by many researchers to assess the robustness of the filter [136],[137],[111]. Subsequently, a comparative analysis is performed, which enables the authors to assess the accuracy of the suggested method. Monte Carlo simulations indicate that the approach is technically feasible and has the potential of yielding positive results in a harsh underwater environment.

4.4. Underwater vehicle navigation system

The complete mathematical modeling of the vehicle navigation system and sensors is discussed in Chapter 2 of this dissertation. For more details, readers may also refer to [22],[138].

The navigation electronics shown in Figure 4.1 carry the main embedded microprocessor and primary sensors used for navigation. The inertial navigation system (INS) is comprised of an inertial measurement unit (IMU) which gives data at a high rate to the processor. The IMU contains gyros and accelerometers, which are referred to as primary sensors. The low-rate secondary sensors include a depth sensor, an electronic compass, and acoustic position transceivers, which enable USBL connection with the vessel and DVL. The system-level diagram of the underwater vehicle is shown in Figure 4.1.

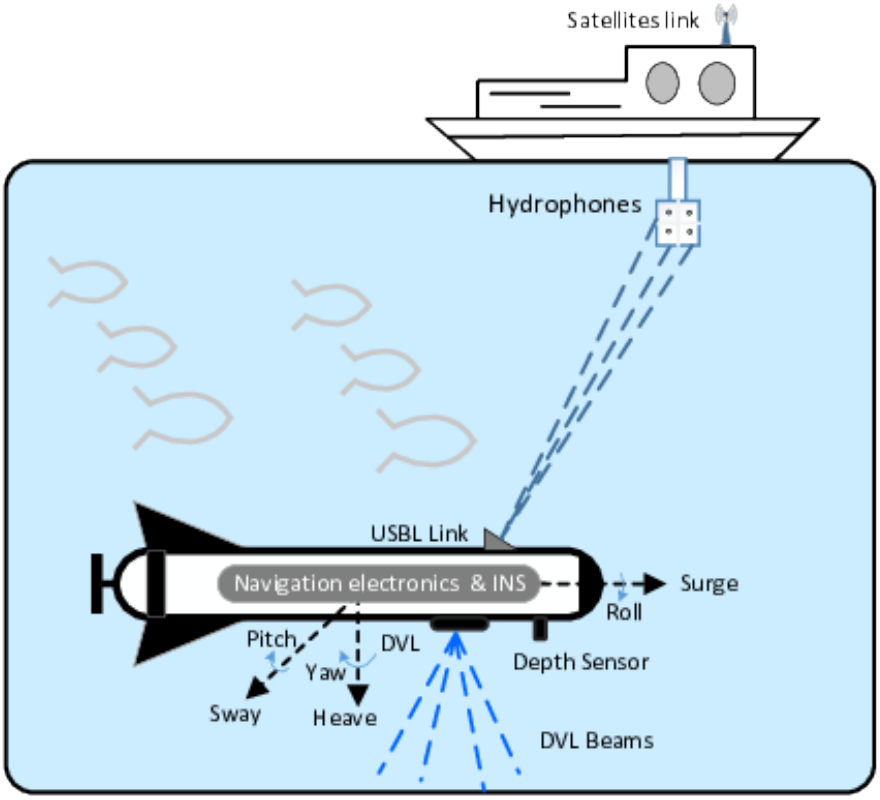

Figure 4.1 System level diagram of sensors used by vehicle while maneuvering underwater

4.5. Shortcomings of Kalman filtering

Nevertheless, when the measurements are contaminated by non-Gaussian noise, such as outliers or impulsive noise inference, EKF will perform poorly and even diverge [139]. The term $(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k^-)$ is known as innovation. It is the difference between the measurement error vector $\mathbf{z}_k$ and its predicted error vector $\mathbf{H}_k\mathbf{x}_k^-$. If the heavy outliers impact measurement or process modeling by including errors caused by nonlinearity, the innovation term will produce erroneous results, causing the filter to diverge. Specifically, in underwater conditions, a major limitation for using Kalman filtering is the limited prior knowledge of process noise and measurement noise covariances. Incorrect initialization of covariance

can cause filters to diverge. Moreover, the statistics of noise can change according to underwater conditions. Consequently, the adaptation of process and measurement covariance is necessary to get explicit navigation accuracy of the underwater vehicles.

4.6. Correntropy-based fuzzy multi-sensor fusion

The proposed modifications improve the performance of Kalman filter-based multi-sensor fusion by utilizing the strengths of correntropy and fuzzy logic. Fuzzy logic has been shown to control nonlinear processes by using human linguistic expressions, and this capability is combined with Kalman filters to solve divergence problems and improve accuracy [105],[130]. Covariance matching has been a popular methodology for adaptive fuzzy Kalman filtering in earlier research [131],[109],[140]; however, there is no reliable way for matching covariance when data have significant nonlinearity and heavy outliers, as in the case of an underwater vehicle navigation system. Therefore, we propose a correntropy-based covariance matching for fuzzy system input because of its robustness to outliers and non-Gaussian noise [140].The top-level block diagram of our proposed multi-sensor fusion algorithm is depicted in Figure 4.2 below

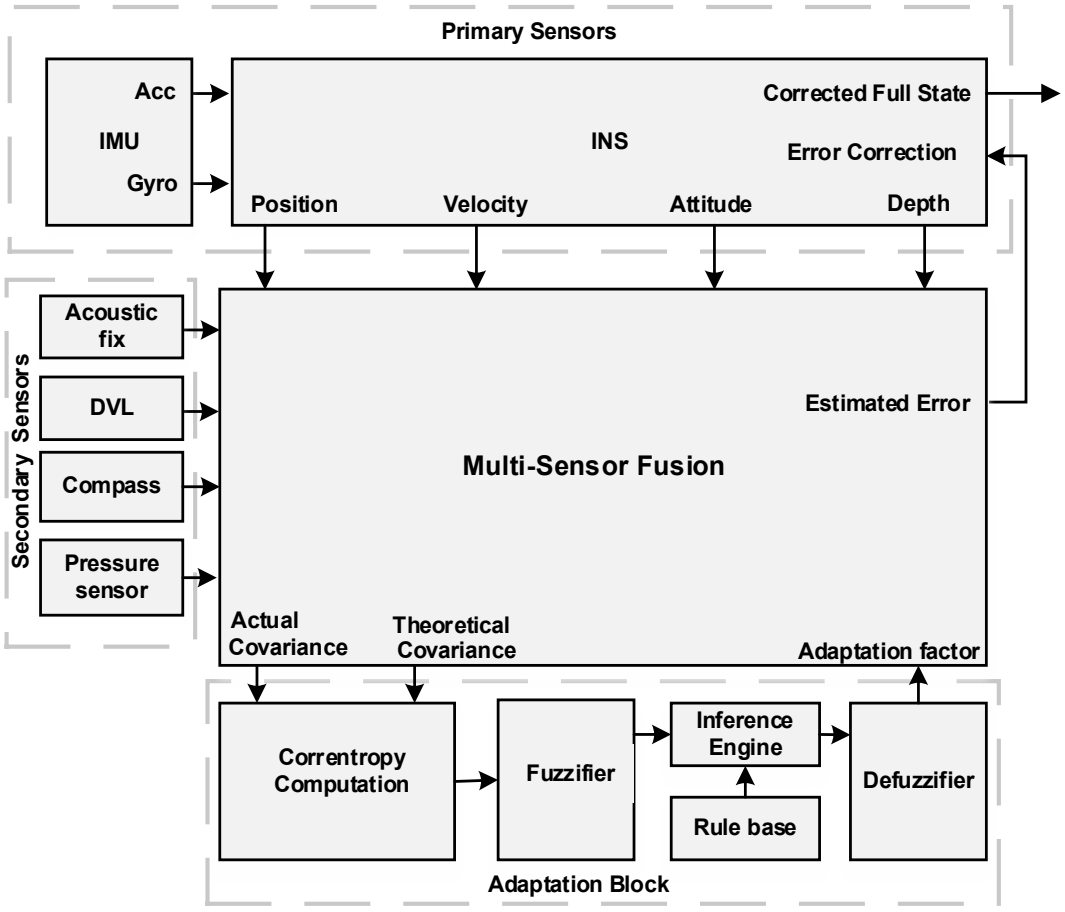

Figure 4.2 Top-level diagram of proposed integrated navigation architecture for underwater vehicle

Figure 4.2, the primary sensors data from IMU and auxiliary sensors, such as the depth sensor, DVL, the acoustic position from USBL, and compass, is fed into the Kalman-based fusion algorithm. The fusion algorithm works on the error dynamics of input data. The correntropy computation block is the part of the adaptation block that receives actual and theoretical covariance as input from the fusion block. The correntropy block provides a similarity measure from 0 to 1, where 1 means maximum similarity. The numerical value from the correntropy block is used for fuzzification. The fuzzified linguistic terms are passed through an inference engine that takes human expert-driven rules base. To obtain the adaption factor, the result of the inference engine is defuzzified. The correction is applied to covariance matrices which protect divergence and improve the accuracy of the filter.

4.6.1 Adaptation by covariance matching

The most commonly used approaches for adaptation are based on the covariance matching approach, which makes the theoretical value and actual value consistent with each other. The approach for adaptation of covariance matrices involves the simultaneous adaptation, both of the process and measurement noise covariances, or the adaptation of either of those two covariances if only one is known.

The theoretical covariance is used as a basis for comparison. It is given as

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k. \tag{4.1}$$

The actual covariance is calculated by taking a moving windows average of measurement innovation [115]. It is given as

$$\mathbf{C}_k = \frac{1}{\lambda} \sum_{i=i_0}^{\lambda} \mathbf{s}_k \mathbf{s}_k^T, \tag{4.2}$$

where λ is the size of the window.

The most commonly used criteria used for fuzzy adaptive Kalman filter are **DOM** and DOD [133][141][134]. They are mathematically given by the following equations:

$$DOD = Tr(\mathbf{S}_k) - Tr(\mathbf{C}_k) \tag{4.3}$$

$$\mathbf{DOM} = \mathbf{S}_k(j,j) - \mathbf{C}_k(j,j). \tag{4.4}$$

The DOD gives one scalar value by subtracting the trace of theoretical and actual covariance matrices. On the other hand, **DOM** is a vector as a result of the diagonal difference of theoretical and actual covariance matrices.

When theoretical and actual covariance is perfectly matched, the DOD and **DOM** are close to zero. Moreover, the positive or negative values of these metrics indicate the positive or negative direction of tuning for covariance matrices.

There are two major problems with this approach. Firstly, it ignores outliers and nonlinearity, both of which are frequent in underwater settings. Secondly, innovation's autocorrelation does not reflect actual covariance. As a result, using **DOM** or DOD, the impact of outliers is reflected in the tuning process, which negatively influences the filter response. Furthermore, in the case of impulsive non-Gaussian noise, they do not provide accurate results.

4.6.2 Correntropy-based robust adaptation of process noise covariance

Correntropy is a similarity metric between two random variables [142]. It is based on kernel methods that take into account both a statistical distribution and temporal structure [110]. It is defined as

$$M(A,B) = E\{\kappa(a_k, b_k)\}, \tag{4.5}$$

where κ is the kernel that satisfies Mercer conditions [143], and E is the expectation operator. Random variables are represented by A and B. In a practical situation, the joint distribution is not available and correntropy is calculated by using finite samples of random variables. It is calculated by using N samples of distributions as

$$\widehat{M}(A,B) = \frac{1}{N}\sum_{i=1}^{N} \kappa_\sigma(a_k, b_k). \tag{4.6}$$

The most commonly used kernel is Gaussian, which reaches the maximum value when $a_k = b_k$. Moreover, the correntropy function based on the Gaussian is positive and bounded. It is written as

$$\kappa_\sigma(a_k, b_k) = \exp\left(-\frac{\|a_k - b_k\|^2}{2\sigma^2}\right), \tag{4.7}$$

where σ is the width of the kernel. The robustness will be good, but the convergence speed will be slow if the kernel width is too small; conversely, if the kernel width is too large, the convergence speed will be rapid, but iterations may take a longer time. Selecting the appropriate kernel width is vital to the performance.

