



НАЦИОНАЛЕН ВОЕНЕН УНИВЕРСИТЕТ „ВАСИЛ ЛЕВСКИ“
ФАКУЛТЕТ „АРТИЛЕРИЯ, ПВО И КИС“

9700 гр. Шумен, ул. „Карел Шкорпил“ №1
телефон: (054) 801 040; тел.факс:(054) 877 463;
email: decanat@aadcf.nvu.bg

асистент инж. Владислав Милков Генов

**ИНТЕГРИРАНА НАВИГАЦИОННА СИСТЕМА
ЗА УПРАВЛЕНИЕ НА БЕЗПИЛОТНИ ЛЕТАТЕЛНИ АПАРАТИ
С ИЗПОЛЗВАНЕ НА ИЗКУСТВЕН ИНТЕЛЕКТ**

АВТОРЕФЕРАТ

на дисертационен труд за присъждане на
образователно-научна степен "ДОКТОР"

Докторска програма по научна специалност:

„Автоматизирани системи за обработка на информация и управление“

НАУЧНА ОБЛАСТ: 5. ТЕХНИЧЕСКИ НАУКИ

ПРОФЕСИОНАЛНО НАПРАВЛЕНИЕ 5.3. КОМУНИКАЦИОННА И КОМПЮТЪРНА
ТЕХНИКА

Научен ръководител:

Подп. доц. д-р инж. Красимир Огнянов Славянов

Научен консултант:

Гл. ас. д-р инж. Иван Атанасов Попов

ШУМЕН

2026

Дисертационния труд се състои от 202 листа
Брой на таблиците – 22
Брой на фигурите – 34
Брой на приложенията – 2
Брой на литературните източници – 96
Брой на публикациите по дисертацията – 4

Защитата на дисертационния труд ще се състои на __. __. 2026 г. от __. __ ч. в
_____ на _____.

Материалите по защитата са на разположение на интересуващите се в _____
на _____, тел. _____.

Дисертационният труд е обсъден на катедрен съвет на катедра „Компютърни системи и технологии“ при факултет „Артилерия, ПВО и КИС“ на Национален военен университет „Васил Левски“ – Велико Търново на __. __. 20... г., и е насрочен за защита пред научно жури за придобиване на образователната и научна степен “Доктор”.

Докторантът е асистент в катедра „Компютърни системи и технологии“ при факултет „Артилерия, ПВО и КИС“ на Национален военен университет „Васил Левски“ – Велико Търново. Основните изследвания по дисертационния труд са проведени в развойната програмна среда MATLAB/Simulink.

Автор: инж. Владислав Милков Генев

Тема: „Интегрирана навигационна система за управление на безпилотни летателни апарати с използване на изкуствен интелект ”.

Тираж __

Отпечатан на __. __. 20__ г.

Издателски комплекс на Национален военен университет „Васил Левски“

I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД.

Мотиви за разработка на дисертационния труд

Глобалното развитие на автономните безпилотни летателни апарати (БЛА; англ. UAV) се характеризира с ясно изразена тенденция към преминаване от изолирани навигационни алгоритми, базирани на силно идеализирани аналитични допускания, към интегрирани навигационни системи и мултисензорно оценяване на състоянието. В този контекст методите на машинното обучение (МО; англ. ML) и дълбокото обучение се използват все по-често за интерпретация на сензорни потоци, моделиране на нелинейни зависимости и подпомагане на процесното моделиране при сложно поведение на системата. Обзорни публикации в областта на приложенията на дълбокото обучение при БЛА подчертават нарастващата роля на данните и на последователната обработка в реално време при навигационни и сродни задачи (Carrío, Sampedro, Rodríguez-Ramos, & Campoy, 2017).

На тази основа изследователската линия „Калманова филтрация + невронни мрежи“ се утвърждава като практически подход за комбиниране на структурираното вероятно оценяване (чрез стандартната схема *прогнозиране–актуализация*) с обучаеми модели на динамика при частично известни или неточно зададени процеси. Представителен пример за това направление са невронно подпомогнатите Калманови методи, при които се запазва state-space структурата на оценителя, а част от операциите или моделите се учат от данни (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022).

Актуалност на проблема

В редица практически приложения със завишени изисквания към сигурността и надеждността се налага постоянен мониторинг на въздушното пространство и контрол на достъпа до обекти със стратегическо значение. В тези условия експлоатацията на БЛА често се осъществява в среда с деградирани или умислено компрометирани GNSS сигнали, включително чрез заглушаване (**jamming**) и подправяне (**spoofing**). Вследствие на това робастната навигация при недостъпен или компрометиран GNSS (**GNSS-denied**/compromised среда) се превръща в съществено инженерно и научно предизвикателство (Elmas, Elif Ece; Alkan, Mustafa, 2024).

Паралелно с това, интеграцията между инерциални навигационни системи (INS) и глобални навигационни спътникови системи (GNSS) остава базов подход за навигация на БЛА, тъй като комбинира автономността на INS с дългосрочната стабилност на GNSS. Въпреки това точността в режими на числение (**dead reckoning**) се ограничава от натрупването на грешки, обусловени от параметрите и температурната стабилност на инерциалния измервателен блок (IMU), включително при MEMS IMU, което прави системата чувствителна при отсъствие на външни корекции.

Цел и задачи на дисертационния труд

Основната цел на дисертационния труд е разработване и експериментална проверка на интегрирана навигационна система за БЛА, при която аналитичният динамичен модел в рамката на Калмановата филтрация се допълва или замества с рекурентна невронна мрежа с дълга краткосрочна памет (Long Short-Term Memory – LSTM). Целта е повишаване на точността и устойчивостта на оценяването при нелинейни динамики, променящи се шумови характеристики и деградации на GNSS сигналите.

Произтичащите от поставената цел основни задачи са:

1. Да се извърши сравнителен анализ на съвременни навигационни подходи и методи за интегриране на сензорни данни при БЛА, с акцент върху ограниченията на KF, EKF и UKF при нелинейност и нестационарни шумове.
2. Да се формализират постановките за интегрирана навигация и да се дефинират критерии за оценка на точност, устойчивост и консистентност при наличен, деградиращ и прекъсващ GNSS.
3. Да се създадат симулационни сценарии (включително GNSS загуби, шумови профили и смущения) и набори от данни за обучение и валидация.
4. Да се извърши сравнителен анализ на съществуващите невронни мрежи и да се осъществи обоснован избор на подходяща невронна архитектура като процесен модел (или като

компонент, подпомагащ процесното моделиране/адаптацията на параметри) в структурата на филтриращия алгоритъм.

5. Да се разработи симулационен експеримент в MATLAB/Simulink за провеждане на числени експерименти и сравнителна оценка спрямо EKF/KF и избрани хибридни подходи.

Предмет на дисертационния труд

Предмет на изследване са алгоритмите и методите за оценяване на навигационното състояние (позиция, скорост и ориентация) на БЛА чрез интегриране на сензорни данни от INS/GNSS, както и включването на обучаеми рекурентни невронни модели за подобряване на процесното моделиране при нелинейности, режимно-зависимо поведение и деградирани измервания.

Средства за постигане на целта

За постигане на целта се използват методи от статистическото оценяване и филтрация на случайни процеси, анализ и моделиране на дискретновремени динамични системи, както и методи от изкуствения интелект (ИИ) за обработка на последователни данни и апроксимация на нелинейни зависимости. Експерименталната част е реализирана в MATLAB/Simulink, включително средства за симулационно моделиране, обучение/валидация на невронни мрежи и анализ на метрики и иновационни последователности.

Начални условия и допускане

Резултатите са получени в симулационна среда и подлежат на допълнителна проверка при по-реалистични условия, включително вариации на параметри, различни профили на движение и несъответствия между симулационни и реални данни. Оценяването състояние в разгледаната постановка е ограничено до позиция и линейна скорост; ориентационни състояния и съответни метрики не са включени в експерименталната част, поради което изводите следва да се тълкуват в този обхват. Представените количествени сравнения са формулирани спрямо базов Калманов филтър - разширен Калманов филтър (Extended Kalman Filter – EKF) в рамките на разгледаните сценарии. Разширено сравнение с допълнителни филтри и хибридни схеми изисква отделно експериментално изследване.

Научна новост

Научната новост се състои в предложена постановка за хибридно навигационно оценяване, при която се запазва Калмановата рамка като структурирана вероятностна схема за интегриране на измервания и оценяване на състоянието, а процесният (преходният) модел или ключови елементи на прогнозната стъпка се реализират чрез обучаем LSTM архитектура. По този начин се цели по-адекватно процесно моделиране при нелинейни режими и променящи се шумови характеристики, включително при деградации на GNSS. Подходът е концептуално съвместим с актуални направления в невронно подпомогнатата Калманова филтрация (напр. KalmanNet и рекурентни разширения), но е фокусиран върху приложимостта при INS/GNSS навигация на БЛА и експериментална проверка в симулационна среда (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022).

Практическа полезност

Практическата полезност на разработката е свързана с изграждане на симулационно-ориентирана софтуерна среда за оценка на навигационна точност и устойчивост при БЛА. Системата комбинира класически филтрационни блокове (KF/EKF/UKF) с обучаем LSTM модел за подпомагане на процесното моделиране в сценарии с шумове, GNSS деградации и прекъсвания. Компонентите са организирани модулно (генериране на сценарии и данни; филтрация и интеграция на измервания; обучение и валидация на невронни модели; сравнение на метрики и визуализации), което позволява поетапно надграждане и бърза проверка на решения при различни профили на мисията и сензорни характеристики.

II. СТРУКТУРА НА ДИСЕРТАЦИОННИЯТ ТРУД.

В дисертацията са анализирани методи от теорията на навигационните системи и от областта на ИИ, с акцент върху рекурентни невронни архитектури за моделиране на последователни данни и тяхната интеграция в задачи за оценка на състоянието. На тази основа е разработен хибриден

навигационен модел, при който LSTM се използва като процесен модел (или ключов елемент от него) в рамката на Калманова филтрация, и е проведен сравнителен анализ при симулационни сценарии със шумове, GNSS деградации и загуби.

- В **предговора** са посочени мотивите за разработка, актуалността на проблема, целта, основните задачи, предметът и средствата, научната новост и практическата полезност на разработката.
- В **първа глава** е представен обзор на навигационните системи за БЛА и подходите за интеграция INS/GNSS и допълващи сензори, както и теоретичната база на Калмановата филтрация и ограниченията ѝ в реалистични условия.
- Във **втора глава** е анализирано развитието на ИИ подходите за моделиране и управление на динамични системи, включително RNN/LSTM/GRU, невронни обикновени диференциални уравнения (Neural Ordinary Differential Equations – Neural ODE) и стохастични диференциални уравнения (Stochastic Differential Equations – SDE), и невронно подпомогнати Калманови методи.
- В **трета глава** е дефинирано предложеното решение и архитектурата на хибридният модел, включително постановката за допълване/замяна на процесния модел с LSTM, организацията на входно-изходните величини, процедурите по обучение и критериите за оценка.
- В **четвърта глава** са описани експерименталната среда и сценарии в MATLAB/Simulink, представени са резултатите от числените експерименти и е направено сравнение на точност и устойчивост при GNSS загуби спрямо EKF/UKF и избрани хибридни подходи, като са формулирани изводи за приложимостта и ограниченията на решението.

III. КРАТКО СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД.

ГЛАВА 1.

ОБЗОР НА СЪВРЕМЕННИТЕ ТЕХНОЛОГИИ ЗА ИНТЕГРИРАНИ НАВИГАЦИОННИ СИСТЕМИ, БЕЗПИЛОТНИ ЛЕТАТЕЛНИ АПАРАТИ И ИЗКУСТВЕН ИНТЕЛЕКТ .

1.1. Класификация на навигационните системи на безпилотни летателни апарати.

В съвременната специализирана литература навигационните системи на безпилотните летателни апарати (БЛА) най-често се разглеждат като архитектури за оценяване на навигационното състояние, които комбинират сензорни измервания и алгоритми за обработка на данни. Основната им цел е надеждно определяне на положение, скорост и ориентация в динамична среда, характеризираща се с шум, смущения и потенциално враждебни въздействия.

Класификацията на навигационните решения може да се извърши по различни признаци, като най-широко използваните са: (1) основен информационен източник, (2) ниво на интеграция между подсистемите и (3) степен на автономност спрямо външни инфраструктури. В рамките на настоящата секция се приема класификация по доминиращ източник на навигационна информация, при която системите условно се разделят на: инерциални навигационни системи (INS), системи на базата на глобални навигационни спътникови системи (GNSS) и интегрирани (хибридни) навигационни системи. Интегрираните решения комбинират множество източници на данни (напр. INS/GNSS, камера, LiDAR, барометрични и магнитометрични измервания, картова или радиобазирани информация) чрез алгоритми за оптимално оценяване, като широко приложение намират варианти на Калманова филтрация (EKF/UKF), както и оптимизационни подходи (напр. факторни графи) в зависимост от изискванията за точност и изчислителен ресурс.

Налице е отчетлива тенденция за преход от единични сензорни модули към интегрирани навигационни решения, тъй като в общия случай отделните подсистеми имат фундаментални ограничения при реална експлоатация. Това се аргументира с натрупването на грешка (дрейф) във времето при INS и уязвимостта на GNSS към смущения, загуба на сигнал и умишлени въздействия

като заглушаване (jamming) и подправяне (spoofing) (Mahmoud & Trilaksono, 2018). Следователно преходът към хибридни архитектури може да се разглежда като технологично обоснована необходимост, особено в сценарии с ограничена или компрометирана GNSS достъпност. За да се обоснове този преход по-пълно, е необходимо първо да се анализират принципите и ограниченията на инерциалната навигация..

Тази теза се потвърждава и в изследванията на Mahmoud и Trilaksono (Mahmoud & Trilaksono, 2018), които акцентират върху фундаменталните ограничения на отделните подсистеми: акумулирането на грешки във времето при INS и уязвимостта на GNSS към смущения, загуба на сигнал или умишлени атаки като заглушаване (jamming) и подправяне (spoofing) (Mahmoud & Trilaksono, 2018). Следователно, логическият преход към хибридни навигационни архитектури се явява технологична необходимост. За да се обоснове напълно този преход обаче, е необходимо първо да се анализират принципите и ограниченията на инерциалната навигация.

1.2. Калманови филтри – теоретични основи.

Калмановият филтър представлява общият рекурсивен Бейсов алгоритъм за филтрация, след допускане, че динамиката на системата и моделът на наблюдение се описват с линейни диференциални и алгебрични уравнения и всички шумове са с нормално (гаусово) разпределение. При тези условия оценката по критерия за минимална средноквадратична грешка (MMSE) може да се изчисли с Калманов филтър. Важно негово предимство е, че може да използва почти всякаква информация от наличните сензори, независимо колко често обновяват данните и колко точни са измерванията им.

Най-общо Калмановата филтрация може да се определи като процес на оптимално оценяване на състоянията на реална динамична система, работеща под влиянието на случайни смущения. По-точно казано, Калмановият филтър представлява линеен рекурсивен алгоритъм, който реконструира неизвестните състояния на системата от наблюдения на зашумения ѝ изход в дискретно реално време (Попов, 2022).

Динамиката на системата се описва чрез линеен дискретен модел представен с израза:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k, \quad (1.18)$$

където \mathbf{x}_k е векторът на състоянието в момент k , \mathbf{u}_k е управляващият вход, а \mathbf{w}_k е процесният шум с ковариационна матрица \mathbf{Q}_k . Матриците \mathbf{F}_k , \mathbf{B}_k и \mathbf{G}_k описват съответно динамиката на състоянието, влиянието на управляващите въздействия и начина, по който процесният шум въздейства върху системата. Измервателният модел се задава съгласно:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \quad (1.19)$$

където \mathbf{z}_k е векторът на измерванията, \mathbf{H}_k е матрицата на измервателния модел, а \mathbf{v}_k е измервателният шум с ковариация \mathbf{R}_k .

Алгоритъмът на Калмановия филтър се реализира чрез рекурсивна процедура в два основни етапа. В прогнозната етап се изчислява априорната оценка на състоянието и съответната ковариационна матрица:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \mathbf{u}_k, \\ \mathbf{P}_k^- &= \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T. \end{aligned} \quad (1.20)$$

Ковариационната матрица \mathbf{P}_k^- представлява статистическа мярка за увереността на прогнозната оценка и отразява натрупаната неопределеност вследствие на процесния шум и динамиката на системата.

При наличие на измерване се извършва корекционният етап, при който прогнозата се актуализира чрез иновационния вектор и коефициента на усилване на Калман:

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-), \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \end{aligned} \quad (1.21)$$

Калмановият коефициент на усилване \mathbf{K}_k изпълнява ролята на тегловен фактор, който балансира влиянието на измерванията спрямо прогнозата на модела. При ниска измервателна

неопределеност корекцията се доминира от измерванията, докато при висока шумова компонента филтърът разчита в по-голяма степен на динамичния модел.

Тази формална структура осигурява стабилна и оптимална оценка при линейни системи с гаусови шумове и представлява теоретичната основа, върху която се изграждат разширените и хибридни варианти на Калмановата филтрация, разглеждани в следващите подраздели (Попов, 2022).

1.2.1. Разширен Калманов филтър (ЕКФ) – линеаризация на нелинейни системи.

В реални навигационни приложения динамиката на системата и измервателните зависимости рядко могат да бъдат описани чрез строго линейни модели. Това налага използването на разширения Калманов филтър (Extended Kalman Filter – EKF), който представлява обобщение на класическия Калманов филтър за нелинейни системи чрез локална линеаризация около текущата оценка на състоянието. Нелинейният дискретен модел на системата се задава във вида:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{G}_k \mathbf{w}_k, \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \end{aligned} \quad (1.22)$$

където $\mathbf{f}(\cdot)$ и $\mathbf{h}(\cdot)$ са нелинейни функции, описващи съответно динамиката на състоянието и измервателния процес. Процесният и измервателният шум запазват същите статистически допускания, както при класическия Калманов филтър.

Основната идея на EKF се състои в апроксимиране на нелинейните функции чрез първия член на реда на Тейлър, развит около текущата апостериорна оценка на състоянието. Това води до дефиниране на Якобианите на динамичната и измервателната функция:

$$\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k}, \mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_k}. \quad (1.23)$$

След тази линеаризация прогнозният етап на EKF формално запазва структурата на класическия филтър:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k), \\ \mathbf{P}_k^- &= \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T. \end{aligned} \quad (1.24)$$

Корекционният етап също следва стандартната Калманова схема, като използва Якобиана \mathbf{H}_k вместо линейна измервателна матрица. По този начин EKF позволява обработка на нелинейни системи, като комбинира нелинейна прогноза със статистически оптимална корекция.

Въпреки широкото си приложение, EKF притежава съществени ограничения. Линеаризацията е валидна само локално и предполага малки отклонения около работната точка. При силно нелинейна динамика или при бързи маневри Якобианите могат да се окажат лоша апроксимация, което води до систематични грешки и дори до загуба на устойчивост на филтъра. Допълнително, изчисляването на аналитични Якобиани усложнява реализацията и увеличава риска от моделни неточности, особено при сложни многомерни системи.

Тези ограничения са ясно идентифицирани в класическите приложения на EKF за навигация на подвижни платформи и представляват основна мотивация за търсене на алтернативни или хибридни методи, при които нелинейната динамика се описва чрез обучаеми модели, а не чрез аналитична линеаризация (Попов, 2022).

1.3. Ограничения на класическите и разширени Калманови филтри.

Въпреки широкото си приложение и доказаната си ефективност, класическите Калманови филтри и техните разширени варианти притежават редица фундаментални ограничения, които се проявяват особено ясно при навигационни задачи за безпилотни летателни апарати. Тези ограничения не са следствие от неправилна реализация, а произтичат директно от математическите предпоставки, върху които е изградена теорията на Калмановата филтрация.

На първо място, всички варианти на Калмановия филтър предполагат, че **динамичният модел на системата е известен и адекватно описва реалната физика на процеса**. При линейния Калманов филтър това изискване се формулира чрез матриците на прехода \mathbf{F} и управлението \mathbf{B} ,

докато при EKF и UKF същата роля се изпълнява от нелинейната функция $f(\cdot)$. В реални навигационни системи обаче динамиката на БЛА е силно нелинейна, зависима от аеродинамични ефекти, турбулентност, вибрации и променливи режими на управление, които трудно могат да бъдат описани с фиксиран аналитичен модел. В резултат на това възниква **структурна грешка на модела**, която не може да бъде компенсирана единствено чрез настройка на шумовите ковариации.

Второ съществено ограничение е свързано с **шумовите допускания**. Класическата теория предполага гаусови, бели и стационарни шумове както в процесния модел, така и в измерванията. В практиката на INS/GNSS навигацията тези условия рядко са изпълнени. Шумът на инерциалните сензори проявява температурна зависимост, дрейф и корелирани компоненти, а GNSS измерванията са подложени на деградации, многолъчево разпространение на сигнала (multipath) и внезапни загуби на сигнал. При такива условия фиксираните матрици Q и R водят до некоректна оценка на несигурността и до субоптимално поведение на филтъра.

Допълнително, при EKF възникват проблеми, свързани с **линейната апроксимация на нелинейната динамика**. Якобианите се изчисляват около текущата оценка на състоянието, което означава, че при по-големи грешки или при силно нелинейни режими апроксимацията става неточна. Това може да доведе до нестабилност, забавена конвергенция или дори до дивергенция на филтъра. Макар UKF да редуцира влиянието на грешките от линейната апроксимация чрез преобразуване по метода на сигма-точките (unscented transform), той също остава зависим от коректността на аналитичния модел и от правилния избор на параметрите на сигма-точките.

Съществен аспект при навигацията на БЛА е поведението на оценителните филтри в режим на счисление (dead reckoning), когато абсолютните измервания (напр. GNSS) временно отсъстват. В такива случаи корекцията от абсолютен източник е недостъпна и оценяването се опира предимно на процесната динамика и инерциалните измервания, поради което грешките от тип bias, нестационарни шумове и несъответствия на модела се акумулират във времето и водят до нарастваща грешка в позицията и скоростта. Обзорни изследвания показват, че при продължителни GNSS-отпадания и силно нелинейна динамика класическите филтрационни схеми могат да проявят съществена деградация на точността и/или консистентността, което мотивира използването на устойчиви и адаптивни подходи.

Всички разгледани ограничения очертават ясно, че проблемът не е в самата филтрационна рамка на Калмановия подход, а в **невъзможността на аналитичния динамичен модел да улови сложните, нелинейни и нестационарни зависимости**, характерни за реалната динамика на БЛА. Това води до логичния извод, че за повишаване на точността и устойчивостта на навигационната система е необходимо динамичният модел да бъде допълнен или заместен с по-гъвкава апроксимационна структура.

Тази необходимост формира теоретичната основа за преминаване към **хибридни модели**, в които Калмановата филтрация се запазва като оптимален механизъм за интегриране на информация и управление на несигурността, а описанието на динамиката се реализира чрез обучаеми невронни мрежи. В този контекст рекурентните архитектури от тип LSTM се явяват естествен избор, тъй като са способни да моделират последователни данни, нелинейни зависимости и времеви корелации, които остават извън обхвата на класическите модели.

1.4. Приложение на изкуствен интелект при съвременните навигационни системи.

През последното десетилетие изкуственият интелект се превръща във водещ фактор за развитието на алгоритмите за навигация на безпилотни летателни апарати. Класическите модели, основани на аналитични уравнения и предпоставки за линейност, вече не са достатъчни в условията на силно нелинейни динамики, високочестотни шумове, сензорни откази и GNSS-смущения. Системите за състояние на БЛА демонстрират значителни отклонения, когато реалната среда не

съответства на предварително зададените модели, а това е характерно за типичната оперативна среда: турбулентност, вибрации, електронно противодействие, временни заслепявания на IMU и внезапни промени в аеродинамичните натоварвания.

Натрупването на големи масиви от данни, както и възможността за синтетично генериране на тренировъчни случаи в MATLAB/Simulink, създава предпоставки за преминаване към базирани на данни (data-driven) и хибридни AI-подходи. Тези методи заемат междинна позиция между “white-box” физичните модели и “black-box” архитектурите от дълбокото обучение, като целта е да се намали моделната грешка и да се повиши прецизността на прогнозирането на динамиката върху широк диапазон от режими на полет. Така например в презентацията от MATLAB EXPO 2024–2025 (Jawahar & Ulusoy, 2024) ясно подчертава, че невронните state-space модели и редуцираните AI-модели (AI-ROM) вече са част от индустриалния стандарт за управление на сложни системи, включително БЛА.

Така възниква изследователска парадигма, в която AI не просто допълва класическия Калманов филтър, а преобразува самия модел на състоянието, като позволява адаптивна структура, по-устойчива на шум, динамична несигурност и нелинейност.

1.4.3. Хибридни AI-Kalman подходи.

Хибридните AI-Kalman подходи са една от водещите линии в съвременните навигационни алгоритми, тъй като комбинират физично-обоснованата структура на Калмановата филтрация с адаптивността на методите на дълбокото обучение. Основната идея е да се интегрират обучаеми компоненти в ключови блокове на рекурсивното оценяване - най-често в прогнозната стъпка (модела на прехода на състоянието), в модела/статистиката на шума на процеса, както и в адаптивни механизми за настройване на ковариациите \mathbf{Q} и \mathbf{R} . Така се запазва вероятностната схема „прогнозиране–корекция“, като едновременно се намалява зависимостта от строгата аналитичност на модела и от допускането за линейност (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022).

Както беше разгледано по-рано в главата, в класическия линеен KF динамичният модел е зададен съгласно:

$$x_{k+1} = F_k x_k + B_k u_k + w_k, \quad (1.33)$$

където F_k , B_k и статистиката на w_k са известни или моделирани априори. На практика при БЛА реалната динамика често е:

- силно нелинейна,
- променяща се във времето,
- зависеща от фактори, които трудно се описват физично (турбулентност, механични вибрации, температурни нелинейности),
- подложена на неидеални и в общия случай негаусови въздействия.

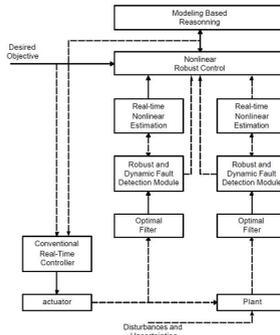
Един от широко използваните хибридни подходи е **апроксимация или замяна на преходния модел** с обучаема нелинейна функция:

$$x_{k+1} = f_{\theta}(x_k, u_k) + w_k, \quad (1.34)$$

където f_{θ} е невронен компонент (напр. рекурентна мрежа), обучен върху реални или симулирани данни. Представителен пример за този клас е KalmanNet, при който се запазва рекурсивната структура на Калмановото оценяване, но се използва невронна мрежа, за да се повиши устойчивостта към нелинейности и несъответствия (model mismatch) при частично известна динамика (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022).

Наред с това съществуват и хибридни подобрения, които **не заменят** динамичния модел с невронна мрежа, а въвеждат **адаптивност** в настройката на филтъра (Фигура 11). В публикацията *Kalman Filter Enhancement for UAV Navigation* Johnson, Sasiadek и Zalewski (Johnson, Roger; Sasiadek, Jerzy; Zalewski, Janusz, 2002) предлагат:

- fuzzy-логика за адаптивно настройване на параметрите/„noise strengths“ с цел ограничаване на дивергенцията на EKF;
- паралелизация за ускоряване на изчисленията, като демонстрират ефект чрез симулационни резултати.

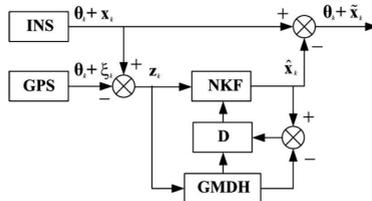


Фигура 11. Основна архитектура на системата за управление.

Източник: *Kalman Filter Enhancement for UAV Navigation* (Johnson, Roger; Sasiadek, Jerzy; Zalewski, Janusz, 2002).

Подобни идеи за базирано на данни обучавемо моделиране на нелинейна динамика се срещат в литературата за **ядрово-базирани (kernel) байесови филтри**, при които преходният и/или измервателният модел се описват в пространство на характеристики (RKHS), а рекурсивната схема „прогнозиране–корекция“ се реализира чрез kernel-операции. Представителни разработки в тази насока са *Kernel Kalman rule / Kernel Kalman filter* (Gebhardt, G.H.W., Kupcsik, A. & Neumann, G., 2019) и *Adaptive Kernel Kalman Filter* (Sun, M., Davies, M. E., Proudler, I., & Hoptgood, J. R., 2023), които формализират оценяването чрез ядрови враждания и позволяват работа при нелинейни зависимости.

В публикацията *Navigation complex with adaptive non-linear Kalman filter for unmanned flight vehicle* (Selezneva, Neusypin, & Proletarsky, 2019) е разработен **адаптивен нелинеен Калманов филтър** (Фигура 12) с параметрична идентификация на модела на грешките на INS, като авторите сравняват няколко реализации (KF, NKF, NKF+OLS, NKF+GMDH и др.). Представените резултати показват съществено повишаване на точността спрямо класически KF (напр. намаляване на грешката на местоположението от порядъка 12–15 m при KF до около 4–5 m при най-добрите варианти в техния експериментален сценарий).



Фигура 12. Блокова схема на NC с NKF при използване на GMDH.

Източник: *Navigation complex with adaptive non-linear kalman filter For unmanned flight vehicle* (Selezneva, Neusypin, & Proletarsky, 2019).

Като пример за съвременни хибридни реализации може да се посочи KalmanNet, при който невронен модул се обучава да оценява **Калмановото усиление K_k** (Kalman gain), докато рекурсивната структура на филтрирането се запазва.

От гледна точка на GNSS-устойчивостта, в INS/GNSS литературата широко се използва подходът, при който ИИ генерира **псевдо-измервания** (pseudo-measurements) по време на GNSS прекъсвания, които след това се подават към EKF като заместител/подпомагаща информация. Например Fang et al. (Fang, W., Jiang, J., Lu, S., Gong, Y., Tao, Y., Tang, Y., Yan, P., Luo, H., & Liu, J., 2020) използват LSTM за генериране на псевдо GNSS-инкременти при отпадане на GNSS сигнала, а Zhi et al. (Zhi, Z., Liu, D., & Liu, L., 2022) прилагат CNN–LSTM за компенсация на деградацията на

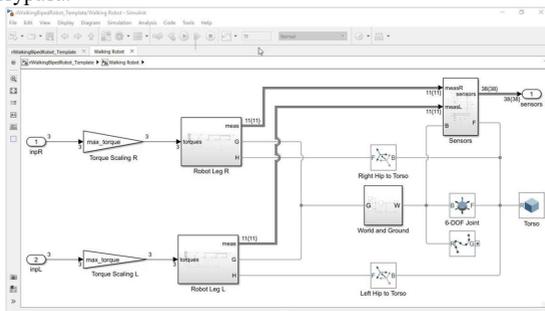
GPS/INS интеграцията по време на GPS outages. В този клас системи логиката е: **INS осигурява динамиката, ИИ оценява заместваща/компенсираща наблюдателна информация, а EKF извършва рекурсивната корекция.**

Отделна, но свързана линия са EKF-базирани архитектури за сензорно интегриране без ИИ, използвани като базова референция и отправна точка за разширения. Пример за такава практическа постановка е EKF-базирано интегриране на GNSS/IMU/барометрични данни в MATLAB/Simulink, разгледано в работа на Motayed (Motayed M. S., 2025).

Що се отнася до диагностика на аномалии и откази, методи на базата на невронни мрежи могат да се използват като **детектори на нетипични режими** в поток от измервания. Макар Djaghoul et al. (Djaghoul, Tehrani, & Vurpillot, 2025) да демонстрират подхода в контекст на силова електроника (инвертор), принципът за откриване на аномални шаблони чрез 1D CNN/LSTM е методологично преносим и към навигационни сензорни потоци.

За устойчиво **оценяване на позицията** в среда с липса на GSNN-сигнал (GNSS-denied) се предлага използване на **хибридна архитектура 1D CNN/LSTM**. 1D CNN слоя извлича високочестотни характеристики от потоците на инерциалните сензори, докато последващият LSTM слой моделира тяхната дългосрочна динамика (Woyano, F., Park, S., Jordanov, V. B., & Lee, S., 2023). Този подход подобрява точността при внезапна загуба на GNSS сигнал.

Накрая, при задачи по управление (Фигура 13), обучение с утвърждаване (RL) често се комбинира с блок за оценяване на състоянието в управленския контур; в презентацията на MathWorks (Baru, MathWorks) това е показано концептуално чрез наличието на отделен блок „State Estimation“ в архитектурата.



Фигура 13. Simulink модел за RL-базирано управление на робот.

Източник: *Reinforcement Learning: Leveraging Deep Learning for Controls* (Baru, MathWorks).

Всички разгледани хибридни модели имат значителни предимства пред традиционните Калманови модели, но за по-голяма яснота в е направен анализ на предимствата на хибридните AI-Kalman подходи

Таблица 6. Синтезирано предимство на хибридните AI-Kalman подходи.

Функционален елемент	Класически KF/EKF/UKF	Hybrid AI-Kalman (AI-KF)
Прогнозиране (prediction)	Динамичен модел $f(\cdot)$ е аналитично зададен (линеен при KF; нелинеен с линеинизация при EKF; сигма-точки при UKF). Точността зависи силно от адекватността на модела	Преходният модел може да се апроксимира/коригира чрез ANN $f0(\cdot)$ или да се използва обучаем корекционен член (residual). Намалвява чувствителността към model mismatch
Шумови модели Q, R	Обикновено се приемат константни или ръчно настроени; възможна е класическа адаптация (innovation-based), но с ограничена гъвкавост	Q и/или R могат да се оценяват адаптивно от данни (напр. ANN/регресионни модели), включително зависимост от режим, динамика и качество на измерванията
Работа при нелинейности	EKF/UKF позволяват нелинейни модели, но апостериорът обикновено е гаусова апроксимация ; устойчивостта зависи от	Обучаемите компоненти могат да моделират посложни нелинейни зависимости и да подобрят устойчивостта при model mismatch; често се запазва гаусовата апроксимация, но с по-добре адаптирани параметри/модели

	линейнизацията/разпределенията и от качеството на модела	
<i>Аномалии / outliers</i>	Чувствителност към outliers при стандартни допускания; изисква допълнителни устойчиви механизми (gating, M-estimators, robust loss)	Възможно е добавяне на AI-детектори за аномалии и/или обучаемо gating/претегляне на измерванията, което подпомага устойчивото отхвърляне или преценка на доверие
<i>Компенсация на дрейф и систематични грешки</i>	Компенсацията зависи от включените вектора на състоянието bias/scale и от коректността на моделите; ограничена при немоделирани ефекти	ANN/времеви модели (напр. LSTM) могат да описват времево-корелирани дрейфове и немоделирани зависимости (температура, вибрации), като добавят корекционни членове или адаптация на шумовете
<i>Поведение при GNSS прекъсвания</i>	Устойчивостта се определя от INS грешките и настройката на Q; при по-дълги прекъсвания грешката нараства поради интеграционен дрейф	Често се използват псевдо-измервания от AI, допълнителни модалности (VIO/SLAM) или адаптивни модели за дрейф, което може да ограничи нарастването на грешката при прекъсвания
<i>Изчислителна сложност и внедряване</i>	Относително предсказуема сложност; добре установени реализации и инструменти (напр. MATLAB/Simulink)	Допълнителна сложност от обучаемите блокове (обучение/инференс, данни, валидиране); при добре избрана архитектура остава съвместимо с реално време, но изисква повече инженеринг
<i>Интерпретируемост и верификация</i>	Висока интерпретируемост на модела и параметрите; по-лесна формална верификация и диагностика	Рекурсивната структура остава интерпретируема, но обучаемите компоненти са частично „черна кутия“; изискват допълнителни процедури за валидация, устойчивост и безопасност Kalman

В обобщение (Таблица 6), разгледаните разработки показват, че хибридният ИИ–Калман подход са **една от най-перспективните** линии за навигационни системи на БЛА: те запазват рекурсивната вероятностна структура на филтрирането, като използват ИИ за моделиране/адаптация на нелинейности, шумови структури и дрейфове, които трудно се описват аналитично.

1.5. ИЗВОДИ ОТ ГЛАВА 1.

