Додаток 1

ПОРІВНЯЛЬНА ТАБЛИЦЯ

із наведенням фрагментів дисертації Коваленко Ю. Б.
та відповідних фрагментів опублікованих текстів інших авторів
без зазначення авторства

Збіги текстів виділені жовтим кольором, перефразування та синоніми – бірюзовим, перестановки слів місцями – зеленим; твердження особи про те, що це нібито вона щось пропонує, оцінює, розробила чи робить висновки – фіолетовим.

Червоним шрифтом поданий коментар щодо фрагментів дисертації Коваленко Ю. Б.

№	**Фрагменти тексту дисертації, у якій виявлено факти порушення академічної доброчесності**	**Фрагменти опублікованих текстів інших авторів (без зазначення в дисертації** Коваленко Ю. Б **посилань на джерело)**
	Вид виявленого порушення: **плагіат**	
1	**Коваленко Ю. Б.** **Інформаційна підтримка процесу проектування та експлуатації комплексу бортового обладнання інтегрованої модульної авіоніки.** – Дис. … доктора технічних наук. – Київ, 2025. (https://duikt.edu.ua/uploads/p_86_93515477.pdf)	**Сильянов Н. В.** **Диагностическое обеспечение многофункциональных бортовых вычислительных систем на основе графовых и алгебраических моделей.** – Дисс. … канд. техн. наук. – Нижний Новгород, 2019.
	С. 36.	**С. 19.**
	Розглянемо процес автоматичного контролю який складається з двох основних етапів. Перший етап полягає в сприйнятті інформації про стан об'єкта і зовнішні умови (первинне перетворення) і в перетворенні її на вигляд, зручний для подальшої обробки (проміжне перетворення). Другий етап полягає у виявленні одержаної інформації ознак контрольованої події, тобто тих специфічних особливостей, які відрізняють цю подію від всіх інших, і в формуванні сигналу про настання цієї події при наявності повної сукупності, що характеризують його ознаки. **Коваленко в переписаному чужому тексті видалила покликання.** **Помилка перекладу: російський вислів «**совокупности характеризующих его признаков**» Коваленко переклала як «**сукупності, що характеризують його ознаки**».** **Смішний плагіат.**	Согласно [14], процесс автоматического контроля состоит из двух основных этапов. Первый этап заключается в восприятии информации о состоянии объекта и внешних условиях (первичное преобразование) и в преобразовании ее к вида, удобному для последующей обработки (промежуточное преобразование). Второй этап состоит в обнаружении в поступающей информации признаков контролируемого события, т.е. тех специфических особенностей, которые отличают данное событие от всех других, и в формировании сигнала о наступлении этого события при наличии полной совокупности характеризующих его признаков.
	С. 36–37.	**С. 19.**
	У [10] розглядається метод аналізу функціонально пов'язаних безперервних систем з метою визначення мінімального числа точок контролю. За блочно-функціональною схемою будується логічна модель, строгий математичний опис у термінах алгебри логіки. Наводиться алгоритм визначення мінімального числа точок контролю. Наводяться приклади складання таблиці несправностей і визначення мінімальних сукупностей виходів. Ступінь зв'язаності елементів в системі є одним із чинників, що визначають методологію випробувань [16]. Одним із недоліків передчасного переходу до випробувань системи є те, що в цьому випадку буде дуже багато незрозумілих відмов елементів, а це ду-	В [15] рассматривается метод анализа функционально связанных непрерывных систем с целью определения минимального числа точек контроля. По блочно-функциональной схеме строится логическая модель (строгое математическое описание в терминах алгебры логики). Приводится алгоритм определения минимального числа точек контроля. Приводятся примеры составления таблицы неисправностей и определения минимальных совокупностей выходов. Степень связанности элементов в системе является одним из факторов, определяющих методологию испытаний [16]. Одним из недостатков преждевременного перехода к испытаниям (функ-

	же ускладнює дослідження статистики відмов. Інший недолік – відмови одних елементів зроблять жорсткішим режим роботи інших. **Коваленко замінила в переписаному чужому тексті покликання [15] на фальшиве [10].** **Коваленко залишила в переписаному чужому тексті покликання [16], але під цим номером – інше джерело.** **Фальсифікація джерел.** **Плагіат.**	циональным) системы является то, что в этом случае будет очень много необъяснимых отказов элементов, а это очень затрудняет исследование статистики отказов. Другой недостаток - отказы одних элементов сделают более жестким режим работы других.
	С. 37.	**С. 19–20.**
	У [11] наводиться методика аналізу одного класу систем для визначення мінімального числа контрольованих виходів, що забезпечують перевірку працездатності або діагностику ушкоджень. Модель також має передбачає блочно-функціональну структуру. На розбиття системи на блоки накладається низка обмежень. Зокрема, йдеться про системи без зворотного зв'язку і без резервування, оскільки в іншому випадку в системі будуть невиразні несправності. У [12] розглядається задача виділення невідомого автомата з заданого класу автоматів шляхом подачі послідовностей вхідних символів і спостереження реакцій. Завдання розглядається для випадку, коли автомати з істотними несправностями відрізняються. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	В [17] приводится методика анализа одного класса систем для определения минимального числа контролируемых выходов, обеспечивающих проверку работоспособности или диагностику повреждений. Модель также подразумевает блочно-функциональную структуру. На разбиение системы на блоки накладывается ряд ограничений. В частности подразумеваются системы без обратных связей и без резервирования, поскольку в противном случае в системе будут неразличимые неисправности. В [18] рассматривается задача выделения неизвестного автомата из заданного класса автоматов путем подачи последовательностей входных символов и наблюдения реакций. Задача рассматривается для случая, когда автоматы с существенными неисправностями отличимы.
	С. 37–38.	**С. 20.**
	Ручні засоби контролю мають недоліки: низьку швидкість, недостатню достовірність і в певних випадках високу вартість перевірок [13]. Автоматичні системи контролю дозволяють в значній мірі усунути ці недоліки. Складність процесу контролю визначається складністю об'єкта. Для опису функціонування об'єкта створюється математична функціонально-статистична модель. Модель задається системою рівнянь, що описують залежність параметрів об'єкта від зовнішніх і внутрішніх впливів при функціонуванні. Загальна ймовірність працездатності об'єкта після його r-й перевірки і настройки в [14] визначається за формулою 1: $$P^{(r)} = \prod_{i=1}^{n_1} D_i^{(r)} \prod_{i=1}^{n_1} P_{\sigma i/i-1} \prod_{i=n_1+1}^{n} P_{\sigma i/i-1}, \quad (1)$$ де $D_i^{(r)}$ – достовірність r-й перевірки і налаштування i-го параметра об'єкта, $P_{\sigma i/1-1}$ – умовна ймовірність безвідмовної роботи по i-му контрольованим параметром, n – загальне число параметрів, що характеризують працездатність об'єкта, n1 – число параметрів, що контролюються.	Ручные средства контроля имеют очевидные недостатки: низкую скорость, недостаточную достоверность и в ряде случаев высокую стоимость проверок [19]. Автоматические системы контроля позволяют в значительной мере устранить эти недостатки. Сложность процесса контроля определяется сложностью объекта. Для описания функционирования объекта создается математическая функционально-статистическая модель. Модель задается системой уравнений, описывающих зависимость параметров объекта от внешних и внутренних воздействий при функционировании. Общая вероятность работоспособности объекта после его r-й проверки и настройки в [19] определяется по формуле 1.3: $$P^{\{z\}} = \text{Цо}\backslash\text{"ЦР}_{17}, \; , \; ! \prod_{i=1}^{\text{Щ}} 1\text{-}_{t=1}^{Til} \; {}_{i=ttt4T}^{\text{п}} \quad (!.3)$$ где $D^{(r)}$ — достоверность r-й проверки и настройки r-го параметра объекта, $Pcri/i_{-1}$ - условная вероятность безотказной работы по i-му контролируемому параметру, *п* общее число параметров, характеризующих работоспособность объекта, *п*\ число контролируемых параметров.

	Коваленко замінила в переписаному чужому тексті покликання на фальшиві. **Фальсифікація джерел.** **Плагіат.**	
	С. 38.	**С. 20–21.**
	У загальному випадку достовірність r-й перевірки і налаштування і-го контрольованого параметра об'єкта залежить від ймовірності невиявлених і тих неправдивих відмов, які при налаштуванні переходять до невиявлених відмов. Достовірність перевірки і налаштування визначається за формулою 2: $$D_i^{(r)} = 1 - P_{hi}^{(r)} - P_{f \to hi}^{(r)}, \quad (2)$$ де $P_{hi}^{(r)}$ – ймовірність існування невиявленої відмови, $P^{(r)}_{f \to hi}$ – імовірність невиявлених і тих неправдивих відмов, які при налаштуванні переходять до невиявлених відмов.	В общем случае достоверность r-й проверки и настройки r-го контролируемого параметра объекта зависит от вероятности необнаруженных и тех ложных отказов, которые при настройке переходят в необнаруженные отказы. Достоверность проверки и настройки определяется по формуле 1.4: $$D_i^{(r)} = 1 - P_{hi}^{(r)} - P_{f \to hi}^{(r)} \quad (1.4)$$ где $Pf_d^{(r)}$ — вероятность существования необнаруженного отказа, (r) — вероятность необнаруженных и тех ложных отказов, которые при настройке переходят в необнаруженные отказы.
	С. 38.	**С. 21.**
	У [15] також наголошується, що одним із шляхів підтримки необхідного рівня надійності складних обчислювальних пристроїв у процесі їхньої експлуатації є контроль працездатності та пошук несправностей. Важливе значення має також контроль в процесі виробництва складних пристроїв. Незалежно від того, якими способами і засобами мусить здійснюватися контроль, завжди важливо мати якомога менше витрат часу і апаратури контролю. У зв'язку з цим велике значення має впровадження методів отримання мінімальних сукупностей впливів і контрольних параметрів, а також методів оптимізації програм перевірок, таблиця 1.1. У [16] розглядається метод побудови мінімальних і мінімізованих тестів. Наводяться оцінки існування тестів меншої довжини щодо мінімізованих тестів, а також імовірні оцінки близькості отриманих тестів до мінімальних. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	В [20] также отмечается, что одним из путей поддержания требуемого уровня надежности сложных вычислительных устройств в процессе их эксплуатации является контроль работоспособности и поиск неисправностей. Важное значение имеет также контроль в процессе производства сложных устройств. Независимо от того, какими способами и средствами должен производиться контроль, всегда важно иметь как можно меньше затрат времени и аппаратуры контроля. В связи с этим имеет большое значение внедрение методов получения минимальных совокупностей воздействий и контрольных параметров, а также методов оптимизации программ проверок. В [21] рассматривается метод построения минимальных и минимизированных тестов. Приводятся оценки существования тестов меньшей длины относительно минимизированных тестов, а также вероятностные оценки близости полученных тестов к минимальным.
	С. 39.	**С. 21.**
	Розвиток апаратури призводить до ускладнення вирішуваних завдань, що спричиняє ускладнення схем. Для правильного визначення працездатності важлива кількісна оцінка з високою точністю, а для локалізації несправності – більша кількість діагностичної інформації. З'явилася блочно-модульна побудова, яка забезпечує більш швидке усунення несправностей. Успішне вирішення завдань забезпечувалося для невеликих пристроїв із малим числом функціональних блоків. **Коваленко в переписаному чужому тексті видалила покликання.** **Плагіат.**	Первое упоминание принципов построения автоматических систем контроля и их классификация найдены в [22]. Развитие аппаратуры приводит к усложнению решаемых задач, что приводит к усложнению схем. Для правильного определения работоспособности требуется количественная оценка с высокой точностью, а для локализации неисправности большее количество диагностической информации. Появилось блочно-модульное построение, которое обеспечивает более быстрое устранение неисправностей. Успешное решение задач обеспечивалось для небольших устройств с малым числом функциональных блоков.
	С. 39–40.	**С. 21–22.**

	У [17] пропонується знайти функціональну залежність між складовими вектора і параметрами елементів пристрою, яка є одним з m співвідношень, що характеризують складові вектора, отримані при випробуваннях. Результат будь-якого вимірювання вихідного параметра містить в собі зміну його значення як через випадкові зміни параметрів елементів у межах поля допусків, так і внаслідок пошкоджень елементів. Для вирішення завдання пошуку елемента, що відмовив, необхідно досліджувати всі можливі відмови елементів з тим, щоб визначити, яке ушкодження найтісніше пов'язані з отриманою величиною вихідного параметра. Згідно [18] автоматичний контроль в поєднанні з можливістю усунення несправностей є одним із способів підвищення надійності апаратури. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	В [22] предлагается найти функциональную зависимость между составляющими вектора Y и параметрами элементов устройства, которая является одним из m соотношений, характеризующих составляющие вектора Y, полученные при испытаниях. Результат любого измерения выходного параметра содержит в себе изменение его значения как из-за случайных изменений параметров элементов в пределах поля допусков, так и вследствие повреждений элементов. Для решения задачи поиска отказавшего элемента необходимо исследовать все возможные отказы элементов с тем, чтобы определить, какое повреждение наиболее тесно связано с полученной величиной выходного параметра. Согласно [22] автоматический контроль в сочетании с возможностью устранения неисправностей является одним из способов повышения надежности аппаратуры.
	С. 40.	**С. 22.**
	У [19] зазначається, що класичні методи діагностики розглядають комбінаційні і послідовні схеми з метою визначення мінімального набору тестів. Метод чорного ящика дає лише функціональну перевірку справності, без можливості пошуку місця несправності. Інший підхід заснований на створенні діагностичних словників. Пропонується розглядати систему окремо на поведінковому і структурному рівнях. На структурному рівні зручно застосувати відображення у вигляді графів. Будь-яка дискретна послідовна система ізоморфна орієнтованому графу. Вершини відображають функціональні елементи (комбінаційні або послідовні), дуги з'єднання між ними. За графу можна побудувати матрицю суміжності, а з неї, відповідно, отримати матрицю досяжності. У складній системі проводиться сегментація, цикли і ланцюги графа замінюються узагальненими вершинами. Контрольні точки мають обиратися таким чином, щоб тестова процедура ізолювала підсистему від її сусідів з дотриманням умов спостереження та управління. **Коваленко замінила в переписаному чужому тексті покликання на фальшиве.** **Фальсифікація джерел.** **Плагіат.**	В [23] указывается, что классические методы диагностики рассматривают комбинационные и последовательностные схемы с целью определения минимального набора тестов. Метод черного ящика дает только функциональную проверку исправности, без возможности поиска места неисправности. Другой подход основан на создание диагностических словарей. Предлагается рассматривать систему отдельно на поведенческом и структурном уровнях. На структурном уровне удобно применить отображение в виде графов. Любая дискретная последовательная система изоморфна ориентированному графу. Вершины отображают функциональные элементы (комбинационные или последовательностные), дуги — соединения между ними. По графу можно построить матрицу смежности, а из нее в свою очередь получить матрицу достижимости. В сложной системе проводится сегментация, циклы и цепи графа заменяются обобщенными вершинами. Контрольные точки должны выбираться так, чтобы тестовая процедура изолировала подсистему от ее соседей с соблюдением условий управляемости и наблюдаемости.
	С. 40.	**С. 22.**
	У [20] наводиться метод побудови програм перевірки дискретних пристроїв з пам'яттю, робота яких описується моделлю кінцевого автомата у вигляді таблиць переходів і виходів. Відповіді автоматів на тестові впливи є значеннями ознак, за якими проводиться розбиття множини автоматів на класи. Наводиться алгоритм побудови діагностичної програми. **Коваленко замінила в переписаному чужому тексті покликання на фальшиве.** **Фальсифікація джерел.** **Плагіат.**	В [24] приводится метод построения программ проверки дискретных устройств с памятью, работа которых описывается моделью конечного автомата в виде таблиц переходов и выходов. Ответы автоматов на тестовые воздействия являются значениями признаков, по которым производится разбиение множества автоматов на классы. Приводится алгоритм построения диагностической программы.

	С. 40–41.	**С. 22–23.**
	У [21] розглядається задача контролю працездатності і діагностики несправностей в дискретних блокових об'єктах. При цьому об'єкти представляються логічними моделями, що відображають причинно-наслідкові зв'язки, що існують в об'єкті при наявності несправностей. Досліджуються питання побудови універсальних діагностичних пристроїв для локалізації несправностей з точністю до змінних блоків. Створені раніше методи належать переважно до комбінаційних схем. Тестовий контроль дискретних об'єктів вимагає виконання великого числа обчислювальних операцій. Перспективним шляхом слід вважати апаратні способи діагностики. Наводяться моделі для синхронних і асинхронних дискретних об'єктів. Розглядаються питання реалізації універсального пристрою діагностики для застосування його в якості зовнішнього пристрою, як вбудованого пристрою для об'єктів з резервуванням, як вбудованого пристрою для об'єктів, що мають вбудований контроль в блоках. **Коваленко замінила в переписаному чужому тексті покликання на фальшиве.** **Фальсифікація джерел.** **Плагіат.**	В [25] рассматривается задача контроля работоспособности и диагностики неисправностей в дискретных блочных объектах. При этом объекты представляются логическими моделями, отражающими причинно-следственные связи, существующие в объекте при наличии неисправностей. Исследуются вопросы построения универсальных диагностических устройств для локализации неисправностей с точностью до сменных блоков. Созданные ранее методы касаются в основном комбинационных схем. Тестовый контроль дискретных объектов требует выполнения большого числа вычислительных операций. Перспективным путем следует считать аппаратные способы диагностики. Приводятся модели для синхронных и асинхронных дискретных объектов. Рассматриваются вопросы реализации универсального устройства диагностики для применения его в качестве внешнего устройства, в качестве встроенного устройства для объектов с резервированием, в качестве встроенного устройства для объектов, имеющих встроенный контроль в блоках.
	С. 41.	**С. 23.**
	У [22] розглядається методика побудови оптимальних універсальних систем контролю на основі теорії статистичних рішень. Структурно система контролю складається з формувача (перетворює простір станів на простір сигналів), класифікатора (що вказує прийнятим за отриманими сигналами приналежність стану до відповідного класу) і інформатора (видає повідомлення на основі прийнятих сигналів і вказівок класифікатора). **Коваленко замінила в переписаному чужому тексті покликання на фальшиве і видалила згадку про рисунок.** **Фальсифікація джерел.** **Плагіат.**	В [26] рассматривается методика построения оптимальных универсальных систем контроля на основе теории статистических решений. Приведенная в [26] классификация систем контроля показана на рисунке 1.1. Структурно система контроля состоит из формирователя (преобразующего пространство состояний в пространство сигналов), классификатора (указывающего по принятым сигналам принадлежность состояния к соответствующему классу) и информатора (выдающего сообщение на основе принятых сигналов и указаний классификатора).
	С. 41.	**С. 23, 25.**
	У [23] розглядається діагностика несправностей комбінаційних схем на основі еквівалентної нормальної форми. Згідно [24] основними характеристиками якості розпізнавання є помилки розпізнавання і середні втрати. Там же наводяться статистичні критерії виявлення: мінімального ризику Байеса, ідеального спостерігача Зигерта-Котельникова, максимальної правдоподібності Фішера, мінімаксний, спостерігача Неймана-Пірсона, мінімальної тривалості експерименту Вальда; а також статистичні критерії розпізнавання: мінімального ризику і сумарного перевищення. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	В [27] рассматривается диагностика неисправностей комбинационных схем на основе эквивалентной нормальной формы (ЭНФ). Согласно [28] основными характеристиками качества распознавания являются ошибки распознавания и средние потери. Там же приводятся статистические критерии обнаружения: минимального риска Байеса, идеального наблюдателя Зигерта-Котелышкова, максимального правдоподобия Фишера, минимаксный, наблюдателя Неймана-Пирсона, минимальной длительности эксперимента Вальда; а также статистические критерии распознавания: минимального риска и суммарного превышения.

	С. 41.	**С. 25.**
	У [25] дається класифікація методів технічного діагностування обчислювальних пристроїв. Розрізняють універсальні та спеціальні методи. Універсальні методи побудови перевіряючих і діагностичних тестів розбиваються на три основних види: – метод теорії експериментів; – метод таблиць функцій несправностей; – метод істотних шляхів. **Коваленко замінила в переписаному чужому тексті покликання на фальшиве.** **Фальсифікація джерел.** **Плагіат.**	В [29] дается классификация методов технического диагностирования вычислительных устройств. Различают универсальные и специальные методы. Универсальные методы построения проверяющих и диагностических тестов разбиваются на три основных вида: метод теории экспериментов; метод таблиц функций неисправностей; метод существенных путей.
	С. 41–42.	**С. 25.**
	У [26] розглядається оптимальний пошук несправностей. Система, що розглядається як об'єкт контролю, це сукупність складових її елементів, з'єднаних між собою функціональними зв'язками. Кожен з елементів системи може знаходитися в двох станах: справності або відмови. Контроль системи полягає в послідовному застосуванні спеціальних тестів, кожен з яких перевіряє справність цілком певної підмножини елементів. Перевірка проводиться або для підтвердження справності, або для віднайдення несправності. Тест, який перевіряє всю систему , буває неможливий або недоцільний, з цього вигідніше використовувати послідовність декількох простих тестів. У загальному випадку підмножини елементів, що перевіряються різними тестами, можуть перетинатися [27]. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	В [30] рассматривается оптимальный поиск неисправностей. Система, рассматриваемая как объект контроля, представляет совокупность составляющих ее элементов (множество), соединенных между собой функциональными связями. Каждый из элементов системы может находиться в двух состояниях: исправности или отказа. Контроль системы заключается в последовательном применении специальных тестов, каждый из которых проверяет исправность вполне определенного подмножества элементов. Проверка проводится либо для подтверждения исправности (обнаружения любой неисправности), либо для отыскания неисправности (локализации всех отказавших элементов). Тест, проверяющий всю систему (глобальный) бывает невозможен или нецелесообразен, поэтому бывает выгоднее использовать последовательность нескольких простых тестов. В общем случае подмножества элементов, проверяемые различными тестами могут пересекаться.
	С. 42.	**С. 25–26.**
	Наводяться алгоритми віднайдення єдиного несправного елемента, а також невідомого числа несправних елементів при довільних пересічних тестах і при поелементній перевірці. У загальному вигляді тестова стратегія ґрунтується на впорядкування тестів по тимчасових витратах і по можливостям відмов елементів. Аналогічний підхід застосовується і для виявлення несправності. У [28] наводиться метод побудови повних перевіряючих і діагностичних тестів комбінаційних пристроїв. Пропонується спосіб редукції повного діагностичного тесту до повного перевіряючого тесту. При побудові перевіряючого тесту очікувана реакція випробуваного пристрою вважається такою, яка збігається з реакцією справного пристрою. В іншому випадку перевірка закінчується і пристрій вважається несправним. Для діагностичного експерименту реакція випробуваного пристрою заздалегідь невідома. **Коваленко замінила в переписаному чужому тексті покликання на фальшиве.** **Фальсифікація джерел.** **Плагіат.**	Приводятся алгоритмы отыскания единственного неисправного элемента, а также неизвестного числа неисправных элементов при произвольных пересекающихся тестах и при поэлементной проверке. В общем виде тестовая стратегия основывается на упорядочивании тестов по временным затратам и по вероятностям отказов элементов. Аналогичный подход применяется и для обнаружения неисправности. В [311 приводится метод построения полных проверяющих и диагностических тестов комбинационных устройств. Предлагается способ редукции полного диагностического теста до полного проверяющего теста. При построении проверяющего теста ожидаемая реакция испытуемого устройства считается совпадающей с реакцией исправного устройства, В противном случае проверка заканчивается и устройство считается неисправным. Для диагностического эксперимента реакция испытуемого устройства заранее неизвестна. **Помилка оцифровки дисертації: написано покликання «[311» (треба «[31]»).**

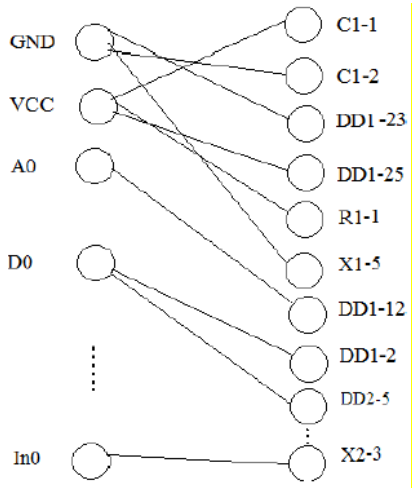
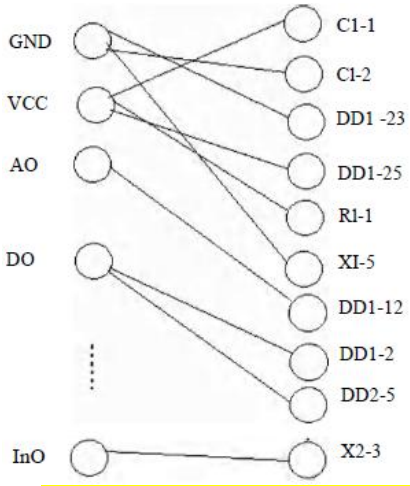
	С. 42.	**С. 26.**
	У [29] наводиться зв'язок достовірності контролю з імовірністю безвідмовної роботи. Достовірність як правильне відображення реального стану об'єкта контролю вважається основним критерієм оцінки якості контролю. Поряд з економічними і експлуатаційними характеристиками достовірність визначає ефективність контролю. **Коваленко замінила в переписаному чужому тексті покликання на фальшиве.** **Фальсифікація джерел.** **Плагіат.**	В [32] приводится связь достоверности контроля с вероятностью безотказной работы. Достоверность как правильное отображение реального состояния объекта контроля считается основным критерием оценки качества контроля. Наряду[7] с экономическими и эксплуатационными характеристиками достоверность определяет эффективность контроля.
	С. 42–43.	**С. 27.**
	У [30] першочергова увага приділяється логічним аспектам технічної діагностики. По-перше, для більшості завдань побудови алгоритмів діагнозу, так само як і методів їхнього вирішення, притаманний комбінаторний характер. По-друге, допустимі способи діагнозу технічного стану безперервних об'єктів завдяки простоті, яка властива для цих способів реалізації, набули найширшого розповсюдження. По-третє, дискретні об'єкти – це великий клас пристроїв і систем, в основу якого має бути покладений логічний підхід. Більшу частину технічних засобів діагнозу представляють пристрої дискретної дії. **Коваленко замінила в переписаному чужому тексті покликання на фальшиве.** **Фальсифікація джерел.** **Плагіат.**	В [34] преобладающее внимание уделяется логическим аспектам технической диагностики. Во-первых, большинство задач построения алгоритмов диагноза, равно как и методов их решения, носит комбинаторный характер. Во-вторых, допусковые способы диагноза технического состояния непрерывных объектов благодаря присущей этим способам простоте реализации получили самое широкое распространение. В-третьих, дискретные объекты представляют большой класс устройств и систем, в основу которого должен быть положен логический подход. Большую часть технических средств диагноза представляют устройства дискретного действия.
	С. 43.	**С. 28–29.**
	В якості універсальної математичної моделі розглядаються таблиця функцій несправностей, де за рядками відкладаються елементарні перевірки, за стовпцями – технічні стани об'єкта або, що те ж саме, – функції, що реалізуються об'єктом в різних станах. Логічна схема зіставляється з орієнтованим графом, вершинам якого відповідають логічні елементи, вхідні і вихідні полюси, а дугам – з'єднання, причому впорядковані, тому що функція, що реалізується елементом, в загальному випадку несиметрична щодо перестановок вхідних змінних. Логічна мережа може бути представлена в дужковій формі, а також в еквівалентній нормальній формі. Під фізичною несправністю можна розуміти наслідки певної події, яка перетворює справний пристрій на несправний. Можливі наслідки, які є несправностями при структурному описі і не є такими при його функціональному описі. Серед найбільш відомих несправностей з'єднань елементів виділяють наступні: – обрив з'єднання; – замикання з'єднання з шиною живлення; – переплутування зв'язків; – поява зайвих зв'язків; – замикання декількох зв'язків між собою; – сукупність несправностей.	В качестве универсальной математической модели рассматривается таблица функций неисправностей, где по строкам откладываются элементарные проверки, по столбцам — технические состояния объекта или, что то же самое функции, реализуемые объектом в различных состояниях. Логической схеме сопоставляется ориентированный граф, вершинам которого соответствуют логические элементы, входные и выходные полюсы, а дугам соединения, причем упорядоченные, т.к. функция, реализуемая элементом, в общем случае несимметрична относительно перестановок входных переменных. Логическая сеть может быть представлена в скобочной форме, а также в эквивалентной нормальной форме. Под физической неисправностью можно понимать последствия некоторого события, которое преобразует исправное устройство в неисправное. Возможны последствия, которые являются неисправностями при структурном описании и не являются таковыми при его функциональном описании. Среди наиболее часто встречающихся неисправностей соединений элементов выделяют следующие: обрыв соединения; замыкание соединения с шиной питания; перепутывайие связей; появление лишних связей; замыкание нескольких связей между собой; совокупность неисправностей.
	С. 43–44.	**С. 29.**
	Дані несправності можуть бути еквівалентні по-	Данные неисправности могут быть эквивалент-

	дачі на вхід постійного сигналу 0 або 1, зміні функції елемента, зміні класу пристрою. Таблиця функцій несправностей є явною моделлю об'єкта діагнозу. При роботі з ТФН багато завдань діагнозу вирішуються просто в обмін на великі обсяги обчислень і пам'яті.	ны подаче на вход постоянного сигнала 0 или 1, изменению функции элемента, изменение класса устройства. Таблица функций неисправностей (ТФН) является явной моделью объекта диагноза. При работе с ТФН многие задачи диагноза решаются просто в обмен на большие объемы вычислений и памяти.
	С. 44.	**С. 29.**
	Іншим способом пошуку несправностей є використання діагностичного словника. Повний діагностичний словник має містити значення виходів комбінаційного пристрою при всіх можливих несправностях і на всій множині вхідних наборів пристрою. За збігом фактичних вихідних значень випробуваного пристрою зі значеннями зі словника визначається технічний стан. У [31] описана автоматизована система забезпечення надійності функціонування технологічних установок безперервної дії. Автоматизації зазнали операції діагностування, визначення надійності та прийняття рішення про корегувальні впливи. Застосування подібних систем стрімко знижує час відновлення, виключає небезпечні наслідки і причини відмов апаратури або зменшує їхню ймовірність. **Коваленко замінила в переписаному чужому тексті покликання на фальшиве.** **Фальсифікація джерел.** **Плагіат.**	Другим способом поиска неисправностей является использование диагностического словаря. Полный диагностический словарь должен содержать значения выходов комбинационного устройства при всех возможных неисправностях на всем множестве входных наборов устройства. По совпадению фактических выходных значений испытуемого устройства со значениями из словаря определяется техническое состояние. В [35] описана автоматизированная система обеспечения надежности функционирования технологических установок непрерывного действия. Автоматизации подвергались операции диагностирования, определения надежности и принятия решения о корректирующих воздействиях. Применение подобных систем резко снижает время восстановления, исключает опасные последствия и причины отказов аппаратуры или уменьшает их вероятность.
	С. 44.	**С. 29–30.**
	Завдання побудови діагностичної моделі зводиться до оптимізації процесу розпізнавання несправностей в об'єкті і визначенню множин: діагностичних і режимних параметрів, класів несправностей, описів несправностей в просторі ознак, алгоритмів формування ознак і розпізнавання класів несправностей. У [32] наводиться методика динамічного контролю. Динамічний контроль – певні дії, спрямовані на визначення технічного стану системи за результатами вимірювання істотно змінних у часі сигналів. Динамічний контроль знаходить основне застосування для визначення працездатності інерційних об'єктів. Згідно [33] метою технічної діагностики є підвищення надійності ресурсу технічних систем. Основне завдання – розпізнавання стану технічної системи в умовах обмеженої інформації. Теоретичний фундамент – теорія розпізнавання образів, завдання класифікації. Алгоритми розпізнавання ґрунтуються на діагностичних моделях, що встановлюють зв'язок між станами технічної системи і їхнім відображеннями у просторі діагностичних сигналів. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	Задача построения диагностической модели сводится к оптимизации процесса распознавания неисправностей в объекте и определению множеств: диагностических и режимных параметров, классов неисправностей, описаний неисправностей в пространстве признаков, алгоритмов формирования признаков и распознавания классов неисправностей. В [36] приводится методика динамического контроля. Динамический контроль ряд действий, направленных на определение технического состояния системы по результатам измерения существенно переменных во времени сигналов. Динамический контроль находит основное применение для определения работоспособности инерционных объектов. Согласно [37] целью технической диагностики является повышение надежности ресурса технических систем. Основная задача — распознавание состояния технической системы в условиях ограниченной информации. Теоретический фундамент теория распознавания образов, задачи классификации. Алгоритмы распознавания основываются на диагностических моделях, устанавливающих связь между состояниями технической системы и их отображениями в пространстве диагностических сигналов.
	С. 45.	**С. 30.**
	У [34, 35] алгоритми виявлення несправностей є	В [38, 39] алгоритмы обнаружения неисправно-

	частиною методики реалізації відмовостійкості. Вони мають забезпечувати виявлення всіх необхідних класів несправностей, а також достатню ймовірність своєчасного виявлення. Перший крок забезпечення відмовостійкості – чітка ідентифікація несправностей, що становлять інтерес. Перший етап виявлення несправностей – початковий контроль, що здійснюється перед нормальною експлуатацією і служить для виявлення несправних елементів апаратури, дефекти яких виникли під час виробництва або збірки. Повний контроль на логічному рівні економічно неможливий. Зростає роль імовірнісних підходів і комплексного логіко-функціонального контролю. Вартість початкового контролю становить значну частину загальної вартості цифрових схем. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	стей являются частью методики реализации отказоустойчивости. Они должны обеспечивать выявление всех требуемых классов неисправностей, а также достаточную вероятность своевременного выявления. Первый шаг обеспечения отказоустойчивости четкая идентификация представляющих интерес неисправностей. Первый этап обнаружения неисправностей начальный контроль, осуществляемый перед нормальной эксплуатацией и служащий для выявления неисправных элементов аппаратуры, дефекты которых возникли в ходе производства или сборки. Полный контроль на логическом уровне экономически неосуществим. Возрастает роль вероятностных подходов и комплексного логико-функционального контроля. Стоимость начального контроля представляет значительную часть общей стоимости цифровых схем.
	С. 45.	**С. 33.**
	У [36] наводиться аналіз впливу допусків пасивних елементів ланцюгів на ненадійні характеристики системи. У роботі застосовуються методи теорії графів і матриць. У [37] в якості прикладу типових задач дослідження операцій наводиться організація вибіркового контролю продукції для забезпечення заданого рівня якості при мінімальних витратах на контроль. Щоб порівнювати між собою по ефективності різні рішення, потрібно мати певний кількісний критерій, показник ефективності, цільову функцію. Для вибіркового контролю такий показник – середні очікувані витрати R на контроль за одиницю часу за умови, що система контролю забезпечує заданий рівень якості, наприклад, середній відсоток браку не вище заданого. При цьому R потрібно мінімізувати. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	В [42] приводится анализ влияния допусков пассивных элементов цепей на надежностные характеристики системы. В работе применяются методы теории графов и матриц. В [43] в качестве примера типичных задач исследования операций (системного анализа) приводится организация выборочного контроля продукции для обеспечения заданного уровня качества при минимальных расходах на контроль. Чтобы сравнивать между собой по эффективности разные решения, нужно иметь какой-то количественный критерий, показатель эффективности, целевую функцию. Для выборочного контроля такой показатель — средние ожидаемые расходы R на контроль за единицу времени, при условии, что система контроля обеспечивает заданный уровень качества, например, средний процент брака не выше заданного. При этом R требуется минимизировать.
	С. 45–46.	**С. 33–34.**
	У [38] акцентується на тому, що навіть неповний літературний огляд показує, що запропонована досить велика кількість методів оцінки працездатності та ступеня працездатності об'єктів і систем управління. Однак практичне втілення цих методів стикається з такими труднощами, як значне число непрямих вимірювань з обробкою для визначення поточних значень параметрів системи. При великому числі контрольованих параметрів обсяг обчислень надзвичайно великий, а процес обробки громіздкий і непридатний для реалізації за прийнятний час. Зменшення числа контрольованих параметрів і використання наближених залежностей дає лише наближене уявлення про працездатність системи. У [39] розглядається задача перевірки правильності функціонування безперервних динамічних об'єктів в робочому режимі. Основну увагу приділено групі методів, в яких діагностування виконується шляхом перевірки алгебраїчних співвідношень між вихідними сигналами об'єкта і пристрою діагносту-	В [44] указывается на то, что даже не полный литературный обзор показывает, что предложено достаточно большое число методов оценки работоспособности и степени работоспособности объектов и систем управления. Однако практическое воплощение этих методов сталкивается с такими трудностями, как значительное число косвенных измерений с обработкой для определения текущих значений параметров системы. При большом числе контролируемых параметров объем вычислений чрезвычайно велик, а процесс обработки громоздкий и непригодный для реализации за приемлемое время. Уменьшение числа контролируемых параметров и использование приближенных зависимостей дает лишь приближенное представление о работоспособности системы. В [45] рассматривается задача проверки правильности функционирования непрерывных динамических объектов в рабочем режиме. Основное внимание уделено группе методов, в которых диагностирование выполняется путем проверки алгеб-

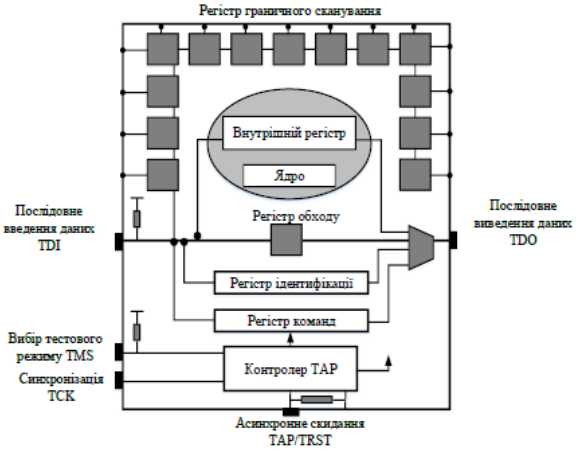
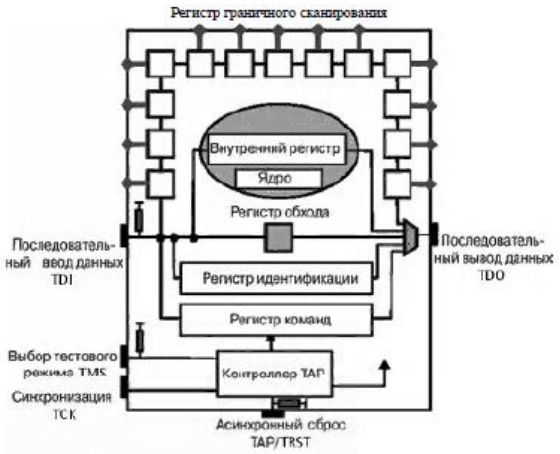
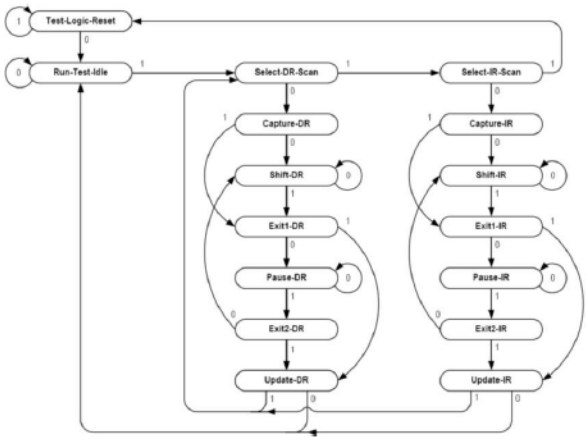
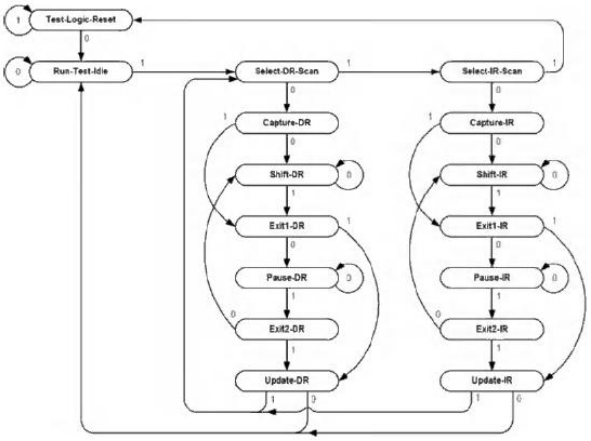
	вання. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	раических соотношений между' выходными сигналами объекта и устройства диагностирования.
	С. 46.	**С. 38.**
	У [40, 41] методи виявлення помилок в цифрових схемах об'єднуються в наступні групи: – модифікація схем дублювання і порівняння, – використання спеціальних кодів, – алгебраїчний метод, – контроль за перебігом процесу. **Коваленко замінила в переписаному чужому тексті покликання на фальшиві.** **Фальсифікація джерел.** **Плагіат.**	В [47, 48] методы обнаружения ошибок в цифровых схемах объединяются в следующие группы: модификация схем дублирования и сравнения, использование специальных кодов, алгебраический метод, контроль за протеканием процесса.
	С. 46.	**С. 38.**
	У [42] викладаються питання автоматизації контролю цифрових функціональних модулів засобів обчислювальної техніки та радіоелектронної апаратури. Розглядаються технологічні процеси виробництва цифрових функціональних модулів, методи і процедури пошуку несправностей, принципи побудови автоматизованих систем контролю. Підвищення надійності виробів, що виготовляються при одночасному зниженні трудомісткості їхнього контролю, можливо тільки на основі автоматизації операцій контролю і налагодження. Частка трудомісткості контрольних операцій збільшується, що пояснюється вдосконаленням елементної бази, підвищенням її ступеня інтеграції, зростанням складності електронних блоків і модулів. **Коваленко замінила в переписаному чужому тексті покликання на фальшиве.** **Фальсифікація джерел.** **Плагіат.**	В [49] излагаются вопросы автоматизации контроля цифровых функциональных модулей средств вычислительной техники и радиоэлектронной аппаратуры. Рассматриваются технологические процессы производства цифровых функциональных модулей, методы и процедуры поиска неисправностей, принципы построения автоматизированных систем контроля. Повышение надежности изготавливаемых изделий при одновременном снижении трудоемкости их контроля возможно только на основе автоматизации операций контроля и наладки. Доля трудоемкости контрольных операций увеличивается, что объясняется совершенствованием элементной базы, повышением ее степени интеграции, возрастанием сложности электронных блоков и модулей.
	С. 118–119.	**С. 59.**
	Потреба в новій базовій моделі обумовлена необхідністю поліпшення показників і характеристик технічного діагностування електронного модуля бортового обчислювача. Класичні базові моделі ґрунтуються на блочній структурі досліджуваного об'єкта, а в разі неможливості виділення такої структури - на графі причинно-наслідкових зв'язків [34, 52, 101]. Бортові обчислювальні системи, що розглядаються, складаються з декількох конструктивно закінчених електронних модулів. У цьому випадку вони теж можуть бути представлені блочною моделлю. Однак, така модель більше підходить для етапу експлуатації. **Коваленко переписала чужий текст разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела.** **Фальсифікація джерел.** **Плагіат.**	Потребность в новой базовой модели обусловлена необходимостью улучшения показателей и характеристик технического диагностирования электронного модуля бортового вычислителя. Классические базовые модели основываются на блочной структуре исследуемого объекта, а в случае невозможности выделения такой структуры — на графе причинно-следственных связей [34, 52, 101]. Рассматриваемые бортовые вычислительные системы состоят из нескольких конструктивно законченных электронных модулей. В этом случае они тоже могут быть представлены блочной моделью. Однако, такая модель больше подходит для этапа эксплуатации.