In this work, a new metric based on the correntropy Gaussian kernel is introduced to calculate a new comparison parameter named Degree of Convergence (DOC), which is the opposite of DOD and has better performance in nonlinear conditions. Furthermore, it solves two major problems by having the property to suppress the negative effects of the large outliers and providing better results in non-Gaussian conditions. The theoretical and actual covariance are matched using the DOC function, which is defined as

$$DOC = \frac{1}{N}\sum_{i=1}^{N} \kappa_\sigma(e_k), \tag{4.7}$$

where e_k is calculated as

$$e_k = Tr(\mathbf{S}_k) - Tr(\mathbf{C}_k). \tag{4.8}$$

4.6.3 Fuzzification of Degree of Convergence

Fuzzification is the process of converting crisp values into fuzzy sets based on vague linguistic variables [144]. Fuzzification of DOC is based on its properties. In particular, DOC is maximum when theoretical and actual covariance is matched, and there is no need to tune $\mathbf{Q}$. Furthermore, DOC is a symmetric positive function; therefore, fuzzy variables are defined in a symmetric manner. Fuzzification of DOC used eight input linguistic terms: Positively Full Converge (PFC), Negatively Full Converge (NFC), Positively Moderate Converge (PMC), Negatively Moderate Converge (NMC), Positively Slight Converge (PSC), Negatively Slight Converge (NSC), Positively Diverge (PD), and Negatively Diverge (ND). The output linguistic variables are no change (NC), Moderate Decreased (MD), Moderate Increased (MI), Limited Decrease (LMD), Limited Increase (LMI), Significant Decrease (SD), and Significant Increase (SI). The fuzzy adaptation parameter for process noise covariance is donated by α_k. The fuzzy rules for DOC are given as

- IF DOC is PFC THEN NC in α_k
- IF DOC is NFC THEN NC in α_k
- IF DOC is PMC THEN MD α_k
- IF DOC is NMC THEN MI α_k
- IF DOC is PSC THEN LMD α_k
- IF DOC is NSC THEN LMI α_k
- IF DOC is PD THEN SD in α_k
- IF DOC is ND THEN SI in α_k

A fuzzy set can be visually represented using membership functions. Different forms are determined by different sorts of mathematical formulas when expressing fuzzy sets with membership functions. The range [0, 1] is used to define fuzzy sets. The membership function of a fuzzy set J can be represented by

$$J = \{(l, \mu_J(l)), \quad \text{suchthat} l \in L\}. \tag{4.9}$$

The membership value of the element l in fuzzy subset J is denoted as $\mu_J(l)$. The universe L contains the crisp variable l.

The triangular function is defined by the following equations:

$$\mu_j(l) = \begin{cases} 0, & l \le a \\ \frac{l-a}{m-a}, & a < l \le m \\ \frac{b-l}{b-m}, & m < l < b \\ 0, & l \ge b \end{cases} \tag{4.10}$$

where the lower limit is defined by a, an upper limit b, and a value m, where $a < m < b$.

The trapezoidal functions are used on the extreme left and right. The right trapezoidal function is defined as

$$\mu_J(l) = \begin{cases} 0, & l > d \\ \frac{d-l}{d-c}, & c \le l \le d \\ 1, & l < c \end{cases} \tag{4.11}$$

The left trapezoidal function is defined as

$$\mu_J(l) = \begin{cases} 0, & l < a \\ \frac{l-a}{b-a}, & a \le l \le b \\ 1, & l > b \end{cases} \tag{4.12}$$

Fuzzy inference is a method for determining how probable an input is to correspond to a specific output. The work utilizes Mamdani inferencing, which assumes that the output membership functions are fuzzy sets. Fuzzy "and" operation is computed Zadeh-min, taking the minimum of the two membership values.

Defuzzification is the process of transforming a fuzzy output, which cannot be used directly in a distinct crisp value. The center of gravity (COG) method [52] is used to get a crisp value of α_k, given by the following equation

$$\alpha_k = \frac{\sum_{i=1}^{n} \Delta_i \times e_i}{\sum_{i=1}^{n} \Delta_i}, \tag{4.13}$$

where n depends on the partition of linguistic rules. Δ_i represent the area under the membership function (i), and e_i is the ith centroid.

The adaptation of process noise covariance is given by the following equation [141],[109]:

$$\mathbf{Q}_{k+1} = (g_q \alpha_k + 1)\widetilde{\mathbf{Q}}_k, \tag{4.14}$$

where $\widetilde{\mathbf{Q}}_k = diag[\mathbf{Q}_k]$ and g_q are gain scaling factors.

Algorithm 4.1 shows iterative steps for **Q** adaptation using fuzzy correntropy-based Kalman filtering by the Gaussian kernel.

Algorithm 4.1 Fuzzy Correntropy-based Kalman Filtering by Gaussian Kernel

Initialize: Initialization of KF state and covariance variables Initialization of fuzzy correntropy variables

Time update:

$$\mathbf{x}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{x}_k^{+}$$

Time propagation of covariance

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_k^{+} \mathbf{\Phi}_k^{\top} + \mathbf{Q}_k$$

Calculation of Kalman gain: Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_k^{-} \mathbf{H}_k^{\top} \left(\mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^{\top} + \mathbf{R}_k \right)^{-1}$$

Innovation calculation: The difference between the measured and predicted value

$$\mathbf{s_k} = \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^{-}$$

Measurement Update: State is corrected by using Kalman gain and innovation

$$\mathbf{x}_k^{+} = \mathbf{x}_k^{-} + \mathbf{K}_k \mathbf{s_k}$$

State covariance corrected by Kalman gain

$$\mathbf{P}_k^{+} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^{-}$$

Theoretical covariance calculation

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^{T} + \mathbf{R}_k$$

Actual approximated covariance in moving windows

$$\mathbf{C}_k = \frac{1}{\lambda} \sum_{i=i_0}^{\lambda} \mathbf{s}_k \mathbf{s}_k^{T}$$

Correntropy based Degree of Convergence (DOC) calculation by using Gaussian kernel

$$DOC = \frac{1}{N} \sum_{i=1}^{N} \kappa_\sigma(\mathbf{s}_k, \mathbf{c}_k)$$

$$\kappa_\sigma(\mathbf{s}_k, \mathbf{c}_k) = \exp\left(-\frac{\|e_k\|^2}{2\sigma^2} \right)$$

Fuzzy adaptation of Q Fuzzification using triangular and L and R type trapezoidal functions.

Application of rules using inference engine. Defuzzification to crisp output for α_k by the center of gravity (COG) method. Updated process noise covariance

$$\mathbf{Q}_{k+1} = (g_q \alpha_k + 1) \tilde{\mathbf{Q}}_k$$

Next iteration (posterior becomes prior)

4.7 Correntropy-based robust adaptation of measurement noise covariance

Multi-path is one of the most challenging problems that acoustic systems face underwater. In shallow water, the signal propagates by reflections from the surface and bottom, in addition to the direct channel, which causes the multi-path effect. On the other hand, in deep waters, a multi-path is created mainly as a result of ray bending due to refraction. Moreover, the sound speed profile, which depends on temperature, depth, and salinity, is also a critical contributor of multi-paths. Furthermore, air bubbles and marine animals have their parts for outliers. Figure 4.3 shows the multi-path in deep and shallow water.

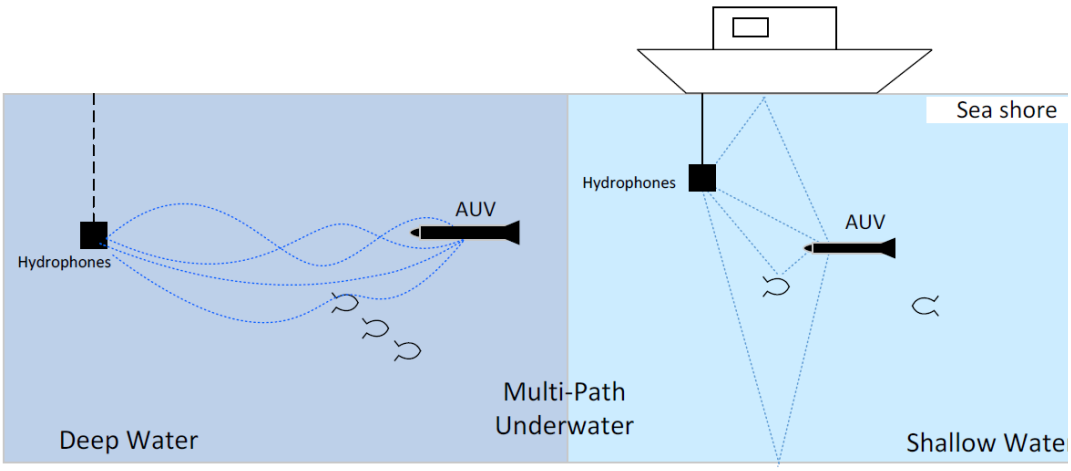

Figure 4.3 Multi-path effect in deep sea and shallow water

The heavy-tailed distribution is a feasible representation for data that has been corrupted by multi-path outliers, which is common in underwater acoustics. The direct subtraction for covariance matching does not reduce the effects of outliers. These outliers' heavier tail represents how the probability of extreme outcomes is higher in the tails than in the normal distribution. The Versoria correntropy kernel is well-suited in an underwater environment for its robustness to outliers. The tail of the Versoria function is heavier than Gaussian and Student's t distributions [145]. Moreover, the Versoria kernel error converges faster than the exponential-based kernel. In addition, it has less computation complexity as compared to the Gaussian Kernel. We define another new metric called the Degree of Similarity (**DOS**), which is used to calculate correntropy using the Versoria function, given as:

$$\mathbf{DOS} = \frac{1}{N}\sum_{i=1}^{N} \kappa_{\sigma}\left(\epsilon_k^i\right). \tag{4.15}$$

The Versoria function is given as

$$\kappa_\sigma(\epsilon_k) = \frac{A^3}{A^2+(|\epsilon_k^i|)^2}, \tag{4.16}$$

where $A = 2r$ and r are the radii of the circle located at $(0, r)$ and $\boldsymbol{\epsilon}_k$ represents the error, which is given as

$$\epsilon_k^i = \mathbf{S}_k^i(j,j) - \mathbf{C}_k^i(j,j), \tag{4.17}$$

where $\mathbf{S}_k(j,j)$,$\mathbf{C}_k(j,j)$ i to N samples are drawn from the storage of diagonal elements of the covariance matrix. The value of the DOS is positively bounded, and the direction of tuning is determined by the positive or negative sign of ϵ_k^i.

The alternate representation of the Versoria function with the shaping factor τ is given as

$$\kappa_\sigma(\epsilon_k) = \frac{2r}{1+\tau(|\epsilon_k^i|)^2}, \tag{4.18}$$

where the shaping factor $\tau = \frac{1}{(2r)^2}$ is constant, and it is dependent on the diameter of the circle.

The adaptation of measurement noise is performed by the following equation [141],[109]:

$$\mathbf{R}_{k+1}(j,j) = \zeta(j)\widetilde{\mathbf{R}}_k(j,j), \tag{4.19}$$

where $\widetilde{\mathbf{R}}_k = diag[\mathbf{R}_k]$ and $\zeta(j)$ are given as

$$\zeta(j) = g_r(j)\gamma_k(j) + 1, \tag{4.20}$$

where $g_r(j)$ is the scaling factor and $\gamma_k(j)$ is the fuzzy adaptation parameter for the j_{th} element of the measurement noise covariance matrix.

4.7.1 Fuzzification of Degree of Similarity

Six linguistic terms are defined as Positive Perfect Matched (PPM), Negative Perfect Matched (NPM), Positive Moderate Match (PMM), Negative Moderate Match (NMM), Positive Mismatch (PM), and Negative Mismatch (NMS). The positive and negative terms are defined by signs of the ϵ_k. The output linguistics terms are defined as No Change (NC), Moderately Decrease (MOD), Moderately Increase (MOI), Large Decrease (LD), and Large Increase (LI).

4.7.2 Fuzzy rules and membership functions

Fuzzy rules for **DOS** are defined by IF-THEN statements which are based on the knowledge of the system using linguistic variables. The rules draw conclusions based on one or more premises that act as an input to the system.

- IF **DOS** is PPM THEN NC in γ_k
- IF **DOS** is NPM THEN NC in γ_k
- IF **DOS** is PMM THEN MOD γ_k
- IF **DOS** is NMM THEN MOI γ_k
- IF **DOS** is PMS THEN LD in γ_k
- IF **DOS** is NMS THEN LI in γ_k

Fuzzy membership functions are defined using a combination of triangular and trapezoidal curves. The outer left and right are trapezoidal, and the inner curves are triangular. These curves have the advantage of faster processing time as compared to other types. The inference is used to assess how probable it is that an input correlates to a specific output; in this case, each of the rules employed only has one premise by utilizing a minimum fuzzy operator. Defuzzification is performed by the COG method..

The Algorithm 4.2 shows iterative steps of measurement noise covariance **R** adaptation using fuzzy correntropy-based Kalman filtering by the Versoria kernel.