1. Анализът в тази глава показва, че нито INS, нито GNSS, нито визуалните методи поотделно осигуряват устойчиво решение във всички режими на полет, поради което интегрираните архитектури са методологично обоснован избор; в частност при инерциално счисление (dead reckoning) систематичните грешки (bias) на IMU, температурната нестабилност и стохастичните отклонения в измерванията определят темпа на натрупване на навигационната грешка и съответно точността на оценките за позиция и скорост, особено при продължителни интервали без външни корекции (напр. GNSS или други абсолютни измервания).
2. Инерциалните навигационни системи осигуряват висока автономност, но поради интегрирането на инерциалните измервания неизбежно акумулират дрейф, като темпът на натрупване зависи от класа, калибрацията и температурната стабилност на инерциалния измервателен блок (IMU), включително при MEMS-IMU.
3. GNSS навигацията осигурява висока абсолютна точност при нормални условия на прием, но е уязвима при заглушаване (jamming), подправяне (spoofing) и деградации (вкл. многолъчево разпространение на сигнала - multipath), като тези смущения са реалистичен оперативен фактор в среди с радиоелектронно противодействие (радиоелектронна борба - РЕБ) и следва да се отчетат при проектиране на навигационни архитектури.
4. Визуалните навигационни методи разширяват възможностите за автономна локализация и картографиране, а интеграцията им с INS чрез EKF/UKF повишава устойчивостта спрямо единични източници, но остава чувствителна към условията на наблюдение (осветеност, текстура, динамика на движението) и не осигурява елиминиране на дрейфа без абсолютни корекции или допълнителни глобални ограничения (напр. GNSS, маяци, карти/loop closure), особено при частична наблюдаемост.
5. EKF разчита на линеаризация на нелинейните модели и при силна нелинейност или неточно моделиране може да доведе до субоптимални оценки или дивергенция. UKF намалява риска чрез сигма-точки; точността и консистентността му зависят от коректната настройка на шумовите ковариации и параметризацията на състоянието, особено при нестационарни шумове без адаптивни механизми.

6. Калмановите филтри (вкл. EKF/UKF) са ефективни при добре дефинирани модели и контролирани предпоставки, но могат да деградират в реална експлоатация при деградирани измервания, нестационарни шумове, аномални измервания (outliers) и динамични промени в достъпността на наблюденията, ако не се използват робастни и/или адаптивни механизми.
7. Реалните сценарии за БЛА се характеризират със силна нелинейност, нестационарни шумове и интермитентна (прекъсваема) достъпност на GNSS измерванията, зависеща от средата и сценария
8. При системно нарушаване на линейно-гаусовите и стационарни допускания моделите и филтрите от Калманов тип могат да деградират; поради това са необходими робастни и/или адаптивни механизми
9. Методите от изкуствения интелект предоставят възможности за адаптация, моделиране на нелинейни зависимости и извличане на закономерности от сензорни данни, включително при трудно моделирани режими, но изискват прецизно обучение, валидиране и контрол на обобщаването (generalization).
10. Необходимостта от интегрирани навигационни решения обосновава използването на робастни и адаптивни архитектури, включително подходи, комбиниращи Калманови филтри с методи от ИИ, например за детекция на аномалии, адаптация на шумови модели или подпомагащи модели за оценяване.

ГЛАВА 2.

АЛГОРИТМИ ЗА ИЗКУСТВЕН ИНТЕЛЕКТ ПРИ УПРАВЛЕНИЕ НА БЕЗПИЛОТНИ ЛЕТАТЕЛНИ АПАРАТИ.

2.1. Моделиране на динамични системи с невронни мрежи.

В съвременните системи за автономно управление на БЛА нараства дялът на базираните на данни методи, при които динамиката се описва чрез обучаеми модели вместо (или заедно със) строго аналитични уравнения. Реалното поведение на летателната платформа често е нелинейно, контекстно зависимо и времевариантно, а неопределеностите, породени от аеродинамични ефекти, вибрации и промени в работната среда, не се улавят надеждно от опростени физични модели. В този смисъл дълбокото обучение не „отменя“ класическата теория на управлението, а предоставя механизъм за идентификация и адаптация на компонентите, които са трудни за моделиране априори (Cohen & Klein, 2024).

Една от изследователските линии са дълбоките state-space модели (deep state-space models - deep SSM), при които преходът на състоянието се задава чрез невронна мрежа (или чрез невронни подмодели), а обучението се извършва върху времеви редове вход-изход. Този клас модели може да се разглежда като обобщение на класическото описание на пространството на състоянието (state-space), но с по-висока изразителност при нелинейни системи и възможност за вероятностно (стохастично) моделиране на неопределеността. Gedon и съавт. (Gedon, D., Wahlström, N., Schön, T. V., & Ljung, L., 2021) демонстрират, че deep SSM са приложими като „черна кутия“ за нелинейна системна идентификация и оценяват няколко разновидности върху референтни тестови набори (benchmark datasets), като изрично подчертават ролята на невронните параметризации за представяне на широк клас динамики.

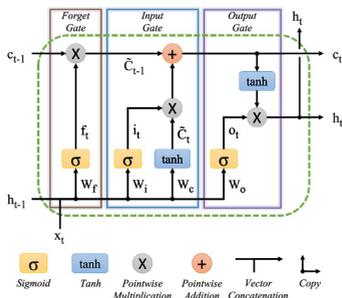
Втора изследователска линия, която разширява възможностите за моделиране на динамиката с невронни мрежи в контекста на управлението, е използването на латентни динамични модели за планиране и прогнозиране при частична наблюдаемост. Подходът на Hafner и съавт. (Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., & Davidson, J., 2019) използва рекурентен модел на състоянието в латентно пространство и обучава преходен модел, който позволява многостъпкови прогнози и планиране на действия без нужда от експлицитно зададена физична динамика. Това е особено релевантно за БЛА, защото навигационните и управляващите задачи често са частично наблюдаеми (ограничена видимост, временни отпадания на GNSS и др.), а латентното състояние може да играе ролята на компактно обобщение на историята от наблюдения.

Трета важна линия са Коорпан-подходите, при които чрез невронна мрежа се намира преобразуване на състоянието към пространство с по-висока размерност, така че нелинейната динамика да се апроксимира с линейна динамика в това пространство, което улеснява синтеза на управление. Han и съвт. (Han, Y., Nao, W., & Vaidya, U., 2020) разглеждат DNN-базирано научаване на Коорпан-представяне с цел синтез на управление и подчертават предимството да се комбинират базирана на данни идентификация и класически методи за управление (напр. linear quadratic regulator - LQR/model predictive control - MPC) върху получената линейна апроксимация. Практическата стойност за БЛА е, че подобен модел може да улесни проектирането на регулатори при силни нелинейности и несигурни параметри, без да се изисква пълно физично моделиране на всички ефекти.

В навигационните приложения дълбокото обучение най-често се използва като допълнение към класически структури (напр. филтриране и сензорно интегриране), като подпомага калибрация, потискане на шум (denoising) и частично научаване на параметри или компоненти на модела, вместо те да се задават изцяло аналитично. В този аспект обзорът на Cohen и Klein (Cohen & Klein, 2024) систематизира тенденциите при интеграция на дълбоко обучение с инерциална навигация и интегриране на сензорни данни, включително подходи, при които се научават компоненти на моделно-базирани алгоритми, вместо те да се дефинират изцяло аналитично.

2.1.1. RNN и LSTM.

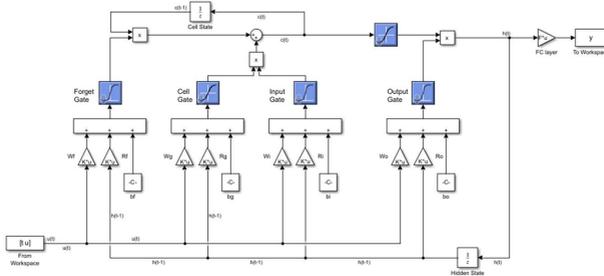
Рекурентните невронни мрежи (RNN) са подходящи за моделиране на динамични системи, тъй като използват вътрешно състояние, чрез което се описват зависимостите във времето между последователни входове. В контекста на БЛА това е важно, тъй като инерциалните измервания, визуалните наблюдения и остатъците от навигационното оценяване (navigation residuals) обичайно се интерпретират като времева последователност, която носи информация за текущото движение и натрупващите се грешки.



Фигура 16. LSTM-клетката може да обработва данни последователно и да запазва скритото си състояние във времето.

Източник: ResearchGate.

Класическите RNN често работят добре при краткосрочни зависимости, но при дълги последователности могат да проявят проблеми от типа затихване/експлозия на градиента, което затруднява обучението. Поради това архитектурата LSTM (Long Short-Term Memory) се използва широко като практическо решение за моделиране на дълги времеви зависимости (Фигура 16), типични за IMU сигнали и визуално-инерциални редове при полет (Hochreiter, S., & Schmidhuber, J., 1997).



Фигура 17. Simulink имплементация на LSTM слой, последван от напълно свързан слой (fully connected layer - FC).

Източник: MathWorks.

В анализа на MathLab *LSTM Layer as a Nonlinear Dynamical System* (Сампа, 2023) се представя как един слой LSTM може да се разглежда като нелинейна динамична система (Фигура 17) със специфична структура и дали и как LSTM слоевете могат да апроксимират линейни динамични системи:

$$\begin{aligned}
 f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f), \\
 i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), \\
 o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o), \\
 \tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c), \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\
 h_t &= o_t \odot \tanh(c_t),
 \end{aligned}
 \tag{2.1}$$

където:

Входове/състояния:

- x_t - входен вектор в момент t (напр. IMU измервания, визуални признаци, иновации/остатъци, или конкатенация от такива величини);
- h_{t-1} - скрито (изходно) състояние на LSTM от предходния момент $t-1$;
- h_t - скрито (изходно) състояние на LSTM в момент t ; това е „изходът“ на клетката, който често се подава към следващ слой или към регресор за прогнозиране;
- c_{t-1} - клетъчно (вътрешно) състояние от момент $t-1$, което носи „паметта“ на LSTM;
- c_t - клетъчно (вътрешно) състояние в момент t , обновено на база текущия вход и предходните състояния.

Порти (gates) и кандидат за клетъчно състояние (candidate cell state):

- f_t - *forget gate* (порта за забравяне): определя каква част от c_{t-1} да се запази;
- i_t - *input gate* (входна порта): определя колко от новата информация да се запише в клетката;
- o_t - *output gate* (изходна врата): определя каква част от актуалното клетъчно състояние да се „покаже“ като изход h_t ;
- \tilde{c}_t - кандидат за ново клетъчно съдържание представлява нелинейно преобразувана „нова информация“, която може да бъде добавена в паметта.

Нелинейности и операции:

- $\sigma(\cdot)$ - логистична сигмоидна функция: ограничава стойностите на вратите в интервала $[0,1]$;
- $\tanh(\cdot)$ - хиперболичен тангенс: ограничава стойностите в $[-1,1]$ и стабилизира обновяването на състоянията;

- \odot - поелементно умножение (Hadamard product).

Тази структура е подходяща за моделиране на динамика на БЛА, тъй като вътрешното състояние c_t може да кодира дългосрочни зависимости и натрупващи се ефекти (например бавни промени, свързани с дрейф/систематични компоненти), докато изходното състояние h_t е векторно представяне на текущата динамика, което се подава към следващия времеви момент или към изходен слой.

В задачите за оценяване на състоянието невронни модели от типа RNN/LSTM могат да се използват като модул, който реконструира/апроксимира динамиката на движение от историческа информация и така подпомага или допълва класически оценители. В *Deep Learning-Based Neural Network Training for State Estimation Enhancement: Application to Attitude Estimation* (Al-Sharman, и др., 2019) се демонстрира, че подобни модели могат да подобряват оценяването на ориентация при сложни и нестационарни измервателни условия, което е релевантно и за БЛА.

Допълнително, устойчивостта към аномални измервания е критична при сензори като GNSS/IMU и при визуални артефакти. В *Outlier-Resistant Remote State Estimation for RNNs with Mixed Time-Delays* (Li, Wang, Dong, & Ghinea, 2020) се разглеждат рекурентни модели и процедури, насочени към устойчиво оценяване при наличие на аномалии, като идеята е приложима към навигационни редове (например при деградация на GNSS, насищане на IMU или проблеми във визуални наблюдения).

Отделно, LSTM може да се използва и за прогнозиране (а не само за филтриране), като прогнозира бъдещи състояния/пози (или други кинематични величини) на база последователност от наблюдения. Пример за такава постановка е работата на Torkar (Torkar, 2019), която демонстрира времево моделиране с LSTM за прогнозиране в контекст на последователни наблюдения; концепцията е методологично преносима към БЛА при наличие на подходящи данни и коректно дефинирано състояние.

Обобщението на направените проучвания е, че RNN и LSTM архитектурите са подходящ инструмент за моделиране на динамиката на БЛА при наличие на времеви зависимости и нестационарни условия. Те най-често се използват не като пълна замяна на физичните модели, а **като допълващ модул - за апроксимация на процесния модел**, компенсирани на систематични ефекти, или повишаване на устойчивостта при деградирани измервания, в рамката на хибридни навигационни/оценителни архитектури.

2.2. ANN като заместващ динамичен модел на KF.

Калмановата филтрация (KF, EKF, UKF) изисква процесен (преходен) модел, който описва еволюцията на състоянието във времето и определя качеството на прогнозната стъпка. При моделна непълнота, силна нелинейност или несъответствие между аналитичния модел и реалната динамика (model mismatch) прогнозната грешка може да нараства, а в условия на ограничена наблюдаемост или временно отсъствие на абсолютна корекция (напр. при счисление) филтърът може да прояви съществена деградация на точността и/или неконсистентност. Това мотивира използването на обучаеми компоненти, при които процесният модел не се задава аналитично, а се параметризира чрез невронна мрежа и се обучава от данни, например: $x_{k+1} = f(x_k, u_k) + w_k$. В литературата са предложени редица хибридни постановки, в които невронни модели подпомагат или частично заместват динамиката/параметрите на филтъра и в експериментални условия показват подобрения при нелинейности и model mismatch. Резултатите от публикациите, отнасящи се до хибридни модели между класическите Калманови филтри и изкуствени невронни мрежи постигат отлични резултати по отношение на точност, както и на ефективност (Feng, S., Li, X., Zhang, S., Jian, Z., Duan, H., & Wang, Z., 2023). Използваните в изследванията относително компактни RNN, могат да бъдат обучени с относително малък набор от данни, което ги прави приложими за високоразмерни state-space модели и устройства с ограничени изчислителни възможности (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022). Експерименталните резултати показват перспективите на хибридният подход по отношение повишаването на точност, устойчивост и интерпретируемост (Chen, C., Lu, X., Wang, B., Trigoni, N., & Markham, A., 2021).

2.2.2. ANN като прогнозен модел.

Използването на изкуствени невронни мрежи (ANN) като прогнозни модели в рамките на навигационни алгоритми е направление с нарастващ интерес в литературата за автономни системи и интегриране на сензорни данни. За разлика от подходите, при които ANN замества процесния модел (пеходната функция) във филтъра, тук невронната мрежа изпълнява ролята на допълнителен прогностичен/корективен модул, който работи паралелно със статистическата рамка на Калмановата филтрация. В този смисъл ANN може да подпомага прогнозната стъпка чрез допълнителна информация, извлечена от исторически измервания, контекст и/или латентни зависимости, без задължително да се нарушава predict-update структурата на филтъра.

Практическата мотивация за прогностични ANN слоеве е свързана с ограничената приложимост на фиксирани аналитични модели в условия на силни нелинейности, нестационарни шумове и несъответствие между модела и реалната динамика (model mismatch). В такива режими прогнозната грешка може да придобие систематична компонента, която не се компенсира ефективно единствено чрез класическата настройка на ковариациите. Особено критични са режимите на числене (dead reckoning) при временна липса на абсолютна корекция (напр. GNSS отпадане), когато натрупването на грешки се ускорява и зависимостта от прогнозата нараства (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Dunik, J., Straka, O., Closas, P., & Eldar, Y. C., 2025); (Kim, S., Petrunin, I., & Shin, H.-S., 2022).

Една от ключовите идеи в този клас подходи е ANN да прогнозира не само самото състояние, а и грешката на модела/оценката (model-error / bias / residual prediction). Вместо невронната мрежа да замества изцяло динамиката, тя може да се обучи да оценява корекционен член Δx_k или еквивалентна поправка в прогнозата, която компенсира систематични несъответствия на аналитичния процесен модел (Farchi, A., Vocquet, M., & Brajard, J., 2021); (Brajard, J., Carrassi, A., Vocquet, M., & Bertino, L., 2020). Аналогично, ANN може да се използва за оценка/адаптация на шумови характеристики (напр. ефективни ковариации), когато предпоставките за стационарност на шумовете са нарушени (Levy, A., & Klein, I., 2025); (Sun, B., Zhang, Z., Qiao, D., Mu, X., & Hu, X., 2022); (Wang, H., Deng, Z., Feng, B., Ma, H., & Xia, Y., 2017). Този тип постановки се разглеждат като част от по-общия клас AI-aided Kalman filtering методи, при които се запазва филтрационната структура, но се добавя обучаем механизъм за компенсация или адаптация.

В контекста на интегрирана навигационна система ANN може да се използва по няколко типични начина, които се разглеждат в литературата за хибридни филтрационни архитектури (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Dunik, J., Straka, O., Closas, P., & Eldar, Y. C., 2025); (Kim, S., Petrunin, I., & Shin, H.-S., 2022):

- **Пряк прогнозен модел на състоянието**, който предоставя допълнителна прогноза \hat{x}_{k+1} (или корекционен член към нея) и се комбинира с филтъра (Farchi, A., Vocquet, M., & Brajard, J., 2021); (Brajard, J., Carrassi, A., Vocquet, M., & Bertino, L., 2020);
- **Прогнозен модел на наблюденията**, например прогнозиране на бъдещи IMU/инкрементални измервания или на иновациите (innovations), с цел стабилизиране на оценяването при деградирани данни (Sun, B., Zhang, Z., Qiao, D., Mu, X., & Hu, X., 2022);
- **Прогнозен модел на грешката/несъответствието**, при който ANN оценява bias или структурна грешка на процесния модел и я използва за корекция на прогнозната стъпка или за адаптация на шумовите параметри (Levy, A., & Klein, I., 2025); (Wang, H., Deng, Z., Feng, B., Ma, H., & Xia, Y., 2017).

Всеки от тези подходи може да бъде интегриран в хибридна архитектура при условие на ясна постановка за обучение, валидация и контрол на обобщаването, тъй като прогностичните ANN компоненти могат да подобрят точността в разгледани сценарии, но могат и да внесат систематични отклонения при домейн-разлика или при непредставителни обучителни данни. Поради това в инженерна реализация ANN се разглежда като допълнителен модул, който намалява зависимостта от опростени аналитични модели, при запазване на структурирана оценителна схема и ясни критерии за устойчивост (Kim, S., Petrunin, I., & Shin, H.-S., 2022); (Shlezinger, N., Revach, G., Ghosh,

A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025); (Choi, G., Park, J., Shlezinger, N., Eldar, Y. C., & Lee, N., 2023).

Така ANN в ролята на прогнозен модел действа като механизъм, който разширява прогнозната способност на Калмановата филтрация отвъд ограниченията на фиксираните аналитични модели, като предоставя допълнителна прогноза, обучена върху реални или симулирани данни (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025). В резултат хибридната система може да прояви висока устойчивост в сценарии с висока неопределеност, нелинейност и влошаване на качеството на сензорните данни — ситуации, типични за реалните условия на работа на (Sun, B., Zhang, Z., Qiao, D., Mu, X., & Hu, X., 2022); (Levy, A., & Klein, I., 2025).

За да бъде ролята на ANN като прогнозен модел формално и ясно дефинирана в рамката на Калмановата филтрация, най-често се използват постановки от следните два типа.

(А) Корекция на прогнозната стъпка чрез прогнозиране на моделна грешка

Нека класическият процесен модел (аналитичен или приближен) е:

$$x_{k+1} = f_{\theta}(x_k, u_k) + w_k, \quad w_k \sim N(0, Q_k). \quad (2.12.1)$$

В прогнозната (aiding) постановка ANN се обучава да прогнозира **корекционен член** $\Delta_{\theta}(\cdot)$, който компенсира структурна грешка/непълнота на модела:

$$\hat{x}_{k+1|k} = f_{\theta}(\hat{x}_{k|k}, u_k) + \Delta_{\theta}(\xi_k), \quad (2.12.2)$$

където ξ_k е вектор от входни признаци (напр. прозорец от IMU измервания, предишни иновации, оценки на отклонението (bias), контекстни параметри). Тази постановка е особено полезна в режим на счисление (dead reckoning), когато липсва абсолютна корекция (напр. при GNSS отпадане) и систематичната компонента на прогнозната грешка доминира акумулирането на дрейф.

Така ANN не замества изцяло процесния модел, а изпълнява ролята на **коректив към прогноза**, който намалява систематичните отклонения при непълни/неточни аналитични зависимости. Подобни идеи за **online model error correction** и комбиниране на ML с асимилация/оценяване се разглеждат и валидират експериментално в литературата (Farchi, A., Vocquet, M., & Brajard, J., 2021); (Brajard, J., Carrassi, A., Vocquet, M., & Bertino, L., 2020). В обзорни постановки за **AI-aided Kalman filtering** този подход се разглежда като част от клас методи, при които обучаем модул компенсира несъответствия на модела при запазване на филтрационната структура (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025).

(В) Адаптация на шумовите ковариации чрез ANN (нестационарни шумове)

В класическата Калманова филтрация ковариациите на процесния и измервателния шум Q_k и R_k често се приемат стационарни или се задават чрез предварителна настройка. При реални навигационни системи за БЛА обаче шумовите характеристики могат да бъдат нестационарни (напр. промяна на вибрационен режим, турбулентност, деградация на GNSS, промяна в качеството на измерванията), което мотивира адаптивна постановка.

Една стандартна линия е онлайн адаптация на Q_k и/или R_k чрез статистика на остатъците/иновациите (innovation-based adaptive filtering), където ковариациите се коригират така, че статистическите свойства на иновационната последователност да съответстват на предпоставките на филтъра. Този подход и негови варианти са широко разглеждани в адаптивни KF схеми и в контекста на интегрирани INS/GNSS системи (Sun, B., Zhang, Z., Qiao, D., Mu, X., & Hu, X., 2022); (Wang, H., Deng, Z., Feng, B., Ma, H., & Xia, Y., 2017).

В ANN-базирана постановка адаптацията може да се параметризира чрез обучаеми функции, които оценяват ефективните ковариации като функции на признаци:

$$Q_k = \mathcal{Q}_{\theta}(\xi_k), \quad R_k = \mathcal{R}_{\phi}(\zeta_k), \quad (2.12.3)$$

където:

Q_k - ковариационна матрица на **процесния шум** (process noise) в дискретния момент k .

R_k - ковариационна матрица на **измервателния шум** (measurement noise) в момента k .

k - дискретен времеви индекс (итерация/стъпка на филтъра).

\mathcal{Q}_{θ} и \mathcal{R}_{ϕ} - **обучаеми функции** (например ANN), които връщат Q_k и R_k .

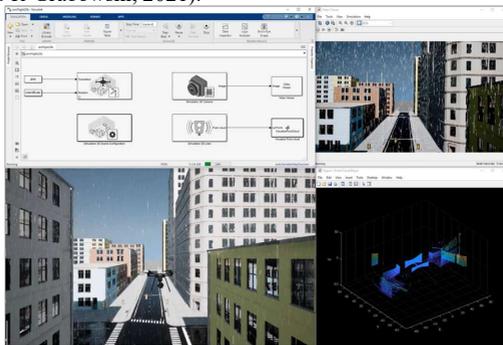
θ и ϕ - параметри (тегла) на съответните невронни модели.

ξ_k и ζ_k - входни признаци/вектор от характеристики (features) за момента k , от които ANN прогнозира Q_k и R_k , като ζ_k може да включва индикатори за качество на наблюденията (напр. C/N₀, геометрия, детектирани смущения, статистика на иновациите). Това позволява филтърът да „пренастройва“ тежестта между прогнозата и измерванията в реално време при деградирани или променливи условия. Подобни **невронно-адаптивни** постановки (напр. neural/adaptive UKF) се разглеждат като средство за адаптация на филтъра при нестационарни шумове. (Levy, A., & Klein, I., 2025) Общият клас методи за AI-aided Kalman filtering включва и постановки за учене/адаптация на шумови параметри и статистики, като целта е устойчивост при реалистични условия и model mismatch (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025).

2.2.3. Обучение върху симулирани или реални данни.

Един от ключовите практически въпроси при изграждането на ANN като компонент на хибридна навигационна система е изборът на стратегия за обучение: използване на симулирани данни, използване на реални измервания, или комбиниран подход. Двата източника на данни не са взаимно изключващи се, но се различават съществено по отношение на контролируемостта на условията, статистическата реалистичност и риска от разминаване между обучението и експлоатационната среда.

Симулационните данни предоставят важно предимство, тъй като позволяват систематично управление на полетните режими, динамичните въздействия и параметрите на сензорните модели, включително настройване на шумовите характеристики и провеждане на повторяеми експерименти в контролирана среда. Това прави симулацията особено подходяща за **първоначално разработване и валидиране** на алгоритми, както и за генериране на големи обеми данни при сценарии, които са трудни, опасни или ресурсно скъпи за реализиране в реални условия. В този контекст MATLAB/Simulink (Фигура 21) се използва като инженерна платформа за моделиране, симулация и оценяване на автономни БЛА системи и за виртуално тестване на решения за автономност и сензорни модели (Antoniou & Grabowski, 2021).



Фигура 21. Средства и инструменти в MATLAB и Simulink за разработка на автономно управлявани дронове.

Източник: MatlabEXPO Autonomous UAV Development and Evaluation with MATLAB and Simulink (Antoniou & Grabowski, 2021).

Наред с това, симулационните данни неизбежно носят риск от идеализиран характер на тези данни. Дори при внимателно конструирани сензорни модели симулацията често не възпроизвежда в достатъчна степен комплексни ефекти като дългосрочни дрейфове, температурни зависимости, структурни вибрации и други нестационарни компоненти на грешката. Това води до разминаване между симулация и реалност (sim-to-real gap), при което даден алгоритъм или обучен модел показва различно поведение при работа със симулирани спрямо реални измервания. Конкретно за GNSS и IMU сензори е предложен количествен подход за оценяване на това разминаване чрез сравнение на поведението на state-estimator върху реални и симулирани експерименти при различни шумови модели, като се изолира приносът на симулационната неточност на сензорите (Mahajan, I.,

Unjhawala, H., Zhang, H., Zhou, Z., Young, A., Ruiz, A., Caldararu, S., Batagoda, N., Ashokkumar, S., & Negrut, D., 2024).

Реалните измервания предоставят „статистически реализъм“, тъй като съдържат действителните свойства на шумовите процеси и артефактите на конкретния хардуер. Изследвания върху MEMS IMU показват, че грешките на инерциалните сензори представляват композиция от различни стохастични компоненти (напр. random walk, bias instability и др.), които се оценяват и характеризират чрез специализирани методи като Allan variance, PSD и модели от тип Gauss-Markov/AR. Поради това ANN, обучени върху реални IMU времеви редове, могат да „видят“ зависимости и структури, които не са представени в опростени симулационни сценарии (Zhao, Noremuz, & Sjöberg, 2011).

В практиката това води до компромис: реалните данни са по-репрезентативни, но са по-трудни за събиране в достатъчен обем и често са с непълно покритие на режимите на движение (особено за редки или екстремни сценарии). Поради това обзорни изследвания в областта на дълбокото обучение за инерциална навигация и sensor fusion подчертават ролята на подходи, които „подсилват“ модел-базираните схеми чрез обучение върху данни и обсъждат необходимостта от адекватна подготовка на обучителните набори при реални условия и различни домейни на движение (Cohen & Klein, 2024).

Комбинираният подход обикновено се реализира като **двустепенно обучение**: (i) предварително обучение (pretraining) върху симулирани данни, с цел усвояване на глобалните зависимости в динамиката и основните причинно-следствени връзки между състояние, управление и наблюдения, и (ii) последващо адаптиране (fine-tuning) върху реални измервания, с цел калибриране на модела към реалните шумови характеристики, структурни артефакти и неизбежни несъответствия на симулационния модел. Такава стратегия е мотивирана от наличието на измерим sim-to-real gap за GNSS/IMU и от факта, че точността на сензорната симулация има директно отражение върху качеството на оценяване на състоянието (Mahajan, I., Unjhawala, H., Zhang, H., Zhou, Z., Young, A., Ruiz, A., Caldararu, S., Batagoda, N., Ashokkumar, S., & Negrut, D., 2024).

В обобщение, обучението на ANN за навигационни цели следва да се планира така, че да се използват предимствата както на симулационните, така и на реалните данни. Симулацията е ефективна за начално обучение и покриване на контролирани сценарии, докато реалните измервания са критични за статистическа достоверност и за адаптация към реалните шумови процеси. Следователно за ANN компоненти в хибридна INS/KF навигационна схема най-надеждна е комбинирана стратегия, при която предварителното обучение върху симулация се допълва от фино настройване върху реални полетни данни (Mahajan, I., Unjhawala, H., Zhang, H., Zhou, Z., Young, A., Ruiz, A., Caldararu, S., Batagoda, N., Ashokkumar, S., & Negrut, D., 2024); (Cohen & Klein, 2024).

2.2.4. Грешка и устойчивост.

В хибридните архитектури, комбиниращи невронни мрежи с Калманова филтрация, анализът на грешката и устойчивостта придобива централно значение поради въвеждането на **допълнителен механизъм на неопределеност**: поведението на обучаемия компонент извън обучаващото разпределение (domain shift) и потенциални грешки при генерализация. Същевременно именно обучаемите модули се използват, за да компенсират недостатъци на аналитичните модели при **нелинейна динамика, моделно несъответствие и нестационарни шумове**, като целта е да се запази структурната интерпретируемост и рекурсивната ефективност на KF/EKF/UKF, но да се повиши устойчивостта при реални условия. Това направление е систематизирано в обзорни трудове за AI-подпомогнати Калманови филтри, които разграничават подходи за (i) научаване/корекция на динамиката и/или измервателния модел и (ii) научаване/адаптация на статистиката на шумовете и неопределеността (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Dunik, J., Straka, O., Closas, P., & Eldar, Y. C., 2025).

2.3. Хибридни модели (KF + ANN).

Изследванията върху комбинирането на Калманова филтрация и изкуствени невронни мрежи (ANN) се развиват интензивно през последното десетилетие, като основната цел е да се съчетаят рекурсивната структура и статистическата обосновааност на KF/EKF/UKF с

апроксимационните възможности на ANN при нелинейна динамика, моделно несъответствие и нестационарни шумове. В обзорната литература това направление се разглежда като ИИ подпомагана Калманова филтрация (AI-aided Kalman filtering), обхващащо методи за (i) научаване/корекция на процесния и измервателния модел и (ii) адаптиране на статистическите параметри (напр. шумови ковариации) с цел по-устойчиво оценяване на състоянието.

Класическият (линейно-гаусов) Калманов филтър е оптимален при линейни модели и гаусови шумови процеси; при навигационни задачи с изразена нелинейност, частична наблюдаемост, деградирани измервания и/или нестационарни грешки се налагат обобщени варианти (EKF/ESKF/UKF) и/или допълнителни устойчиви и адаптивни механизми, както и при известни шумови ковариации и стандартните допускания (напр. бял шум). В този контекст ANN се използват не като „заместител на филтрацията“, а като обучаеми компоненти, които подпомагат оценителя в части, където аналитичният модел е непълен или трудно параметризируем. Такива решения се обсъждат и в обзорите за дълбоко обучение в инерциалната навигация и сензорното интегриране, където се подчертават едновременно потенциалните предимства и необходимостта от внимателна валидация на генерализацията и консистентността.

В архитектурен план хибридните модели KF+ANN могат да се разглеждат като запазване на прогнозиране–корекция (predict–update) логиката на филтъра с въвеждане на обучаем модул в една или повече ключови стъпки. Най-често ANN участва чрез:

- **обучаемо изчисляване на Калмановия коефициент на усилване** (напр. $K_k = g_\phi(\cdot)$);
- **резидуална (добавъчна) корекция** на процесния/измервателния модел (напр. $x_{k+1} = Fx_k + Bu_k + \hat{f}_\theta(x_k, u_k) + w_k$ или $y_k = Hx_k + \hat{g}_\phi(x_k) + v_k$);
- **адаптивно претегляне/гейтинг** при интегриране на оценки от различни канали (напр. $\hat{x}_k^{fusion} = \alpha_k \hat{x}_k^{phys} + (1 - \alpha_k) \hat{x}_k^{NN}$).

Представителен пример за невронно подпомогнат оценител с Калманова структура е KalmanNet, който интегрира обучаем компонент в рекурсивната схема прогнозиране–корекция (predict–update) с цел оценяване на състоянието (state estimation) при нелинейни системи и частично известен модел.

В рамките на настоящия подраздел хибридните решения KF+ANN се структурират в три основни архитектурни линии според ролята на обучаемия компонент:

1. **Recursive KalmanNet (R–KalmanNet)** – рекурсивна невронна архитектура за оценяване, ориентирана към обучаване gain и квантифициране на неопределеността;
2. **Двуканални (dual-branch) системи** – паралелен физически и невронен клон с адаптивно интегриране на оценките;
3. **KF за линейната компонента + ANN за нелинейната/немоделирана част** – резидуален модел, който допълва аналитичната динамика/измерване.

Важно е да се подчертае, че хибридните методи **целят** повишаване на точността и устойчивостта при model mismatch и променливи шумови характеристики, но консистентността на неопределеността не следва да се приема априори; тя се проверява чрез анализ на иновационните последователности и стандартни статистически критерии (напр. NIS/NEES), особено при режими с деградирани измервания и dead-reckoning.

2.3.3. KF за линейност + ANN за нелинейности.

Този хибриден подход се оформя като една от най-устойчивите и практически приложими архитектури в съвременните навигационни системи за БЛА. Идеята му е да раздели задачата на две концептуално различни нива: **линейната динамика се обработва от класическия Калманов филтър, а нелинейните компоненти и трудните за аналитично моделиране зависимости се поемат от невронна мрежа.** Този принципен компромис между аналитичната строгост и обучаемата гъвкавост позволява да се избегнат линеаризационните грешки на EKF (Якобиани) и/или тежката настройка/чувствителност на sigma-point подходите (UKF) при определени режими/висока размерност/нестационарни шумове.

В литературата този модел се среща в няколко разновидности. Част от публикациите разглеждат невронната мрежа като коригиращ елемент, който компенсира систематични грешки,

дрейф или неидентифицирани параметри на динамичния модел. Именно такъв пример се наблюдава в работата на Li (Li, Wang, Dong, & Ghinea, 2020), където рекурентна архитектура се използва за компенсиране на нелинейните изкривявания в процеса, докато базовият линейно-гаусов модел продължава да се оценява чрез KF. Същата идея е видима и при Al-Sharman и неговия екип (Al-Sharman, и др., 2019), където авторите демонстрират, че добре обучена ANN може да замести аналитичните нелинейни функции и да подаде към KF по-коректна „ефективна“ оценка на процесното ускорение.

Ролята на KF тук остава ясно дефинирана: филтърът осигурява оптималност при линейни отношения и гаусов шум, подпомага стабилизиране на оценката и често е по-устойчив, дори когато ANN проявява частична грешка или недостатъчна генерализация. Обратната роля на невронната мрежа е да предоставя пропуснатите динамични зависимости, които не присъстват в аналитичния модел. Така например остатъчната грешка между наблюдаваното ускорение и прогнозното ускорение от KF може да бъде използвана като вход към ANN, която да прогнозира нелинейната компонента и да я връща обратно в цикъла за корекция.

Математически архитектурата се формулира чрез модифициран процесен модел:

$$x_{k+1} = Fx_k + Bu_k + \hat{f}_\theta(x_k, u_k) + w_k, \quad (2.15)$$

където \hat{f}_θ е обучаемата невронна мрежа, която моделира нелинейните отклонения от линейната динамика. В някои случаи се използва и корекция върху изхода:

$$y_k = Hx_k + \hat{g}_\phi(x_k) + v_k, \quad (2.16)$$

където:

- u_k е векторът на измерванията (изходите) на системата в дискретния момент k ;
- H е матрицата на линейната част на измервателния модел, която свързва състоянието с измерванията;
- x_k е векторът на състоянието на системата в момент k ;
- $\hat{g}_\phi(\cdot)$ е параметризирана нелинейна функция с параметри ϕ , реализирана чрез невронна мрежа, която моделира нелинейната част на измервателния процес;
- v_k е векторът на измервателния шум в момент k , който отчита сензорните грешки и смущения.

Паралелният невронен клон позволява да се моделират немоделирани нелинейности, температурно-зависими дрейфове и измервания с деградирано качество, които са типични за MEMS-IMU. Редица изследвания по GNSS/INS интеграция, включително *GNSS/INS Integration Methods* (Angrisano, 2010) и *Integrating INS and GNSS Sensors to Provide Reliable Surface Positioning* (Russell, 2012), докладват съществено ограничаване на натрупващата се (интеграционна) грешка и повишена устойчивост към сензорен шум спрямо самостоятелна инерциална навигация. При сценарии с прекъсвания на GNSS хибридни подходи с изкуствена невронна мрежа (ANN) могат да подпомогнат филтъра да запази консистентността на оценките чрез моделиране на остатъчни нелинейни отклонения и систематични грешки, които иначе биха довели до ускорено натрупване на грешка при **счисление**.