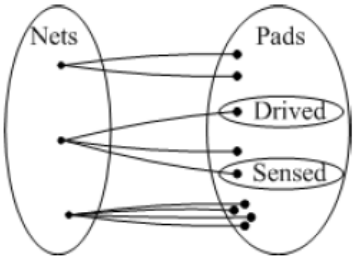
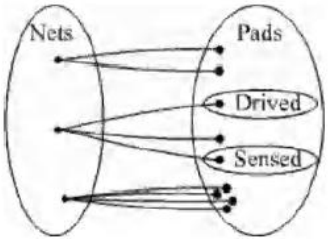
	С. 119.	С. 59–60.
	Для етапів розробки, налагодження і виготовлення існує необхідність в інший базової моделі. Перед тестуванням системи обов'язково тестується кожен модуль окремо. Кожен модуль створюється, починаючи зі схеми електричної принципової. Більшість сучасних схем створюються в тій чи іншій САПР. Потім конструкторам-розробникам друкованої плати передається ця схема, перелік елементів і, найголовніше, список з'єднань або список ланцюгів. Список ланцюгів формується САПР після закінчення розробки схеми і є однозначним описом цієї схеми. Різні САПР створюють списки в різних форматах. Список з'єднань в форматі за стандартом Protel99 1.1 представлений на рис. 29. Спочатку в квадратних дужках перераховуються компоненти: мікросхеми, резистори, конденсатори, діоди та ін. Вказується їх позначення в схемі (DD1, R5, C14, VD2...), типи корпусів, найменування, процентні відхилення та ін. Потім в круглих дужках перераховуються ланцюги. Перший рядок після круглої дужки - найменування ланцюга (NET00009, NET00003, GND, VCC, ADDR1, DATA7...). Наступні рядки аж до круглої дужки, що закриває - контактні площадки компонентів, з'єднаних даним ланцюгом.	Для этапов разработки, отладки и изготовления существует необходимость в другой базовой модели. Перед тестированием системы обязательно тестируется каждый модуль в отдельности. Каждый модуль создается, начиная со схемы электрической принципиальной (в ЕСКД обозначается Э3). Большинство современных схема создаются в той или иной САПР. Затем коиструкторам-раз- работчикам печатной платы передается эта схема, перечень элементов и, самое главное, список соединений или список цепей (NETLIST). Список цепей формируется САПР по окончании разработки схемы и является однозначным описанием этой схемы. Разные САПР создают списки в разных форматах. Список соединений в формате по стандарту Protel99 1.1 представлен на рисунке 2.3. <…> Сначала в квадратных скобках перечисляются компоненты: микросхемы, резисторы, конденсаторы, диоды и пр. Указывается их обозначение в схеме (DD1, R5, C14, VD2...), типы корпусов, наименования, процентные отклонения и др. Затем в круглых скобках перечисляются цепи. Первая строка после круглой скобки - наименование цепи (NET00009, NET00003, GND, VCC, ADDR1, DATA7...). Следующие строки вплоть до закрывающей круглой скобки - контактные площадки компонентов, соединенных данной цепью.
	С. 119.	**С. 60.**
	[DD1 4244.256-3 1986VE8T] … (NET00009 R36-2 S10-3) (NET00003 DA2-4 C9-1) … (…) Рис. 2.9. Фрагмент списку з'єднань за стандартом Protel99 1.1	[DD1 4244.256-3 1986VE8T] (NET00009 R36-2 SI 0-3) (NE TOO 003 DA2-4 C9-1) ( ) Рис. 2.3. Фрагмент списка соединений по стандарту Protel99 1.1
	С. 119–120.	**С. 60–61.**
	Фактично, цей список є моделлю електронного модуля з точністю до контактного випадку (майданчику). Вивчивши структуру в формі списку з'єднань, можна відзначити, що цей список встановлює відповідність *U* між двома множинами: множиною ланцюгів *N* в схемі і множиною контактних майданчиків *P* електронних компонентів схеми. Тоді цю модель умовно можна зобразити у вигляді двудольного графа *G = (N,P,U)*, як показано на рисунок 2.10	Фактически, этот список представляет собой модель электронного модуля с точностью до контактной площадки. 2.3. Построение базовой модели МБВС для этапов разработки, отладки и изготовления Изучив структуру в форме списка соединений, можно отметить, что этот список устанавливает соответствие *U* между двумя множествами: множеством цепей *N* в схеме и множеством контактных площадок *P* электронных ком поцентов схемы. Тогда эту модель условно можно изобразить в виде двудольного графа G = (N, P. U). как показано на рисунке 2.4.

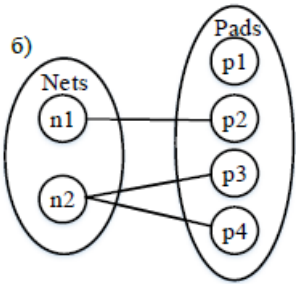
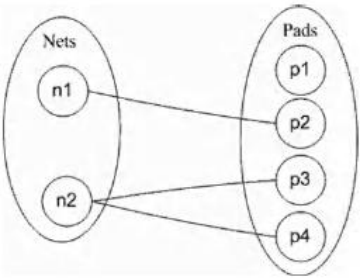
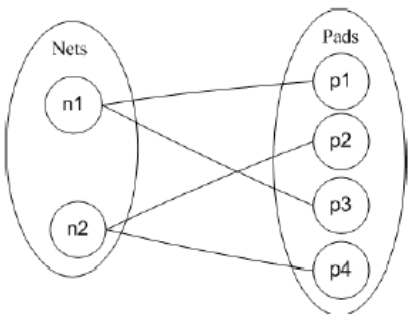
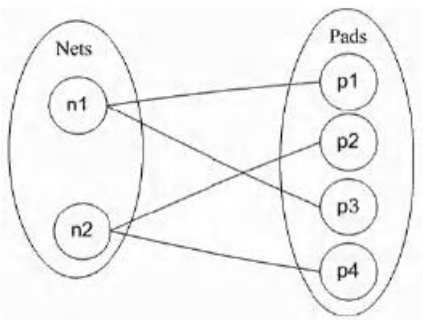
	С. 120.	С. 61.
	 Рис. 2.10. Двочастковий граф ланцюгів і контактних майданчиків	 Рис. 2.4. Двудольный граф цепей и контактных площадок
	С. 120.	**С. 61.**
	Необхідні відомості з теорії графів наводяться згідно [104]. Граф *G* - це впорядкована пара множин *G(V,E)*, де *V* - це множина вершин, а *E* - безліч ребер. Дводольний граф визначається як звичайний граф з хроматичним числом не більше двох і заданим правильним розфарбуванням вершин в два кольори. Іншими словами, граф є дводольним, якщо існує таке розбиття множини його вершин *V*, що $N \cup P = V$ і $N \cap P = \varnothing$, і ребра *U* можуть з'єднувати тільки вершини *N* з вершинами *P*. **Коваленко переписала чужий текст, видаливши заголовок.** **Помилка перекладу: російський вислів «**множество рёбер**» Коваленко переклала як «**безліч ребер**» замість «множина ребер», що спричинено нерозумінням наукової термінології** **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	**2.3.1. Необходимые сведения из теории графов** Необходимые сведения из теории графов приводятся согласно |104]. Граф *G* — это упорядоченная пара множеств *G(V, E)*, где *V* — это множество вершин, а *E* — множество рёбер. Двудольный граф определяется как обыкновенный граф с хроматическим числом не более двух и заданной правильной раскраской вершин в два цвета. Другими словами, граф является двудольным, если существует такое разбиение множества его вершин *V*, что $N \cup P = V$ и $N \cap P = \varnothing$, и ребра *U* могут соединять только вершины *N* с вершинами *P*.
	С. 120–121.	**С. 61–62.**
	Відносини між елементами графа можуть бути виражені в матричному вигляді. Матрицею суміжності називається матриця (4) $$A = \|a_{ij}\|, i, j = \overline{1, |P|},$$ (2.33) елементи якої будуть визначатися відповідно до виразу: $a_{ij} = \begin{cases} 0, \text{ якщо ці вершини не суміжні} \\ 1, \text{ якщо вершини } p_i \text{ та } p_j \text{ суміжні (з'єднані ребром),} \end{cases}$	Отношения между элементами графа могут быть выражены в матричном виде. Матрицей смежности называется матрица (2.1) $$A = \|<ц,-\|, i, j = 1, |P|.$$ (2.1) элементы которой будут определяться согласно выражению: $a_{ij} = \begin{cases} 1, \text{ если вершины } pi \text{ и } pj \text{ смежны (соединены ребром),} \\ О, \text{ — если эти вершины не смежны.} \end{cases}$
	С. 121.	**С. 62.**
	Отже, базова модель у вигляді двудольного графа, згідно рисунку 2.9, вершини в лівій долі графа відповідають елементам множинності лан-	**2.3.2. Базовая модель в виде двудольного графа** Согласно рисунку 2.4, вершины в левой доли

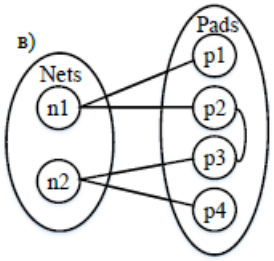
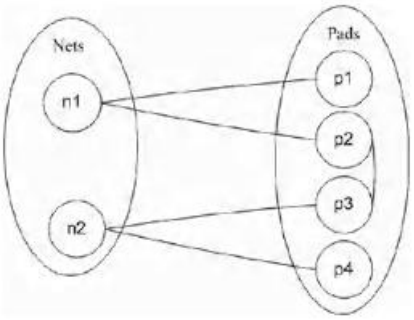
	ціогів електронного модуля $n_i \in N, i = \overline{1,\lvert N\rvert}$, вершини справа – елементам множини контактних майданчиків $p_j \in P, j = \overline{1,\lvert P\rvert}$. Ребра графу $u_k \in U, k = \overline{1,\lvert U\rvert}$ відображають відповідність ланцюгів контактних площадок згідно зі схемою електричної принципової. **Коваленко пише «Отже», ніби це вона робить якийсь висновок, а насправді переписує чужий текст. Коваленко включила заголовок чужого розділу в свій поточний текст.** **Помилка перекладу: російський вислів «согласно схеме электрической принципиальной» Коваленко неправильно переклала як «згідно зі схемою електричної принципової», тоді як треба «згідно зі схемою електричною принциповою».** **Плагіат.**	графа соответствуют элементам множества цепей электронного модуля *щ* € *N, г* = 1, |iV|, вершины справа элементам множества контактных площадок *pj* £ *P, j* = 1,|P|. Ребра графа *Uk* € *U. к* = 1, |C/| отображают соответствие цепей контактным площадкам согласно схеме электрической принципиальной. **Через помилки оцифровки тексту формули записані неправильно.**
	С. 121.	**С. 62.**
	Відповідність справа наліво є однозначною, тобто кожен елемент множини справа (контактна площадка) має тільки один зв'язок з елементом множини зліва (ланцюгом). Іншими словами, одна контактна площадка не може бути з'єднана відразу з двома ланцюгами - це буде означати коротке замикання ланцюгів. При обриві ланцюга в розглянутому графі буде спостерігатися ізольована від інших вершина, що відповідає певному контактному майданчику. Заплутування ланцюгів буде еквівалентно новому графу, відмінному від того, який був заданий схемою електричної принципової. Таким чином, будь-який структурний дефект призведе до порушень інцидентності або навіть зв'язності вихідного графа. **Помилка перекладу: російський вислів «задан схемой электрической принципиальной» Коваленко неправильно переклала як «заданий схемою електричної принципової», тоді як треба «заданий схемою електричною принциповою».** **Коваленко пише «Таким чином», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Плагіат.**	Соответствие справа налево является однозначным, т.е. каждый элемент множества справа (контактная площадка) имеет только одну связь с элементом множества слева (цепью). Другими словами, одна контактная площадка не может быть соединена сразу с двумя цепями — это будет означать короткое замыкание цепей. При обрыве цепи в рассматриваемом графе будет наблюдаться изолированная от других вершина, соответствующая некой контактной площадке. Перепутывание цепей будет эквивалентно новому графу, отличному от того, который был задан схемой электрической принципиальной. Таким образом, любой структурный дефект приведет к нарушениям инцидентности или даже связности исходного графа.
	С. 121.	**С. 62–63.**
	Базова модель у вигляді графа є наочною і представляється зручною для аналізу, але це справедливо лише для списків ланцюгів дуже малих розмірів. У реальності, навіть прості пристрої, що представляються невеликими схемами електричними принциповими, мають сотні контактних майданчиків і десятки ланцюгів. При цьому очевидно, що побудова моделі у вигляді графа і подальший аналіз стають дуже утрудненими. За списком з'єднань можна також побудувати базову модель у вигляді матриці суміжності, елементи якої будуть визначатися в такий спосіб	Базовая модель в виде графа является наглядной и представляется удобной для анализа, но это справедливо лишь для списков цепей очень малых размеров. В реальности, даже простые устройства, представляемые небольшими схемами электрическими принципиальными, имеют сотни контактных площадок и десятки цепей. При этом очевидно, что построение модели в виде графа и последующий анализ становятся весьма затруднительными. По списку соединений можно также построить базовую модель в виде матрицы смежности (2.1), элементы которой будут определяться следующим образом:
	С. 122.	**С. 63.**
	$a_{ij} = \begin{cases} 1, & \text{якщо контактні майданчики } p_i \text{ і } p_j \text{ з'єднані ланцюгом} \\ 0, & \text{якщо з'єднання між майданчиками відсутнє} \end{cases}$	$\begin{cases} 1, & \text{если контактные площадки р* и } pj \text{ соединены цепью,} \\ 0, & \text{— если соединение между площадками отсутствует.} \end{cases}$

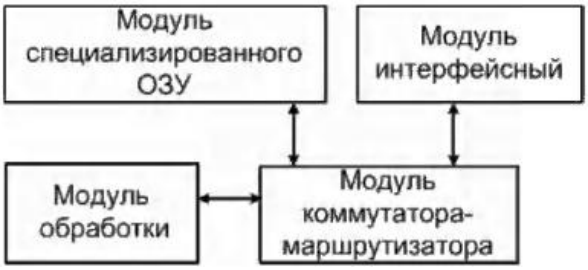
	Матриця суміжностей є симетричною з нулями на головній діагоналі. Елементи матриці згруповані за належністю до ланцюгів схеми. Більшість ланцюгів в схемі з'єднує між собою дві або три контактні площадки. На цьому фоні часто виділяються два ланцюги, що охоплюють відносно велику кількість контактів, - це ланцюги напруги живлення і «землі». Побудована на основі інформації зі списку з'єднань базова модель електронного модуля в графовому або матричному представленні дає потенційну можливість локалізації структурних дефектів модуля з точністю до контактної площадки елемента. Класичні базові моделі, засновані на блочно-функціональних схемах або причинно-наслідкові зв'язки, не дозволяють забезпечити такий ступінь деталізації.	Матрица смежностей является симметричной с нулями на главной диагонали. Элементы матрицы сгруппированы по принадлежности к цепям схемы. Большинство цепей в схеме соединяет между собой две или три контактные площадки. На этом фоне часто выделяются две цепи, охватывающие относительно большое количество контактов, — это цепи напряжения питания и “земли”. Построенная на основе информации из списка соединений базовая модель электронного модуля в графовом или матричном представлении дает потенциальную возможность локализации структурных дефектов модуля с точностью до контактной площадки элемента. Классические базовые модели, основанные на блочно-функциональных схемах или причинно-следственных связях, не позволяют обеспечить такую степень детализации.
	С. 122.	**С. 63–64.**
	Діагностична модель розробляється на основі базової та дозволяє за рахунок інформації зі списку з'єднань (що лежить в основі базової моделі) досягти високої точності локалізації. Ключовим моментом для побудови діагностичної моделі є наявність у схемі електронного модуля хоча б однієї мікросхеми, що підтримує технологію граничного сканування за стандартом IEEE 1149.1 [60]. Технологія граничного сканування також широко відома по найменуванню об'єднаної групи розробки методів тестування, що створила її. **Коваленко переписала чужий текст, видаливши заголовок.** **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	**2.4. Описание технологии граничного сканирования по стандарту IEEE Std. 1149.1-2001** Диагностическая модель разрабатывается на основе базовой и позволяет за счет информации из списка соединений (лежащей в основе базовой модели) достичь высокой точности локализации (вплоть до контактной площадки). Ключевым моментом для построения диагностической модели является наличие в схеме электронного модуля хотя бы одной микросхемы, поддерживающей технологию граничного сканирования по стандарту IEEE 1149.1 [60]. Технология граничного (периферийного) сканирования также широко известна по наименованию создавшей ее объединенной группы разработки методов тестирования (Joint Test Action Group JTAG).
	С. 122–123.	**С. 64–65.**
	В мікросхему з підтримкою стандарту IEEE 1149.1 вбудовується додаткова тестова логіка, основою якої є послідовний зсувний регістр (boundary scan register), що розташовується між функціональним ядром і контактами введення-виведення мікросхеми (на кордоні). Тестова логіка обов'язково включає TAP-контроллер (Test Access Port - порт тестового доступу), регістр команд (Instruction Register) і набір регістрів даних (Data Registers). Обов'язковими за стандартом регістрами даних є регістр граничного сканування (Boundary-Scan Register) і регістр обходу (Bypass Register). Також в тестову логіку можуть бути включені додаткові регістри даних: регістр ідентифікації (Identification Register), допоміжні тестові регістри (при наявності вбудованих в системну логіку тестових засобів, наприклад, для забезпечення покрокового налагодження). **Коваленко переписала чужий текст, видаливши покликання в двох абзацах.**	В микросхему с поддержкой стандарта IEEE 1149.1 встраивается дополнительная тестовая логика, основой которой является последовательный сдвиговый регистр (boundary scan register), располагающийся между функциональным ядром и контактами ввода-вывода микросхемы (на границе) [63]. Тестовая логика обязательно включает TAP-контроллер (Test Access Port порт тестового доступа), регистр команд (Instruction Register) и набор регистров данных (Data Registers). Обязательными по стандарту регистрами данных являются регистр граничного сканирования (Boundary-Scan Register) и регистр обхода (Bypass Register). Также в тестовую логику могут быть включены дополнительные регистры данных: регистр идентификации (Identification Register), вспомогательные тестовые регистры (при наличии встроенных в системную логику тестовых средств, например, для обеспечения пошаговой отладки) [64].

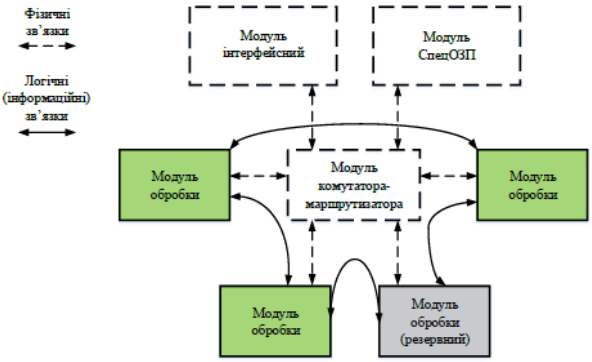
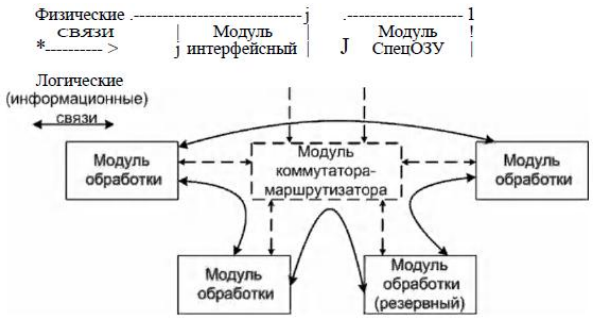
	Помилка перекладу: Коваленко не змогла перекласти російський вислів «ТАР-контроллер». Фальсифікація джерел. Плагіат.	
	С. 123.	**С. 64.**
	Особливості технології граничного сканування на рисинку 2.11.  Рис. 2.11 Структура підтримки технології граничного сканування	Особенности технологии граничного сканирования иллюстрируются в соответствии с рисунком 2.5.  Рис. 2.5. Микросхема с поддержкой технологии граничного сканирования
	С. 123.	**С. 65.**
	ТАР-контролер представляє собою кінцевий автомат з 16 станами: Test-Logic-Reset, Run-Test-Idle і дві ідентичних групи станів (Select, Capture, Shift, Exit1, Pause, Exit2, Update), що відрізняються суфіксом: -IR (для операцій з регістром команд) або -DR (для операцій з регістрами даних). Схема кінцевого автомата показана на рисинку 2.12.	ТАР-контроллер представляет собой конечный автомат с 16 состояниями: Test-Logic-Reset, Run-Test-Idle и две идентичных группы состояний (Select, Capture, Shift, Exitl, Pause, Exit2, Update), отличающихся суффиксом: -IR (для операций с регистром команд) или -DR (для операций с регистрами данных). Схема конечного автомата показана на рисунке 2.6.
	С. 123.	**С. 65.**
	 Рис. 2.12. Кінцевий автомат ТАР-контролера	 Рис. 2.6. Конечный автомат ТАР-контроллера
	С. 124.	**С. 66.**
	Зміна станів автомата здійснюється в залежності від значення сигналу TMS по кожному фронту сигналу TCK. Стандарт IEEE 1149.1 визначає наступні сигнали тестового доступу: – TDI (Test Data Input), – TDO (Test Data Output), – TMS (Test Mode Select input), – TCK (Test ClocK input), – TRST (Test ReSeT input). Через контакти TDI, TDO здійснюється послідовне введення-виведення даних в перераховані	Смена состояний автомата осуществляется в зависимости от значения сигнала TMS по каждому фронту сигнала TCK. Стандарт IEEE 1149.1 определяет следующие сигналы тестового доступа: TDI (Test Data Input), TDO (Test Data Output), TMS (Test Mode Select input), TCK (Test ClocK input), - ,/TRST (Test ReSeT input). Через контакты TDI, TDO осуществляется последовательный ввод-вывод данных в перечис-

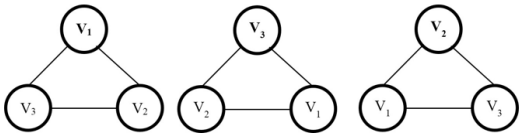
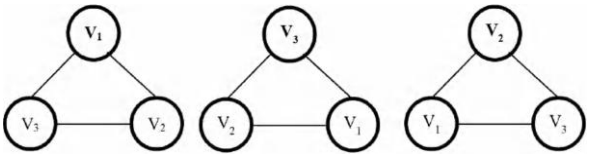
	вище тестові регістри. Через контакт TMS здійснюється управління TAP-контролером. За сигналом TCK здійснюється зміна станів TAP-контролера і зрушення даних від TDI до TDO через будь-який тестовий регістр. Необов'язковий сигнал / TRST надає можливість асинхронного скидання тестової логіки.	ленные выше тестовые регистры. Через контакт TMS осуществляется управление TAP-контроллером. По сигналу TCK осуществляется смена состояний TAP-контроллера и сдвиг данных от TDI к TDO через какой- либо тестовый регистр. Необязательный сигнал /TRST предоставляет возможность асинхронного сброса тестовой логики.
	С. 124.	**С. 66.**
		2.5. Разработка диагностической модели МБ В С для этапов разработки, отладки и изготовления
	Технологія IEEE 1149.1 дозволяє виставляти на контакти мікросхеми, завантажені через регістр граничного сканування тестові вектори і через цей же регістр зчитувати з контактів реакцію на впливи. Функціональне ядро мікросхеми при цьому залишається (повинно залишатися) незадіяним. Пропонована діагностична модель полягає в порівнянні (знаходженні деякого показника близькості) інформації, отриманої в результаті діагностичного експерименту з використанням регістра граничного сканування, з базовою моделлю, заздалегідь створеної за списком ланцюгів даного модуля.	Технология IEEE 1149.1 позволяет выставлять на контакты микросхемы загруженные через регистр граничного сканирования тестовые векторы и через этот же регистр считывать с контактов реакцию на воздействия. Функциональное ядро микросхемы при этом остается (должно оставаться) незадействованным. Предлагаемая диагностическая модель заключается в сравнении (нахождении некоторого показателя близости) информации, полученной в результате диагностического эксперимента с использованием регистра граничного сканирования, с базовой моделью, заранее созданной по списку цепей данного модуля.
	С. 124–125.	**С. 67.**
	З точки зору підтвердження справності шляхом перевірки відсутності в електронному модулі структурних дефектів типу замикань, обривів, змішування ланцюгів доцільно в якості базової моделі розглянути зв'язки між двома множинами: множина ланцюгів *N* (nets) і множина контактних майданчиків *P* (pads). У множині контактних майданчиків можна виділити дві (що представляють інтерес) підмножини: керованих *D* (drived) і спостережуваних *S* (sensed) за допомогою регістра граничного сканування. За допомогою майданчиків підмножини *D* можна виставляти тестові впливи, за допомогою майданчиків підмножини *S* можна зчитувати реакцію. Співвідношення множин ілюструється у відповідності з рисунок 2.13  Рис. 2.13 Множина тестованих ланцюгів і множини керованих і спостережуваних контактних майданчиків	С точки зрения подтверждения исправности путем проверки отсутствия в электронном модуле структурных дефектов типа замыканий, обрывов, перепутывания цепей целесообразно в качестве базовой модели рассмотреть связи между двумя множествами: множество цепей N (nets) и множество контактных площадок P (pads). В множестве контактных площадок можно выделить два (представляющих интерес) подмножества: управляемых D (drived) и наблюдаемых S (sensed) с помощью регистра граничного сканирования. С помощью площадок подмножества D можно выставлять тестовые воздействия, с помощью площадок подмножества S можно считывать реакцию. Соотношение множеств иллюстрируется в соотвествии с рисунком 2.7.  Рис. 2.7. Множество тестируемых цепей и множества управляемых и наблюдаемых контактных площадок
	С. 125.	**С. 67.**
	Як випливає з рис. 2.13, далеко не всі ланцюги можуть бути протестовані. В цьому відношенні, чим більше мікросхем з підтримкою технології граничного сканування містить схема, тим краще. У справної схеми кола не перетинаються, кожному контактному майданчику ставиться у відповідність один ланцюг. Кожному ланцюгу може відповідати кілька контактних майданчиків. Еталонна матриця *A* (без несправностей) складається за схемою. Обмеженням даної моделі є те, що елементами еталонної матриці можуть бути тільки майданчики, які керуються і спостерігаються за	Как следует из рисунка 2.7, далеко не все цепи могут быть протестированы. В этом отношении, чем больше микросхем с поддержкой технологии граничного сканирования содержит схема, тем лучше. В исправной схеме цепи не пересекаются, каждой контактной площадке ставится в соответствие одна цепь. Каждой цепи может соответствовать несколько контактных площадок. Эталонная матрица A (без неисправностей) составляется по схеме. Ограничением данной модели является то, что элементами эталонной матрицы могут являться

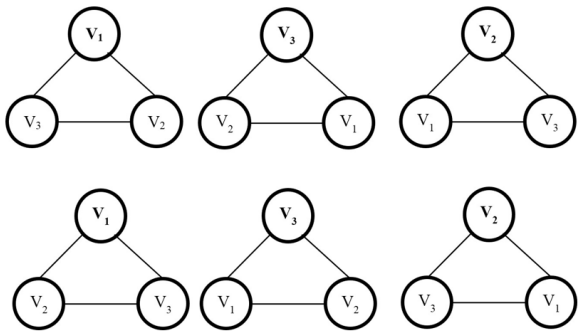
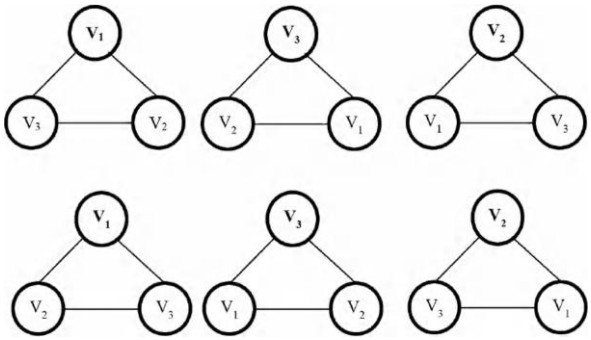
	допомогою технології граничного сканування. Наприклад, на рис. 28 показаний граф, що відповідає справній схемі, в якій майданчики p_1 і p_2, а також p_3 і p_4 з'єднані двома різними ланцюгами.	только площадки, управляемые и наблюдаемые с помощью технологии граничного сканирования. Например, на рисунке 2.8 показан граф, соответствующий исправной схеме, в которой площадки рі и p_2. а также рз и р± соединены двумя разными цепями.
	С. 125–126.	**С. 69.**
	При цьому матриця суміжності графа буде відображатися у вигляді: $A = \begin{Vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{Vmatrix}$	При этом матрица смежности графа будет отображаться в виде (2.2). $\begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$
	С. 126.	**С. 68.**
	Інформація, отримана в результаті діагностичного експерименту, може бути представлена у вигляді матриці . Наприклад, на рисинку 2.14 показана діагностична модель у вигляді графа, коли в схемі є обрив ланцюгів.  Рис. 2.14 Діагностична модель у вигляді графа для обриву ланцюга	Информация, полученная в результате диагностического эксперимента, может быть представлена в виде матрицы В. Например, на рисунке 2.9 показана диагностическая модель в виде графа, когда в схеме есть обрыв цепей.  Рис. 2.9. Диагностическая модель в виде графа для обрыва цепи
	С. 126.	**С. 69.**
	Матриця суміжності для цього випадку буде відображатися у вигляді: $B_o = \begin{Vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{Vmatrix}$ Таким чином, при обриві ланцюга в матриці суміжності будуть 0 замість 1.	Матрица смежности для этого случая будет отображаться в виде (2.3). $B_0 = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$ (2.3) Таким образом, при обрыве цепи в матрице смежности будут 0 вместо 1
	С. 126.	**С. 69.**
	На рисунку 2.15 показаний граф для змішування ланцюгів.  Рис. 2.15 Діагностична модель змішування ланцюгів	На рисунке 2.10 показан граф для перепутывания цепей.  Рис. 2.10. Диагностическая модель перепутывания цепей

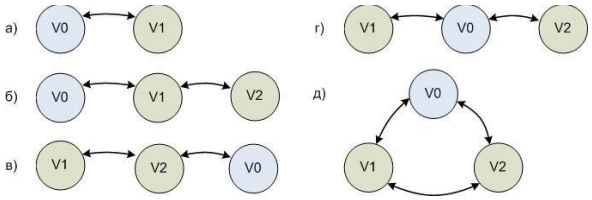
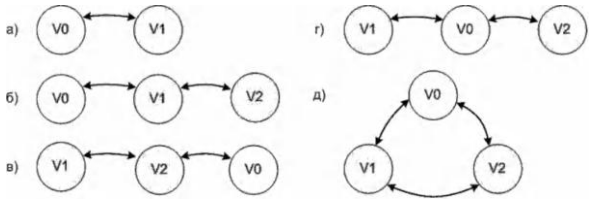
	С. 127.	**С. 69.**
	Матриця суміжності для змішування ланцюгів буде представлена у вигляді: $B_c = \begin{Vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{Vmatrix}$	Матрица смежности для перепутывания цепей будет представлена в виде (2-4) $B_c = \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix}$ (2.4)
	С. 127.	**С. 69–70.**
	На рисунку 2.16 проілюстрована діагностична модель для короткого замикання ланцюгів. При цьому матриця суміжності графа буде відображатися у вигляді: $B_s = \begin{Vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{Vmatrix}$  Рис. 2.16 Діагностична модель у вигляді графа для короткого замикання ланцюгів	На рисунке 2.11 проиллюстрирована диагностическая модель для короткого замыкания цепей. При этом матрица смежности графа будет отображаться в виде (2.5). $B^* = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$ (2.5)  Рис. 2.11. Диагностическая модель в виде графа для короткого замыкания цепей
	С. 127–128.	**С. 70.**
	При замиканні або переплутуванні ланцюгів в матриці суміжності будуть або зайві 1, або 1 не на своєму місці. Пошук структурного дефекту зводиться до обчислення виразу $C = A \oplus B$, де $\oplus$ - операція «що виключає АБО». При $C = 0$ виходить збіг інформації, отриманої в результаті діагностичного експерименту, з еталонною. При $C \neq 0$ в схемі міститься структурний дефект, стан якого відображається ненульовими елементами матриці C. Наприклад, для обриву ланцюга буде справедливо: $C = A \oplus B = \begin{Vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{Vmatrix}$ Представлені діагностичні моделі можуть застосовуватися для електронних модулів як МБВС-М, так і МБВС-П.	При замыкании или перепутывании цепей в матрице смежности будут или лишние 1, или 1 не на своем месте. Поиск структурного дефекта сводится к вычислению выражения (2.6) $C = A \phi B$, (2.6) где ® операция “исключающее ИЛИ”. При $C = 0$ получается совпадение информации, полученной в результате диагностического эксперимента, с эталонной. При $C \Phi 0$ в схеме содержится структурный дефект, положение которого отображается ненулевыми элементами матрицы C. Например, для обрыва цепи будет справедливо (2.7) $C — A ® B\ o — \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}$ (2.7) Представленные диагностические модели применимы для электронных модулей как МБВС-М, так и МБВС-П.
	С. 128.	**С. 71.**
		2.6. Разработка базовой модели МБВС для этапа эксплуатации

	Для аналізу на етапі експлуатації в якості вихідної обрана структура бортової обчислювальної системи, спрощено зображеної на рисинку 2.17  Рис. 2.17. Вихідна структура бортової обчислювальної системи **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	Для анализа на этапе эксплуатации в качестве исходной выбрана структура бортовой вычислительной системы, упрощенно изображенная на рисунке 2.12.  Рис. 2.12. Исходная структура бортовой вычислительной системы
	С. 128.	**С. 71.**
	Система будується на основі високошвидкісної комутаційної мережі SpaceWire. Детальна структурна схема і елементна база такої системи розглянуті в розділі 2.1. Модуль комутатора-маршрутизатора має вільні порти, які можуть бути використані для масштабування або резервування. При нарощуванні модулів обробки структуру можна віднести до класу багатомашинних, коли кожен процесор, що обробляється, має власну пам'ять (як постійну, так і оперативну) [75]. Багатомашинні структури дозволяють забезпечити одночасне підвищення продуктивності та відмовостійкості. При цьому, зрозуміло, алгоритми управління повинні допускати можливість розпаралелювання. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	Система строится на основе высокоскоростной коммутационной сети Space Wire. Подробная структурная схема и элементная база такой системы рассмотрены в разделе 2.1 в части описания МБВС-П. Модуль коммутатора-маршрутизатора имеет свободные порты, которые могут быть использованы для масштабирования или резервирования. При наращивании модулей обработки структуру можно отнести к классу многомашинных, когда каждый обрабатывающий процессор имеет собственную память (как постоянную, так и оперативную) |75]. Многомашинные структуры позволяют обеспечить одновременное повышение производительности и отказоустойчивости. При этом, разумеется, алгоритмы управления должны допускать возможность распараллеливания.
	С. 128–129.	**С. 71–72.**
	В даному прикладі надмірність вводиться тільки для модулів обробки в припущенні, що вони з більшою ймовірністю схильні до відмов в порівнянні з іншими модулями. Припущення базується на традиційному уявленні, що найменш надійними елементами обчислювальної техніки є мікросхеми оперативних запам'ятовуючих пристроїв [105]. Мікросхеми, що застосовуються в модулі спеціалізованого ОЗУ для зберігання проміжних результатів, мають підвищений рівень безперебійної роботи, проте їх інформаційна ємність недостатня для застосування в модулях обробки. Модулі інтерфейсний і комутатора-маршрутизатора побудовані на основі спеціалізованих НВІС, які (в порівнянні з ОЗУ) також менш схильні до збоїв або відмов. Тому надмірність далі розглядається тільки відносно модулів обробки.	В данном примере избыточность вводится только для модулей обработки в предположении, что они с большей вероятностью подвержены отказам по сравнению с другими модулями. Предположение основано на традиционном представлении, что наименее надежными элементами вычислительной техники являются микросхемы оперативных запоминающих устройств [105]. Микросхемы, применяемые в модуле специализированного ОЗУ для хранения промежуточных результатов, имеют повышенный уровень бессбойной работы, однако их информационная емкость недостаточна для применения в модулях обработки. Модули интерфейсный и коммутатора-маршрутизатора построены на основе специализированных СБИС, которые (по сравнению с ОЗУ) также менее подвержены сбоям или отказам. Поэтому избыточность далее рассматривается только в отношении модулей обработки.

	Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело – її власна стаття. **Фальсифікація джерел.** **Плагіат.**	
	С. 129.	**С. 72.**
	Також слід зазначити, що в даній роботі розглядається мінімальна надмірність, а саме та, яка дозволяє зберегти працездатність при відмові тільки одного модуля. Відповідно, в структурній схемі додається тільки один резервний модуль обробки, як показано на рисунку 2.18  Рис. 2.18 Багатомашинна структура з резервним модулем Відмовостійкість в більш загальному випадку, тобто коли відмовляють модулів, розглядається в [106].	Также следует отметить, что в данной работе рассматривается минимальная избыточность, а именно позволяющая сохранить работоспособность при отказе только одного модуля. Соответственно, в структурной схеме добавляется только один резервный модуль обработки, как показано на рисунке 2.13.  Рис. 2.13. Многомашинная структура с резервным модулем Отказоустойчивость в более общем случае, т.е. когда отказывают к модулей, рассматривается в [106].
	Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело – її власна стаття. **Фальсифікація джерел.** **Плагіат.**	
	С. 129–130.	**С. 72–73.**
	Згідно рисунку 2.18 апаратна частина бортової обчислювальної системи має повнозв'язну структуру. Організація логічних зв'язків, тобто розподіл потоків даних і керуючої інформації при паралельній обробці, менш очевидна і може бути реалізована різними способами при розробці програмного забезпечення системи. Пошук спільних рекомендацій з проектування системи на логічному рівні, що дозволяють забезпечити розглянутий рівень відмовостійкості, і становить предмет подальшого розгляду. Структура моделюється у вигляді графа, вершини якого відповідають модулям обробки, а ребра - логічним (інформаційним) зв'язкам між ними. Для визначеності резервний модуль обробки відповідає вершині *V*0. Застосовувана модель передбачає, що відмовам схильні тільки вершини графа.	Согласно рисунку 2.13, аппаратная часть бортовой вычислительной системы имеет полносвязную структуру. Организация логических связей, т.е. распределение потоков данных и управляющей информации при параллельной обработке, менее очевидна и может быть реализована разными способами при разработке программного обеспечения системы. Поиск общих рекомендаций по проектированию системы на логическом уровне, позволяющих обеспечить рассмотренный уровень отказоустойчивости, и составляет предмет дальнейшего рассмотрения. Структура моделируется в виде графа, вершины которого соответствуют модулям обработки, а ребра - логическим (информационным) связям между ними. Для определенности резервный модуль обработки соответствует вершине VO. Применяемая модель предполагает, что отказам подвержены только вершины графа.
	С. 130.	**С. 73.**
		2.7. Необходимые сведения из теории групп

	Абстрактною групою називається множина, кінцева або нескінченна, для якої виконуються умови [107]: 1. $g_i * g_j = g_s \in G,$ де * асоціативна операція 2. $\exists g_k = e \in G$ такий, що $e * g_k \in G$ 3. $\exists g_i \in G, \exists g_i^{-1} * g_i = e$ **Коваленко переписала чужий текст, видаливши заголовок.** **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело – її власна стаття.** **Фальсифікація джерел.** **Плагіат.**	Абстрактной группой G называется множество, конечное или бесконечное, для которого выполняются условия 1107]: 1. *cjj * gj = g_s* € G, где * ассоциативная операция, 2. Hgk = e € G такой, что e* дк = дк € G, 3. Зд{ в G, Эg~^{l} € G и д~^{1} * д, = е. **У цьому файлі з дисертацією Сильянова – помилки оцифровки тексту в формулах і в номері покликання: треба [107].**
	С. 130.	**С. 73.**
	Група симетрії деякого об'єкта (багатогранника або множини точок з метричного простору) - це група всіх рухів, для яких даний об'єкт є інваріантом, з композицією в якості групової операції. Далі будуть розглянуті циклічна, діедральна та тетраедральна групи симетрії [108]. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело – її власна стаття.** **Фальсифікація джерел.** **Плагіат.**	Группа симметрии некоторого объекта (многогранника или множества точек из метрического пространства) — это группа всех движений, для которых данный объект является инвариантом, с композицией в качестве групповой операции. Далее будут рассмотрены циклическая, диэдральная и тетраэдральная группы симметрии [108].
	С. 130.	**С. 74.**
	Циклічна група (C_n) симетрії являє собою групу циклічних поворотів на кут $\alpha = \frac{360^o}{N}$ навколо осі, що проходить через центр О системи, як показано на рисунку 2.19.  Рис. 2.19 Група автоморфізмів графа G_3 циклічної групи симетрії	Циклическая группа симметрии (Сп) представляет собой группу циклических поворотов на угол а = 360°^ вокруг оси, проходящей через центр О системы, как показано на рисунке 2.14.  Рис. 2.14. Группа автоморфизмов графа G_3 циклической группы симметрии
	С. 131.	**С. 74.**
	Дана група має порядок $\|C_n\|=N=n$ отже група містить автоморфізмів [108]. Діедральна група (D_n) симетрії складається з власних обертань в просторі на кути $\alpha = \frac{360^o}{2N}$, навколо осі, що проходить через центр О і на 180° навколо N/2 осей другого порядку. Порядок групи $\|D_n\|=N=2n$ отже, група містить 2n автоморфізмів, як показано на рисунку 2.20: тотожний автоморфизм; – 1 поворотів на кути 2πk/n, де = 1 . . . −1; відображень щодо осей, що проходять через центр О і утворюючі між собою кути, кратні 2π/n [108]. **Коваленко переписала чужий текст, двічі разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело – її власна**	Данная группа имеет порядок \$C_n$\ = N = п, следовательно группа содержит *п* автоморфизмов [108]. Диэдральная группа симметрии (Dn) состоит из собственных вращений в пространстве на углы *а* = 360°1 V, вокруг оси, проходящей через центр О, и на 180 °вокруг *N/2* осей второго порядка. Порядок группы \$D_n$\ = N = 2п, следовательно, группа содержит 2п автоморфизмов, как показано на рисунке 2.15: тождественный автоморфизм; п— 1 поворотов на углы 2т*тк/п*: где *к* = 1... п—1; *п* отражений относительно осей, проходящих через центр *О* и образующих между собой углы, кратные *2п/п* [108]. **У цьому файлі з дисертацією Сильянова – помилки оцифровки тексту.**

	стаття. **Фальсифікація джерел.** **Плагіат.**	
	С. 131.	**С. 74.**
	Тетраедральна група симетрії тісно пов'язана з представленням плоского графа у вигляді просторового. Тетраедр має 4 осі симетрії, що проходять через його вершини і центри протилежних граней. Навколо кожної осі крім тотожного можливі ще два обертання. Таким чином, група обертань тетраедра має 12 перестановок, включаючи тотожне [108]. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело – її власна стаття.** **Фальсифікація джерел.** **Плагіат.**	Тетраэдральную группы симметрии (Т) тесно связана с представлением плоского графа в виде пространственного. Тетраэдр имеет 4 оси симметрии, проходящие через его вершины и центры противолежащих граней. Вокруг каждой оси кроме тождественного возможны еще два вращения. Таким образом, группа вращений тетраэдра имеет 12 перестановок, включая тождественное [108].
	С. 131.	**С. 75.**
	 Рис. 2.20. Група автоморфізмів графа G_3 діедральної групи симетрії	 Рис. 2.15. Группа автоморфизмов графа G$ диэдральной группы симметрии
	С. 131–132.	**С. 75.**
	Структура МБВС-П моделюється у вигляді графа, вершини якого відповідають модулям обробки, а дуги - логічним (інформаційним) зв'язкам між ними. Для визначеності резервний модуль обробки відповідає вершині …V0. Застосовувана модель передбачає, що до відмов схильні тільки вершини графа. Комутаційна мережа до відмов не схильна, тобто при відмові модуля обробки можна здійснити реконфігурацію системи відповідно до рисунку 2.21 за рахунок перебудови роботи модуля комутатора - маршрутизатора і замінити модуль обробки, що відмовив на резервний.	2.8. Разработка диагностической модели МБВС для этапа эксплуатации Структура МБВС-П моделируется в виде графа, вершины которого соответствуют модулям обработки, а дуги - логическим (информационным) связям между ними. Для определенности резервный модуль обработки соответствует вершине V0. Применяемая модель предполагает, что отказам подвержены только вершины графа. Коммутационная сеть отказам не подвержена, т.е. при отказе модуля обработки можно осуществить реконфигурацию системы в соответствии с рисунком 2.13 за счет перестройки работы модуля коммутатора-маршрутизатора и заменить отказавший модуль обработки на резервный.
	С. 132.	**С. 75–76.**
	У найпростішому випадку в системі два модуля, при цьому резервний модуль заміщає той, що відмовив, як показано на рис. 2.21 а. Такий граф відповідає формату двоканального обчислювача з «холодним» резервом. При організації логічних зв'язків в структурі з трьома модулями слід врахувати, що в ланцюжках на рис. 2.21 б і рис. 2.21 в резервний модуль може замістити лише один з основних. Більшою відмовостійкістю будуть володіти структур и з організацією логічних зв'язків, показані на рис. 2.21 г і рис. 2.21 д. У цих випадках резервний модуль здатний замінити будь-який з основних.	В простейшем случае в системе два модуля, при этом резервный модуль замещает отказавший, как показано на рисунке 2.16а. Такой граф соответствует двухканальному вычислителю с “холодным” резервом, показанному на рисунке 1.12. При организации логических связей в структуре с тремя модулями следует учесть, что в цепочках на рисунке 2.166 и рисунке 2.16в резервный модуль может заместить лишь один из основных. Большей отказоустойчивостью будут обладать структуры с организацией логических связей, показанные на рисунке 2.16г и рисунке 2.16д. В этих

		случаях резервный модуль способен заменить любой из основных.
	С. 132.	**С. 76.**
	 Рис. 2.21: Варіанти надлишкових структур з двома і трьома модулями обробки	 Рис. 2.16. Варианты избыточных структур с двумя и тремя модулями обработки
	С. 132–133.	**С. 76.**
	При збільшенні розмірності системи відповідно зростає кількість варіантів з'єднань між модулями, при цьому організація логічних зв'язків стає менш очевидною. Як приклад на рисунку 2.20 показані структури з чотирма модулями обробки. На рис. 2.21а – 2.21г вершина V0 є суміжною лише з однією з вершин графа, тобто резервний модуль здатний замінити тільки один з основних. Кращі можливості щодо забезпечення відмовостійкості відповідно до рис. 2.21ж є у повнозв'язного графа. При подальшому зростанні розмірності системи організація такої структури буде ставати складніше і дорожче. Аналізуючи два приклади, що залишились, слід зазначити, що замінити будь-який з основних модулів на резервний, не порушивши відносин інцидентності, можна лише для випадку на рис. 2.21д.	При увеличении размерности системы соответственно возрастает количество вариантов соединений между модулями, при этом организация логических связей становится менее очевидной. В качестве примера на рисунке 2.17 показаны структуры с четырьмя модулями обработки. На рисунках 2.17а-2.17г вершина V0 является смежной лишь с одной из вершин графа, т.е. резервный модуль способен заменить только один из основных. Лучшие возможности по обеспечению отказоустойчивости в соответствии с рисунком 2.17ж имеются у полносвязного графа. При дальнейшем росте размерности системы организация такой структуры будет становиться сложнее и дороже. Анализируя оставшиеся два примера, следуют отметить, что заменить любой из основных модулей на резервный, не нарушив отношений инцидентности, можно лишь для случая на рисунке 2.17д.
	С. 133.	**С. 76–77.**
	У загальному випадку показано, що відмовостійкість з мінімально можливою надмірністю можна реалізувати тільки в структурах, що володіють певного роду математичною симетрією. Заміна модуля, що відмовив, на резервний в рамках застосовуваної моделі означає перейменування логічних імен вершин графа таким чином, що надлишкова вершина заміщає ту, що відмовила, без зміни інцидентності між новими логічними іменами[109]. Таке перетворення графа є інваріантним і також називається автоморфизмом. Сукупність автоморфізмів утворює групу симетрії. Знаючи групу симетрії, якою володіє система, можна скласти таблицю групових перетворень. При цьому число перетворень дорівнює порядку групи симетрії. Зазначена таблиця буде однозначно визначати процес реконфігурації після відмови. Для системи, показаної на рис. 2.21, діагностику і реконфігурацію повинен здійснювати модуль комутатора-маршрутизатора, відповідно таблиці реконфігурації повинні зберігатися у нього в пам'яті. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело – її власна стаття.** **Фальсифікація джерел.** **Плагіат.**	В общем случае в [109] показано, что отказоустойчивость с минимально возможной избыточностью реализуема только в структурах, обладающих определенного рода математической симметрией. Замена отказавшего модуля на резервный в рамках применяемой модели означает переименование логических имен вершин графа таким образом, что избыточная вершина замещает отказавшую без изменения инцидентности между новыми логическими именами. Такое преобразование графа является инвариантным и также называется автоморфизмом. Совокупность автоморфизмов образует группу симметрии. Далее в качестве примера будут рассмотрены циклическая, диэдральная и тетраэдральная группы симметрий. Зная группу симметрии, которой обладает система, можно составить таблицу групповых преобразований. При этом число преобразований равно порядку группы симметрии. Указанная таблица будет однозначно определять процесс реконфигурации после отказа. Для системы, показанной на рисунке 2.13, диагностику и реконфигурацию должен осуществлять модуль коммутатора-маршрутизатора, соответственно таблицы реконфигурации должны храниться у него в памяти.