Algorithm 4.2 Fuzzy correntropy-based Kalman filtering by the Versoria kernel.

Initialize: Initialization of KF state and covariance variables Initialization of fuzzy correntropy variables **Time update:** Time update state

$$\mathbf{x}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{x}_k^{+}$$

Time propagation of covariance

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_k^{+} \mathbf{\Phi}_k^{\top} + \mathbf{Q}_k$$

Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_k^{-} \mathbf{H}_k^{\top} \left(\mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^{\top} + \mathbf{R}_k \right)^{-1}$$

Innovation calculation: The difference between the measured and predicted value

$$\mathbf{s_k} = \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^{-}$$

Measurement update: State is corrected by using Kalman gain and innovation

$$\mathbf{x}_k^{+} = \mathbf{x}_k^{-} + \mathbf{K}_k \mathbf{s_k}$$

State covariance corrected by Kalman gain

$$\mathbf{P}_k^{+} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^{-}$$

Theoretical covariance calculation

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^{T} + \mathbf{R}_k$$

Actual approximated covariance in moving windows

$$\mathbf{C}_k = \frac{1}{\lambda} \sum_{i=i_0}^{\lambda} \mathbf{s}_k \mathbf{s}_k^{T}$$

Correntropy based Degree of Similarity (DOS) calculation by using Versoria kernel

$$\mathbf{DOS} = \frac{1}{N} \sum_{i=1}^{N} \kappa_\sigma \left(\epsilon_k^i \right)$$

$$\kappa_\sigma(\epsilon_k) = \frac{2r}{1 + \tau(|\epsilon_k^i|)^2}$$

Fuzzy adaptation of R Fuzzification using triangular and L and R type trapezoidal functions

Application of rules using inference engine. Defuzzification to crisp output for γ_k is done by COG method. Update measurement noise covariance

$$\mathbf{R}_{k+1}(j,j) = \zeta(j) \widetilde{\mathbf{R}}_k(j,j)$$

Next iteration (posterior becomes prior)

Figure 4.4 shows a flow chart representation of Algorithm 4.1 and Algorithm 4.2 working together for process noise covariance and measurement noise covariance adaptation.

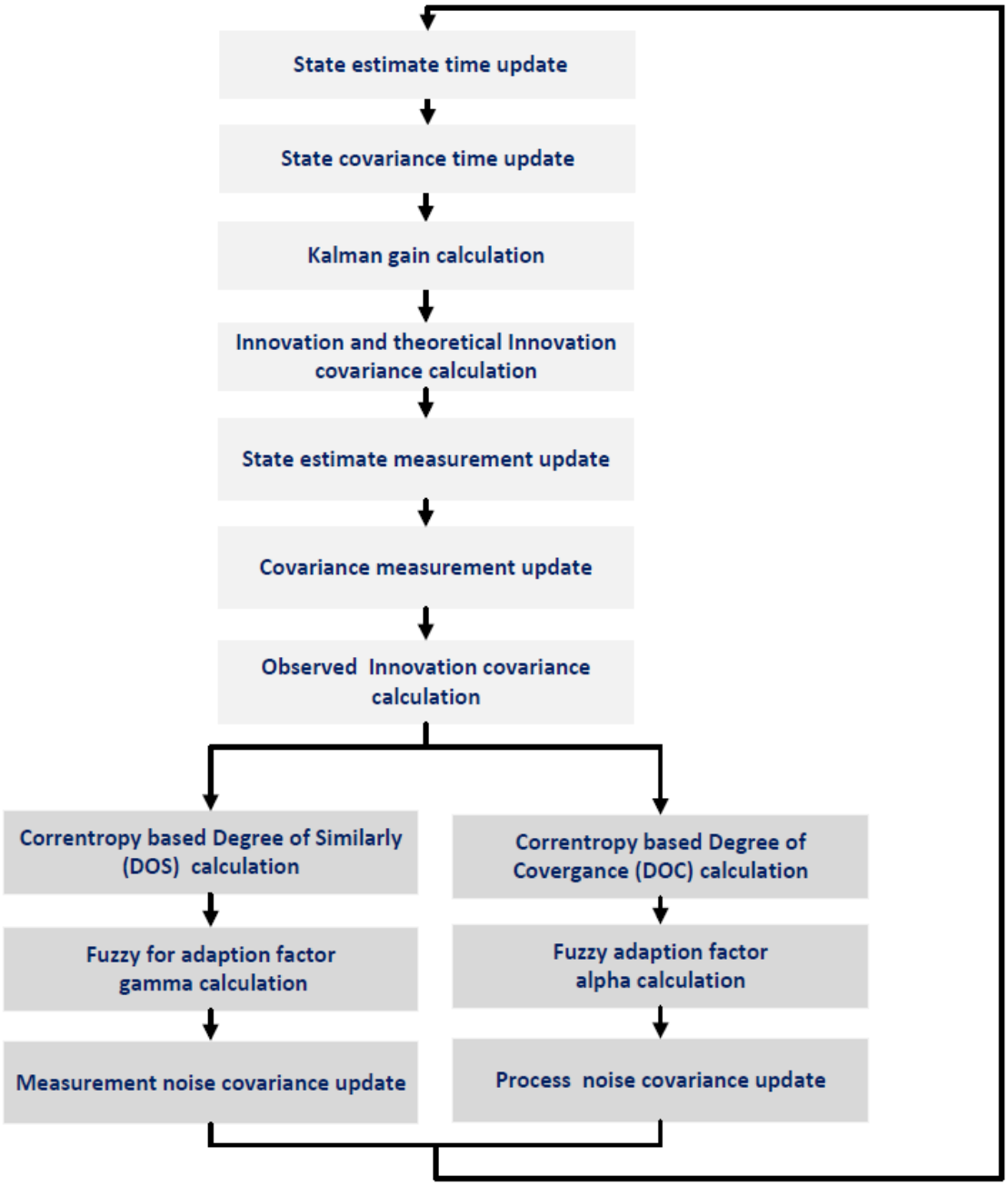

Figure 4.4 Flow chart representation of the proposed FC-MSF method

4.8 Simulation results and discussion

This section is divided into two subsections. The first, Section briefly explains the simulation scenario, and second, the Section discusses the simulation results.

4.8.1 Simulation scenario

The performance of the proposed fuzzy correntropy-based multi-sensor algorithm is compared with Kalman-based multi-sensor fusion and fuzzy multi-sensor fusion using Monte Carlo simulation. The Root Mean Square Error (RMSE) was chosen as the main metric for comparison because it compares overall filter estimation performance over a longer length of time. Calculating the RMSE for the ith state of the estimated state vector x_{est} and reference state x_{ref} is given by

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ref}^{i} - x_{est}^{i}\right)^{2}} \tag{4.21}$$

In order to assess the combined effects of the north, east, and down position and velocity, where the average *RMSE* is computed as

$$\text{AverageRMSE} = \frac{1}{3}\left(RMSE^{north} + RMSE^{east} + RMSE^{down}\right). \tag{4.22}$$

The trajectory of the underwater vehicle is simulated by using different values of acceleration and angular velocities. All filters are tested on similar conditions with the same input data for valid comparison. The vehicle is assumed to be in the normal mode of operation, with no onboard or off-board sensor failures. The initial velocities of all three axes are zero. For position Initialization, the Latitude is initialized at 50°, Longitude is initialized at 5°, and height is initialized at zero. The attitude is represented in quaternion, and it is initialized at [1 0 0 0] in the ECEF frame. The INS simulation model employs raw IMU measurement data and a starting position, velocity, and attitude to build an underwater vehicle navigation profile. All the units used in this study are based on SI units. The position is represented in meters and velocity in meters per second. The input and output triangular and trapezoidal membership functions used in this work are displayed in Figure 4.5 to Figure 4.6.

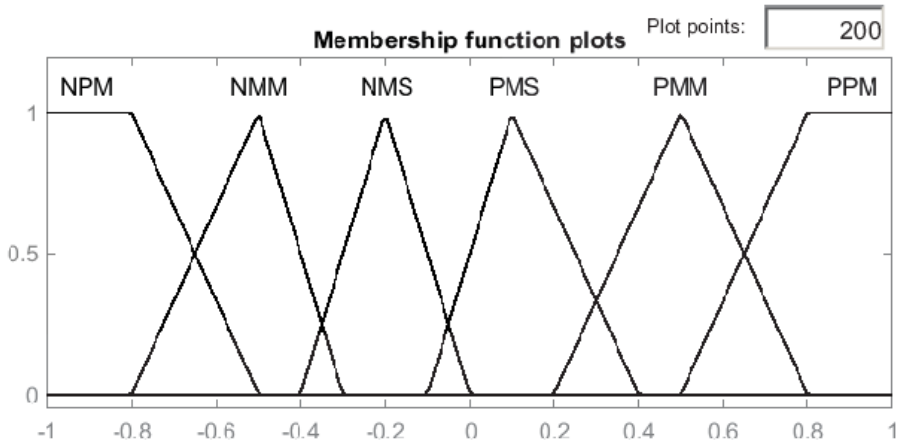

*Figure 4.5 Input membership functions for **DOS***

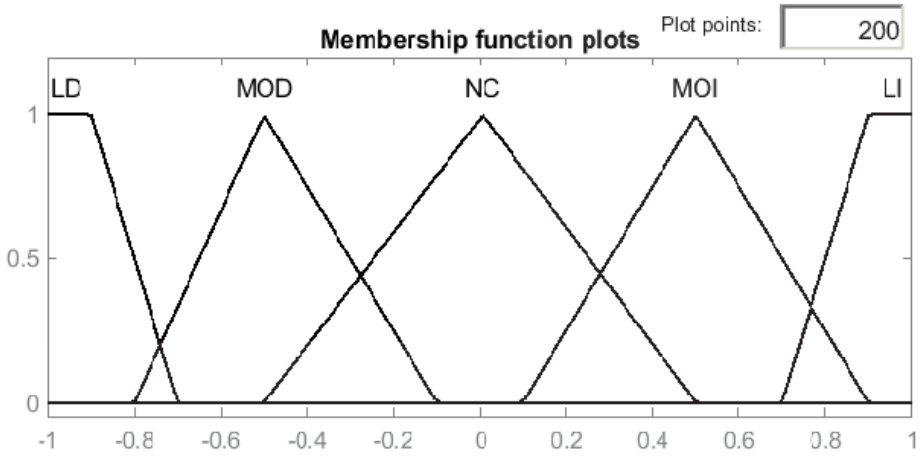

Figure 4.6 Output membership functions for DOS

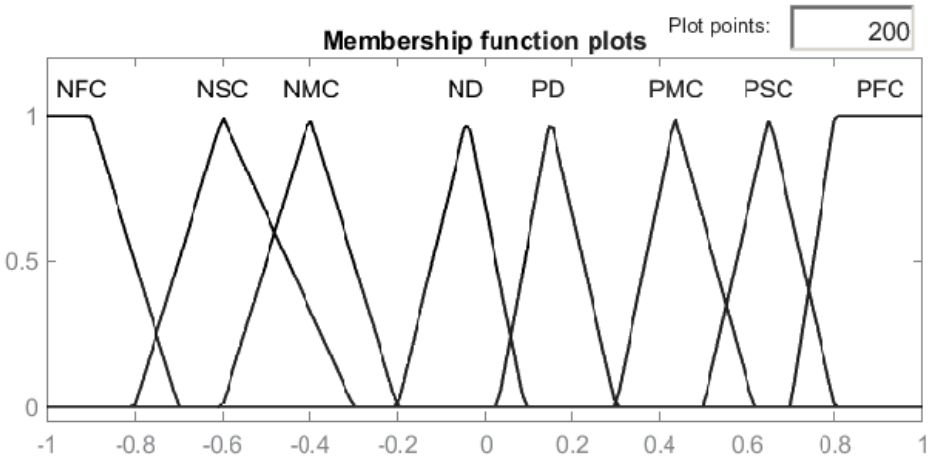

Figure 4.7 Input membership functions for DOC

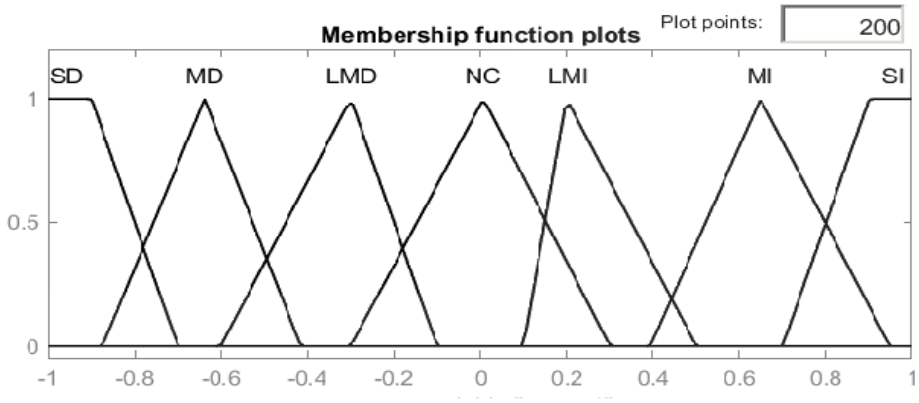

Figure 4.8 Output membership functions for DOC.