Тази хибридна двуканална архитектура е относително компактна и запазва стандартната структура на Калмановия филтър (KF) (прогнозиране–корекция), като в прогнозната стъпка динамичният модел се апроксимира от невронна мрежа, т.е. $x_{k+1} = f_\theta(x_k, u_k)$. Интегрирането на оценките остава отделен, ясно дефиниран етап, което улеснява анализа и настройката на системата. Архитектурата е компактна и подходяща за реализация в MATLAB/Simulink, поради което е уместна за Глава 3, където ще бъде изграден симулационен модел на хибридна навигационна система.

Един от най-критичните проблеми при реалното внедряване на Калманов филтър в навигационни системи е изборът на матриците на процесния и измервателния шум. В класическия подход параметрите Q и R се определят ръчно въз основа на лабораторни данни или чрез инженерна оценка. В условията на динамични експлоатационни среди – като вибрации, температурни градиенти, GNSS смущения или паразитни смущения върху IMU – тези параметри престават да бъдат постоянни. Това води до некоректни оценки и нарушава оптималността на филтъра.

Съвременните методи на изкуствения интелект предоставят механизъм за автоматично, адаптивно и базираното на данни (data-driven) определяне на шумовите характеристики, като допълват структурните предимства на KF с обучаеми модели.

2.5. Обучение с утвърждаване за навигация – сравнителен анализ.

Парадигмата на обучение с утвърждаване (Reinforcement Learning, RL) предлага коренно различен подход към проблема за навигация в сравнение с филтрационните методи. Докато KF-базираните системи се стремят да оценят текущото състояние чрез моделиране на процесните и измервателните шумове, RL разглежда задачата като оптимизационен проблем на поведенческа политика. Целта е да се научи стратегия $\pi(a | s)$, която максимизира дългосрочната възвръщаемост, вместо да се изгражда експлицитна оценка на фазовото пространство.

В работата *Continuous Deep Q-Learning with Model-based Acceleration* (Gu, Lillicrap, Sutskever, & Levine, 2016) се изследват методи за обучение на агенти в непрекъснато управлявани пространства, което е от особена важност за БЛА, където управлението не е дискретно. Авторите показват, че комбинацията между моделно-базирана симулационна динамика и дълбоко Q-обучение ускорява сходимостта и намалява нуждата от експериментални полети.

Освен това в Лияк и Бион (Liaq & Yuun, 2019) разглеждат практическите аспекти на RL за UAV – особено влиянието на частично наблюдаеми среди, грешки в сензорите и динамични препятствия. Резултатите показват, че RL-политики могат да превъзхождат класическия навигационен подход при сложни условия, при които моделът е несигурен или непълен.

Въпреки това RL има ограничения. Политиките често са трудни за интерпретиране, изискват голямо количество симулации, а тяхната стабилност при реални смущения не винаги е гарантирана. Важно е да се подчертае, че RL не замества Калмановата филтрация, а по-скоро може да я допълва при задачи за управление на високо ниво, където целевите движения и маневриране се учат чрез политика, докато ниското ниво на оценка на състоянието остава отговорност на KF/ANN моделите.

Сравнителен анализ на KF, ANN и RL подходи при навигация на БЛА е направен в **Error!** **Reference source not found.** в приложението към дисертационния труд.

2.6. ИЗВОДИ ОТ ГЛАВА 2.

1. При невронните архитектури се наблюдават предимства спрямо класическите аналитични модели в редица сценарии за навигация на БЛА при нелинейна динамика, нестационарни шумове и частична наблюдаемост, особено когато моделната неопределеност е трудно описуема аналитично. В този контекст рекурентните невронни мрежи от тип RNN, GRU и LSTM моделират времеви зависимости и акумулират контекст във времето, което е критично при навигационни задачи с дрейф в оценката и натрупващи се грешки при счисление.
2. LSTM невронната мрежа позволява моделиране на дългосрочни зависимости и е приложима при силно нелинейни процеси. При наличие на неравномерни времеви интервали и спорадични наблюдения е необходимо експлицитно отчитане на времевата скала, като непрекъснато-времевите подходи от типа Neural ODE/Neural SDE предоставят естествена формулировка за такива режими.
3. Анализът на литературата по оценяване на състоянието и прогнозиране на движение показва, че LSTM моделите постигат добра генерализация между симулационни и реални траектории при подходящо обучение, валидиране и контрол на разликите между симулационна и реална среда.
4. Комбинирането на Калманова филтрация (KF) с изкуствени невронни мрежи позволява запазване на структурираното интегриране на сензорни данни и вероятностната интерпретация на оценката, при условие на коректно моделиране и калибриране на шумовите характеристики.
5. Невронните модели в рамките на KF архитектури компенсират неточности в динамичния модел (напр. чрез моделиране на остатъчни отклонения) и да подпомогнат адаптацията към нестационарни шумове, например чрез оценяване/адаптиране на матриците на

ковариация на процесния и измервателния шум Q и R , както и чрез детекция на аномални измервания (outliers), които се отклоняват съществено от очакваното поведение на сензорите.

6. Анализът в настоящата глава обосновава избора на интегриран навигационен модел, при който преходната функция (динамичният модел) на филтъра се параметризира от LSTM мрежа, обучена със сензорни данни и симулационни траектории, като неопределеността на модела се представя чрез оценяване/адаптация на процесния шум.

ГЛАВА 3. ПРОЕКТИРАНЕ НА ИНТЕГРИРАНА НАВИГАЦИОННА СИСТЕМА В MATLAB/SIMULINK.

Настоящата глава представя процеса на проектиране, моделиране и симулиране на интегрирана навигационна постановка за безпилотен летателен апарат в средата MATLAB/Simulink. Въз основа на теоретичните постановки от Глава 1 и разработените подходи от Глава 2 се изгражда интегриран симулационен модел, който реализира съвместна обработка на инерциални и спътникови (GNSS) измервания в рамка на Калманова филтрация, подпомогната от невронен процесен предиктор. Разработената система демонстрира хибриден подход, при който класическият EKF се комбинира с дълбока рекурентна мрежа от тип LSTM, използвана като процесен модел в прогнозната стъпка на филтъра.

В тази експериментална постановка MATLAB/Simulink се използва като интеграционна платформа, осигуряваща математическо моделиране, визуална блокова реализация и симулация на сензорни потоци. Върху базова схема на инерциална навигационна система (INS) са реализирани блокове, представящи IMU, GNSS и измервателни шумове, както и EKF модул за оценяване на състоянието. Добавен е невронен блок, обучен в MATLAB чрез Deep Learning Toolbox, който изпълнява ролята на нелинеен прогнозен оператор за процесната динамика в прогнозната стъпка.

Построеният модел позволява директно сравнение между резултатите, получени чрез аналитичния процесен модел и тези, генерирани от невронната апроксимация, при идентични входни условия. По този начин се осигурява експериментална основа за количествен анализ на точността и устойчивостта на оценяването при различни шумови профили и при временна загуба на GNSS сигнал.

В следващите раздели са представени концептуалната архитектура на системата, математическият модел на процесите и измерванията, параметризацията на шумовете, както и интеграцията на LSTM мрежата в цикъла на Калмановата филтрация. Специално внимание е отделено на реализацията в средата Simulink, където отделните компоненти са оформени в блокова структура, позволяваща последователно тестване и визуализация на резултатите.

3.1. Математически модел на системата.

Математическото моделиране на интегрираната навигационна система има за цел да опише взаимовръзката между движението на летателния апарат, измерванията от инерциалните сензори и външните наблюдения, както и влиянието на шумовите процеси. Разработеният модел е формулиран така, че да бъде едновременно съвместим със симулационната среда в MATLAB/Simulink и адаптивен към невронната архитектура, използвана за прогнозиране на състоянието чрез LSTM предиктор.

Математическият модел се базира на класическата структура на инерциално-навигационните системи, включваща **динамично уравнение** за еволюцията на състоянието, **измервателно уравнение** за описване на наблюденията, **модел на шумовете** за симулиране на сензорните несъвършенства и **Extended Kalman Filter (EKF)** като стандартен метод за оценка на състоянието.

Динамичният модел на системата описва връзката между вектора на състоянието и управляващите въздействия. За целите на интегрираната инерциална навигация се приема шестизмерен вектор на състоянието:

$$\mathbf{x}_k = [p_x \ p_y \ p_z \ v_x \ v_y \ v_z]^T, \quad (3.1)$$

където p_x, p_y, p_z са координатите на позицията, а v_x, v_y, v_z — компонентите на скоростта в инерциална рамка.

Управляващите въздействия \mathbf{u}_k се определят от IMU / сензорни данни:

$$\mathbf{u}_k = [a_x \ a_y \ a_z \ \omega_x \ \omega_y \ \omega_z]^T, \quad (3.2)$$

където a_i и ω_i са проектираните ускорения и ъглови скорости; в настоящия опростен модел се приема, че ускоренията a_k са предварително приведени в инерциалната рамка (или еквивалентно – че тяловата и инерциална рамка съвпадат), поради което ω_k не участва в (3.3).

Динамичното уравнение в дискретна форма е формулирано съгласно:

$$\mathbf{x}_{k+1} = f_{dyn}(\mathbf{x}_k, \mathbf{u}_k, \Delta t) = [\mathbf{p}_k + \mathbf{v}_k \Delta t + \frac{1}{2} \mathbf{a}_k \Delta t^2] + \mathbf{w}_k, \quad (3.3)$$

$$\mathbf{v}_k + \mathbf{a}_k \Delta t$$

където:

- \mathbf{x}_{k+1} е векторът на състоянието на системата в дискретния момент $k+1$;
- \mathbf{x}_k е векторът на състоянието в момент k ;
- $f_{dyn}(\cdot)$ е динамичната функция на процесния модел, която описва еволюцията на състоянието за интервал Δt ;
- \mathbf{u}_k е векторът на управляващите входове в момент k ;
- Δt е времеви интервал между два последователни дискретни момента;
- \mathbf{p}_k е векторът на позицията в момент k ;
- \mathbf{v}_k е векторът на скоростта в момент k ;
- \mathbf{a}_k е векторът на ускорението в момент k ;
- \mathbf{w}_k е векторът на процесния шум, който моделира немоделирани динамики и външни въздействия;
- $[\cdot]$ обозначава векторната форма на процесния модел, включваща уравненията за позиция и скорост.

В Simulink този модел е реализиран чрез MATLAB Function блока **f_dyn**, който приема текущото състояние \mathbf{x}_k , управляващия вектор \mathbf{u}_k и времевата стъпка $T_s = 0.01$ s, изчислявайки \mathbf{x}_{k+1} .

Паралелно на него в системата е интегриран и невронният модел:

$$\hat{\mathbf{x}}_{k+1} = f_{LSTM}([\mathbf{x}_{k-L+1}, \dots, \mathbf{x}_k], [\mathbf{u}_{k-L+1}, \dots, \mathbf{u}_k]), \quad (3.4)$$

където:

- $\hat{\mathbf{x}}_{k+1}$ е прогнозната оценка на вектора на състоянието в дискретния момент $k+1$;
- $f_{LSTM}(\cdot)$ е нелинейна функция, реализирана чрез LSTM невронна мрежа, която моделира динамиката на системата на база последователни данни;
- $\mathbf{x}_{k-L+1}, \dots, \mathbf{x}_k$ са векторите на състоянието за предходните L дискретни момента, формиращи входната времева последователност;
- $\mathbf{u}_{k-L+1}, \dots, \mathbf{u}_k$ са векторите на управляващите входове за съответните L дискретни момента;
- L е дължината на времеви прозорец (look-back window), използван от LSTM мрежата за извличане на времеви зависимости.

Моделът представлява нелинейна апроксимация на динамиката чрез обучената дълбока рекурентна мрежа. Двата модела - аналитичният и LSTM - са структурно идентични по отношение на входно-изходните връзки и позволяват директно сравнение на производителността.

Измервателното уравнение описва връзката между наблюдаваните величини \mathbf{z}_k и истинското състояние \mathbf{x}_k . В интегрираната навигационна система се комбинират данните от GNSS и IMU, като GNSS осигурява абсолютни координати, а IMU — относителни изменения на скоростта.

В общ вид измервателното уравнение се записва съгласно израза:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (3.5)$$

където \mathbf{v}_k е векторът на измервателния шум. При линеаризация около текущото състояние:

$$\mathbf{z}_k = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \Delta t + \mathbf{v}_k, \quad (3.6)$$

където:

- \mathbf{z}_k е векторът на измерванията в дискретния момент k ;
- \mathbf{p}_x , \mathbf{p}_y , \mathbf{p}_z са компонентите на вектора на позицията по съответните оси на координатната система;
- \mathbf{v}_x , \mathbf{v}_y , \mathbf{v}_z са компонентите на вектора на скоростта по съответните оси;
- Δt е времевият интервал между два последователни дискретни момента;
- $[\cdot]$ обозначава векторна форма на измервателния модел;
- \mathbf{v}_k е векторът на измервателния шум в момент k , който отчита сензорните грешки и смущения.

Резултатът съответства на изходите от блока **gPos** в Simulink модела. GNSS шумът се моделира чрез диагонална матрица на дисперсията $R = \text{diag}(1.5^2, 1.5^2, 2.0^2)$, като вертикалният компонент има двойно по-голяма дисперсия спрямо хоризонталните. Тези стойности се реализират в блока **diag([1.5 1.5 2.0])u*, което осигурява реалистично поведение на навигационните измервания.

Измервателното уравнение в LSTM клона е еквивалентно, но прогнозата $\hat{\mathbf{x}}_{k+1}$ се подава като вход към същия наблюдателен механизъм. Това осигурява пълна съпоставимост между двата модела и идентична обработка на измерванията.

Шумовете в системата се разделят на два основни типа — **процесен шум** и **измервателен шум**. Тяхното моделиране има съществено значение за реалистичната симулация и за коректността на обучителните данни, използвани от LSTM предиктора.

Процесният шум \mathbf{w}_k отразява неопределеността в динамиката и е моделиран като гаусово разпределен вектор с нулево средно и ковариационна матрица Q : $\mathbf{w}_k \sim \mathcal{N}(0, Q)$, $Q = \text{diag}(0.01^2, 0.01^2, 0.02^2, 0.01^2, 0.01^2, 0.01^2)$. Този шум се въвежда чрез блоковете **nAcc** и **nGyro**, които генерират случайни сигнали с дисперсия, зададена от матриците **Gacc** и **Ggyro**.

Измервателният шум \mathbf{v}_k се прилага върху изходите на GNSS блока **gPos**, с диагонални коефициенти 1.5, 1.5 и 2.0, което създава вертикална грешка два пъти по-голяма от хоризонталната.

В комбинация, тези шумове осигуряват приближение до реалните условия при работа на инерциални и спътникови навигационни системи, включително влиянието на сензорните дрейфове и несъвършенства.

Шумовите процеси са от ключово значение и при обучението на LSTM мрежата. Тъй като данните за обучение са генерирани от симулация, съдържаща реалистичен шум, невронната мрежа научава статистическите зависимости между измерванията и скритите състояния дори при стохастични отклонения. Това осигурява по-добра генерализация при използване в експерименталната симулация.

За валидиране на динамичния модел е имплементиран **разширен Калманов филтър (ЕКФ)**, който осигурява оптимална оценка на състоянието в условия на нелинейност и шум. ЕКФ комбинира прогнозата от модела f_{dyn} с наблюденията от измервателната подсистема, като минимизира средноквадратичната грешка чрез последователни итерации на прогнозиране и корекция.

Алгоритъмът се състои от два етапа:

(1) Прогнозиране:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= f_{dyn}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}), \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + Q_k,\end{aligned}\quad (3.7)$$

където $F_k = \frac{\partial f}{\partial \mathbf{x}}$ е якобианът на динамичния модел.

(2) Корекция:

$$\begin{aligned}K_k &= P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}, \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})), \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1},\end{aligned}\quad (3.8)$$

където $H_k = \frac{\partial h}{\partial \mathbf{x}}$ е матрицата на наблюдение.

В контекста на симулирания модел EKF е приложен върху резултатите от блока **f_dyn**, като служи за оценка на точността на динамичния модел и потвърждава стабилността на изчислените състояния. Тъй като LSTM моделът няма аналитична форма, неговото включване в EKF се разглежда като нелинеен черен блок, чиито частни производни могат да бъдат апроксимирани чрез числени методи.

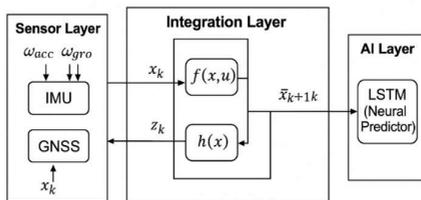
Комбинацията между EKF и LSTM модел открива перспектива за **хибридни филтри**, при които невронната мрежа замества функцията f_{dyn} , а класическата EKF рамка осигурява корекция спрямо измерванията. Така системата може да съчетава обучаеми нелинейни зависимости и статистически оптимална оценка на състоянието.

3.2. Концептуална архитектура на интегрираната система

Концептуалната архитектура на интегрираната навигационна система е изградена върху принципа на комбиниране на класически алгоритми за филтриране и оценяване на състоянието с методи на изкуствения интелект за нелинейно моделиране на динамиката. Системата включва взаимодействащи подсистеми за измерване, обработка и прогнозиране, реализирани в MATLAB/Simulink, които възпроизвеждат основните функционални модули на бордовия навигационен комплекс на безпилотен летателен апарат.

На концептуално ниво архитектурата се състои от три основни слоя. В първия слой се намира сензорната подсистема, включваща инерциално измервателно устройство (IMU) и GNSS приемник, моделирани чрез блокове за линейно и ъглово ускорение, както и съответните шумови процеси. Вторият слой представлява интегриращия навигационен филтър, базиран на разширен Калманов филтър (EKF), който обединява данните от различните източници и оценява текущото състояние на апарата. Третият слой реализира невронния предиктор – дълбока рекурентна мрежа от тип LSTM, която замества аналитичния динамичен модел в прогнозната (predict) част на филтъра.

Връзките между тези подсистеми са организирани така, че изходите на IMU и GNSS блоковете да се подават към EKF, който чрез рекурентна структура и невронно прогнозиране осигурява актуализирани стойности на позицията и линейната скорост. MATLAB/Simulink се използва като платформа за интеграция, позволяваща едновременното симулиране на физичните процеси, алгоритмите за оценка и обучените модели на изкуствен интелект в рамките на единен блоков модел (Фигура 23).



Фигура 23. Концептуална архитектура на Интегрирана Навигационна Система

3.2.1. Сензори: IMU, GNSS, шумове

В основата на симулираната интегрирана навигационна система стои моделът на сензорния слой, който възпроизвежда работата на инерциалното измервателно устройство (IMU) и спътниковата навигационна система (GNSS). Тези два източника на информация осигуряват съответно висока честота на измерване, но с натрупваща се грешка, и ниска честота с абсолютна корекция, което е класическият принцип на интеграция при системите от тип INS/GNSS. В настоящата симулация този слой е реализиран в MATLAB/Simulink посредством комбинация от функционални блокове, описващи динамиката на акселерометри, жirosкопи и позиционни сензори, както и съответните шумови процеси, моделирани стохастично.

IMU модел

Инерциалното измервателно устройство се състои от три акселерометъра и три жirosкопа, които измерват линейните и ъгловите ускорения по трите оси на тялото на апарата. В Simulink те са

представени чрез два отделни канала – **Gacc** и **Ggyro**, всеки реализиран с диагонална матрица за мащабиране на сигналите.

Моделът на акселерометъра се описва с израза:

$$\mathbf{a}_m = \mathbf{G}_{acc} \mathbf{a}_{true} + \mathbf{n}_{acc}, \quad (3.9)$$

където:

- \mathbf{a}_m е измереното ускорение;
- $\mathbf{G}_{acc} = \text{diag}([0.2, 0.2, 0.3])$ е матрицата на чувствителността;
- \mathbf{a}_{true} е реалното ускорение;
- \mathbf{n}_{acc} – добавеният шум.

Моделът на жирокопа е аналогичен:

$$\boldsymbol{\omega}_m = \mathbf{G}_{gyro} \boldsymbol{\omega}_{true} + \mathbf{n}_{gyro}, \quad (3.10)$$

където $\mathbf{G}_{gyro} = \text{diag}([0.01, 0.01, 0.02])$.

В Simulink тези изрази се реализират чрез блокове **Gain** (Gacc, Ggyro) и **Sum** (sumAcc, sGyro), които събират реалния сигнал и шумовите компоненти, подадени от източниците **nAcc** и **nGyro**.

Шумовите процеси се моделират като бели гаусови шумове с нулево средно и стандартно отклонение, съответстващо на реалните характеристики на MEMS сензори. При симулацията е зададено средно квадратично отклонение $\sigma_{acc} = 0.2 \text{ m/s}^2$ и $\sigma_{gyro} = 0.01 \text{ rad/s}$.

GNSS модел

GNSS-позиционирането е реализирано чрез блок **gPos**, който симулира измерванията на спътниковата навигация като линейна комбинация на реалното положение и добавен измервателен шум. Математически моделът се задава съгласно:

$$\mathbf{p}_m = \mathbf{G}_{pos} \mathbf{p}_{true} + \mathbf{n}_{pos}, \quad (3.11)$$

където $\mathbf{G}_{pos} = \text{diag}([1.5, 1.5, 2.0])$ задава мащабирането на грешката по трите координатни оси, а \mathbf{n}_{pos} е шумът на псевдодалечината, симулиращ неточността на GNSS решението.

Този модел отразява реалистичната динамика на системите GPS/GLONASS при нисковисочинни БЛА, при които грешката по вертикалата е обикновено два пъти по-голяма от тази по хоризонталните оси. Във времевата област GNSS блокът генерира дискретни измервания със стъпка на дискретизация $T_s = 0.01 \text{ s}$, което съответства на честота 100 Hz.

Интеграция на шумовете

Шумовите източници са въведени отделно за всяка група сензори и се подават чрез сигнални линии, които влизат в съответните блокове за сумиране. Това позволява контролирано въвеждане на различни типове смущения – както адитивни (бели шумове), така и мултипликативни (вариращи коефициенти на усилване).

В процеса на симулация тези шумове осигуряват реалистично поведение на сензорния слой, което от своя страна позволява адекватно обучение и тестване на LSTM модела. Именно чрез комбинацията от реален динамичен модел и стохастични компоненти се постига балансиран сценарий, при който невронната мрежа се обучава да прогнозира състоянието в условия на несигурност, подобна на тази при реален полет.

На Фигура 24 схематично е представена интеграцията на сензорите и шумовете в модела.

прави неподходящ за директно вграждане в реалновремени филтрационен цикъл. От друга страна, предварителното обучение позволява мрежата да бъде изложена на широк спектър от динамични режими и смущения, които трудно биха се появили в рамките на единичен експеримент или полет. По този начин се постига по-добра обобщаваща способност и по-висока устойчивост на прогнозите при реална експлоатация.

Обучаващият набор от данни е генериран чрез симулация на референтен динамичен модел на БЛА в MATLAB/Simulink. Използвани са разнообразни траектории и режими на управление, включващи праволинейно движение, ускорителни и спиращи маневри, както и комбинирани линейни и ъглови движения. Към симулираните данни са добавени адитивни шумове и смущения, съответстващи на реалистични характеристики на MEMS инерциални сензори, с цел мрежата да усвои не идеализиран, а „ефективен“ динамичен модел, включващ както номиналната динамика, така и типичните системни отклонения.

Входът към LSTM мрежата е организиран като последователност от вектори, съдържащи нормализирани стойности на оцененото състояние \mathbf{x} и управляващия вектор \mathbf{u} за последните L дискретни времеви момента. Нормализацията на входните данни е извършена спрямо статистиките на обучаващия набор (средна стойност и стандартно отклонение), което е стандартна практика при обучение на рекурентни мрежи и допринася за числена стабилност и по-бърза конвергенция на алгоритъма за обучение. По този начин мрежата работи с мащабно съпоставими входове, независимо от физическите размерности на отделните компоненти на състоянието и управлението.

Изходът на LSTM мрежата е дефиниран като едностъпкова прогноза на шестизмерното състояние $\hat{\mathbf{x}}_{k+1|k}$. Изборът на едностъпкова прогноза е съзнателен и е ясно свързан с интеграцията на мрежата в структурата на Калмановия филтър. Калмановата филтрация работи итеративно, като във всяка дискретна стъпка извършва прогнозиране и корекция. Използването на едностъпков прогнозен модел осигурява съвместимост с тази структура и ограничава натрупването на грешки, характерно за многостъпковите невронни прогнози, при които малки отклонения могат да се усилят във времето. По този начин LSTM мрежата замества аналитичната функция на преход $f(\cdot)$, без да нарушава рекурсивния характер на филтъра.

Шестизмерното състояние е избрано с оглед балансиране между информативност и числена устойчивост на филтрационния процес. В разглеждания опростен модел шестизмерното състояние включва позиция и линейна скорост на БЛА в инерциална рамка. Този избор позволява LSTM мрежата да моделира директно динамиката на движение, без да се въвеждат излишни състояния, които биха усложнили обучението и биха увеличили риска от пренапасване. В същото време шестизмерното състояние е достатъчно информативно, за да отрази основните нелинейни зависимости между управление, инерциални ефекти и кинематичен отговор на системата.

По време на симулацията блокът **nn_predict_lstm** поддържа плъзгащ прозорец върху буфер от последните L входни вектори. При всяка дискретна стъпка този прозорец се обновява, след което LSTM мрежата извършва едностъпкова прогноза на състоянието $\hat{\mathbf{x}}_{k+1|k}$. Получената прогноза се използва директно в прогнозната стъпка на Калмановия филтър, като по този начин аналитичният модел на динамиката се замества с обучаем, базиран на данни (data-driven) модел. Коригиращата стъпка на филтъра остава непроменена, което гарантира статистическа консистентност и позволява ефективно комбиниране на невронната прогноза със сензорните измервания.

Превключвателят **Manual Switch** играе основна роля в архитектурата – чрез него се избира кой модел се използва за изчисляване на прогнозното състояние. В позиция „нагоре“ системата използва динамичния модел f_{dyn} , а в позиция „надолу“ – LSTM предиктора. Това позволява незабавно сравнение между двата подхода при идентични входни въздействия и начални условия.

Блокът **selAcc** подава селектираните измервания от акселерометрите, докато **selGyro** осигурява сигналите от жirosкопите. Изходите им се подават към съответните шумови генератори **nAcc** и **nGyro**, чиято цел е да въведат стохастична компонента, симулираща несъвършенствата на реалните MEMS сензори. Тези шумове се усилят чрез матрици на чувствителност (**Gacc**, **Ggyro**) и се сумират със съответните реални сигнали посредством блокове **sumAcc** и **sGyro**.

Получените сигнали представляват реалистични измервания на инерциалната платформа и се обединяват в шесткомпонентен вектор u_{noisy} , който постъпва едновременно към двата модела — динамичния и невронния.

На изхода на блоковете **f_dyn** и **nn_predict_lstm** се формира новото състояние x_{next} , което се записва в съответните логове **out.log_x_f** и **out.log_x_nn**. Логът **out.log_x** съхранява референтното „истинско“ състояние, а чрез разликата между прогнозите и него се оценява ефективността на двата модела.

Позиционните измервания от GNSS се реализират чрез блок **gPos**, който съдържа диагонална матрица на шумовете $\text{diag}([1.5, 1.5, 2.0])$, като вертикалната грешка е два пъти по-голяма от хоризонталните. Тези измервания се обединяват с резултатите от INS блока в **sumPos**, за да се формира смесеният вектор z_k , съответстващ на реалната интегрирана навигация.

Кодът в **train_lstm.m** и **nn_predict_lstm.m** реализира методологията за обучение и интеграция на невронната мрежа в Simulink. Обучението се базира на предварително подготвени последователности $[x_k, u_k] \rightarrow x_{k+1}$, които отразяват динамиката на аппарата в условия на шум. Невронната архитектура включва входен слой от 12 признака, един скрит LSTM слой със 64 неврона и регресионен изходен слой от 6 параметъра, съответстващи на вектора на състоянието.

След обучението мрежата се експортира във файл **lstm_dyn.mat**, който се зарежда по време на симулация от **nn_predict_lstm.m**. В този файл се съхраняват и параметрите на нормализация ($\mu_x, \sigma_x, \mu_u, \sigma_u$), осигуряващи съвместимост на данните между MATLAB и Simulink средата (параметрите са описани в **Error! Reference source not found.** на приложението).

При симулация в реално време MATLAB Function блокът **nn_predict_lstm** извиква метода **predict()** на обучената мрежа и връща едностъпкова прогноза. Тази интеграция осигурява затворен контур между машинното обучение и физическата симулация – важен принцип при изследванията на автономни системи.

Тази архитектура е избрана с цел да позволи паралелна работа на два независими процесни модела – класически и невронен, при пълна съвместимост на входовете и изходите. Превключвателят служи като експериментален механизъм за оценка на устойчивостта и точността на LSTM модела в сравнение с аналитичния динамичен модел.

Системата дава възможност за:

- сравнение на динамични траектории при еднакви условия;
- изчисляване на грешките в реално време чрез RMSE критерий;
- визуализация на състоянията x, y, z в тримерно пространство;
- оценка на устойчивостта при шумови смущения.

3.2.3. Интеграция на LSTM предиктора.

Интеграцията на LSTM предиктора в средата на MATLAB/Simulink реализира преходът от аналитично описание на динамиката към машинно-обучен модел. Основната цел на тази интеграция е създаването на хибридна симулационна платформа, в която невронната мрежа поема ролята на процесен модел f_{dyn} , като прогнозира състоянието на системата въз основа на времеви последователности от векторите на състояние и управление.

Обучението се извършва с оптимизатор **Adam**, 35 епохи и размер на минипакетите 128. Алгоритъмът **Adam** (Adaptive Moment Estimation) е метод за стохастично градиентно спускане, който се е наложил като стандарт в обучението на дълбоки рекурентни архитектури. Изборът на този оптимизатор е продиктуван от способността му да адаптира динамично скоростта на обучение (learning rate) за всеки отделен параметър на модела, базирайки се на оценки на първия и втория момент на градиентите.

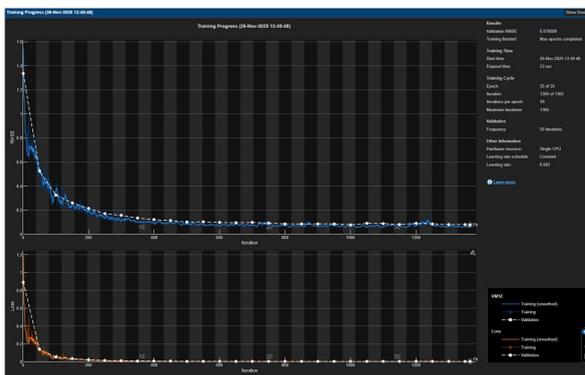
За разлика от класическото стохастично градиентно спускане (SGD), където стъпката на обучение е фиксирана за всички тегла, Adam комбинира предимствата на методите **AdaGrad** (Adaptive Gradient Algorithm) и **RMSProp** (Root Mean Square Propagation). Алгоритъмът поддържа експоненциално затихващи средни стойности на миналите градиенти (аналог на инерционен момент) и на техните квадрати (аналог на нецентрирана дисперсия). Това позволява ефективна

навигация в повърхнината на грешката, като се ускорява сходимостта в "равните" участъци и се предотвратяват резките осцилации в стръмните направления – критично предимство при обучението на LSTM мрежи върху зашумени сензорни данни.

Хиперпараметрите на тренировъчния процес са настроени емпирично за постигане на баланс между скорост на сходимост и генерализационна способност:

- **Продължителност:** Обучението се провежда в рамките на **35 епохи**, което осигурява достатъчно итерации за минимизиране на функцията на загубите (Loss Function), без да се достига до преобучение (overfitting).
- **Размер на мини-пакета (Mini-batch size):** Използван е размер от **128 примера**. Този обем осигурява достатъчно точна стохастична оценка на градиента, намалявайки шума при обновяването на теглата, като същевременно позволява ефективно паралелизиране на изчисленията върху графичния ускорител (GPU).

Исходният модел **lstm_dyn.mat** съдържа както обучените параметри, така и нормализационните коефициенти, осигуряващи съвместимост между тренировъчната и симулационната среда. На Фигура 26 е показана нормализационната графика при обучение на модела.



Фигура 26. Графика на обучение на модела в Mathlab Simulink.

След обучението моделът се интегрира чрез MATLAB Function блока **nn_predict_lstm**, който се извиква във всеки времеви цикъл от Simulink.

След приключване на симулационните експерименти, генерираните масиви от данни се подлагат на последваща обработка (post-processing) и анализ в програмната среда MATLAB. Методологията за валидация включва изчисляване на средноквадратичната грешка (Root Mean Square Error – RMSE) като обективен количествен критерий за точност, както и сравнителна визуализация на оценените траектории спрямо еталонните (Ground Truth).

След корекция на конфигурацията на симулационния модел и повторно провеждане на експериментите се наблюдава промяна в поведението на LSTM предиктора, която е съвместима с коректната му интеграция в прогнозната стъпка на Калмановия филтър. Корекцията се състои в промяна на начина, по който невронният модел е включен в прогнозната стъпка, като превключвателят в симулационната схема е конфигуриран така, че изходът на LSTM да участва директно в изчисляването на априорната оценка на състоянието вместо аналитичния динамичен модел. По този начин е реализирана предвидената хибридна структура LSTM+KF, при която невронният модел изпълнява ролята на предиктор, а измерванията се използват в корекционната стъпка на филтъра.

Получените резултати индикират, че високата грешка, наблюдавана в началните тестове, е била в значителна степен повлияна от некоректна интеграция/конфигурация на невронния модел в

хибридна структура. При коректна настройка невронният модел възпроизвежда динамиката на системата по-адекватно спрямо некоректния режим на работа.

Въпреки това, както е представено в Таблица 8, при финалното сравнение между двата подхода аналитичният динамичен модел f_dyn демонстрира по-висока точност в разглеждания сценарий. Например за шестата компонента на състоянието (State 6) е отчетено $RMSE_f_dyn = 0.0031$, докато при LSTM е $RMSE_LSTM = 0.2652$. Аналогична тенденция се наблюдава и за останалите компоненти, което показва, че при дадените условия аналитичният модел запазва значително по-висока точност..

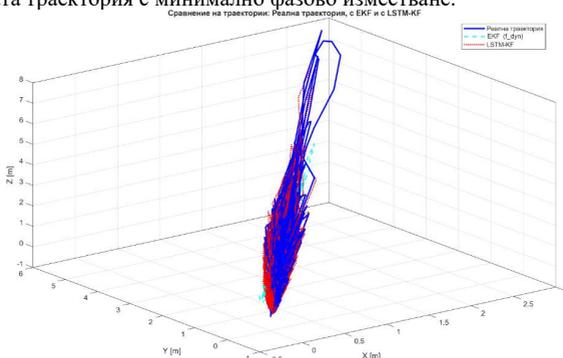
Таблица 1. Статистическа оценка на точността на двата модела.

State	RMSE f_dyn	RMSE LSTM
1	0.0021	0.1985
2	0.0037	0.3839
3	0.0054	0.6107
4	0.0011	0.1344
5	0.0021	0.1448
6	0.0031	0.2652

Източник: MathLab симулация в конзолата на приложението.

Резултатите подчертават, че LSTM подходът не следва да се разглежда като директна и универсална замяна на аналитичния динамичен модел, а като допълващ инструмент, чиято ефективност зависи от конфигурацията, режима на работа и конкретния сценарий. В условия, при които физическият модел е добре дефиниран и системата не проявява изразено нестандартно поведение, моделно-базираният предиктор остава по-надежден по отношение на точността

Фигура 27 показва динамичното поведение на двата модела. При достатъчно количество тренировъчни данни и коректно зададен буфер, дължината на LSTM прозореца осигурява прогноза, която следва реалната траектория с минимално фазово изместване.



Фигура 27. Графика сравняваща траекторията на синтетичните данни и LSTM модела.

В таблица 22Error! Reference source not found. на приложение №1 към дисертационния труд са представени всички параметри на обучението и производителността на модела. Интеграцията на LSTM предиктора в Simulink демонстрира възможността за създаване на хибридна навигационна система, в която машинното обучение не просто имитира, а частично замества аналитичните уравнения на движението. Така разработената система позволява експериментално сравнение на двете парадигми — класическа и невронна, при идентични условия, което представлява съществен принос в областта на интелигентните INS архитектури.

3.3. ИЗВОДИ ОТ ГЛАВА 3.