	С. 134.	**С. 78.**
	(2.5 Висновки до другого розділу) Для розробки, налагодження і виготовлення важливо забезпечити пошук дефекту з глибиною пошуку до ланцюга або до контактної площадки. Тому елементами графа є ланцюги і відповідні їм контакти. При цьому побудована діагностичної моделі на основі списків з'єднань і з застосуванням технології граничного сканування. Це дозволяє збільшити глибину пошуку структурних дефектів (таких як обриви, короткі замикання або змішування ланцюгів), що знижує ризики втрат різного виду. Для етапу експлуатації бортових систем важливим є забезпечення відмовостійкості при впливі несприятливих факторів. В цьому випадку діагностична модель не вимагає високої деталізації. Необхідно встановити факт правильного функціонування, а в разі порушення роботи виконати реконфігурацію системи. Тому елементами графа в цьому випадку є модулі обробки. При цьому як витрати на апаратурну надмірність, так і витрати часу на відновлення повинні бути мінімальні. Це можна досягти для систем, структура яких описується певною алгебраїчної моделлю. **Доказом плагіату є помилка перекладу російської фрази «**целесообразно построение диагностической модели**», яку Коваленко переклала зі стилістичною помилкою: «**побудована діагностичної моделі**». Вислів «**перепутывания цепей**» неправильно перекладений як «**змішування ланцюгів**».** **Плагіат.**	Для разработки, отладки и изготовления важно обеспечить поиск дефекта с глубиной поиска до цепи или до контактной площадки. Поэтому элементами графа являются цепи и соответствующие им контакты. При этом целесообразно построение диагностической модели на основе списков соединений и с применением технологии граничного сканирования. Это позволяет увеличить глубину поиска структурных дефектов (таких как обрывы, короткие замыкания или перепутывания цепей), что снижает риски потерь различного вида. Для этапа эксплуатации бортовых систем важно обеспечение отказоустойчивости при воздействии неблагоприятных факторов (климатических, механических, излучений и помех различного рода). В этом случае диагностическая модель не требует высокой детализации. Необходимо установить факт правильного функционирования, а в случае нарушения работы выполнить реконфигурацию системы. Поэтому элементами графа в этом случае являются модули обработки. При этом как затраты на аппаратурную избыточность, так и затраты времени на восстановление должны быть минимальны. Это достижимо для систем, структура которых описывается определенной алгебраической моделью. Например, организация взаимодействия составных частей должна соответствовать некой группе симметрии.
	С. 231.	**С. 32.**
	Найбільш доцільним методом контролю є дублювання, тому що обидва автомата ідентичні, їхні входи об'єднані і обидва працюють від єдиної системи синхронізації. Мажоритарні схеми застосовуються для контролю найбільш відповідальних вузлів у тих випадках, коли інші способи застосувати важко. Найбільш поширені на практиці способи, засновані на зіставленні вихідних сигналів (числовий контроль за модулем, корегувальні коди). Кодовий контроль за модулем відрізняється від числового тим, що в якості контрольних слів використовуються залишки від ділення суми цифр даного слова на обраний модуль контролю (парність-непарність).	Наиболее простым методом контроля является дублирование, т.к. оба автомата идентичны, их входы объединены и оба работают от единой системы синхронизации. Мажоритарные схемы применяются для контроля наиболее ответственных узлов в тех случаях, когда другие способы применить трудно. Наиболее распространены на практике способы, основанные на сопоставлении выходных сигналов (числовой контроль по модулю, корректирующие коды). Кодовый контроль по модулю отличается от числового тем, что в качестве контрольных слов используются остатки от деления суммы цифр данного слова на выбранный модуль контроля (четность-нечетность). Корректирующие коды применяются при передаче и хранении информации, при этом автомат А информацию не перерабатывает. При этом получается меньший объем аппаратуры и выше эффективность ее работы.
	С. 231.	**С. 31–32.**
	Розробили класифікацію методів апаратурного контролю, рисунок 4.10	Выбрав в качестве признака контрольные соотношения, положенные в основу построения автомата В и узла сопоставления .D, можно произвести классификацию методов аппаратурного контроля согласно рисунку 1.5.

Рис. 4.10. Класифікація методів апаратурного контролю

Коваленко пише «Розробили класифікацію», а насправді переписує чужий текст і чужу схему. Плагіат.

Рис. 1.5. Классификация методов аппаратурного контроля

У цьому рисунку – дефект через оцифровку тексту.

С. 232. | **С. 33.**

Визначено набір показників, які можна використовувати при проєктуванні систем діагностування, а також для порівняння варіантів систем діагностування. Вимоги ефективності передбачають вимоги з надійності, аналіз яких для більшості випадків показує необхідність діагностування та відновлення. Від системи діагностування потрібно зафіксувати місце виникнення дефекту і сформувати команду на реконфігурацію об'єкта.

Коваленко пише «Визначено набір показників», нібито це вона таке зробила, а насправді переписує чужий текст. Плагіат.

В [41] излагаются принципы выбора показателей системы диагностирования дискретных объектов. Предлагается набор показателей, которые можно использовать при проектировании систем диагностирования, а также для сравнения вариантов систем диагностирования. Требования эффективности влекут требования по надежности, анализ которых для большинства случаев показывает необходимость диагностирования и восстановления. От системы диагностирования требуется зафиксировать место возникновения дефекта и сформировать команду на реконфигурацию объекта.

С. 232. | **С. 34.**

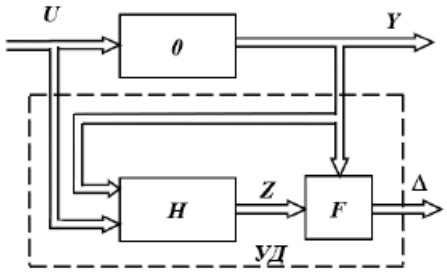
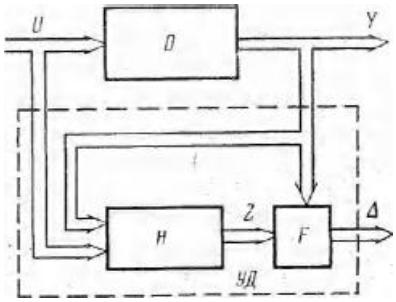
Класифікація методів і принципів функціонального діагностування показана на рис. 19.

Рис. 19. Класифікація методів функціонального діагностування

Классификация методов и принципов функционального диагностирования показана на рисунке 1.6:

Рис. 1.6. Классификация методов функционального диагностирования

	С. 232.	**С. 32–33.**
	Корегувальні коди застосовуються при передачі і зберіганні інформації, при цьому автомат A інформацію не переробляє. У той же час виходить менший обсяг апаратури і підвищується ефективність її роботи. Між тестами для перевірки та діагностичними тестами немає чіткої межі. Як правило, діагностичні тести будуються на базі тестів для перевірки, тобто використовують відомості про стан пристрою, одержані в результаті застосування тестів для перевірки. Для діагностики складних пристроїв, особливо якщо необхідного часу діагностики замало, необхідно застосовувати умовні послідовні тести. Пристрій, що підлягає діагностиці, розглядається як кінцевий автомат з відомою внутрішньою структурою.	<…> Корректирующие коды применяются при передаче и хранении информации, при этом автомат А информацию не перерабатывает. При этом получается меньший объем аппаратуры и выше эффективность ее работы. Между проверяющими и диагностическими тестами нет четкой границы. Как правило, диагностические тесты строятся на базе проверяющих, т.е. используют сведения о состоянии устройства, полученные в результате применения проверяющих тестов. Для диагностики сложных устройств, особенно если требуемое время диагностики мало, необходимо применять условные последовательные тесты. Устройство, подлежащее диагностике, рассматривается как конечный автомат с известной внутренней структурой.
	С. 233–234.	**С. 34–35.**
	При контролі технічного стану об'єкта в просторі параметрів визначаються їхні поточні значення (коефіцієнти передавальних функцій, постійні часу та ін.) і оцінюється відхилення їх від номінального значення. Номінальні значення зазвичай бувають відомі, труднощі пов'язані зі складністю вимірювань поточних значень. При контролі в просторі сигналів перевіряється відхилення вихідних сигналів об'єкта і його блоків від теоретичних значень. При цьому проблема, навпаки, полягає в необхідності безперервного визначення номінальних значень вихідних сигналів для поточних значень вхідних сигналів. Основне завдання ідентифікації полягає в отриманні або уточненні математичного опису об'єкта за вимірюваннями його вхідного і вихідного сигналів. Методи ідентифікації спрощуються при використанні апріорної інформації про номінальні значення параметрів справного об'єкта і моделі дефектів. При розпізнаванні образів поточний стан об'єкта характеризується вектором у просторі діагностичних ознак, які розділені на області, що відповідають тим чи іншим дефектам.	При контроле технического состояния объекта в пространстве параметров определяются их текущие значения (коэффициенты передаточных функций, постоянные времени и др.) и оценивается отклонение их от номинального значения. Номинальные значения обычно бывают известны, трудность связана со сложностью измерений текущих значений. При контроле в пространстве сигналов проверяется отклонение выходных сигналов объекта и его блоков от теоретических значений. При этом проблема наоборот состоит в необходимости непрерывного определения номинальных значений выходных сигналов для текущих значений входных сигналов.Основная задача идентификации состоит в получении или уточнении математического описания объекта по измерениям его входного и выходного сигналов. Методы идентификации упрощаются при использовании априорной информации о номинальных значениях параметров исправного объекта рі модели дефектов. При распознавании образов текущее состояние объекта характеризуется вектором в пространстве диагностических признаков, которые разбиты на области, соответствующие тем или иным дефектам.
	С. 234.	**С. 35–36.**
	Метод словників дефектів або діагностичних таблиць (таблиць несправностей) передбачає складання списків найбільш характерних несправностей і відповідних їм значень діагностичних ознак. Зазвичай під несправністю розуміють неприпустиме відхилення деякого параметра від номінального значення, а її ознакою служить перевищення граничного рівня деяким сигналом. Тому даний метод застосовується і в просторі ознак, і в просторі сигналів. Функціональне діагностування динамічних об'єктів в просторі сигналів точніше відповідає змістовній меті перевірки правильності функціонування об'єкта. В першу чергу для об'єктів, призначення яких – у перетворенні вхідних сигналів на вихідні. При використанні простору параметрів виконання цієї основної функції перевіряється побічно за значеннями параметрів об'єкта. У просторі сигналів правильність вироблення вихідних сигналів перевіряється безпосередньо шляхом їхнього аналізу. Одним з методів при цьому є аналіз характеристик сигналів (амплітуди, частотних властивостей та ін.). Недоліками цієї групи є необхідність апріорної інформації про характеристики вихідних сигналів,	Метод словарей дефектов или диагностических таблиц (таблиц неисправностей) подразумевает составление списков наиболее характерных неисправностей и соответствующих им значений диагностических признаков. Обычно под неисправностью понимают недопустимое отклонение некоторого параметра от номинального значения, а ее признаком служит превышение предельного уровня некоторым сигналом. Поэтому данный метод применяется и в пространстве признаков, и в пространстве сигналов. Функциональное диагностирование динамических объектов в пространстве сигналов точнее соответствует содержательной цели проверки правильности функционирования объекта. В первую очередь для объектов, назначение которых в преобразовании входных сигналов в выходные. При использовании пространства параметров выполнение этой основной функции проверяется косвенно по значениям параметров объекта. В пространстве сигналов правильность выработки выходных сигналов проверяется непосредственно путем их анализа. Одним из методов при этом является анализ характеристик сигналов (амплитуды, ча-

	неминуча залежність характеристик від вхідних сигналів (поведінка яких заздалегідь найчастіше невідома), недостатня повнота контролю.	стотных свойств и др.). Недостатками этой группы являются необходимость априорной информации о характеристиках выходных сигналов, неизбежная зависимость характеристик от входных сигналов (поведение которых заранее чаще всего неизвестно), недостаточная полнота контроля.
	С. 234–235.	**С. 36.**
	Зазначені недоліки не притаманні групі методів, заснованій на використанні алгебраїчних інваріантів. У цих методах діагностування здійснюється шляхом перевірки деяких алгебраїчних співвідношень (контрольних умов), яким має задовольняти сукупність вихідних сигналів об'єкта, доповнена при необхідності надлишковими сигналами. Інваріантність контрольних умов полягає в тому, що при відсутності дефектів вони зобов'язані виконуватися для будь-яких вхідних сигналів і в будь-який момент часу. Загальна схема функціонального діагностування на основі алгебраїчних інваріантів показана на рисунку 4.11.	От указанных недостатков свободна группа методов, основанная на использовании алгебраических инвариантов. В этих методах диагностирование осуществляется путем проверки некоторых алгебраических соотношений (контрольных условий), которым должна удовлетворять совокупность выходных сигналов объекта, дополненная при необходимости избыточными сигналами. Инвариантность контрольных условий состоит в том, что при отсутствии дефектов они обязаны выполняться для любых входных сигналов и в любой момент времени. Общая схема функционального диагностирования на основе алгебраических инвариантов показана на рисунке 1.7.
	С. 235.	**С. 36.**
	 Рис. 4.11. Функціональне діагностування на основі алгебраїчних інваріантів	 Рис. 1.7. Функциональное диагностирование на основе алгебраических инвариантов
	С. 235–236.	**С. 36–37.**
	На пристрій діагностування надходять вхідні U і вихідні Y-сигнали об'єкта, що перевіряється O. На їхній основі блок H виробляє допоміжні (надлишкові) сигнали Z, що задовольняють алгебраїчному рівнянню $\Delta=F(Y,Z)=0$, яке інваріантне щодо вектора вхідних сигналів. Якщо в результаті збою вектор Y буде спотворений, це призведе до появи ненульового сигналу Δ. На практиці замість порівняння з нулем аналізується перевищення допуску (через похибки) $\|\Delta\|\le\varepsilon$. Найбільш відомими методами цієї групи є дублювання і резервування. При цьому в якості блоку H виступає другий примірник об'єкта (Y на вхід не надходять), діагностування полягає в порівнянні виходів за умовою $\Delta=Y-Z=0$. Для зменшення апаратурної надмірності можна використовувати не дублікат об'єкта, а його спрощену або апроксимувальну модель, зменшувати число виходів, перевіряти правильність функціонування в окремих режимах. Сюди ж належить побудова за структурними або функціональними схемами логічних моделей, що відбивають причинно-наслідкові зв'язки між блоками. Тут застосовується і кінцевоавтоматна апроксимація. Іншим напрямком є побудова моделей, що використовують в якості діагностичних ознак внутрішні, безпосередньо не спостерігаються координати об'єкта. Іноді вихідні сигнали можуть задовольняти відомому алгебраїчному співвідношенню виду $F(Y)=0$, при цьому пристрій діагностування буде містити тільки блок F. Такі об'єкти мають природну надмірність. Існують критерії аналізу об'єкта діагностування з виявленням в них	На устройство диагностирования поступают входные U и выходные Y сигналы проверяемого объекта О. На их основе блок Н вырабатывает вспомогательные (избыточные) сигналы Z, удовлетворяющие алгебраическому уравнению Д = F(Y\ Z) = 0, которое инвариантно по отношению к вектору входных сигналов. Если в результате сбоя вектор Y будет искажен, это приведет к появлению ненулевого сигнала Д. На практике вместо сравнения с нулем, анализируется превышение допуска (из-за погрешностей) |Д| < е. Наиболее известными методами этой группы являются дублирование и резервирование. При этом в качестве блока Я выступает второй экземпляр объекта (Y на вход не поступают), диагностирование состоит в сравнении выходов по условию Д = Y — Z = 0. Для уменьшения аппаратурной избыточности можно использовать не дубликат объекта, а его упрощенную или аппроксимирующую модель, уменьшать число выходов, проверять правильность функционирования в отдельных режимах. Сюда же относится построение по структурным или функциональным схемам логических моделей, отражающих причинно-следственные связи между блоками. Здесь же применяется конечноавтоматная аппроксимация. Другим направлением является построение моделей, использующих в качестве диагностических признаков внутренние, непосредственно не наблюдаемые координаты объекта. Иногда выходные сигналы могут удовлетворять известному алгебраическому соотношению вида

	природної надмірності, яка дозволяє будувати прості пристрої діагностування за рахунок індивідуальних особливостей об'єкта. У методі надлишкових змінних вводиться штучна надмірність так, щоб розширена сукупність змінних задовольняла контрольній умові.	F(Y) = 0, при этом устройство диагностирования будет содержать только блок F. Такие объекты обладают естественной избыточностью. Существуют критерии анализа объекта диагностирования по обнаружению в них естественной избыточности, которая позволяет строить простые устройства диагностирования за счет индивидуальных особенностей объекта. В методе избыточных переменных вводится искусственная избыточность так, чтобы расширенная совокупность переменных удовлетворяла контрольному условию.
	С. 236.	**С. 37–38.**
	Одним із способів забезпечення надійної роботи пристрою є проведення тестування, тобто перевірки пристрою шляхом подачі на вхід спеціальних (тестових) впливів і аналіз реакцій на виході. Значна частина фізичних дефектів проявляється у вигляді одиночних константних несправностей. Існують ефективні методи тестування цього класу. Тестування більш складних класів стикається зі значними труднощами.	В [46] рассматривается автоматизация тестового диагностирования цифровых устройств. Одним из способов обеспечения надежной работы устройства является проведение тестирования, т.е. проверки устройства путем подачи на вход специальных (тестовых) воздействий и анализ реакций на выходе. Значительная часть физических дефектов проявляется в виде одиночных константных неисправностей. Существуют эффективные методы тестирования этого класса. Тестирование более сложных классов сталкивается со значительными трудностями.
2	**Коваленко Ю. Б.** **Інформаційна підтримка процесу проектування та експлуатації комплексу бортового обладнання інтегрованої модульної авіоніки.** – Дис. … доктора технічних наук. – Київ, 2025. (https://duikt.edu.ua/uploads/p_86_93515477.pdf)	**Хакимов Д. В.** **Автоматизация проектирования структуры функций комплексов бортового оборудования, построенных на принципах интегральной модульной авионики.** – Дисс. … канд. техн. наук. – Ульяновск, 2017.
	С. 47–48.	**С. 149.**
	Зауважимо, що основним методом підвищення відмовобезпеки і надійності на сьогоднішній день є резервування. Застосування резервування дозволяє ефективно підвищити показники надійності слабких механізмів і вузлів до необхідного рівня. Метод резервування добре вивчений і існує безліч різних способів його реалізації. Основними недоліками методу резервування є: – зниження ефективності при збільшенні кратності резервування; – сильне зростання вартості обладнання зі збільшенням кратності резервування; – сильне зростання вагогабаритних характеристик обладнання зі збільшенням кратності резервування; – основний і резервний канал як правило ідентичні і мають однаковий час наробітку до відмови, термін служби та ін. Це спричиняє високу ймовірність синхронної відмови всіх резервних каналів. **Коваленко додає до чужого тексту «**Зауважимо, що**».** **Нахабний плагіат.**	Основным методом повышения отказобезопасности и надежности на сегодняшний день является резервирование. Применение резервирования позволяет эффективно повысить показатели надежности слабых механизмов и узлов до требуемого уровня. Метод резервирования хорошо изучен и существует множество различных способов его реализации. Основными недостатками метода резервирования являются: ● Снижение эффективности при увеличении кратности резервирования; ● Сильный рост стоимости оборудования с увеличением кратности резервирования; ● Сильный рост массогабаритных характеристик оборудования с увеличением кратности резервирования; ● Основной и резервный канал как правило идентичны и имеют одинаковое время наработки на отказ, срок службы и т. д. Это приводит к высокой вероятности синхронного отказа всех резервных каналов.
	С. 48–49.	**С. 149.**
	Сукупність цих недоліків веде до того, що застосування резервування є небажаним і дорогим методом підвищення надійності. Архітектура ІМА дозволяє значною мірою знизити кількість резервного обладнання в складі КБО. Це властивість ІМА є одна з найбільш значущих і	Совокупность данных недостатков ведет к тому, что применение резервирования является нежелательным и дорогостоящим методом повышения надежности. Архитектура ИМА позволяет в значительной мере снизить количество резервного оборудования

	обумовлена концепцією застосування уніфікованих ОМ в поєднанні з властивістю незалежності ПЗ і АЗ [32-35]. Однак, аналізуючи структурну схему типового крейта ІМА, приходимо до висновку, що резервуванню підлягає весь крейт цілком. При такому підході резервуються не тільки критичні функції комплексу, а й некритичні. Тобто резервування надмірне. **Коваленко переписала чужий текст, додавши фальшиві покликаня.** **Коваленко пише «**Однак, аналізуючи… приходимо до висновку**», нібито це вона робить якийсь аналіз та висновки, а насправді переписує чужий текст.** **Фальсифікація джерел.** **Плагіат.**	в составе КБО. Это свойство ИМА является одним из наиболее значимых и обусловлено концепцией применения унифицированных ВМ в сочетании со свойством независимости ПО и АО. Однако, анализируя структурную схему типового крейта ИМА видно, что резервированию подвергается весь крейт целиком. При таком подходе резервируются не только критические функции комплекса, но и не критические. То есть резервирование избыточно.
	С. 49.	**С. 150.**
	Причиною, яка зумовлює доцільність такого підходу до резервування, є розподіл критичних функцій комплексу між ОМ крейта. У підсумку надмірність ПЗ і АЗ КБО ІМА надвисока. Ґрунтуючись на принципах роботи алгоритму проєктування структури функцій, можна запропонувати метод зниження надмірності ПЗ і АЗ комплексу. РГП функцій є одним з критеріїв проєктування структури функцій. На етапі визначення складу ГФ облік РГП дозволяє максимально сконцентрувати критичні функції комплексу в рамках однієї групи. Тобто з отриманої множини ГФ лише мала частина матиме в своєму складі критичні функції. Тобто при реалізації ПЗ і АЗ крейта вийде, що лише один ОМ і реалізовані на ньому програмні застосунки будуть вимагати резервування. Це означає, що можна мінімізувати комплектність резервного крейта.	Причиной, обуславливающей целесообразность такого подхода к резервированию является распределение критических функций комплекса между ВМ крейта. В итоге избыточность ПО и АО КБО ИМА очень высока. Основываясь на принципах работы алгоритма проектирования структуры функций можно предложить метод снижения избыточности ПО и АО комплекса. УГП функций является одним из критериев проектирования структуры функций. На этапе определения состава ГФ учет УГП позволяет максимально сконцентрировать критические функции комплекса в рамках одной группы. То есть из полученного множества ГФ лишь малая часть будет иметь в своем составе критические функции. На основе рисунка 32 и из таблицы 7 и видно, что лишь одна ГФ включает критические функции комплекса. То есть, при реализации ПО и АО крейта получиться, что лишь один ВМ и реализуемые на нем программные приложения будут требовать резервирования. А значит можно минимизировать комплектность резервного крейта.
	С. 49.	**С. 150.**
	Тепер розглянемо механізм реконфігурації КБО ІМА. Цей механізм є основою і методом скорочення кількості резервного обладнання в складі крейта при значному підвищенні його відмовобезпеки [100, 101, 118, 119]. Реконфігурація заснована на процедурі перерозподілу програмних за стосунків між справними ОМ крейта [102-104]. Підставою для виконання реконфігурації є відмова одного або більше ОМ [105, 106]. Метою реконфігурації при критичних режимах експлуатації ПС, в разі пошкодження великого числа ОМ, є перерозподіл програмних застосунків таким чином, щоб забезпечити працездатність найбільш критичних функції [107, 118]. **Коваленко переписала чужий текст разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела, у тому числі її власні статті (покликання №100, 101, 102, 103, 104, 105, 106 і 107).** **У цьому фрагменті її дисертації всі покликання фальшиві.**	Теперь рассмотрим механизм реконфигурации КБО ИМА. Этот механизм является основой и методом сокращения количества резервного оборудования в составе крейта при значительном повышении его отказобезопасности [100, 101, 118, 119]. Реконфигурация основана на процедуре перераспределения программных приложений между исправными ВМ крейта [102 - 104]. Основанием для выполнения реконфигурации является отказ одного или более ВМ [105, 106]. Целью реконфигурации при критических режимах эксплуатации ВС, в случае повреждения большого числа ВМ является перераспределение программных приложений таким образом, чтобы обеспечить работоспособность наиболее критичных функции [107, 118]. **Покликання [100] – це:** Дегтярев А. Р. Перспективные динамически реконфигурирующиеся комплексы бортового оборудования на основе интегрированной модульной авионики // Научные чтения по авиации, посвященные памяти Н. Е. Жуковского – 2015. – №3. – С. 371–375.

	Помилка перекладу: в російському тексті слово «функции» має помилку (буква «и» замість «й»), і у Коваленко вийшла фраза: «працездатність найбільш критичних функції». **Фальсифікація джерел.** **Плагіат.**	
	С. 49–50.	**С. 150–151.**
	Для ефективної роботи алгоритмів реконфігурації потрібно розбиття функцій КБО на якомога більш дрібні функціональні застосунки. Однак при цьому складність визначення оптимальної конфігурації багаторазово зростає, аж до неприйнятного рівня. Це є перешкодою на шляху реалізації високоефективного реконфігуруючого КБО ІМА. Крім того, сам алгоритм реконфігурації при такому підході вимагає його реалізації як програмного за стосунку комплексу. А значить, до нього пред'являються ті ж вимоги, що і до всіх інших функцій КБО, що робить реалізацію властивості реконфігурації вкрай складним завданням [105-109]. **Коваленко переписала чужий текст разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела – її власні статті.** **Фальсифікація джерел.** **Плагіат.**	Для эффективной работы алгоритмов реконфигурации требуется разбиение функций КБО на как можно более мелкие функциональные приложения. Однако при этом сложность определения оптимальной конфигурации многократно возрастает, вплоть до неприемлемого уровня. Это является препятствием на пути реализации высоко эффективного реконфигурирующегося КБО ИМА. Кроме того, сам алгоритм реконфигурации при таком подходе требует его реализации как программного приложения комплекса. А значит к нему предъявляются те же требования, что и ко всем другим функциям КБО, что делает реализацию свойства реконфигурации крайне сложной задачей [105 - 109].
	С. 50.	**С. 151.**
	Однак, якщо реалізувати кожну ГФ як один програмний додаток і призначити відповідні пріоритети між ними, то можна максимально спростити реалізацію властивості реконфігурації комплексу. При виконанні цих умов сутність реконфігурації полягатиме у виконанні максимально можливої кількості програмних застосунків відповідно до їхнього пріоритету.	Однако, если реализовать каждую ГФ как одно программное приложение и назначить соответствующие приоритеты между ними, то можно максимально упростить реализацию свойства реконфигурации комплекса. При выполнении этих условий сущность реконфигурации будет состоять в выполнении максимально возможного количества программных приложений в соответствии с их приоритетом.
	С. 50.	**С. 151.**
	Аналізуючи масив даних, які формуються при проєктуванні структури функцій, можна зробити висновок про те, що зберігання цих даних можна здійснювати в базі даних (БД) реляційного типу, які широко застосовуються в наш час [110, 117]. **Коваленко переписала чужий текст разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела, у тому числі [110] – її власна стаття.** **Фальсифікація джерел.** **Плагіат.**	**4.3 Возможные варианты реализации САПР по проектированию структуры функций** Анализируя массив данных формируемых при проектировании структуры функций можно сделать вывод о том, что хранение этих данных можно осуществлять в базе данных (БД) реляционного типа, которые широко применяются в наше время [110, 117].
	С. 50.	**С. 151.**
	Для роботи з реляційними БД існує безліч систем управління БД (СУБД): IMS, DB2, Informix, Oracle, Database, Microsoft SQL Server, Adaptive Server Enterprise, Teradata Database, Firebird, Postgre SQL, MySQL, SQLite, Microsoft Access, Visual FoxPro, Линтер, CouchDB, MongoDB, Cache та ін. Вибір конкретної СУБД визначається можливостями і	Для работы с реляционными БД существует множество систем управления БД (СУБД): IMS, DB2, Informix, Oracle, Database, Microsoft SQL Server, Adaptive Server Enterprise, Teradata Database, Firebird, Postgre SQL, MySQL, SQLite, Microsoft Access, Visual FoxPro, Линтер, CouchDB, MongoDB, Cache и т. д. Выбор конкретной СУБД определяет-

	потребами підприємства. Всі СУБД відрізняються один від одної набором функцій для обслуговування та управління БД. Тому доцільно визначити низку вимог, яким мусить відповідати СУБД [20, 27, 111, 112]. **Коваленко переписала чужий текст разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела, у тому числі [111, 112] – її власні статті.** **Фальсифікація джерел.** **Плагіат.**	ся возможностями и потребностями предприятия. Все СУБД отличаются друг от друга предоставляемым набором функций для обслуживания и управления БД. Поэтому целесообразно определить ряд требований, которым должна отвечать СУБД [20, 27, 111, 112].
	С. 50–51.	**С. 151–152.**
	Для формування масиву вихідних даних і побудови графа первинної структури функцій потрібна участь чималої кількості фахівців. Це фахівці різних рівнів проєктування – ПС, комплекс, система, модуль. Фахівці вузького і широкого профілів, які займаються розробкою програмних застосунків, функціональних вузлів, друкованих плат тощо. Кількість фахівців, залучених до процесу проєктування структури функцій, визначається складністю КБО і глибиною розкладання комплексних функцій на підфункції. При проєктуванні складних КБО ІМА може виникнути необхідність залучення і сторонніх фахівців. Крім цього, очевидно, що необхідно забезпечити доступ до БД як замовника, так і виконавця ДКР. А це вже є питанням міжміської взаємодії. Тоді вже відіграє важливу роль питання збереження таємниці розробки і підприємств. Таким чином, СУБД має забезпечуватися можливість мережевого доступу до БД хоча б у рамках мережі підприємства і мати вбудовані інструменти для управління політикою доступу до даних. **Коваленко пише «Таким чином», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Плагіат.**	Для формирования массива исходных данных и построения графа первичной структуры функций требуется участие достаточно большого количества специалистов. Это специалисты различных уровней проектирования – ВС, комплекс, система, модуль. Специалисты узкого и широкого профилей, которые занимаются разработкой программных приложений, функциональных узлов, печатных плат и т. д. Количество специалистов, вовлеченных в процесс проектирования структуры функций, определяется сложностью КБО и глубиной разложения комплексных функций на подфункции. При проектировании сложных КБО ИМА может возникнуть необходимость привлечения и сторонних специалистов. Кроме этого очевидно, что необходимо обеспечить доступ к БД как заказчика, так и исполнителя ОКР. А это может является вопросом междугороднего взаимодействия. Тогда начинает играть важную роль вопрос секретности разработки и предприятий. Таким образом СУБД должна обеспечивать возможность сетевого доступа к БД хотя бы в рамках сети предприятия и иметь встроенные инструменты для управления политикой доступа к данным.
	С. 51.	**С. 152.**
	Другим важливим аспектом є питання надійності зберігання даних. Інформація про структуру функцій проєктованого КБО дуже цінна і об'ємна. Крім того, масиви вихідних даних про деякі функції і фрагменти графа первинної структури функцій є з великою ймовірністю типовими. У зв'язку з чим згодом на підприємстві буде сформована бібліотека даних про функції комплексу. Втрата даних є важко відновлюваною і вимагає великих трудовитрат. Тому вкрай важливо, щоб СУБД містила стандартні інструменти підвищення надійності зберігання даних, перевірки коректності внесених змін та ін.	Вторым важным аспектом является вопрос надежности хранения данных. Информация о структуре функций проектируемого КБО весьма ценна и объемна. Кроме того, массивы исходных данных о некоторых функциях и фрагменты графа первичной структуры функций являются с большой вероятностью типовыми. В виду чего со временем на предприятии будет сформирована библиотека данных о функциях комплекса. Потеря данных является тяжело восстанавливаемой и требует больших трудозатрат. Поэтому крайне важно, чтобы СУБД содержала стандартные инструменты повышения надежности хранения данных, проверки корректности вносимых изменений и т. д.
	С. 51.	**С. 153.**
	Спеціалізоване ПЗ як правило пишеться для конкретних, вузькоспеціалізованих завдань, з жорстким математичним і алгоритмічним набором функцій. Ступінь складності завдань, що реалізуються таким методом, може бути досить висока. Кількість операторів при роботі з таким ПЗ як правило не перевищує однієї людини [113].	Специализированное ПО как правило пишется для конкретных, узкоспециализированных задач, с жестким математическим и алгоритмическим набором функций. Степень сложности задач, реализуемых таким методом, может быть достаточно высока. Количество операторов при работе с таким ПО как правило не превышает одного человека.

	Коваленко до переписаного чужого тексту за 2017-й рік додала покликання [113] на свою власну статтю за 2024-й рік. **Фальсифікація джерел.** **Плагіат.**	
	С. 51–52.	**С. 55–56.**
	На сьогоднішній день САПР є найефективнішим інструментом для вирішення трудомістких завдань проєктування. Найбільш значущі ефекти від застосування САПР наступні: – Скорочення трудомісткості проєктування. – Скорочення циклу проєктування. – Скорочення собівартості проєктування. – Покращення якості проєктування. – Скорочення витрат на натурне моделювання та випробування. – Скорочення трудомісткості розробки. – Скорочення трудомісткості адаптації до умов експлуатації. – Скорочення трудомісткості супроводу.	**1.2 Анализ применяемых САПР при проектировании КБО** На сегодняшний день САПР является самым эффективным инструментом для решения трудоемких задач проектирования. Наиболее значимые эффекты от применения САПР следующие: – Сокращение трудоемкости проектирования; – Сокращение цикла проектирования; – Сокращение себестоимости проектирования; – Улучшение качества проектирования; – Сокращение затрат на натурное моделирование и испытания; – Сокращение трудоемкости разработки; – Сокращение трудоемкости адаптации к условиям эксплуатации; – Сокращение трудоемкости сопровождения.
	С. 52.	**С. 56.**
	Високу ефективність САПР отримали в процесі інтенсивного розвитку упродовж останніх двох десятиліть. Перші САПР з'явилися на початку 60-х років XX століття і були набором частково пов'язаних між собою прикладних програм вузької спеціалізації. Тоді були розроблені програми для вирішення завдань будівельної механіки, аналізу електронних схем і проєктування друкованих плат. Висока ефективність і величезні перспективи розвитку САПР залучили велику кількість компаній-розробників в дану сферу. Незабаром вигляд САПР змінився докорінно. Вже до кінця XX століття системи САПР зазнали якісних змін і мали вигляд мобільних і чітко організованих систем, здатних до оптимізації, в залежності від функціоналу. Сьогодні САПР є складними математичними об'єктами, заснованими на поєднанні безлічі прикладних наук. Розробка математичного апарату САПР вимагає використання величезного розмаїття методів обчислювальної математики, статистики, дискретної математики, математичного аналізу, методів математичного програмування, імовірнісних розрахунків, застосування складних алгоритмів та ін.	Высокую эффективность САПР приобрели в процессе интенсивного развития в течении последних двух десятилетий. Первые САПР появились в начале 60-х годов 20-го века и представляли собой набор частично связанных между собой прикладных программ узкой специализации. Тогда были разработаны программы для решения задач строительной механики, анализа электронных схем и проектирования печатных плат. Высокая эффективность и огромные перспективы развития САПР привлекли большое количество компаний разработчиков в данную сферу. В скором времени облик САПР изменился коренным образом. Уже к концу 20 века системы САПР претерпели качественные изменения и стали представлять собой мобильные и строго организованные системы, способные к оптимизации под задачи предметной области. Сегодня САПР являются сложными математическими объектами, основанными на сочетании множества прикладных наук. Разработка математического аппарата САПР требует использования огромного разнообразия методов вычислительной математики, статистики, дискретной математики, математического анализа, методов математического программирования, вероятностных расчетов, применения сложных алгоритмов и т. д.
	С. 52.	**С. 56–57.**
	Сучасні САПР можна класифікувати за безліччю ознак. Класифікація відповідно до ISO/IEC 19501 UML: – типом об'єкта проєктування; – складністю об'єкта проєктування; – рівнем автоматизації проєктування; – комплексністю автоматизації проєктування; – характером документів, що випускаються; – кількістю випущених документів за рік; – кількістю рівнів технічного забезпечення.	Современные САПР можно классифицировать по множеству признаков. В РФ САПР классифицируются в соответствии с ГОСТ 23501.108-85 по [33]: ● типу объекта проектирования; ● сложности объекта проектирования; ● уровню автоматизации проектирования; ● комплексности автоматизации проектирования; ● характеру выпускаемых документов; ● количеству выпускаемых документов за год; ● количеству уровней технического обеспечения.