The design parameters for fuzzy logic are selected empirically, where the input gain for fuzzy scaling for process covariance adaptation is selected as 1, and the output gain is selected as 0.001. For process noise adaptation, fuzzy input scaling is selected as 1, and the output is selected as 0.01. Another critical design parameter is the width of the kernel. In general, a wider kernel width provides a quicker convergence speed, but it generally results in worse shot noise performance. The kernel width of the Gaussian correntropy function is empirically selected as 0.01, and the Versoria shaping factor is chosen as 0.5.

The IMU was simulated at a data rate of 100 Hz, and the other secondary sensors at 10 Hz. For simulation of gyros bias and noise power, spectral density is assumed to be $1°/\text{h}$ and $0.1°/\text{s}/\sqrt{\text{Hz}}$ respectively. The accelerometers bias is considered to be 250 g, and noise power spectral density is 30 $\sqrt{\text{Hz}}$. The DVL standard deviation is assumed to be $\pm 0.005\ \text{m/s}$ and random noise $0.1\ \text{m/s}$, whereas the electronic compass was supposed to have a bias 5° and random noise 1°. Lastly, the depth sensor had 0.2 m random noise.

The outliers were modeled as shot noise that is simulated by the amplitude of the noise, which abruptly increases or decreases [136].

$$w_k = (0, \mathbf{Q}_k) + \text{Shotnoise} \tag{4.23}$$

$$v_k = (0, \mathbf{R}_k) + \text{Shotnoise} \tag{4.24}$$

Figure 4.9 shows shot noise added to the system for simulations

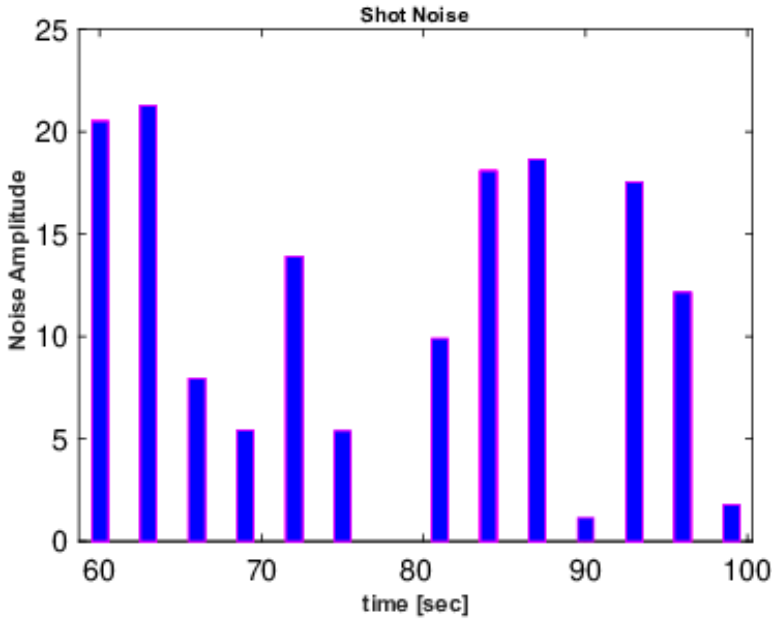

Figure 4.9 Shot noise added to the system

4.8.2 Simulation results

The performances of KF-MSF, Fuzzy (F-MSF), and correntropy-based fuzzy (FC-MSF) are compared after running 200 Monte Carlo simulation.

Table 4.1 Comparison of RMSE for position and velocity in the presence of shot noise with both R and Q adaptation.

RMSE	KF-MSF	F-MSF	FC-MSF
North Position	26.887	2.145	0.345
East Position	39.562	2.469	0.412
Down Position	9.513	0.353	0.051
Avg Position	25.321	1.655	0.269
North Velocity	1.608	0.388	0.146
East Velocity	1.529	0.485	0.121
Down Velocity	0.159	0.148	0.067
Avg Velocity	1.331	0.308	0.125

The RMSE of the position displayed in Table 4.1 Comparison of RMSE for position and velocity in the presence of shot noise with both R and Q adaptation., it can be observed that estimation of KF-MSF has a very large error due to its inability to cover the correct position with the application of shot noise. However, F-MSF and FC-MSF have much lesser position errors as compared to KF-MSF. In comparison, the RMSE of the FC-MSF for the north and east positions is significantly better than F-MSF because it does not provide robust similarity measures as with correntropy-based similarity metrics. As compared to the north and east positions, the depth error is less since it does not suffer from random bias. Nevertheless, estimation of depth from FC-MSF is far better than KF-MSF and F-MSF.

The estimation of velocity from FC-MSF showed considerable improvement compared to the KF-MSF and FC-MSF. The overall northern velocity error was almost twice improved, like that of the F-MSF. Noticeable improvements were seen in east and downward velocities. The RMSE of the FC-MSF was found to be roughly twofold better than the ESKF for the north, east, and down velocities. Overall, with our proposed FC-MSF algorithm, the average of all RMSE velocities was almost two times better than F-MSF. In comparison to F-MSF and FC-MSF, the KF-MSF has inferior estimation performance due to the lack of a robust adaptation mechanism.

These results support the hypothesis that the correntropy-based fuzzy multi-sensor fusion is less susceptible to disturbances than the F-MSF and KF-MSF. Furthermore, the robustness property of the correntropy kernel allows the FC-MSF to perform better when shot noise is present. Moreover, FC-MSF estimation results are much better than position and velocity since the correntropy has the potential to capture high-order information. Conversely, conventional fuzzy Logic correction and KF-MSF without correntropy have a negative impact on the performance, given that these solutions are exposed to the same noise.

Figure 4.10 to Figure 4.12 show a comparison of KF-MSF, F-MSF, and FC-MSF where we estimated north, east, and down velocities. It can be observed FC-MSF velocities errors remain close to the actual, while KF-MSF significantly diverges from the actual value.

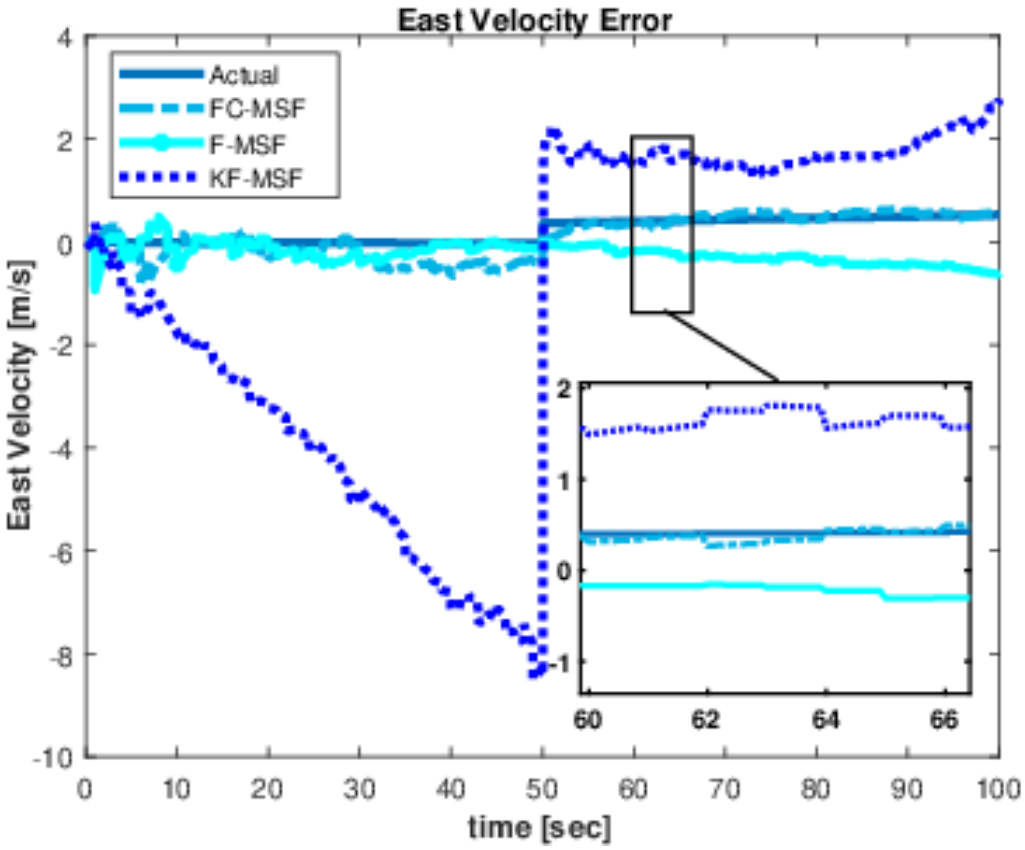

Figure 4.10 Comparison of east velocity errors

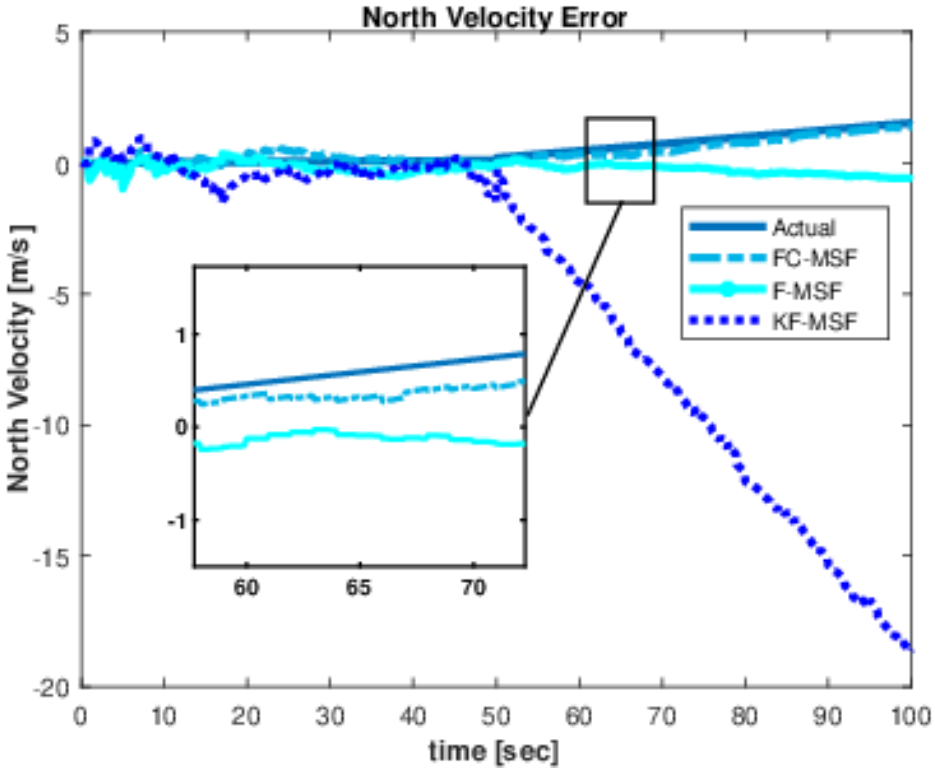

Figure 4.11 Comparison of north velocity errors

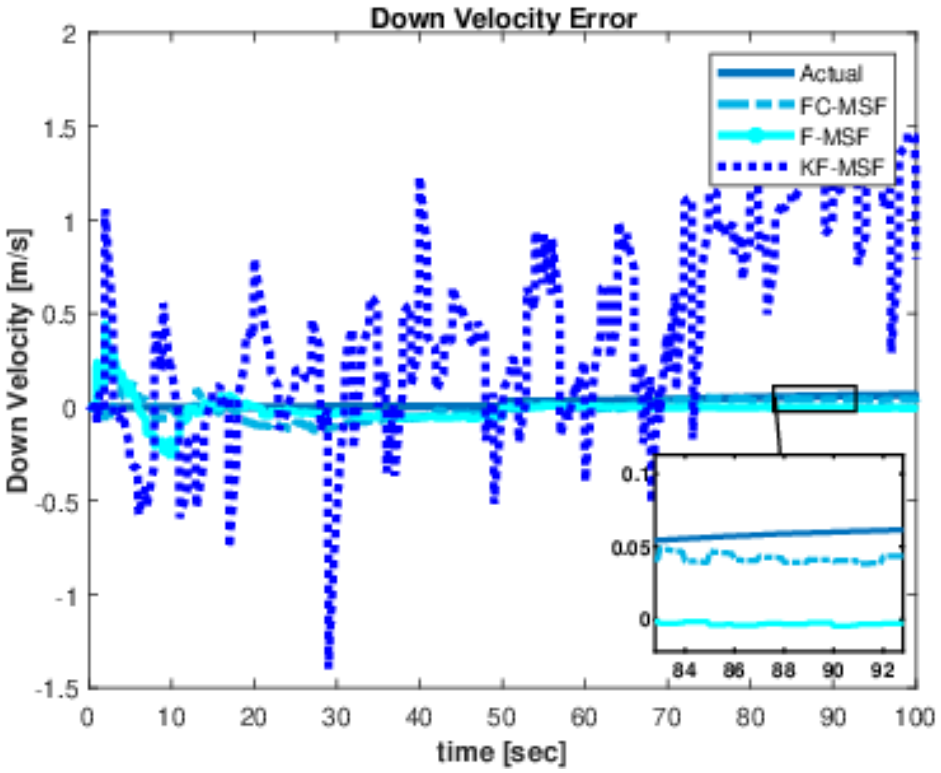

Figure 4.12 Comparison of down velocity errors

Figure 4.14 and Figure 4.13 show position errors of the north and east, and it is evident that shot noise negatively influences KF-MSF and F-MSF position estimations