1. Предложеният модел, реализирана в MATLAB/Simulink, комбинира рекурсивния механизъм за интегриране на сензорни данни (predict-update) на Калмановата филтрация с базирано на данни моделиране на процесната динамика чрез дълбока рекурентна мрежа, като LSTM-базираният процесен блок функционира като алтернатива на

аналитичния процесен модел в прогнозната стъпка, корекционната стъпка остава в рамката на филтрационния алгоритъм, невронният и филтрационният модул работят върху унифицирана времева база със съгласувани интерфейси и размерности на входно–изходните сигнали.

2. При разгледаните симулационни сценарии изпълнението на модела протича без числени аномалии (невалидни или некоректни числови стойности) на изхода на невронния блок, като е предвидена защитна проверка за невалидни стойности в прогнозата на LSTM. За разглеждания сценарий резултатите от таблица 8 показват RMSE за f_{dyn} в интервала 0.0011 - 0.0054 по компоненти, а за LSTM - 0.1344 - 0.6107, което характеризира разлика от порядък между двата процесни предиктора при зададените условия на експеримента.
3. При експерименталното сравнение на аналитичния процесен модел f_{dyn} и процесния предиктор, реализиран с рекурентна невронна мрежа от тип LSTM, като едностъпкови предсказатели на шестизмерното състояние (позиция и линейна скорост), получените RMSE спрямо референтното състояние показват съществена разлика в точността при зададените условия, като f_{dyn} дава стойности от порядък 10^{-3} , а LSTM - от порядък 10^{-1} по компонентите (табл. 8).
4. Получените симулационни резултати обосновават необходимост от допълнителна валидация при различни профили на движение и условия, включително вариации на параметри, временна загуба на измервания и несъответствия между симулационни и реални данни.

ГЛАВА 4. ЕКСПЕРИМЕНТАЛНО ИЗСЛЕДВАНЕ НА ПРЕДЛОЖЕНАТА НЕВРОННО- ПОДСИЛЕНА ИНТЕГРИРАНА НАВИГАЦИОННА СИСТЕМА.

4.1. Описание на експерименталната среда.

Симулационните експерименти са проведени в развойната програмна среда MATLAB/Simulink, като са използвани инструментите от MATLAB and Simulink Student Suite. Пакетът включва MATLAB, Simulink и десет широко използвани допълнителни библиотеки, които осигуряват необходимия набор от средства за числена симулация, обработка на времеви редове, прототипиране и директно изпълнение на модели върху нискобюджетен хардуер. Тази конфигурация предоставя функционална среда за изграждане, тестване и оптимизация на хибридни навигационен модел, който комбинира класическа Калманова филтрация с невронна архитектура от тип LSTM.

Симулациите са изпълнени върху система с архитектура x64, работеща под Windows 11 Pro (Версия 25H2, компилация 26200.7623). Хардуерната платформа включва Intel(R) Core(TM) i7-14700HX (2.10 GHz) процесор с тактова честота 2.10 GHz и 32 GB RAM, което позволява едновременно изпълнение на сложни MATLAB задачи, оптимизация на невронни мрежи и визуализация в реално време на резултатите от Simulink модела (фиг. 25, Глава 3). Наличната RAM памет осигурява достатъчен ресурс за обработка на големи времеви редове, включително синтетично генерирани IMU данни и шумови профили, без да се наблюдават ограничения в производителността. Операционната система и нейните устройства за управление на изчислителните ресурси способстват за стабилно поведение при продължителни експерименти, включително при невронно обучение с mini-batch режим.

Експерименталните данни се състоят от набори синтетични и реалистично моделирани измервания от IMU с триосни ускорения и ъглови скорости, които са генерирани така, че да представят нелинейна динамика, характерна за малки безпилотни апарати. Входните сигнали включват контролни действия, профили на ускорение, ъглова скорост на завъртане (yaw rate), маневри и промени на височината. Данните са структурирани като времеви редове с дискретизация, съответстваща на реален IMU (100–200 Hz), което позволява ефективно тестване на LSTM блока при различни сценарии на стабилност и смущения.

Симуляционната среда допуска контролирано задаване на шумови модели, включително бял гаусов шум, температурно зависим дрейф, отклонения от типа *bias instability* и модел на случайното блуждаене (Random Walk). Тези параметри са реализирани в MATLAB чрез стохастични процеси, за да се апроксимира реалното поведение на MEMS IMU. В отделни тестови сценарии са включени и изкуствено усилените смущения, свързани с високофреквентни вибрации, които често нарушават точността на жироскопите при динамични маневри.

Един съществен аспект на експерименталната настройка е моделирането на GNSS загуби и деградации. В Simulink е конфигуриран модул за симулация на GNSS dropout, включващ пълна загуба на позиционни измервания за интервали от 5 до 40 секунди. Паралелно с това е включен и модел на GNSS шум със спектрални характеристики, които имитират jamming и multipath ефекти, характерни за градски каньони или електронно смущение. Това позволява да се изследва устойчивостта на хибридният модел спрямо отклонения в режим на dead-reckoning и да се оцени дали невронният блок успешно компенсира структурните слабости на класическия процесен модел.

Настройките на експерименталната среда позволяват многократно възпроизвеждане на тестовите сценарии, както и извършване на чувствителностни анализи върху параметрите. MATLAB предлага вградени инструменти за анализ на грешките, визуализация на траектории, както и автоматично изчисляване на показатели като RMSE, иновационна ковариация и скорост на конвергенция. Този интегриран подход позволява детайлна оценка на хибридната архитектура преди преминаване към комплексната симулация на пълната интегрирана навигационна система в следващите раздели.

4.2. Сценарий 1: Нормални условия.

Настоящият експеримент има за цел да оцени поведението на стандартния **Extended Kalman Filter (EKF)** и на предложени **LSTM-усилен Калманов филтър (LSTM-KF)** при нормални условия на движение, без откази в сензорите и при номинални стойности на шумовете в акселерометъра, жироскопа и GNSS модула. Средата на експеримента е реализирана изцяло в MATLAB / Simulink, като блоковата схема на модела съответства на описаната в раздел 3.2.2, а динамичният модел и измервателните уравнения са формулирани съгласно раздел 3.

4.2.1. Методика на провеждане.

Генерираният набор от данни $X_{true}(t)$, $U(t)$ и $Z(t)$ описва движението на безпилотната система в равнинна траектория с постепенно нарастваща скорост и постоянен времеви интервал на дискретизация $\Delta t = 0.01$ s. Данните са използвани за две независими симулации:

1. **EKF с аналитичен модел (f_dyn)** – за прогнозиране на състоянието се използва дискретното уравнение на движението:

$$\mathbf{x}_{k+1} = f_{dyn}(\mathbf{x}_k, \mathbf{u}_k, \Delta t), \quad (4.1)$$

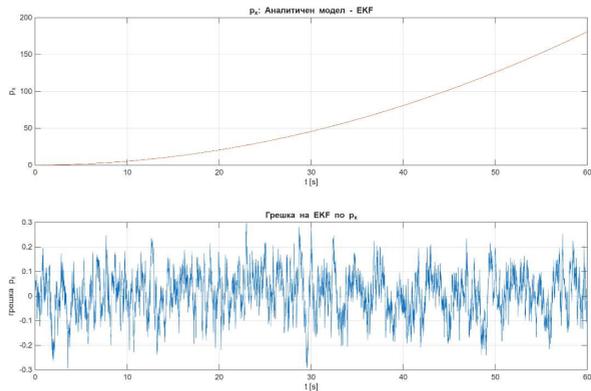
реализирано в скрипта `fdyn_step.m` (вж. Приложението).

2. **EKF с невронен предиктор (LSTM-KF)** – вместо аналитичния модел се използва обучената рекурентна невронна мрежа, имплементирана чрез функцията `nn_predict_lstm.m` (вж. приложението). Мрежата е обучена по скрипта `train_lstm.m` (вж. приложението) върху последователности с дължина $L = 20$, с входове $[\mathbf{x}_k; \mathbf{u}_k]$ и целеви изход \mathbf{x}_{k+1} .

За двата варианта са използвани еднакви начални условия и шумови параметри $\mathbf{Q} = \text{diag}(0.01, 0.01, 0.02, 0.01, 0.01, 0.01)$, $\mathbf{R} = \text{diag}(1.5^2, 1.5^2, 2.0^2)$.

4.2.2. Резултати за EKF с аналитичен модел.

Фигура 28 показва времевите зависимости на координатата p_x и съответната грешка на EKF. Наблюдава се стабилна конвергенция и малка амплитуда на флукуациите около нулата, което свидетелства за адекватно функциониране на филтъра при правилно зададен динамичен модел.



Фигура 28. Динамика на позицията и грешката при EKF с аналитичен модел ($f_{\text{дуп}}$).

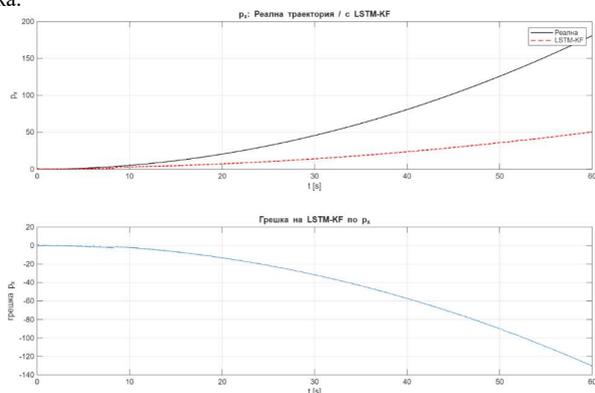
Таблица 9. Средноквадратична грешка (RMSE) на EKF по състояния при нормални условия.

State	RMSE_KF
1	0.0906
2	0.0906
3	0.1203
4	0.0752
5	0.0752
6	0.0741

Средната стойност (Таблица 9) на общата грешка $RMSE_{KF, overall} = 0.087656667$ потвърждава висока точност на прогнозирането при отсъствие на нелинейни ефекти и динамични нарушения.

4.2.3. Резултати за LSTM-KF.

Фигура 29 илюстрира резултатите при използване на LSTM-предиктора. При този сценарий невронната мрежа възпроизвежда динамиката на системата, обучена с данните от Simulink-модела, без пряко познаване на физическите уравнения. Макар че формата на траекторията следва тенденцията на реалната, се наблюдава системно изоставане, което води до натрупване на отрицателна грешка.



Фигура 29. Динамика на позицията и грешката при EKF с LSTM предиктор (LSTM-KF).

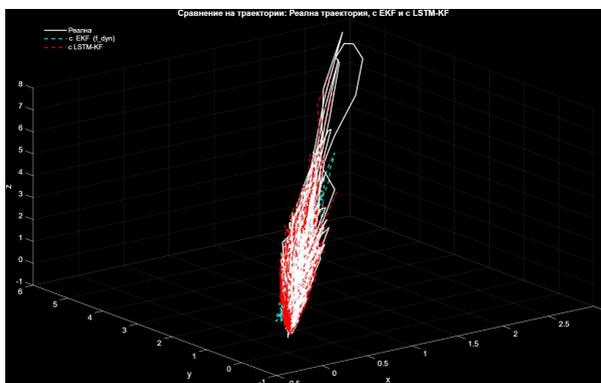
Таблица 10. Средноквадратична грешка (RMSE) на LSTM-KF по състояния при нормални условия.

State	RMSE LSTM
1	57.537
2	114.970
3	171.980
4	3.398
5	5.528
6	8.813

Общата средна грешка е $RMSE_{LSTM, overall} = 87.77$, което е значително по-голяма стойност в сравнение с класическия EKF (Таблица 10). Основната причина е, че LSTM-предикторът е обучен върху ограничен набор от данни и не е извършена допълнителна денормализация в реално време, което води до подценяване на ускорението и следователно до кумулативна позиционна грешка.

4.2.4. Сравнителен анализ.

На Фигура 30 е показана триизмерната траектория на движението при двата филтъра и референтния модел. EKF (сyan) практически съвпада с реалната траектория (бяла линия), докато LSTM-KF (червена линия) следва подобна форма, но с отчетливо систематично изместване по оста Y и Z.



Фигура 30. Сравнение на траекториите при EKF и LSTM-KF в нормални условия.

Анализът на резултатите показва, че при номинални условия и стабилна динамика класическият EKF постига значително по-висока точност благодарение на точното физическо описание на системата. LSTM-усиленият филтър, макар и да демонстрира коректно поведение на скоростните компоненти, изисква допълнително обучение и оптимизация на нормализационните коефициенти, за да постигне сравнима производителност.

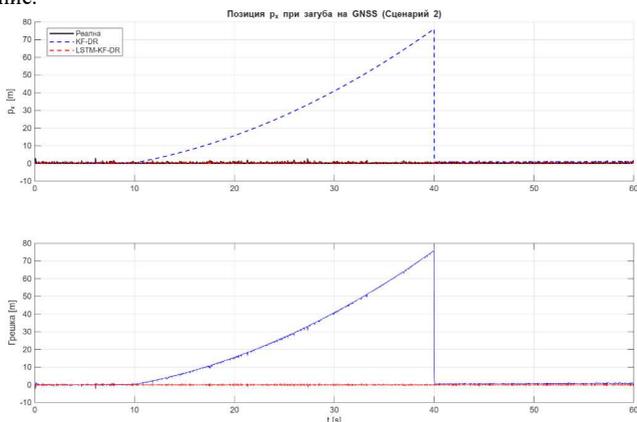
Въпреки това експериментът доказва функционалната съвместимост на невронния предиктор със структурата на EKF и създава предпоставки за последващи сценарии, в които моделът на динамиката е нарушен или GNSS измерванията отсъстват. Именно в тези условия се очаква LSTM-компонентът да прояви своето предимство, компенсирайки неопределеността в системния модел.

4.3. Сценарий 2: Загуба на GNSS / DR режим.

В този експеримент се моделира ситуация, при която бордовата интегрирана навигационна система губи GNSS наблюдения за определен времеви интервал ($t = 10-40$ s), като преминава в режим *Dead Reckoning* (DR). Целта е да се анализира способността на двата подхода — стандартен Калманов филтър (EKF) и невронно-подсилен LSTM-KF — да поддържат точност и стабилност на оценките при отсъствие на външни корекции.

GNSS измерванията, записани в масива `out.log_z`, са изкуствено занулени в посочения интервал, което симулира реална загуба на спътников сигнал. При EKF алгоритъмът се свежда до чисто прогнозиране на състоянието чрез динамичния модел f_{dyn} , докато при LSTM-KF предикцията се извършва от невронната мрежа `nn_predict_lstm`, обучена върху последователности от сензорни данни. Тази архитектура позволява използването на паметта на мрежата като заместител на липсващата GNSS информация.

На Фигура 31 е показана времевата еволюция на координатата p_x и съответната грешка за двата филтъра. Видно е, че EKF (цианова линия) показва нарастващо отклонение, което достига приблизително 80 m в края на загубата на GNSS сигнала, докато LSTM-KF (червена линия) запазва оценката практически съпадаща с реалната траектория. След възстановяване на спътниковите данни при $t > 40$ s, EKF бавно намалява грешката си, докато LSTM-KF вече е стабилизан около реалното състояние.



Фигура 31. Времени редове на позицията p_x и грешките по p_x при загуба на GNSS в интервала 10–40 s за EKF и LSTM-KF.

Количественото сравнение при пълна загуба на GNSS се извършва чрез корен средноквадратичната грешка (RMSE), изчислена поотделно за шестте компоненти на вектора на състоянието, както и чрез агрегирана метрика за целия вектор (Combined RMS). В Таблица 11 са представени резултатите за базовия филтър EKF/KF и за хибридният подход LSTM-KF.

Таблица 11. RMSE на EKF/KF и LSTM-KF при загуба на GNSS.

Състояние	RMSE KF-DR	RMSE LSTM-DR
1	26.69	0.209
2	53.09	0.396
3	79.62	0.632
4	3.33	0.136
5	6.67	0.148
6	10.02	0.270

Агрегираната грешка (Combined RMS) за целия вектор на състоянието ϵ :

$$\text{RMSE KF-DR} = 40.8783$$

$$\text{RMSE LSTM-DR} = 0.34478$$

Метриката е изчислена чрез квадратично усредняване на компонентните $RMSE_{model}$:

$$\text{Combined RMS} = \sqrt{\frac{1}{n} \sum RMSE_i^2}, \quad n = 6, \quad (4.2)$$

което поставя по-голяма тежест върху компонентите с по-големи отклонения.

Стойностите в Таблица 11 показват, че **хибридната архитектура (LSTM-KF)** постига по-ниски RMSE във всички шест състояния и редуцира Combined RMS спрямо класическата архитектура EKF/KF (40.878 \rightarrow 0.345). При продължителна липса на GNSS сигнал хибридният

филтър ограничава нарастването на грешката в сравнение с базовия подход, при който отсъствието на външна корекция води до натрупване на инерциални грешки и дрейф.

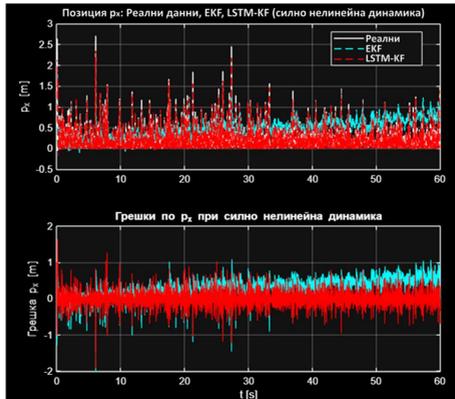
След възстановяване на GNSS сигнала, LSTM-KF се връща към номинален режим в рамките на един-два цикъла на обновяване, без изразени преходни отклонения в оценката. За сравнение, при базовия филтър се наблюдава по-дълъг преходен период до стабилизиране на оценката, свързан с повторно установяване на коректно претегляне между прогнозата и възстановените измервания.

4.4. Сценарий 3: Силно нелинейна динамика.

В сценарий 3 се изследва поведението на интегрираната навигационна система при силно нелинейна динамика на обекта спрямо опростения модел, използван в стандартния филтър на Калман. „Реалната“ динамика на БЛА се генерира от нелинейния блок f_dyn в модела `ins_baseline`, докато предикторът в EKF използва линеен дискретен модел `fdyn_linear_step`, реализиращ двойна интеграция на измерените ускорения. По този начин се създава умишлен модел-mismatch между реалната система и математическия модел в оценителя, което е типична ситуация при реални безпилотни платформи с сложна аеродинамика и неидеални сензори. Невронно-подсилената схема LSTM-KF използва същия EKF, но динамичният модел в прогнозната стъпка е заменен с обучен LSTM-предиктор (`nn_predict_lstm`), който апроксимира прехода $x_{k+1} = f(x_k, u_k)$ на база на последователности от нормализирани вектори $[x_k, u_k]$. В приложението към дисертационния труд са описани скриптовете с които е реализиран експеримента и са генерирани визуалните резултати.

Симулацията се изпълнява върху същите измервания от инерциалния модул и псевдоданните от GNSS, както в предходните сценарии. Времевият хоризонт е $T = 60$ s, с дискретизация $T_s = 0,01$ s. Коварнациите на процесния шум Q и на измервателния шум R , както и измервателната матрица H , съвпадат с конфигурацията в базовия EKF. Разликата е единствено в динамичния модел: при KF-NL се използва линейният оператор `fdyn_linear_step`, а при LSTM-KF-NL – невронният предиктор върху плъзгащ прозорец с дължина $L = 20$ такта.

На Фигура 32 е представена еволюцията на координатата p_x във времето и съответните грешки на двата филтъра.



Фигура 32. Еволюция на координатата p_x във времето и съответните грешки на двата филтъра.

В горния панел са представени референтната позиция (бяла линия), оценката на KF-NL (циан) и оценката на LSTM-KF-NL (червена линия). При разглеждания сценарий траекторията е силно нелинейна, което води до моделно несъответствие спрямо използваната линейна динамика; и двата метода възпроизвеждат общата тенденция на движението, като оценката на LSTM-KF-NL се наблюдава да остане по-близо до референтната крива. В долния панел на фигурата са изобразени грешките:

$$e_{p_x}^{KF}(k) = \hat{p}_{x,k}^{KF} - p_{x,k} \text{ и } e_{p_x}^{LSTM}(k) = \hat{p}_{x,k}^{LSTM} - p_{x,k}. \quad (4.3)$$

Наблюдава се, че при KF-NL грешката нараства във времето и проявява устойчива ненулева компонента (bias), докато при LSTM-KF-NL отклоненията остават ограничени в по-тесен интервал около нулата, а емпиричната средна стойност на грешката е близка до нула.

Таблица 12 представя корен средноквадратичната грешка (RMSE) по компоненти на състоянието при силно нелинейна динамика (резки маневри) за два оценяващи алгоритъма: стандартен Extended Kalman Filter (EKF) и предложен LSTM-усилен Калманов филтър (LSTM-KF). В настоящия раздел означението **RMSE_KF_NL** съответства на RMSE на **EKF** при силно нелинейна динамика, а **RMSE_LSTM_NL** — на RMSE на **LSTM-KF** при същите условия.

Таблица 12. RMSE по състояния при силно нелинейна динамика.

State	RMSE_KF_NL	RMSE_LSTM_NL
1	0.42443	0.19829
2	0.83446	0.36873
3	1.10370	0.57603
4	3.33130	0.13490
5	6.66500	0.14691
6	10.02300	0.26516

Агрегираната грешка на оценяване за шесткомпонентния вектор на състоянието е съответно:

$$RMSE_{KF-NL} = 5.1329, \quad RMSE_{LSTM-KF-NL} = 0.3207.$$

Тези стойности са получени чрез квадратично усредняване на компонентните $RMSE_{model}$ според израза (4.2), което поставя по-голяма тежест върху RMSE по компонентите с по-големи отклонения.

Сравнението между EKF и LSTM-KF не се ограничава до отделни компоненти на състоянието, а се базира на агрегирана оценка за целия вектор на състоянието върху целия времеви интервал на симулацията. Двата алгоритъма са оценени при идентични симулационни сценарии, начални условия, шумови характеристики и входни въздействия, което осигурява съпоставимост на резултатите. Получените стойности за $RMSE_{KF-NL}$ и $RMSE_{LSTM-KF-NL}$ показват съществено различие в интегралната грешка на оценяване при силно нелинейна динамика, като отношението между тях е приблизително $5.1329 / 0.3207 \approx 16$.

Наблюдаваното намаляване на агрегиращия показател при LSTM-KF е свързано с начина, по който двата подхода описват динамиката в прогнозната стъпка. При EKF, когато използваният процесен модел не отразява адекватно нелинейните ефекти при резки маневри, възниква структурно моделно несъответствие, което води до систематично изкривяване на априорната прогноза. При такива условия корекционната стъпка (чрез иновацията) може да редуцира част от грешката, но остава чувствителна към качеството на априорната оценка, което може да се прояви като натрупване на отклонения във времето и повишена грешка в целия вектор на състоянието.

При LSTM-KF, процесният модел в прогнозната стъпка е реализиран чрез обучаемата рекурентна невронна мрежа (LSTM), която е обучена да апроксимира ефективната нелинейна зависимост между предходните състояния и входните въздействия в разглеждания режим на движение. Това води до по-точна априорна прогноза и позволява корекционният етап на Калмановата филтрация да работи при по-малки остатъчни несъответствия, вместо да компенсира груби моделни грешки. В резултат грешката остава по-ограничена във времето и агрегиращият показател (Combined RMS) е съществено по-нисък.

По-ниската обща RMSE при LSTM-KF-NL се интерпретира като индикатор за по-добро съответствие между използвания динамичен модел и реалната нелинейна динамика в разглеждания експеримент, което води до по-стабилно оценяване при резки маневри и наличие на моделно несъответствие.

Анализът във времевата област (напр. по траектории и грешки във времето) е в съответствие с количествените резултати: при EKF с процесен модел, който не описва достатъчно точно нелинейния режим, се наблюдава тенденция към дрейф и нарастващи дългосрочни отклонения, докато при LSTM-KF грешките остават ограничени и без ясно изразена систематична компонента. Това подкрепя хипотезата на изследването, че при интегрирана навигационна система и силно нелинейни режими, невронно подпомогнатото моделиране на процесната динамика може да подобри точността и устойчивостта на рекурсивното оценяване спрямо стандартен EKF с аналитичен процесен модел.

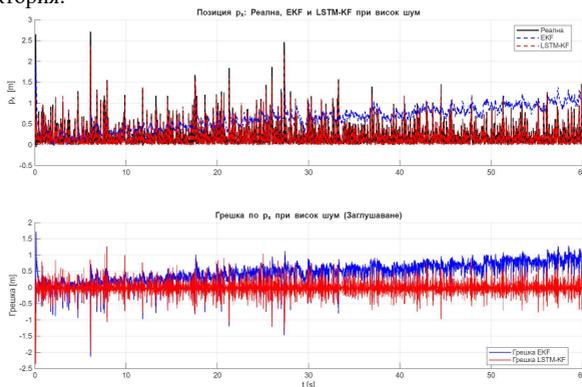
4.5. Сценарий 4: Висок шум / несигурност на измерванията.

В четвъртия експериментален сценарий се оценява поведението на стандартния Extended Kalman Filter (EKF) и предложения невронно-подсилен филтър LSTM-KF при повишен шум и несигурност на измерванията. Целта е да се анализира устойчивостта на двата алгоритъма при деградирало качество на сензорните данни, което е типично за реални условия като повишен шум в GNSS и IMU каналите, електромагнитни смущения и/или вибрации на платформата.

Симуляционната постановка използва същия модел *ins_baseline* и идентични входни въздействия (управление и профил на движение), като разликата спрямо предходните сценарии е в параметрите на шумовите ковариации. Ковариацията на процесния шум Q е увеличена петкратно, ковариацията на измервателния шум R - десетократно, а началната ковариация на грешката P_0 - петдесеткратно, с цел моделиране на по-голяма неопределеност в началните условия. По този начин двата оценителя се поставят в режим на висока неопределеност както по отношение на динамиката, така и по отношение на измерванията. В приложението към дисертационния труд са приведени използваните скриптове за експеримента и визуализациите на резултатите.

EKF запазва стандартната си структура с аналитичен динамичен модел f_{dyn} и линеен измервателен модел с матрица $H = [I_3 \ 0_{3 \times 3}]$, като LSTM-KF използва същия алгоритъм на рекурсивно обновяване, но реализира прогнозната стъпка чрез обучена рекурентна невронна мрежа (LSTM), имплементирана чрез функцията *nn_predict_lstm*. Двата филтъра работят при еднакви начални условия и използват идентични шумови параметри (Q, R, P_0) в рамките на сценария.

На Фигура 33 е показана еволюцията на координатата p_x във времето за двата филтъра и референтната траектория.



Фигура 33. Зависимост на координатата p_x във времето при висок шум при двата филтъра и референтната траектория.

В горния панел се наблюдава, че при повишен шум оценката на EKF (цианова линия) проявява по-големи флуктуации около референтната стойност, докато оценката на LSTM-KF (червена линия) остава по-близо до референтната траектория (бяла линия). В долния панел са показани грешките на оценките спрямо референтната стойност, като при EKF се наблюдава по-голяма вариабилност на грешката, докато при LSTM-KF отклоненията остават по-ограничени около нулата.

Таблица 13 представя количествените резултати за корен средноквадратичната грешка (RMSE) по компонентите на вектора на състоянието (позиционни компоненти - States 1-3 и скоростни компоненти - States 4-6).

Таблица 13. RMSE по състояния при висок шум в измерванията.

State	RMSE KF HN	RMSE LSTM HN
1	0.55893	0.19904
2	1.1077	0.37673

3	1.4398	0.58926
4	3.3313	0.13481
5	6.6650	0.14670
6	10.023	0.26482

$$\text{RMSE}_{\text{KF-HN}} = 5.1575, \quad \text{RMSE}_{\text{LSTM-KF-HN}} = 0.3262$$

Агрегираната оценка на точността за целия шесткомпонентен вектор на състоянието е изчислена чрез квадратично усредняване (Combined RMS) на компонентните RMSE (4.2). Използването на квадратично усредняване (вместо аритметично усредняване) акцентира върху компонентите с по-големи отклонения и позволява по-адекватна оценка на поведението при сценарии с висок шум.

Стойностите в Таблица 13 показват, че при значително увеличен шум EKF демонстрира съществено влошаване на точността по скоростните компоненти, като максималната стойност достига $\text{RMSE}=10.023 \text{ m/s}$ (State 6). За сравнение, LSTM-KF ограничава скоростните грешки под 0.3 m/s за States 4-6 ($0.13481 - 0.26482 \text{ m/s}$) и поддържа позиционните грешки в субметров диапазон за States 1-3 ($0.19904 - 0.58926 \text{ m}$).

Спрямо агрегиращия показател за целия вектор на състоянието, EKF достига $\text{RMSE} \approx 5.16$, докато LSTM-KF постига $\text{RMSE} \approx 0.33$, което съответства на приблизително $5.1575 / 0.3262 \approx 16$ -кратно намаляване на агрегираната грешка (Combined RMS) в условия на деградирано качество на измерванията.

Получените резултати са в съответствие с хибридната архитектура, при която прогнозната стъпка се подпомага от обучаем модел, приближаващ ефективната динамика в шумна среда. Поточната априорна прогноза намалява чувствителността на рекурсивната оценка към шумни измервания в корекционната стъпка и води до по-ниска обща грешка при високи нива на несигурност в GNSS/IMU каналите.

Сценарий 4 потвърждава, че в разглежданата постановка LSTM-KF демонстрира по-висока устойчивост към шум и неопределеност на измерванията спрямо стандартния EKF, както по компонентните RMSE, така и по агрегиращия показател за целия вектор на състоянието.

4.6. Количествен анализ

Целта на анализа е да се извърши статистическа и сравнителна оценка на EKF и LSTM-KF по четири метрики: (i) агрегирана корен средноквадратична грешка за вектора на състоянието (Combined RMS), (ii) RMSE за скоростните компоненти, и (iii) скорост на конвергенция. Метриците за ориентация не са включени в настоящата постановка поради липса на ориентационни състояния/измервания в модела.

4.6.1. Агрегирана грешка за вектора на състоянието (Combined RMS)

Първият етап от анализа обобщава агрегиращия показател **Combined RMS** за целия вектор на състоянието при всички сценарии (табл. 14). Метриката се изчислява чрез квадратично усредняване на компонентните RMSE (4.2). Този показател акцентира върху компонентите с по-големи отклонения и предоставя интегрална оценка на точността във времето.

Промяната на LSTM-KF спрямо EKF се изчислява с израз:

$$I[x] = \frac{\text{CombinedRMS}_{\text{(EKF)}}}{\text{CombinedRMS}_{\text{(LSTM-KF)}}}. \quad (4.4)$$

Таблица 14. Усреднени RMSE за позиция по сценарии.

Сценарий	Combined RMS (EKF)	Combined RMS (LSTM-KF)	Отношение на агрегираната грешка (I) на LSTM-KF спрямо EKF
1. Нормални условия	0.09	0.32	0.28
2. Загуба на GNSS (DR)	40.88	0.35	116.80
3. Силно нелинейна динамика	5.13	0.32	16.03
4. Висок шум	5.16	0.33	15.64

За сценарий 1 стойността е положителна в смисъл на **увеличение на грешката** при LSTM-KF (влошаване), докато за сценарии 2–4 стойността представлява **редукция** на грешката.

Данните показват, че при номинални условия (Сценарий 1) **EKF** постига по-ниска *Combined RMS* спрямо **LSTM-KF** (0.09 срещу 0.32). Това е съвместимо с факта, че при добре описана динамика и блиски до гаусови шумове класическите филтри са конкурентни, а добавянето на обучаем модел не е необходимо условие за подобрене. При критичните сценарии обаче (загуба на GNSS, силна нелинейност и висок шум) **LSTM-KF** постига по-ниска *Combined RMS* спрямо **EKF**, като при загуба на GNSS се наблюдава най-голям ефект (40.88 → 0.35; ≈ 116.8).

4.6.2. Грешка по скоростните компоненти

При вторият етап от проведения експеримент се анализира RMSE по скоростните компоненти v_x, v_y, v_z . В проведените експерименти при EKF се наблюдава повишена грешка в сценарии с нелинейна динамика и шумови смущения, като при неблагоприятни условия RMSE по скорост достига порядъка 5–10 m/s. За сравнение, при LSTM-KF скоростните грешки остават значително по-ниски (типично под 0.3 m/s в най-тежките сценарии), което е в съответствие с точна априорна прогноза в прогнозната стъпка и по-ниска чувствителност към шумни измервания.

Агрегираната скоростна грешка може да се дефинира като:

$$Combined\ RMS_v = \sqrt{\frac{1}{3} \sum_{i=4}^6 RMSE_i^2}, \quad (4.5)$$

където $Combined\ RMS_v$ са RMSE по компонентите v_x, v_y, v_z .

4.7. Сравнение с модели от литературата.

Предложената в настоящия труд архитектура LSTM-KF се вписва в съвременната линия на **хибридни state-space оценители**, при които се запазва рекурсивната predict-update структура на Калмановата филтрация, а част от модела се допълва/параметризира чрез обучаеми функции (напр. невронни мрежи). Подобни подходи се разглеждат както в рамките на model-based deep learning за оценяване на състоянието, така и в рамките на neural state-space модели, където динамиката f_θ се учи от данни (Krishnan, R. G., Shalit, U., & Sontag, D., 2015); (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022)

4.7.3. LSTM при GNSS-отпадане и счисление (dead reckoning): пряко релевантни резултати.

Най-пряко релевантни към сценарии с GNSS-деградация са публикации, в които LSTM се използва за **поддържане на навигационното решение по време на GNSS-прекъсване**, т.е. за ограничаване на грешката при режим на „счисление“. Fang et al. (2020) предлагат LSTM-базиран подход за генериране на псевдо GNSS позиционни инкременти при отпадане на GNSS, които подпомагат INS/KF корекцията (Fang, W., Jiang, J., Lu, S., Gong, Y., Tao, Y., Tang, Y., Yan, P., Luo, H., & Liu, J., 2020).

Тази линия е съвместима с интерпретацията на резултатите от глава 4: при неблагоприятни режими (GNSS-отпадане, висока шумова деградация, силна нелинейност) обучаемият компонент може да намали чувствителността към model mismatch и да ограничи акумулирането на грешка. Важно уточнение (и методично коректно заключение) е, че при номинални условия и добре описан модел, класическият EKF може да остане конкурентен спрямо хибридните модели, което е наблюдавано и в нашите експерименти (Сценарий 1) (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022); (Fang, W., Jiang, J., Lu, S., Gong, Y., Tao, Y., Tang, Y., Yan, P., Luo, H., & Liu, J., 2020).

4.7.4. Обобщение.

В обобщение, LSTM-KF може да бъде позициониран като хибриден state-space оценител, който е методологично съпоставим с (i) model-based deep learning подходи за филтрация (напр. KalmanNet) и (ii) neural state-space модели (напр. Deep Kalman Filters), както и с (iii) LSTM-

подпомогнати INS/GNSS методи при GNSS-отпадания (Revach, G., Shlezinger, N., Ni, X., López Escoriza, A., van Sloun, R. J. G., & Eldar, Y. C., 2022); (Krishnan, R. G., Shalit, U., & Sontag, D., 2015); (Fang, W., Jiang, J., Lu, S., Gong, Y., Tao, Y., Tang, Y., Yan, P., Luo, H., & Liu, J., 2020).

С оглед на собствените резултати от глава 4, коректната интерпретация е следната: **LSTM-KF** показва предимство спрямо базов EKF в неблагоприятни режими (GNSS-прекъсване/висок шум/силна нелинейност), докато при номинален режим EKF може да демонстрира по-ниска агрегирана грешка. Следователно приносът следва да се формулира като повишена устойчивост при деградирани условия, а не като универсално превъзходство при всички сценарии.

4.8. ИЗВОДИ ОТ ГЛАВА 4.

1. Резултатите от проведените симулационни експерименти сочат, че в неблагоприятни режими на работа (загуба на GNSS/режим dead reckoning, повишен IMU шум, нестационарна и силно нелинейна динамика и бързи динамични промени) LSTM-KF постига по-ниски стойности на агрегирания показател **Combined RMS** и на **RMSE за позиция и скорост** спрямо базовия EKF, като едновременно достига зададения критерий за установяване на грешката за по-кратко време и демонстрира устойчиво поведение; при номинален режим (нормални условия) EKF постига по-ниска агрегирана грешка спрямо LSTM-KF, което е съгласувано с представените резултати по сценарии.
2. Анализът на иновационните последователности показва намалена систематична компонента на остатъците и съгласуване между прогноза и измервания по използваните показатели, което е индикатор за по-адекватно процесно моделиране чрез LSTM в разгледаната постановка.
3. При продължителни режими на счисление (dead reckoning) LSTM-KF ограничава натрупването на грешката спрямо EKF по агрегирания показател Combined RMS и по компонентните RMSE в рамките на разгледания експеримент.
4. В рамките на използваните критерии за оценка се наблюдава повишаване на устойчивостта на навигационната оценка при деградирани условия (GNSS загуби/повишен шум), което потвърждава изследователската хипотеза за полезността на LSTM-подпомогнат процесен модел в Калманова структура.
5. Предложената архитектура представлява основа за последваща интеграция и валидация в реални условия, включително Hardware-in-the-loop (HIL) и полети тестове, както и разработване на по-сложни сценарии и сензорни конфигурации.