	Коваленко в переписаному чужому тексті видалила покликання. Плагіат.	
	С. 52–53.	**С. 57.**
	Відповідно до цієї класифікації завдання проєктування структури КБО є комплексною і за ознакою типу об'єкта проєктування належить до проєктування виробів приладобудування. За ознакою складності об'єкта проєктування, в залежності від конкретного виробу, може належати як до об'єктів середньої складності, так і до об'єктів високої складності. При проєктуванні КБО застосовується безліч САПР, різних за своїм цільовим призначенням: – Засоби проєктування CAD (Computer Aided Design). – Засоби інженерного аналізу CAE (Computer Aided Engineering). – Засоби управління документообігом PDM (Product Document Management). – Геоінформаційні системи GIS (Geo informatics Systems).	В соответствии с данной классификацией задача проектирования структуры КБО является комплексной и по признаку типа объекта проектирования, относится к проектированию изделий приборостроения. По признаку сложности объекта проектирования, в зависимости от конкретного изделия может относится как к объектам средней сложности, так и к объектам высокой сложности. При проектировании КБО применяется множество САПР, различных по своему целевому назначению: ● Средства проектирования CAD (Computer Aided Design); ● Средства инженерного анализа CAE (Computer Aided Engineering); ● Средства управления документооборотом PDM (Product Document Management); ● Геоинформационные системы GIS (Geo informatics Systems).
	С. 53.	**С. 57–58.**
	У свою чергу, серед усіх CAD-рішень за галузевим призначенням вони поділяються на: – Машинобудівні CAD – MCAD (Mechanical Computer Aided Design). – САПР електронних пристроїв, EDA (Electronic Design Automation). А за системами CAE наступні: – Засоби виконання розрахунків на міцність. – Засоби виконання теплових розрахунків. – Засоби побудови гідроаеродинамічних моделей (CFD, Computational Fluid Dynamics). – Засоби виконання кінематичного аналізу. – Засоби виконання механічної симуляції (MES, Mechanical Event Simulation). – Засоби виконання електромагнітних і електродинамічних розрахунків.	В свою очередь, из всего множества CAD-решений выделим по отраслевому назначению следующие: ● Машиностроительные CAD - MCAD (Mechanical Computer Aided Design); ● САПР электронных устройств, EDA (Electronic Design Automation); А из систем CAE следующие: ● Средства выполнения прочностных расчетов; ● Средства выполнения тепловых расчетов; ● Средства построения гидроаэродинамических моделей (CFD, Computational Fluid Dynamics); ● Средства выполнения кинематического анализа; ● Средства выполнения механической симуляции (MES, Mechanical Event Simulation); ● Средства выполнения электромагнитных и электродинамических расчетов.
	С. 53–54.	**С. 58.**
	Нижче представлені приклади САПР, які отримали на сьогоднішній день найбільш широке поширення: – Для проєктування електричних систем: AutoCAD Electrical, E3.series, CADdy++ – Для тривимірного проєктування конструкцій апаратури БРЕО: Autodesk Inventor, SolidWorks, UniGraphics, КОМПАС, CATIA та ін. – Для інженерних розрахунків і моделювання теплових полів, створюваних апаратурою БРЕО: BETAsoft, Sauna, АСОНІКА-Т та ін. – Для моделювання електромагнітних полів, що створюються апаратурою БРЕО: OrCAD Family Release, GENESYS та ін. – Для проведення спектрального аналізу радіочастотних сигналів, що генеруються і прийнятих апаратурою БРЕО – TESLA та ін. – Для моделювання міцності і резонансних характеристик конструкцій авіаційних виробів: Samcef, Mecano, Boss Quattro та ін. – Для автоматичного трасування друкованих плат з урахуванням тривимірного компонування елементів монтажу: ACCELEDA, P-CAD, Altium Designer та ін.	Ниже представлены примеры САПР получивших на сегодняшний день наиболее широкое распространение: ● Для проектирования электрических систем: AutoCAD Electrical, E3.series, CADdy++ и др.; ● Для трехмерного проектирования конструкций аппаратуры БРЭО: Autodesk Inventor, SolidWorks, UniGraphics, КОМПАС, CATIA и др.; ● Для инженерных расчетов и моделирования тепловых полей, создаваемых аппаратурой БРЭО: BETAsoft, Sauna, Асоника-Т и др.; ● Для моделирования электромагнитных полей, создаваемых аппаратурой БРЭО: OrCAD Family Release, GENESYS и др.; ● Для проведения спектрального анализа радиочастотных сигналов, генерируемых и принимаемых аппаратурой БРЭО – TESLA и др.; ● Для моделирования прочностных и резонансных характеристик конструкций авиационных изделий: Samcef, Mecano, Boss Quattro и др.; ● Для автоматической трассировки печатных плат с учетом трехмерной компоновки элементов монтажа: ACCELEDA, P-CAD, Altium Designer и др.;

	С. 54.	**С. 58–59.**
	– Для конструювання і розводки джгутів в тривимірному просторі корпусів апаратури БРЕО: UG/Wiring, Autodesk Inventor та ін.; – Для моделювання гідродинамічних процесів у системах охолодження авіаційних виробів з урахуванням тепловиділення електронних радіоелементів і фізичних процесів теплопереносу: Fine/Turbo та ін. – Для електронного моделювання радіотехнічних сигналів і ланцюгів: Cadence Design, Mentor Graphics, XILINX Foundation, ALTERA, MicroCap . – Для проєктування програмного забезпечення: C/C++/C#, ADA . – Для аналізу і розрахунку надійності, готовності та ремонтопридатності: АРБІТР, АРМ, АСОНІКА-К, AnyGraph, CRISS, Block Sim, ITEM Software, RAM Commander, Reliability Workbench, Windchill.	● Для конструирования и разводки жгутов в трехмерном пространстве корпусов аппаратуры БРЭО: UG / Wiring, Autodesk Inventor и др.; ● Для моделирования гидродинамических процессов в системах охлаждения авиационных изделий с учетом тепловыделения электронных радиоэлементов и физических процессов теплопереноса: Fine/Turbo др.; ● Для электронного моделирования радиотехнических сигналов и цепей: Cadence Design, Mentor Graphics, XILINX Foundation, ALTERA, MicroCap и др.; ● Для проектирования программного обеспечения: C/C++/C#, ADA и др.; ● Для анализа и расчета надежности, готовности и ремонтопригодности: АРБИТР, АРМ, АСОНИКА-К, AnyGraph, CRISS, Block Sim, ITEM Software, RAM Commander, Reliability Workbench, Windchill и др.
	С. 54–55.	**С. 59.**
	Наведені приклади програмних пакетів САПР за рівнем комплексності є одноетапними. Якщо класифікувати їх за рівнем автоматизації в рамках відповідного етапу проєктування КБО, то їх можна віднести до класу високоавтоматизованих САПР. Якщо розглядати дані САПР у межах всього процесу розробки, то загальний рівень автоматизації буде середнім. Впровадження САПР дозволяє скоротити час процесу проєктування приблизно на 20%. А впровадження принципів уніфікації та стандартизації апаратно-програмних розробок підприємства на засадах принципів інтегрованого середовища САПР дозволяє пришвидшити процес проєктування десь нана 50%.	Приведенные примеры программных пакетов САПР по уровню комплексности являются одноэтапными. Если классифицировать их по уровню автоматизации в рамках соответствующего этапа проектирования КБО, то их можно отнести к классу высокоавтоматизированных САПР. Если рассматривать данные САПР в рамках всего процесса разработки, то общий уровень автоматизации будет средним. Внедрение САПР позволяет сократить время процесса проектирования примерно на 20%. Внедрение же принципов унификации и стандартизации аппаратно-программных разработок предприятия на основе принципов интегрированной среды САПР позволяет сократить время процесса проектирования примерно на 50%.
	С. 55.	**С. 62.**
	Провівши порівняльний аналіз САПР в основних існуючих аналогів сформувався перелік програм, що дозволяє досягти оптимального рівня автоматизації процесу проєктування КБО [35-41]. Перелік програм представлений в табл. 1.3. **Коваленко пише «Провівши порівняльний аналіз», ніби це вона робить якесь порівняння з аналізом, а насправді переписує чужий текст.** **Коваленко переписала чужий текст разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела.** **Фальсифікація джерел.** **Плагіат.**	Проведя сравнительный анализ САПР применяемых в АО «УКБП» и НПП «ЦРТС» и основных существующих аналогов сформировался перечень программ, позволяющих достичь оптимального уровня автоматизации процесса проектирования КБО [35 - 41]. Перечень программ представлен в таблице 1.
	С. 55.	**С. 63.**
	Таблиця 1.3 Перелік САПР для автоматизації процесу проєктування КБО	Таблица 1 – Перечень САПР для автоматизации процесса проектирования КБО

Етап проєктування	Завдання проєктування	Засіб САПР	Ступінь автоматизації
Ескізне проєктування	Розробка архітектури виробу	–	Низька
	Схемотехнічне опрацювання виробу	AutoCAD Electrical	Висока
	Проєктування друкованих плат	Altium Designer	Висока
	Тривимірне проєктування конструкції виробу	Autodesk Inventor	Висока
	Аналіз і розрахунок надійності	RAM Commander	Висока
Технічне проєктування	Доопрацювання архітектури виробу	AutoCAD Electrical	Низька
	Схемотехнічне доопрацювання виробу	AutoCAD Electrical	Висока
	Доопрацювання друкованих плат	Altium Designer	Висока
	Доопрацювання тривимірних моделей конструкції виробу	Autodesk Inventor	Висока
	Аналіз і розрахунок надійності	RAM Commander	Висока

Этап проектирования	Задача проектирования	Средство САПР	Степень автоматизации
Эскизное проектирование	Разработка архитектуры изделия	–	низкая
	Схемотехническая проработка изделия	AutoCAD Electrical	высокая
	Проектирование печатных плат	Altium Designer	высокая
	Трехмерное проектирование конструкции изделия	Autodesk Inventor	высокая
	Анализ и расчет надежности	RAM Commander	высокая
Техническое проектирование	Доработка архитектуры изделия	AutoCAD Electrical	низкая
	Схемотехническая доработка изделия	AutoCAD Electrical	высокая
	Доработка печатных плат	Altium Designer	высокая
	Доработка трехмерных моделей конструкции изделия	Autodesk Inventor	высокая
	Анализ и расчет надежности	RAM Commander	высокая

С. 56.	**С. 62.**
Виходячи з даних, представлених у табл. 1, можна зробити висновок про те, що сучасний рівень розвитку САПР дуже високий [42]. При виборі САПР надано перевагу продукції компанії Autodesk. Це пов'язано з високим ступенем популярності продукції даної компанії на українському ринку, якістю технічної підтримки, пристосованістю до роботи за стандартами України, високим авторитетом виробника. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	Исходя из данных, представленных в таблице 1 можно сделать вывод о том, что современный уровень развития САПР очень высок [42]. При выборе САПР было отдано предпочтение продукции компании Autodesk. Это связано с высокой степенью популярности продукции данной компании на Российском рынке, каче-ством технической поддержки, приспособленностью к работе по стандартам РФ, высоким авторитетом производителя.
С. 56.	**С. 63–64.**
САПР на сьогоднішній день дуже сильно інтегровані в усі типові процеси життєвого циклу виробу та їхня роль вкрай велика. Розробка більшості сучасних виробів була б неможлива без САПР. До таких виробів належить і КБО. Аналіз ступеня автоматизації проєктування КБО показав, що найменш автоматизованим і складним завданням проєктування є розробка структури комплексу. Типовий процес розробки завжди починається з вибору аналогів і прототипу нового виробу [43]. Таким чином, виробник використовує вже готову структуру виробу, лише допрацьовуючи її. Ступінь доопрацювання прототипу вкрай рідко перевищує 20%, що дозволяє в повному обсязі скористатися принципом успадкування найкращих технічних рішень наявних розробок. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	САПР на сегодняшний день очень сильно интегрированы во все типовые процессы жизненного цикла изделия и их роль крайне велика. Разработка большинства современных изделий была бы невозможна без САПР. К таким изделиям относится и КБО. Анализ степени автоматизации проектирования КБО показал, что наименее автоматизированной и сложной задачей проектирования является разработка структуры комплекса. Типовой процесс разработки всегда начинается с выбора аналогов и прототипа нового изделия [43]. Таким образом, производитель использует уже готовую структуру изделия, лишь дорабатывая ее. Степень доработки прототипа крайне редко превышает 20%. Это позволяет производителю в полном объеме использовать принцип наследования наилучших технических решений прошлых разработок.
С. 56.	**С. 64.**
Такий підхід є актуальним, коли йдеться про не-	Такой подход актуален, когда речь идет о не-

	великий пристрій з типовим набором функцій. Мета розробки такого пристрою, як і раніше, це покращення його якісних характеристик. При розгляді КБО як об'єкта проєктування очевидно, що лише частина його функцій є типовими. Кожен новий КБО відмінний від аналогів не лише за якісними показниками, а й функціонально. Розглядаючи архітектуру ІМА, завдання розробки структури комплексу ускладнюється. Виробники випускають КБО ІМА, маючи в своєму розпорядженні величезний досвід розробок лише федеративних комплексів. Водночас певні особливості архітектури змушують виробника кожен КБО ІМА розробляти практично «з нуля».	большом устройстве, выполняющем типовой набор функций. Целью разработки такого устройства вновь, как правило, является повышение его качественных характеристик. При рассмотрении КБО как объекта проектирования, очевидно, что лишь часть его функций являются типовыми. Каждый новый КБО отличен от аналогов не только качественными показателями, но и функционально. Рассматривая архитектуру ИМА задача разработки структуры комплекса становиться еще более сложной. Производители выпускающие КБО ИМА, имеют в своем распоряжении огромный опыт разработок лишь федеративных комплексов. К тому же особенности самой архитектуры приводят к тому, что производитель вынужден каждый КБО ИМА разрабатывать практически «с нуля».
	С. 67.	**С. 72.**
	Типовий процес проєктування КБО регламентується безліччю різних нормативних документів міжнародного, державного та галузевого рівнів. Тому при розробці методики проєктування структури функцій КБО ІМА необхідно врахувати вимоги даних документів. Виконаємо аналіз нормативної документації, що регламентує типовий процес проєктування КБО. **Коваленко пише «Виконаємо аналіз», ніби це вона буде робити якийсь аналіз, а насправді переписує чужий текст.** **Плагіат.**	**2 ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЦЕССУ ПРОЕКТИРОВАНИЯ СТРУКТУРЫ ФУНКЦИЙ КБО ИМА** Типовой процесс проектирования КБО регламентируется множеством различных нормативных документов международного, государственного и отраслевого уровней. Поэтому, при разработке методики проектирования структуры функций КБО ИМА необходимо учесть требования данных документов. Выполним анализ нормативной документации регламентирующей типовой процесс проектирования КБО.
	С. 67.	**С. 72.**
	Відповідно до стандартів, процес проєктування авіоніки поділяється на кілька етапів [53]: – формування вимог до КБО. – Розробка концепції проєктування. – Формування технічного завдання. – Ескізне проєктування виробу. – Технічне проєктування виробу. – Формування робочої документації. – Введення в дію (підготовка і проведення випробувань КБО). – Гарантійний супровід і післягарантійне обслуговування. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	**2.1 Анализ нормативной документации регламентирующей типовой процесс проектирования КБО** В соответствии с ГОСТ 34.601-90 процесс проектирования авионики делится на несколько этапов [53]: ● Формирование требований к КБО; ● Разработка концепции проектирования; ● Формирование технического задания; ● Эскизное проектирование изделия; ● Техническое проектирование изделия; ● Формирование рабочей документации; ● Ввод в действие (подготовка и проведение испытаний КБО); ● Гарантийное сопровождение и послегарантийное обслуживание.
	С. 67.	**С. 72–73.**
	Кожен етап закінчується формуванням відповідної документації з фіксацією результатів виконаної роботи. Перехід від одного етапу до іншого здійснюється за підсумками оцінки виконаної роботи під час завершеного етапу і ухвалення осново-	Каждый этап заканчивается формированием соответствующей документации с фиксацией результатов проделанной работы. Переход от одного этапа к другому осуществляется по итогам оценки проделанной работы в ходе завершенного этапа и

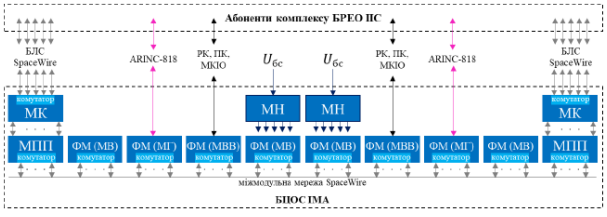
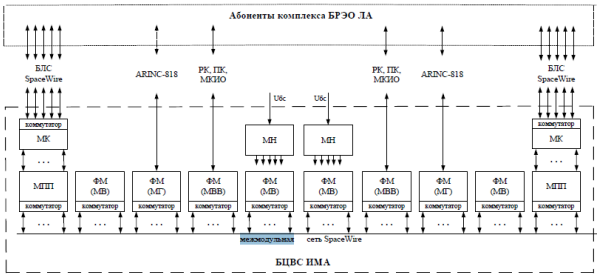
	положних рішень про проведення наступного. Проєктування авіоніки має здійснюватися на основі стандартів і нормативної документації, до яких належать: – Документація EASA і FAA з норм льотної придатності і сертифікації авіаційного обладнання. – Документація FAA, EASA, ARINC, RTCA і SAE з процесів розробки, випробувань, сертифікації та виробництва бортового устаткування. – Національні стандарти України. – Стандарти ANSI/VITA і нормативні документи, що розробляються ОАК для НДР «Конструктор КБО», «ІМА-Конструктор», «ІМА-Інтеграція».	принятия основополагающих решений о проведении следующего. Проектирование авионики для ВСГА должно осуществляться на основе стандартов и руководящей документации, к которым относятся: ● Документация АР МАК, EASA и FAA по нормам летной годности и сертификации авиационного оборудования; ● Документация АР МАК, FAA, EASA, ARINC, RTCA и SAE по процессам разработки, испытаний, сертификации и производства бортового оборудования; ● Национальные стандарты РФ; ● Стандарты ANSI/VITA и разрабатываемые нормативные документы ОАК по НИР «Конструктор КБО», «ИМА-Конструктор», «ИМА-Интеграция».
	С. 68.	**С. 73.**
	– Докладний список даних нормативних документів наведено нижче: – Авіаційні правила. Частина 21. Процедури сертифікації авіаційної техніки . – Авіаційні правила. Частина 25. Норми льотної придатності літаків транспортної категорії (АП-25). – Керівництво з сертифікації систем електронної індикації літаків транспортної категорії. – Технічні вимоги до літаків транспортної категорії, що виконують всепогодні польоти. **Коваленко переписує чужий текст, не розуміючи його. Вступне речення перед переліком перетворюється в неї в один із пунктів цього переліку: «**– Докладний список даних нормативних документів наведено нижче:**».** **Недолугий плагіат.**	Подробный список данных нормативных документов представлен ниже: ● АР МАК Авиационные правила. Часть 21. Процедуры сертификации авиационной техники (АП-21); ● АР МАК Авиационные правила. Часть 25. Нормы летной годности самолетов транспортной категории (АП-25); ● АР МАК Р-25-11А. Руководство по сертификации систем электронной индикации самолетов транспортной категории; ● АР МАК ТТ-ВП. Технические требования к самолетам транспортной категории, выполняющие всепогодные полеты;
	С. 68.	**С. 74.**
	– RTCA DO-248B / EUROCAE ED-94B, Final Report for Clarification of DO- 178B «Software Considerations in Airborne Systems and Equipment Certification». – EASA Certification Memorandum (SWCEH-002). Software Aspects of Certification. – ARINC 653. Avionics Application Software Standard Interface. – IEEE Std 830-1998. Recommended Practice for Software Requirements Specification. – IEEE Std 1233-1998. Guide for Developing System Requirements Specification. – ISO/IEC 15289 Systems and software engineering – Content of systems and software life cycle process information products (Documentation).	● RTCA DO-248B/EUROCAE ED-94B, Final Report for Clarification of DO-178B «Software Considerations in Airborne Systems and Equipment Certification»; ● EASA Certification Memorandum (SWCEH-002). Software Aspects of Certification; ● ARINC 653. Avionics Application Software Standard Interface; ● IEEE Std 830-1998 Recommended Practice for Software Requirements Specification; ● IEEE Std 1233-1998 Guide for Developing System Requirements Specification; ● ISO/IEC 15289 Systems and software engineering – Content of systems and software life cycle process information products (Documentation);
	С. 68.	**С. 74.**
	– ДЕСТ 19.201-78. Технічне завдання, вимоги до змісту та оформлення. – ДЕСТ 19.202-78. Специфікація. Вимоги до змісту та оформлення.	● ГОСТ 19.201-78 Техническое задание, требования к содержанию и оформлению; <…> ● ГОСТ 19.202-78 Спецификация. Требования к содержанию и оформлению;
	С. 68.	**С. 75.**
	– ДЕСТ 2.004-88. Єдина система конструкторської документації. Загальні вимоги до виконання конструкторських і технологічних документів на друкованих і графічних пристроях виведення ЕОМ.	● ГОСТ 2.004-88 Единая система конструкторской документации. Общие требования к выполнению конструкторских и технологических документов на печатающих и графических устройствах вывода

	– ДЕСТ 2.102-68. Єдина система конструкторської документації. Види і комплектність конструкторських документів. – ДЕСТ 2.104-68. Єдина система конструкторської документації. Основні написи.	ЭВМ; ● ГОСТ 2.102-68 Единая система конструкторской документации. Виды и комплектность конструкторских документов; ● ГОСТ 2.104-68 Единая система конструкторской документации. Основные надписи;
	С. 69.	**С. 75.**
	– ДЕСТ 2.105-95. Єдина система конструкторської документації. Загальні вимоги до текстових документів. – ДЕСТ 2.106-96 Єдина система конструкторської документації. Текстові документи. – ДЕСТ 7.32-2001. Система стандартів з інформації, бібліотечної та видавничої справи. Звіт про науково-дослідну роботу. Структура і правила оформлення. – ДЕСТ 2.124-85. Єдина система конструкторської документації. Порядок застосування покупних виробів. – ДЕСТ ВТ 0019-001-2006. Програмне забезпечення вбудованих систем. Вимоги до змісту та оформлення документів.	● ГОСТ 2.105-95 Единая система конструкторской документации. Общие требования к текстовым документам; ● ГОСТ 2.106-96 Единая система конструкторской документации. Тексто-вые документы; ● ГОСТ 7.32-2001 Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления; ● ГОСТ 2.124-85 Единая система конструкторской документации. Поря-док применения покупных изделий; ● ГОСТ ВТ 0019-001-2006 Программное обеспечение встроенных систем. Требования к содержанию и оформлению документов;
	С. 69.	**С. 75.**
	– DOC 9613 AN/937. Керівництво з навігації, заснованої на характеристиках (PBN). – DOC 8168 OPS/611. Здійснення польотів повітряних суден. – DOC 4444 ATM/501. Керівництво з розробки функцій авіоніки. Організація повітряного руху. – DOC 8071. Керівництво з випробувань радіонавігаційних засобів. Том II. Випробування супутникових радіонавігаційних систем (GNSS). – DOC 9863 AN/461. Керівництво з бортової системи попередження зіткнень (БСПЗ).	● DOC 9613 AN/937 Руководство по навигации, основанной на характеристиках (PBN); ● DOC 8168 OPS/611 Производство полетов воздушных судов; ● DOC 4444 ATM/501 Руководство по разработке функций авионики. Организация воздушного движения; ● DOC 8071 Руководство по испытаниям радионавигационный средств. Том II. Испытания спутниковых радионавигационных систем (GNSS); ● DOC 9863 AN/461 Руководство по бортовой системе предупреждения столкновений (БСПС).
	С. 69–70.	**С. 76.**
	– При розробці КБО ІМА додатково набувають чинності такі документи: – ANSI/VITA 46.0-2007. VPX Baseline Standard (Базовий стандарт VPX). – ANSI/VITA 48.0-2010. Mechanical Specification for Microcomputers Using Ruggedized Enhanced Design Implementation (REDI) (Конструкції мікрокомп'ютерів для жорстких умов експлуатації). – VITA 46.9 PMC/XMC/Ethernet Signal Mapping to 3U/6U on VPX User IO (Призначення сигналів вводу-виводу і Ethernet-мезонін PMC/XMC для стандарту VPX з форматом модулів 3U і 6U). – IEEE 1386.1-2001. Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC (Стандарт на фізичний рівень і навколишні умови для мезонінних плат с інтерфейсом PCI). **Коваленко переписує чужий текст, не розуміючи його. Вступне речення перед переліком перетворюється в неї в один із пунктів цього переліку: «**– При розробці КБО ІМА додатково набувають чинності такі документи:**».** **Недолугий плагіат.**	При разработке КБО ИМА дополнительно вступают в силу следующие документы: ● ANSI/VITA 46.0-2007 VPX Baseline Standard (Базовый стандарт VPX); ● ANSI/VITA 48.0-2010 Mechanical Specification for Microcomputers Using Ruggedized Enhanced Design Implementation (REDI) (Конструкции микрокомпьютеров для жестких условий эксплуатации); ● VITA 46.9 PMC/XMC/Ethernet Signal Mapping to 3U/6U on VPX User IO (Назначение сигналов ввода-вывода и Ethernet-мезонин PMC/XMC для стандарта VPX с форматом модулей 3U и 6U); ● IEEE 1386.1-2001 Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC (Стандарт на физический уровень и окружающие условия для мезонинных плат с интерфейсом PCI).
	С. 70.	**С. 76.**
	Представлений перелік нормативної документації дозволяє типізувати процес розробки авіоніки і зробити його повністю контрольованим. З цією метою у процес проєктування впроваджуються необ-	Представленный перечень нормативной документации позволяет типизировать процесс разработки авионики и сделать его полностью контролируемым. С этой целью в процесс проектирования

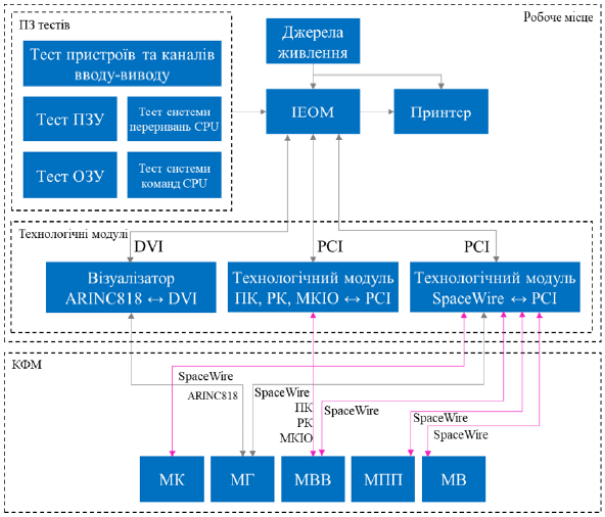
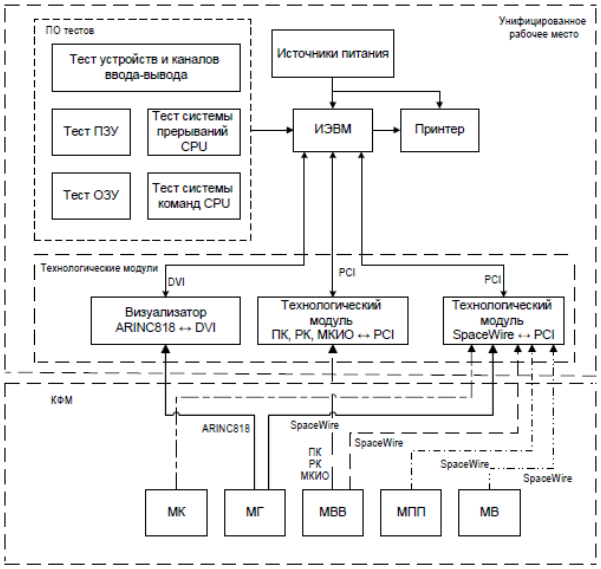
	хідні для контролю якості процеси. Одним з основних таких процесів є оцінка безпеки. Роботи з оцінки безпеки проєктування КБО відбуваються відповідно за такими документами:	внедряются необходимые для контроля качества процессы. Одним из основных таких процессов является – процесс оценки безопасности. Работы по оценке безопасности проектирования КБО производятся в соответствии со следующими документами:
	С. 70.	**С. 76.**
	– Окремо процеси, які є частиною процесу оцінки безпеки, регламентуються документами: **Коваленко переписує чужий текст, не розуміючи його. Вступне речення перед переліком перетворюється в неї в один із пунктів цього переліку.** **Недолугий плагіат.**	Отдельные под процессы, являющиеся частью процесса оценки безопасности, регламентируются документами:
	С. 70.	**С. 77.**
	– ДЕСТ 1 02776-2001. Експлуатація технічна авіаційної техніки за станом. Основні положення.	● ОСТ 1 02776-2001 Эксплуатация техническая авиационной техники по состоянию. Основные положения;
	С. 72.	**С. 77.**
	До сучасних КБО в обов'язковому порядку пред'являється низка вимог з уніфікації виробів і їхньої взаємозамінності, тому в технічному завданні (ТЗ) на комплькекс закладаються вимоги зі стандартизації, уніфікації і каталогізації.	К современным КБО в обязательном порядке предъявляется ряд требований по унификации изделий и их взаимозаменяемости, поэтому в техническом задании (ТЗ) на комплекс закладываются требования по стандартизации, унификации и каталогизации в соответствии с:
	С. 220.	**С. 78.**
	Процес проєктування – це складний ітераційний процес послідовного характеру, який тісно пов'язаний з процесом оцінки безпеки. Весь процес розділений на етапи і рівні. Етапи проєктування зазначені в моделі загальними формулюваннями і не суперечать ДЕСТ 34.601-90 [54]. У моделі поділ на рівні передбачає розмежування процесів проєктування ПС, КБО, систем і модулів один від одного і визначає механізми взаємозв'язків між ними. Між рівнями проєктування існують жорсткі ієрархічні зв'язки. Це забезпечує єдність концепцій проєктування модулів у рамках системи, систем у рамках КБО. Зв'язки між рівнями відображені в документації, яка містить вимоги. Вимоги до будь-якої складової частини виробу формуються поєднанням вимог верхнього рівня і приватних вимог. Виділимо два основних види вимог: – Технічні вимоги. – Вимоги безпеки.	В соответствии с руководствами процесс проектирования представляет собой сложный итерационный процесс последовательного характера, который тесно связан с процессом оценки безопасности. Весь процесс разделен на этапы и уровни. Этапы проектирования указаны в модели общими формулировками и не противоречат ГОСТ 34.601-90 [54]. В модели разделение на уровни предусматривает разграничение процессов проектирования ВС, КБО, систем и модулей друг от друга и определяет механизмы взаимосвязей между ними. Между уровнями проектирования существуют жесткие иерархические связи. Это обеспечивает единство концепций проектирования модулей в рамках системы, систем в рамках КБО и т. д. [55]. Связи между уровнями отражены в документации, содержащей требования. Требования к любой составной части изделия формируются сочетанием требований верхнего уровня и частных требований. Выделим два основных вида требований: ● Технические требования; ● Требования безопасности.
	С. 220–221.	**С. 79.**
	Технічні вимоги визначають основні технічні характеристики, а саме: функції, конструктивне виконання, дизайн виробу, міцність та ін. Вимоги безпеки закріплюють певні характеристики надійності виробу, формують перелік заходів і процедур для забезпечення і гарантії безпеки проєктування і подальшої експлуатації. Вимоги безпеки на перший погляд мають менш важливий пріоритет, ніж вимоги технічні. Такий стан справ у більшості галузей промисловості. Через високу ймовірність загибелі людей у разі відмови обладнання ПС в авіаційній промисловості вимоги безпеки мають надвисокий пріоритет.	Технические требования определяют основные технические характеристики, такие как функции, конструктивное исполнение, дизайн изделия, прочность и т. д. Требования безопасности закрепляют определенные характеристики надежности изделия, формируют перечень мер и процедур для обеспечения и гарантии безопасности проектирования и последующей эксплуатации. Требования безопасности на первый взгляд имеют менее важный приоритет, чем требования технические. Такое положение дел обстоит в большинстве отраслей промышленности. В силу высокой вероятности гибели людей в слу-

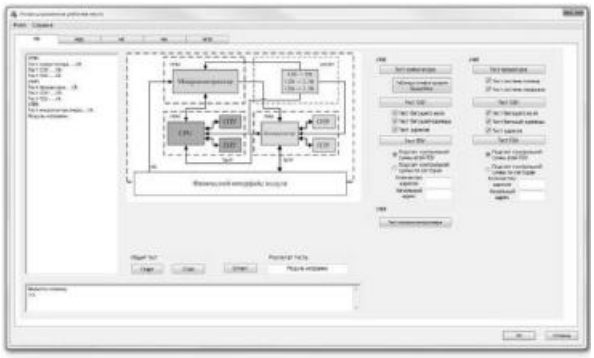
	Представлена модель, рисунок 4.7 – є поєднаною моделлю процесів проєктування і оцінки безпеки. З неї можна бачити, як тісно сучасний процес проєктування КБО пов'язаний з процесом оцінки його безпеки.	чае отказа оборудования ВС в авиационной промышленности требования безопасности имеют крайне высокий приоритет. Модель, представленная в Р-4754 и Р-4761 является совмещенной моделью процессов проектирования и оценки безопасности. Из нее видно, сколь тесно современный процесс проектирования КБО связан с процессом оценки его безопасности.
	С. 221.	**С. 79.**
	Численні оцінки безпеки на кожному з рівнів проєктування виконують роль головного критерію переходу між ними. Причому критерієм обмежений не лише процес завершення етапу проєктування, а і його початок. Настільки висока контролююча роль процесу оцінки безпеки є результатом реалізації концепції наскрізного проєктування авіоніки [65, 66]. **Коваленко переписала чужий текст, замінивши покликання [55, 56] на [65, 66], під якими в її дисертації знаходиться зовсім інші джерела – її власні статті.** **Фальсифікація джерел.** **Плагіат.**	<…> Множественные анализы оценки безопасности на каждом из уровней проектирования выполняют роль главного критерия перехода между ними. Причем критерием ограничен не только процесс завершения этапа проектирования, но и его начало. Столь высокая контролирующая роль процесса оценки безопасности является результатом реализации концепции сквозного проектирования авионики [55, 56].
3	**Коваленко Ю. Б.** **Інформаційна підтримка процесу проектування та експлуатації комплексу бортового обладнання інтегрованої модульної авіоніки.** – Дис. … доктора технічних наук. – Київ, 2025. (https://duikt.edu.ua/uploads/p_86_93515477.pdf)	**Книга Е. В.** **Алгоритмы контроля вычислительной системы интегрированной модульной авионики и ее функциональных элементов.** – Дисс. … канд. техн. наук. – Санкт-Петербург, 2015.
	С. 56–57.	**С. 38.**
	Для створення реконфігурованих багатопроцесорних бортових обчислювальних систем з характеристиками, які будуть відповідати вимогам перспективних комплексів бортового обладнання, розробляються бортові системи, структура яких заснована на використанні уніфікованих функціональних модулів різного призначення, причому їх тип і кількість не повинні впливати на принципи організації обчислювальної системи і принципи її функціонування. При створенні засобів і методів контролю КБО класу ІМА необхідно використовувати принципи уніфікації і стандартизації, засновані на внутрішніх функціональних вузлах компонентів КБО. Роботи з проектування функціональних модулів для КБО перспективних ПС в даний час ведуться за двома різними підходами до апаратної реалізації, які визначають внутрішню структуру функціональних модулів класу ІМА.	Для создания реконфигурируемых многопроцессорных бортовых вычислительных систем с характеристиками, которые будут соответствовать требованиям перспективных комплексов бортового оборудования, разрабатываются бортовые системы, структура которых основана на использовании унифицированных функциональных модулей различного назначения, причем их тип и количество не должны влиять на принципы организации вычислительной системы и принципы ее функционирования. При создании средств и методов контроля БЦВС класса ИМА необходимо использовать принципы унификации и стандартизации, основанные на внутренних функциональных узлах компонентов БЦВС. Работы по проектированию функциональных модулей для БЦВС перспективных ЛА в настоящее время ведутся по двум разным подходам к аппаратной реализации, которые определяют внутреннюю структуру функциональных модулей класса ИМА.
	С. 57.	**С. 39.**
	Обчислювальні модулі мають відкриту масштабовану архітектуру, що дозволяэ програмно реконфігурувати обчислювальну систему. Основу модулів становить елементи (в залежності від модуля кількість варіюється від 16 до 64 шт.), які з'єднані швидкодіючими каналами зв'язку. При розташовані в вузлах плоскої сітки і з'єднані «прямокутної» системою зв'язків. При цьому з'єднуються тільки сусідні елементи ПЛІС, а дані між несуміжними елементами ПЛІС передаються «транзитом» через проміжні елементи. На основі базового модуля може бути побудо-	Вычислительные модули имеют открытую масштабируемую архитектуру, позволяющую программно реконфигурировать вычислительную систему. Основу модулей составляет элементы ПЛИС (в зависимости от модуля количество варьируется от 16 до 64 шт.), которые соединены быстродействующими каналами связи (LVDS, Ethernet и др.). ЭРИ расположены в узлах плоской сетки и соединены «прямоугольной» системой связей. При этом соединяются только соседние элементы ПЛИС, а данные между несмежными элементами ПЛИС передаются «транзитом» через промежуточ-

	вана обчислювальна система, що містить кілька базових модулів та запрограмована для вирішення функціональної задачі авіоніки.	ные элементы. На основе базового модуля 16V5-75 может быть построена вычислительная система, содержащая несколько базовых модулей и запрограммированная для решения функциональной задачи авионики.
	С. 57–58.	**С. 39.**
	Запропонована розробниками в модулі апаратна реалізація має очевидні переваги у вигляді високої швидкодії, малих габаритних розмірів базових модулів, легко реалізовану програмно-керовану реконфігурацію. Разом з цим підхід на основі використання тільки апаратного обладнання має суттєві недоліки у вигляді великого енергоспоживання, що відіб'ється на значне збільшення габаритних розмірів блоку в зв'язку з необхідністю введення складної системи охолодження, та у вигляді істотної залежності проектного рішення від використовуваної імпортної елементної бази, так як вибір вітчизняних аналогів обмежений. Крім того, потрібна розробка складових проекту: нового програмного забезпечення, операційної системи, компіляторів і комплексів ПЗ для програмування обчислювальної системи.	Предложенная разработчиками в модуле аппаратная реализация имеет очевидные преимущества в виде высокого быстродействия, малых габаритных разметов базовых модулей, легко реализуемую программно-управляемую реконфигурацию. Вместе с этим подход на основе использования только ПЛИС имеет существенные недостатки в виде большого энергопотребления, что отразится на значительном увеличении габаритных размеров блока в связи с необходимостью введения сложной системы охлаждения, и в виде существенной зависимости проектного решения от используемой импортной элементной базы, так как выбор отечественных аналогов ограничен. Кроме того, требуется разработка составляющих проекта: нового программного обеспечения, операционной системы, компиляторов и комплексов ПО для программирования вычислительной системы.
	С. 58.	**С. 43.**
	Відповідно до другого підходу елементом, який виконує основні функціональні завдання, покладені на функціональний модуль, є процесор. Додатково до нього при необхідності можуть бути встановлені співпроцесори і / або мікросхеми ПЛІС.	Согласно второму подходу элементом, выполняющим основные функциональные задачи, возлагаемые на функциональный модуль, является процессор. Дополнительно к нему при необходимости могут быть установлены сопроцессоры и/или микросхемы ПЛИС.
	С. 58.	**С. 47.**
	З огляду на той факт, що в даний час основна увага приділяється на реалізації заміни імпорту в вітчизняних виробах (в тому числі модулів бортових обчислювальних систем ІМА), даний підхід є більш перспективним, ніж реалізація модулів ІМА виключно на елементах.	Из анализа рис. 1.8 и табл. 1.4 видно, что технические характеристики модулей, в основе которых заложен процессорный элемент, не уступают техническим характеристикам модулей, в основе которых лежат элементы ПЛИС, а по некоторым параметрам превышает (потребляемая мощность, габаритные размеры, масса и др.). Учитывая тот факт, что в настоящее время большой упор стоит на реализации замены импорта в отечественных изделиях (в том числе модулей бортовых вычислительных систем ИМА), данный подход является более перспективным, чем реализация модулей ИМА исключительно на элементах ПЛИС.
	С. 58.	**С. 47.**
	Обчислювальні системи, що використовуються на ПС, виконують безліч складних функцій в польоті. Таким чином, їх працездатність потрібно відстежувати в кожен момент часу. Використовувані засоби для контролю та діагностики для систем попередніх поколінь не можуть бути використані для контролю та діагностики перспективних обчислювальних систем. Таким чином, необхідно розвивати напрямки дослідження засобів контролю та діагностики бортових цифрових обчислювальних систем класу ІМА та розробляти нові алгоритми для цих цілей.	Вычислительные системы, используемые на ЛА, выполняют множество сложных функций в полете. Таким образом, их работоспособность нужно отслеживать в каждый момент времени. Используемые средства для контроля и диагностики для систем предыдущих поколений не могут быть использованы для контроля и диагностики перспективных вычислительных систем. Таким образом, необходимо развивать направления исследования средств контроля и диагностики бортовых цифровых вычислительных систем класса ИМА и разрабатывать новые алгоритмы для этих целей.
	С. 183.	**С. 96.**
	Отже, сформовані різні підходи до побудови засобів тестування. Тести апаратури бортового обладнання 4-го покоління не підходять для контролю	**3.6 Выводы** 1. К настоящему времени сформировались различные подходы к построению средств тестирования. Тесты аппаратуры бортового оборудования 4-

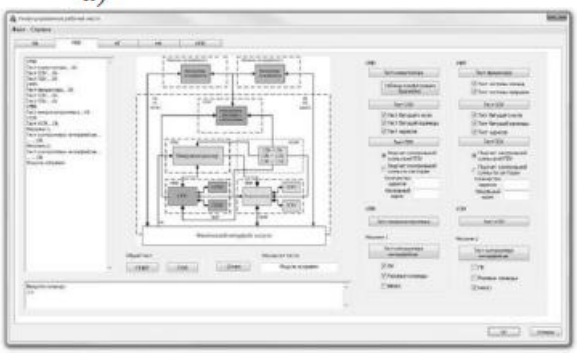
	обчислювальних машин 5-го покоління у зв'язку з істотною відмінністю в організації архітектури обчислювальних систем і збільшення вимог по надійності і відмовостійкості. Тестування апаратури бортового обладнання 5-го покоління слід організовувати з урахуванням нових підходів до архітектури обчислювальної системи. Конструктивно-функціональні модулі, об'єднані в єдину локальну мережу, слід тестувати шляхом зовнішнього функціонального контролю з введенням процедури мажорування. Причому, проведення тесту ініціюється стороннім модулем, що входять до складу обчислювальної системи. Модульна побудови тестів конструктивно-функціональних модулів дозволяє скласти набір тестів, з яких можна скласти функціональний тест для будь-якого з типів обчислювальних модулів.	ого поколения не подходят для контроля вычислительных систем 5-ого поколения в связи с существенным отличием в организации архитектуры вычислительных систем и увеличении требований по надежности и отказоустойчивости. 2. Тестирование аппаратуры бортового оборудования 5-ого поколения следует организовывать с учетом новых подходов к архитектуре вычислительной системы. Конструктивно-функциональные модули, объединенные в единую локальную сеть, следует тестировать путем внешнего функционального контроля с введением процедуры мажорирования. Причем, проведение теста инициируется сторонним модулем, входящим в состав вычислительной системы. <…> 5. Модульное построения тестов конструктивно-функциональных модулей позволяет составить набор тестов, из которых можно составить функциональный тест для любого из типов вычислительных модулей.
	С. 255.	**С. 97.**
	5.1. Оцінка надійності обчислювальних структур інтегрованої модульної авіоніки. Суттєвими для оцінки надійності обчислювачів класу IMA є внутрішня структура обчислювача [62-64], в якості якої розглядається схема зв'язку ФМ у виробі, і λ-характеристики надійності елементної бази ФМ, що входить у виріб. При цьому оцінка надійності виробу може бути виконана шляхом аналітичного виведення виразу для ймовірності P(t) безвідмовної роботи виробу і побудови сімейства графіків залежності цієї ймовірності на заданому часовому інтервалі. Часовий інтервал характеризує очікуваний час безперервної роботи виробу в експлуатації. На основі універсальної функціональної схеми обчислювача класу IMA на практиці розробляють різні варіанти внутрішніх структур виробів авіоніки, що реалізуються на практичному рівні, рисунок 5.1. **Коваленко переписала текст із російської дисертації 2015-го року.** **Фальшиві покликання [62–64] – це власні праці Коваленко 2017–2018 рр.** **Фальсифікація джерел.** **Плагіат.**	**4.1 Оценка надежности вычислительных структур** Существенными для оценки надежности вычислителей класса ИМА являются внутренняя структура вычислителя [8-14, 71], в качестве которой рассматривается схема связи ФМ в изделии, и λ-характеристики надежности элементной базы ФМ, входящей в изделие. При этом оценка надежности изделия может быть выполнена путем аналитического вывода выражения для вероятности ()tP безотказной работы изделия и построения семейства графиков зависимости этой вероятности на заданном временном интервале. Временной интервал характеризует ожидаемое время непрерывной работы изделия в эксплуатации. На основе универсальной функциональной схемы вычислителя класса ИМА (см. рис. 2.1) на практике разрабатываются различные варианты внутренних структур изделий авионики, реализуемых на логическом уровне.
	С. 255.	**С. 52.**
	 Рис 5.1 Структурно-функціональна схема міжмодульної взаємодії комплексу бортового радіоелектронного обладнання (БРЕО) на основі мережі SpaceWire в архітектурі IMA	 Рисунок 2.1 - Функциональная схема БЦВС (Uбс – напряжение бортовой электрической сети питания).
	С. 255.	**С. 97.**
	Необхідно відзначити, що фізичне середовище	<…> Необходимо отметить, что физическая среда

	поширення інформації в виробі відповідає схемі зв'язку, а логічне середовище поширення інформації організовується програмними засобами. Таким чином, не кожна фізична лінія передачі інформації виявляється задіяною в логічному протоколі обміну в ОС. **Коваленко пише «Таким чином», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Плагіат.**	распространения информации в изделии соответствует схеме связи рис. 1.7, а логическая среда распространения информации организуется программными средствами. Таким образом, не каждая физическая линия передачи информации оказывается задействованной в логическом протоколе обмена в ВС.
	С. 286.	**С. 85–86.**
	5.3. Уніфіковане автоматизоване робоче місце з перевірки конструктивно-фунціонального модуля ІМА КБО В процесі виробництва КБО виробники стикаються з проблемою контролю якості продукції, що випускається. Традиційно контроль якості продукції на авіаприборобудівльному підприємстві здійснюється за допомогою перевірки кожного виготовленого виробу в складі автоматизованого робочого місця [102]. **Коваленко переписала текст із російської дисертації 2015-го року.** **Фальшиве покликання [102] – це власна праця Коваленко 2021 р.** **Фальсифікація джерел.** **Плагіат.**	**3.4.2 Унифицированное автоматизированное рабочее место по проверке КФМ** В процессе производства КФМ изготовители сталкиваются с проблемой контроля качества выпускаемой продукции. Традиционно контроль качества продукции на авиаприборостроительном предприятии осуществляется посредством проверки каждого изготовленного изделия в составе автоматизированного рабочего места.
	С. 286.	**С. 86.**
	АРМ з перевірки КБО має забезпечувати: – імітацію процесів інформаційного обміну по інтерфейсів SpaceWire для всіх типів модулів, по ПК, РК, МКІО для модулів введення-виведення, по інтерфейсу Fibre Channel для модуля графічного і модуля-комутатора; – контроль процесу тестування; – завантаження програмного забезпечення, тестування і ведення файлів звіту по тестуванню кожного модуля. – Функціональна схема АРМ представлена на рисунку 5.7 АРМ з перевірки КБО містить: – інструментальну електронно-обчислювальну машину, що забезпечує установку інструментальних і програмних засобів; – технологічне обладнання (технологічні модулі, які забезпечують сполучення інтерфейсів КБО і інтерфейсів ІЕВМ, комплект з'єднай-них джгутів для з'єднання рами з встановленим КБО з технологічним обладнанням і джерелом живлення); – програмне забезпечення для тестування кожного модуля, принтер для виведення на друк результатів тестування; – джерело живлення для подачі напруги на КБО; – комплект експлуатаційної документації, що містить інструкції для перевірки і завантаження тестового програмного забезпечення. **Коваленко переписала чужий текст, у якому неправильно переклала абревіатуру «ИЭВМ» як «ІЕВМ», тоді як треба «ІЕОМ» – нерозуміння авторкою термінології.**	АРМ по проверке КФМ должно обеспечивать: - имитацию процессов информационного обмена по интерфейсам SpaceWire для всех типов модулей, по ПК, РК, МКИО для модулей ввода-вывода, по интерфейсу Fibre Channel для модуля графического и модуля-коммутатора; - контроль процесса тестирования; - загрузку программного обеспечения, тестирование и ведение файлов отчета по тестированию каждого модуля. Функциональная схема АРМ представлена на рис. 3.8. АРМ по проверке КФМ содержит: - инструментальную электронно-вычислительную машину, обеспечивающую установку инструментальных и программных средств; - технологическое оборудование (технологические модули, обеспечивающие сопряжение интерфейсов КФМ и интерфейсов ИЭВМ, комплект соединительных жгутов для соединения рамы с установленным КФМ с технологическим оборудованием и источником питания); - программное обеспечение для тестирования каждого модуля, принтер для вывода на печать результатов тестирования; - источник питания для подачи напряжения на КФМ; - комплект эксплуатационной документации, содержащий инструкции для проверки и загрузки тестового программного обеспечения.