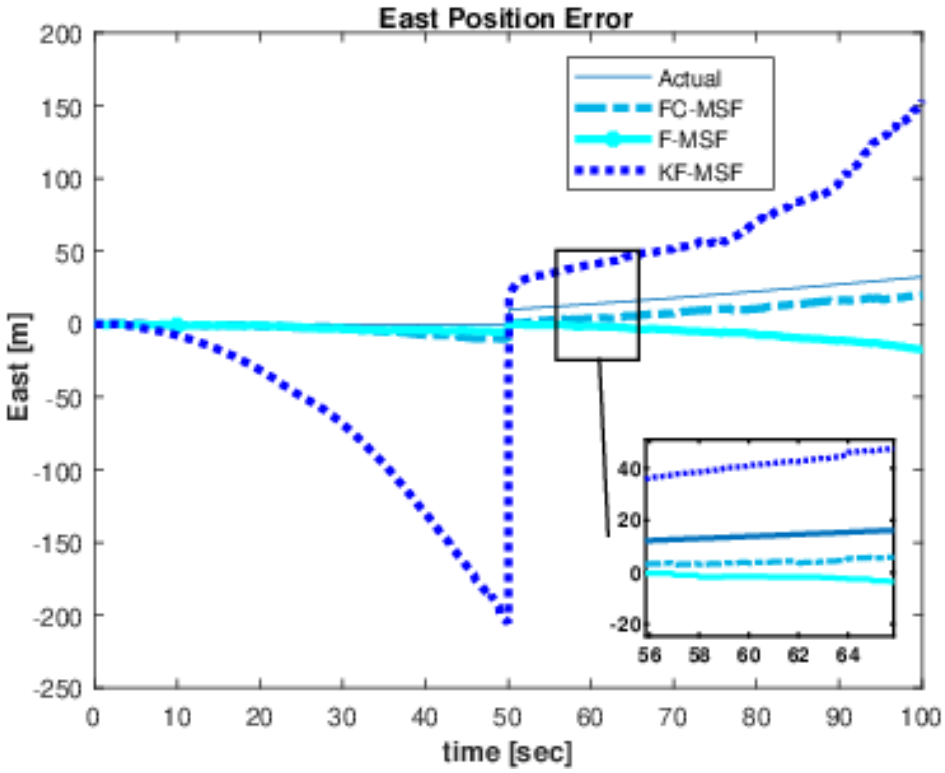

Figure 4.13 Comparison of east position errors

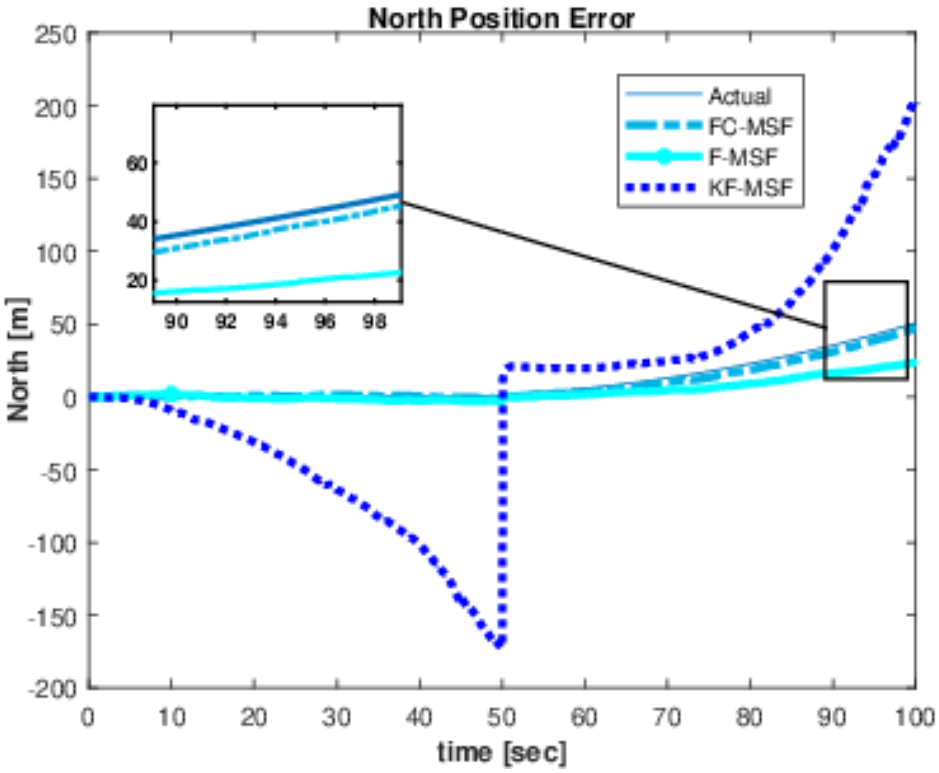

Figure 4.14 Comparison of north position errors

The above-mentioned graphs clearly illustrate that FC-MSF has superior performance, and errors are far less than KF-MSF and F-MSF. However, the suggested method is not restricted to underwater positioning applications; it may also be utilized to improve aerial positioning and navigation. Furthermore, autonomous cars can be another potential use for this approach. Additionally, satellite attitude estimates can be improved with the proposed technique.

4.9 Conclusion

This research study attempted to bridge a gap by providing a novel adaptive fusion method for underwater vehicle positioning by taking advantage of fuzzy and correntropy. The performance of the proposed algorithm is compared with Kalman and fuzzy-based sensor fusion techniques. It was found to have a better position and velocity estimation under the negative influence of shot noise. The primary aim of this work was to take advantage of correntropy and improve the covariance matching technique by using two new metrics, degree of similarity and degree of convergence. The purpose was to improve the estimation performance of conventional methods for underwater vehicle positioning. The two proposed metrics help to enhance estimation accuracy through better matching of theoretical and actual covariance. The suggested technique is designed for use in underwater seabed mapping applications for ocean exploration.

Chapter 5 : Conclusions and Future Work

The purpose of this chapter is to conclude the work presented in this dissertation and indicate some interesting directions for future work.

5.1 Conclusions

The purpose of this thesis is to study a multi-sensor data fusion approach that combines neural networks, fuzzy logic, information-theoretic learning, and Kalman filtering techniques to improve the positioning accuracy of underwater vehicles. Most underwater applications are nonlinear and defy the assumptions made in the development of the Kalman Filter. Our proposed methods mark an important step forward in the development of hybrid navigation algorithms for underwater vehicles. Moreover, the hybrid methods proposed in the thesis performed better than traditional Kalman filtering.

This study first presented a hybrid EKF based fusion method, which uses RBFNN as the basic part of the conceptual framework, to overcome the limitations of traditional methods that rely solely on Kalman filter variants for underwater multi-sensor fusion. This main aims is to enhance traditional Kalman filter estimate performance for underwater vehicle location, velocity, and attitude by utilizing neural networks. Compare to traditional multi-sensor fusion methods, a proposed hybrid neural network-based method learns underwater vehicle error state information over time, enabling it to enhance navigation performance. Based on online learning, the proposed method can make real-time corrections and improve predictions. It can automatically learn underwater conditions which reduce the error in innovation terms in the Kalman filter. Using the proposed hybrid fusion integrated system, the test results used in this study demonstrated that this system is capable of providing higher positioning accuracy during protracted USBL and DVL outages. Its overall performance revealed its benefits and possibilities for application in the creation of next-generation hybrid underwater vehicle navigation software as an alternative traditional fusion architecture.

This second part of this study focuses on developing an adaptive Kalman filter by taking the benefits of fuzzy logic and correntropy. The priori knowledge of process and measurements noise covariance (**Q** and **R**) is required for the estimation to be accurate, but when operating in underwater conditions, this is difficult to obatin as underwater conditions are changing. Therefore, designing an efficient adaptive multi-sensor fusion algorithm has become an essential requirement of underwater applications. The proposed FC-MSF method aimed to improve the performance by incorporating fuzzy logic, which can handle nonlinearity with expert knowledge, correntropy for robust handling of outliers, and Kalman filter real-time estimation. The work also introduces new metrics based on correntropy, which use high-order moments to improve the adaptation of covariance matching. Correntropy provides a similarity measure based on kernels, specifically the Gaussian and Versoria kernels, because of their distinct advantages to the heavier tail underwater vehicle stochastic data. A simulation study has demonstrated the superior performance of the proposed FC-MSF algorithm and validated that the novel correntropy-based metrics improved outlier-influenced underwater navigation in the absence of global satellite-based positioning systems.

Although the methods presented in this thesis was designed for underwater vehicles in the future. It can easily be adapted to autonomous vehicles, unmanned aerial vehicles, and subsurface mining applications. These novel approaches to underwater navigation algorithms laid the groundwork for the subsequent development of an artificially intelligent underwater fusion architecture that can overcome some of the limitations of conventional methods.

5.2 Future work

This section will provide a few specific recommendations for future work in a concise manner.

Future work could include integrating the navigation methods into an underwater guidance, navigation, and control system and then testing it in real time. Moreover, in the future, the study may be further enhanced to automatically calculate the number of hidden neurons and kernel width during the learning process, depending on the complexity of the problem. This approach will enhance overall performance of the system. Furthermore, there may be future opportunities to optimize the membership function parameters based on a set of performance criteria to improve multi sensor fusion based on the applications.

Bibliography

[1] H.-T. Choi and J. Yuh, "Underwater Robots," in *Springer Handbook of Robotics*, Cham: Springer International Publishing, 2016, pp. 595–622.

[2] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV Navigation and Localization: A Review," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 131–149, Jan. 2014, doi: 10.1109/JOE.2013.2278891.

[3] Paul D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Artech House, 2013.

[4] J. H. Kepper, B. C. Claus, and J. C. Kinsey, "A Navigation Solution Using a MEMS IMU, Model-Based Dead-Reckoning, and One-Way-Travel-Time Acoustic Range Measurements for Autonomous Underwater Vehicles," *IEEE J. Ocean. Eng.*, vol. 44, no. 3, pp. 664–682, Jul. 2019, doi: 10.1109/JOE.2018.2832878.

[5] J. Hwang, N. Bose, and S. Fan, "AUV adaptive sampling methods: A review," *Appl. Sci.*, vol. 9, no. 15, pp. 1–30, 2019, doi: 10.3390/app9153145.

[6] Ø. Hegrenæs, A. Ramstad, T. Pedersen, and D. Velasco, "Validation of a new generation DVL for underwater vehicle navigation," *Auton. Underw. Veh. 2016, AUV 2016*, pp. 342–348, 2016, doi: 10.1109/AUV.2016.7778694.

[7] Ø. Hegrenæs and E. Berglund, "Doppler water-track aided inertial navigation for autonomous underwater vehicle," *Ocean. '09 IEEE Bremen Balanc. Technol. with Futur. Needs*, 2009, doi: 10.1109/OCEANSE.2009.5278307.