IV. ЗАКЛЮЧЕНИЕ.

Настоящият дисертационен труд изследва възможността за конструиране на интегрирана навигационна система за безпилотни летателни апарати (БЛА), при която процесният (преходният) модел в рамката на Калмановата филтрация се параметризира чрез обучаемата рекурентна невронна мрежа от тип LSTM. Изследването обхваща литературен анализ, разработване на симулационен модел в MATLAB/Simulink, реализация на хибридна архитектура LSTM-KF и провеждане на експериментални симулационни сценарии с различни шумови профили, смущения и режими на частична или прекъсваща достъпност на GNSS измерванията. Резултатите в разгледаните условия показват, че подходът е технологично осъществим и може да предостави измерими подобрения спрямо базовия EKF при неблагоприятни режими на работа, характеризирани се с нестационарни шумове и деградирани измервания.

Обобщените резултати от експериментите показват, че разработената хибридна архитектура с използване на невронна мрежа постига по-ниски стойности на използваните метрики за отчитане

на грешките при оценяване на позиция и скорост при деградиращи или прекъсващи GNSS наблюдения и при режим на счисление (dead reckoning), достига по-бързо (фиг. 31) установен режим по избрания критерий и демонстрира устойчиво поведение при повишен IMU шум и нестационарна динамика спрямо класическата архитектура. Същевременно, при номинален режим (нормални условия на измервания) базовият EKF постига по-ниска агрегирана грешка (табл. 14) спрямо LSTM-KF, което потвърждава сценарий-зависим характер на наблюдаваните предимства. Невронният компонент подпомага процесното моделиране чрез апроксимация на нелинейни и режимно-зависими зависимости, които са трудно описуеми с ръчно зададени аналитични модели при реални ограничения. Това се проявява като подобро поведение на оценяването в сценарии с GNSS деградация, повишен шум и силно нелинейна динамика.

Хибридният навигационен модел, при който рекурентна невронна мрежа (RNN) от тип LSTM (Long Short-Term Memory) не изпълнява единствено коригираща функция, а участва като ключов елемент в процесния модел на филтъра чрез обучение върху сензорни данни и симулационни траектории. Предложеният подход съчетава структурирания механизъм за оценяване и интегриране на измерванията в Калмановата рамка с моделиране на динамиката, базирано на данни, чрез дълбока рекурентна архитектура. Постигнатите резултати показват потенциал за използване на обучаеми процесни модели при оценяване на състоянието в реално време, при условие на последващо профилиране, оптимизация и валидация върху целеви хардуер.

На базата на извършения анализ и проведените експерименти могат да се формулират следните насоки за бъдещи изследвания: (i) включване на обучаеми компоненти в режими с вариращи шумови характеристики и непълни знания за динамиката при осигурен контрол на обобщаването и оценка/калибриране на неопределеността; (ii) систематично сравнение на различни рекурентни структури (LSTM и GRU), както и на алтернативни невронни модели в пространството на състоянията, включително Neural ODE / Neural SDE (невронни диференциални / стохастични диференциални модели), по показатели за точност, устойчивост и изчислителна сложност; (iii) разширяване към мултимодална интеграция на сензорни данни (визуални, LiDAR и Ultra-Wideband – UWB), с цел ограничаване на грешките при продължителни режими на счисление и при деградирани GNSS условия.

Възможностите за развитие включват обучение и валидация върху реални полетни данни, добавяне на механизми за онлайн адаптация и изпитване в по-реалистична експериментална среда (напр. hardware-in-the-loop – HIL и полеви тестове). Допълнително, интеграцията с методи на обучение с утвърждаване може да бъде разгледана в контекста на съвместно оптимизиране на навигация и управление, така че да се постигне адаптация на контролната политика при променящи се условия на средата и динамиката.

Приложимостта на предложения подход е потенциално широка и включва автономна навигация на БЛА, роботизирани системи и автономни платформи, при които надеждното оценяване на състоянието е критично. Разработената архитектура може да бъде адаптирана за внедряване при ограничени изчислителни ресурси при условие на профилиране и оптимизация, тъй като LSTM компонентът допуска компактна реализация, а филтрационната рамка поддържа ефективна онлайн обработка. Получените резултати подкрепят извода, че комбинирането на аналитично моделиране и невронни архитектури е перспективна посока за създаване на по-надеждни и адаптивни навигационни системи за БЛА, особено при нелинейности, нестационарни шумови режими и деградирани наблюдения.

ОБЩИ ИЗВОДИ ОТ ДИСЕРТАЦИОННИЯ ТРУД

1. Анализът на навигационните подходи показва, че решения, базирани на единичен сензорен канал, проявяват структурни ограничения при деградирани или прекъсващи наблюдения, поради което интегрираните архитектури за комбиниране на инерциални и външни измервания са методологично обоснован избор за устойчивост в широк диапазон от режими на полет.

2. При инерциално счисление доминиращ фактор за коректната работа на модела са характеристиките на IMU – чувствителност спрямо систематични отклонения, температурна

нестабилност, стохастични компоненти, които водят до натрупване на навигационна грешка при липса на външни корекции и предполагат използването на корекционни канали и адаптивни модели.

3. Параметризирането на процесния модел в прогнозната стъпка на Калмановата схема чрез обучаем предиктор е концептуално обосновано и инженерно реализируемо - LSTM-базиран модел може да изпълнява ролята на процесен блок при запазване на рекурсивната структура predict-update и корекционната стъпка на филтъра.

4. Симулационният модел показва, че коректната интеграция на невронната мрежа в априорната прогноза е критична за коректността на резултатите и за сравнителния анализ на стандартния аналитичен и предложения адаптивен LSTM процесен модел при идентични входни условия.

5. Резултатите от сравнителната оценка между EKF и хибридната конфигурация LSTM-KF (Калманова схема, в която процесният модел в прогнозната стъпка се реализира от LSTM предиктор) характеризират сценарий-зависимо поведение: при номинални условия EKF постига по-ниска стойност на агрегирания показател за грешка (Combined RMS) спрямо LSTM-KF, докато при неблагоприятни режими (прекъсване на GNSS/режим dead reckoning, силна нелинейност и повишен шум) LSTM-KF дава по-ниски стойности на използваните метрики за оценка на грешката и достига по-бързо установен режим по избрания критерий за установяване спрямо EKF (фиг. 31, таблица 14, израз 4.2).

6. Анализът на иновационните последователности предоставя допълнителна диагностична информация за съгласуването между прогноза и измервания, като наблюдаваното ограничаване на систематичната компонента на остатъците е съвместимо с адекватно процесно моделиране чрез обучаемия компонент в разгледания модел.

НАСОКИ ЗА БЪДЕЩА РАБОТА

1. Разширяване на експерименталната валидация към по-широк набор от траектории и режими, включително по-сложни маневри, продължителни GNSS прекъсвания и разнообразни шумови профили, с цел оценка на обобщаването и устойчивостта на предложената архитектура.

2. Включване на ориентационни състояния и допълнителни измервания (напр. магнитометър, барометър, визуални/лидарни наблюдения), както и разширяване към мултисензорни конфигурации, за да се оцени ефектът на обучаемия процесен модел при по-пълна навигационна постановка.

3. Изследване на варианти за адаптация на неопределеността и робастност (напр. адаптивно оценяване на шумови ковариации и детекция на атипични измервания), както и анализ на статистическата консистентност чрез стандартни критерии, когато се включат ковариационни оценки и проверими доверителни граници.

4. Преминаване към по-реалистични експериментални условия чрез hardware-in-the-loop (HIL) и полеви тестове, включително профилиране и оптимизация за целеви хардуер, за да се оцени приложимостта на архитектурата при онлайн обработка и ресурсни ограничения.

V. ПРИНОСИ.

Научни приноси

1. Разработена е **хибридна навигационна система за интегрирана INS/GNSS навигация на БЛА**, при която процесният (преходният) модел в прогнозната стъпка на Калмановата схема (predict-update) се параметризира чрез рекурентна невронна мрежа от тип LSTM, а корекционната стъпка се реализира от корекционния модул на филтъра. Системата е концептуално обоснована в условия на нелинейна динамика и нестационарни шумове и е реализирана и оценена чрез симулационни сценарии.

Научно-приложни приноси

1. Дефинирани са условията на приложимост на хибридната INS/GNSS навигационна конфигурация LSTM–KF (Калманова схема, в която процесният модел в прогнозната стъпка се реализира от **рекурентна невронна мрежа от тип LSTM**) спрямо базов EKF: при деградирани или прекъсващи GNSS наблюдения, повишен IMU шум и силно нелинейна динамика LSTM–KF постига по-ниски стойности на използваните показатели за грешка за позиция и линейна скорост, докато при номинален режим EKF дава по-ниска стойност на агрегирания показател за грешка.
2. **Разработен е симулационен модел в развойната програмна среда MATLAB/Simulink** на интегрирана навигационна архитектура, който поддържа паралелни реализации на процесния модел – аналитичен f_{dyn} и процесен предиктор, реализиран с рекурентна невронна мрежа от тип LSTM – със съгласувани входно-изходни интерфейси, унифицирана времева база и управляем избор на процесен модел.
3. **Реализиран е възпроизводим работен поток за изграждане на хибридният навигационен модел (данни–обучение–интеграция)**, включващ автоматизирано извличане на симулационни данни, формиране на последователности за обучение, нормализация и съхраняване на параметрите за мащабиране, обучение на **рекурентна невронна мрежа от тип LSTM** и внедряване на обучената мрежа в Simulink чрез функционален блок за едностъпкова прогноза, съвместим с дискретната схема на филтъра.
4. **Введена е методология за експериментална оценка на хибридната INS/GNSS навигационна архитектура**, чрез агрегирана метрика (Combined RMS), компонентни RMSE за състоянието и критерий за установяване на грешката (напр. праг 5%), позволяващи сравнимост на резултатите при идентични входни условия и различни сценарии на GNSS достъпност и шумови профили.
5. **Реализирана е последваща обработка на резултатите от симулацията за статистическа оценка на прогнозата в хибридният навигационен модел**, чрез сравнение на аналитичния процесен модел f_{dyn} и LSTM-базиран процесен предиктор (**рекурентна невронна мрежа от тип LSTM**) спрямо референтното състояние, с изчисляване и таблично представяне на RMSE по компоненти (позиция и линейна скорост).
6. **Представен е диагностичен анализ на интеграцията на LSTM-базирания процесен предиктор** (рекурентна невронна мрежа от тип LSTM) в прогнозната стъпка на хибридната навигационна схема, включително разграничаване между ефектите на обучаемия процесен модел и ефектите на конфигурацията на симулационната интеграция, с цел коректно интерпретиране на резултатите и оптимизация на параметрите на системата.

СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИЯТА:

- [1] Генов, В. (2024). Съвременни технологии за изкуствен интелект при управлението на безпилотни летателни апарати. *Defense Technology Forum DTF24 (DefTech 2024)*, (стр. 668-674). Шумен
- [2] Генов, В. (2025). Съвременни технологии за интегрирани навигационни системи за безпилотни летателни апарати. *Defense Technology Forum DTF25 (DefTech 2025)*, (стр. 691-700). Шумен
- [3] Генов, В. (2025). Приложения на невронни мрежи при работата на Калманов филтър. *Defense Technology Forum DTF25 (DefTech 2025)*, (стр. 701-713). Шумен
- [4] Генов, В. (2025). Симулиране на интегрирана навигационна система за управление на безпилотен летателен апарат в Matlab. *Defense Technology Forum DTF25 (DefTech 2025)*, (стр. 755-766). Шумен



VASIL LEVSKI NATIONAL MILITARY UNIVERSITY
FACULTY OF ARTILLERY, AIR DEFENSE AND CIS

9700 Shumen, 1 Karel Shkorpil St. Phone: (054) 801 040; Fax: (054) 877 463
telephone: (054) 801 040; fax: (054) 877 463;
Email: decanat@aadcf.nvu.bg

Assist. Prof. Eng. Vladislav Milkov Genoff

**INTEGRATED NAVIGATION SYSTEM
FOR THE CONTROL OF UNMANNED AERIAL VEHICLES
USING ARTIFICIAL INTELLIGENCE**

THESIS ABSTRACT

for the award of the educational and scientific degree “Doctor” (PhD)

Doctoral programme in the scientific specialty:

"Automated Systems for Information Processing and Control"

SCIENTIFIC FIELD: 5. TECHNICAL SCIENCES

PROFESSIONAL FIELD 5.3. COMMUNICATION AND COMPUTER TECHNOLOGY

Scientific supervisor:

Assoc. Prof. Eng. Krasimir Ognyanov Slavyanov, PhD

Scientific consultant:

Senior Assistant Prof. Eng. Ivan Atanasov Popov, PhD

SHUMEN

2026

The dissertation consists of 202 pages
Number of tables – 22
Number of figures – 34
Number of appendices – 2
Number of references – 96
Number of publications related to the dissertation – 4

The defense of the dissertation will take place on __.__.2026 at __.__. a.m. in _____ at _____.

The materials related to the defense are available to interested parties at _____ in _____, tel. _____.

The dissertation was discussed at a departmental council meeting of the Department of Computer Systems and Technologies at the Faculty of Artillery, Air Defense, and CIS at the Vasil Levski National Military University – Veliko Tarnovo on __.__.20... , and is scheduled for defense before a scientific jury for the acquisition of the educational and scientific degree "Doctor" (PhD).

The doctoral student is an assistant at the Department of Computer Systems and Technologies at the Faculty of Artillery, Air Defense and CIS at the National Military University "Vasil Levski" – Veliko Tarnovo. The main research for the dissertation was conducted in the MATLAB/Simulink development environment.

Author: Assist. Prof. Eng. Vladislav Milkov Genoff

Topic: "Integrated navigation system for controlling unmanned aerial vehicles using artificial intelligence."

Circulation __

Printed on __.__. 20__

Publishing complex of the Vasil Levski National Military University

I. GENERAL CHARACTERISTICS OF THE DISSERTATION.

Reasons for developing the dissertation

The global development of autonomous unmanned aerial vehicles (UAVs) is characterized by a clear trend towards a transition from isolated navigation algorithms based on highly idealized analytical assumptions to integrated navigation systems and multisensor state estimation. In this context, machine learning (ML) and deep learning methods are increasingly used to interpret sensor streams, model nonlinear dependencies, and support process modeling in complex system behavior. Review publications in the field of deep learning applications in UAVs emphasize the growing role of data and sequential real-time processing in navigation and related tasks (Carrio, Sampedro, Rodriguez-Ramos, & Campoy, 2017).

On this basis, the research line "Kalman filtering + neural networks" is establishing itself as a practical approach for combining structured probabilistic estimation (through the standard *prediction-update* scheme) with learnable models of dynamics in partially known or inaccurately specified processes. A representative example of this direction are the neural-assisted Kalman methods, in which the state-space structure of the estimator is preserved, and part of the operations or models are learned from data (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022).

Relevance of the problem

In a number of practical applications with increased security and reliability requirements, constant monitoring of airspace and control of access to strategically important sites is necessary. Under these conditions, UAVs are often operated in environments with degraded or deliberately compromised GNSS signals, including through jamming and spoofing. As a result, robust navigation in GNSS-denied/compromised environments have become a significant engineering and scientific challenge (Elmas, Elif Ece; Alkan, Mustafa, 2024).

At the same time, the integration of inertial navigation systems (INS) and global navigation satellite systems (GNSS) remains a basic approach to UAV navigation, as it combines the autonomy of INS with the long-term stability of GNSS. However, accuracy in **dead reckoning** modes is limited by the accumulation of errors caused by the parameters and temperature stability of the inertial measurement unit (IMU), including MEMS IMUs, which makes the system sensitive in the absence of external corrections.

Purpose and objectives of the dissertation

The main objective of the dissertation is to develop and experimentally verify an integrated navigation system for UAVs, in which the analytical dynamic model within the framework of Kalman filtering is supplemented or replaced by a recurrent neural network with long short-term memory (LSTM). The aim is to improve the accuracy and stability of estimation in nonlinear dynamics, changing noise characteristics, and GNSS signal degradation.

The main tasks arising from the set goal are:

1. To perform a comparative analysis of modern navigation approaches and methods for integrating sensor data in UAVs, with an emphasis on the limitations of KF, EKF, and UKF in the case of nonlinearity and non-stationary noise.
2. To formalize the settings for integrated navigation and define criteria for assessing accuracy, stability, and consistency in the presence of degraded and interrupted GNSS.
3. Create simulation scenarios (including GNSS losses, noise profiles, and interference) and training and validation datasets.
4. Perform a comparative analysis of existing neural networks and make an informed choice of a suitable neural architecture as a process model (or as a component of the , supporting process modeling/parameter adaptation) in the structure of the filtering algorithm.
5. Develop a simulation experiment in MATLAB/Simulink to conduct numerical experiments and comparative evaluation against EKF/KF and selected hybrid approaches.

Subject of the dissertation

The subject of the study is the algorithms and methods for estimating the navigation state (position, velocity, and orientation) of UAVs by integrating sensor data from INS/GNSS, as well as the inclusion of trainable recurrent neural models to improve process modeling in the case of nonlinearities, mode-dependent behavior, and degraded measurements.

Means of achieving the objective

To achieve the objective, methods of statistical estimation and filtering of random processes, analysis and modeling of discrete-time dynamic systems, as well as artificial intelligence (AI) methods for processing sequential data and approximating nonlinear dependencies are used. The experimental part is implemented in MATLAB/Simulink, including tools for simulation modeling, training/validation of neural networks, and analysis of metrics and innovation sequences.

Initial conditions and assumptions

The results were obtained in a simulation environment and are subject to further verification under more realistic conditions, including parameter variations, different motion profiles, and discrepancies between simulation and real data. The evaluated state in the considered setting is limited to position and linear velocity; orientation states and corresponding metrics are not included in the experimental part, therefore the conclusions should be interpreted within this scope. The quantitative comparisons presented are formulated with respect to a basic Kalman filter - Extended Kalman Filter (EKF) within the scenarios considered. An extended comparison with additional filters and hybrid schemes requires a separate experimental study.

Scientific novelty

The scientific novelty consists of a proposed hybrid navigation estimation framework, in which the Kalman framework is retained as a structured probabilistic scheme for integrating measurements and estimating the state, while the process (transitional) model or key elements of the prediction step are implemented through a trainable LSTM architecture. This aims to achieve more adequate process modeling in nonlinear modes and changing noise characteristics, including GNSS degradation. The approach is conceptually compatible with current trends in neural-assisted Kalman filtering (e.g., KalmanNet and recurrent extensions), but focuses on applicability in UAV INS/GNSS navigation and experimental verification in a simulation environment (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022).

Practical utility

The practical utility of the development is related to the construction of a simulation-oriented software environment for assessing navigation accuracy and stability in UAVs. The system combines classical filter blocks (KF/EKF/UKF) with a trainable LSTM model to support process modeling in scenarios with noise, GNSS degradation, and interruptions. The components are organized in a modular way (scenario and data generation; measurement filtering and integration; neural model training and validation; metric comparison and visualization), which allows for step-by-step upgrading and quick verification of solutions for different mission profiles and sensor characteristics.

II. STRUCTURE OF THE DISSERTATION.

The dissertation analyzes methods from the theory of navigation systems and the field of AI, with an emphasis on recurrent neural architectures for modeling sequential data and their integration into state estimation tasks. On this basis, a hybrid LSTM-based navigation model has been developed, in which LSTM is used as a process model (or a key element of it) within the framework of Kalman filtering, and a comparative analysis has been performed in simulation scenarios with noise, GNSS degradation, and losses.

- **The preface** outlines the motives for the development, the relevance of the problem, the objective, the main tasks, the subject and means, the scientific novelty and practical usefulness of the development.
- **The first chapter** presents an overview of navigation systems for UAVs and approaches for INS/GNSS integration and complementary sensors, as well as the theoretical basis of Kalman filtering and its limitations in realistic conditions.
- **The second chapter** analyzes the development of AI approaches for modeling and controlling dynamic systems, including RNN/LSTM/GRU, Neural Ordinary Differential Equations (Neural ODE) and Stochastic Differential Equations (SDE), and neural-assisted Kalman methods.
- **The third chapter** defines the proposed solution and the architecture of the hybrid model, including the setup for supplementing/replacing the process model with LSTM, the organization of input-output values, training procedures, and evaluation criteria.

- **The fourth chapter** describes the experimental environment and scenarios in MATLAB/Simulink, presents the results of the numerical experiments, and compares the accuracy and stability in GNSS losses with EKF/UKF and selected hybrid approaches, formulating conclusions about the applicability and limitations of the solution.

III. BRIEF SUMMARY OF THE DISSERTATION.

CHAPTER 1.

OVERVIEW OF CURRENT TECHNOLOGIES FOR INTEGRATED NAVIGATION SYSTEMS, UNMANNED AERIAL VEHICLES, AND ARTIFICIAL INTELLIGENCE.

1.1. Classification of navigation systems for unmanned aerial vehicles.

In contemporary specialized literature, navigation systems for unmanned aerial vehicles (UAVs) are most often considered as architectures for assessing the navigation state, combining sensor measurements and data processing algorithms. Their main purpose is to reliably determine position, velocity, and orientation in a dynamic environment characterized by noise, interference, and potentially hostile influences.

Navigation solutions can be classified according to various criteria, the most widely used being: (1) primary information source, (2) level of integration between subsystems, and (3) degree of autonomy from external infrastructures. Within the scope of this section, a classification based on the dominant source of navigation information is adopted, in which systems are conditionally divided into: inertial navigation systems (INS), global navigation satellite system (GNSS)-based systems, and integrated (hybrid) navigation systems. Integrated solutions combine multiple data sources (e.g., INS/GNSS, camera, LiDAR, barometric and magnetometric measurements, map or radio-based information) through optimal estimation algorithms, with Kalman filtering variants (EKF/UKF) and optimization approaches (e.g., factor graphs) being widely used depending on the accuracy and computational resource requirements.

There is a clear trend towards a transition from single sensor modules to integrated navigation solutions, as individual subsystems generally have fundamental limitations in real-world operation. This is justified by the accumulation of error (drift) over time in INS and the vulnerability of GNSS to interference, signal loss, and deliberate actions such as jamming and spoofing (Mahmoud & Trilaksono, 2018). Therefore, the transition to hybrid architectures can be seen as a technologically justified necessity, especially in scenarios with limited or compromised GNSS availability. To justify this transition more fully, it is necessary to first analyze the principles and limitations of inertial navigation.

This thesis is also confirmed in the studies by Mahmoud and Trilaksono (Mahmoud & Trilaksono, 2018), which emphasize the fundamental limitations of individual subsystems: the accumulation of errors over time in INS and the vulnerability of GNSS to interference, signal loss, or deliberate attacks such as jamming and spoofing (Mahmoud & Trilaksono, 2018). Therefore, the logical transition to hybrid navigation architectures is a technological necessity. However, in order to fully justify this transition, it is necessary to first analyze the principles and limitations of inertial navigation.

1.2. Kalman filters – theoretical foundations.

The Kalman filter is the general recursive Bayesian filtering algorithm, assuming that the system dynamics and the observation model are described by linear differential and algebraic equations and all noises are normally (Gaussian) distributed. Under these conditions, the minimum mean square error (MMSE) estimate can be calculated using a Kalman filter. An important advantage of this filter is that it can use almost any information from the available sensors, regardless of how often they update the data and how accurate their measurements are.

In general, Kalman filtering can be defined as the process of optimally estimating the states of a real dynamic system operating under the influence of random disturbances. More precisely, the Kalman

filter is a linear recursive algorithm that reconstructs the unknown states of the system from observations of its noisy output in discrete real time (Popov, 2022).

The dynamics of the system are described by a linear discrete model represented by the expression:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k, \quad (1.18)$$

where \mathbf{x}_k is the state vector at time k , \mathbf{u}_k is the control input, and \mathbf{w}_k is the process noise with covariance matrix \mathbf{Q}_k . The matrices \mathbf{F}_k , \mathbf{B}_k , and \mathbf{G}_k describe, respectively, the dynamics of the state, the influence of the control actions, and the way in which the process noise affects the system. The measurement model is specified according to:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \quad (1.19)$$

where \mathbf{z}_k is the measurement vector, \mathbf{H}_k is the measurement model matrix, and \mathbf{v}_k is the measurement noise with covariance \mathbf{R}_k .

The Kalman filter algorithm is implemented through a recursive procedure in two main stages. In the prediction stage, the prior estimate of the state and the corresponding covariance matrix are calculated:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \mathbf{u}_k, \\ \mathbf{P}_k^- &= \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T. \end{aligned} \quad (1.20)$$

The covariance matrix \mathbf{P}_k^- is a statistical measure of the confidence of the forecast and reflects the accumulated uncertainty due to process noise and system dynamics.

When a measurement is available, the correction stage is performed, during which the forecast is updated using the innovation vector and the Kalman gain coefficient:

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-), \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \end{aligned} \quad (1.21)$$

The Kalman gain coefficient \mathbf{K}_k acts as a weighting factor that balances the influence of measurements on the model forecast. With low measurement uncertainty, the correction is dominated by the measurements, while with a high noise component, the filter relies more on the dynamic model.

This formal structure provides a stable and optimal estimate for linear systems with Gaussian noise and represents the theoretical basis on which the extended and hybrid variants of Kalman filtering, discussed in the following subsections, are built (Popov, 2022).

1.2.1. Extended Kalman filter (EKF) – linearization of nonlinear systems.

In real navigation applications, the dynamics of the system and the measurement dependencies can rarely be described by strictly linear models. This requires the use of an Extended Kalman Filter (EKF), which is a generalization of the classical Kalman filter for nonlinear systems through local linearization around the current state estimate. The nonlinear discrete model of the system is specified as follows:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{G}_k \mathbf{w}_k, \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \end{aligned} \quad (1.22)$$

where $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are nonlinear functions describing the dynamics of the state and the measurement process, respectively. The process and measurement noise retain the same statistical assumptions as in the classical Kalman filter.

The main idea of the EKF is to approximate the nonlinear functions using the first term of the Taylor series expanded around the current posterior state estimate. This leads to the definition of the Jacobians of the dynamic and measurement functions:

$$\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k}, \quad \mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_k}. \quad (1.23)$$

After this linearization, the prediction stage of the EKF formally retains the structure of the classical filter:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k), \\ \mathbf{P}_k^- &= \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T. \end{aligned} \quad (1.24)$$

The correction stage also follows the standard Kalman scheme, using the Jacobian \mathbf{H}_k instead of a linear measurement matrix. In this way, the EKF allows the processing of nonlinear systems by combining nonlinear prediction with statistically optimal correction.

Despite its wide application, the EKF has significant limitations. Linearization is only valid locally and assumes small deviations around the operating point. In highly nonlinear dynamics or during rapid maneuvers, the Jacobians may prove to be a poor approximation, leading to systematic errors and even loss of filter stability. Furthermore, the calculation of analytical Jacobians complicates implementation and increases the risk of model inaccuracies, especially in complex multidimensional systems.

These limitations are clearly identified in classic EKF applications for mobile platform navigation and are a major motivation for seeking alternative or hybrid methods in which nonlinear dynamics are described by learnable models rather than analytical linearization (Popov, 2022).

1.3. Limitations of classical and extended Kalman filters.

Despite their wide application and proven effectiveness, classical Kalman filters and their extended variants have a number of fundamental limitations that are particularly evident in navigation tasks for unmanned aerial vehicles. These limitations are not a consequence of incorrect implementation, but stem directly from the mathematical assumptions on which the theory of Kalman filtering is based.

First, all variants of the Kalman filter assume that **the dynamic model of the system is known and adequately describes the real physics of the process**. In the linear Kalman filter, this requirement is formulated through the transition matrices \mathbf{F} and the control matrices \mathbf{B} , while in EKF and UKF the same role is performed by the nonlinear function $\mathbf{f}(\cdot)$. In real navigation systems, however, the dynamics of UAVs are highly nonlinear, dependent on aerodynamic effects, turbulence, vibrations, and variable control modes, which are difficult to describe with a fixed analytical model. As a result, **a structural error arises in the model** that cannot be compensated for solely by adjusting the noise covariances.

A second significant limitation is related to **noise assumptions**. Classical theory assumes Gaussian, white, and stationary noise in both the process model and the measurements. In INS/GNSS navigation practice, these conditions are rarely met. The noise of inertial sensors exhibits temperature dependence, drift, and correlated components, while GNSS measurements are subject to degradation, multipath propagation, and sudden signal loss. Under such conditions, the fixed matrices \mathbf{Q} and \mathbf{R} lead to incorrect uncertainty estimation and suboptimal filter performance.

Additionally, problems arise in EKF related to **the linear approximation of nonlinear dynamics**. Jacobians are calculated around the current state estimate, which means that for larger errors or highly nonlinear regimes, the approximation becomes inaccurate. This can lead to instability, delayed convergence, or even divergence of the filter. Although UKF reduces the influence of errors from linear approximation by transforming using the unscented transform method, it also remains dependent on the correctness of the analytical model and the correct choice of unscented transform parameters.

An important aspect of UAV navigation is the behavior of the estimation filters in dead reckoning mode when absolute measurements (e.g., GNSS) are temporarily unavailable. In such cases, correction from an absolute source is unavailable, and estimation relies primarily on process dynamics and inertial measurements, which is why bias errors, non-stationary noise, and model mismatches accumulate over time and lead to increasing errors in position and velocity. Review studies show that during prolonged GNSS outages and highly nonlinear dynamics, classical filtering schemes can exhibit significant degradation in accuracy and/or consistency, which motivates the use of robust and adaptive approaches.

All the limitations discussed clearly indicate that the problem is not in the Kalman filter framework itself, but in **the inability of the analytical dynamic model to capture the complex, nonlinear, and non-stationary dependencies** characteristic of the real dynamics of UAVs. This leads to the logical conclusion

that in order to improve the accuracy and stability of the navigation system, the dynamic model needs to be supplemented or replaced with a more flexible approximation structure.

This necessity forms the theoretical basis for the transition to **hybrid models**, in which Kalman filtering remains the optimal mechanism for integrating information and managing uncertainty, while the description of the dynamics is implemented through trainable neural networks. In this context, LSTM-type recurrent architectures are a natural choice, as they are capable of modeling sequential data, nonlinear dependencies, and temporal correlations that remain outside the scope of classical models.

1.4. Application of artificial intelligence in modern navigation systems.

Over the last decade, artificial intelligence has become a leading factor in the development of navigation algorithms for unmanned aerial vehicles. Classic models based on analytical equations and linearity assumptions are no longer sufficient in conditions of highly nonlinear dynamics, high-frequency noise, sensor failures, and GNSS interference. UAV state systems demonstrate significant deviations when the real environment does not correspond to the predefined models, which is typical for the typical operating environment: turbulence, vibrations, electronic countermeasures, temporary IMU blindings, and sudden changes in aerodynamic loads.

The accumulation of large data sets, as well as the ability to synthetically generate training cases in MATLAB/Simulink, creates the conditions for a transition to data-driven and hybrid AI approaches. These methods occupy an intermediate position between "white-box" physical models and "black-box" deep learning architectures, with the aim of reducing model error and increasing the accuracy of dynamics prediction over a wide range of flight modes. For example, in a presentation at MATLAB EXPO 2024–2025 (Jawahar & Ulusoy, 2024) clearly emphasizes that neural state-space models and reduced AI models (AI-ROM) are now part of the industry standard for controlling complex systems, including UAVs.

This gives rise to a research paradigm in which AI not only complements the classical Kalman filter, but transforms the state model itself, allowing for an adaptive structure that is more robust to noise, dynamic uncertainty, and nonlinearity.

1.4.3. Hybrid AI-Kalman approaches.

Hybrid AI-Kalman approaches are one of the leading lines in modern navigation algorithms, as they combine the physically grounded structure of Kalman filtering with the adaptability of deep learning methods. The main idea is to integrate learnable components into key blocks of recursive estimation - most often in the prediction step (the state transition model), in the process noise model/statistics, as well as in adaptive mechanisms for tuning the \mathbf{Q} and \mathbf{R} covariances. This preserves the probabilistic "forecasting-correction" scheme while reducing the dependence on the strict analytical nature of the model and the assumption of linearity (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022).

As discussed earlier in the chapter, in the classical linear KF, the dynamic model is specified as follows:

$$x_{k+1} = F_k x_k + B_k u_k + w_k, \quad (1.33)$$

where F_k, B_k , and the statistics of w_k are known or modeled a priori. In practice, with UAVs, the actual dynamics are often:

- highly nonlinear,
- changing over time,
- dependent on factors that are difficult to describe physically (turbulence, mechanical vibrations, temperature nonlinearities),
- subject to non-ideal and, in general, non-Gaussian influences.

One of the widely used hybrid approaches is **approximation or replacement of the transient model** with a trainable nonlinear function:

$$x_{k+1} = f_{\theta}(x_k, u_k) + w_k, \quad (1.34)$$

where f_{θ} is a neural component (e.g., a recurrent network) trained on real or simulated data. A representative example of this class is KalmanNet, which preserves the recursive structure of Kalman estimation but uses

a neural network to increase robustness to nonlinearities and model mismatches in partially known dynamics (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022).

In addition, there are hybrid enhancements that **do not replace** the dynamic model with a neural network but introduce **adaptability** in the filter tuning (Figure 11). In the publication *Kalman Filter Enhancement for UAV Navigation* Johnson, Sasiadek, and Zalewski (Johnson, Roger; Sasiadek, Jerzy; Zalewski, Janusz, 2002) propose:

- fuzzy logic for adaptive tuning of parameters/noise strengths to limit EKF divergence;
- parallelization to accelerate calculations, demonstrating the effect through simulation results.

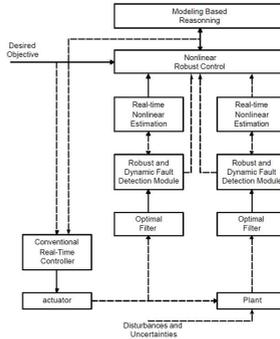


Figure 11. Basic architecture of the control system.

Source: *Kalman Filter Enhancement for UAV Navigation* (Johnson, Roger; Sasiadek, Jerzy; Zalewski, Janusz, 2002).

Similar ideas for data-based learnable modeling of nonlinear dynamics can be found in the literature on **kernel-based Bayesian filters**, in which the transition and/or measurement model is described in a feature space (RKHS), and the recursive "prediction-correction" scheme is implemented through kernel operations. Representative developments in this direction are *the Kernel Kalman rule / Kernel Kalman filter* (Gebhardt, G.H.W., Kupcsik, A. & Neumann, G., 2019) and *the Adaptive Kernel Kalman Filter* (Sun, M., Davies, M. E., Proudler, I., & Hopgood, J. R., 2023), which formalize estimation through kernel embeddings and allow operation with nonlinear dependencies.

In the publication *Navigation complex with adaptive non-linear Kalman filter for unmanned flight vehicle* (Selezneva, Neusypin, & Proletarsky, 2019), **an adaptive nonlinear Kalman filter** (Figure 12) with parametric identification of the INS error model is developed, with the authors comparing several implementations (KF, NKF, NKF+OLS, NKF+GMDH, etc.). The results presented show a significant improvement in accuracy compared to the classical KF (e.g., a reduction in position error from 12–15 m with KF to about 4–5 m with the best variants in their experimental scenario).

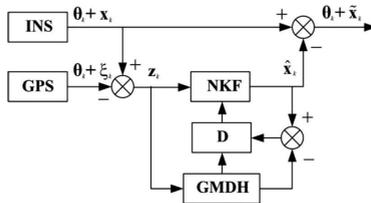


Figure 12. Block diagram of NC with NKF using GMDH.

Source: *Navigation complex with adaptive non-linear Kalman filter for unmanned flight vehicle* (Selezneva, Neusypin, & Proletarsky, 2019).

An example of modern hybrid implementations is KalmanNet, in which a neural module is trained to estimate **the Kalman gain K_k** (Kalman gain), while the recursive structure of the filtering is preserved.

From a GNSS-resilient perspective, an approach widely used in INS/GNSS literature involves AI generating pseudo-measurements during GNSS outages, which are then fed into the EKF as substitute/auxiliary information. For example, Fang et al. (Fang, W., Jiang, J., Lu, S., Gong, Y., Tao, Y., Tang, Y., Yan, P., Luo, H., & Liu, J., 2020) use LSTM to generate pseudo GNSS increments when the GNSS signal is lost, and Zhi et al. (Zhi, Z., Liu, D., & Liu, L., 2022) apply CNN–LSTM to compensate for the degradation of GPS/INS integration during GPS outages. In this class of systems, the logic is: **INS provides the dynamics, AI evaluates replacement/compensating observational information, and EKF performs recursive correction.**

A separate but related line of work involves EKF-based architectures for sensor integration without AI, used as a baseline reference and starting point for extensions. An example of such a practical implementation is EKF-based integration of GNSS/IMU/barometric data in MATLAB/Simulink, discussed in Motayed's work (Motayed M. S., 2025).