	Коваленко переписує чужий текст, не розуміючи його. Вступне речення перед переліком перетворюється в неї в один із пунктів цього переліку: «– Функціональна схема АРМ представлена на рисунку 5.7 АРМ з перевірки КБО містить:**». Плагіат.**	
	С. 287.	**С. 87.**
		
	Рис. 5.7. Модель уніфікованого автоматизованого робочого місця (DVI – Digital Visual Interface)	Рисунок 3.8 – Функциональная схема унифицированного автоматизированного рабочего места (DVI – Digital Visual Interface).
	С. 291.	**С. 87–88.**
	Тестування починається з подачі на КБО напруги харчування. Потім завантажується і запускається необхідне програмне забезпечення для тестування модуля. Під час тестування модуля перевіряються всі вузли КБО: осередки пам'яті ОЗУ, осередки пам'яті ПЗУ, система команд процесора, система переривань процесора, пристрої та канали введення-виведення. Всі результати тестування видаються на екран, а також заносяться в файл звіту[115]. Розрізняють 2 види алгоритмів перевірки модулів в складі АРМ: в послідовному режимі (рис. 5.10 а) і в паралельному режимі (рис. 5.10 б). Інструментальна програма перевірки КБО забезпечує занесення в КБО тестового ПО за технологічним каналу і обмін інформацією між ІЕВМ і КБО. Зовнішній вигляд робочих вікон програм САПР з перевірки КБО представлені на рисунку 5.10 – 5.11 . **Коваленко переписала текст із російської дисертації 2015-го року.** **Фальшиве покликання [115] – це власна праця Коваленко 2024 р.** **Фальсифікація джерел.** **Плагіат.**	Тестирование начинается с подачи на КФМ напряжения питания. Затем загружается и запускается необходимое программное обеспечение для тестирования модуля. Во время тестирования модуля проверяются все узлы КФМ: ячейки памяти ОЗУ, ячейки памяти ПЗУ, система команд процессора, система прерываний процессора, устройства и каналы ввода-вывода. Все результаты тестирования выдаются на экран, а также заносятся в файл отчета. Различают 2 вида алгоритмов проверки модулей в составе АРМ: в последовательном режиме (рис. 3.9,а) и в параллельном режиме (рис. 3.9,б). Инструментальная программа проверки КФМ обеспечивает занесение в КФМ тестового ПО по технологическому каналу и обмен информацией между ИЭВМ и КФМ. Внешний вид рабочих окон программ САПР по проверке КФМ представлены на рис. 3.10 - 3.12.
	С. 291.	**С. 90.**

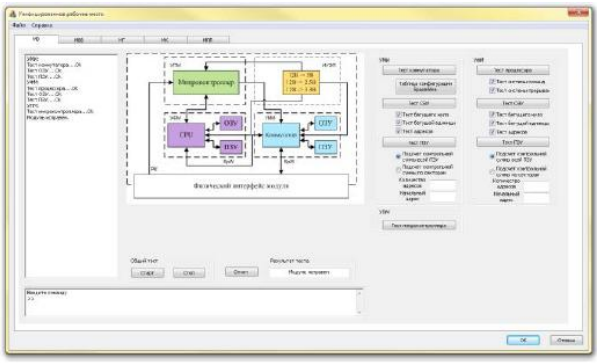


а)

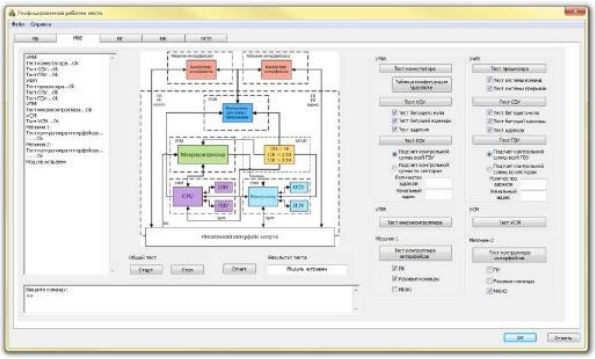


б)

Рис. 5.10. Робоче вікно програми САПР для перевірки та усунення несправностей: а) модуль обчислювальний ;б) модуль вводу-виводу



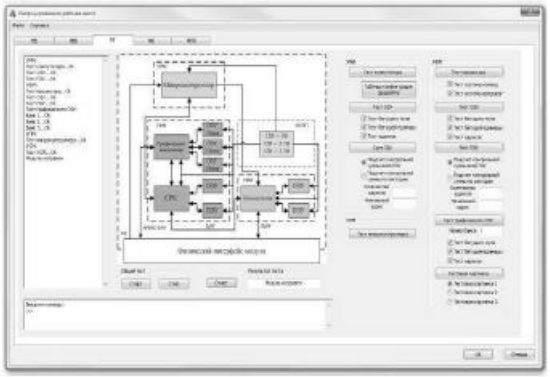
а)



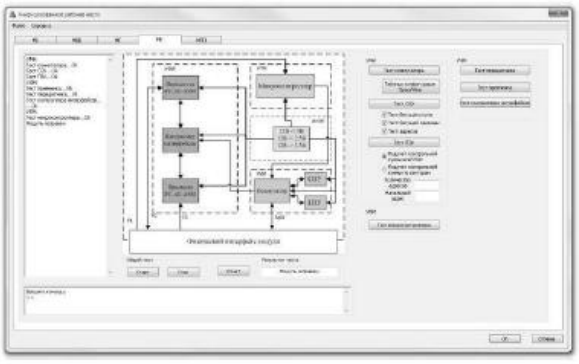
б)

Рисунок 3.10 – Рабочее окно программы САПР для проверки КФМ:
а) Модуль вычислительный (полный тест);
б) Модуль ввода-вывода (полный тест).

С. 292. **С. 91.**

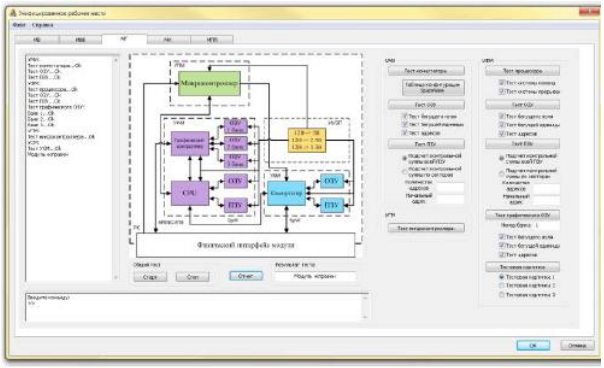


а)

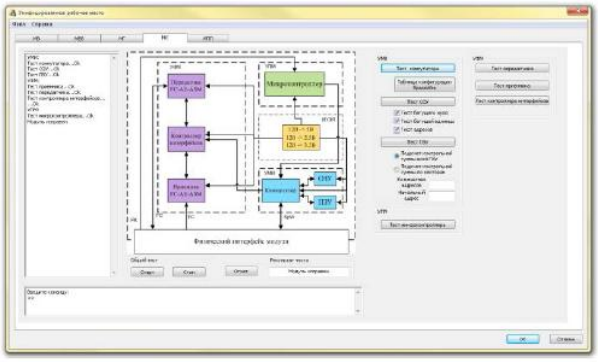


б)

Рис. 5.11 Робоче вікно програми САПР для перевірки та усунення несправностей: а) модуль гра-

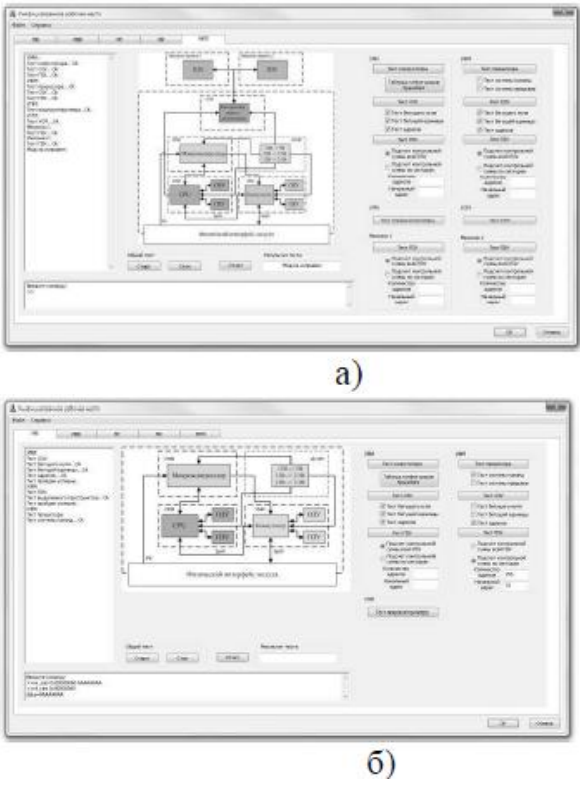
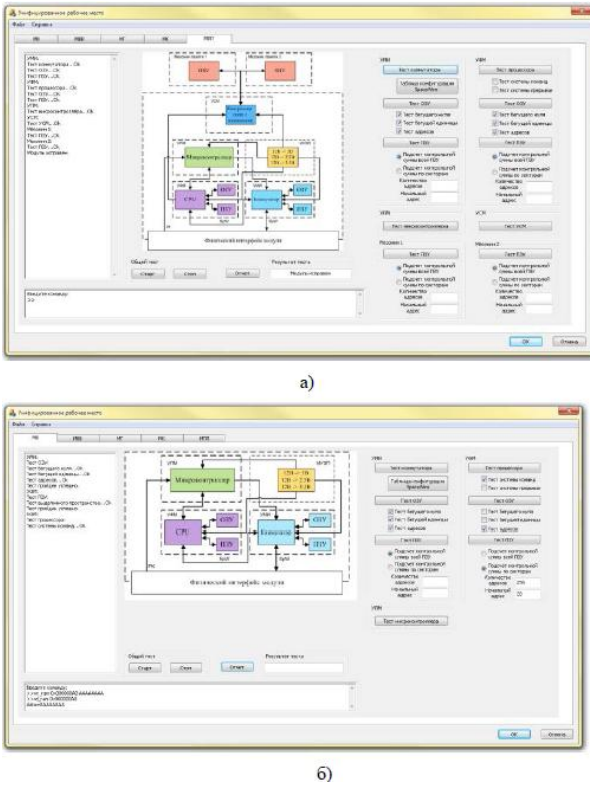


а)



б)

Рисунок 3.11 – Рабочее окно программы САПР для проверки КФМ:
а) Модуль графический (полный тест)

	фічний б) модуль комутатор	б) Модуль коммутатор (полный тест)
	С. 292.	**С. 88.**
	Робоче вікно інструментальної програми розділене на кілька фрагментів. Зліва знаходить лог проведених тестів. Знизу командний рядок для проведення окремих операцій, наприклад, читання або запис в ОЗУ / ПЗУ. У центрі робочого вікна знаходить структурна схема модуля, що тестується [116]. Під структурною схемою знаходяться кнопки початку тесту і зупинки тесту, а також кнопка для отримання звіту, а в додаткове вікно виводиться інформація про результат повного тесту: «модуль справний» або «модуль відмовив». Справа в робочому вікні знаходяться кнопки для проведення тестів окремих вузлів модуля з додатковими настройками. **Коваленко переписала текст із російської дисертації 2015-го року.** **Фальшиве покликання [116] – це власна праця Коваленко 2024 р.** **Фальсифікація джерел.** **Плагіат.**	Рабочее окно инструментальной программы разделено на несколько фрагментов. Слева находит лог проведенных тестов. Снизу командная строка для проведения отдельных операций, например, чтение или запись в ОЗУ/ПЗУ. В центре рабочего окна находит структурная схема тестируемого модуля. Под структурной схемой находятся кнопки начала теста и остановки теста, а также кнопка для получения отчета, а в дополнительное окно выводится информация о результате полного теста: «модуль исправен» или «модуль отказал». Справа в рабочем окне находятся кнопки для проведения тестов отдельных узлов модуля с дополнительными настройками.
	С. 292–293.	**С. 92.**
	 а) б) Рис. 5.12. Робоче вікно програми САПР для перевірки та усунення несправностей: а) модуль постійної пам'яті; б) модуль обчислювальний.	 а) б) Рисунок 3.12 – Рабочее окно программы САПР для проверки КФМ: а) Модуль постоянной памяти (полный тест) б) Модуль вычислительный (выполнение отдельных тестов и команд).
4	**Коваленко Ю. Б.** **Інформаційна підтримка процесу проектування та експлуатації комплексу бортового обладнання інтегрованої модульної авіоніки.** – Дис. … доктора технічних наук. – Київ, 2025. (https://duikt.edu.ua/uploads/p_86_93515477.pdf)	**Чжо Зин Хтут.** **Редуцированные динамические экспертные системы и алгоритмы повышения отказоустойчивости прицельно-навигационных комплексов летательных аппаратов.** – Дисс. … канд. техн. наук. – Москва, 2018.

	С. 58–59.	**С. 6.**
	Вимірювальні системи, що входять до складу бортових обчислювальних систем ІМА, мають похибки, обумовлені їх конструкціями та умовам, в яких функціонує ПС. Для компенсації похибок КБО використовуються відомі алгоритмічні методи, а також можливості реконфігурування структури та адаптації до зовнішніх умов і внутрішньому стану КБО. До складу бортового обладнання ПС входять базові автономні навігаційні системи, зокрема ІНС, СНС, радіонавігаційні системи, геофізичні та оптичні системи, а також комплексна обробка інформації. Однією з важливих задач при експлуатації КБО ПС є збереження його працездатності та високої ефективності. Для визначення працездатності та якості функціонування КБО застосовуються різні системи контролю.	Измерительные системы, входящие в состав ПНК, имеют погрешности, вызываемые конструктивными особенностями, а также условиями функционирования ЛА. Для компенсации погрешностей на практике используются известные алгоритмические методы, а также возможности реконфигурирования структуры и адаптации к внешним условиям и внутреннему состоянию ПНК. В состав ПНК ЛА входят базовые автономные навигационные системы, в частности инерциальные навигационные системы (ИНС), спутниковые навигационные системы (СНС), радионавигационные системы, геофизические и оптические системы, БЦВМ, а также комплексная обработка информации (КОИ). Одной из важных задач при эксплуатации ПНК ЛА является сохранение его работоспособности и высокой эффективности. Для определения работоспособности и качества функционирования ПНК применяются различные системы контроля, повышения отказоустойчивости и помехозащиты.
	С. 59.	**С. 6–7.**
	Умови експлуатації сучасних ПС характеризуються високими швидкостями, великими висотами та дальністю польоту, дією різноманітних зовнішніх збурень. В таких умовах ймовірність відмов систем зростає. Тому необхідно жорстко контролювати характеристики КБО і окремих систем авіоніки. Для цього застосовуються різні системи контролю на різних етапах експлуатації КБО. Автоматизовані бортові системи контролю, що включають вбудовані засоби інструментального контролю та системи інформаційного контролю. Використовуються ієрархічні системи засобів контролю, які добре зарекомендували себе на практиці, в яких оцінюється працездатність і достовірність інформації окремих систем і комплексу бортового обладнання в цілому.	Современные ЛА функционируют на высоких скоростях, больших высотах и дальностях полета, в условиях разнообразных и многочисленных внешних факторов. Требования к точности и надежности навигационного оборудования постоянно возрастают. Обеспечение тактико-технических характеристик ЛА и жесткие условия эксплуатации накладывают ограничения на физические, технические и эксплуатационные характеристики авионики. Важной и актуальной задачей при эксплуатации бортового оборудования ЛА является оценка состояния ПНК, диагностика его отказов и управление функционированием всей авионики. Для этого применяются различные системы повышения отказоустойчивости, помехозащиты и контроля на различных этапах эксплуатации ПНК. Используются иерархические системы, хорошо зарекомендовавшие себя на практике, в которых оценивается работоспособность и достоверность информации отдельных систем и комплекса бортового оборудования в целом.
	С. 59.	**С. 7.**
	Однак при вирішенні завдання контролю бортового обладнання доцільно знати не тільки момент відмови бортових систем, але і передбачити момент виникнення аварійної ситуації, а також інтервали недостовірної роботи обладнання. Вирішення цього завдання за допомогою апріорних прогнозують моделей вимагає проведення тривалих дорогих експериментів, не дозволяє враховувати особливості конкретних систем і здійснювати ефективний контроль високоманеврових ПС. Тому для здійснення контролю бортового устаткування перспективних ПС доцільно використовувати комплексні системи контролю на базі ДЕС, які дозволяють враховувати режими польоту ПС, мають велику базу даних і ансамбль оціночних критеріїв. Застосування ДЕС на борту ПС пов'язане зі складнощами реалізації, вимогами підвищеної продуктивності КБО. КБО ПС повинні мати бути помехо- та відмовостійкими, видавати достовірну навігаційну інформацію в усіх режимах роботи.	Однако при решении задачи сохранения работоспособности бортового оборудования целесообразно знать не только момент отказа бортовых систем, но и предвидеть момент возникновение аварийной ситуации, а также интервалы недостоверной работы оборудования. Решение этой задачи с помощью априорных прогнозирующих моделей требует проведения длительных дорогостоящих экспериментов, не позволяет учитывать особенности конкретных систем и осуществлять эффективный контроль высокоманевренных ЛА. Поэтому для повышения отказоустойчивости и осуществления контроля бортового оборудования перспективных маневренных ЛА целесообразно использовать комплексные системы контроля на базе динамических экспертных систем (ДЭС), которые позволяют учитывать режимы полета ЛА, имеют богатую базу данных и ансамбль оценочных критериев. Применение ДЭС на борту ЛА сопряжено со сложностями реализации, требованиями повышенной производительности БЦВС.
	С. 60.	**С. 35.**

	Таким чином, виділений комплекс проблем, який доцільно вирішити в процесі дисертаційного дослідження: розробити структуру системи контролю працездатності КБО ПС на основі принципів функціонування ДЕС, а також високоточні алгоритми контролю стану КБО і окремих навігаційних систем ПС. При синтезі систем контролю та діагностики необхідно враховувати можливість їх реалізації в умовах обмежень потужності КБО. **Коваленко пише «**Таким чином, виділений комплекс проблем**», а насправді переписує чужий текст.** **Плагіат.**	Таким образом, выделен комплекс проблем, который целесообразно решить в процессе диссертационного исследования: разработать структуру системы контроля работоспособности ПНК ЛА на основе принципов функционирования ДЭС, а также высокоточные алгоритмы контроля состояния ПНК и отдельных навигационных систем ЛА. При синтезе систем контроля и диагностики необходимо учитывать возможность их реализации в условиях ограничений мощности БЦВС.
	С. 60.	**С. 36.**
	На основі проведеного аналізу систем контролю та їх алгоритмічного забезпечення зроблено висновок про необхідність розробки універсального алгоритмічного засобу контролю та діагностики стану сучасних вимірювальних комплексів КБО ПС. **Коваленко пише «**На основі проведеного аналізу … зроблено висновок**», а насправді переписує чужий текст.** **Плагіат.**	На основе проведенного анализа систем контроля и их алгоритмического обеспечения сделан вывод о необходимости разработки универсального алгоритмического средства контроля и диагностики состояния современных измерительных комплексов и ПНК маневренных ЛА.
	С. 185.	**С. 76.**
	В розділі описано синтез системи діагностики працездатності вимірювальних приладів ПС. Розглянуто технології розробки діагностичної системи на основі моделі похибок приладів, які використовуються для вимірювання параметрів ПС. Вимоги по надійності точності, що пред'являються до вимірювальних систем ПС, постійно зростають. Для забезпечення необхідних характеристик вимірювальних систем і приладів необхідно мати інформацію про їх стан, точностних характеристиках і перспективах зміни цих параметрів в майбутньому. Зазначені завдання вирішуються за допомогою різних систем діагностики і контролю [2, 49, 43]. Синтез системи діагностики доцільно здійснювати з використанням підходу алгоритмічного конструювання, що дозволяє досягти необхідного результату за короткий час з мінімальними фінансовими витратами.	В разделе описан синтез системы диагностики работоспособности измерительных приборов ЛА. Рассмотрены технологии разработки диагностической системы на основе модели погрешностей гироприборов, которые используются для измерения параметров ЛА. Требования по над жности и точности, предъявляемые к измерительным системам ЛА, постоянно возрастают. Для обеспечения требуемых характеристик измерительных систем и приборов необходимо иметь информацию об их состоянии, точностных характеристиках и перспективах изменения этих параметров в будущем. Указанные задачи решаются с помощью различных систем диагностики и контроля [2, 49, 43]. Синтез системы диагностики целесообразно осуществлять с использованием подхода алгоритмического конструирования, позволяющим достичь требуемого результата за короткое время с минимальными финансовыми затратами.
	С. 185.	**С. 76.**
	Реалізація систем контролю і діагностики ПС здійснюється в бортовий цифровий обчислювальної машині. Тобто, створення прототипу діагностичної системи визначення та прогнозування стану вимірювальної техніки ПС. Життєздатність та розвиток інформаційних технологій пояснюється тим, що сучасний ПС вкрай чутливий до помилок в управлінні. Помилки і збої в роботі алгоритмічного забезпечення знижують якість функціонування систем управління і можуть привести до відмов і фатальним результатам.	Реализация систем контроля и диагностики ЛА осуществляется в бортовой цифровой вычислительной машине (БЦВМ). Цель данной работы - создание прототипа диагностической системы определения и прогнозирования состояния измерительной техники ЛА. Жизнеспособность и развитие информационных технологий объясняется тем, что современный ЛА крайне чувствителен к ошибкам в управлении. Ошибки и сбои в работе алгоритмического обеспечения снижают качество функционирования систем управления и могут привести к отказам и фатальным результатам.
	С. 185.	**С. 76–77.**

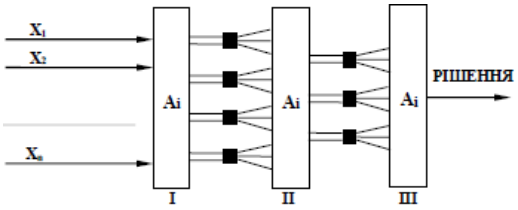
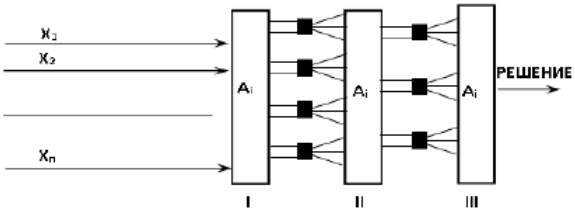
	Тому надзвичайно важливо зрозуміти, які вимірювальні прилади вийшли з ладу, рівень їх точності і перспективи зміни точностних характеристик. Ця інформація дозволяє прийняти рішення про способи використання сигналів вимірювальних приладів в системі управління ПС. Система автоматичного діагностування являє собою комплекс програмних і апаратурних засобів і є складовою частиною ДЕС.	Поэтому чрезвычайно важно понять, какие измерительные приборы вышли из строя, уровень их точности и перспективы изменения точностных характеристик. Эта информация позволяет принять решение о способах использования сигналов измерительных приборов в системе управления ЛА. Система автоматического диагностирования представляет собой комплекс программных и аппаратурных средств и является составляющей частью ДЭС.
	С. 186.	**С. 77.**
	Розглянемо синтез систем діагностики на прикладі задачі оцінки і прогнозування похибок БРЕО на борту ПС. Методи оцінки і прогнозування інструментальних похибок БРЕО. Обурення викликаючі кутову швидкість догляду Ω БРЕО, яку можна розділити на швидкість догляду Ωн, не залежну від прискорення, і швидкість догляду Ωр, яка залежить від прискорення і пропорційна розбалансування. Ωр складається з швидкості догляду Ω_{pI} і ΩpII, пропорційні осьової і радіальної розбалансуванням відповідно [61]. Найбільший вплив на стан обладнання надає складова Ω_{pI}. **Помилка перекладу: російський вислів «Возмущения вызывающие угловую скорость ухода Ω гироприборов» Коваленко переклала як «Обурення викликаючі кутову швидкість догляду Ω БРЕО», і далі теж кілька разів слово «скорость ухода» переклала як «швидкість догляду».** **Коваленко залишила в переписаному чужому тексті покликання [61], але під цим номером знаходиться її власна стаття.** **Фальсифікація джерел.** **Плагіат.**	Рассмотрим синтез систем диагностики на примере задачи оценки и прогнозирования погрешностей гироприборов на борту ЛА. **Методы оценки и прогнозирования инструментальных погрешностей гироприборов ПНК.** Возмущения вызывающие угловую скорость ухода Ω гироприборов, которую можно разделить на скорость ухода $\Omega_н$, не зависящую от ускорения, и скорость ухода $\Omega_р$, которая зависит от ускорения и пропорциональна разбалансировке. $\Omega_р$ состоит из скорости ухода Ω_{pI} и ΩpII , пропорциональные осевой и радиальной разбалансировкам соответственно [61]. Наибольшее влияние на состояние гироприбора оказывает составляющая Ω_{pI} .
	С. 186.	**С. 77.**
	Тому для оцінки та прогнозування інструментальних похибок БРЕО можна використовувати, наприклад, оцінку і прогнозування працездатності апаратного забезпечення, яка залежить від швидкості догляду Ω_{pI}. Даний метод полягає в визначенні швидкості догляду Ω_{pI} за інформацією, яка вимірювалася в попередніх випробуваннях і по вимірах при запуску приладів для введення поправок. Далі проводиться прогноз величини стабільної складової Ω_{pI}. Прогноз здійснюється на наступний запуск приладів і обчислюється оцінка відповідності його допуску. На інтервалі польоту проводиться визначення та прогноз точностних характеристик приладів. Розрахунок змінної складової Ω_{pI} проводиться в заданому допуску. Останні два пункти дозволяють прийняти рішення про можливість постановки приладів на борт ПС [78]. **Помилка перекладу: російський вислів «скорость ухода» Коваленко переклала як «швидкість догляду».** **Коваленко до переписаного чужого тексту додала покликання [78].** **Фальсифікація джерел.** **Плагіат.**	Поэтому для оценки и прогнозирования инструментальных погрешностей гироприборов, применяемых в ПНК перспективных ЛА, можно использовать, например, оценку и прогнозирование работоспособности гироприбора, которая зависит от скорости ухода Ω_{pI} . Данный метод заключается в определение скорости ухода Ω_{pI} по информации, которая измерялась в предыдущих испытаниях и по замерам при запуске гироприбора для ввода поправок. Далее проводится прогноз величины стабильной составляющей Ω_{pI} («скачка»). Прогноз дается на последующий запуск гироприбора и вычисляется оценка соответствия его допуску. На интервале полета проводится определение и прогноз точностных характеристик гироприбора. Расчет переменной составляющей Ω_{pI} проводится в заданном допуске. Последние два пункта позволяют принять решение о возможности постановки гироприбора на борт ЛА.

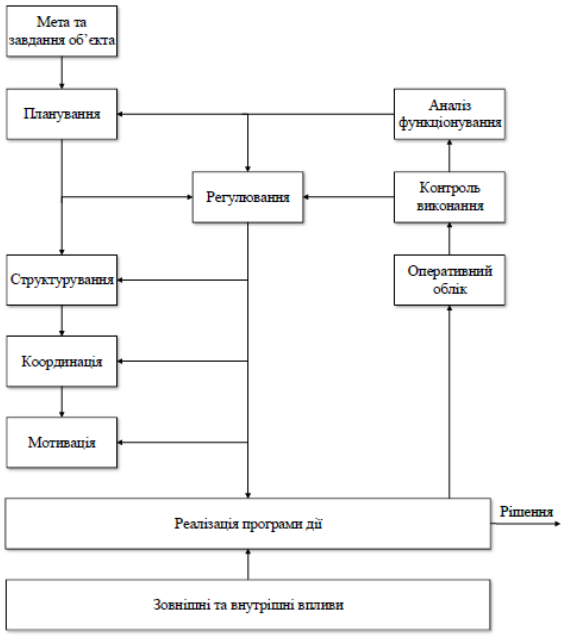
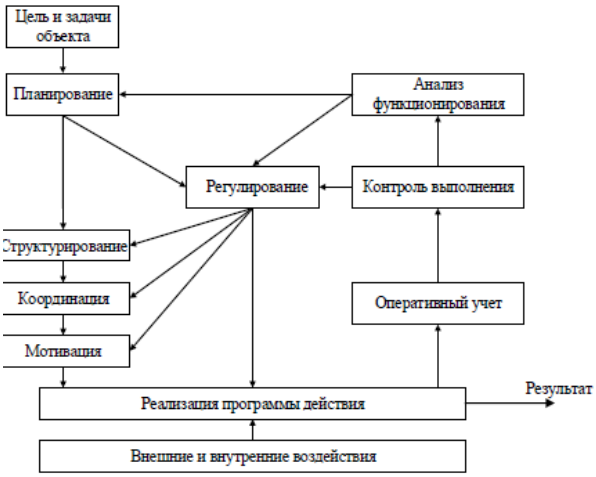
	С. 186.	**С. 77–78.**
	У практичних додатках проводиться рішення задачі оцінки і прогнозу інструментальних похибок приладів паралельно: визначається стабільна складова швидкості догляду Ω_{pl} при запуску, яка використовується в алгоритмах управління для формування поправок; обчислюється оцінка і прогноз характеристик приладів на інтервалі польоту ПС [99]. **Помилка перекладу: російський вислів «скорость ухода» Коваленко переклала як «швидкість догляду».** **Коваленко до переписаного чужого тексту додала покликання [99], але під цим номером знаходиться її власна стаття 2020-го року. Джерело плагіату – 2018-го року.** **Фальсифікація джерел.** **Плагіат.**	В практических приложениях проводится решение задачи оценки и прогноза инструментальных погрешностей гироприборов ПНК параллельно: определяется стабильная составляющая скорости ухода Ω_{pl} при запуске, которая используется в алгоритмах управления для формирования поправок; вычисляется оценка и прогноз характеристик гироприбора на интервале полёта ЛА.
	С. 186–187.	**С. 78.**
	Модель швидкості догляду приладів Ω_{pl} має вигляд: $\Omega_{M_t} = \overset{з}{\ell}(\Omega_t^з) + \alpha_\Delta(\Delta\Omega_t^з) + \overset{с}{\ell}(\Omega_t^с),$ (4.1) де Ω_{Mt} – швидкість догляду приладів Ω_p; $\Omega^з_t$, $\Delta\Omega^з_t$ – стабільна і змінна складові при конкретному запуску приладів; $\Omega^с_t$ – значення «стрибка» стабільної складової швидкості догляду приладів при дослідженні на різних запусках; $\overset{з}{\ell}, \alpha_\Delta, \overset{с}{\ell}$ – параметри моделі. **Коваленко з переписаного чужого тексту видалила покликання [78],** **Помилка перекладу: російський вислів «скорость ухода» Коваленко переклала як «швидкість догляду».** **Плагіат.**	Модель скорости ухода гироприбора Ω_{pl} имеет вид [78]: $\Omega_{M_t} = \ell^з(\Omega_t^з) + \alpha_\Delta(\Delta\Omega_t^з) + \ell^с(\Omega_t^с)$, (4.1) где Ω_{Mt} – скорость ухода гироприбора Ω_p ; $\Omega^з_t$, $\Delta\Omega^з_t$ – стабильная и переменная составляющие при конкретном запуске гироприбора; $\Omega^с_t$ – значение «скачка» стабильной составляющей скорости ухода гироприбора при исследовании на различных запусках; $\ell^з, \alpha_\Delta, \ell^с$ - параметры модели.
	С. 187.	**С. 78.**
	Введення поправок зазвичай здійснюється на Ωз, тому $\Omega_{M_t} = \overset{з}{\ell}(\Omega_t^з)$, якість здійснення корекції за допомогою стабільної складової швидкості характеризується ступенем наближення розрахункової швидкості догляду до е фактичному значенню, тобто $/\Omega_t^з - \Omega_{M_t}/ \le \varepsilon_\Omega \quad \text{при} \quad t_{вим} \le t_{доп},$ (4.2) де ε_Ω – точність оцінки стабільної складової швидкості виходу на інтервалі е вимірювання $t_{вим}$, що не перевищує допустимого значення $t_{доп}$.	Ввод поправок обычно осуществляется на $\Omega^з_t$, поэтому $\Omega_{M_t} = \ell^з(\Omega_t^з)$, качество осуществления коррекции с помощью стабильной составляющей скорости характеризуется степенью приближения расчетной скорости ухода к её фактическому значению, т.е. $/\Omega_t^з - \Omega_{M_t}/ \le \varepsilon_\Omega \quad \text{при} \quad t_{изм} \le t_{доп.},$ (4.2) где ε_Ω – точность оценки стабильной составляющей скорости ухода на интервале её измерения $t_{изм}$, не превышающего допустимого значения $t_{доп.}$.
	С. 187.	**С. 78–79.**
	В умовах вдало сформованої похибка оцінки Ω_{Mt} залежить тільки від точності визначення її параметрів. Важливою характеристикою при оцінці параметрів моделі є їх стабільність в сенсі наступних критеріїв:	В условиях удачно сформированной (4.1) погрешность оценки Ω_{Mt} зависит только от точности определения ее параметров. Важной характеристикой при оценке параметров модели является их стабильность в смысле следующих критериев [99]:

	$K_I = \left\|M\left[\ell^3\right] - \ell_i^3\right\| \le \Delta;\quad K_{II} = \sqrt{\frac{\sum_{i=1}^{n}(\ell_i^3 - M\left[\ell_i^3\right])^2}{n-1}} - M \le \eta_\ell;$ $K_{III} = \sqrt{\frac{\sum_{i=1}^{n}(\ell_i^3 - \ell_{i-1}^3)^2}{n}} \le \varepsilon_\ell,$ (4.3) де $M[\ell^3]$ – математичне очікування; Δ, η_ℓ, ε_ℓ – задані точності визначення параметрів моделі. Ці параметри визначають виходячи з умови забезпечення мінімального впливу інструментальних похибок БРЕО на точність переміщення ПС в задану область фазового простору; i = 1,2, ..., n - послідовність оцінок параметрів моделі. **Коваленко з переписаного чужого тексту видалила покликання [99],** **Плагіат.**	$K_I = \left\|M[\ell^3] - \ell_i^3\right\| \le \Delta;$ $K_{II} = \sqrt{\frac{\sum_{i=1}^{n}(\ell_i^3 - M[\ell^3])^2}{n-1}} \le \eta_\ell;$ $K_{III} = \sqrt{\frac{\sum_{i=1}^{n}(\ell_i^3 - \ell_{i-1}^3)^2}{n}} \le \varepsilon_\ell,$ (4.3) где M[l^3] - математическое ожидание; Δ, η_l, ε_l – заданные точности определения параметров модели. Эти параметры определяют исходя из условия обеспечения минимального влияния инструментальных погрешностей ПНК на точность перемещения ЛА в заданную область фазового пространства; i = 1,2,...,n - последовательность оценок параметров модели.
	С. 187–188.	**С. 79.**
	Критерій K_{III} є чутливий до зміни параметрів моделі (4.1), яка використовується для оцінки і прогнозу працездатності приладів БОІС. Процес зміни «стрибка» характеризує член $\Omega_{M_t} = \ell^c(\Omega_t^c)$. Другий член (4.1) характеризує зміну змінної складової швидкості догляду приладів. При цьому час знаходження $\Delta\Omega_t^3$ в межах допуску можна обчислити за допомогою змішаного розподілу Вейбулла [104]. **Коваленко залишила в переписаному чужому тексті покликання [104], але під цим номером знаходиться її власна стаття 2020-го року. Джерело плагіату – 2018-го року.** **Фальсифікація джерел.** **Плагіат.**	Критерий K_{III} является чувствителен к изменению параметров модели (4.1), которая используется для оценки и прогноза работоспособности гироприбора ПНК. Процесс изменения «скачка» характеризует член $\Omega_{M_t} = \ell^c(\Omega_t^c)$ Второй член (4.1) характеризует изменение переменной составляющей скорости ухода гироприбора. При этом время нахождения $\Delta\Omega_t^3$ в пределах допуска можно вычислить посредством смешанного распределения Вейбулла [104]
	С. 188.	**С. 79.**
	$f(t) = \left\{\frac{\beta_\Omega}{\alpha_\Omega}(e-\gamma_\Omega)^{\beta_\Omega - 1}\exp\left[-\frac{(t-\gamma_\Omega)^{\beta_\Omega}}{\alpha_\Omega}\right]\right\}; t \ge \gamma_\Omega, \beta_\Omega > 0, \alpha_\Omega > 0,$ (4.4) 0 – в інших випадках, де α_Ω, β_Ω – параметри масштабу і форми, що характеризують, відповідно, час знаходження $\Delta\Omega^3_t$ в межах допуску і ступінь зносу приладів; γ_Ω – параметр положення, що визначає кінцевий час, до якого вихід $\Delta\Omega^3_t$ за допуск неможливий.	$f(t) = \left\{\frac{\beta_\Omega}{\alpha_\Omega}(e-\gamma_\Omega)^{\beta_\Omega - 1}\exp\left[-\frac{(t-\gamma_\Omega)^{\beta_\Omega}}{\alpha_\Omega}\right]; \; t \ge \gamma_\Omega, \; \beta_\Omega > 0, \; \alpha_\Omega > 0\right.$ $\left. 0 - \text{в остальных случаях,}\right.$ (4.4) где α_Ω, β_Ω - параметры масштаба и формы, характеризующие, соответственно, время нахождения $\Delta\Omega^3_t$ в пределах допуска и степень износа гироприбора; γ_Ω – параметр положения, определяющий конечное время, до которого выход $\Delta\Omega^3_t$ за допуск невозможен.
	С. 188.	**С. 79.**
	При $\gamma_\Omega \approx 0$ ймовірність P(t) і середній час μ_t знаходження $\Delta\Omega^3_t$ в заданому допуску обчислюються наступним чином $P(t) = \exp\left[-\frac{t^{\beta_\alpha}}{\alpha_\Omega}\right]; \mu t = (\alpha_\Omega)\frac{1}{\beta_\Omega} - \left(1 + \frac{1}{\beta_\Omega}\right),$ (4.5) де $\Gamma(o)$ – гамма функція. **Коваленко переписала чужу формулу з помилкою: замість піднесення до степеня – помножила.** **Недолугий плагіат.**	При $\gamma_\Omega \approx 0$ вероятность P(t) и средне время μ_t нахождения $\Delta\Omega^3_t$ в заданном допуске вычисляются следующим образом $P(t) = \exp\left[-\frac{t^{\beta_\Omega}}{\alpha_\Omega}\right]; \; \mu_t = (\alpha_\Omega)^{\frac{1}{\beta_\Omega}} - \left(1 + \frac{1}{\beta_\Omega}\right),$ (4.5) где $\Gamma(o)$ – гамма-функция.
	С. 188.	**С. 79–80.**