[8] Q. Wang, X. Cui, Y. Li, and F. Ye, "Performance enhancement of a USV INS/CNS/DVL integration navigation system based on an adaptive information sharing factor federated filter," *Sensors (Switzerland)*, vol. 17, no. 2, 2017, doi: 10.3390/s17020239.

[9] T. Zhang, L. Chen, and Y. Li, "AUV underwater positioning algorithm based on interactive assistance of SINS and LBL," *Sensors (Switzerland)*, vol. 16, no. 1, 2016, doi: 10.3390/s16010042.

[10] K. Vickery, "Acoustic positioning systems - a practical overview of current systems," *Proc. IEEE Symp. Auton. Underw. Veh. Technol.*, pp. 5–17, 1998, doi: 10.1109/auv.1998.744434.

[11] A. Alcocer, P. Oliveira, and A. Pascoal, "Study and implementation of an EKF GIB-based underwater positioning system," *Control Eng. Pract.*, vol. 15, no. 6, pp. 689–701, 2007, doi: 10.1016/j.conengprac.2006.04.001.

[12] J. Melo and A. Matos, "On the use of particle filters for terrain based navigation of sensor-limited AUVs," in *2013 MTS/IEEE OCEANS - Bergen*, Jun. 2013, pp. 1–8, doi: 10.1109/OCEANS-Bergen.2013.6607997.

[13] *Kjetil Bergh Ånonsen Advances in Terrain Aided Navigation for Underwater Vehicles Kjetil Bergh Ånonsen Advances in Terrain Aided Navigation for Underwater Vehicles Thesis for the degree of philosophiae doctor*. 2010.

[14] L. Silveira *et al.*, "An Open-source Bio-inspired Solution to Underwater SLAM★," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 212–217, 2015, doi: 10.1016/j.ifacol.2015.06.035.

[15] X. E. Gros, "Data Fusion – A Review," in *NDT Data Fusion*, Elsevier, 1997, pp. 5–42.

[16] H. B. Mitchell, *Data Fusion: Concepts and Ideas*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[17] D. Fusion, *Multi-Sensor Data Fusion With M - Jitendra R. Raol.pdf*. .

[18] N. Shaukat, A. Ali, M. J. Iqbal, M. Moinuddin, and P. Otero, "Multi-sensor fusion for underwater vehicle localization by augmentation of rbf neural network and error-state kalman

filter," *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–26, 2021, doi: 10.3390/s21041149.

[19] N. Shaukat, A. Ali, M. Moinuddin, and P. Otero, "Underwater Vehicle Localization by Hybridization of Indirect Kalman Filter and Neural Network," in *2021 7th International Conference on Mechatronics and Robotics Engineering, ICMRE 2021*, 2021, pp. 111–115, doi: 10.1109/ICMRE51691.2021.9384844.

[20] N. Shaukat, M. Moinuddin, and P. Otero, "Underwater Vehicle Positioning by Correntropy-Based Fuzzy Multi-Sensor Fusion," *Sensors*, vol. 21, no. 18, p. 6165, Sep. 2021, doi: 10.3390/s21186165.

[21] N. Shaukat, M. Moinuddin, and P. Otero, "Underwater vehicle positioning by correntropy-based fuzzy multi-sensor fusion," *Sensors*, vol. 21, no. 18, 2021, doi: 10.3390/s21186165.

[22] J. A. Farrell, *AIDED NAVIGATION GPS with High Rate Sensors*, 1st ed. The McGraw-Hill Companies, 2008.

[23] P. D. Groves, *Principles of GNSS , Inertial , and Multisensor Integrated*. 2008.

[24] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Circumventing Dynamic Modeling : Evaluation of the Error-State Kalman Filter applied to Mobile Robot Localization," 1999, doi: 10.1109/ROBOT.1999.772597.

[25] R. M. Rogers, *Applied Mathematics in Integrated Navigation Systems, Third Edition*. Reston ,VA: American Institute of Aeronautics and Astronautics, 2007.

[26] P. A. Miller, J. A. Farrell, Y. Zhao, and V. Djapic, "Autonomous underwater vehicle navigation," *IEEE J. Ocean. Eng.*, vol. 35, no. 3, pp. 663–678, 2010, doi: 10.1109/JOE.2010.2052691.

[27] H. T. H. T. Foss and E. T. Meland, "Sensor Integration for Nonlinear Navigation System in Underwater Vehicles," Norwegian University of Science and Technology, 2007.

[28] M. Emami and M. R. Taban, "A Low Complexity Integrated Navigation System for Underwater Vehicles," *J. Navig.*, vol. 71, no. 5, pp. 1161–1177, Sep. 2018, doi: 10.1017/S0373463318000140.

[29] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. The Institution of Engineering and Technology, Michael Faraday House, Six Hills Way, Stevenage SG1 2AY, UK: Institution of Engineering and Technology, 2004.

[30] "Emcore IMU." https://emcore.com/products/sdi500-tactical-grade-imu-inertial-measurement-unit/ (accessed Jan. 11, 2021).

[31] "LN200 IMU." https://www.northropgrumman.com/what-we-do/air/ln-200-fog-family-advanced-airborne-imu-ahrs/ (accessed Nov. 01, 2021).

[32] J. L. W. David H. Titterton, J. L. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. American Institute of Aeronautics and Astronautics, 2002.

[33] "Hipap502." https://www.kongsberg.com/maritime/products/Acoustics-Positioning-and-Communication/acoustic-positioning-systems/hipap-models/hipap-502-high/ (accessed Nov. 03, 2021).

[34] O. Hegrenaes, A. Ramstad, T. Pedersen, and D. Velasco, "Validation of a new generation DVL for underwater vehicle navigation," in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, Nov. 2016, pp. 342–348, doi: 10.1109/AUV.2016.7778694.

[35] "DVL-500." https://www.nortekgroup.com/products/dvl500-300-m (accessed Nov. 03, 2021).

[36] "WHN-300." http://www.teledynemarine.com/Documents/Brand Support/RD INSTRUMENTS/Technical Resources/Manuals and Guides/WorkHorse Navigator/Navigator Integration Guide_Jun18.pdf (accessed Nov. 03, 2021).

[37] "Depth Sensor." https://www.comm-tec.com/Prods/mfgs/Paros/Series8000.pdf (accessed Nov.

03, 2021).

[38] "e-compass." https://jewellinstruments.com/products/inertial-tilt-sensors/ecompass-imu/ecs-ecompass-series/ (accessed Nov. 02, 2021).

[39] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960, doi: 10.1115/1.3662552.

[40] M. Liggins, D. Hall, and J. Llinas, *Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition (Electrical Engineering & Applied Signal Processing Series)*. 2008.

[41] M. Farrell, J and Barth, *The Global Positioning System & Inertial Navigation*, 1st ed. New York: McGrawHill Professional 1999, 1999.

[42] N. J. Salkind, "Statistics for people who (think they) hate statistics: Using Microsoft Excel 2016.," p. 544, 2017.

[43] A. Barrau and S. Bonnabel, "Invariant Kalman Filtering," *Annu. Rev. Control. Robot. Auton. Syst.*, vol. 1, no. 1, pp. 237–257, 2018, doi: 10.1146/annurev-control-060117-105010.

[44] J. M. Hansen, T. A. Johansen, N. Sokolova, and T. I. Fossen, "Nonlinear Observer for Tightly Coupled Integrated Inertial Navigation Aided by RTK-GNSS Measurements," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1084–1099, 2019, doi: 10.1109/TCST.2017.2785840.

[45] Y. Bar-shalom, X. R. Li, and T. Kirubarajan, "Booktext@Id89013304Placeboie," vol. 9, 2001.

[46] R. K. Mehra, "Approaches to Adaptive Filtering," *IEEE Trans. Automat. Contr.*, vol. 17, no. 5, pp. 693–698, 1972, doi: 10.1109/TAC.1972.1100100.

[47] D. L. Alspach, "Comments on 'On the Identification of Variances and Adaptive Kalman Filtering,'" *IEEE Trans. Automat. Contr.*, vol. 17, no. 6, pp. 843–845, 1972, doi: 10.1109/TAC.1972.1100153.

[48] J. M. Mendel, "General type-2 fuzzy logic systems made simple: A tutorial," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 5, pp. 1162–1182, 2014, doi: 10.1109/TFUZZ.2013.2286414.

[49] L. A. Páramo-Carranza *et al.*, "Discrete-time Kalman filter for Takagi–Sugeno fuzzy models," *Evol. Syst.*, vol. 8, no. 3, pp. 211–219, 2017, doi: 10.1007/s12530-017-9181-0.

[50] J. Z. Sasiadek and Q. Wang, "Fuzzy adaptive kalman filtering for INS/GPS data fusion," *1999 Guid. Navig. Control Conf. Exhib.*, no. Isic, pp. 1911–1918, 1999, doi: 10.2514/6.1999-4307.

[51] J. Jantzen, *Foundations of Fuzzy Control: A Practical Approach: Second Edition*, 2nd ed. 2013.

[52] D. Fusion, *Sensor and Data Fusion*. I-Tech Education and Publishing, 2009.

[53] O. Martin T. Hagan, Howard B. Demuth, Mark H. Beale, *Neural Network Design*, 2nd editio. 2014.

[54] Ding Xiao, Xu Li, Xiuqin Lin, and Chuan Shi, "A time Series Prediction method based on self-adaptive RBF neural network," in *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, Dec. 2015, no. Iccsnt, pp. 685–688, doi: 10.1109/ICCSNT.2015.7490837.

[55] L. Chang, B. Hu, G. Chang, and A. Li, "Robust derivative-free Kalman filter based on Huber's M-estimation methodology," *J. Process Control*, vol. 23, no. 10, pp. 1555–1561, 2013, doi: 10.1016/j.jprocont.2013.05.004.

[56] H. Wang, W. Zhang, J. Zuo, and H. Wang, "Outlier-robust Kalman filters with mixture correntropy," *J. Franklin Inst.*, vol. 357, no. 8, pp. 5058–5072, 2020, doi: 10.1016/j.jfranklin.2020.03.042.

[57] J. Opara, "Information Theoretic State Estimation in Power Systems," 2013.

[58] C. Xu *et al.*, "A novel self-adapting filter based navigation algorithm for autonomous underwater vehicles," *Ocean Eng.*, vol. 187, no. 2017, p. 106146, Sep. 2019, doi: 10.1016/j.oceaneng.2019.106146.

[59] B. Allotta *et al.*, "Development of a navigation algorithm for autonomous underwater vehicles," *IFAC-PapersOnLine*, vol. 28, no. 2, pp. 64–69, 2015, doi: 10.1016/j.ifacol.2015.06.011.

[60] H. Chen, Y. Gong, X. Hong, and S. Chen, "A Fast Adaptive Tunable RBF Network For Nonstationary Systems," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2683–2692, Dec. 2016, doi: 10.1109/TCYB.2015.2484378.

[61] K. Tomczyk, M. Piekarczyk, and G. Sokal, "Radial basis functions intended to determine the upper bound of absolute dynamic error at the output of voltage-mode accelerometers," *Sensors (Switzerland)*, vol. 19, no. 19, pp. 12–14, 2019, doi: 10.3390/s19194154.

[62] S. Lu and T. Başar, "Robust nonlinear system identification using neural-network models," *IEEE Trans. Neural Networks*, vol. 9, no. 3, pp. 407–429, 1998, doi: 10.1109/72.668883.

[63] D. M. Li and F. C. Li, "Identification of chaotic systems with noisy data based on RBF neural networks," *Proc. 2009 Int. Conf. Mach. Learn. Cybern.*, vol. 5, no. July, pp. 2578–2581, 2009, doi: 10.1109/ICMLC.2009.5212655.

[64] P. D. Groves, "Navigation using inertial sensors [Tutorial]," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 30, no. 2, pp. 42–69, Feb. 2015, doi: 10.1109/MAES.2014.130191.

[65] J. Melo and A. Matos, "Survey on advances on terrain based navigation for autonomous underwater vehicles," *Ocean Eng.*, vol. 139, no. September 2015, pp. 250–264, Jul. 2017, doi: 10.1016/j.oceaneng.2017.04.047.

[66] J. González-García, A. Gómez-Espinosa, E. Cuan-Urquizo, L. G. García-Valdovinos, T. Salgado-Jiménez, and J. A. Escobedo Cabello, "Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions," *Appl. Sci.*, vol. 10, no. 4, 2020, doi: 10.3390/app10041256.

[67] I. Ullah, J. Chen, X. Su, C. Esposito, and C. Choi, "Localization and Detection of Targets in Underwater Wireless Sensor Using Distance and Angle Based Algorithms," *IEEE Access*, vol. 7, pp. 45693–45704, 2019, doi: 10.1109/ACCESS.2019.2909133.