With regard to anomaly and failure diagnostics, neural network-based methods can be used as **detectors of atypical modes** in a stream of measurements. Although Djaghoul et al. (Djaghoul, Tehrani, & Vurpillot, 2025) demonstrate the approach in the context of power electronics (inverter), the principle of detecting anomalous patterns using 1D CNN/LSTM is methodologically transferable to navigation sensor streams.

For robust **position estimation** in a GNSS-denied environment, a **hybrid 1D CNN/LSTM architecture** is proposed. The **1D CNN** layer extracts high-frequency characteristics from the inertial sensor streams, while the subsequent **LSTM** layer models their long-term dynamics (Woyano, F., Park, S., Iordanov, V. B., & Lee, S., 2023). This approach improves accuracy in the event of sudden GNSS signal loss.

Finally, in control tasks (Figure 13), reinforcement learning (RL) is often combined with a state estimation block in the control loop; in the MathWorks presentation (Baru, MathWorks) , this is shown conceptually by the presence of a separate "State Estimation" block in the architecture.

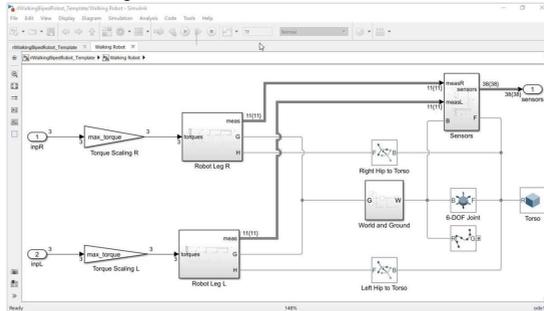


Figure 13. Simulink model for RL-based robot control.

Source: Reinforcement Learning: Leveraging Deep Learning for Controls (Baru, MathWorks).

All hybrid models considered have significant advantages over traditional Kalman models, but for greater clarity, an analysis of the advantages of hybrid AI–Kalman approaches has been performed

Table 6. Synthesized advantage of hybrid AI–Kalman approaches.

Functional element	Classic KF/EKF/UKF	Hybrid AI–Kalman (AI–KF)
<i>Prediction</i>	The dynamic model $f(\cdot)$ is analytically specified (linear for KF; nonlinear with linearization for EKF; sigma points for UKF). Accuracy depends heavily on the adequacy of the model	The transition model can be approximated/corrected using ANN $\hat{f}(\cdot)$ or a trainable correction term (residual) can be used. Reduces sensitivity to model mismatch
<i>Noise models Q, R</i>	Usually assumed to be constant or manually tuned; classical adaptation (innovation-based) is possible, but with limited flexibility	Q and/or R can be estimated adaptively from data (e.g., ANN/regression models), including dependence on mode, dynamics, and measurement quality
<i>Operation under nonlinearities</i>		Trainable components can model more complex nonlinear dependencies and improve robustness in model mismatch;

	EKF/UKF allow nonlinear models, but the posterior is usually a Gaussian approximation ; robustness depends on linearization/distributions and model quality	Gaussian approximation is often retained, but with better adapted parameters/models
<i>Anomalies / outliers</i>	Sensitivity to outliers under standard assumptions; requires additional robust mechanisms (gating, M-estimators, robust loss)	AI detectors for anomalies and/or trainable gating/weighting of measurements can be added to support robust rejection or confidence re-evaluation
<i>Compensation for drift and systematic errors</i>	Compensation depends on the included bias/scale state vectors and the correctness of the models; limited for unmodeled effects	ANN/time models (e.g., LSTM) can describe time-correlated drifts and unmodeled dependencies (temperature, vibrations) by adding correction terms or noise adaptation
<i>Behavior during GNSS interruptions</i>	Robustness is determined by INS errors and Q tuning; during longer interruptions, the error increases due to integration drift	Pseudo-measurements from AI, additional modalities (VIO/SLAM), or adaptive drift models are often used, which can limit the increase in error during interruptions
<i>Computational complexity and implementation</i>	Relatively predictable complexity; well-established implementations and tools (e.g., MATLAB/Simulink)	Additional complexity from trainable blocks (training/inference, data, validation); with a well-chosen architecture, it remains compatible with real time, but requires more engineering
<i>Interpretability and verification</i>	High interpretability of the model and parameters; easier formal verification and diagnostics	The recursive structure remains interpretable, but the trainable components are partially a "black box"; require additional procedures for validation, robustness, and safety

In summary (Table 6), the developments discussed show that hybrid AI-Kalman approaches are **one of the most promising** lines for UAV navigation systems: they preserve the recursive probabilistic structure of filtering, using AI to model/adapt nonlinearities, noise structures, and drifts that are difficult to describe analytically.

1.5. CONCLUSIONS FROM CHAPTER 1.

1. The analysis in this chapter shows that neither INS, nor GNSS, nor visual methods alone provide a stable solution in all flight modes, which is why integrated architectures are a methodologically sound choice; in particular, in dead reckoning, the systematic errors (bias) of the IMU, temperature instability, and stochastic measurement deviations determine the rate of navigation error accumulation and, accordingly, the accuracy of position and velocity estimates, especially during prolonged intervals without external corrections (e.g., GNSS or other absolute measurements).
2. Inertial navigation systems provide high autonomy, but due to the integration of inertial measurements, they inevitably accumulate drift, with the rate of accumulation depending on the class, calibration, and temperature stability of the inertial measurement unit (IMU), including MEMS-IMU.
3. GNSS navigation provides high absolute accuracy under normal reception conditions, but is vulnerable to jamming, spoofing, and degradation (including multipath signal propagation), as these disturbances are a realistic operational factor in environments with electronic countermeasures (electronic warfare - EW) and should be taken into account when designing navigation architectures.
4. Visual navigation methods expand the possibilities for autonomous localization and mapping, and their integration with INS via EKF/UKF increases resilience to single sources, but remains sensitive to observation conditions (illumination, texture, motion dynamics) and does not ensure drift elimination without absolute corrections or additional global constraints (e.g., GNSS, beacons, maps/loop closure), especially in partial observability.
5. EKF relies on linearization of nonlinear models and, in the case of strong nonlinearity or inaccurate modeling, can lead to suboptimal estimates or divergence. UKF reduces risk through sigma points; its accuracy and consistency depend on the correct tuning of noise covariances and state parameterization, especially in the case of non-stationary noise without adaptive mechanisms.
6. Kalman filters (including EKF/UKF) are effective with well-defined models and controlled assumptions, but can degrade in real operation with degraded measurements, non-stationary noise, outliers, and dynamic changes in the availability of observations if robust and/or adaptive mechanisms are not used.

7. Real UAV scenarios are characterized by strong nonlinearity, non-stationary noise, and intermittent (discontinuous) availability of GNSS measurements, depending on the environment and scenario
8. In the event of a systematic violation of the linear-Gaussian and stationary assumptions, Kalman-type models and filters may degrade; therefore, robust and/or adaptive mechanisms are necessary
9. Artificial intelligence methods provide opportunities for adaptation, modeling of nonlinear dependencies, and extraction of patterns from sensor data, including in difficult-to-model modes, but require precise training, validation, and generalization control.
10. The need for integrated navigation solutions justifies the use of robust and adaptive architectures, including approaches that combine Kalman filters with AI methods, for example for anomaly detection, noise model adaptation, or support models for evaluation.

CHAPTER 2. ALGORITHMS FOR ARTIFICIAL INTELLIGENCE IN THE MANAGEMENT OF UNMANNED AERIAL VEHICLES.

2.1. Modeling of dynamic systems with neural networks.

In modern autonomous UAV control systems, there is a growing share of data-based methods in which dynamics are described by trainable models instead of (or together with) strictly analytical equations. The actual behavior of the flying platform is often nonlinear, context-dependent, and time-varying, and the uncertainties caused by aerodynamic effects, vibrations, and changes in the operating environment are not reliably captured by simplified physical models. In this sense, deep learning does not "replace" classical control theory, but provides a mechanism for identifying and adapting components that are difficult to model a priori (Cohen & Klein, 2024).

One line of research is deep state-space models (deep SSM), in which the state transition is specified by a neural network (or neural submodels), and training is performed on input-output time series. This class of models can be viewed as a generalization of the classical description of state space, but with higher expressiveness in nonlinear systems and the possibility of probabilistic (stochastic) modeling of uncertainty. Gedon et al. (Gedon, D., Wahlström, N., Schön, T. B., & Ljung, L., 2021) demonstrate that deep SSMs are applicable as a "black box" for nonlinear system identification and evaluate several variants on benchmark datasets, explicitly emphasizing the role of neural parameterizations in representing a wide class of dynamics.

A second line of research that expands the possibilities for modeling dynamics with neural networks in the context of control is the use of latent dynamic models for planning and forecasting under partial observability. The approach of Hafner et al. (Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., & Davidson, J., 2019) uses a recurrent state model in latent space and trains a transition model that allows for multi-step predictions and action planning without the need for explicitly specified physical dynamics. This is particularly relevant for UAVs because navigation and control tasks are often partially observable (limited visibility, temporary GNSS outages, etc.), and the latent state can serve as a compact summary of the observation history.

A third important line of research is Koopman approaches, in which a neural network is used to transform the state to a higher-dimensional space so that nonlinear dynamics can be approximated by linear dynamics in that space, which facilitates control synthesis. Han et al. (Han, Y., Hao, W., & Vaidya, U., 2020) consider DNN-based learning of Koopman representation for control synthesis and highlight the advantage of combining data-based identification and classical control methods (e.g., linear quadratic regulator - LQR/model predictive control - MPC) on the resulting linear approximation. The practical value for UAVs is that such a model can facilitate the design of controllers under strong nonlinearities and uncertain parameters without requiring a complete physical modeling of all effects.

In navigation applications, deep learning is most often used as a supplement to classical structures (e.g., filtering and sensor integration), assisting in calibration, denoising, and partial learning of model parameters or components, rather than setting them entirely analytically. In this regard, the review by Cohen and Klein, "systematizes trends in the integration of deep learning with inertial navigation and sensor data

integration, including approaches in which components of model-based algorithms are learned rather than defined entirely analytically.

2.1.1. RNN and LSTM.

Recurrent neural networks (RNN) are suitable for modeling dynamic systems because they use an internal state to describe the temporal dependencies between successive inputs. In the context of UAVs, this is important because inertial measurements, visual observations, and navigation residuals are typically interpreted as a time sequence that carries information about current motion and accumulating errors.

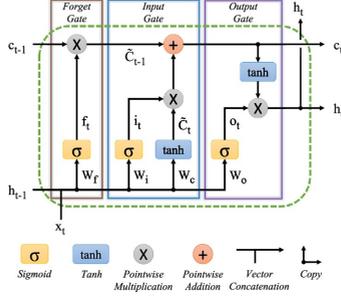


Figure 16. The LSTM cell can process data sequentially and maintain its hidden state over time.
Source: ResearchGate.

Classic RNNs often work well with short-term dependencies, but with long sequences they can exhibit problems such as gradient vanishing/explosion, which makes training difficult. For this reason, the LSTM (Long Short-Term Memory) architecture is widely used as a practical solution for modeling long-term dependencies (Figure 16), typical of IMU signals and visual-inertial sequences during flight (Hochreiter, S., & Schmidhuber, J., 1997).

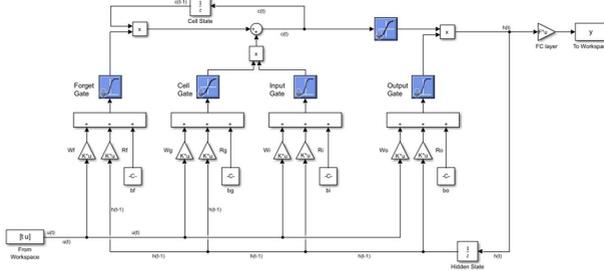


Figure 17. Simulink implementation of an LSTM layer followed by a fully connected layer (FC).
Source: MathWorks.

The MathLab analysis *LSTM Layer as a Nonlinear Dynamical System* (Campa, 2023) shows how an LSTM layer can be viewed as a nonlinear dynamic system (Figure 17) with a specific structure and whether and how LSTM layers can approximate linear dynamic systems:

$$\begin{aligned}
 f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f), \\
 i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), \\
 o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o), \\
 \tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c), \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\
 h_t &= o_t \odot \tanh(c_t),
 \end{aligned}
 \tag{2.1}$$

where:

Inputs/states:

- x_t - input vector at time t (e.g., IMU measurements, visual features, innovations/residuals, or concatenation of such quantities);
- h_{t-1} - hidden (initial) state of the LSTM from the previous moment $t-1$;
- h_t - hidden (output) state of the LSTM at time t ; this is the "output" of the cell, which is often fed to the next layer or to a regressor for prediction;
- c_{t-1} - cell (internal) state from time $t-1$, which carries the "memory" of the LSTM;
- c_t - cell (internal) state at time t , updated based on the current input and previous states.

Gates and candidate cell state:

- f_t - *forget gate*: determines how much of the c_{t-1} to retain;
- i_t - *input gate*: determines how much of the new information to write to the cell;
- o_t - *output gate*: determines what part of the current cell state should be "shown" as output h_t ;
- \tilde{c}_t - candidate for new cell content represents nonlinearly transformed "new information" that can be added to memory.

Nonlinearities and operations:

- $\sigma(\cdot)$ - logistic sigmoid function: limits the values of the gates in the interval $[0,1]$;
- $\tanh(\cdot)$ - hyperbolic tangent: limits values in $[-1,1]$ and stabilizes state updates;
- \odot - element-wise multiplication (Hadamard product).

This structure is suitable for modeling UAV dynamics, as the internal state c_t can encode long-term dependencies and accumulating effects (e.g., slow changes related to drift/systematic components), while the output state h_t is a vector representation of the current dynamics, which is fed to the next time step or to the output layer.

In state estimation tasks, RNN/LSTM-type neural models can be used as a module that reconstructs/approximates the motion dynamics from historical information, thus assisting or complementing classical estimators. In *Deep Learning-Based Neural Network Training for State Estimation Enhancement: Application to Attitude Estimation* (Al-Sharman, et al., 2019), it is demonstrated that such models can improve orientation estimation under complex and non-stationary measurement conditions, which is also relevant for UAVs.

Additionally, resistance to anomalous measurements is critical for sensors such as GNSS/IMU and visual artifacts. In *Outlier-Resistant Remote State Estimation for RNNs with Mixed Time-Delays* (Li, Wang, Dong, & Ghinea, 2020), recurrent models and procedures aimed at robust estimation in the presence of anomalies are discussed, with the idea being applicable to navigation sequences (e.g., GNSS degradation, IMU saturation, or problems in visual observations).

Separately, LSTM can also be used for prediction (not just filtering), predicting future states/positions (or other kinematic quantities) based on a sequence of observations. An example of such an approach is the work of Torkar (Torkar, 2019), which demonstrates time modeling with LSTM for forecasting in the context of sequential observations; the concept is methodologically transferable to UAVs in the presence of appropriate data and a correctly defined state.

The summary of the studies conducted is that RNN and LSTM architectures are a suitable tool for modeling the dynamics of UAVs in the presence of time dependencies and non-stationary conditions. They are most often used not as a complete replacement for physical models, but as a **complementary module - for approximating the process model**, compensating for systematic effects, or increasing robustness in degraded measurements, within the framework of hybrid navigation/estimation architectures.

2.2. ANN as a replacement dynamic model of KF.

Kalman filtering (KF, EKF, UKF) requires a process (transient) model that describes the evolution of the state over time and determines the quality of the forecast step. In the case of model incompleteness, strong nonlinearity, or mismatch between the analytical model and the actual dynamics (model mismatch), the forecast error may increase, and in conditions of limited observability or temporary absence of absolute correction (e.g., in calculation), the filter may exhibit significant degradation in accuracy and/or

inconsistency. This motivates the use of trainable components, in which the process model is not specified analytically but is parameterized by a neural network and trained from data, for example: $x_{k+1} = f(x_k, u_k) + w_k$. A number of hybrid approaches have been proposed in the literature, in which neural models support or partially replace the dynamics/parameters of the filter and show improvements in nonlinearities and model mismatch under experimental conditions. The results of publications related to hybrid models between classical Kalman filters and artificial neural networks achieve excellent results in terms of accuracy and efficiency (Feng, S., Li, X., Zhang, S., Jian, Z., Duan, H., & Wang, Z., 2023). The relatively compact RNNs used in the studies can be trained with a relatively small dataset, making them applicable to high-dimensional state-space models and devices with limited computing capabilities (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022). The experimental results show the potential of the hybrid approach in terms of improving accuracy, robustness, and interpretability (Chen, C., Lu, X., Wang, B., Trigoni, N., & Markham, A., 2021).

2.2.2. ANN as a predictive model.

The use of artificial neural networks (ANN) as prediction models within navigation algorithms is an area of growing interest in the literature on autonomous systems and sensor data integration. Unlike approaches where ANN replaces the process model (transfer function) in the filter, here the neural network acts as an additional predictive/corrective module that works in parallel with the statistical framework of Kalman filtering. In this sense, ANN can assist the predictive step by providing additional information extracted from historical measurements, context, and/or latent dependencies, without necessarily disrupting the predict–update structure of the filter.

The practical motivation for predictive ANN layers is related to the limited applicability of fixed analytical models in conditions of strong nonlinearities, non-stationary noise, and model mismatch. In such modes, the prediction error can acquire a systematic component that cannot be effectively compensated for solely by classical covariance tuning. Particularly critical are dead reckoning modes in the temporary absence of absolute correction (e.g., GNSS failure), when the accumulation of errors accelerates and the dependence on the forecast increases (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025); (Kim, S., Petrunin, I., & Shin, H.-S., 2022).

One of the key ideas in this class of approaches is for ANN to predict not only the state itself, but also the model error/bias/residual prediction. Instead of the neural network completely replacing the dynamics, it can be trained to estimate a correction term Δx_k or an equivalent correction in the forecast that compensates for systematic inconsistencies in the analytical process model (Farchi, A., Bocquet, M., & Brajard, J., 2021); (Brajard, J., Carrassi, A., Bocquet, M., & Bertino, L., 2020). Similarly, ANN can be used to estimate/adapt noise characteristics (e.g., effective covariances) when the assumptions of noise stationarity are violated (Levy, A., & Klein, I., 2025); (Sun, B., Zhang, Z., Qiao, D., Mu, X., & Hu, X., 2022); (Wang, H., Deng, Z., Feng, B., Ma, H., & Xia, Y., 2017). These types of setups are considered part of the more general class of AI-aided Kalman filtering methods, in which the filtering structure is preserved but a learnable compensation or adaptation mechanism is added.

In the context of an integrated navigation system, ANN can be used in several typical ways, which are discussed in the literature on hybrid filtering architectures (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025); (Kim, S., Petrunin, I., & Shin, H.-S., 2022):

- **A direct state prediction model** that provides an additional prediction \hat{x}_{k+1} (or a correction term to it) and is combined with the filter (Farchi, A., Bocquet, M., & Brajard, J., 2021); (Brajard, J., Carrassi, A., Bocquet, M., & Bertino, L., 2020);
- **Predictive model of observations**, e.g., prediction of future IMU/incremental measurements or innovations, in order to stabilize estimation in degraded data (Sun, B., Zhang, Z., Qiao, D., Mu, X., & Hu, X., 2022);
- **A predictive error/inconsistency model** in which ANN estimates the bias or structural error of the process model and uses it to correct the prediction step or adapt the noise

parameters (Levy, A., & Klein, I., 2025); (Wang, H., Deng, Z., Feng, B., Ma, H., & Xia, Y., 2017).

Each of these approaches can be integrated into a hybrid architecture provided there is a clear framework for training, validation, and generalization control, as predictive ANN components can improve accuracy in the scenarios considered, but they can also introduce systematic biases in the case of domain differences or unrepresentative training data. Therefore, in engineering implementation, ANN is considered an additional module that reduces dependence on simplified analytical models while maintaining a structured assessment scheme and clear sustainability criteria (Kim, S., Petrunin, I., & Shin, H.-S., 2022) A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025); (Choi, G., Park, J., Shlezinger, N., Eldar, Y. C., & Lee, N., 2023).

Thus, ANN in the role of a predictive model acts as a mechanism that extends the predictive power of Kalman filtering beyond the limitations of fixed analytical models by providing an additional forecast trained on real or simulated data (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025). As a result, the hybrid system may exhibit higher resilience in scenarios with high uncertainty, nonlinearity, and deterioration of sensor data quality—situations typical of real-world operating conditions (Sun, B., Zhang, Z., Qiao, D., Mu, X., & Hu, X., 2022); (Levy, A., & Klein, I., 2025).

In order for the role of ANN as a predictive model to be formally and clearly defined within the framework of Kalman filtering, the following two types of formulations are most commonly used.

(A) Correction of the forecast step by predicting model error

Let the classical process model (analytical or approximate) be:

$$x_{k+1} = f_{\theta}(x_k, u_k) + w_k, \quad w_k \sim N(0, Q_k). \quad (2.12.1)$$

In the predictive (aiding) setting, the ANN is trained to predict a **correction term** $\Delta_{\theta}(\cdot)$, which compensates for structural error/incompleteness of the model:

$$\hat{x}_{k+1|k} = f_{\theta}(\hat{x}_{k|k}, u_k) + \Delta_{\theta}(\xi_k), \quad (2.12.2)$$

where ξ_k is a vector of input features (e.g., a window of IMU measurements, previous innovations, bias estimates, contextual parameters). This formulation is particularly useful in dead reckoning mode, when there is no absolute correction (e.g., GNSS failure) and the systematic component of the prediction error dominates the accumulation of drift.

Thus, ANN does not completely replace the process model, but acts as a **corrective to the forecast**, reducing systematic deviations in incomplete/inaccurate analytical dependencies. Similar ideas for **online model error correction** and combining ML with assimilation/estimation are discussed and experimentally validated in the literature (Farchi, A., Bocquet, M., & Brajard, J., 2021); (Brajard, J., Carrassi, A., Bocquet, M., & Bertino, L., 2020). In review articles on **AI-aided Kalman filtering**, this approach is considered part of a class of methods in which a learnable module compensates for model mismatches while preserving the filtering structure (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025).

(B) Adaptation of noise covariances via ANN (non-stationary noises)

In classical Kalman filtering, the process and measurement noise covariances Q_k and R_k are often assumed to be stationary or set by default. However, in real UAV navigation systems, noise characteristics can be non-stationary (e.g., change in vibration mode, turbulence, GNSS degradation, change in measurement quality), which motivates an adaptive approach.

A standard line is an online adaptation of Q_k and/or R_k through residual/innovation statistics (innovation-based adaptive filtering), where the covariances are adjusted so that the statistical properties of the innovation sequence match the filter assumptions. This approach and its variants have been widely discussed in adaptive KF schemes and in the context of integrated INS/GNSS systems (Sun, B., Zhang, Z., Qiao, D., Mu, X., & Hu, X., 2022); (Wang, H., Deng, Z., Feng, B., Ma, H., & Xia, Y., 2017).

In an ANN-based setting, adaptation can be parameterized through learnable functions that estimate effective covariances as functions of features:

$$Q_k = \mathcal{Q}_{\theta}(\xi_k), \quad R_k = \mathcal{R}_{\phi}(\zeta_k), \quad (2.12.3)$$

where:

Q_k - covariance matrix of the process noise at discrete time k .

R_k - covariance matrix of the measurement noise at time k .

k - discrete time index (iteration/filter step).

Q_θ and \mathcal{R}_ϕ - trainable functions (e.g., ANN) that return Q_k and R_k .

θ and ϕ - parameters (weights) of the respective neural models.

ξ_k and ζ_k - input features/vector of characteristics for moment k , from which ANN predicts Q_k and R_k , where ζ_k may include indicators of observation quality (e.g., C/N₀, geometry, detected interference, innovation statistics). This allows the filter to "readjust" the weight between the forecast and real-time measurements under degraded or changing conditions. Similar neural-adaptive settings (e.g., neural/adaptive UKF) are considered as a means of adapting the filter to non-stationary noise. (The general class of methods for AI-aided Kalman filtering also includes approaches for learning/adapting noise parameters and statistics, with the goal of robustness under realistic conditions and model mismatch (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Dunik, J., Straka, O., Closas, P., & Eldar, Y. C., 2025).

2.2.3. Training on simulated or real data.

One of the key practical issues in building an ANN as a component of a hybrid navigation system is the choice of training strategy: using simulated data, using real measurements, or a combined approach. The two data sources are not mutually exclusive, but they differ significantly in terms of controllability of conditions, statistical realism, and the risk of mismatch between training and operational environments.

Simulation data provides an important advantage in that it allows for systematic control of flight modes, dynamic effects, and sensor model parameters, including noise tuning and repeatable experiments in a controlled environment. This makes simulation particularly suitable for **initial development and validation** of algorithms, as well as for generating large amounts of data in scenarios that are difficult, dangerous, or resource-intensive to implement in real conditions. In this context, MATLAB/Simulink (Figure 21) is used as an engineering platform for modeling, simulating, and evaluating autonomous UAV systems and for virtually testing autonomy solutions and sensor models (Antoniou & Grabowski, 2021).

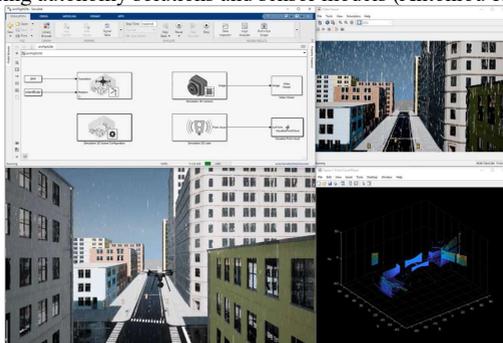


Figure 21. Tools and instruments in MATLAB and Simulink for developing autonomously controlled drones.

Source: *MatlabEXPO Autonomous UAV Development and Evaluation with MATLAB and Simulink* (Antoniou & Grabowski, 2021).

In addition, simulation data inevitably carries the risk of idealizing the nature of this data. Even with carefully constructed sensor models, simulation often fails to adequately reproduce complex effects such as long-term drifts, temperature dependencies, structural vibrations, and other non-stationary components of error. This leads to a gap between simulation and reality (sim-to-real gap), where an algorithm or trained model behaves differently when working with simulated versus real measurements. Specifically for GNSS and IMU sensors, a quantitative approach has been proposed to assess this gap by comparing the behavior of the state estimator on real and simulated experiments under different noise models, isolating the contribution of sensor simulation inaccuracy (Mahajan, I., Unjhawala, H., Zhang, H., Zhou, Z., Young, A., Ruiz, A., Caldararu, S., Batagoda, N., Ashokkumar, S., & Negrut, D., 2024).

Real measurements provide "statistical realism" because they contain the actual properties of noise processes and artifacts of the specific hardware. Studies on MEMS IMUs show that inertial sensor errors are a composition of various stochastic components (e.g., random walk, bias instability, etc.), which are evaluated and characterized using specialized methods such as Allan variance, PSD, and Gauss–Markov/AR models. Therefore, ANNs trained on real IMU time series can "see" dependencies and structures that are not represented in simplified simulation scenarios (Zhao, Horemuz, & Sjöberg, 2011).

In practice, this leads to a compromise: real data is more representative, but it is more difficult to collect in sufficient volume and often does not fully cover all modes of motion (especially for rare or extreme scenarios). Therefore, review studies in the field of deep learning for inertial navigation and sensor fusion emphasize the role of approaches that "reinforce" model-based schemes through data training and discuss the need for adequate preparation of training sets under real conditions and different motion domains (Cohen & Klein, 2024).

The combined approach is usually implemented as a **two-stage training**: (i) pretraining on simulated data to learn the global dependencies in the dynamics and the main causal relationships between state, control, and observations, and (ii) subsequent fine-tuning on real measurements, with the aim of calibrating the model to the actual noise characteristics, structural artifacts, and inevitable discrepancies of the simulation model. Such a strategy is motivated by the existence of a measurable sim-to-real gap for GNSS/IMU and by the fact that the accuracy of sensor simulation has a direct impact on the quality of state estimation (Mahajan, I., Unjhwala, H., Zhang, H., Zhou, Z., Young, A., Ruiz, A., Caldararu, S., Batagoda, N., Ashokkumar, S., & Negrut, D., 2024).

In summary, ANN training for navigation purposes should be planned to take advantage of both simulation and real data. Simulation is effective for initial training and covering controlled scenarios, while real measurements are critical for statistical reliability and adaptation to real noise processes. Therefore, for ANN components in a hybrid INS/KF navigation scheme, the most reliable strategy is a combined one in which preliminary training on simulation is complemented by fine-tuning on real flight data (Mahajan, I., Unjhwala, H., Zhang, H., Zhou, Z., Young, A., Ruiz, A., Caldararu, S., Batagoda, N., Ashokkumar, S., & Negrut, D., 2024); (Cohen & Klein, 2024).

2.2.4. Error and resilience.

In hybrid architectures combining neural networks with Kalman filtering, error analysis and robustness become central due to the introduction of an **additional mechanism of uncertainty**: the behavior of the learnable component outside the training distribution (domain shift) and potential generalization errors. At the same time, it is precisely the learnable modules that are used to compensate for the shortcomings of analytical models in **nonlinear dynamics**, **model mismatch**, and **non-stationary noise**, with the aim of preserving the structural interpretability and recursive efficiency of KF/EKF/UKF, but increasing robustness under real conditions. This direction is systematized in review papers on AI-assisted Kalman filters, which distinguish between approaches for (i) learning/correcting the dynamics and/or measurement model and (ii) learning/adapting the noise and uncertainty statistics (Shlezinger, N., Revach, G., Ghosh, A., Chatterjee, S., Tang, S., Imbiriba, T., Duník, J., Straka, O., Closas, P., & Eldar, Y. C., 2025).

2.3. Hybrid models (KF + ANN).

Research on combining Kalman filtering and artificial neural networks (ANN) has developed intensively over the last decade, with the main goal being to combine the recursive structure and statistical validity of KF/EKF/UKF with the approximation capabilities of ANN in nonlinear dynamics, model mismatch, and non-stationary noise. In the literature, this direction is considered as AI-aided Kalman filtering, covering methods for (i) learning/correcting the process and measurement model and (ii) adapting statistical parameters (e.g., noise covariances) for more robust state estimation.

The classic (linear-Gaussian) Kalman filter is optimal for linear models and Gaussian noise processes; for navigation tasks with pronounced nonlinearity, partial observability, degraded measurements, and/or non-stationary errors, generalized variants (EKF/ESKF/UKF) and/or additional robust and adaptive mechanisms are required, as well as in cases of known noise covariances and standard assumptions (e.g., white noise). In this context, ANNs are used not as a "substitute for filtering" but as

trainable components that assist the estimator in areas where the analytical model is incomplete or difficult to parameterize. Such solutions are also discussed in reviews of deep learning in inertial navigation and sensor integration, where both the potential advantages and the need for careful validation of generalization and consistency are emphasized.

In terms of architecture, hybrid KF+ANN models can be viewed as preserving the predict–update logic of the filter with the introduction of a learnable module in one or more key steps. Most often, ANN participates through:

- **trainable calculation of the Kalman gain coefficient** (e.g., $K_k = g_\phi(\cdot)$);
- **residual (additional) correction** of the process/measurement model (e.g., $x_{k+1} = Fx_k + Bu_k + \hat{f}_\theta(x_k, u_k) + w_k$ or $y_k = Hx_k + \hat{g}_\phi(x_k) + v_k$);
- **adaptive weighting/gating** when integrating estimates from different channels (e.g., $\hat{x}_k^{fusion} = \alpha_k \hat{x}_k^{phys} + (1 - \alpha_k) \hat{x}_k^{NN}$).

A representative example of a neural-assisted estimator with a Kalman structure is KalmanNet, which integrates a learnable component into the recursive predict–update scheme for state estimation in nonlinear systems and partially known models.

Within this subsection, hybrid KF+ANN solutions are structured into three main architectural lines according to the role of the learnable component:

1. **Recursive KalmanNet (R–KalmanNet)** – a recursive neural architecture for estimation, oriented towards trainable gain and quantification of uncertainty;
2. **Dual-branch systems** – parallel physical and neural branches with adaptive integration of estimates;
3. **KF for the linear component + ANN for the nonlinear/unmodeled part** – residual model that complements the analytical dynamics/measurement.

It is important to emphasize that hybrid methods **aim** to increase accuracy and robustness in model mismatch and variable noise characteristics, but the consistency of uncertainty should not be assumed a priori; it is verified by analyzing innovation sequences and standard statistical criteria (e.g., NIS/NEES), especially in modes with degraded measurements and dead reckoning.

2.3.3. KF for linearity + ANN for nonlinearities.

This hybrid approach is emerging as one of the most robust and practical architectures in modern UAV navigation systems. The idea is to divide the task into two conceptually different levels: **linear dynamics are handled by the classic Kalman filter**, while **nonlinear components and dependencies that are difficult to model analytically are handled by a neural network**. This fundamental compromise between analytical rigor and trainable flexibility avoids the linearization errors of EKF (Jacobi) and/or the difficult tuning/sensitivity of sigma-point approaches (UKF) in certain modes/high dimensions/non-stationary noises.

This model is found in several variations in the literature. Some publications consider the neural network as a corrective element that compensates for systematic errors, drift, or unidentified parameters of the dynamic model. Such an example can be found in the work of Li (Li, Wang, Dong, & Ghinea, 2020), where a recurrent architecture is used to compensate for nonlinear distortions in the process, while the basic linear Gaussian model continues to be estimated using KF. The same idea is visible in Al-Sharman and his team's work (Al-Sharman, et al., 2019), where the authors demonstrate that a well-trained ANN can replace analytical nonlinear functions and provide the KF with a more accurate "effective" estimate of the process acceleration.

The role of KF here remains clearly defined: the filter ensures optimality in linear relationships and Gaussian noise, helps stabilize the estimate, and is often more robust, even when ANN exhibits partial error or insufficient generalization. The reverse role of the neural network is to provide the missing dynamic dependencies that are not present in the analytical model. For example, the residual error between the observed acceleration and the predicted acceleration from KF can be used as input to the ANN, which predicts the nonlinear component and returns it back to the correction cycle.

Mathematically, the architecture is formulated through a modified process model:

$$x_{k+1} = Fx_k + Bu_k + \hat{f}_\theta(x_k, u_k) + w_k, \quad (2.15)$$

where \hat{f}_θ is the trainable neural network that models nonlinear deviations from linear dynamics. In some cases, a correction is also applied to the output:

$$y_k = Hx_k + \hat{g}_\phi(x_k) + v_k, \quad (2.16)$$

where:

- y_k is the vector of measurements (outputs) of the system at the discrete moment k ;
- H is the matrix of the linear part of the measurement model that links the state to the measurements;
- x_k is the vector of the system state at time k ;
- $\hat{g}_\phi(\cdot)$ is a parameterized nonlinear function with parameters ϕ , implemented through a neural network, which models the nonlinear part of the measurement process;
- v_k is the vector of measurement noise at time k , which accounts for sensor errors and disturbances.

The parallel neural branch allows modeling unmodeled nonlinearities, temperature-dependent drifts, and degraded quality measurements, which are typical for MEMS-IMUs. A number of studies on GNSS/INS integration, including *GNSS/INS Integration Methods* (Angrisano, 2010) and *Integrating INS and GNSS Sensors to Provide Reliable Surface Positioning* (Russell, 2012), report a significant reduction in cumulative (integration) error and increased resistance to sensor noise compared to standalone inertial navigation. In scenarios with GNSS interruptions, hybrid approaches with artificial neural networks (ANN) can help the filter maintain the consistency of estimates by modeling residual nonlinear deviations and systematic errors that would otherwise lead to accelerated error accumulation in **calculations**.

This hybrid dual-channel architecture is relatively compact and retains the standard structure of the Kalman filter (KF) (prediction–correction), with the dynamic model approximated by a neural network in the prediction step, i.e., $x_{k+1} = f_\theta(x_k, u_k)$. The integration of the estimates remains a separate, clearly defined stage, which facilitates the analysis and tuning of the system. The architecture is compact and suitable for implementation in MATLAB/Simulink, which is why it is appropriate for Chapter 3, where a simulation model of a hybrid navigation system will be built.

One of the most critical issues in the actual implementation of a Kalman filter in navigation systems is the selection of the process and measurement noise matrices. In the classical approach, the parameters Q and R are determined manually based on laboratory data or through engineering assessment. In dynamic operating environments—such as vibrations, temperature gradients, GNSS interference, or parasitic interference on the IMU—these parameters cease to be constant. This leads to incorrect estimates and compromises the optimality of the filter. Modern artificial intelligence methods provide a mechanism for automatic, adaptive, and data-driven determination of noise characteristics, complementing the structural advantages of KF with learnable models.