	Зміни швидкості догляду в запусках $\Omega_{M_t} = \ell^c(\Omega_t^c)$ описаний авторегресійною моделлю: $\Omega_k^з = (1+\ell_1^з)\Omega_{k-1} + \ell_1^з\Omega_{k-2},$ (4.6) а процес зміни «стрибків» стабільної складової швидкості догляду $\Omega_{M_t} = \ell^c(\Omega_t^c)$ моделлю $\Omega_k^c = \ell_1^c\Omega_{k-1},$ (4.7) **Знову та сама помилка перекладу: російський вислів «скорость <u>ухода</u>» Коваленко переклала як «швидкість <u>догляду</u>».** **Плагіат.**	Изменения скорости ухода в запусках $\Omega_{M_t} = \ell^з(\Omega_t^з)$ описан авторегрессионной моделью: $\Omega_k^з = (1 + \ell_1^з)\Omega_{k-1} + \ell_1^з\Omega_{k-2},$ (4.6) а процесс изменения «скачков» стабильной составляющей скорости ухода $\Omega_{M_t} = \ell^c(\Omega_t^c)$ моделью $\Omega_k^c = \ell_1^c\Omega_{k-1},$ (4.7)
	С. 188.	**С. 80.**
	Параметри моделей визначаються в наступному вигляді $\ell_t^з = \frac{\sum_{i=2}^{k}\omega_i\omega_{i-1}}{\sum_{i=2}^{k}\omega_{i-1}^2};\ \ell_1^c = \frac{\sum_{i=2}^{k}\Omega_i\Omega_{i-1}}{\sum_{i=2}^{k}\Omega_{i-1}^2},$ (4.8) де $\omega_k=\Omega_k-\Omega_{k-1}$ – різниця першого порядку.	Параметры моделей определяются в следующем виде $\ell_t^з = \frac{\sum_{i=2}^{k}\omega_i\omega_{i-1}}{\sum_{i=2}^{k}\omega_{i-1}^2};\quad \ell_1^c = \frac{\sum_{i=2}^{k}\Omega_i\Omega_{i-1}}{\sum_{i=2}^{k}\Omega_{i-1}^2},$ (4.8) где $\omega_k = \Omega_k - \Omega_{k-1}$ – разность первого порядка.
	С. 188–189.	**С. 80.**
	Параметри Вейбулла в виразі (4.8) визначимо методом максимальної правдоподібності. Формалізовані залежності, які використані для оцінки параметрів α_Ω і β_Ω мають такий вигляд: $n\alpha_\Omega - \sum_{i=1}^{n} t_i^{\beta_\Omega} = 0;$ $\frac{n}{\beta_\Omega} = \sum_{i=1}^{n}\ln t_i - \frac{1}{\alpha_\Omega}\sum_{i=1}^{n} t_i^{\beta_\Omega}\ln t_i = 0,$ (4.9) де n – кількість інформації про час, коли прилад знаходиться в межах робочого допуску; t_i – час знаходження в допуску за результатами випробувань приладу (i = 1, 2, ..., n).	Параметры Вейбулла в выражении (4.4) определим методом максимального правдоподобия. Формализованные зависимости, используемые для оценки параметров α_Ω и β_Ω имеют следующий вид: $n\alpha_\Omega - \sum_{i=1}^{n} t_i^{\beta_\Omega} = 0;$ $\frac{n}{\beta_\Omega} + \sum_{i=1}^{n}\ln t_i - \frac{1}{\alpha_\Omega}\sum_{i=1}^{n} t_i^{\beta_\Omega}\ln t_i = 0,$ (4.9) где n – количество информации о времени, когда прибор находится в пределах рабочего допуска; t_i – время нахождения в допуске по результатам испытаний прибора (i=1, 2,…, n).
	С. 189.	**С. 80.**
	Рішення рівнянь проводиться з використанням методу випадкового пошуку. Знаходимо мінімальну нев'язку $\Delta E = /E_1/ - / E_2/,$ (4.10) де E_1, E_2 – є, відповідно, першим і другим рівняннями системи . На кожному кроці пошуку значень параметрів α_Ω і β_Ω приймаються величини $\overline{h}^{(N)} = \ell\xi^{(N)},$ (4.11) де $\overline{h} = (\alpha_\Omega, \beta_\Omega)^N$, N – кількість пошукових кроків; ℓ – масштаб можливих меж змін $\overline{h}$; ξ – нормальний розподіл.	Решение уравнений (4.9) проводится с использованием метода случайного поиска. Находим минимальную невязку $\Delta E = /E_1 / - / E_2/,$ (4.10) где E_1, E_2 – являются, соответственно, первым и вторым уравнениями системы (4.9). На каждом шаге поиска значений параметров α_Ω и β_Ω принимаются величины $\overline{h}^{(N)} = \ell\xi^{(N)},$ (4.11) где $\overline{h} = (\alpha_\Omega, \beta_\Omega)^T$; N – количество поисковых шагов; ℓ – масштаб возможных пределов изменения $\overline{h}$; ξ – нормальное распределение.
	С. 189.	**С. 80–81.**
	Якщо умова $\Delta E<\varepsilon_2$, де ε_2 – задана точність рішення, виконується, то пошук завершується. Отримані значення α_Ω і β_Ω дозволяють визначити показники (30). Розглянутий метод за допомогою (4.3) і (4.4) дозволяє оцінити і спрогнозувати стан приладів в складі БЦОМ. Необхідно виконання умов: $\Omega_k^c \le \Omega_{доп}^c;\ P(t) \ge P_{зад};\ \mu_t \ge t_{пол},$ (4.12) де $\Omega_{доп}^c$ – допустиме значення «стрибка» Ω_{PI}; $P_{зад}$ – задана ймовірність знаходження змінної складової швидкості догляду в діапазоні на інтер-	Если условие $\Delta E < \varepsilon_2$, где ε_2 – заданная точность решения, выполняется, то поиск завершается. Полученные значения α_Ω и β_Ω позволяют определить показатели (4.5). Рассмотренный метод посредством (4.5) и (4.7) позволяет оценить и спрогнозировать состояние гироприбора в составе ПНК. Необходимо выполнение условий: $\Omega_k^c \le \Omega_{доп}^c;\quad P(t) \ge P_{зад};\ \mu_t \ge t_{пол},$ (4.12) где $\Omega_{доп}^c$ – допустимое значение «скачка» Ω_{PI} ; $P_{зад}$ – заданная вероятность нахождения переменной составляющей скорости ухода в диапазоне на

	валі польоту ПС $t_{пол}$. В процесі випробувань приладів уточнюються параметри Вейбулла за наступним алгоритмом $\alpha_\Omega = \alpha_\Omega^{ср} - \frac{1}{n+K}(\alpha_\Omega^{ср} - \alpha_T);\ \ \beta_\Omega = \beta_\Omega^{ср} - \frac{1}{n+K}(\beta_\Omega^{ср} - \beta_T),$ (4.13) де K, α_T, β_T – визначаються в результаті випробувань кількість запусків і параметри.	интервале полёта ЛА $t_{пол}$ В процессе испытаний гироприбора уточняются параметры Вейбулла по следующему алгоритму $\alpha_\Omega = \alpha_\Omega^{ср} - \frac{1}{n+K}(\alpha_\Omega^{ср} - \alpha_T);$ $\beta_\Omega = \beta_\Omega^{ср} - \frac{1}{n+K}(\beta_\Omega^{ср} - \beta_T),$ (4.13) где K,α_T, β_T – определяемые в результате испытаний количество запусков и параметры.
	С. 190.	**С. 81.**
	Після уточнення α_Ω і β_Ω розраховуються P(t), μ_t і відповідно до виразом (4.13) за результатами випробувань приладів приймається рішення про можливість його установки на борт ПС. Оцінку Ω_t^3 вводять в алгоритми управління за допомогою моделі (4.1) при значенні параметра, визначеному за результатами заводських випробувань гприладів. Це дозволяє істотно скоротити необхідний інтервал для оцінки і прогнозування швидкості догляду приладів. **Коваленко у переписаному чужому тексті неправильно вказала номер моделі (4.1 замість 4.6) і пропустила параметр ℓ_i^3 .** **Недолугий плагіат.**	После уточнения α_Ω и β_Ω рассчитываются P(t), μ_t и в соответствии с выражением (4.12) по результатам испытаний гироприбора принимается решение о возможности его установки на борт ЛА. Оценку Ω_t^3 вводят в алгоритмы управления с помощью модели (4.6) при значении параметра ℓ_i^3, определённом по результатам заводских испытаний гироприбора. Это позволяет существенно сократить требуемый интервал для оценки и прогнозирования скорости ухода гироприбора.
	С. 190.	**С. 81.**
	Прогнозування стану БРЕО ПС здійснюється на основі апріорної математичної моделі. На сучасному етапі з'явилася можливість використовувати на борту ПС еволюційні алгоритми побудови прогнозуючих моделей, які базуються на методах самоорганізації, генетичних підходах і нейронних мережах [31, 25, 45]. Також широко використовуються методи, засновані на ідеях ідентифікації параметрів моделі з заданою структурою [44]. **Коваленко переписала чужий текст з тими самими номерами покликань, але під цими номерами знаходяться зовсім інші джерела.** **Фальсифікація джерел.** **Плагіат.**	**Методы прогнозирования.** Прогнозирование состояния ПНК ЛА осуществляется на основе априорной математической модели. На современном этапе появилась возможность использовать на борту ЛА эволюционные алгоритмы построения прогнозирующих моделей, которые базируются на методах самоорганизации, генетических подходах и нейронных сетях [31, 25, 45]. Также широко используются методы, основанные на идеях идентификации параметров модели с заданной структурой [44].
	С. 190.	**С. 82.**
	Застосування апріорних моделей виду (4.1) відрізняється простотою і досить високою точністю, тому що базуються на великому статистичному матеріалі. В алгоритмах прогнозу, заснованих на ідентифікації параметрів моделі з апріорно заданою структурою, основна складність полягає в попередньому виборі адекватної апріорної інформації про структуру моделі. Алгоритми самоорганізації - це багаторядні алгоритми, що базуються на гіпотезі селекції. Необхідно вирішити систему нормалізованих рівнянь Гаусса: $\begin{cases} \sum_{i=1}^{N} y_i = \alpha_0 N + \alpha_1 \sum_{i=1}^{N} g_{0,j}(x_i) + \alpha_2 \sum_{i=1}^{N} g_{1,j}(x_i); \\ \sum_{i=1}^{N} y_i g_{0,j}(x_i) = \alpha_0 \sum_{i=1}^{N} g_{0,j}(x_i) + \alpha_1 \sum_{i=1}^{N} g_{0,j}^2(x_i) + \alpha_2 \sum_{i=1}^{N} g_{0,j}(x_i) g_{1,j}(x_i); \\ \sum_{i=1}^{N} y_i g_{1,j}(x_i) = \alpha_0 \sum_{i=1}^{N} g_{1,j}(x_i) + \alpha_1 \sum_{i=1}^{N} g_{0,j}(x_i) g_{1,j}(x_i) + \alpha_2 \sum_{i=1}^{N} g_{1,j}^2(x_i). \end{cases}$ (4.14)	Применение априорных моделей вида (4.1) отличается простотой и достаточно высокой точностью, так как базируются на большом статистическом материале. В алгоритмах прогноза, основанных на идентификации параметров модели с априорно заданной структурой, основная сложность заключается в предварительном выборе адекватной априорной информации о структуре модели. Алгоритмы самоорганизации - это многорядные алгоритмы, базирующиеся на гипотезе селекции. Необходимо решить систему нормализованных уравнений Гаусса: $\begin{cases} \sum_{i=1}^{N} y_i = a_0 N + a_1 \sum_{i=1}^{N} g_{0,j}(x_i) + a_2 \sum_{i=1}^{N} g_{1,j}(x_i) \\ \sum_{i=1}^{N} y_i g_{0,j}(x_i) = a_0 \sum_{i=1}^{N} g_{0,j}(x_i) + a_1 \sum_{i=1}^{N} g_{0,j}^2(x_i) + a_2 \sum_{i=1}^{N} g_{0,j}(x_i) g_{1,j}(x_i) \\ \sum_{i=1}^{N} y_i g_{1,j}(x_i) = a_0 \sum_{i=1}^{N} g_{1,j}(x_i) + a_1 \sum_{i=1}^{N} g_{0,j}(x_i) g_{1,j}(x_i) + a_2 \sum_{i=1}^{N} g_{1,j}^2(x_i) \end{cases}$ (4.14)

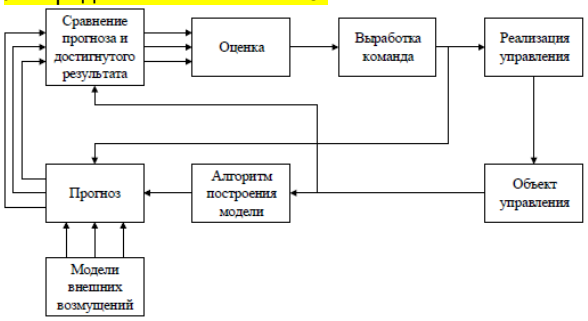
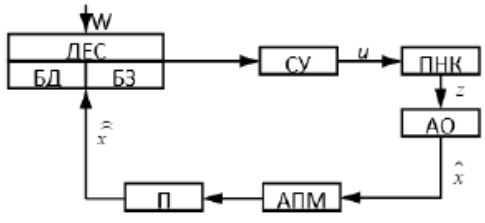
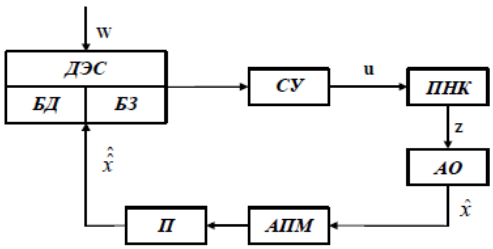
	С. 191.	С. 82.
	При введенні в базис константи від вільного члена в рівняннях можна відмовитися: $y_i = \alpha_{0,j} g_{0,j}(x_i) + \alpha_1 g_{1,j}(x_i), j = 1..C_N^2.$ $\sum_{i=1}^{N} y_i g_{0,j}(x_i) = \alpha_0 \sum_{i=1}^{N} g_{0,j}^2(x_i) + \alpha_1 \sum_{i=1}^{N} g_{0,j}(x_i) g_{1,j}(x_i)$ $\sum_{i=1}^{N} y_i g_{1,j}(x_i) = \alpha_0 \sum_{i=1}^{N} g_{0,j}(x_i) g_{1,j}(x_i) + \alpha_1 \sum_{i=1}^{N} g_{1,j}^2(x_i).$ (4.15) Обчислення припиняються, коли досягається мінімум використовуваного групи критеріїв. Результатом є краща модель в останньому ряду. Функціональна схема алгоритму МГУА представлена на рисунку 3.14. **Помилка перекладу: російський вислів «**используемого ансамбля критериев**» Коваленко неправильно переклала як «**використовуваного групи критеріїв**», тоді як треба або «використовуваної групи критеріїв», або «використовуваного ансамблю критеріїв».** **Смішний плагіат.**	При введении в базис константы от свободного члена в уравнениях можно отказаться. $y_i = a_{0,j} g_{0,j}(x_i) + a_{1,j} g_{1,j}(x_i), \quad j = 1..C_N^2.$ $\begin{cases} \sum_{i=1}^{N} y_i g_{0,j}(x_i) = a_0 \sum_{i=1}^{N} g_{0,j}^2(x_i) + a_1 \sum_{i=1}^{N} g_{0,j}(x_i) g_{1,j}(x_i) \\ \sum_{i=1}^{N} y_i g_{1,j}(x_i) = a_0 \sum_{i=1}^{N} g_{0,j}(x_i) g_{1,j}(x_i) + a_1 \sum_{i=1}^{N} g_{1,j}^2(x_i) \end{cases}$ (4.15) Вычисления прекращаются, когда достигается минимум используемого ансамбля критериев. Результатом является лучшая модель в последнем ряду. Функциональная схема алгоритма МГУА представлена на Рис. 4.1.
	С. 191.	**С. 83.**
	 Рис. 4.1. Функціональна схема модуля головного управління авіонікою (МГУА) де: Ai – порогові самовідбори корисної інформації; I, II, III – критерії самовідбору. Отримана модель використовується для прогнозу і подальшої корекції в вихідному сигналі ІНС БЦОМ ПС. Еволюційні алгоритми реалізують механізм природного відбору. В процесі функціонування алгоритму пристосованість моделей-претендентів в цілому зростає. Еволюційні алгоритми застосовуються для вирішення завдань, що мають багато локальних мінімумів, а їх недоліками, при використанні в системах діагностики на борту ПС, є підвищені вимоги до продуктивності обчислювальних засобів при реалізації на борту ПС, а також необхідність обліку особливостей проектування алгоритмів (вдалий вибір групи критеріїв селекції алгоритму самоорганізації [50], вибір функції пристосованості генетичного алгоритму). **Коваленко переписала чужий текст з тим самим номером покликання, але під цим номером знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	 Рис.4.1. Функциональная схема МГУА где: Ai – пороговые самоотборы полезной информации; I, II, III – критерии самоотбора. Полученная модель используется для прогноза и последующей коррекции в выходном сигнале ИНС ПНК ЛА. Эволюционные алгоритмы реализуют механизм естественного отбора. В процессе функционирования алгоритма приспособленность моделей-претендентов в целом возрастает. Эволюционные алгоритмы применяются для решения задач, имеющих много локальных минимумов, а их недостатками, при использовании в системах диагностики на борту ЛА, являются повышенные требования к производительности вычислительных средств при реализации на борту ЛА, а также необходимость учёта особенностей проектирования алгоритмов (удачный выбор ансамбля критериев селекции алгоритма самоорганизации [50], выбор функции приспособленности генетического алгоритма и др.).
	С. 192.	**С. 83.**
	Таким чином, розроблений підхід до задачі синтезу системи діагностики вимірювальної апаратури ПС. Технологія синтезу продемонстрована на прикладі створення автоматичної системи діагностики приладів ВК перспективного ПС. Для оцінювання та прогнозування стану приладів використовуються	Таким образом, разработан подход к задаче синтеза системы диагностики измерительной аппаратуры ЛА. Технология синтеза продемонстрирована на примере создания автоматической системы диагностики гироприборов ИК перспективного ЛА. Для оценивания и прогнозирования состоя-

	апріорні моделі, алгоритми ідентифікації та еволюційні алгоритми побудови прогнозуючих моделей. **Коваленко пише «Таким чином», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Плагіат.**	ния гироприборов используются априорные модели, алгоритмы идентификации и эволюционные алгоритмы построения прогнозирующих моделей.
	С. 192.	**С. 84.**
	При використанні подібної структури в системі управління необхідно враховувати деякі особливості рисунок 4.2. Отже, необхідність постійно мати адекватну модель, її доводиться будувати заново або коригувати наявну модель при виникненні суттєвих змін зовнішнього і внутрішнього середовища. Така необхідність пояснюється вимогами швидко реагувати на зміни, зумовлені постійно мінливих умов функціонування. Остання особливість призводить до необхідності тримати ситуацію під постійним контролем, використовувати постійну корекцію і не дозволяє застосовувати комплекс довгострокових заходів, використовувати ефективні сценарії.	**4.2. Разработка системы повышения отказоустойчивости на основе теории функциональных систем.** При использовании подобной структуры в системе управления необходимо учитывать некоторые особенности. Например, необходимость постоянно иметь адекватную модель, ее приходится строить заново или корректировать имеющуюся модель при возникновении ощутимых изменений внешней и внутренней среды. Также при функционировании динамического объекта с подобной системой управления, органы управления должны иметь высокую степень сензитивности. Такая необходимость объясняется требованиями быстро реагировать на изменения, обусловленные постоянно меняющимися условиями функционирования. Последняя особенность приводит к необходимости держать ситуацию под постоянным контролем, использовать постоянную коррекцию и не позволяет применять комплекс долгосрочных мер, использовать эффективные сценарии.
	С. 192–193.	**С. 84.**
	 Рис. 4.2 Функціональна схема циклу управління БРЕО	 Рис.4.2. Функциональная схема цикла управления
	С. 193.	**С. 85.**
	При функціонуванні системи, можливо, цілеспрямовано змінювати її стан за допомогою підсистем управління. Управління динамічним об'єктом здійснюється з будь-якою метою, яка реалізується за допомогою групи різних критеріїв або в формалізованому вигляді як функціонал якості. Шляхом мінімізації або максимізації функціоналу якості визначаються оп-	При функционировании системы, возможно, целенаправленно изменять ее состояние с помощью подсистем управления. Управление динамическим объектом осуществляется с какой-либо целью, которая реализуется посредством ансамбля различных критериев или в формализованном виде как функционал качества. Путем минимизации или максимизации функцио-

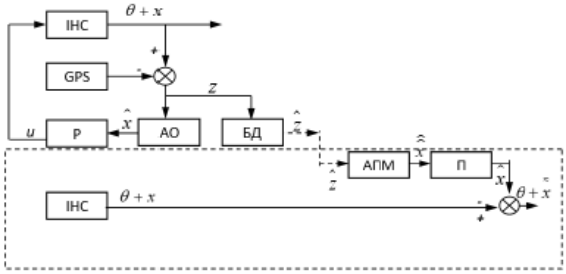
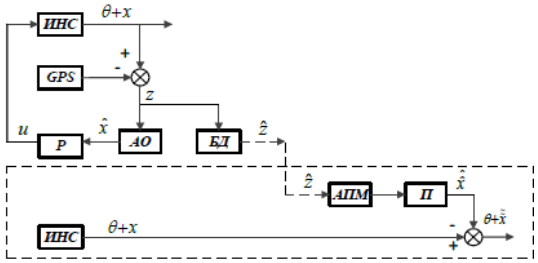
	тимальні параметри об'єкта. **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	нала качества определяются оптимальные параметры объекта. Реализацию системы управления можно осуществить, например, в рамках теории функциональных систем Анохина [5, 59].
	С. 193.	**С. 85.**
	Функціональні системи реалізують принцип саморегуляції [5]. Вони вибірково об'єднують підсистеми і їх елементи з метою досягнення корисних результатів. У функціональній системі присутня зворотна інформація про результати її дії, що забезпечує пристосувальний ефект, який призводить до досягнення мети, яка є системоутворюючим фактором. **Коваленко переписала чужий текст з тим самим номером покликання, але під цим номером знаходиться зовсім інше джерело – ДСТУ (1998).** **Фальсифікація джерел.** **Плагіат.**	П.К. Анохин обратил внимание на то, что системы живых организмов осуществляют избирательное объединение входящих в них элементов для реализации функций организма. Такие системы он назвал функциональными системами. Функциональные системы реализуют принцип саморегуляции [5]. Они избирательно объединяют подсистемы и их элементы с целью достижения полезных результатов. В функциональной системе присутствует обратная информация о результатах ее действия, что обеспечивает приспособительный эффект, который приводит к достижению цели, являющейся системообразующим фактором. **Покликання [5] – це:** Анохин П.К. Биология и нейрофизиология условного рефлекса. М: Медицина, 1968.
	С. 193.	**С. 85–86.**
	Механізми корекції параметрів системи необхідно синтезувати з урахуванням можливостей реалізації саморегуляції. У колі зворотної аферентації основним алгоритмом є алгоритм побудови прогнозуючих моделей, заснований на методі самоорганізації. Прогнозуюча модель має вигляд: $$\Phi(x) = \sum_{i=1}^{N} a_i \, \mu_{ni}(fix), \quad (4.16)$$ де n – число базисних функцій; μ_n – базисні функції з параметризовані множини F_p. $F_p = \{a_i \mu_i(f_i x) \mid i = 1, L\{\}\}$, набір базисних функцій. Кожній базисній функції ставиться у відповідність двомірний вектор параметрів $(a, f)^T$, де a – амплітуда, f – частота, доцільно здійснювати на основі аналізу простих прогнозуючих моделей. Використання модифікацій алгоритму самоорганізації дозволяє підвищити точність прогнозу, а отже, і точність діагностики приладового комплексу ПС. Таким чином, розроблена перспективна система діагностики (управління) приладового комплексу ПС, яка дозволяє здійснювати діагностику протягом усього його життєвого циклу.	<…> Механизмы коррекции параметров системы необходимо синтезировать с учетом возможностей реализации саморегуляции. В цепи обратной афферентации основным алгоритмом является алгоритм построения прогнозирующих моделей, основанный на методе самоорганизации. Прогнозирующая модель имеет вид: $$\Phi(\mathrm{x}) = \sum_{i=1}^{N} a_i \mu_{ni}(f_i \, x), \quad (4.16)$$ где n – число базисных функций; μ_n – базисные функции из параметризованного множества $\mathrm{F_p} = \{\mathrm{a_i}\mu_i(\mathrm{f_i x}) \mid \mathrm{i}=1,\ \mathrm{L}\}$, набор базисных функций. Каждой базисной функции ставится в соответствие двухмерный вектор параметров $(a, f)^T$, где a – амплитуда, f – частота, целесообразно осуществлять на основе анализа простых прогнозирующих моделей. Использование модификаций алгоритма самоорганизации позволяет повысить точность прогноза, а следовательно, и точность диагностики приборного комплекса ЛА. Таким образом, разработана перспективная система диагностики (управления) приборного комплекса ЛА, которая позволяет осуществлять диагностику в течение всего его жизненного цикла.
	С. 212.	**С. 86.**
	4.3. Розробка редукованої динамічної експертної системи для автоматизованого контролю бортового устаткування Системи контролю базуються на аналізі кількісних і альтернативних ознак. При контролі за альтернативною ознакою не потрібно знати фактичне значення контрольованого параметра, а потрібно встановити факт відпо-	**4.3. Редуцированная динамическая экспертная система с интеллектуальной компонентой для контроля прицельно-навигационного комплекса летательного аппарата.** Системы контроля базируются на анализе количественных и альтернативных признаков [29, 52, 72, 88]. При контроле по альтернативному признаку не требуется знать фактическое значение контроли-

	відності або невідповідності цього параметра встановленим вимогам. При експлуатації сучасних ПС необхідно реалізувати систему контролю працездатності авіоніки. **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	руемого параметра, а нужно установить факт соответствия или несоответствия этого параметра установленным требованиям. При эксплуатации современных ЛА необходимо реализовать систему контроля работоспособности авионики.
	С. 212.	**С. 86–87.**
	Для цього застосовуються різноманітні системи контролю на різних етапах експлуатації бортового обладна. Автоматизовані бортові системи контролю, що включають вбудовані засоби інструментального контролю і системи інформаційного контролю, системи автоматичного контролю бортового устаткування і ін. Використовуються ієрархічні системи засобів контролю, які добре зарекомендували себе на практиці, в яких оцінюється працездатність і достовірність інформації окремих систем і комплексу бортового обладнання в цілому. Однак при вирішенні завдання контролю бортового устаткування доцільно знати не тільки момент відмови бортових систем, але і передбачити момент виникнення аварійної ситуації, а також інтервали недостовірної роботи обладнання. **Коваленко не змогла перекласти російське слово «**общесамолетные**», а тому просто його викинула.** **Плагіат.**	Для этого применяются разнообразные системы контроля на различных этапах эксплуатации ПНК. Автоматизированные бортовые системы контроля, включающие встроенные средства инструментального контроля и системы информационного контроля, общесамолетные системы контроля бортового оборудования и др. Используются иерархические системы средств контроля, которые хорошо зарекомендовали себя на практике, в которых оценивается работоспособность и достоверность информации отдельных систем и комплекса бортового оборудования в целом. Однако при решении задачи контроля бортового оборудования целесообразно знать не только момент отказа бортовых систем, но и предвидеть момент возникновение аварийной ситуации, а также интервалы недостоверной работы оборудования.
	С. 212–213.	**С. 87.**
	Вирішення цього завдання за допомогою апріорних прогнозуючих моделей вимагає проведення тривалих дорогих експериментів, не дозволяє враховувати особливості конкретних систем і здійснювати ефективний контроль високоманеврових ПС. Тому для здійснення контролю бортового устаткування перспективних маневрених ПС доцільно використовувати комплексні системи контролю на базі ДЕС, які дозволяють враховувати режими польоту ПС, мають багату базу даних і група оціночних критеріїв. **Знову та сама помилка перекладу: російський вислів «**имеют богатую базу данных и ансамбль оценочных критериев**» Коваленко переклала як «**мають багату базу даних і група оціночних критеріїв**», тоді як треба або «мають … групу оціночних критеріїв», або «мають … ансамбль оціночних критеріїв».** **Смішний плагіат.**	Решение этой задачи с помощью априорных прогнозирующих моделей требует проведения длительных дорогостоящих экспериментов, не позволяет учитывать особенности конкретных систем и осуществлять эффективный контроль высокоманевренных ЛА. Поэтому для осуществления контроля бортового оборудования перспективных маневренных ЛА целесообразно использовать комплексные системы контроля на базе ДЭС, которые позволяют учитывать режимы полета ЛА, имеют богатую базу данных и ансамбль оценочных критериев.
	С. 213.	**С. 87.**
	В сучасних БРЕО для забезпечення високої якості формованої інформації використовується апріорна структурна адаптація, під якою розуміється вибір найкращої конфігурації оцінюваної частини повного вектора стану, а також матриць моделі досліджуваної системи для конкретних умов. Адаптація здійснюється за допомогою коваріаційного аналізу на основі мінімаксного критерію якості.	В современных КОИ ПНК для обеспечения высокого качества формируемой информации используется априорная структурная адаптация, под которой понимается выбор наилучшей конфигурации оцениваемой части полного вектора состояния, а также матриц модели исследуемой системы для конкретных условий [6, 23]. Адаптация осуществляется с помощью ковариационного анализа

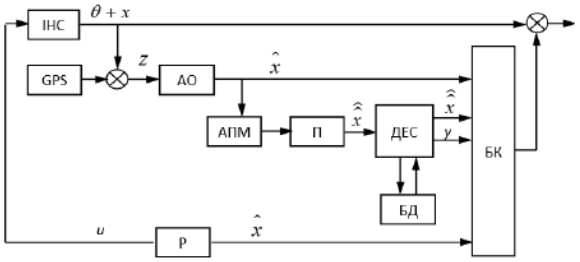
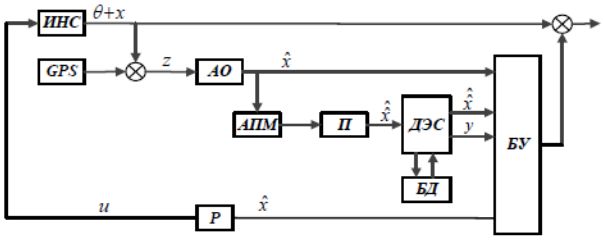
	Коваленко в переписаному чужому тексті видалила покликання. **Плагіат.**	на основе минимаксного критерия качества.
	С. 213.	**С. 88.**
	Для ефективного виділення динамічних складових часто потрібні дорогі і трудомісткі льотні експерименти. Наприклад, для достовірного виділення компонент динамічного дрейфу необхідна велика кількість льотних експериментів. Для використання в системі контролю доцільно формувати редукований вектор стану з використанням концепції системного синтезу [47, 58]. Базуючись на концепції системного синтезу здійснюється скорочення числа параметрів моделі, що характеризують досліджуваний об'єкт, виключаючи їх з вектора стану моделі. З вектора стану моделей виділяються ключові параметри за допомогою загальних і спеціальних критеріїв [60]. Загальні критерії є універсальними, а спеціальні критерії, як правило, визначають якість досліджуваного процесу на основі будь-яких фізичних характеристик, притаманних аналізованого процесу. При функціонуванні ПС відбираються параметри, які мають максимальні ступеня спостережливості і керованості [42], що дозволяє відбирати для здійснення контролю БЦОМ тільки ефективно керовані і достовірно оцінюючі параметри. **Коваленко переписала чужий текст разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела, у тому числі під №60 – її власна стаття 2016-го року.** **У цьому фрагменті її дисертації всі покликання фальшиві.** **Фальсифікація джерел.** **Плагіат.**	Для эффективного выделения динамических составляющих часто требуются дорогостоящие и трудоемкие летные эксперименты. Например, для достоверного выделения компонент динамического дрейфа необходимо большое количество летных экспериментов. Для использования в системе контроля целесообразно формировать редуцированный вектор состояния с использованием концепции системного синтеза [47, 58]. Базируясь на концепции системного синтеза осуществляется сокращение числа параметров модели, характеризующих исследуемый объект, исключая их из вектора состояния модели. Из вектора состояния моделей выделяются ключевые параметры с помощью общих и специальных критериев [60]. Общие критерии являются универсальными, а специальные критерии, как правило, определяют качество исследуемого процесса на основе каких-либо физических характеристик, присущих анализируемому процессу. При функционирования ЛА отбираются параметры, имеющие максимальные степени наблюдаемости и управляемости [42], что позволяет отбирать для осуществления контроля ПНК только эффективно управляемые и достоверно оцениваемые параметры.
	С. 213–214.	**С. 88.**
	В процесі функціонування бортового устаткування, склад редукованого вектора стану моделі може змінюватися. Компоненти редукованого вектора стану, що підлягають контролю, вибираються на основі аналізу якісних характеристик кожної компоненти, які залежать від режиму польоту ПС, зовнішніх збурювань, власного стану БЦОМ, а також можливостей безпосереднього вимірювання за допомогою наявних засобів. Зазвичай в вектор стану для здійснення контролю включають параметри, що характеризують працездатність БРЕО. Але в редукованому векторі стану використовуються тільки частина цього вектора стану – ключові параметри: безпосередньо вимірювані і компоненти, що ефективно спостерігаються.	В процессе функционирования ПНК состав редуцированного вектора состояния модели может изменяться. Компоненты редуцированного вектора состояния, подлежащие контролю, выбираются на основе анализа качественных характеристик каждой компоненты, которые зависят от режима полета ЛА, внешних возмущений, собственного состояния ПНК, а также возможностей непосредственного измерения с помощью имеющихся средств. Обычно в вектор состояния для осуществления контроля включают параметры, характеризующие работоспособность ПНК. Но в редуцированном векторе состояния используются только часть этого вектора состояния - ключевые параметры: непосредственно измеряемые и эффективно наблюдаемые компоненты.
	С. 214.	**С. 89.**
	Для здійснення всеохоплюючого контролю бортового устаткування доцільно використовувати	**4.4. Разработка редуцированной динамической экспертной системы для контроля бортового оборудования.** Для осуществления всеобъемлющего контроля бортового оборудования целесообразно использо-

	концепції, які закладені в ДЕС і інтелектуальні технології. В якості інтелектуальної технології використовуються алгоритми акцептора дії. Реалізація функції контролю в ДЕС передбачає її модифікацію з метою зменшення обсягу пам'яті використовуємо структуру системи управління з інтелектуальною компонентою, побудована на основі теорії функціональних систем, передбачає побудову прогнозуючих моделей стану об'єкта управління і моделей зовнішнього середовища. **Коваленко в переписаному чужому тексті видалила покликання.** **Плагіат.**	вать концепции, заложенные в ДЭС и интеллектуальные технологии. В качестве интеллектуальной технологии используются алгоритмы акцептора действия. Реализация функции контроля в ДЭС предполагает ее модификацию с целью уменьшения объема памяти БЦВМ. Известна структура системы управления с интеллектуальной компонентой, построенная на основе теории функциональных систем, предусматривает построение прогнозирующих моделей состояния объекта управления и моделей внешней среды [7, 9, 48, 55].
	С. 214.	**С. 89.**
	Побудувавши моделі зовнішнього середовища функціонування бортового обладнання, з'являється можливість розробляти сценарії управління на тривалі проміжки часу його функціонування. Сценарії управління припускають реконфігурування обладнання ПС. Функціональна схема системи контролю БРЕО ПС представлена на рисунку 4.3.  Рис. 4.3. Модель функціональна схема системи контролю з урахуванням моделей зовнішнього середовища	Построив модели внешней среды функционирования ПНК ЛА, появляется возможность разрабатывать сценарии управления на длительные интервалы времени его функционирования. Сценарии управления предполагают реконфигурирование оборудования ЛА. Функциональная схема системы контроля ПНК ЛА представлена на Рис.4.3.  Рис.4.3. Функциональная схема системы контроля с учетом моделей внешней среды
	С. 215.	**С. 90.**
	На рис. 4.3 в блоці прогнозу здійснюється прогнозування ключових змінних стану БРЕО ПС з урахуванням різних моделей зовнішнього середовища. Далі прогноз стану БЦОМ порівнюється з критичними значеннями його стану, які визначаються в залежності від режиму польоту ПС. Функціональна структура рисунку 4.3 використана при синтезі редукованої ДЕС ПС. Структура редукованої ДЕС з інтелектуальною компонентою представлена на рисунку 4.4.	На Рис. 4.3 в блоке прогноза осуществляется прогнозирование ключевых переменных состояния ПНК ЛА с учетом различных моделей внешней среды. Далее прогноз состояния пнк сравнивается с критическими значениями его состояния, которые определяются в зависимости от режима полета ЛА. Функциональная структура Рис. 4.3 использована при синтезе редуцированной ДЭС ЛА. Структура редуцированной ДЭС с интеллектуальной компонентой представлена на Рис.4.4.
	С. 215.	**С. 90.**
	 Рис. 4.4: Структура редукованої ДЕС з інтелектуальною компонентою СУ – система управління; П – алгоритм прогнозу; БД – база даних; БЗ – база знань; w - вектор зовнішніх збурювань; u – вектор управління; z – вектор вимірювань; x оцінка редукованого вектора стану	 Рис.4.4. Структура редуцированной ДЭС с интеллектуальной компонентой На Рис.4.4 обозначено: СУ - система управления; П - алгоритм прогноза; БД - база данных; БЗ - база знаний; w - вектор внешних возмущений; u – вектор управления; z - вектор измерений; $\hat{x}$ - оцен-

	БРЕО; x- скорочений вектор прогнозних значень стану БРЕО.	ка редуцированного вектора состояния ПНК; $\hat{\hat{x}}$ - редуцированный вектор прогнозных значений состояния ПНК.
	С. 215.	**С. 90–91.**
	ДЕС ПС працює на основі інформації БД і БЗ. Інформація про результати функціонування БРЕО ПС надходить в БД, де виконується початкова обробка даних для забезпечення ефективної побудови моделей, виявлення і відсіювання аномальних вимірювань, формування вимірювань в вибірках та ін. В БД інформація зберігається в упорядкованій формі в двох частинах пам'яті - довгострокової і короткострокової пам'яті. У довгостроковій пам'яті зберігаються шаблони, тобто сукупність даних, відповідно чітко обумовленому режиму функціонування ПС і прогнозуючі моделі, які використовуються для конкретних умов польоту. У короткостроковій пам'яті зберігаються всі динамічні дані, які оперативно оновлюються під час вступу нових вимірів.	ДЭС ЛА работает на основе информации БД и БЗ. Информация о результатах функционирования ПНК ЛА поступает в БД, где выполняется первоначальная обработка данных для обеспечения эффективного построения моделей, выявление и отсеивание аномальных измерений, формирование измерений в выборках и т.д. В БД информация сохраняется в упорядоченной форме в двух частях памяти - долгосрочной и краткосрочной памяти. В долгосрочной памяти сохраняются шаблоны, т.е. совокупность данных, соответственно четко обусловленному режиму функционирования ЛА и прогнозирующие модели, используемые для конкретных условий полета. В краткосрочной памяти сохраняются все динамические данные, которые оперативно обновляются при поступлении новых измерений.
	С. 215–216.	**С. 91.**
	Попередньо оброблені в БРЕО вибірки надходять в акцептор дії, де відбуваються оцінювання ключових параметрів, побудова прогнозуючих моделей і прогноз його стану, а також звірення результатів прогнозу з реальними результатами дії. Оцінювання ключових параметрів БРЕО ПС здійснюється за допомогою алгоритму оцінювання, наприклад фільтра Калмана. Для отримання нерозходжуючихся оцінок в практичних додатках використовують різні адаптивні модифікації фільтра Калмана. В якості алгоритму побудови прогнозуючих моделей використовуються нейронні мережі, алгоритми самоорганізації і генетичні алгоритми [26, 50, 115]. Отримані прогнозуючі моделі для даної конкретної ситуації використовуються для отримання прогнозних значень ключових параметрів БРЕО. Результати прогнозу надходять в БЗ, де порівнюються зі значеннями, встановленими для виконуваного режиму функціонування ПС. Також прогнозними значеннями ключових параметрів БРЕО поповнюються БД ДЕС. З ДЕС інформація про майбутні порушеннях функціонування КБО передається для формування керуючого впливу на КБО. **Коваленко в переписаному чужому тексті видалила покликання [48, 93, 96, 113].** **Коваленко переписала чужий текст разом із номерами покликань [26, 50, 115], під якими в її дисертації знаходяться зовсім інші джерела, у тому числі під №115 – її власна стаття 2024-го року.** **У цьому фрагменті її дисертації всі покликання фальшиві.** **Фальсифікація джерел.** **Плагіат.**	Предварительно обработанные в ПНК выборки поступают в акцептор действия, где происходят оценивание ключевых параметров ПНК, построение прогнозирующих моделей и прогноз его состояния, а также сличение результатов прогноза с реальными результатами действия. Оценивание ключевых параметров ПНК ЛА осуществляется с помощью алгоритма оценивания, например фильтра калмана. Для получения нерасходящихся оценок в практических приложениях используют различные адаптивные модификации фильтра калмана [48, 93, 96, 113]. В качестве алгоритма построения прогнозирующих моделей используются нейронные сети, алгоритмы самоорганизации и генетические алгоритмы [26, 50, 115]. Полученные прогнозирующие модели для данной конкретной ситуации используются для получения прогнозных значений ключевых параметров ПНК. Результаты прогноза поступают в БЗ, где сравниваются со значениями, установленными для выполняемого режима функционирования ЛА. Также прогнозными значениями ключевых параметров ПНК пополняются БД ДЭС. С ДЭС информация о будущих нарушениях функционирования ПНК передается в СУ для формирования управляющего воздействия на ПНК.
	С. 216.	**С. 91–92.**
	Як у випадку плавного виходу параметрів за	**4.5. Разработка ДЭС с функцией восстановления.** Как в случае плавного выхода параметров за

	межі допуску, так і в разі різких збоїв відновлення навігаційного комплексу ПС проводиться шляхом підстроювання коефіцієнтів або параметрів алгоритмічного забезпечення КБО. Виявлення неприйнятного рівня похибок здійснено системою контролю ПС з використанням алгоритму самоорганізації. Відновлення працездатності системи проведено за допомогою регулятора в структурі ІНС або прогнозу і компенсації похибок на виході ІНС. Вимірювальний комплекс з функцією регенерації представлений на рисунку 4.5.	границы допуска, так и в случае резких сбоев восстановление навигационного комплекса ЛА проводится путем подстройки коэффициентов или параметров алгоритмического обеспечения ПНК. Выявление неприемлемого уровня погрешностей осуществлено системой контроля ЛА с использованием алгоритма самоорганизации. Восстановление работоспособности системы проведено с помощью регулятора в структуре ИНС или прогноза и компенсации погрешностей на выходе ИНС. Измерительный комплекс с функцией регенерации представлен на Рис.4.5.
	С. 216–217.	**С. 92.**
	 Рис. 4.5. Вимірювальний комплекс з функцією регенерації θ – справжня навігаційна інформація; x – похибки ІНС; z – вектор вимірювань; $\hat{z}$ – вимірювальна вибірка, АО – алгоритм оцінювання; АПМ – алгоритм побудови моделей; БД – база даних; П – алгоритм прогнозу, Р – регулятор; $\hat{x}$ – оцінка вектора похибок ІНС; $\hat{\hat{x}}$ – прогноз похибок ІНС; x̃ – помилка прогнозування; u – вектор управління.	 Рис.4.5. Измерительный комплекс с функцией регенерации На Рис.4.5 обозначено: θ – истинная навигационная информация; x – погрешности ИНС; ẑ - измерительная выборка; П - алгоритм прогноза, Р - регулятор; $\tilde{\hat{X}}$ - ошибка прогнозирования.
	С. 217.	**С. 92.**
	Для діагностики нелінійних моделей КБО ПС використаний алгоритм побудови прогнозуючої моделі похибок з використанням ГА або алгоритму самоорганізації. В сучасних КІО КБО для забезпечення високої якості формованої інформації використовується апріорна структурна адаптація, під якою розуміється вибір найкращої конфігурації оцінюваної частини повного вектора стану, а також матриць моделі досліджуваної системи для конкретних умов [9]. **Коваленко переписала чужий текст разом із номером покликання, під якими в її дисертації знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	Для диагностики нелинейных моделей ПНК ЛА использован алгоритм построения прогнозирующей модели погрешностей с использованием ГА или алгоритма самоорганизации. В современных КОИ ПНК для обеспечения высокого качества формируемой информации используется априорная структурная адаптация, под которой понимается выбор наилучшей конфигурации оцениваемой части полного вектора состояния, а также матриц модели исследуемой системы для конкретных условий [9].
	С. 217.	**С. 92–93.**
	Адаптація здійснюється за допомогою коваріаційного аналізу на основі критерію якості мінімального та максимального. Для ефективного виділення динамічних складових вектора стану часто потрібні дорогі і трудомісткі льотні експерименти. Наприклад, для достовірного виділення компонент динамічного дрейфу необхідна велика кількість льотних експериментів. Для використання в системі контролю доцільно формувати редукований вектор стану з використанням концепції системного синтезу. У відповідність з цією концепцією необхідно раціональне скорочення числа параметрів моделі, що характеризують досліджуваний об'єкт. Здійснити таке скоро-	Адаптация осуществляется с помощью ковариационного анализа на основе минимаксного критерия качества. Для эффективного выделения динамических составляющих вектора состояния часто требуются дорогостоящие и трудоемкие летные эксперименты. Например, для достоверного выделения компонент динамического дрейфа необходимо большое число летных экспериментов. Для использования в системе контроля целесообразно формировать редуцированный вектор состояния с использованием концепции системного синтеза [10]. В соответствие с этой концепцией необходимо рациональное сокращение числа параметров модели, характеризующих исследуемый