[68] U. M. Qureshi *et al.*, "RF path and absorption loss estimation for underwaterwireless sensor networks in differentwater environments," *Sensors (Switzerland)*, vol. 16, no. 6, 2016, doi: 10.3390/s16060890.

[69] R. Diversi, R. Guidorzi, and U. Soverini, "Kalman filtering in extended noise environments," *IEEE Trans. Automat. Contr.*, vol. 50, no. 9, pp. 1396–1402, Sep. 2005, doi: 10.1109/TAC.2005.854627.

[70] J. Almeida, B. Matias, A. Ferreira, C. Almeida, A. Martins, and E. Silva, "Underwater localization system combining iusbl with dynamic sbl in ¡vamos! trials," *Sensors (Switzerland)*, vol. 20, no. 17, pp. 1–23, 2020, doi: 10.3390/s20174710.

[71] N. Y. Ko, S. Jeong, S. S. Hwang, and J. Y. Pyun, "Attitude estimation of underwater vehicles using field measurements and bias compensation," *Sensors (Switzerland)*, vol. 19, no. 2, pp. 1–21, 2019, doi: 10.3390/s19020330.

[72] H. Huang, X. Chen, Z. Zhou, Y. Xu, and C. Lv, "Study of the algorithm of backtracking decoupling and adaptive extended kalman filter based on the quaternion expanded to the state variable for underwater glider navigation," *Sensors (Switzerland)*, vol. 14, no. 12, pp. 23041–23066, 2014, doi: 10.3390/s141223041.

[73] A. Miller, B. Miller, and G. Miller, "On AUV control with the aid of position estimation algorithms based on acoustic seabed sensing and DOA measurements," *Sensors (Switzerland)*, vol. 19, no. 24, pp. 1–21, 2019, doi: 10.3390/s19245520.

[74] A. Tal, I. Klein, and R. Katz, "Inertial navigation system/doppler velocity log (INS/DVL) fusion with partial dvl measurements," *Sensors (Switzerland)*, vol. 17, no. 2, pp. 1–20, 2017, doi: 10.3390/s17020415.

[75] M. Zhang, K. Li, B. Hu, and C. Meng, "Comparison of Kalman Filters for Inertial Integrated Navigation.," *Sensors (Basel).*, vol. 19, no. 6, Mar. 2019, doi: 10.3390/s19061426.

[76] C. Sun, Y. Zhang, G. Wang, and W. Gao, "A new variational bayesian adaptive extended kalman filter for cooperative navigation," *Sensors (Switzerland)*, vol. 18, no. 8, 2018, doi: 10.3390/s18082538.

[77] Linzhouting Chen and Jiancheng Fang, "A Hybrid Prediction Method for Bridging GPS Outages in High-Precision POS Application," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 6, pp. 1656–1665, Jun. 2014, doi: 10.1109/TIM.2013.2292277.

[78] Z. Jingsen, Z. Wenjie, H. Bo, and W. Yali, "Integrating Extreme Learning Machine with Kalman Filter to Bridge GPS Outages," *Proc. - 2016 3rd Int. Conf. Inf. Sci. Control Eng. ICISCE 2016*, pp. 420–424, 2016, doi: 10.1109/ICISCE.2016.98.

[79] X.-L. Huang, X. Ma, and F. Hu, "Editorial: Machine Learning and Intelligent Communications," *Mob. Networks Appl.*, vol. 23, no. 1, pp. 68–70, Feb. 2018, doi: 10.1007/s11036-017-0962-2.

[80] E. E. Tsiropoulou, G. Mitsis, and S. Papavassiliou, "Interest-aware energy collection & resource management in machine to machine communications," *Ad Hoc Networks*, vol. 68, no. September, pp. 48–57, 2018, doi: 10.1016/j.adhoc.2017.09.003.

[81] X. Zhang, X. Mu, H. Liu, B. He, and T. Yan, "Application of Modified EKF Based on Intelligent Data Fusion in AUV Navigation," in *2019 IEEE Underwater Technology (UT)*, Apr. 2019, pp. 1–4, doi: 10.1109/UT.2019.8734414.

[82] M. T. Sabet, H. M. Daniali, A. Fathi, and E. Alizadeh, "Identification of an Autonomous Underwater Vehicle Hydrodynamic Model Using the Extended, Cubature, and Transformed Unscented Kalman Filter," *IEEE J. Ocean. Eng.*, vol. 43, no. 2, pp. 457–467, 2018, doi: 10.1109/JOE.2017.2694470.

[83] B. Allotta *et al.*, "A new AUV navigation system exploiting unscented Kalman filter," *Ocean Engineering*, vol. 113. pp. 121–132, 2016, doi: 10.1016/j.oceaneng.2015.12.058.

[84] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Automat. Contr.*, vol. 45, no. 3, pp. 477–482, Mar. 2000, doi: 10.1109/9.847726.

[85] M. Karimi, M. Bozorg, and A. R. Khayatian, "A comparison of DVL/INS fusion by UKF and EKF to localize an autonomous underwater vehicle," in *2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, Feb. 2013, pp. 62–67, doi: 10.1109/ICRoM.2013.6510082.

[86] J. V. Tsyganova and M. V. Kulikova, "SVD-Based Kalman Filter Derivative Computation," *IEEE Trans. Automat. Contr.*, vol. 62, no. 9, pp. 4869–4875, Sep. 2017, doi: 10.1109/TAC.2017.2694350.

[87] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "On the complexity and consistency of UKF-based SLAM," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 4401–4408, doi: 10.1109/ROBOT.2009.5152793.

[88] D. Simon, "Training radial basis neural networks with the extended Kalman filter," *Neurocomputing*, vol. 48, no. 1–4, pp. 455–475, 2002, doi: 10.1016/S0925-2312(01)00611-7.

[89] Y. Wang, S. Chai, and H. D. Nguyen, "Experimental and numerical study of autopilot using Extended Kalman Filter trained neural networks for surface vessels," *Int. J. Nav. Archit. Ocean Eng.*, vol. 12, pp. 314–324, 2020, doi: 10.1016/j.ijnaoe.2019.11.004.

[90] T. Kurban and E. Beşdok, "A Comparison of RBF Neural Network Training Algorithms for Inertial Sensor Based Terrain Classification," *Sensors*, vol. 9, no. 8, pp. 6312–6329, Aug. 2009, doi: 10.3390/s90806312.

[91] X. Dong, J. Wu, S. Wang, and T. Chen, "An improved CDKF algorithm based on RBF neural network for satellite attitude determination," *Proc. 2012 Int. Conf. Image Anal. Signal Process. IASP 2012*, pp. 155–161, 2012, doi: 10.1109/IASP.2012.6425013.

[92] V. Pesce, S. Silvestrini, and M. Lavagna, "Radial basis function neural network aided adaptive extended Kalman filter for spacecraft relative navigation," *Aerosp. Sci. Technol.*, vol. 96, p. 105527, 2020, doi: 10.1016/j.ast.2019.105527.

[93] Tianping Chen and Hong Chen, "Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks," *IEEE Trans. Neural Networks*, vol. 6, no. 4, pp. 904–910, Jul. 1995, doi: 10.1109/72.392252.

[94] A. P. Markopoulos, S. Georgiopoulos, and D. E. Manolakos, "On the use of back propagation and radial basis function neural networks in surface roughness prediction," *J. Ind. Eng. Int.*, vol. 12, no. 3, pp. 389–400, Sep. 2016, doi: 10.1007/s40092-016-0146-x.

[95] H. Yu, T. Xie, S. Paszczynski, and B. M. Wilamowski, "Advantages of Radial Basis Function Networks for Dynamic System Design," *IEEE Trans. Ind. Electron.*, vol. 58, no. 12, pp. 5438–5450, Dec. 2011, doi: 10.1109/TIE.2011.2164773.

[96] Y. Wu, H. Wang, B. Zhang, and K.-L. Du, "Using Radial Basis Function Networks for Function Approximation and Classification," *ISRN Appl. Math.*, vol. 2012, pp. 1–34, 2012, doi: 10.5402/2012/324194.

[97] J. Solà, "Quaternion kinematics for the error-state Kalman filter," *arXiv*, Nov. 2017, [Online]. Available: http://www.billion.uk.com/downloads/user manual/annex_m.pdf.

[98] H. A. Bjaili, M. Moinuddin, and A. M. Rushdi, "A State-Space Backpropagation Algorithm for Nonlinear Estimation," *Circuits, Syst. Signal Process.*, vol. 38, no. 8, pp. 3682–3696, 2019, doi: 10.1007/s00034-019-01031-2.

[99] Y. Zhao, D. Wang, and L. Wang, "Convolution Accelerator Designs Using Fast Algorithms," *Algorithms*, vol. 12, no. 5, p. 112, May 2019, doi: 10.3390/a12050112.

[100] F. Dul, P. Lichota, and A. Rusowicz, "Generalized Linear Quadratic Control for a Full Tracking Problem in Aviation," *Sensors*, vol. 20, no. 10, p. 2955, May 2020, doi: 10.3390/s20102955.

[101] L. Wei, C. Cappelle, and Y. Ruichek, "Camera/Laser/GPS Fusion Method for Vehicle Positioning Under Extended NIS-Based Sensor Validation," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 11, pp. 3110–3122, Nov. 2013, doi: 10.1109/TIM.2013.2265476.

[102] M. D. Pham, K. S. Low, S. T. Goh, and S. Chen, "Gain-scheduled extended kalman filter for nanosatellite attitude determination system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 2, pp. 1017–1028, Apr. 2015, doi: 10.1109/TAES.2014.130204.

[103] T. Zhang, L. Chen, and Y. Yan, "Underwater Positioning Algorithm Based on SINS/LBL Integrated System," *IEEE Access*, vol. 6, pp. 7157–7163, 2018, doi: 10.1109/ACCESS.2018.2795799.

[104] R. Woo, E. J. Yang, and D. W. Seo, "A fuzzy-innovation-based adaptive Kalman filter for enhanced vehicle positioning in dense urban environments," *Sensors (Switzerland)*, vol. 19, no. 5, 2019, doi: 10.3390/s19051142.

[105] G. Abdelnour, S. Chand, and S. Chiu, "Applying fuzzy logic to the Kalman filter divergence problem," in *Proceedings of IEEE Systems Man and Cybernetics Conference - SMC*, 1993, pp. 630–635, doi: 10.1109/ICSMC.1993.384814.

[106] Z. L. Ren, L. G. Wang, and L. Bi, "Improved Extended Kalman Filter Based on Fuzzy

Adaptation for SLAM in Underground Tunnels," *Int. J. Precis. Eng. Manuf.*, vol. 20, no. 12, pp. 2119–2127, 2019, doi: 10.1007/s12541-019-00222-w.

[107] J. Lalk, "Intelligent adaptation of Kalman filters using fuzzy logic," *IEEE Int. Conf. Fuzzy Syst.*, vol. 2, pp. 744–749, 1994, doi: 10.1109/fuzzy.1994.343829.

[108] C. H. Do and H. Y. Lin, "Incorporating neuro-fuzzy with extended Kalman filter for simultaneous localization and mapping," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 5, pp. 1–13, 2019, doi: 10.1177/1729881419874645.

[109] C. T. Fraser and S. Ulrich, "A fuzzy adaptive kalman filter for spacecraft formation navigation," *Proc. Am. Control Conf.*, vol. 2019-July, pp. 2527–2533, 2019, doi: 10.23919/acc.2019.8814948.

[110] A. Gunduz and J. C. Principe, "Correntropy as a novel measure for nonlinearity tests," *Signal Processing*, vol. 89, no. 1, pp. 14–23, 2009, doi: 10.1016/j.sigpro.2008.07.005.

[111] G. Wang, Z. Gao, Y. Zhang, and B. Ma, "Adaptive maximum correntropy gaussian filter based on variational bayes," *Sensors (Switzerland)*, vol. 18, no. 6, pp. 1–14, 2018, doi: 10.3390/s18061960.

[112] A. Sahoo, S. K. Dwivedy, and P. S. Robi, "Advancements in the field of autonomous underwater vehicle," *Ocean Eng.*, vol. 181, no. March, pp. 145–160, 2019, doi: 10.1016/j.oceaneng.2019.04.011.

[113] L. Zhang, D. Sidoti, A. Bienkowski, K. R. Pattipati, Y. Bar-Shalom, and D. L. Kleinman, "On the identification of noise covariances and adaptive kalman filtering: A new look at a 50 year-old problem," *IEEE Access*, vol. 8, no. February, pp. 59362–59388, 2020, doi: 10.1109/ACCESS.2020.2982407.