2.5. Reinforcement Learning for navigation – a comparative analysis.

The reinforcement learning (RL) paradigm offers a fundamentally different approach to the navigation problem compared to filtering methods. While KF-based systems seek to estimate the current state by modeling process and measurement noise, RL treats the task as an optimization problem of behavioral policy. The goal is to learn a strategy $\pi(a | s)$ that maximizes long-term return, rather than constructing an explicit estimate of the phase space.

The paper *Continuous Deep Q-Learning with Model-based Acceleration* (Gu, Lillicrap, Sutskever, & Levine, 2016) explores methods for training agents in continuously controlled spaces, which is particularly important for UAVs, where control is not discrete. The authors show that the combination of model-based simulation dynamics and deep Q-learning accelerates convergence and reduces the need for experimental flights.

Furthermore, in Liaq and Byun (Liaq & Byun, 2019), the practical aspects of RL for UAVs are examined—particularly the impact of partially observable environments, sensor errors, and dynamic obstacles. The results show that RL policies can outperform the classical navigation approach in complex conditions where the model is uncertain or incomplete.

However, RL has limitations. Policies are often difficult to interpret, require a large number of simulations, and their stability under real-world disturbances is not always guaranteed. It is important to

emphasize that RL does not replace Kalman filtering, but rather can complement it in high-level control tasks where target movements and maneuvers are learned through policy, while low-level state estimation remains the responsibility of KF/ANN models.

A comparative analysis of KF, ANN, and RL approaches to UAV navigation is provided in Table 21 in the appendix to the dissertation.

2.6. CONCLUSIONS FROM CHAPTER 2.

1. Neural architectures offer advantages over classical analytical models in a number of UAV navigation scenarios involving nonlinear dynamics, non-stationary noise, and partial observability, especially when model uncertainty is difficult to describe analytically. In this context, recurrent neural networks of the RNN, GRU, and LSTM types model temporal dependencies and accumulate context over time, which is critical in navigation tasks with drift in estimation and accumulating errors in computation.
2. The LSTM neural network allows modeling of long-term dependencies and is applicable to highly nonlinear processes. In the presence of uneven time intervals and sporadic observations, it is necessary to explicitly take into account the time scale, as continuous-time approaches such as Neural ODE/Neural SDE provide a natural formulation for such regimes.
3. A review of the literature on state estimation and motion prediction shows that LSTM models achieve good generalization between simulated and real trajectories with appropriate training, validation, and control of differences between the simulated and real environments.
4. Combining Kalman filtering (KF) with artificial neural networks allows for the preservation of structured sensor data integration and probabilistic interpretation of the estimation, provided that the noise characteristics are correctly modeled and calibrated.
5. Neural models within KF architectures compensate for inaccuracies in the dynamic model (e.g., by modeling residual deviations) and assist in adapting to non-stationary noise, e.g. by estimating/adapting the matrices of covariance of the process and measurement noise Q and R , as well as by detecting anomalous measurements (outliers) that deviate significantly from the expected behaviour of the sensors.
6. The analysis in this chapter justifies the choice of an integrated navigation model in which the transition function (dynamic model) of the filter is parameterized by an LSTM network, trained with sensor data and simulation trajectories, with the uncertainty of the model represented by estimation/adaptation of the process noise.

CHAPTER 3.

DESIGNING AN INTEGRATED NAVIGATION SYSTEM IN MATLAB/SIMULINK.

This chapter presents the process of designing, modeling, and simulating an integrated navigation setup for an unmanned aerial vehicle in the MATLAB/Simulink environment. Based on the theoretical concepts from Chapter 1 and the approaches developed in Chapter 2, an integrated simulation model is constructed that implements joint processing of inertial and satellite (GNSS) measurements within a Kalman filter framework, assisted by a neural process predictor. The developed system demonstrates a hybrid approach in which the classical EKF is combined with a deep recurrent LSTM network used as a process model in the prediction step of the filter.

In this experimental setup, MATLAB/Simulink is used as an integration platform providing mathematical modeling, visual block implementation, and simulation of sensor streams. Blocks representing IMU, GNSS, and measurement noise, as well as an EKF module for state estimation. A neural block trained in MATLAB using Deep Learning Toolbox has been added, which acts as a nonlinear predictive operator for the process dynamics in the prediction step.

The constructed model allows a direct comparison between the results obtained through the analytical process model and those generated by the neural approximation under identical input conditions. This provides an experimental basis for quantitative analysis of the accuracy and robustness of the estimation under different noise profiles and temporary loss of GNSS signal.

The following sections present the conceptual architecture of the system, the mathematical model of the processes and measurements, the parameterization of the noise, and the integration of the LSTM network into the Kalman filtering cycle. Special attention is paid to the implementation in the Simulink environment, where the individual components are arranged in a block structure that allows for sequential testing and visualization of the results.

3.1. Mathematical model of the system.

The mathematical modeling of the integrated navigation system aims to describe the relationship between the motion of the aircraft, the measurements from the inertial sensors and external observations, as well as the influence of noise processes. The developed model is formulated in such a way that it is both compatible with the simulation environment in MATLAB/Simulink and adaptable to the neural architecture used for state prediction via an LSTM predictor.

The mathematical model is based on the classical structure of inertial navigation systems, including **a dynamic equation** for state evolution, **a measurement equation** for describing observations, **a noise model** for simulating sensor imperfections, and **an Extended Kalman Filter (EKF)** as a standard method for state estimation.

The dynamic model of the system describes the relationship between the state vector and the control actions. For the purposes of integrated inertial navigation, a six-dimensional state vector is assumed:

$$\mathbf{x}_k = [p_x \ p_y \ p_z \ v_x \ v_y \ v_z]^T, \quad (3.1)$$

where p_x, p_y, p_z are the position coordinates, and v_x, v_y, v_z are the velocity components in an inertial frame.

\mathbf{u}_k are determined by IMU/sensor data:

$$\mathbf{u}_k = [a_x \ a_y \ a_z \ \omega_x \ \omega_y \ \omega_z]^T, \quad (3.2)$$

where a_i and ω_i are the designed accelerations and angular velocities; in this simplified model, it is assumed that the accelerations a_k are pre-converted to the inertial frame (or equivalently, that the body and inertial frames coincide), so ω_k does not participate in (3.3).

The dynamic equation in discrete form is formulated as follows:

$$\mathbf{x}_{k+1} = f_{dyn}(\mathbf{x}_k, \mathbf{u}_k, \Delta t) = \left[\mathbf{p}_k + \mathbf{v}_k \Delta t + \frac{1}{2} \mathbf{a}_k \Delta t^2 \right] + \mathbf{w}_k, \quad (3.3)$$

$$\mathbf{v}_k + \mathbf{a}_k \Delta t$$

where:

- \mathbf{x}_{k+1} is the state vector of the system at the discrete moment $k+1$;
- \mathbf{x}_k is the state vector at time k ;
- $f_{dyn}(\cdot)$ is the dynamic function of the process model, which describes the evolution of the state for the interval Δt ;
- \mathbf{u}_k is the vector of control inputs at time k ;
- Δt is the time interval between two consecutive discrete moments;
- \mathbf{p}_k is the position vector at time k ;
- \mathbf{v}_k is the velocity vector at time k ;
- \mathbf{a}_k is the acceleration vector at time k ;
- \mathbf{w}_k is the vector of process noise that models unmodeled dynamics and external influences;
- $[\cdot]$ denotes the vector form of the process model, including the equations for position and velocity.

In Simulink, this model is implemented through the MATLAB Function block **f_dyn**, which accepts the current state \mathbf{x}_k , the control vector \mathbf{u}_k , and the time step $T_s = 0.01$ s, calculating \mathbf{x}_{k+1} .

In parallel with this, the neural model is also integrated into the system:

$$\hat{\mathbf{x}}_{k+1} = f_{LSTM}([\mathbf{x}_{k-L+1}, \dots, \mathbf{x}_k], [\mathbf{u}_{k-L+1}, \dots, \mathbf{u}_k]), \quad (3.4)$$

where:

- $\hat{\mathbf{x}}_{k+1}$ is the predicted estimate of the state vector at the discrete moment $k+1$;
- $f_{LSTM}(\cdot)$ is a nonlinear function implemented through an LSTM neural network that models the dynamics of the system based on sequential data;

- $\mathbf{x}_{k-L+1}, \dots, \mathbf{x}_k$ are the state vectors for the previous L discrete moments forming the input time sequence;
- $\mathbf{u}_{k-L+1}, \dots, \mathbf{u}_k$ are the control input vectors for the corresponding L discrete moments;
- L is the length of the look-back window used by the LSTM network to extract temporal dependencies.

The model represents a nonlinear approximation of the dynamics through the trained deep recurrent network. The two models—the analytical and the LSTM—are structurally identical in terms of input-output connections and allow for a direct comparison of performance.

The measurement equation describes the relationship between the observed quantities \mathbf{z}_k and the true state \mathbf{x}_k . The integrated navigation system combines data from GNSS and IMU, with GNSS providing absolute coordinates and IMU providing relative velocity changes.

In general, the measurement equation is written as follows:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (3.5)$$

where \mathbf{v}_k is the measurement noise vector. When linearized around the current state:

$$\mathbf{z}_k = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \Delta t + \mathbf{v}_k, \quad (3.6)$$

where:

- \mathbf{z}_k is the vector of measurements at the discrete moment k ;
- $\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z$ are the components of the position vector along the respective axes of the coordinate system;
- $\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z$ are the components of the velocity vector along the respective axes;
- Δt is the time interval between two consecutive discrete moments;
- $[\cdot]$ denotes the vector form of the measurement model;
- \mathbf{v}_k is the vector of measurement noise at time k , which accounts for sensor errors and disturbances.

The result corresponds to the outputs from the **gPos** block in the Simulink model. GNSS noise is modeled by a diagonal dispersion matrix $R = \text{diag}(1.5^2, 1.5^2, 2.0^2)$, where the vertical component has twice the dispersion of the horizontal components. These values are implemented in the block **diag([1.5 1.5 2.0])u*, which ensures realistic behavior of the navigation measurements.

The measurement equation in the LSTM branch is equivalent, but the forecast $\hat{\mathbf{x}}_{k+1}$ is fed as input to the same observation mechanism. This ensures full comparability between the two models and identical processing of the measurements.

Noises in the system are divided into two main types — **process noise** and **measurement noise**. Their modeling is essential for realistic simulation and for the correctness of the training data used by the LSTM predictor.

The **process noisew_k** reflects the uncertainty in the dynamics and is modeled as a Gaussian distributed vector with zero mean and covariance matrix Q : $\mathbf{w}_k \sim \mathcal{N}(0, Q)$, $Q = \text{diag}(0.01^2, 0.01^2, 0.02^2, 0.01^2, 0.01^2, 0.01^2)$. This noise is introduced through the **nAcc** and **nGyro** blocks, which generate random signals with variance specified by the **Gacc** and **Ggyro** matrices.

The **measurement noisew_k** is applied to the outputs of the GNSS block **gPos**, with diagonal coefficients 1.5, 1.5, and 2.0, which creates a vertical error twice as large as the horizontal one.

In combination, these noises provide an approximation of the real conditions when operating inertial and satellite navigation systems, including the influence of sensor drifts and imperfections.

Noise processes are also key in training the LSTM network. Since the training data is generated from a simulation containing realistic noise, the neural network learns the statistical dependencies between measurements and hidden states even in the presence of stochastic deviations. This provides better generalization when used in experimental simulation.

To validate the dynamic model, an **extended Kalman filter (EKF)** has been implemented, which provides optimal state estimation under conditions of nonlinearity and noise. The EKF combines the forecast from the model f_{dyn} with observations from the measurement subsystem, minimizing the mean square error through successive iterations of prediction and correction.

The algorithm consists of two stages:

(1) Prediction:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= f_{dyn}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}), \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + Q_k,\end{aligned}\quad (3.7)$$

where $F_k = \frac{\partial f}{\partial \mathbf{x}}$ is the Jacobian of the dynamic model.

(2) Correction:

$$\begin{aligned}K_k &= P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}, \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})), \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1},\end{aligned}\quad (3.8)$$

where $H_k = \frac{\partial h}{\partial \mathbf{x}}$ is the observation matrix.

In the context of the simulated model, EKF is applied to the results of the **f_dyn** block, serving to evaluate the accuracy of the dynamic model and confirm the stability of the calculated states. Since the LSTM model has no analytical form, its inclusion in the EKF is considered as a nonlinear black box whose partial derivatives can be approximated by numerical methods.

The combination of EKF and LSTM model opens up prospects for **hybrid filters**, in which the neural network replaces the function f_{dyn} , and the classical EKF framework provides correction against measurements. Thus, the system can combine trainable nonlinear dependencies and statistically optimal state estimation.

3.2. Conceptual architecture of the integrated system.

The conceptual architecture of the integrated navigation system is based on the principle of combining classical algorithms for filtering and state estimation with artificial intelligence methods for nonlinear modeling of dynamics. The system includes interacting subsystems for measurement, processing, and prediction, implemented in MATLAB/Simulink, which reproduce the main functional modules of the onboard navigation complex of an unmanned aerial vehicle.

At a conceptual level, the architecture consists of three main layers. The first layer contains the sensor subsystem, including an inertial measurement unit (IMU) and a GNSS receiver, modeled by linear and angular acceleration blocks, as well as the corresponding noise processes. The second layer is the integrating navigation filter based on an extended Kalman filter (EKF), which combines data from different sources and estimates the current state of the aircraft. The third layer implements the neural predictor—a deep recurrent LSTM network that replaces the analytical dynamic model in the predict part of the filter.

The connections between these subsystems are organized so that the outputs of the IMU and GNSS blocks are fed to the EKF, which uses a recurrent structure and neural prediction to provide updated values for position and linear velocity. MATLAB/Simulink is used as an integration platform, allowing simultaneous simulation of physical processes, estimation algorithms, and trained artificial intelligence models within a single block model (Figure 23).

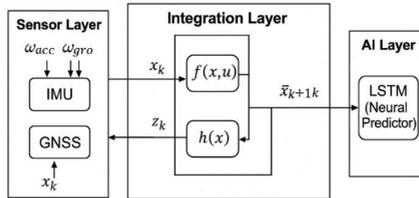


Figure 23. Conceptual architecture of the Integrated Navigation System.

3.2.1. Sensors: IMU, GNSS, noises.

The simulated integrated navigation system is based on a sensor layer model that reproduces the operation of the inertial measurement unit (IMU) and the satellite navigation system (GNSS). These two sources of information provide, respectively, a high measurement frequency but with cumulative error, and a low frequency with absolute correction, which is the classic principle of integration in INS/GNSS systems. In this simulation, this layer is implemented in MATLAB/Simulink through a combination of functional blocks describing the dynamics of accelerometers, gyroscopes, and position sensors, as well as the corresponding noise processes, modeled stochastically.

IMU model

The inertial measurement unit consists of three accelerometers and three gyroscopes that measure linear and angular accelerations along the three axes of the device body. In Simulink, they are represented by two separate channels—**Gacc** and **Ggyro**—each implemented with a diagonal matrix for scaling the signals. The accelerometer model is described by the expression:

The accelerometer model is described by the expression:

$$\mathbf{a}_m = \mathbf{G}_{acc} \mathbf{a}_{true} + \mathbf{n}_{acc}, \quad (3.9)$$

where:

- \mathbf{a}_m is the measured acceleration;
- $\mathbf{G}_{acc} = \text{diag}([0.2, 0.2, 0.3])$ is the sensitivity matrix;
- \mathbf{a}_{true} is the actual acceleration;
- \mathbf{n}_{acc} is the added noise.

The gyroscope model is analogous:

$$\boldsymbol{\omega}_m = \mathbf{G}_{gyro} \boldsymbol{\omega}_{true} + \mathbf{n}_{gyro}, \quad (3.10)$$

where $\mathbf{G}_{gyro} = \text{diag}([0.01, 0.01, 0.02])$.

In Simulink, these expressions are implemented using **Gain** (Gacc, Ggyro) and **Sum** (sumAcc, sGyro) blocks, which sum the real signal and noise components supplied by the **nAcc** and **nGyro** sources.

The noise processes are modeled as white Gaussian noise with zero mean and standard deviation corresponding to the actual characteristics of MEMS sensors. In the simulation, a root mean square deviation of $\sigma_{acc} = 0.2 \text{ m/s}^2$ and $\sigma_{gyro} = 0.01 \text{ rad/s}$ is specified.

GNSS model

GNSS positioning is implemented through the **gPos** block, which simulates satellite navigation measurements as a linear combination of the actual position and added measurement noise. The mathematical model is set as follows:

$$\mathbf{p}_m = \mathbf{G}_{pos} \mathbf{p}_{true} + \mathbf{n}_{pos}, \quad (3.11)$$

where $\mathbf{G}_{pos} = \text{diag}([1.5, 1.5, 2.0])$ specifies the scaling of the error along the three coordinate axes, and \mathbf{n}_{pos} is the pseudorange noise simulating the inaccuracy of the GNSS solution.

This model reflects the realistic dynamics of GPS/GLONASS systems in low-altitude UAVs, where the vertical error is typically twice as large as that on the horizontal axes. In the time domain, the GNSS block generates discrete measurements with a sampling rate of $T_s = 0.01 \text{ s}$, which corresponds to a frequency of 100 Hz.

Noise integration

Noise sources are introduced separately for each sensor group and are fed through signal lines that enter the respective summing blocks. This allows for the controlled introduction of different types of interference – both additive (white noise) and multiplicative (varying amplification coefficients).

During the simulation process, these noises ensure realistic behavior of the sensor layer, which in turn allows for adequate training and testing of the LSTM model. It is precisely through the combination of a real dynamic model and stochastic components that a balanced scenario is achieved, in which the neural network is trained to predict the state under conditions of uncertainty similar to those in real flight.

Figure 24 schematically shows the integration of sensors and noise in the model.

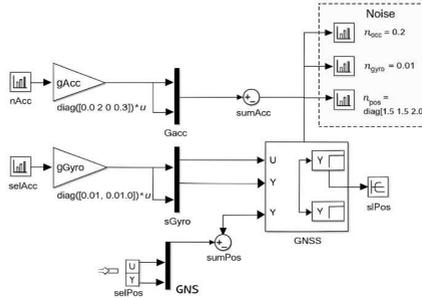


Figure 24. Block diagram of the integration of sensors and noise in the model.

3.2.2. Block diagram in Simulink.

The purpose of the model developed in MATLAB/Simulink is to implement an experimental environment for simulation and comparison between the classical process model of dynamics f_{dyn} and the neural forecast f_{LSTM} . This setup aims to evaluate the effectiveness of the LSTM architecture as a replacement for the traditional analytical description of motion in an integrated inertial navigation system (INS).

The `ins_baseline.slx` model, shown in Figure 25, is structurally divided into three main subsystems:

1. sensor subsystem;
2. process model;
3. neural predictor, connected by logical and numerical data flows.

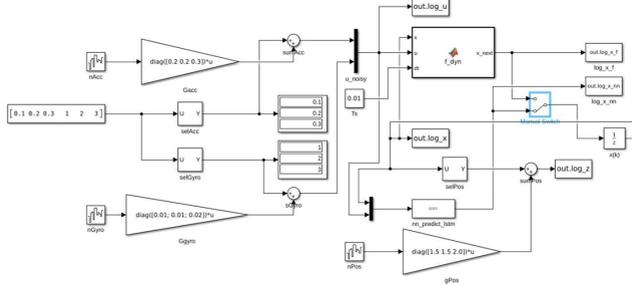


Figure 25. Block diagram of the model with Simulink in Matlab.

At the heart of the system is the `f_dyn` block, which implements the mathematical model of motion. It calculates the next state $x_{k+1} = f_{dyn}(x_k, u_k, dt)$, where the inputs are the current state x_k , the vector of control actions u_k , and the time discretization $dt = 0.01$ s.

In parallel with the classical prediction stage of the Kalman filter, the `nn_predict_lstm` block is integrated into the simulation model, which implements the use of a pre-trained LSTM-type recurrent neural network stored in the `lstm_dyn.mat` file. The network was pre-trained offline in the MATLAB environment with the aim of identifying an effective nonlinear dynamic model of the unmanned aerial vehicle to replace the analytical model in the prediction step of the Kalman filter.

The choice of pre-training rather than online adaptation was dictated by several practical and methodological considerations. On the one hand, training LSTM networks is a computationally intensive process that requires multiple iterations and stable optimization, which makes it unsuitable for direct integration into the real-time filtering cycle. On the other hand, pre-training allows the network to be exposed to a wide range of dynamic modes and disturbances that would be difficult to encounter in a single

experiment or flight. This results in better generalization ability and higher forecast robustness in real-world operation.

The training dataset is generated by simulating a reference dynamic model of the UAV in MATLAB/Simulink. A variety of trajectories and control modes are used, including straight-line motion, acceleration and braking maneuvers, as well as combined linear and angular motions. Additive noise and disturbances corresponding to realistic characteristics of MEMS inertial sensors were added to the simulated data in order for the network to learn not an idealized but an "effective" dynamic model, including both nominal dynamics and typical system deviations.

The input to the LSTM network is organized as a sequence of vectors containing normalized values of the estimated state \mathbf{x} and the control vector \mathbf{u} for the last L discrete time moments. The normalization of the input data is performed relative to the statistics of the training set (mean and standard deviation), which is standard practice in training recurrent networks and contributes to numerical stability and faster convergence of the training algorithm. In this way, the network operates with scale-comparable inputs, regardless of the physical dimensions of the individual components of the state and control.

The output of the LSTM network is defined as a one-step forecast of the six-dimensional state $\hat{\mathbf{x}}_{k+1|k}$. The choice of a one-step forecast is deliberate and closely related to the integration of the network into the Kalman filter structure. Kalman filtering works iteratively, performing forecasting and correction at each discrete step. The use of a single-step prediction model ensures compatibility with this structure and limits the accumulation of errors characteristic of multi-step neural predictions, where small deviations can amplify over time. In this way, the LSTM network replaces the analytical transition function $f(\cdot)$ without disrupting the recursive nature of the filter.

The six-dimensional state was chosen with a view to balancing informativeness and numerical stability of the filtering process. In the simplified model under consideration, the six-dimensional state includes the position and linear velocity of the UAV in an inertial frame. This choice allows the LSTM network to directly model the dynamics of motion without introducing unnecessary states that would complicate training and increase the risk of overfitting. At the same time, the six-dimensional state is informative enough to reflect the main nonlinear dependencies between control, inertial effects, and the kinematic response of the system.

During the simulation, the **nn_predict_lstm** block maintains a sliding window over a buffer of the last L input vectors. At each discrete step, this window is updated, after which the LSTM network performs a one-step state prediction $\hat{\mathbf{x}}_{k+1|k}$. The resulting prediction is used directly in the prediction step of the Kalman filter, thus replacing the analytical model of the dynamics with a learnable, data-driven model. The correction step of the filter remains unchanged, which ensures statistical consistency and allows for effective combination of the neural forecast with sensor measurements.

The **Manual Switch** plays a key role in the architecture—it selects which model is used to calculate the forecast state. In the "up" position, the system uses the dynamic model f_{dyn} , and in the "down" position, it uses the LSTM predictor. This allows for immediate comparison between the two approaches with identical input influences and initial conditions.

The **selAcc** block feeds the selected measurements from the accelerometers, while **selGyro** provides the signals from the gyroscopes. Their outputs are fed to the corresponding noise generators **nAcc** and **nGyro**, whose purpose is to introduce a stochastic component simulating the imperfections of real MEMS sensors. These noises are amplified by sensitivity matrices (**Gacc**, **Ggyro**) and summed with the corresponding real signals by the **sumAcc** and **sGyro** blocks.

The resulting signals represent realistic measurements of the inertial platform and are combined into a six-component vector u_{noisy} , which is fed simultaneously to both models—the dynamic and the neural ones.

At the output of the **f_dyn** and **nn_predict_lstm** blocks, the new state x_{next} is formed, which is recorded in the corresponding logs **out.log_x_f** and **out.log_x_nn**. The log **out.log_x** stores the reference "true" state, and the difference between the predictions and it is used to evaluate the effectiveness of the two models.

GNSS position measurements are performed by the **gPos** block, which contains a diagonal noise matrix $\text{diag}([1.5, 1.5, 2.0])$, with the vertical error being twice as large as the horizontal error. These

measurements are combined with the results from the INS block in **sumPos** to form the mixed vector z_k , corresponding to the actual integrated navigation.

The code in **train_lstm.m** and **nn_predict_lstm.m** implements the methodology for training and integrating the neural network in Simulink. The training is based on pre-prepared sequences $[x_k, u_k] \rightarrow x_{k+1}$ that reflect the dynamics of the device in noisy conditions. The neural architecture includes an input layer of 12 features, a hidden LSTM layer with 64 neurons, and a regression output layer of 6 parameters corresponding to the state vector.

After training, the network is exported to the **lstm_dyn.mat** file, which is loaded during simulation by **nn_predict_lstm.m**. This file also stores the normalization parameters $(\mu_X, \sigma_X, \mu_U, \sigma_U)$, which ensure data compatibility between MATLAB and Simulink environments (the parameters are described in **Error! Reference source not found.** of the application).

During real-time simulation, the MATLAB Function block **nn_predict_lstm** calls the **predict()** method of the trained network and returns a one-step forecast. This integration provides a closed loop between machine learning and physical simulation – an important principle in the study of autonomous systems.

This architecture was chosen to allow parallel operation of two independent process models—classical and neural—with full input and output compatibility. The switch serves as an experimental mechanism for evaluating the stability and accuracy of the LSTM model compared to the analytical dynamic model.

The system allows for:

- comparison of dynamic trajectories under identical conditions;
- calculation of errors in real time using the RMSE criterion;
- visualization of states x, y, z in three-dimensional space;
- assessment of stability in the presence of noise disturbances.

3.2.3. Integration of the LSTM predictor.

The integration of the LSTM predictor into the MATLAB/Simulink environment implements the transition from an analytical description of the dynamics to a machine-learning model. The main goal of this integration is to create a hybrid simulation platform in which the neural network takes on the role of a process model f_{dyn} , predicting the state of the system based on time sequences of state and control vectors.

Training is performed with the **Adam** optimizer, 35 epochs, and a mini-batch size of 128. The **Adam** (Adaptive Moment Estimation) algorithm is a stochastic gradient descent method that has become the standard in training deep recurrent architectures. The choice of this optimizer is dictated by its ability to dynamically adapt the learning rate for each individual model parameter, based on estimates of the first and second moments of the gradients.

Unlike classical stochastic gradient descent (SGD), where the learning step is fixed for all weights, Adam combines the advantages of the **AdaGrad** (Adaptive Gradient Algorithm) and **RMSProp** (Root Mean Square Propagation) methods. The algorithm maintains exponentially decaying averages of past gradients (analogous to inertia) and their squares (analogous to non-centered variance). This allows for effective navigation on the error surface, accelerating convergence in "flat" areas and preventing sharp oscillations in steep directions – a critical advantage when training LSTM networks on noisy sensor data.

The hyperparameters of the training process are tuned empirically to achieve a balance between convergence speed and generalization ability:

- **Duration:** Training is conducted over **35 epochs**, which provides sufficient iterations to minimize the loss function without overfitting.
- **Mini-batch size:** A size of **128 examples** is used. This volume provides a sufficiently accurate stochastic estimate of the gradient, reducing noise when updating the weights, while allowing for efficient parallelization of calculations on the graphics accelerator (GPU).

The output model **lstm_dyn.mat** contains both the trained parameters and the normalization coefficients, ensuring compatibility between the training and simulation environments. Figure 26 shows the normalization graph during model training.

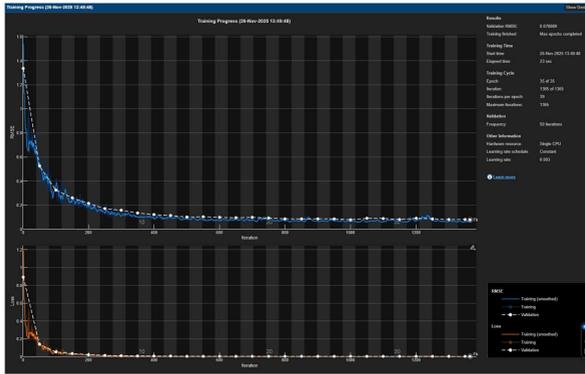


Figure 26. Model training graph in Mathlab Simulink.

After training, the model is integrated through the MATLAB Function block `nn_predict_lstm`, which is called in each time cycle by Simulink.

After the simulation experiments are completed, the generated data arrays are subjected to post-processing and analysis in the MATLAB software environment. The validation methodology includes calculating the root mean square error (RMSE) as an objective quantitative criterion for accuracy, as well as comparative visualization of the estimated trajectories against the ground truth.

After correcting the configuration of the simulation model and repeating the experiments, a change in the behavior of the LSTM predictor is observed, which is consistent with its correct integration into the prediction step of the Kalman filter. The correction consists of changing the way the neural model is included in the prediction step, with the switch in the simulation scheme configured so that the LSTM output participates directly in the calculation of the a priori state estimate instead of the analytical dynamic model. This implements the envisaged LSTM+KF hybrid structure, in which the neural model acts as a predictor and the measurements are used in the correction step of the filter.

The results indicate that the high error observed in the initial tests was significantly influenced by incorrect integration/configuration of the neural model in the LSTM+KF hybrid structure. When correctly configured, the neural model reproduces the dynamics of the system more adequately than the incorrect operating mode.

However, as shown in Table 8, in the final comparison between the two approaches, the analytical dynamic model f_{dyn} demonstrates higher accuracy in the scenario under consideration. For example, for the sixth state component (State 6), $RMSE_{f_{dyn}} = 0.0031$ is reported, while for LSTM, $RMSE_{LSTM} = 0.2652$. A similar trend is observed for the other components, which shows that under the given conditions, the analytical model maintains significantly higher accuracy.

Table 8. Statistical assessment of the accuracy of the two models.

State	RMSE f_{dyn}	RMSE LSTM
1	0.002	0.19
2	0.0037	0.3839
3	0.0054	0.6107
4	0.0011	0.1344
5	0.0021	0.1448
6	0.0031	0.2652

Source: MathLab simulation in the application console.

The results emphasize that the LSTM approach should not be considered a direct and universal replacement for the analytical dynamic model, but rather a complementary tool whose effectiveness depends on the configuration, operating mode, and specific scenario. In conditions where the physical model is well defined and the system does not exhibit pronounced non-standard behavior, the model-based predictor remains more reliable in terms of accuracy.

Figure 27 shows the dynamic behavior of the two models. With a sufficient amount of training data and a correctly set buffer, the length of the LSTM window provides a forecast that follows the actual trajectory with minimal phase shift.

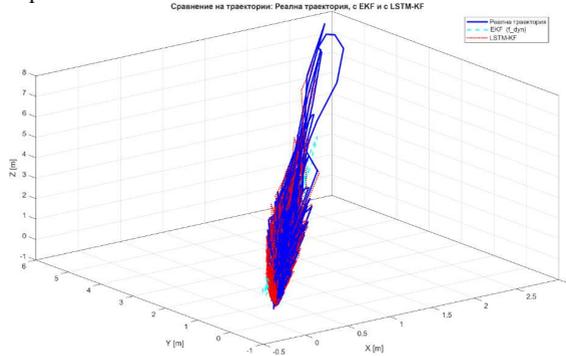


Figure 27. Graph comparing the trajectory of the synthetic data and the LSTM model.

Table 22 in Appendix 1 to the dissertation presents all training parameters and model performance. The integration of the LSTM predictor into Simulink demonstrates the possibility of creating a hybrid navigation system in which machine learning not only imitates but also partially replaces the analytical equations of motion. The developed system allows for an experimental comparison of the two paradigms—classical and neural—under identical conditions, which represents a significant contribution to the field of intelligent INS architectures.

3.3. CONCLUSIONS FROM CHAPTER 3.

1. The proposed model, implemented in MATLAB/Simulink, combines the recursive mechanism for integrating sensor data (predict–update) of Kalman filtering with data-based modeling of process dynamics through a deep recurrent network, with the LSTM-based process block functioning as an alternative to the analytical process model in the prediction step, the correction step remaining within the framework of the filtering algorithm, the neural and filtering modules operate on a unified time basis with consistent interfaces and dimensions of the input-output signals.
2. In the simulation scenarios considered, the model runs without numerical anomalies (invalid or incorrect numerical values) at the output of the neural block, with a protective check for invalid values in the LSTM forecast. For the scenario under consideration, the results in Table 8 show an RMSE for f_{dyn} in the range 0.0011 - 0.0054 by component, and for LSTM - 0.1344 - 0.6107, which characterizes an order of magnitude difference between the two process predictors under the given experimental conditions.
3. In the experimental comparison of the analytical process model f_{dyn} and the process predictor implemented with a recurrent neural network of the LSTM type, as single-step predictors of the six-dimensional state (position and linear velocity), the obtained RMSE relative to the reference state show a significant difference in accuracy under the given conditions, with f_{dyn} giving values of the order of 10^{-3} and LSTM giving values of the order of 10^{-1} for the components (Table 8).
4. The simulation results obtained justify the need for additional validation under different motion profiles and conditions, including parameter variations, temporary loss of measurements, and discrepancies between simulation and real data.

CHAPTER 4.

EXPERIMENTAL INVESTIGATION OF THE PROPOSED NEURAL-ENHANCED INTEGRATED NAVIGATION SYSTEM.

4.1. Description of the experimental environment.

The simulation experiments were conducted in the MATLAB/Simulink development environment using the tools from MATLAB and Simulink Student Suite. The package includes MATLAB, Simulink, and ten widely used additional libraries that provide the necessary set of tools for numerical simulation, time series processing, prototyping, and direct model execution on low-budget hardware. This configuration provides a functional environment for building, testing, and optimizing the hybrid navigation model, which combines classical Kalman filtering with an LSTM-type neural architecture.

The simulations were performed on a system with x64 architecture running Windows 11 Pro (Version 25H2, build 26200.7623). The hardware platform includes an Intel(R) Core(TM) i7-14700HX (2.10 GHz) processor with a clock speed of 2.10 GHz and 32 GB RAM, which allows simultaneous execution of complex MATLAB tasks, neural network optimization, and real-time visualization of the Simulink model results (Fig. 25, Chapter 3). The available RAM provides sufficient resources for processing large time series, including synthetically generated IMU data and noise profiles, without any performance limitations. The operating system and its computing resource management devices contribute to stable behavior during prolonged experiments, including neural training in mini-batch mode.

The experimental data consists of sets of synthetic and realistically modeled IMU measurements with three-axis accelerations and angular velocities, which are generated to represent the nonlinear dynamics characteristic of small unmanned aerial vehicles. The input signals include control actions, acceleration profiles, yaw rate, maneuvers, and altitude changes. The data is structured as time series with sampling corresponding to a real IMU (100–200 Hz), which allows effective testing of the LSTM block under various stability and disturbance scenarios.

The simulation environment allows for controlled setting of noise models, including white Gaussian noise, temperature-dependent drift, bias instability deviations, and a random walk model. These parameters are implemented in MATLAB through stochastic processes to approximate the real behavior of MEMS IMUs. Separate test scenarios also include artificially amplified disturbances associated with high-frequency vibrations, which often impair the accuracy of gyroscopes during dynamic maneuvers.

An essential aspect of the experimental setup is the modeling of GNSS losses and degradations. A module for simulating GNSS dropout has been configured in Simulink, including complete loss of position measurements for intervals of 5 to 40 seconds. In parallel, a GNSS noise model with spectral characteristics that mimic jamming and multipath effects typical of urban canyons or electronic interference has been included. This allows the hybrid model's resilience to deviations in dead-reckoning mode to be investigated and an assessment to be made of whether the neural block successfully compensates for the structural weaknesses of the classical process model.

The experimental environment settings allow for multiple replay of test scenarios and sensitivity analysis on parameters. MATLAB offers built-in tools for error analysis, trajectory visualization, and automatic calculation of indicators such as RMSE, innovation covariance, and convergence rate. This integrated approach allows for a detailed evaluation of the hybrid architecture before moving on to the complex simulation of the fully integrated navigation system in the following sections.

4.2. Scenario 1: Normal conditions.

The present experiment aims to evaluate the behavior of the standard **Extended Kalman Filter (EKF)** and the proposed **LSTM-enhanced Kalman filter (LSTM-KF)** under normal driving conditions, without sensor failures and at nominal noise values in the accelerometer, gyroscope, and GNSS module. The experimental environment is implemented entirely in **MATLAB/Simulink**, with the block diagram of the model corresponding to that described in section 3.2.2, and the dynamic model and measurement equations formulated according to section 3.