	чення можна шляхом використання різних критеріїв. У векторі стану моделей виділяються ключові параметри за допомогою загальних і спеціальних критеріїв. Загальні критерії є універсальними, а спеціальні критерії, як правило, визначають якість досліджуваного процесу на основі будь-яких фізичних характеристик, притаманних аналізованого процесу. **Коваленко в переписаному чужому тексті видалила всі покликання.** **Плагіат.**	объект. Осуществить такое сокращение можно путем использования различных критериев [11]. В векторе состояния моделей выделяются ключевые параметры с помощью общих и специальных критериев [12]. Общие критерии являются универсальными, а специальные критерии, как правило, определяют качество исследуемого процесса на основе каких-либо физических характеристик, присущих анализируемому процессу.
	С. 217–218.	**С. 93.**
	Потім, в залежності від режиму польоту ПС, формуються різні моделі. Наприклад, при вчиненні інтенсивного маневрування ПС вибирають параметри, які мають максимальні ступені спостережливості і керованості [13, 91], що дозволяє відбирати для здійснення контролю КБО тільки ефективно керовані і достовірно оцінюються параметри. **Коваленко переписала чужий текст разом із номерами покликань [13, 91], під якими в її дисертації знаходяться зовсім інші джерела.** **Фальсифікація джерел.** **Плагіат.**	Затем, в зависимости от режима полета ЛА, формируются различные модели. Например, при совершении интенсивного маневрирования ЛА выбирают параметры, имеющие максимальные степени наблюдаемости и управляемости [13, 91], что позволяет отбирать для осуществления контроля ПНК только эффективно управляемые и достоверно оцениваемые параметры.
	С. 218.	**С. 93.**
	В процесі функціонування КБО склад редукованого вектора стану моделі може змінюватися. Компоненти цього вектора, що підлягають контролю, вибираються на основі аналізу якісних характеристик кожної компоненти, які залежать від режиму польоту ПС, зовнішніх збурень, власного стану КБО, а також можливостей безпосереднього вимірювання за допомогою наявних засобів. Зазвичай в вектор стану для здійснення контролю включають параметри, що характеризують роботоздатність КБО. Але в редукованому векторі стану використовуються тільки частини цього вектора стану – ключові параметри: безпосередньо вимірювані і компоненти, що ефективно спостерігаються . **Коваленко в переписаному чужому тексті видалила покликання.** **Плагіат.**	В процессе функционирования ПНК состав редуцированного вектора состояния модели может изменяться. Компоненты этого вектора, подлежащие контролю, выбираются на основе анализа качественных характеристик каждой компоненты, которые зависят от режима полета ЛА, внешних возмущений, собственного состояния ПНК, а также возможностей непосредственного измерения с помощью имеющихся средств. Обычно в вектор состояния для осуществления контроля включают параметры, характеризующие работо-способность ПНК. Но в редуцированном векторе состояния используются только часть этого вектора состояния - ключевые параметры: непосредственно измеряемые и эффективно наблюдаемые компоненты [14].
	С. 218.	**С. 93–94.**
	ДЕС ПС працює на основі інформації бази даних і бази знань. Інформація про результати функціонування КФС ПС надходить в БД, де виконується початкова обробка даних для забезпечення ефективної побудови моделей, виявлення і відсіювання аномальних вимірювань, формування вимірювань в вибірках та ін. В БД інформація зберігається в упорядкованій формі в двох частинах пам'яті – довгострокової і короткострокової. У довгостроковій пам'яті зберігаються шаблони, тобто сукупність даних, відповідна чітко обумовленому режиму функціонування ПС, і прогнозуючі моделі, що використовуються для конкретних умов польоту. У короткостроковій пам'яті зберігаються всі динамічні дані, які оперативно оновлюються під час вступу нових вимірів. Попередньо оброблені в КБО вибірки надходять	ДЭС ЛА работает на основе информации базы данных (БД) и базы знаний. Информация о результатах функционирования ПНК ЛА поступает в БД, где выполняется первоначальная обработка данных для обеспечения эффективного построения моделей, выявление и отсеивание аномальных измерений, формирование измерений в выборках и т.д. В БД информация сохраняется в упорядоченной форме в двух частях памяти - долгосрочной и краткосрочной. В долгосрочной памяти сохраняются шаблоны, т.е. совокупность данных, соответствующая четко обусловленному режиму функционирования ЛА, и прогнозирующие модели, используемые для конкретных условий полета. В краткосрочной памяти сохраняются все динамические данные, которые оперативно обновляются при по-

	в акцептор дії, де відбуваються оцінювання ключових параметрів КБО, побудова прогнозують моделей і прогноз його стану, а також звірення результатів прогнозу з реальними результатами дії.	ступлении новых измерений. Предварительно обработанные в ПНК выборки поступают в акцептор действия, где происходят оценивание ключевых параметров ПНК, построение прогнозирующих моделей и прогноз его состояния, а также сличение результатов прогноза с реальными результатами действия.
	С. 218–219.	**С. 94.**
	Отримані прогнозуючі моделі використовуються для отримання прогнозних значень ключових параметрів КБО. Результати прогнозу надходять в БЗ, де порівнюються зі значеннями, встановленими для виконуваного режиму функціонування ПС. Також прогнозними значеннями ключових параметрів КБО поповнюються БД ДЕС [21, 34]. З ДЕС інформація про майбутні порушення функціонування КБО передається в СУ для формування керуючого впливу на КБО. **Коваленко переписала чужий текст разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела.** **Фальсифікація джерел.** **Плагіат.**	Полученные прогнозирующие модели используются для получения прогнозных значений ключевых параметров ПНК. Результаты прогноза поступают в БЗ, где сравниваются со значениями, установленными для выполняемого режима функционирования ЛА. Также прогнозными значениями ключевых параметров ПНК пополняются БД ДЭС [21, 34]. С ДЭС информация о будущих нарушениях функционирования ПНК передается в СУ для формирования управляющего воздействия на ПНК.
	С. 219.	**С. 94–95.**
	Запропоновано систему контролю і діагностики КБО з функцією відновлення , рисунок 4.6.  Рис. 4.6 Система контролю і діагностики КБО з функцією відновлення БУ – блок управління корекцією; у – керуючий сигнал ДЕС. Діагностуються перевищення порогових значень похибок ІНС, парирування яких дозволяє утримати ІНС в області достовірних значень навігаційної інформації ПС, запобігши розвиток ситуації, що призводить до втрати працездатності КБО. При реалізації системи контролю використана структура ДЕС і FDI-метод, який передбачає використання нечіткої логіки. Передбачається використання даних з БД випробувань КБО, які надходять на вхід математичної моделі, а потім вихідні вектори параметрів моделі порівняти з даними випробувань. Одержуваний вектор відхилень піддається аналізу базою нечітких правил ДЕС, а в результаті формується діагностичне рішення про стан КБО. Елементи нечіткої логіки застосовуються спільно з методом діагностичних матриць (матриць Л. А. Урбана).	Предложена система контроля и диагностики ПНК с функцией восстановления (Рис. 4.6).  Рис.4.6. Система контроля и диагностики ПНК с функцией восстановления На Рис.4.6 обозначено: БУ - блок управления коррекцией; у - управляющий сигнал ДЭС. Диагностируются превышения пороговых значений погрешностей ИНС, парирование которых позволяет удержать ИНС в области достоверных значений навигационной информации ЛА, предотвратив развитие ситуации, приводящей к потере работоспособности ПНК. При реализации системы контроля использована структура ДЭС и FDI-метод, предполагающий использование нечеткой логики. Предполагается использование данных из БД испытаний ПНК, которые поступают на вход математической модели, а потом выходные векторы параметров модели сравнить с данными испытаний. Получаемый вектор отклонений подвергается анализу базой нечетких правил ДЭС, а в результате формируется диагностическое решение о состоянии ПНК. Элементы нечеткой логики применяются совместно с методом диагностических матриц (матриц Л. А. Урбана).
	С. 219–220.	**С. 95.**
	Моделі в векторній формі мають вигляд: $\delta x = A \cdot 1 \cdot B \cdot \delta y,$ (4.65) де δх – вектор стану КБО; δу – вектор діагностичних ознак вимірюваних параметрів КБО; В і А – матриці коефіцієнтів, які встановлюють кількісний взаємозв'язок параметрів на виділених режимах	Модели в векторной форме имеют вид: $\Delta x = A^{-1} \cdot B \cdot \delta y,$ (4.17) где δх - вектор состояния ПНК; δу - вектор диагностических признаков измеряемых параметров ПНК; В и А - матрицы коэффициентов, устанавливающие количественную взаимосвязь параметров

	функціонування КБО. Діагностична матриця конкретного КБО представляє собою таблицю чисельних значень і, що дозволяють на основі відхилень частини вимірювальних параметрів визначати відхилення безпосередньо не вимірюваних параметрів стану. Створення бази нечітких правил здійснюється на основі діагностичної матриці, рядки якої лягли в основу створення БЗ і функцій приналежності відповідних лінгвістичних змінних. **Коваленко скопіювала формулу з помилкою, а також скопіювала текст без позначень вимірювальних параметрів.** **Недолугий плагіат.**	на выделенных режимах функционирования ПНК. Диагностическая матрица конкретного ПНК представляет собой таблицу численных значений a_{ij} и b_{ij}, позволяющих на основе отклонений части измеряемых параметров δy_{ij} определять отклонения непосредственно не измеряемых параметров состояния δx_{ij}. Создание базы нечетких правил осуществляется на основе диагностической матрицы, строки которой легли в основу создания БЗ и функций принадлежности соответствующих лингвистических переменных.
	С. 236–237.	**С. 59.**
	Процес контролю поетапний. Він включає в себе сприйняття значень контрольованих параметрів (отримання інформації про стан організації), зіставлення значень контрольованих параметрів з допусками і видачу результатів зіставлення. Процес контролю передбачає наявність об'єкта контролю, з одного боку, і засоби контролю, з іншого, будемо розуміти під об'єктом контролю будь-який тип об'єкта, інформацію про відповідність стану якого пропонованим вимогам належить отримати, а під засобами контролю – будь-які спеціалізовані ресурси і технічні засоби, що служать для отримання і перетворення цієї інформації. Система контролю - сукупність об'єкта і засоби контролю.	**3.1. Модели процесса контроля.** Процесс контроля поэтапный. Он включает в себя восприятие значений контролируемых параметров (получение информации о состоянии организации), сопоставление значений контролируемых параметров с допусками и выдачу результатов сопоставления (логическая обработка полученной информации). Процесс контроля предполагает наличие объекта контроля, с одной стороны, и средства контроля, с другой, будем понимать под объектом контроля любой тип объекта, информацию о соответствии состояния которого предъявляемым требованиям надлежит получить, а под средствами контроля – любые специализированные ресурсы и технические средства, служащие для получения и преобразования этой информации. Система контроля – совокупность объекта и средства контроля.
	С. 237.	**С. 59.**
	При розгляді процесів контролю, а також при проектуванні засобів - систем контролю виникають наступні завдання: 1. передбачення очікуваного результату процесу контролю в умовах, що включають елемент випадковості; 2. синтез оптимальних засобів контролю, тобто вибір оптимальних з точки зору результату процесу контролю параметрів і елементів застосовуваних засобів; 3. оптимальна організація процесу і системи контролю.	При изучении процессов контроля, а также при проектировании средств – систем контроля возникают следующие задачи: 1) предсказание ожидаемого результата процесса контроля в условиях, включающих элемент случайности; 2) синтез оптимальных средств контроля, т.е. выбор оптимальных с точки зрения результата процесса контроля параметров и элементов применяемых средств; 3) оптимальная организация процесса и системы контроля.
	С. 237.	**С. 59–60.**
	Функціонування систем контролю носить стохастичний характер з огляду на стохастичних властивостей об'єктів контролю (в іншому випадку відпала б необхідність контролю) та обмежених точності і надійності засобів контролю. Тому при формалізації опису функціонування систем контролю для оцінки їх ефективності слід використовувати математичний апарат теорії ймовірностей і теорії оптимального управління. Таким чином, рішення задач, пов'язаних з дослідженням ефективності систем контролю, передбачає вирішення таких питань: вибору показників ефективності; побудови математичної моделі контролю; дослідження корисності результатів виконання операцій; кількісної оцінки показників ефек-	Функционирование систем контроля носит стохастический характер ввиду стохастических свойств объектов контроля (в противном случае отпала бы необходимость контроля) и ограниченных точности и надежности средств контроля. Поэтому при формализации описания функционирования систем контроля для оценки их эффективности следует использовать математический аппарат теории вероятностей и теории оптимального управления. Таким образом, решение задач, связанных с исследованием эффективности систем контроля, предполагает решение следующих вопросов: выбора показателей эффектив-ности; построения математической модели контроля; исследования полезности результатов выполнения

	тивності системи контролю, рисунок 4.12.	операций; количественной оценки показателей эффективности системы контроля.
	С. 238.	**С. 60.**
	Отже, один із ключових аспектів у дослідженні операцій полягає у виборі такого показника ефективності, який би виявляв високу чутливість до параметрів операції, раціональні значення яких необхідно визначити, або які справляють найбільш істотний вплив на досягнення позитивного результату. Водночас цей показник має бути достатньо простим для забезпечення його зручного обчислення та подальшого аналітичного опрацювання	<…> При исследовании операций одной из основных задач является выбор такого показателя эффективности операций, который был бы достаточно чувствителен к тем параметрам операции, рациональные значения которых следует определить (задачи синтеза) или которые оказывают наиболее существенное влияние на успех или неуспех операции (задачи анализа). Вместе с тем показатель эффективности должен быть достаточно прост, чтобы его можно было сравнительно легко вычислить и проанализировать.
	С. 238–239.	**С. 100.**
	ІНС використовуються в складі КБО. Корекція ІНС в вихідному сигналі проводиться за допомогою алгоритмів оцінювання та прогнозу. Працездатність алгоритмів перевіряється за допомогою тестової моделі похибок ІНС [37]. $x_k = \Phi x_{k-1} = W_{k-1}$, (91) де: $x_k = \begin{bmatrix} \delta V_k \\ \varphi_k \\ \varepsilon_k \end{bmatrix}$; $\Phi = \begin{bmatrix} 1 & -gT & 0 \\ \frac{T}{R} & 1 & T \\ 0 & 0 & 1-\beta T \end{bmatrix}$; $W_{k-1} = \begin{bmatrix} B \\ 0 \\ \omega_{k-1} \end{bmatrix}$. Тут – похибки ІНС у визначенні швидкості, φ_k – кути відхилення ГСП від площини горизонту, ε_k – швидкість дрейфу ГСП; g – прискорення вільного падіння; B – зміщення нуля акселерометра; R – радіус Землі; T – період дискретизації; β – середня частота випадкового зміни дрейфу; W_{k-1}- вектор зовнішніх збурень, що представляє собою дискретний аналог білого гаусового шуму. **Коваленко в переписаному чужому тексті забула скопіювати похибки ІНС.** **Плагіат.**	**5.2. Результаты математического моделирования.** ИНС используются в составе ПНК. Коррекция ИНС в выходном сигнале проводится с помощью алгоритмов оценивания и прогноза. Работоспособность алгоритмов проверяется с помощью тестовой модели погрешностей ИНС [37]. $x_k = \Phi x_{k-1} = W_{k-1}$ (5.1) где: $x_k = \begin{bmatrix} \delta V_k \\ \varphi_k \\ \varepsilon_k \end{bmatrix}$; $\Phi = \begin{bmatrix} 1 & -gT & 0 \\ \frac{T}{R} & 1 & T \\ 0 & 0 & 1-\beta T \end{bmatrix}$; $W_{k-1} = \begin{bmatrix} B \\ 0 \\ \omega_{k-1} \end{bmatrix}$. Здесь δV_k – погрешности ИНС в определении скорости, φ_k - углы отклонения ГСП от плоскости горизонта, ε_k - скорость дрейфа ГСП; g - ускорение свободного падения; B - смещение нуля акселерометра; R - радиус Земли; T - период дискретизации; β - средняя частота случайного изменения дрейфа; W_{k-1} – вектор внешних возмущений, представляющий собой дискретный аналог белого гауссового шума.
	С. 239.	**С. 100–101.**
	Представлені результати математичного моделювання помилок ІНС, фільтра Калмана, алгоритму прогнозу. Для здійснення прогнозу похибок ІНС побудова моделі проводиться за допомогою алгоритму самоорганізації і ГА. Так як в результаті проведеного моделювання за допомогою цих алгоритмів отримані близькі по точності результати, представлені результати роботи тільки одного алгоритму. На рис. 4.13 – 4.15 представлені результати роботи адаптивного фільтра Калмана.	Представлены результаты математического моделирования ошибок ИНС, фильтра Калмана, алгоритма прогноза. Для осуществления прогноза погрешностей ИНС построение модели проводится с помощью алгоритма самоорганизации и ГА. Так как в результате проведенного моделирования с помощью этих алгоритмов получены близкие по точности результаты, представлены результаты работы только одного алгоритма. На Рис.5.2-5.4 представлены результаты работы адапивного фильтра калмана.
	С. 239.	**С. 101.**

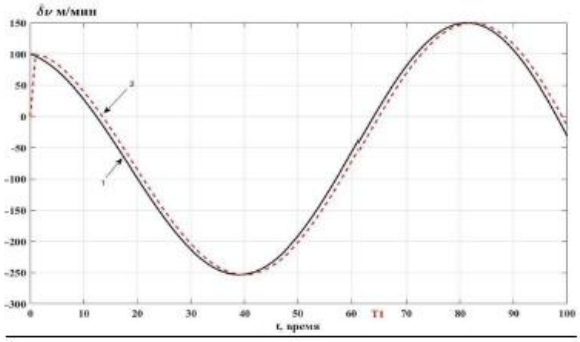


Рис. 4.13. Похибка ІНС у визначенні швидкості і її оцінка

1 – модель помилок ІНС у визначенні швидкості; 2 – оцінка швидкості за допомогою фільтра Калмана.

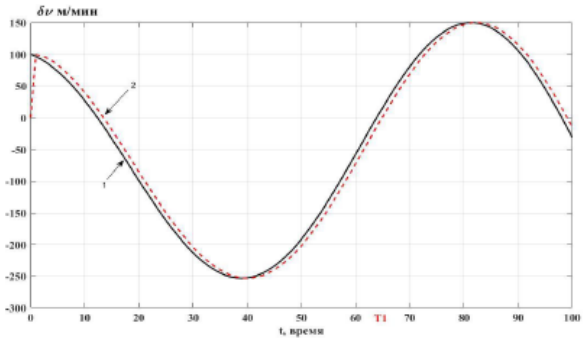


Рис.5.2. Погрешность ИНС в определении скорости и ее оценка

На Рис.5.2 обозначено: 1 - модель ошибок ИНС в определении скорости; 2 - оценка скорости с помощью фильтра калмана.

С. 240. | **С. 101.**

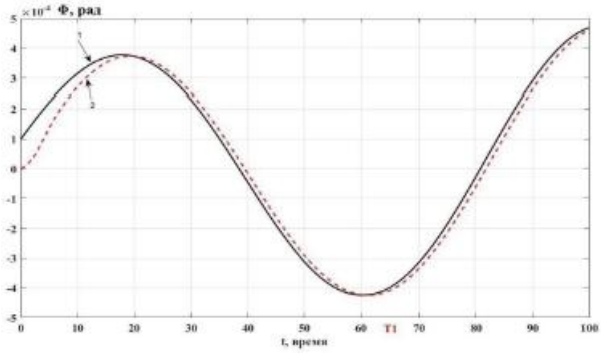


Рис. 4.14. Кут відхилення ГСП і його оцінка

1 – модель помилок ІНС у визначенні кута відхилення ГСП; 2 – оцінка кута відхилення ГСП за допомогою фільтра Калмана.

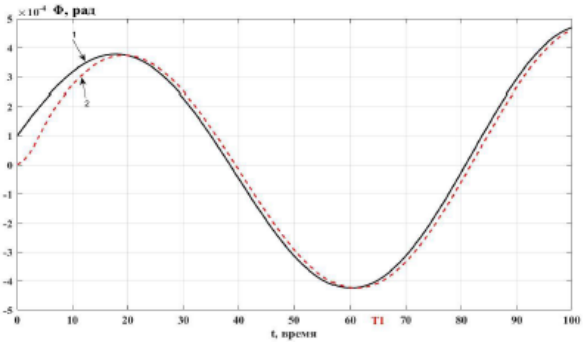


Рис.5.3. Угол отклонения ГСП и его оценка

На Рис.5.3 обозначено: 1 - модель ошибок ИНС в определении угла отклонения ГСП; 2 - оценка угла отклонения ГСП с помощью фильтра калмана.

С. 240. | **С. 102.**

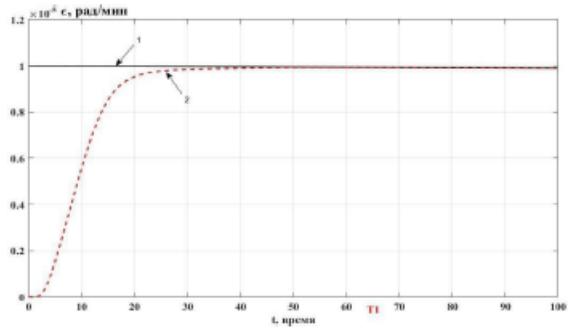


Рис. 4.15. Швидкість дрейфу ГСП і її оцінка

1 – модель помилок ІНС у визначенні дрейфу; 2 – оцінка дрейфу за допомогою фільтра Калмана.

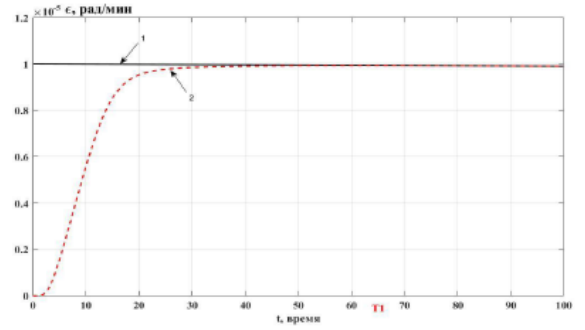


Рис.5.4. Скорость дрейфа ГСП и ее оценка

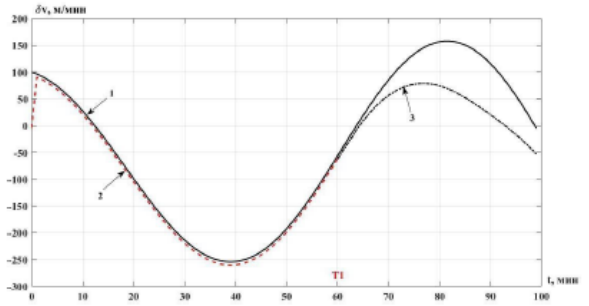
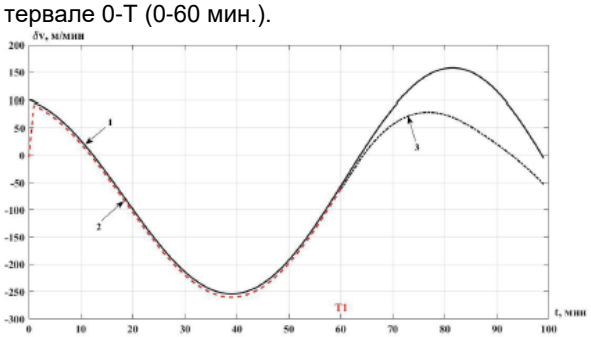
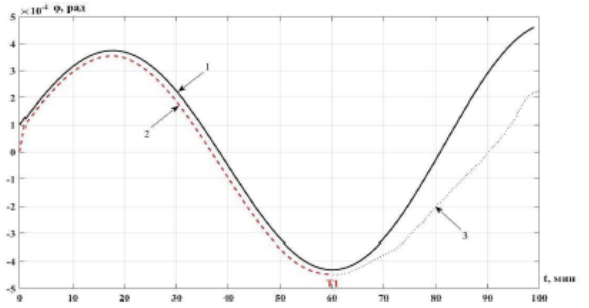
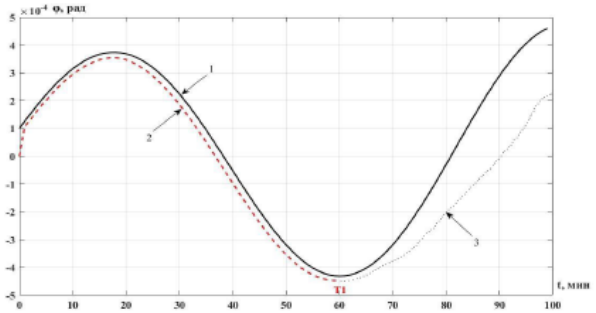
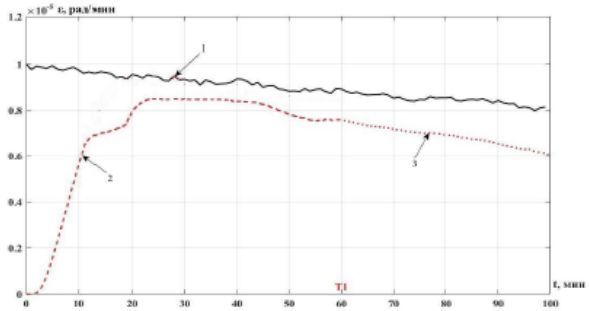
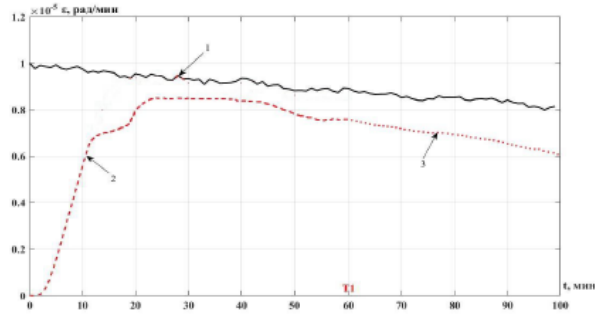
На Рис.5.4 обозначено: 1 - модель ошибок ИНС в определении дрейфа; 2 – оценка дрейфа с помощью фильтра калмана.

С. 240–241. | **С. 102–103.**

У фільтрі Калмана використана апріорна модель похибок ІНС в формі . Застосовувати цю модель для прогнозу не представляється можливим, тому що в практичних додатках характер зміни похибок істотно змінюється, особливо при здійсненні ПС маневрів. Тому для здійснення прогнозу доцільно будувати модель в процесі польоту по наявній вимірювальній вибірці або вибірці оцінок похибок ІНС. На рис. 4.16 – 4.18 представлені результати моделювання похибок ІНС і прогноз по вибірці оцінок. Вибірка оцінок сформована на інтервалі 0-Т (0-60 хв.).

В фильтре калмана использована априорная модель погрешностей ИНС в форме (5.1). Применять эту модель для прогноза не представляется возможным, так как в практических приложениях характер изменения погрешностей существенно меняется, особенно при совершении ЛА маневров. Поэтому для осуществления прогноза целесообразно строить модель в процессе полета по имеющейся измерительной выборке или выборке оценок погрешностей ИНС.

На Рис.5.5-5.7 представлены результаты моделирования погрешностей ИНС и прогноз по выборке оценок. Выборка оценок сформирована на ин-

	 4.16. Похибка ІНС у визначенні швидкості і її прогноз 1 – модель помилок ІНС у визначенні швидкості; 2 – оцінка швидкості за допомогою ГА; 3 – прогноз за допомогою алгоритму ГА.	тервале 0-Т (0-60 мин.).  Рис.5.5. Погрешность ИНС в определении скорости и ее прогноз На Рис.5.5 обозначено: 1 - модель ошибок ИНС в определении скорости; 2 - оценка скорости с помощью ГА; 3 - Прогноз прогноз с помощью алгоритма ГА.
	С. 241.	**С. 103.**
	 Рис. 4.17. Кут відхилення ГСП і його прогноз 1 – модель помилок ІНС у визначенні кута відхилення ГСП; 2 – оцінка кута відхилення ГСП за допомогою ГА; 3 – прогноз за допомогою алгоритму ГА.	 Рис.5.6. Угол отклонения ГСП и его прогноз На Рис.5.6 обозначено: 1 - модель ошибок ИНС в определении угла отклонения ГСП; 2 - оценка угла отклонения ГСП с помощью ГА; 3 - прогноз с помощью алгоритма ГА.
	С. 241–242.	**С. 103.**
	 Рис. 4.18. Швидкість дрейфу ГСП і її прогноз 1 – модель помилок ІНС у визначенні швидкості дрейфу ГСП; 2 – оцінка швидкості дрейфу ГСП за допомогою ГА; 3 – прогноз за допомогою алгоритму ГА.	 Рис.5.7. Скорость дрейфа ГСП и ее прогноз На Рис.5.7 обозначено: 1 - модель ошибок ИНС в определении скорости дрейфа ГСП; 2 - оценка скорости дрейфа ГСП с помощью ГА; 3 - прогноз прогноз с помощью алгоритма ГА.
	С. 242.	**С. 104.**
	На рис. 4.19 – 4.21 представлені результати моделювання похибок ІНС і модель ГА, побудована на основі вимірювальної вибірки z. З 60 хв. здійснений короткостроковий прогноз.	На Рис.5.8-5.10 представлены результаты моделирования погрешностей ИНС и модель ГА, построенная на основе измерительной выборки z. С 60 мин.осуществлен краткосрочный прогноз.

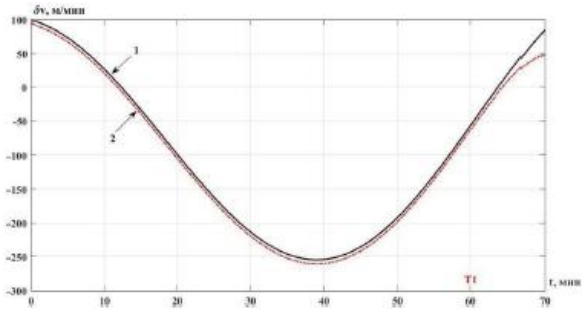


Рис.4.19. Похибка ІНС у визначенні швидкості і її короткостроковий прогноз

На рис. 4.20 позначено: 1 – модель помилок ІНС у визначенні швидкості; 2 – прогноз швидкості за допомогою ГА.

Коваленко переписала чужий текст і зробила помилку: написала про рис. 4.20, тоді як треба про рис. 4.19.

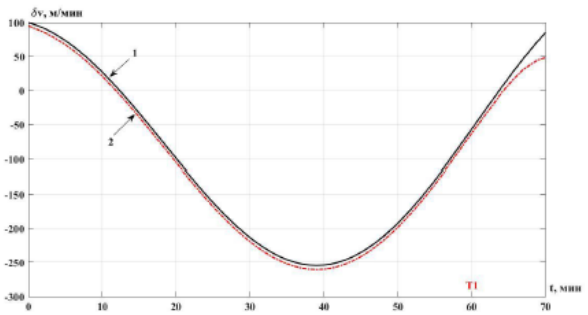


Рис.5.8. Погрешность ИНС в определении скорости и ее краткосрочный прогноз

На Рис.5.8 обозначено: 1 - Модель ошибок ИНС в определении скорости; 2 - прогноз скорости при помощи ГА.

С. 242. | **С. 104.**

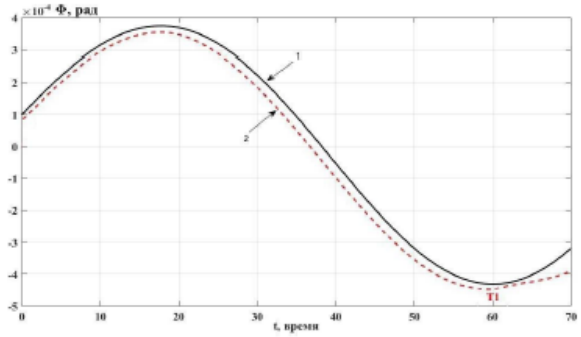


Рис. 4.20. Кут відхилення ГСП і його короткостроковий прогноз

На рис. 66 позначено: 1 – модель помилок ІНС у визначенні кута відхилення ГСП; 2 – прогноз кута відхилення ГСП за допомогою ГА.

Коваленко переписала чужий текст і зробила помилку: написала про рис. 66, тоді як треба про рис. 4.20.

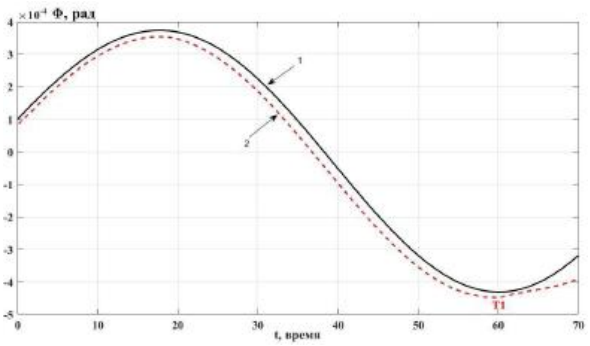


Рис.5.9. Угол отклонения ГСП и его краткосрочный прогноз

На Рис.5.9 обозначено: 1 - модель ошибок ИНС в определении угла отклонения ГСП; 2 - прогноз угла отклонения ГСП с помощью ГА.

С. 243. | **С. 105.**

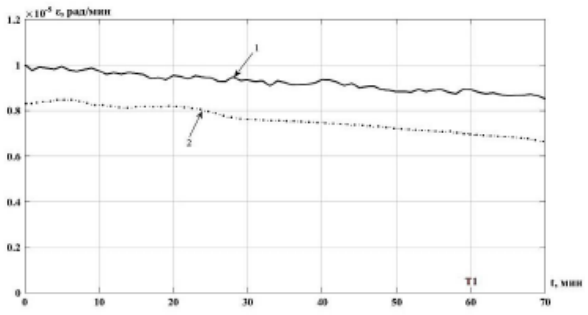


Рис. 4.21. Швидкість дрейфу ГСП і її короткостроковий прогноз

1 – модель помилок ІНС у визначенні швидкості дрейфу ГСП; 2 – прогноз швидкості дрейфу ГСП за допомогою ГА.

На рис. 4.22 – 4.24 наведені результати моделювання похибок ІНС, модель ГА, на основі оціночної вибірки і довгостроковий прогноз. Для побудови моделі використана коротка вибірка.

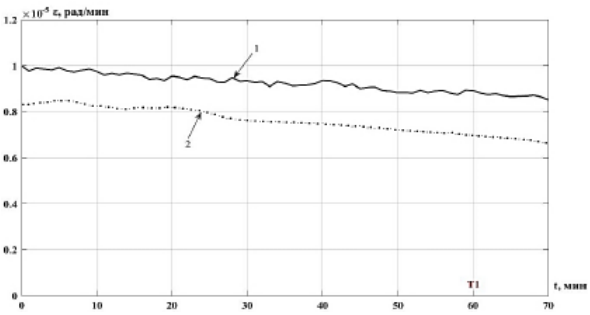
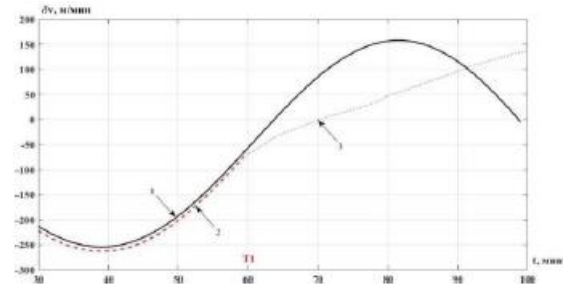
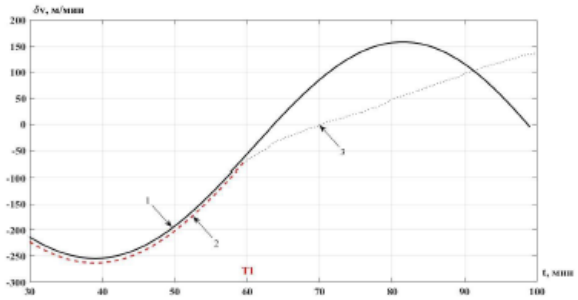
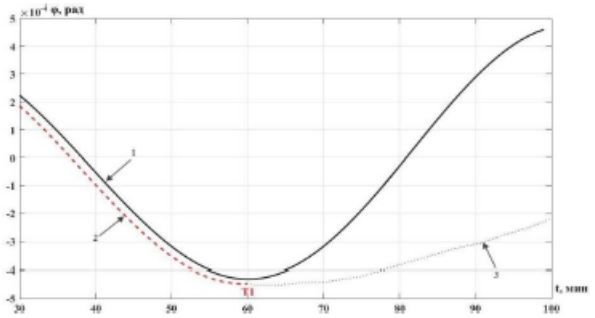
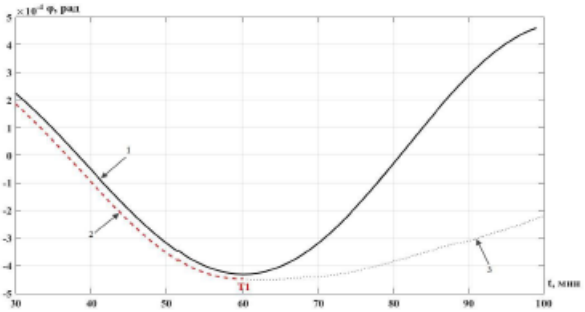
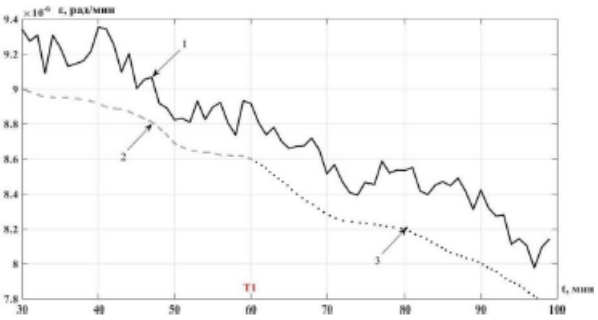
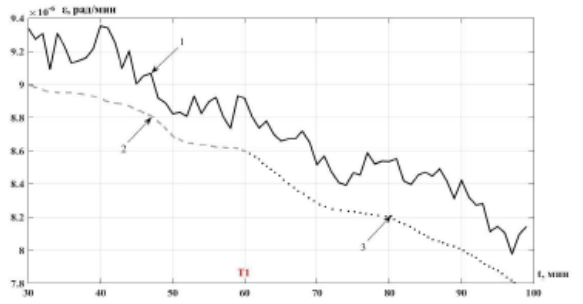


Рис.5.10. Скорость дрейфа ГСП и ее краткосрочный прогноз

На Рис.5.10 обозначено: 1 - модель ошибок ИНС в определении скорости дрейфа ГСП; 2 – прогноз скорости дрейфа ГСП с помощью ГА.

На Рис.5.11-5.13 приведены результаты моделирования погрешностей ИНС, модель ГА, на основе оценочной выборки и долгосрочный прогноз. Для построения модели использована короткая выборка.

	С. 243.	**С. 105.**
	 Рис. 4.22. Похибка ІНС у визначенні швидкості і її довгостроковий прогноз1 – модель помилок ІНС у визначенні швидкості; 2 – оцінка швидкості з ГА; 3 – прогноз.	 Рис.5.11. Погрешность ИНС в определении скорости и ее долгосрочный прогноз На Рис.5.11 обозначено: 1 – модель ошибок ИНС в определении скорости; 2 – оценка скорости с ГА; 3 – прогноз.
	С. 243.	**С. 106.**
	 Рис. 4.23. Кут відхилення ГСП і його довгостроковий прогноз 1 – модель помилок ІНС у визначенні кута відхилення ГСП; 2 – оцінка кута відхилення ГСП за допомогою ГА; 3 – прогноз.	 Рис.5.12. Угол отклонения ГСП и его долгосрочный прогноз На Рис.5.12 обозначено: 1 – модель ошибок ИНС в определении угла отклонения ГСП; 2 – оценка угла отклонения ГСП с помощью ГА; 3 – прогноз .
	С. 244.	**С. 106.**
	 На рис. 70 позначено: 1 – модель помилок ІНС у визначенні швидкості дрейфу ГСП; 2 – оцінка швидкості дрейфу ГСП за допомогою ГА; 3 – прогноз. На рис. 4.25 – 4.27 представлені результати моделювання похибок ІНС і короткостроковий прогноз по моделі, отриманої на основі оціночної вибірки.	 Рис.5.13. Скорость дрейфа ГСП и ее прогноз На Рис.5.13 обозначено: 1 – модель ошибок ИНС в определении скорости дрейфа ГСП; 2 – оценка скорости дрейфа ГСП с помощью ГА; 3 – прогноз. На Рис.5.14-5.16 представлены результаты моделирования погрешностей ИНС и краткосрочный прогноз по модели, полученной на основен оценочной выборки.
	С. 244.	**С. 107.**

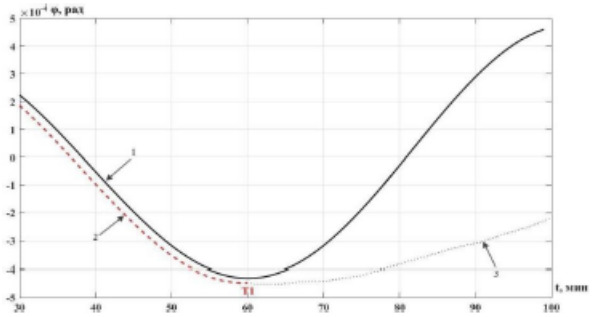


Рис. 4.25. Похибка ІНС у визначенні швидкості і її прогноз
1 – модель помилок ІНС у визначенні швидкості;
2 – прогноз швидкості за допомогою алгоритма ГА.

Переписуючи чужий текст, Коваленко зробила помилку: до підпису про рис. 4.25 вставила рисунок 4.23, який вона вже скопіювала собі раніше на с. 243.
Недолугий плагіат.

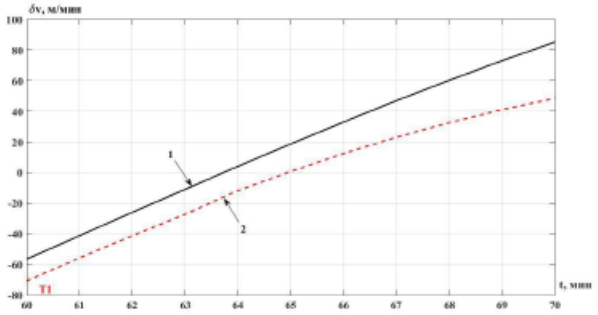


Рис.5.14. Погрешность ИНС в определении скорости и ее прогноз
На Рис.5.14 обозначено: 1 – модель ошибок ИНС в определении скорости; 2 – прогноз скорости при помощи алгоритмы ГА.

С. 245. | **С. 107.**

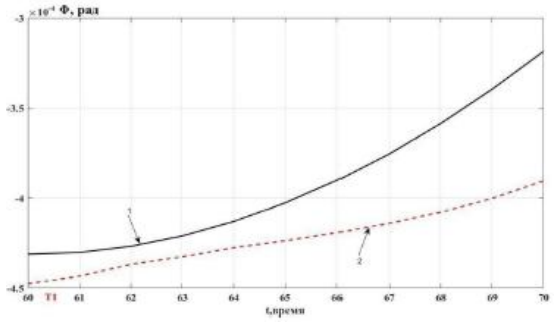


Рис. 4.27. Кут відхилення ГСП і його прогноз
1 – модель помилок ІНС у визначенні кута відхилення ГСП; 2 – прогноз кута відхилення ГСП за допомогою алгоритму ГА.

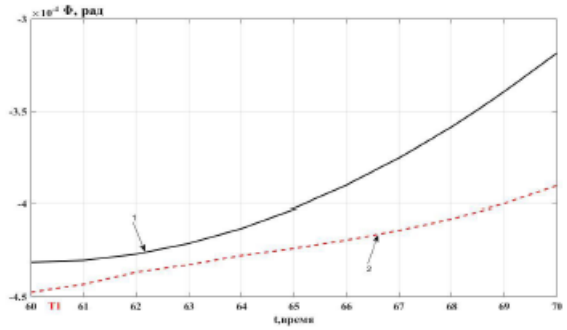


Рис.5.15. Угол отклонения ГСП и его прогноз
На Рис.5.15 обозначено: 1 – модель ошибок ИНС в определении угла отклонения ГСП; 2 – прогноз угла отклонения ГСП с помощью алгоритмы ГА.

С. 245. | **С. 108.**

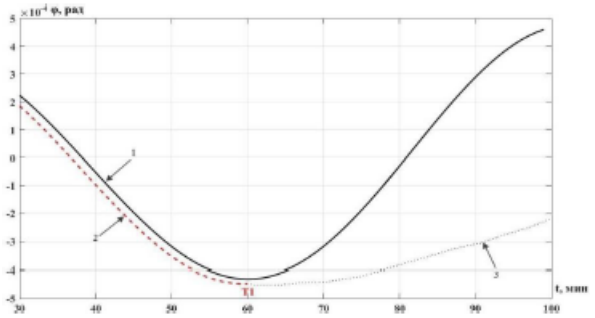


Рис. 4.27. Швидкість дрейфу ГСП і її прогноз
позначено: 1 – модель помилок ІНС у визначенні швидкості дрейфу ГСП; 2 – прогноз швидкості дрейфу ГСП за допомогою ГА.
Алгоритми функціонування ДЕС представлені на рисунку 4.28.