[114] S. Li, X. Jiang, and Y. Liu, "Innovative Mars entry integrated navigation using modified multiple model adaptive estimation," *Aerosp. Sci. Technol.*, vol. 39, pp. 403–413, 2014, doi: 10.1016/j.ast.2014.04.009.

[115] A. H. Mohamed and K. P. Schwarz, "Adaptive Kalman Filtering for INS/GPS," *J. Geod.*, vol. 73, no. 4, pp. 193–203, May 1999, doi: 10.1007/s001900050236.

[116] B. Or, B. Z. Bobrovsky, and I. Klein, "Kalman Filtering with Adaptive Step Size Using a Covariance based Criterion," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, doi: 10.1109/TIM.2021.3063191.

[117] Y. Huang, Y. Zhang, Z. Wu, N. Li, and J. Chambers, "A Novel Adaptive Kalman Filter with Inaccurate Process and Measurement Noise Covariance Matrices," *IEEE Trans. Automat. Contr.*, vol. 63, no. 2, pp. 594–601, 2018, doi: 10.1109/TAC.2017.2730480.

[118] D. Li, D. Ji, J. Liu, and Y. Lin, "A Multi-Model EKF Integrated Navigation Algorithm for Deep Water AUV," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 1, 2016, doi: 10.5772/62076.

[119] Z. C. Deng, X. Yu, H. De Qin, and Z. Ben Zhu, "Adaptive kalman filter-based single-beacon underwater tracking with unknown effective sound velocity," *Sensors (Switzerland)*, vol. 18, no. 12, 2018, doi: 10.3390/s18124339.

[120] D. Wang, X. Xu, and L. Hou, "An Improved Adaptive Kalman Filter for Underwater SINS/DVL System," *Math. Probl. Eng.*, vol. 2020, pp. 1–14, Aug. 2020, doi: 10.1155/2020/5456961.

[121] K. He, H. Liu, and Z. Wang, "A novel adaptive two-stage information filter approach for deep-sea USBL/DVL integrated navigation," *Sensors (Switzerland)*, vol. 20, no. 21, pp. 1–20, 2020, doi: 10.3390/s20216029.

[122] N. Davari and A. Gholami, "Variational Bayesian adaptive Kalman filter for asynchronous multirate multi-sensor integrated navigation system," *Ocean Eng.*, vol. 174, no. December 2017, pp. 108–116, 2019, doi: 10.1016/j.oceaneng.2019.01.012.

[123] X. Zhang, B. He, G. Li, X. Mu, Y. Zhou, and T. Mang, "Navnet: AUV Navigation through Deep Sequential Learning," *IEEE Access*, vol. 8, pp. 59845–59861, 2020, doi: 10.1109/ACCESS.2020.2982272.

[124] X. Mu, B. He, X. Zhang, Y. Song, Y. Shen, and C. Feng, "End-to-end navigation for Autonomous Underwater Vehicle with Hybrid Recurrent Neural Networks," *Ocean Eng.*, vol. 194, no. October, 2019, doi: 10.1016/j.oceaneng.2019.106602.

[125] X. Gao *et al.*, "RL-AKF: An adaptive kalman filter navigation algorithm based on reinforcement learning for ground vehicles," *Remote Sens.*, vol. 12, no. 11, 2020, doi: 10.3390/rs12111704.

[126] G. Salavasidis *et al.*, "Terrain-aided navigation for long-endurance and deep-rated autonomous underwater vehicles," *J. F. Robot.*, vol. 36, no. 2, p. rob.21832, Nov. 2018, doi: 10.1002/rob.21832.

[127] P. Oliveira, "MMAE terrain reference navigation for underwater vehicles using PCA," *Int. J. Control*, vol. 80, no. 7, pp. 1008–1017, Jul. 2007, doi: 10.1080/00207170701242515.

[128] J. Lee and H. Bang, "A Robust Terrain Aided Navigation Using the Rao-Blackwellized Particle Filter Trained by Long Short-Term Memory Networks," *Sensors*, vol. 18, no. 9, p. 2886, Aug. 2018, doi: 10.3390/s18092886.

[129] J. Z. Sasiadek and Q. Wang, "Sensor fusion based on fuzzy Kalman filtering for autonomous robot vehicle," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 4, no. May, pp. 2970–2975, 1999, doi: 10.1109/robot.1999.774048.

[130] B. Liu, J. Xu, B. Fu, Y. Hao, and T. An, "A covariance shaping filtering method for tightly-coupled MIMU/GNSS of UAV," *Aircr. Eng. Aerosp. Technol.*, vol. 91, no. 10, pp. 1257–1267, 2019, doi: 10.1108/AEAT-07-2018-0211.

[131] H. Deilamsalehy and T. C. Havens, "Fuzzy adaptive extended Kalman filter for robot 3D pose estimation," *Int. J. Intell. Unmanned Syst.*, vol. 6, no. 2, pp. 50–68, 2018, doi: 10.1108/IJIUS-12-2017-0014.

[132] D. J. Jwo and T. S. Cho, "A practical note on evaluating Kalman filter performance optimality and degradation," *Appl. Math. Comput.*, vol. 193, no. 2, pp. 482–505, 2007, doi: 10.1016/j.amc.2007.04.008.

[133] D.-J. Jwo, F.-C. Chung, and T.-P. Weng, "Adaptive Kalman Filter for Navigation Sensor Fusion," in *Sensor Fusion and its Applications*, Sciyo, 2010.

[134] C. T. Fraser and S. Ulrich, "Adaptive extended Kalman filtering strategies for spacecraft formation relative navigation," *Acta Astronaut.*, vol. 178, no. July 2020, pp. 700–721, 2021, doi: 10.1016/j.actaastro.2020.10.016.

[135] J. Vaganay, J. J. Leonard, and J. G. Bellingham, "Outlier rejection for autonomous acoustic navigation," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 3, no. April, pp. 2174–2181, 1996, doi: 10.1109/robot.1996.506191.

[136] S. Fakoorian, R. Izanloo, A. Shamshirgaran, and D. Simon, "Maximum Correntropy Criterion Kalman Filter with Adaptive Kernel Size," *Proc. IEEE Natl. Aerosp. Electron. Conf. NAECON*, vol. 2019-July, pp. 581–584, 2019, doi: 10.1109/NAECON46414.2019.9057886.

[137] Y. Huang, Y. Zhang, P. Shi, Z. Wu, J. Qian, and J. A. Chambers, "Robust Kalman Filters Based on Gaussian Scale Mixture Distributions with Application to Target Tracking," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 10, pp. 2082–2096, 2019, doi: 10.1109/TSMC.2017.2778269.

[138] N. Shaukat, A. Ali, M. J. Iqbal, M. Moinuddin, and P. Otero, "Multi-sensor fusion for underwater vehicle localization by augmentation of rbf neural network and error-state kalman filter," *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–26, 2021, doi: 10.3390/s21041149.

[139] J. Z. Sasiadek and J. Khe, "Sensor fusion based on fuzzy Kalman filter," *Proc. 2nd Int. Work. Robot Motion Control. RoMoCo 2001*, no. 7, pp. 275–283, 2001, doi: 10.1109/ROMOCO.2001.973467.

[140] D. J. Jwo and S. H. Wang, "Adaptive fuzzy strong tracking extended Kalman filtering for GPS navigation," *IEEE Sens. J.*, vol. 7, no. 5, pp. 778–789, 2007, doi: 10.1109/JSEN.2007.894148.

[141] A. L. da Silva and J. J. da Cruz, "Fuzzy adaptive extended Kalman filter for UAV INS/GPS data fusion," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 38, no. 6, pp. 1671–1688, 2016, doi: 10.1007/s40430-016-0509-7.

[142] J. Zhao, H. Zhang, and G. Wang, "Projected Kernel Recursive Maximum Correntropy," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 65, no. 7, pp. 963–967, 2018, doi: 10.1109/TCSII.2017.2778302.

[143] R. Izanloo, S. A. Fakoorian, H. S. Yazdi, and D. Simon, "Kalman filtering based on the maximum correntropy criterion in the presence of non-Gaussian noise," *2016 50th Annu. Conf. Inf. Syst. Sci. CISS 2016*, pp. 500–505, 2016, doi: 10.1109/CISS.2016.7460553.

[144] C. C. Tsai, H. H. Lin, and J. C. Hsu, "Fuzzy adaptive extended information filtering," *Int. J. Fuzzy Syst.*, vol. 7, no. 1, pp. 31–38, 2005.

[145] F. Huang, J. Zhang, and S. Zhang, "Maximum Versoria Criterion-Based Robust Adaptive Filtering Algorithm," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 64, no. 10, pp. 1252–1256, 2017, doi: 10.1109/TCSII.2017.2671521.

Appendix A

Curriculum Vitae

Experience

- Faculty Member/Research Engineer, Electrical and Electronics Department 2012–date
  Royal Commission Colleges and Institute, Yanbu Industrial City, Al Madinah, Saudi Arabia
- Manager (Research and Development) 2003–2011
  AERO, Karachi, Pakistan
- Visiting Faculty Member, Electronic Engineering Department 2008–2012
  IQRA University of Engineering & Technology, Karachi, Pakistan

Education

Doctor of Philosophy in Telecommunication Engineering (Thesis submitted)
University of Malaga, Málaga, Spain

Master in Electronics Engineering 2005–2007
NED University of Engineering & Technology, Karachi, Pakistan

Bachelor in Industrial Electronic Engineering 1999–2003
IIEE, NED University of Engineering & Technology, Karachi, Pakistan

Training / certifications

- Project Management Professional (PMP)
- NI Multisim & ELVIS Hands-on training, USA
- FPGA Design Expert, Singapore
- NI LabVIEW Core-II, USA
- NI LabVIEW Core-I, USA
- NI Control Design & Robotics Hands-on training, USA
- NI LabVIEW Hands-on training, USA
- NI FPGA Hands-on training, USA
- Freescale Motorola MPC55xx Microcontroller Architecture
- Advance Research Methods and Experiment design Training
- TPU Enhance Time Processing
- Real-time time embedded operating system
- Six Sigma Yellow Belt
- Programming Logic Controllers
- Matlab and Simulink Trainings
- Huawei Certified Network Associate (HCNA)
- Huawei Advance Telecoms Networks Training
- Wireless Networking Training
- Machine learning and Artificial Neural Networks
- Self-driving car
- Data Fusion
- Telecommunication system design

Publications

- N. Shaukat and P. Otero, “Underwater Vehicle Positioning by Fuzzy and Neural Adaptive Kalman Sensor Fusion,” OCEANS 2021: San Diego – Porto, Sep. 2021, doi: 10.23919/oceans44145.2021.9705963.

- N. Shaukat, M. Moinuddin, and P. Otero, “Underwater Vehicle Positioning by Correntropy-Based Fuzzy Multi-Sensor Fusion,” *Sensors*, vol. 21, no. 18, p. 6165, Sep. 2021.

- N. Shaukat, A. Ali, M. Moinuddin and P. Otero, "Underwater Vehicle Localization by Hybridization of Indirect Kalman Filter and Neural Network," 2021 7th International Conference on Mechatronics and Robotics Engineering (ICMRE), 2021, pp. 111-115, doi: 10.1109/ICMRE51691.2021.9384844.

- N. Shaukat, A. Ali, M. Javed Iqbal, M. Moinuddin, and P. Otero, “Multi-Sensor Fusion for Underwater Vehicle Localization by Augmentation of RBF Neural Network and Error-State Kalman Filter,” *Sensors*, vol. 21, no. 4, p. 1149, Feb. 2021.

- F. A. Banakhr, M. J. Iqbal and N. Shaukat, "Active project based learning pedagogies: Learning hardware, software design and wireless sensor instrumentation," 2018 IEEE Global Engineering Education Conference (EDUCON), 2018, pp. 1870-1874, doi: 10.1109/EDUCON.2018.8363463.

- A. M. Dabroom, M. J. Iqbal, T. T. Tay, N. Shaukat and S. Mahamood, "TTC process control trainer: Motivations and the design of a laboratory trainer to meet Saudi Arabia educational needs," 2014 IEEE Global Engineering Education Conference (EDUCON), 2014, pp. 283-288, doi: 10.1109/EDUCON.2014.6826104.

- N. Shaukat, "PLC based automatic liquid filling process," International Multi Topic Conference, 2002. Abstracts. INMIC 2002., 2002, pp. 39-39, doi: 10.1109/INMIC.2002.1310146.

Professional memberships

- Pakistan Engineering Council, Life time member
- Project Management Institute (PMI)
- Marine Technology Society, USA