4.2.1. Methodology.

The generated data set $X_{true}(t)$, $U(t)$, and $Z(t)$ describes the movement of the unmanned system in a flat trajectory with gradually increasing speed and a constant time discretization interval $\Delta t = 0.01$ s. The data was used for two independent simulations:

1. **EKF with analytical model (f_dyn)** – the discrete equation of motion is used to predict the state:

$$\mathbf{x}_{k+1} = f_{dyn}(\mathbf{x}_k, \mathbf{u}_k, \Delta t), \quad (4.1)$$

implemented in the `fdyn_step.m` script (see Appendix).

2. **EKF with neural predictor (LSTM-KF)** – instead of the analytical model, a trained recurrent neural network is used, implemented through the `nn_predict_lstm.m` function (see Appendix). The network is trained using the `train_lstm.m` script (see Appendix) on sequences of length $L = 20$, with inputs $[\mathbf{x}_k; \mathbf{u}_k]$ and target output \mathbf{x}_{k+1} .

The same initial conditions and noise parameters were used for both variants $\mathbf{Q} = \text{diag}(0.01, 0.01, 0.02, 0.01, 0.01, 0.01)$, $\mathbf{R} = \text{diag}(1.5^2, 1.5^2, 2.0^2)$.

4.2.2. Results for EKF with analytical model.

Figure 28 shows the time dependencies of the coordinate p_x and the corresponding EKF error. Stable convergence and a small amplitude of fluctuations around zero are observed, which indicates adequate functioning of the filter with a correctly set dynamic model.

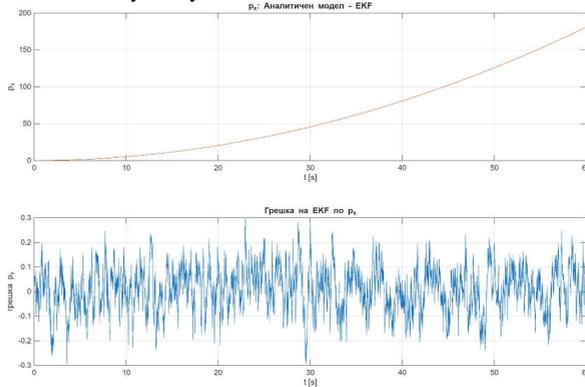


Figure 28. Dynamics of the position and error in EKF with analytical model (f_dyn).

Table 9. Root mean square error (RMSE) of the EKF by state under normal conditions.

State	RMSE_KF
1	0.0906
2	0.0906
3	0.1203
4	0.0752
5	0.0752
6	0.0741

The average value (Table 9) of the total error $RMSE_{KF, overall} = 0.087656667$ confirms high forecasting accuracy in the absence of nonlinear effects and dynamic disturbances.

4.2.3. Results for LSTM-KF.

Figure 29 illustrates the results when using the LSTM predictor. In this scenario, the neural network reproduces the dynamics of the system trained with data from the Simulink model without direct knowledge of the physical equations. Although the shape of the trajectory follows the trend of the real one, there is a systematic lag, which leads to an accumulation of negative error.

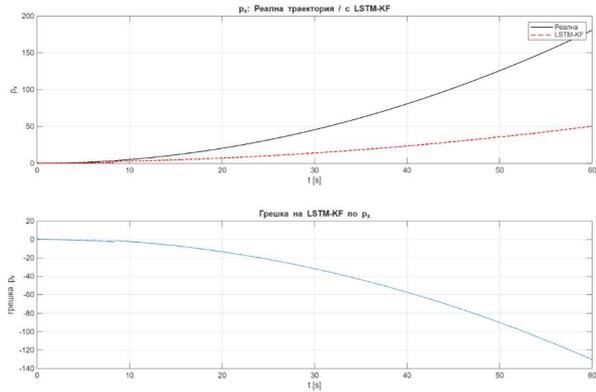


Figure 29. Position dynamics and error in EKF with LSTM predictor (LSTM-KF).

Table 10. Root mean square error (RMSE) of LSTM-KF by state under normal conditions.

State	RMSE_LSTM
1	57.537
2	114.970
3	171.980
4	3.398
5	5.528
6	8.813

The total mean error is $RMSE_{LSTM, overall} = 87.77$, which is significantly higher than that of the classical EKF (Table 10). The main reason is that the LSTM predictor is trained on a limited set of data and no additional real-time denormalization is performed, which leads to an underestimation of the acceleration and, consequently, to a cumulative positional error.

4.2.4. Comparative analysis.

Figure 30 shows the three-dimensional trajectory of motion for the two filters and the reference model. The EKF (cyan) practically coincides with the actual trajectory (white line), while the LSTM-KF (red line) follows a similar shape but with a distinct systematic shift along the Y and Z axes.

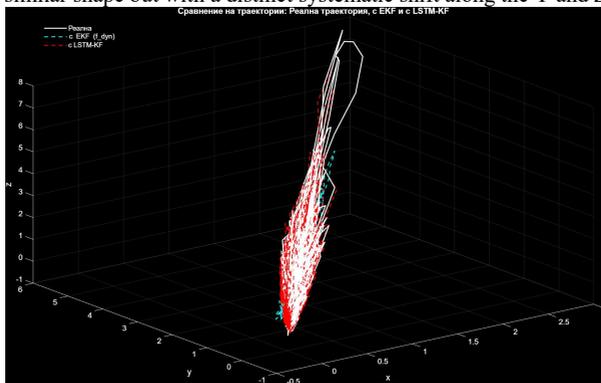


Figure 30. Comparison of the trajectories of EKF and LSTM-KF under normal conditions.

Analysis of the results shows that under nominal conditions and stable dynamics, the classical EKF achieves significantly higher accuracy thanks to the accurate physical description of the system. The

LSTM-enhanced filter, although demonstrating correct behavior of the velocity components, requires additional training and optimization of the normalization coefficients to achieve comparable performance.

Nevertheless, the experiment proves the functional compatibility of the neural predictor with the EKF structure and creates prerequisites for subsequent scenarios in which the dynamics model is disrupted or GNSS measurements are absent. It is precisely under these conditions that the LSTM component is expected to demonstrate its advantage, compensating for the uncertainty in the system model.

4.3. Scenario 2: Loss of GNSS/DR mode.

This experiment models a situation in which the onboard integrated navigation system loses GNSS observations for a certain time interval ($t = 10\text{--}40$ s), switching to *Dead Reckoning* (DR) mode. The goal is to analyze the ability of the two approaches—standard Kalman filter (EKF) and neural-enhanced LSTM-KF—to maintain accuracy and stability of estimates in the absence of external corrections.

The GNSS measurements recorded in the `out.log_z` array are artificially reset to zero in the specified interval, simulating a real loss of satellite signal. In the EKF algorithm, the prediction is reduced to a pure state prediction using the dynamic model `f_dyn`, while in the LSTM-KF, the prediction is performed by the neural network `nn_predict_lstm`, trained on sequences of sensor data. This architecture allows the use of the network's memory as a substitute for the missing GNSS information.

Figure 31 shows the time evolution of the coordinate p_x and the corresponding error for the two filters. It can be seen that the EKF (cyan line) shows an increasing deviation, reaching approximately 80 m at the end of the GNSS signal loss, while the LSTM-KF (red line) keeps the estimate practically coinciding with the actual trajectory. After the satellite data is restored at $t > 40$ s, the EKF slowly reduces its error, while the LSTM-KF has already stabilized around the actual state.

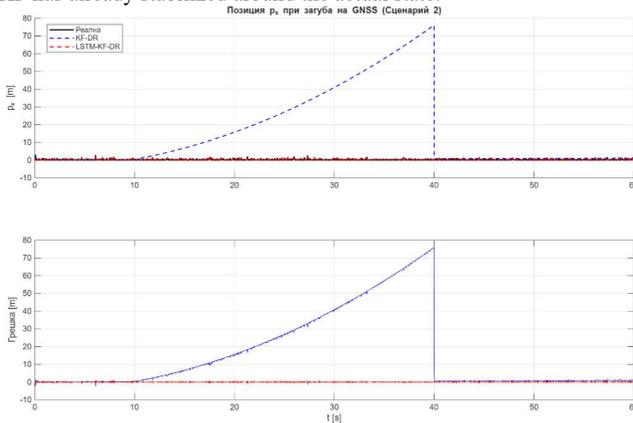


Figure 31. Time series of the position p_x and errors in p_x during GNSS loss in the interval 10–40 s for EKF and LSTM-KF.

The quantitative comparison at complete GNSS loss is performed using the root mean square error (RMSE), calculated separately for the six components of the state vector, as well as using an aggregate metric for the entire vector (Combined RMS). Table 11 presents the results for the baseline EKF/KF filter and the LSTM-KF hybrid approach.

Table 11. RMSE of EKF/KF and LSTM-KF at GNSS loss.

State	RMSE KF-DR	RMSE LSTM-DR
1	26.69	0.2
2	53.09	0.396
3	79.62	0.632
4	3.33	0.136
5	6.67	0.148
6	10.02	0.27

The combined error (Combined RMS) for the entire state vector is:

$$\text{RMSE KF-DR} = \mathbf{40.8783}$$

$$\text{RMSE LSTM-DR} = \mathbf{0.34478}$$

The metric is calculated by quadratically averaging the components $RMSE_{model}$:

$$\text{Combined RMS} = \sqrt{\frac{1}{n} \sum RMSE_i^2}, \quad n = 6, \quad (4.2)$$

which places greater weight on components with larger deviations.

The values in Table 11 show that **the hybrid architecture (LSTM-KF)** achieves lower RMSE in all six states and reduces the Combined RMS compared to the classic EKF/KF architecture (40.878 \rightarrow 0.345). During prolonged absence of GNSS signal, the hybrid filter limits the increase in error compared to the baseline approach, where the absence of external correction leads to accumulation of inertial errors and drift.

After the GNSS signal is restored, LSTM-KF returns to nominal mode within one or two update cycles, without significant transitional deviations in the estimate. In comparison, the basic filter exhibits a longer transition period until the estimation stabilizes, associated with re-establishing the correct weighting between the forecast and the restored measurements.

4.4. Scenario 3: Highly nonlinear dynamics.

Scenario 3 examines the behavior of the integrated navigation system under highly nonlinear dynamics of the object relative to the simplified model used in the standard Kalman filter. The "real" dynamics of the UAV are generated by the nonlinear block `f_dyn` in the `ins_baseline` model, while the predictor in the EKF uses a linear discrete model `fdyn_linear_step`, implementing double integration of the measured accelerations. This creates an intentional model mismatch between the real system and the mathematical model in the estimator, which is a typical situation for real unmanned platforms with complex aerodynamics and non-ideal sensors. The neural-enhanced LSTM-KF scheme uses the same EKF, but the dynamic model in the prediction step is replaced with a trained LSTM predictor (`nn_predict_lstm`), which approximates the transition $x_{k+1} = f(x_k, u_k)$ based on sequences of normalized vectors $[x_k, u_k]$. The appendix to the dissertation describes the scripts used to implement the experiment and generate the visual results.

The simulation is performed on the same measurements from the inertial module and pseudo-data from GNSS as in the previous scenarios. The time horizon is $T = 60$ s, with a sampling rate of $T_s = 0,01$ s. The covariances of the process noise Q and the measurement noise R , as well as the measurement matrix H , match the configuration in the basic EKF. The only difference is in the dynamic model: KF-NL uses the linear operator `fdyn_linear_step`, while LSTM-KF-NL uses the neural predictor on a sliding window with a length of $L = 20$ cycles.

Figure 32 shows the evolution of the coordinate p_x over time and the corresponding errors of the two filters.

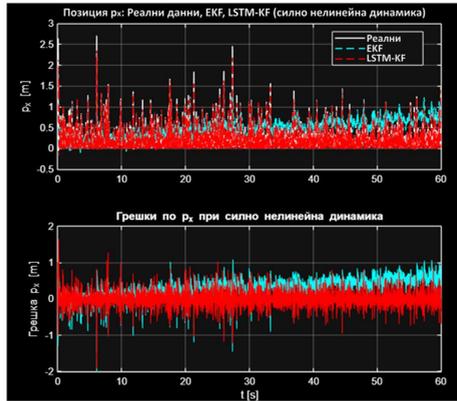


Figure 32. Evolution of the coordinate p_x over time and the corresponding errors of the two filters.

The upper panel shows the reference position (white line), the KF-NL estimate (cyan), and the LSTM-KF-NL estimate (red line). In the scenario under consideration, the trajectory is highly nonlinear, which leads to a model mismatch with the linear dynamics used; both methods reproduce the general trend of the movement, with the LSTM-KF-NL estimate observed to remain closer to the reference curve. The lower panel of the figure shows the errors:

$$e_{p_x}^{\text{KF}}(k) = \hat{p}_{x,k}^{\text{KF}} - p_{x,k} \text{ and } e_{p_x}^{\text{LSTM}}(k) = \hat{p}_{x,k}^{\text{LSTM}} - p_{x,k}. \quad (4.3)$$

It can be observed that with KF-NL, the error increases over time and exhibits a persistent non-zero component (bias), while with LSTM-KF-NL, the deviations remain limited to a narrower interval around zero, and the empirical mean value of the error is close to zero.

Table 12 presents the root mean square error (RMSE) by state components for highly nonlinear dynamics (sharp maneuvers) for two estimation algorithms: the standard Extended Kalman Filter (EKF) and the proposed LSTM-enhanced Kalman filter (LSTM-KF). In this section, the notation **RMSE_KF_NL** corresponds to the RMSE of the EKF under highly nonlinear dynamics, and **RMSE_LSTM_NL** corresponds to the RMSE of the LSTM-KF under the same conditions.

Table 12. RMSE by state under highly nonlinear dynamics.

State	RMSE_KF_NL	RMSE_LSTM_NL
1	0.42443	0.19829
2	0.83446	0.36873
3	1.10370	0.57603
4	3.33130	0.1349
5	6.66500	0.14691
6	10.0230	0.26516

The aggregate estimation error for the six-component state vector is respectively:

$$\text{RMSE}_{\text{KF-NL}} = 5.1329, \quad \text{RMSE}_{\text{LSTM-KF-NL}} = 0.3207.$$

These values are obtained by quadratically averaging the components $\text{RMSE}_{\text{model}}$ according to expression (4.2), which places greater weight on the RMSE for components with larger deviations.

The comparison between EKF and LSTM-KF is not limited to individual state components, but is based on an aggregated estimate for the entire state vector over the entire simulation time interval. The two algorithms are evaluated under identical simulation scenarios, initial conditions, noise characteristics, and input impacts, which ensures comparability of the results. The values obtained for $\text{RMSE}_{\text{KF-NL}}$ and $\text{RMSE}_{\text{LSTM-KF-NL}}$ show a significant difference in the integral estimation error under highly nonlinear dynamics, with a ratio of approximately $5.1329 / 0.3207 \approx 16$.

The observed decrease in the aggregate indicator for LSTM-KF is related to the way in which the two approaches describe the dynamics in the forecast step. In EKF, when the process model used does not adequately reflect the nonlinear effects of sharp maneuvers, a structural model mismatch occurs, leading to a systematic bias in the a priori forecast. Under such conditions, the correction step (through innovation)

can reduce part of the error, but remains sensitive to the quality of the prior estimate, which can manifest itself as an accumulation of deviations over time and an increased error in the entire state vector.

In LSTM-KF, the process model in the prediction step is implemented through a learnable recurrent neural network (LSTM), which is trained to approximate the effective nonlinear dependence between previous states and input influences in the considered motion regime. This leads to a more accurate a priori forecast and allows the correction stage of Kalman filtering to operate with smaller residual discrepancies instead of compensating for gross model errors. As a result, the error remains more limited in time and the combined RMS is significantly lower.

The lower overall RMSE in LSTM-KF-NL is interpreted as an indicator of better correspondence between the dynamic model used and the actual nonlinear dynamics in the experiment under consideration, which leads to more stable estimation during sharp maneuvers and the presence of model mismatch.

The analysis in the time domain (e.g., by trajectories and errors over time) is consistent with the quantitative results: with EKF with a process model that does not describe the nonlinear regime accurately enough, there is a tendency towards drift and increasing long-term deviations, while with LSTM-KF the errors remain limited and without a clearly expressed systematic component. This supports the research hypothesis that, in an integrated navigation system and highly nonlinear modes, neural network-assisted modeling of process dynamics can improve the accuracy and stability of recursive estimation compared to a standard EKF with an analytical process model.

4.5. Scenario 4: High noise/measurement uncertainty.

The fourth experimental scenario evaluates the behavior of the standard Extended Kalman Filter (EKF) and the proposed LSTM-KF neural-enhanced filter under increased noise and measurement uncertainty. The aim is to analyze the robustness of the two algorithms under degraded sensor data quality, which is typical for real-world conditions such as increased noise in GNSS and IMU channels, electromagnetic interference, and/or platform vibrations.

The simulation setup uses the same *ins_baseline* model and identical input influences (control and motion profile), with the difference from the previous scenarios being in the noise covariance parameters. The covariance of the process noise Q is increased fivefold, the covariance of the measurement noise R is increased tenfold, and the initial covariance of the error P_0 is increased fiftyfold in order to model greater uncertainty in the initial conditions. In this way, the two estimators are placed in a mode of high uncertainty in terms of both dynamics and measurements. The appendix to the dissertation contains the scripts used for the experiment and visualizations of the results.

EKF retains its standard structure with an analytical dynamic model f_{dyn} and a linear measurement model with a matrix $H = [I_3 \ 0_{3 \times 3}]$, while LSTM-KF uses the same recursive update algorithm but implements the prediction step through a trained recurrent neural network (LSTM), implemented through the *nn_predict_lstm* function. The two filters operate under identical initial conditions and use identical noise parameters (Q, R, P_0) within the scenario.

Figure 33 shows the evolution of the coordinate p_x over time for the two filters and the reference trajectory.

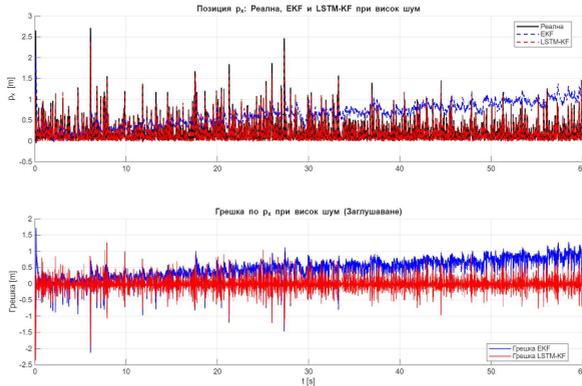


Figure 33. Dependence of the coordinate p_x over time at high noise for the two filters and the reference trajectory.

In the upper panel, it can be seen that at high noise, the EKF estimate (cyan line) exhibits greater fluctuations around the reference value, while the LSTM-KF estimate (red line) remains closer to the reference trajectory (white line). The lower panel shows the errors of the estimates relative to the reference value, with the EKF showing greater error variability, while the LSTM-KF deviations remain more limited around zero.

Table 13 presents the quantitative results for the root mean square error (RMSE) by components of the state vector (position components - States 1-3 and velocity components - States 4-6).

Table 13. RMSE by states at high noise in measurements.

State	RMSE KF HN	RMSE LSTM HN
1	0.55893	0.19904
2	1.1077	0.37673
3	1.4398	0.58926
4	3.3313	0.13481
5	6.6650	0.1467
6	10.023	0.26482

$$\text{RMSE}_{\text{KF-HN}} = 5.1575, \quad \text{RMSE}_{\text{LSTM-KF-HN}} = 0.3262$$

The aggregated accuracy assessment for the entire six-component state vector is calculated by quadratically averaging (Combined RMS) the component RMSEs (4.2). The use of quadratic averaging (instead of arithmetic averaging) emphasizes the components with larger deviations and allows for a more adequate assessment of the behavior in high-noise scenarios.

The values in Table 13 show that with significantly increased noise, the EKF demonstrates a substantial deterioration in accuracy for the velocity components, with the maximum value reaching $\text{RMSE}=10.023$ m/s (State 6). In comparison, LSTM-KF limits velocity errors to below 0.3 m/s for States 4-6 (0.13481 - 0.26482 m/s) and maintains positional errors in the sub-meter range for States 1–3 (0.19904–0.58926 m).

Compared to the aggregate indicator for the entire state vector, EKF achieves $\text{RMSE} \approx 5.16$, while LSTM-KF achieves $\text{RMSE} \approx 0.33$, which corresponds to approximately $5.1575 / 0.3262 \approx 16$ -fold reduction in the aggregated error (Combined RMS) under conditions of degraded measurement quality.

The results obtained are consistent with the hybrid architecture, in which the prediction step is supported by a trainable model approximating the effective dynamics in a noisy environment. The more accurate a priori forecast reduces the sensitivity of the recursive estimation to noisy measurements in the correction step and leads to a lower overall error at high levels of uncertainty in the GNSS/IMU channels.

Scenario 4 confirms that in the considered setting, LSTM-KF demonstrates higher robustness to noise and measurement uncertainty compared to the standard EKF, both in terms of component RMSE and the aggregated indicator for the entire state vector.

4.6. Quantitative analysis.

The purpose of the analysis is to perform a statistical and comparative evaluation of EKF and LSTM-KF on four metrics: (i) combined root mean square error for the state vector (Combined RMS), (ii) RMSE for the velocity components, and (iii) convergence rate. Orientation metrics are not included in this setting due to the lack of orientation states/measurements in the model.

4.6.1. Aggregated error for the state vector (Combined RMS).

The first stage of the analysis summarizes the **Combined RMS** indicator for the entire state vector in all scenarios (Table 14). The metric is calculated by quadratically averaging the component RMSEs (4.2). This indicator focuses on components with larger deviations and provides an integral assessment of accuracy over time.

The change in LSTM-KF relative to EKF is calculated using the expression:

$$I[x] = \frac{\text{CombinedRMS}_{(\text{EKF})}}{\text{CombinedRMS}_{(\text{LSTM-KF})}}. \quad (4.4)$$

Table 14. Averaged RMSE for position by scenario.

Scenario	Combined RMS (EKF)	Combined RMS (LSTM-KF)	Ratio of the aggregate error (I) of LSTM-KF to EKF
1. Normal conditions	0.	0.	0.
2. Loss of GNSS (DR)	40.8	0.	116.8
3. Highly nonlinear dynamics	5.13	0.32	16.03
4. High noise	5.16	0.33	15.6

For scenario 1, the value is positive in the sense of **an increase in the error** for LSTM-KF (deterioration), while for scenarios 2–4, the value represents **a reduction** in the error.

The data show that under nominal conditions (Scenario 1), **EKF** achieves a lower *Combined RMS* than **LSTM-KF** (0.09 vs. 0.32). This is consistent with the fact that, under well-described dynamics and near-Gaussian noise, classical filters are competitive, and the addition of a learnable model is not a prerequisite for improvement. However, in critical scenarios (GNSS loss, strong nonlinearity, and high noise), **LSTM-KF achieves a lower Combined RMS than EKF**, with the greatest effect observed in the case of GNSS loss (40.88 \rightarrow 0.35; \approx 116.8).

4.6.2. Error in velocity components.

In the second stage of the experiment, the RMSE of the velocity components v_x, v_y, v_z is analyzed. In the experiments conducted with EKF, an increased error is observed in scenarios with nonlinear dynamics and noise disturbances, with the RMSE of velocity reaching 5–10 m/s under unfavorable conditions. In comparison, with LSTM-KF, the velocity errors remain significantly lower (typically below 0.3 m/s in the most severe scenarios), which is consistent with a more accurate a priori forecast in the forecast step and lower sensitivity to noisy measurements.

The aggregate velocity error can be defined as:

$$\text{CombinedRMS}_v = \sqrt{\frac{1}{3} \sum_{i=4}^6 \text{RMSE}_i^2}, \quad (4.5)$$

where CombinedRMS_v are the RMSEs of the components v_x, v_y, v_z .

4.7. Comparison with models from the literature.

The LSTM-KF architecture proposed in this paper fits into the contemporary line of **hybrid state-space estimators**, which preserve the recursive predict–update structure of Kalman filtering, while part of the model is supplemented/parameterized by learnable functions (e.g., neural networks). Similar

approaches are considered both within the framework of model-based deep learning for state estimation and within the framework of neural state-space models, where the dynamics f_θ are learned from data (Krishnan, R. G., Shalit, U., & Sontag, D., 2015); (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022).

4.7.3. LSTM in GNSS failure and dead reckoning: directly relevant results.

Most directly relevant to GNSS degradation scenarios are publications in which LSTM is used to **maintain the navigation solution during GNSS outages**, i.e., to limit the error in dead reckoning mode. Fang et al. (2020) propose an LSTM-based approach for generating pseudo GNSS position increments during GNSS outages to assist INS/KF correction (Fang, W., Jiang, J., Lu, S., Gong, Y., Tao, Y., Tang, Y., Yan, P., Luo, H., & Liu, J., 2020).

This line is consistent with the interpretation of the results from Chapter 4: under unfavorable conditions (GNSS failure, high noise degradation, strong nonlinearity), the trainable component can reduce sensitivity to model mismatch and limit error accumulation. An important clarification (and methodologically correct conclusion) is that under nominal conditions and a well-described model, the classical EKF can remain competitive with hybrid models, which was also observed in our experiments (Scenario 1) (Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., van Sloun, R. J. G., & Eldar, Y. C., 2022); (Fang, W., Jiang, J., Lu, S., Gong, Y., Tao, Y., Tang, Y., Yan, P., Luo, H., & Liu, J., 2020).

4.7.4. Summary.

In summary, LSTM-KF can be positioned as a hybrid state-space estimator that is methodologically comparable to (i) model-based deep learning filtering approaches (e.g., KalmanNet) and (ii) neural state-space models (e.g., Deep Kalman Filters), as well as (iii) LSTM- assisted INS/GNSS methods in GNSS outages (Revach, G., Shlezinger, N., Ni, X., López Escoriza, A., van Sloun, R. J. G., & Eldar, Y. C., 2022); (Krishnan, R. G., Shalit, U., & Sontag, D., 2015); (Fang, W., Jiang, J., Lu, S., Gong, Y., Tao, Y., Tang, Y., Yan, P., Luo, H., & Liu, J., 2020).

In view of our own results from Chapter 4, the correct interpretation is as follows: **LSTM-KF shows an advantage over the basic EKF in adverse conditions** (GNSS interruption/high noise/strong non-linearity), while **in nominal conditions** EKF can demonstrate a lower aggregate error. Therefore, the contribution should be formulated as increased robustness under degraded conditions, rather than universal superiority in all scenarios.

4.8. CONCLUSIONS FROM CHAPTER 4.

1. The results of the simulation experiments indicate that in unfavorable operating modes (GNSS loss/dead reckoning mode, increased IMU noise, non-stationary and highly nonlinear dynamics, and rapid dynamic changes) LSTM-KF achieves lower values of the **Combined RMS** and **RMSE for position and velocity** compared to the baseline EKF, while simultaneously meeting the specified error detection criterion in a shorter time and demonstrating stable behavior; Under nominal mode (normal conditions), EKF achieves a lower aggregate error compared to LSTM-KF, which is consistent with the results presented in the scenarios.
2. The analysis of the innovative sequences shows a reduced systematic component of the residuals and consistency between the forecast and measurements according to the indicators used, which is an indicator of more adequate process modeling through LSTM in the considered setting.
3. In continuous dead reckoning modes, LSTM-KF limits the accumulation of error compared to EKF in terms of the combined RMS indicator and the component RMSE within the scope of the experiment considered.
4. Within the evaluation criteria used, an increase in the stability of the navigation estimate under degrading conditions (GNSS losses/increased noise) is observed, which confirms the research hypothesis about the usefulness of an LSTM-assisted process model in a Kalman structure.

5. The proposed architecture provides a basis for subsequent integration and validation in real conditions, including Hardware-in-the-loop (HIL) and field tests, as well as the development of more complex scenarios and sensor configurations.

IV. CONCLUSION.

This dissertation explores the possibility of constructing an integrated navigation system for unmanned aerial vehicles (UAVs), in which the process (transient) model within the Kalman filter framework is parameterized by a trainable LSTM-type recurrent neural network. The research included a literature review, the development of a simulation model in MATLAB/Simulink, the implementation of a hybrid LSTM–KF architecture, and the execution of experimental simulation scenarios with different noise profiles, disturbances, and modes of partial or intermittent availability of GNSS measurements. The results under the conditions considered show that the approach is technologically feasible and can provide measurable improvements over the baseline EKF under adverse operating conditions characterized by non-stationary noise and degraded measurements.

The summarized results of the experiments show that the developed hybrid architecture using a neural network achieves lower values of the metrics used to account for errors in position and velocity estimation under degrading or intermittent GNSS observations and in dead reckoning mode (dead reckoning), reaches a stable mode faster (Fig. 31) according to the selected criterion, and demonstrates stable behavior in the presence of increased IMU noise and non-stationary dynamics compared to the classical architecture. At the same time, in nominal mode (normal measurement conditions), the basic EKF achieves a lower aggregate error (Table 14) compared to LSTM-KF, which confirms the scenario-dependent nature of the observed advantages. The neural component supports process modeling by approximating nonlinear and mode-dependent dependencies that are difficult to describe with manually specified analytical models under real constraints. This manifests itself as improved estimation behavior in scenarios with GNSS degradation, increased noise, and highly nonlinear dynamics.

The hybrid navigation model, in which a recurrent neural network (RNN) of the LSTM (Long Short-Term Memory) type does not perform only a corrective function, but participates as a key element in the process model of the filter through training on sensor data and simulation trajectories. The proposed approach combines the structured mechanism for evaluating and integrating measurements in the Kalman framework with data-based dynamics modeling through a deep recurrent architecture. The results achieved show the potential for using trainable process models in real-time state estimation, subject to subsequent profiling, optimization, and validation on target hardware.

Based on the analysis and experiments conducted, the following guidelines for future research can be formulated: (i) inclusion of learnable components in modes with varying noise characteristics and incomplete knowledge of dynamics, with guaranteed control of generalization and evaluation/calibration of uncertainty; (ii) systematic comparison of different recurrent structures (LSTM and GRU), as well as alternative neural models in the state space, including Neural ODE / Neural SDE (neural differential / stochastic differential models), in terms of accuracy, stability, and computational complexity; (iii) extension to multimodal integration of sensor data (visual, LiDAR, and Ultra-Wideband – UWB) to limit errors during prolonged computation modes and degraded GNSS conditions.

Development opportunities include training and validation on real flight data, adding online adaptation mechanisms, and testing in a more realistic experimental environment (e.g., hardware-in-the-loop – HIL and field tests). Additionally, integration with reinforcement learning methods could be considered in the context of joint navigation and control optimization to achieve adaptation of the control policy to changing environmental conditions and dynamics.

The applicability of the proposed approach is potentially broad and includes autonomous navigation of UAVs, robotic systems, and autonomous platforms where reliable state estimation is critical. The developed architecture can be adapted for implementation with limited computational resources under profiling and optimization conditions, as the LSTM component allows for compact implementation and the

filtering framework supports efficient online processing. The results obtained support the conclusion that combining analytical modeling and neural architectures is a promising direction for creating more reliable and adaptive navigation systems for UAVs, especially in the case of nonlinearities, non-stationary noise regimes, and degraded observations.

GENERAL CONCLUSIONS FROM THE DISSERTATION

1. The analysis of navigation approaches shows that solutions based on a single sensor channel exhibit structural limitations in degraded or intermittent observations, which is why integrated architectures for combining inertial and external measurements are a methodologically sound choice for stability in a wide range of flight modes.

2. In inertial computation, the dominant factor for the correct operation of the model is the characteristics of the IMU – sensitivity to systematic deviations, temperature- instability, stochastic components that lead to the accumulation of navigation error in the absence of external corrections and require the use of correction channels and adaptive models.

3. The parameterization of the process model in the prediction step of the Kalman filter using a trainable predictor is conceptually sound and technically feasible – an LSTM-based model can perform the role of a process block while preserving the recursive predict–update structure and the correction step of the filter.

4. The simulation model shows that the correct integration of the neural network into the a priori forecast is critical for the accuracy of the results and for the comparative analysis of the standard analytical and proposed adaptive LSTM process model under identical input conditions.

5. The results of the comparative assessment between EKF and the hybrid LSTM–KF configuration (Kalman scheme in which the process model in the prediction step is implemented by an LSTM predictor) characterize scenario-dependent behavior: under nominal conditions, EKF achieves a lower value of the combined error metric (Combined RMS) compared to LSTM–KF, while under adverse conditions (GNSS interruption/dead reckoning mode, strong nonlinearity, and increased noise) LSTM–KF gives lower values of the metrics used to evaluate the error and reaches a faster established mode according to the selected establishment criterion compared to EKF (Fig. 31, Table 14, expression 4.2).

6. The analysis of the innovation sequences provides additional diagnostic information on the consistency between the forecast and the measurements, as the observed limitation of the systematic component of the residuals is compatible with adequate process modeling through the learnable component in the model under consideration.

GUIDELINES FOR FUTURE WORK

1. Extend experimental validation to a wider set of trajectories and modes, including more complex maneuvers, prolonged GNSS outages, and diverse noise profiles, to assess the generalization and robustness of the proposed architecture.

2. Include orientation states and additional measurements (e.g., magnetometer, barometer, visual/lidar observations), as well as extend to multi-sensor configurations to evaluate the effect of the learnable process model in a more complete navigation setting.

3. Investigate options for uncertainty adaptation and robustness (e.g., adaptive estimation of noise covariances and detection of atypical measurements), as well as analysis of statistical consistency through standard criteria when covariance estimates and verifiable confidence limits are included.

4. Transition to more realistic experimental conditions through hardware-in-the-loop (HIL) and field tests, including profiling and optimization for target hardware, to assess the applicability of the architecture under online processing and resource constraints.

V. CONTRIBUTIONS.

Scientific contributions

1. **A hybrid navigation system for integrated INS/GNSS navigation of UAVs** has been developed, in which the process (transitional) model in the predictive step of the Kalman filter (predict–update) is parameterized by a recurrent neural network of the LSTM type, and the correction step is implemented by the correction module of the filter. The system is conceptually justified in conditions of nonlinear dynamics and non-stationary noise and is implemented and evaluated through simulation scenarios.

Scientific and applied contributions

1. **The conditions for applicability of the hybrid INS/GNSS navigation configuration LSTM–KF** (Kalman filter in which the process model in the prediction step is implemented by a **recurrent neural network of the LSTM type**) relative to the basic EKF **are defined**: in the case of degraded or interrupted GNSS observations, increased IMU noise, and highly nonlinear dynamics, LSTM–KF achieves lower values of the error indicators used for position and linear velocity, while in nominal mode, EKF gives a lower value of the aggregate error indicator.
2. **A simulation model of an integrated navigation architecture has been developed in the MATLAB/Simulink development environment**, which supports parallel implementations of the process model – analytical f_{dyn} , and process predictor implemented with a recurrent neural network of the LSTM type – with coordinated input-output interfaces, a unified time base, and a controllable choice of process model.
3. **A reproducible workflow for building the hybrid navigation model (data–training–integration) has been implemented**, including automated extraction of simulation data, formation of training sequences, normalization and storage of scaling parameters, training of a **recurrent neural network of the LSTM type**, and implementation of the trained network in Simulink through a functional block for single-step prediction, compatible with the discrete filter circuit.
4. **A methodology for experimental evaluation of the hybrid INS/GNSS navigation architecture has been introduced**, using an aggregate metric (Combined RMS), component RMSE for the state, and an error detection criterion (e.g., 5% threshold), allowing for comparability of results under identical input conditions and different GNSS availability scenarios and noise profiles.
5. **Post-processing of the simulation results for statistical evaluation of the forecast in the hybrid navigation model has been implemented** by comparing the analytical process model f_{dyn} and the **LSTM-based process predictor (LSTM-type recurrent neural network)** against the reference state, with calculation and tabular presentation of RMSE by components (position and linear velocity).
6. **A diagnostic analysis of the integration of the LSTM-based process predictor (LSTM-type recurrent neural network) in the prediction step of the hybrid navigation scheme is presented**, including a distinction between the effects of the trainable process model and the effects of the simulation integration configuration, in order to correctly interpret the results and optimize the system parameters.

LIST OF PUBLICATIONS RELATED TO THE DISSERTATION:

- [1] Генов, В. (2024). Съвременни технологии за изкуствен интелект при управлението на безпилотни летателни апарати. Defense Technology Forum DTF24 (DefTech 2024), (стр. 668-674). Шумен
- [2] Генов, В. (2025). Съвременни технологии за интегрирани навигационни системи за безпилотни летателни апарати. Defense Technology Forum DTF25 (DefTech 2025), (стр. 691-700). Шумен
- [3] Генов, В. (2025). Приложения на невронни мрежи при работата на Калманов филтър. Defense Technology Forum DTF25 (DefTech 2025), (стр. 701-713). Шумен
- [4] Генов, В. (2025). Симулиране на интегрирана навигационна система за управление на безпилотен летателен апарат в Matlab. Defense Technology Forum DTF25 (DefTech 2025), (стр. 755-766). Шумен