Переписуючи чужий текст, Коваленко знову зробила ту саму помилку: до підпису про рис.

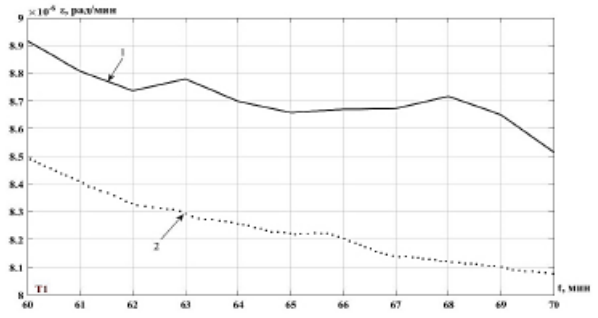
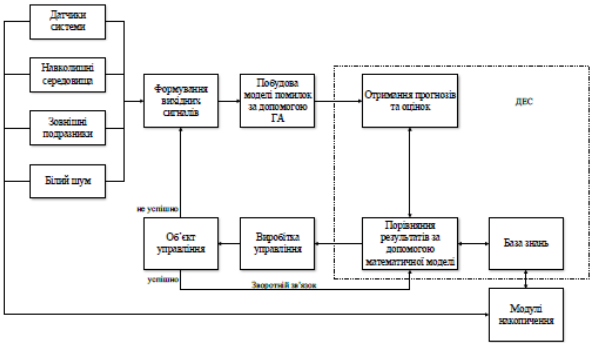
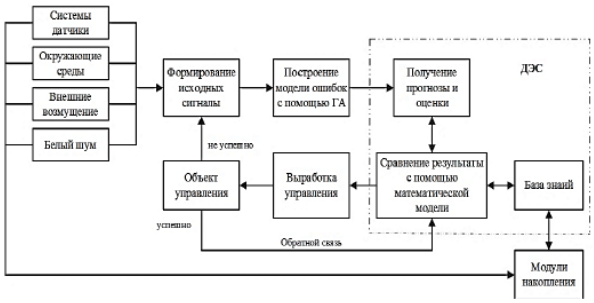
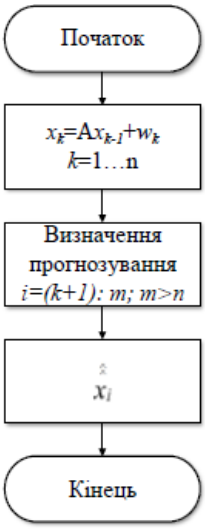
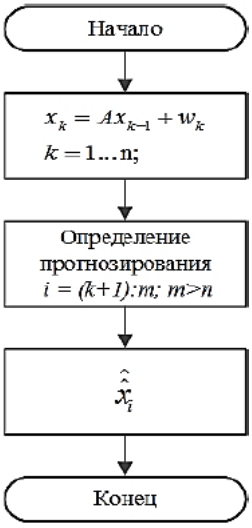


Рис.5.16. Скорость дрейфа ГСП и ее прогноз
На Рис.5.16 обозначено: 1 – модель ошибок ИНС в определении скорости дрейфа ГСП; 2 – прогноз скорости дрейфа ГСП с помощью ГА.
Алгоритмы функционирования ДЭС представлены на Рис.5.17-5.20.

	4.27 вставила той самий рисунок 4.23, який вона вже скопіювала собі раніше на с. 243, а потім на с. 244. **Недолугий плагіат.**	
	С. 246.	**С. 108.**
	 Рис. 4.28. Модель системи контролю та діагностики стану КБО на основі ДЕС з нечіткою логікою	 Рис.5.17. Функциональная структура схема системы контроля и диагностики состояния ПНК на основе ДЭС с нечеткой логикой
	С. 246.	**С. 109.**
	 Рис. 4.29. Алгоритми прогнозу	 Рис.5.18. Алгоритмы прогноза
	С. 247.	**С. 111.**
	Діагностична матриця конкретної КБО на обраному режимі є таблицею чисельних значень коефіцієнтів і, яка дозволяє за відхиленнями деяких вимірювальних параметрів визначати відхилення безпосередньо не вимірюваних параметрів стану , таблиця 4.1 **Коваленко в переписаному чужому тексті забула скопіювати коефіцієнти та параметри.** **Недолугий плагіат.**	Диагностическая матрица конкретной ПНК на выбранном режиме является таблицей численных значений коэффициентов a_{ij} и b_{ij}, позволяющей по отклонениям некоторых измеряемых параметров δy_{ij} определять отклонения непосредственно не измеряемых параметров состояния δx_{ij}.
	С. 247.	**С. 112.**
	Таблиця 4.1 Фрагмент діагностичної матриці ІНС	Таблица 2. Фрагмент диагностической матрицы ИНС

№	D_N_1	D_G_v	D_N_2	D_T_2	D_P_2	D_T_3	D_P_4	D_P_6	D_F_c	D_G_t	D_R	RESULT
1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	ETALON
2	-0,18	0,12	0,16	0,14	0,00	-0,08	-0,33	-0,33	0,41	-0,03	-0,06	АКС 1%
3	-0,46	0,31	0,52	0,39	0,00	-0,27	-0,93	-0,93	1,11	-0,11	-0,18	АКС 3%
4	-0,68	0,57	0,85	0,65	0,00	-0,46	-1,57	-1,57	1,87	-0,19	-0,31	АКС 5%
5	0,19	0,07	-0,28	0,04	0,00	0,16	0,82	0,82	-0,66	0,33	0,49	АКС1 1%
6	0,78	0,24	-0,93	0,18	0,00	0,45	2,57	2,53	-2,08	1,03	1,54	АКС1 3%
7	1,16	0,39	-1,61	0,27	0,00	0,85	4,27	4,28	-3,60	1,76	2,62	АКС1 5%
8	-0,10	0,84	0,13	0,85	0,00	-0,07	0,53	0,51	0,26	0,72	1,02	АКС2 1%
9	-0,29	2,46	0,36	2,49	0,00	-0,15	1,58	1,58	0,79	2,16	3,01	АКС2 3%
10	-4,28	6,80	2,16	0,36	0,00	-1,16	-0,11	-0,14	4,55	3,21	5,14	АКС2 5%
11	-0,78	-0,06	0,06	-0,07	0,00	-0,03	-0,18	-0,18	0,11	-0,10	-0,15	АКС3 1%
12	-2,33	-0,20	0,16	-0,18	0,00	-0,08	-0,53	-0,53	0,30	-0,34	-0,47	АКС3 3%
13	-4,07	-0,37	0,25	-0,32	0,00	-0,13	-0,96	-1,00	0,55	-0,59	-0,84	АКС3 5%
14	0,68	0,21	-0,12	0,25	0,00	0,07	0,52	0,53	-0,28	0,31	0,46	АКС4 1%
15	2,04	1,20	-0,41	0,58	0,00	0,21	1,64	1,64	-0,93	0,94	1,38	АКС4 3%
16	3,41	1,43	-0,71	0,96	0,00	0,36	2,71	2,71	-1,8	1,62	2,30	АКС4 5%

С. 248.

№	D_N_2	D_G_v	D_N_2	D_T_2	D_P_2	D_T_3	D_P_4	D_P_6	D_F_c	D_G_t	D_R	RESULT
1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	ETALON
2	-0,18	0,12	0,16	0,14	0,00	-0,08	-0,33	-0,33	0,41	-0,03	-0,06	АКС 1%
3	-0,46	0,31	0,52	0,39	0,00	-0,27	-0,93	-0,93	1,11	-0,11	-0,18	АКС 3%
4	-0,68	0,57	0,85	0,65	0,00	-0,46	-1,57	-1,57	1,87	-0,19	-0,31	АКС 5%
5	0,19	0,07	-0,28	0,04	0,00	0,16	0,82	0,82	-0,66	0,33	0,49	АКС1 1%
6	0,78	0,24	-0,93	0,18	0,00	0,45	2,57	2,53	-2,08	1,03	1,54	АКС1 3%
7	1,16	0,39	-1,61	0,27	0,00	0,85	4,27	4,28	-3,60	1,76	2,62	АКС1 5%
8	-0,10	0,84	0,13	0,85	0,00	-0,07	0,53	0,51	0,26	0,72	1,02	АКС2 1%
9	-0,29	2,46	0,36	2,49	0,00	-0,15	1,58	1,58	0,79	2,16	3,01	АКС2 3%
10	-4,28	6,80	2,16	0,36	0,00	-1,16	-0,11	-0,14	4,55	3,21	5,14	АКС2 5%
11	-0,78	-0,06	0,06	-0,07	0,00	-0,03	-0,18	-0,18	0,11	-0,10	-0,15	АКС3 1%
12	-2,33	-0,20	0,16	-0,18	0,00	-0,08	-0,53	-0,53	0,30	-0,34	-0,47	АКС3 3%
13	-4,07	-0,37	0,25	-0,32	0,00	-0,13	-0,96	-1,00	0,55	-0,59	-0,84	АКС3 5%
14	0,68	0,21	-0,12	0,25	0,00	0,07	0,52	0,53	-0,28	0,31	0,46	АКС4 1%
15	2,04	1,20	-0,41	0,58	0,00	0,21	1,64	1,64	-0,93	0,94	1,38	АКС4 3%
16	3,41	1,43	-0,71	0,96	0,00	0,36	2,71	2,71	-1,8	1,62	2,30	АКС4 5%

С. 110.

Рис. 4. 30. Основні алгоритми роботи

С. 248.

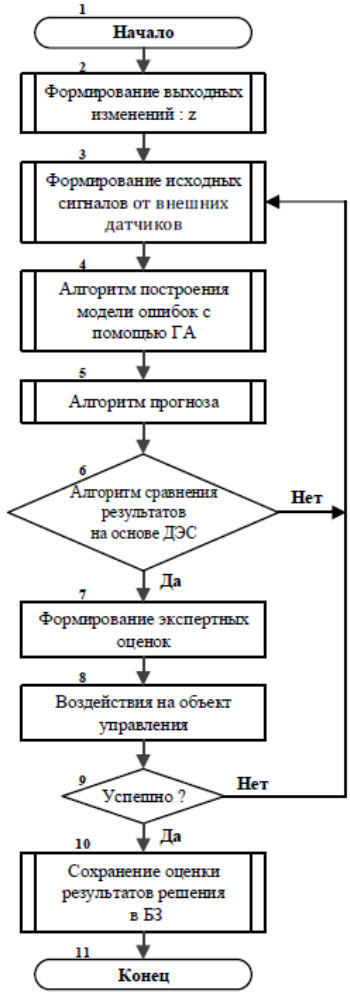


Рис.5.19. Основные алгоритмы работы

С. 111.

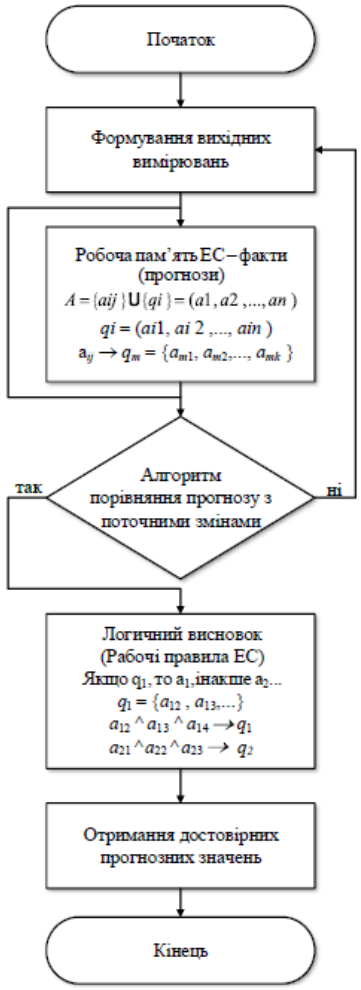
<table>
<tr>
<td></td>
<td> 

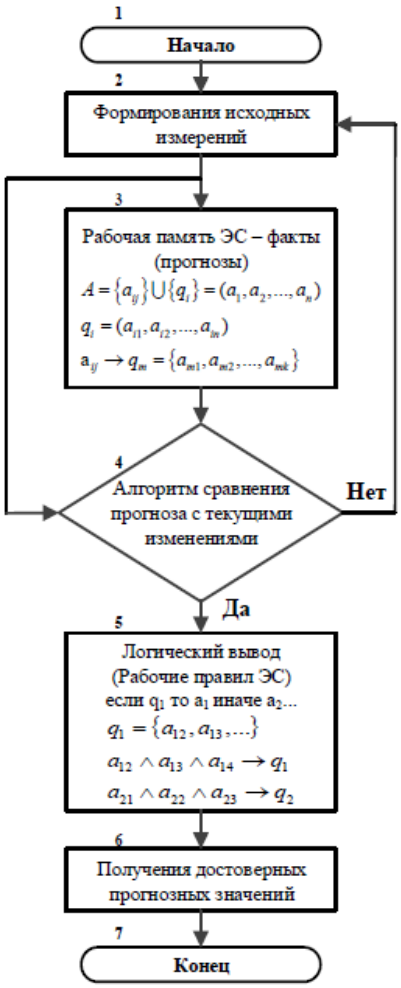
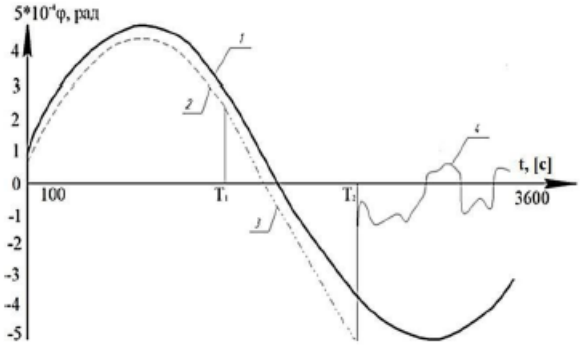
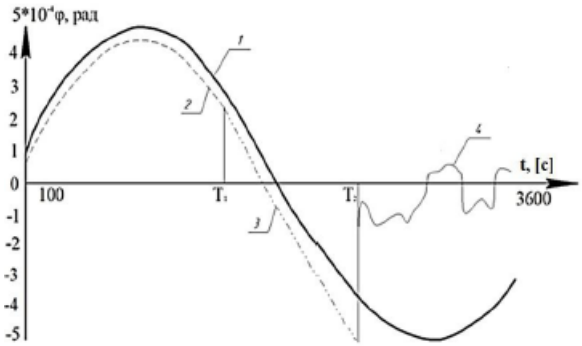
Рис. 4.31 Алгоритми поріняння за допомогою ДЕС</td>
<td> 

Рис.5.20. Алгоритмы сравнения с помощью ДЭС</td>
</tr>
<tr>
<td></td>
<td>С. 249.</td>
<td>С. 112.</td>
</tr>
<tr>
<td></td>
<td>Нехай в процесі контролю технічного стану ІНС були заміряні наступні параметри: n_2 – похибка у визначенні широти місцевості (%); G_v – похибка у визначенні швидкості ПС v_x; T_2* – похибка визначення курсу; P_2* – оцінка швидкості дрейфу ГСП ε_x; T_3* – похибка визначення тангажу; T_4* – похибка визначення крену; P_4* – оцінка швидкості дрейфу ГСП ε_y; F_c – похибка у визначенні довготи місцевості; G_t – похибка у визначенні швидкості ПС v_y; R – уявне прискорення.
У таблиці 4.2 представлені параметри відхилень і відповідні їм лінгвістичні змінні: LN – дуже мале; MN – невелике; Z – близько нуля; MP – середнє; LP – дуже велике. Процес тестування працездатності ДЕС перевіряються за тими рядками бази правил, які не увійшли до навчальної множини: по двох попереднім рядкам (з відхиленням параметрів стану ІНС в вузлах на 1 і 3% відповідно).

Помилка перекладу: російський вислів «по двум предшествующим строчкам» Коваленко неправильно переклала як «по двох попереднім рядкам», тоді як треба «по двох попередніх рядках».
Плагіат.</td>
<td>Пусть в процессе контроля технического состояния ИНС были замерены следующие параметры: n_2 - погрешность в определении широты местности (%); G_v - погрешность в определении скорости ЛА v_x; T_2* - погрешность определения курса; P_2* - оценка скорости дрейфа ГСП ε_x; T_3* - погрешность определения тангажа; T_4* - погрешность определения крена; P_4* - оценка скорости дрейфа ГСП ε_y; F_c - погрешность в определении долготы местности; G_t - погрешность в определении скорости ЛА v_y; R - кажущееся ускорение.
В таблице 2 представлены параметры отклонений и соответствующие им лингвистические переменные: LN - очень малое; MN - небольшое; Z - около нуля; MP - среднее; LP - очень большое. Процесс тестирования работоспособности ДЭС проверяются по тем строчкам базы правил (Таблица 3), которые не вошли в обучающее множество: по двум предшествующим строчкам (с отклонением параметров состояния ИНС в узлах на 1 и 3 % соответственно).</td>
</tr>
<tr>
<td></td>
<td>С. 249.</td>
<td>С. 113.</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>Таблиця 4.2
Фрагмент бази нечітких експертних правил
<table>
<tr><th>Атрибути і їх значення</th><th>Результат</th></tr>
<tr><td>Якщо</td><td>То=АКС</td></tr>
<tr><td>Якщо</td><td>То=АКС1</td></tr>
<tr><td>Якщо</td><td>То=АКС2</td></tr>
<tr><td>Якщо</td><td>То= АКС3</td></tr>
<tr><td>Якщо</td><td>То= АКС4</td></tr>
</table>
Коваленко в переписаній чужій таблиці не змогла скопіювати в першому стовпчику атрибути та їхні значення, а в другому – різні Y.
Недолугий нахабний плагіат.</td>
<td>Таблица 3.
Фрагмент базы нечетких экспертных правил
<table>
<tr><th>Атрибуты и их значения</th><th>Результат</th></tr>
<tr><td>Если $(\Delta T_2 = Z) \wedge (\Delta P_4 = Z) \wedge (\Delta P_6 = Z) \wedge (\Delta F_c = MP)$</td><td>То Y_1=АКС</td></tr>
<tr><td>Если $(\Delta N_2 = Z) \wedge (\Delta T_2 = Z) \wedge (\Delta F_c = MN) \wedge (\Delta G_t = MP)$</td><td>То Y_2=АКС1</td></tr>
<tr><td>Если $(\Delta N_2 = LN) \wedge (\Delta T_2 = Z) \wedge (\Delta P_2 = LP) \wedge (\Delta R = LP)$</td><td>То Y_3=АКС2</td></tr>
<tr><td>Если $(\Delta N_2 = LN) \wedge (\Delta G_v = Z) \wedge (\Delta P_2 = Z) \wedge (\Delta R = Z)$</td><td>То Y_4= АКС3</td></tr>
<tr><td>Если $(\Delta N_2 = MP) \wedge (\Delta G_v = Z) \wedge (\Delta T_2 = Z) \wedge (\Delta R = MP)$</td><td>То Y_5= АКС4</td></tr>
</table></td>
</tr>
<tr>
<td></td>
<td>С. 249–250.</td>
<td>С. 113.</td>
</tr>
<tr>
<td></td>
<td>В процесі наповнення БД, крім поточних вимірювань, додатково використані прогнозні значення аналізованих параметрів КБО. ДЕС з алгоритмом побудови прогнозуючих моделей, методом діагностичних матриць і правил нечіткої логіки дозволяє визначати ступінь достовірності навігаційної інформації ПС, а блок прийняття рішень, регулятор відновлюють працездатність КБО, використання оцінок і прогнозних значень похибок ІНС підвищує достовірність навігаційної інформації ПС.
Представлені результати роботи алгоритму діагностики похибок ІНС. На рис 4.31. представлена похибка ІНС у визначенні швидкості, швидкість дрейфу ГСП, оцінки адаптивним фільтром Калмана, а з моменту Т1 – прогноз, отриманий за допомогою ГА. При моделюванні для отримання похибки ІНС використана тестова математична модель. Так як в реальних умовах є інформація про оцінку, то вона використовується для діагностики стану КБО. Запропоновано використовувати прогноз похибки ІНС, щоб заздалегідь виявити момент виходу систем із зони стійкої роботи.
Коваленко пише «Представлені результати роботи алгоритму» та «Запропоновано використовувати…», а насправді переписує чужі результати та пропозиції.
Плагіат.</td>
<td>В процессе наполнения БД, помимо текущих измерений, дополнительно использованы прогнозные значения анализируемых параметров ПНК. ДЭС с алгоритмом построения прогнозирующих моделей, методом диагностических матриц и правил нечеткой логики позволяет определять степень достоверности навигационной информации ЛА, а блок принятия решений, регулятор восстанавливают работоспособность ПНК, использование оценок и прогнозных значений погрешностей ИНС повышает достоверность навигационной информации ЛА.
Представлены результаты работы алгоритма диагностики погрешностей ИНС. На Рис.5.16 представлена погрешность ИНС в определении скорости, скорость дрейфа ГСП, оценки адаптивным фильтром Калмана, а с момента Т1 - прогноз, полученный с помощью ГА. При моделировании для получения погрешности ИНС использована тестовая математическая модель. Так как в реальных условиях имеется информация об оценке, то она используется для диагностики состояния ПНК. Предложено использовать прогноз погрешности ИНС, чтобы заранее выявить момент выхода систем из зоны устойчивой работы.</td>
</tr>
<tr>
<td></td>
<td>С. 250.</td>
<td>С. 113–114.</td>
</tr>
<tr>
<td></td>
<td>Застосування ДЕС значно підвищує ефективність діагностування КБО ПС. Оперативний аналіз різноманітної інформації про поточну ситуацію дозволяє приймати рішення про стан і перспективи даного КБО. Результат роботи системи контролю та діагностики КБО з функцією відновлення представлений на рис. 4.31..</td>
<td>Применение ДЭС значительно повышает эффективность диагности- рования ПНК ЛА. Оперативный анализ разнообразной информации о текущей ситуации позволяет принимать решения о состоянии и перспективах данного ПНК.
Результат работы системы контроля и диагностики ПНК с функцией восстановления представлен на Рис.5.21.</td>
</tr>
</table>

	 Рис. 4.31. Кут відхилення ГСП, його оцінка фільтром Калмана, прогноз і похибка ІНС з регулятором позначено: 1 – математична модель зміни кута відхилення ГСП щодо супроводжуючого тригранника; 2 – оцінка кута відхилення ГСП адаптивним фільтром Калмана; 3 – прогноз кута відхилення ГСП; 4 – зміна кута відхилення ГСП при корекції в структурі ІНС.	 Рис.5.21. Угол отклонения ГСП, его оценка фильтром калмана, прогноз и погрешность ИНС с регулятором На Рис.5.21 обозначено: 1 – математическая модель изменения угла отклонения ГСП относительно сопровождающего трехгранника; 2 – оценка угла отклонения ГСП адаптивным фильтром калмана; 3 – прогноз угла отклонения ГСП; 4 – изменение угла отклонения ГСП при коррекции в структуре ИНС.
	С. 250–251.	**С. 114.**
	На основі прогнозу похибок ІНС КБО ПС в момент часу Т2 приймається рішення про переключення робочого контуру корекції (на інтервалі Т1-Т2 використана корекція в вихідному сигналі ІНС, а з моменту Т2 використана корекція в структурі ІНС за допомогою регулятора). У порівнянні з ІНС КБО використання ІНС з розробленою системою контролю в стохастичних умовах дозволяє підвищити точність за результатами математичного моделювання в середньому на 8-11%. Досліджено особливості вирішення задач контролю і діагностики КБО з використанням нечіткої ДЕС, процес формування БД, а також реалізація FDI-методу. Побудована діагностична матриця ДЕС. **Коваленко пише «**Досліджено особливості вирішення задач алгоритму**» та «**Побудована діагностична матриця**», а насправді переписує чужі результати.** **Плагіат.**	На основе прогноза погрешностей ИНС ПНК ЛА в момент времени Т2 принимается решение о переключении рабочего контура коррекции (на интервале Т1-Т2 использована коррекция в выходном сигнале ИНС, а с момента Т2 использована коррекция в структуре ИНС с помощью регулятора). По сравнению с ИНС ПНК использование ИНС с разработанной системой контроля в стохастических условиях позволяет повысить точность по результатам математического моделирования в среднем на 8-11%. Исследованы особенности решения задач контроля и диагностики ПНК с использованием нечеткой ДЭС, процесс формирования БД, а также реализация FDI-метода. Построена диагностическая матрица ДЭС.
	С. 251.	**С. 114–115.**
	База нечітких правил сформована на основі діагностичної матриці, рядки якої лягли в основу створення БЗ і функцій приналежності відповідних лінгвістичних змінних. Для 1% -го відхилення оцінки швидкості дрейфу ГСП, на основі бази нечітких правил ДЕС і операцію перетину нечітких множин. Тоді отримаємо: $\min = \begin{pmatrix} \mu_z(\Delta T_2^{(k)}) \& & \mu_z(\Delta P_4^{(k)}) \\ \& \mu_z(\Delta P_6^{(k)}) & \& \mu_z(\Delta F_c^{(k)}) \end{pmatrix} \to 0,85.$ (4.66) Достовірність прийняття рішення про справності ІНС складає 0,85. Для рядка діагностичної матриці, відповідної 3% відхилення оцінки швидкості дрейфу ГСП, отримаємо $\min(\mu_{Z_{\varepsilon_i}}) \to 0,59.$ (4.77)	База нечетких правил сформирована на основе диагностической матрицы, строки которой легли в основу создания БЗ и функций принадлежности соответствующих лингвистических переменных. Для 1%-го отклонения оценки скорости дрейфа ГСП, на основе базы нечетких правил ДЭС и операцию пересечения нечетких множеств. Тогда получим: $min \begin{pmatrix} \mu_z\left(\Delta T_2^{(k)}\right) \& & \mu_z\left(\Delta P_4^{(k)}\right) \\ \& \mu_z\left(\Delta P_6^{(k)}\right) & \& \mu_z\left(\Delta F_c^{(k)}\right) \end{pmatrix} \to 0,85$ (5.2) Достоверность принятия решения о исправности ИНС составляет 0,85. Для строки диагностической матрицы, соответствующей 3 % отклонению оценки скорости дрейфа ГСП, получим $\min(\mu_{Z_{\varepsilon_i}}) \to 0,59.$ (5.3)
	С. 251.	**С. 115.**

	Достовірність прийняття рішення про справність ІНС складає 0,59. При несправному стані ІНС коефіцієнт довіри становить 0,24. У розглянутій ситуації висока ймовірність виходу ІНС із зони стійкої роботи. Результати аналізу ДЕС збігаються з результатами досліджень, проведених в ході стендових випробувань ІНС. Результати моделювання показали високу працездатність запропонованих алгоритмів. Аналіз результатів моделювання продемонстрував, що розроблене алгоритмічне забезпечення систем контролю і діагностики дозволяють підвищити ефективність навігаційних систем і КБО ПС за рахунок завчасного виявлення моменту, коли інформація стає недостовірною і утримання КБО в зоні стійкої роботи. **Коваленко пише «**Результати моделювання показали**» та «**Аналіз результатів моделювання продемонстрував**», а насправді переписує як чужі результати, так і їх аналіз.** **Нахабний плагіат.**	Достоверность принятия решения об исправности ИНС составляет 0,59. При неисправном состоянии ИНС коэффициент доверия составляет 0,24. В рассмотренной ситуации высока вероятность выхода ИНС из зоны устойчивой работы. Результаты анализа ДЭС совпадают с результатами исследований, проведенных в ходе стендовых испытаний ИНС. Результаты моделирования показали высокую работоспособность предложенных алгоритмов. Анализ результатов моделирования продемонстрировал, что разработанное алгоритмическое обеспечение систем контроля и диагностики позволяют повысить эффективность навигационных систем и ПНК ЛА за счет заблаговременного выявления момента, когда информация становится недостоверной и удержания ПНК в зоне устойчивой работы.
5	**Коваленко Ю. Б.** **Інформаційна підтримка процесу проектування та експлуатації комплексу бортового обладнання інтегрованої модульної авіоніки.** – Дис. … доктора технічних наук. – Київ, 2025. (https://duikt.edu.ua/uploads/p_86_93515477.pdf)	**Парамонов П. П., Гатчин Ю. А., Жаринов И. О., Жаринов О. О., Дейко М. С.** **Принципы построения отраслевой системы автоматизированного проектирования в авиационном приборостроении** // Научно-технический вестник информационных технологий, механики и оптики, 2012, №6 (82), с. 111–117. (https://cyberleninka.ru/article/n/printsipy-postroeniya-otraslevoy-sistemy-avtomatizirovannogo-proektirovaniya-v-aviatsionnom-priborostroenii)
	С. 135–136.	**С. 111.**
	Сучасні тенденції розвитку приладобудівної галузі в області проектування бортового радіоелектронного обладнання нерозривно пов'язані із застосуванням систем автоматизованого проектування (САПР). Розробники САПР пропонують різні універсальні програмні інструменти, призначені для автоматизації окремих етапів проектування БРЕО. Відомі, наприклад, пакети прикладних програм: – тривимірного проектування конструкцій апаратури БРЕО – AutoCAD, Solidworks, UniGraphics; – інженерних розрахунків і моделювання теплових полів, створюваних апаратурою БРЕО: BETAsoft, Sauna, АСОНІКА-Т; – моделювання електромагнітних полів, що створюються апаратурою БРЕО –OrCAD Family Release, GENESYS; – спектрального аналізу радіочастотних сигналів, що генеруються і приймаються апаратурою БРЕО – TESLA; – моделювання міцності і резонансних характеристик конструкцій авіаційних виробів - Samcef, Mecano, Boss Quattro; – автоматичного трасування друкованих плат з урахуванням тривимірного компонування елементів монтажу – ACCEL EDA, PCAD; – конструювання та розводки джгутів в тривимірному просторі корпусів апаратури БРЕО - UG / Wiring; – моделювання гідродинамічних процесів в системах охолодження авіаційних виробів з урахуванням тепловиділення електрорадіоелементів і фізичних процесів теплопереносу – Fine / Turbo;	Современные тенденции развития приборостроительной отрасли в области проектирования бортового радиоэлектронного оборудования (БРЭО) неразрывно связаны с применением систем автоматизированного проектирования (САПР). Разработчики САПР предлагают различные универсальные программные инструменты, предназначенные для автоматизации отдельных этапов проектирования БРЭО. Известны, например, пакеты прикладных программ: – трехмерного проектирования конструкций аппаратуры БРЭО – AutoCAD, Solidworks, UniGraphics; – инженерных расчетов и моделирования тепловых полей, создаваемых аппаратурой БРЭО: BETAsoft, Sauna, Асоника-Т; – моделирования электромагнитных полей, создаваемых аппаратурой БРЭО – OrCAD Family Release, GENESYS; – спектрального анализа радиочастотных сигналов, генерируемых и принимаемых аппаратурой БРЭО – TESLA; – моделирования прочностных и резонансных характеристик конструкций авиационных изделий – Samcef, Mecano, Boss Quattro; – автоматической трассировки печатных плат с учетом трехмерной компоновки элементов монтажа – ACCEL EDA, PCAD; – конструирования и разводки жгутов в трехмерном пространстве корпусов аппаратуры БРЭО – UG/Wiring; – моделирования гидродинамических процес-

	– електронного моделювання радіотехнічних сигналів і ланцюгів - Cadence Design, Mentor Graphics, XILINX Foundation, ALTERA, MicroCap; – проектування програмного забезпечення C / C ++ / C #, ADA і ін.	сов в системах охлаждения авиационных изделий с учетом тепловыделения электрорадиоэлементов и физических процессов теплопереноса – Fine/Turbo; – электронного моделирования радиотехнических сигналов и цепей – Cadence Design, Mentor Graphics, XILINX Foundation, ALTERA, MicroCap; – проектирования программного обеспечения C/C++/C#, ADA и др.
	С. 136–137.	**С. 111.**
	Дані програмні інструменти складають основу автоматизації проектних технологій і є закінченими програмними рішеннями. Специфічність форматів зберігання і представлення даних результатів проектування в таких пакетах програм істотно впливає на рівень автоматизації процесу проектування БРЕО в цілому. Це пов'язано в першу чергу з тим, що формат представлення даних результатів проектування будь-якої складової частини виробу, отриманого із застосуванням САПР одного виду, виявляється неузгодженим з форматом зберігання і представлення даних в САПР іншого виду, що застосовується на наступному, більш пізньому, етапі проектування.	Данные программные инструменты составляют основу автоматизации проектных технологий и являются законченными программными решениями. Специфичность форматов хранения и представления данных результатов проектирования в таких пакетах программ существенно влияет на уровень автоматизации процесса проектирования БРЭО в целом. Это связано в первую очередь с тем, что формат представления данных результатов проектирования какой-либо составной части изделия, полученного с применением САПР одного вида, оказывается несогласованным с форматом хранения и представления данных в САПР другого вида, применяемого на последующем, более позднем, этапе проектирования.
	С. 137.	**С. 111.**
	У таких випадках виникає потреба введення в технологічні маршрути проектування виробів «ручних» процедур підготовки (переформатування) окремих електронних документів, що істотно знижує ефект застосування САПР і підвищує сумарну трудомісткість розробки конструкторської, технологічної та програмної документації на виріб в цілому. Для виключення або мінімізації обсягів «ручних» робіт в процесі проектування всі застосовувані при розробці БРЕО кошти САПР повинні інтегруватися в єдину галузеву САПР [66], що володіє сумісними форматами уявлення, зберігання і передачі даних і підтримує всі етапи життєвого циклу БРЕО - від маркетингових досліджень доцільності розробки нових видів виробів до їх подальшої утилізації. Відповідна функціональна схема взаємодії різнорідних засобів САПР в складі єдиної галузевої САПР підтримки життєвого циклу БРЕО приведена на рисунку 3.1. **Покликання [66] – це:** Коваленко Ю., Козлюк І., «Структурний аналіз технологічного проектування і управління технологічним проектом у авіаційній галузі», Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка, № 59, С. 22-31, 2018. **Коваленко замінила в переписаному чужому тексті 2012-го року покликання [1] на фальшиве джерело [66] – власну статтю 2018-го року.** **Фальсифікація джерел.** **Плагіат.**	В таких случаях возникает потребность введения в технологические маршруты проектирования изделий «ручных» процедур подготовки (переформатирования) отдельных электронных документов, что существенно снижает эффект применения САПР и повышает суммарную трудоемкость разработки конструкторской, технологической и программной документации на изделие в целом. Для исключения или минимизации объемов «ручных» работ в процессе проектирования все применяемые при разработке БРЭО средства САПР должны интегрироваться в единую отраслевую САПР [1], обладающую совместимыми форматами представления, хранения и передачи данных и поддерживающую все этапы жизненного цикла БРЭО – от маркетинговых исследований целесообразности разработки новых видов изделий до их последующей утилизации. Соответствующая функциональная схема взаимодействия разнородных средств САПР в составе единой отраслевой САПР поддержки жизненного цикла БРЭО приведена на рис. 1. **Покликання [1] – це:** Гатчин Ю.А., Жаринов И.О., Жаринов О.О. Архитектура программного обеспечения автоматизированного рабочего места разработчика бортового авиационного оборудования // Научно-технический вестник информационных технологий, механики и оптики. – 2012. – № 2 (78). – С. 140–141.
	С. 137.	**С. 112.**

<table>
<tr>
<td></td>
<td>


Рис. 3.1. Функціональна схема взаємодії засобів САПР в складі єдиної галузевої САПР підтримки життєвого циклу БРЕО

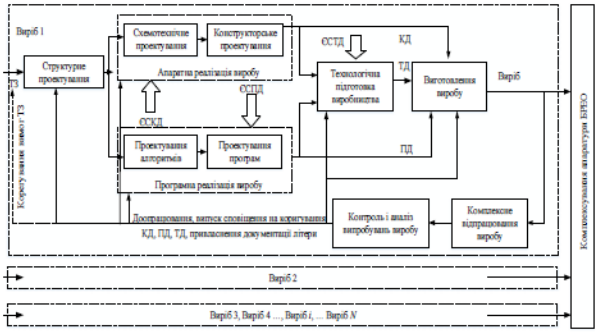
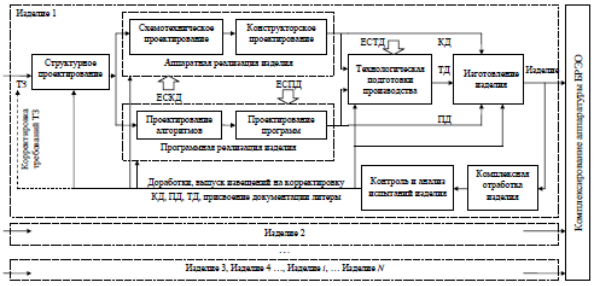
Коваленко переписала чужу схему без пояснення скорочень.
Плагіат.</td>
<td>

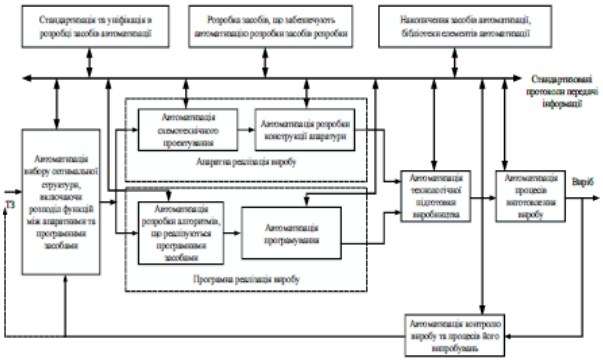

Рис. 1. Функциональная схема взаимодействия разнородных средств САПР в составе единой отраслевой САПР поддержки жизненного цикла БРЭО (CAE – Computer Aided Engineering; CAD – Computer AidedDesign; CAM – Computer Aided Manufacturing; ECAD – Electronic Computer Aided Design; PDM – Product Data Management; ERP – Enterprise Resource Platting; MRP – Manufacturing Requirement Platting; MES – Manufacturing Executing System; SCM – Supply Chain Management; CRM – Customer Relationship Management; SCADA – Supervisory Control and Data Acquisition; CNC – Computer Numerical Control; S&SM – Sales and Service Management; CPC – Collaborative Product Commerce; CASE – Computer Aided Software Engineering)</td>
</tr>
<tr>
<td></td>
<td>С. 138.</td>
<td>С. 112.</td>
</tr>
<tr>
<td></td>
<td>Інтеграція різнорідних засобів САПР передбачає об'єднання технологічних операцій, що реалізуються кожною з них окремо, в єдину проектну середу «проектування-виробництво-експлуатація». Єдине проектне середовище забезпечить автоматизовану підтримку етапів життєвого циклу БРЕО і дозволить дослідити і розробити «наскрізні» проектні технології створення виробів, що входять в БРЕО, до їх реалізації у вигляді готових фізичних об'єктів; розробити технологічний цикл розробки, виробництва і випробувань апаратури БРЕО;

Коваленко в переписаному чужому тексті видалила покликання.
Коваленко приєднала до переписаного речення заголовок чужого розділу, закінчивши абзац крапкою з комою замість крапки.
Плагіат.</td>
<td>Как следует из рис. 1, интеграция разнородных средств САПР предполагает объединение технологических операций, реализуемых каждой из них в отдельности, в единую проектную среду «проектирование–производство–эксплуатация» [2]. Единая проектная среда обеспечит автоматизированную поддержку этапов жизненного цикла БРЭО и позволит исследовать и разрабатывать «сквозные» проектные технологии создания изделий, входящих в БРЭО, до их реализации в виде готовых физических объектов.

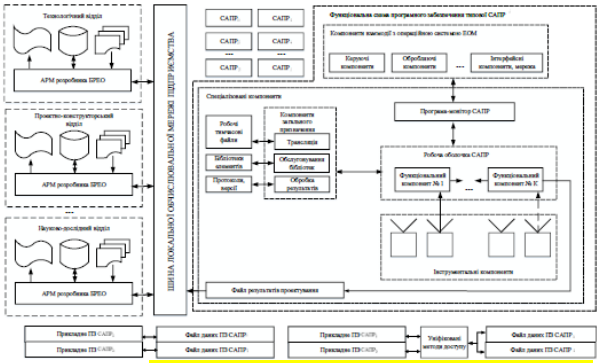
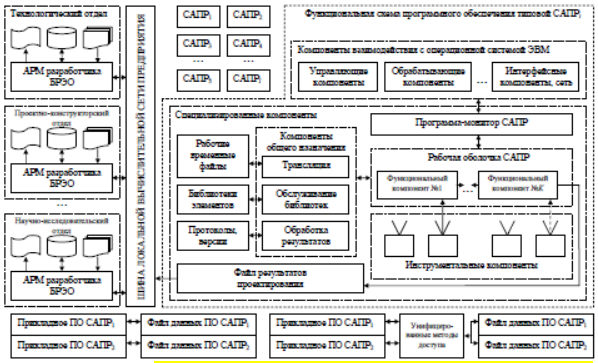
Технологический цикл разработки, производства и испытаний БРЭО на основе отраслевой САПР</td>
</tr>
<tr>
<td></td>
<td>С. 138.</td>
<td>С. 112–113.</td>
</tr>
<tr>
<td></td>
<td>Технологічний цикл розробки, виробництва і випробувань апаратури БРЕО на основі галузевої САПР авіаційного приладобудування представлений на рисунку 3.2</td>
<td>Технологический цикл разработки, производства и испытаний аппаратуры БРЭО на основе отраслевой САПР авиационного приборостроения представлен на рис. 2.

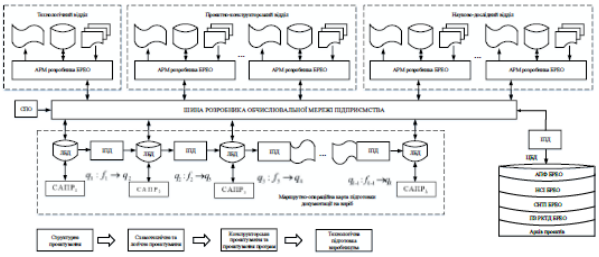
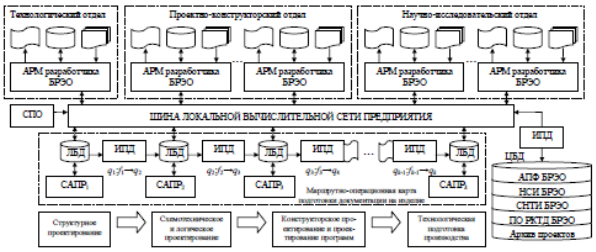
<…></td>
</tr>
</table>

	 Рис. 3.2 Технологічний цикл розробки, виробництва і випробувань апаратури БРЕО Де, КД – конструкторська документація; ТД – технологічна документація; ПД – програмна документація; ЄСКД – єдина система конструкторської документації; ЄСТД – єдина система технологічної документації; ЄСПД – єдина система програмної документації; ТЗ – технічне завдання.	 Рис. 2. Технологический цикл разработки, производства и испытаний аппаратуры БРЭО (КД – конструкторская документация; ТД – технологическая документация; ПД – программная документация; ЕСКД – единая система конструкторской документации; ЕСТД – единая система технологической документации; ЕСПД – единая система программной документации; ТЗ – техническое задание)
	С. 138–139.	**С. 112.**
	Отже, основу технологічного циклу розробки, виробництва і випробувань апаратури БРЕО складають проектні процедури структурного проектування окремих виробів, реалізації їх апаратних і програмних компонентів, підтримки завдань технологічної підготовки виробництва, виготовлення і випробувань виробу. Суттєвою складовою проектування БРЕО є також етап комплексування апаратури [3]. **Коваленко пише «Отже», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Покликання [3] – це:** ДСТУ 2862-94. Надійність техніки. Методи розрахунку показників надійності. К.: Держстандарт України, 1995, 39 с.20. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	Как следует из рис. 2, основу технологического цикла разработки, производства и испытаний аппаратуры БРЭО составляют проектные процедуры структурного проектирования отдельных изделий, реализации их аппаратных и программных компонентов, поддержки задач технологической подготовки производства, изготовления и испытаний изделия. Существенной составляющей проектирования БРЭО является также этап комплексирования аппаратуры [3]. **Покликання [3] – це:** Андреев Л.В., Богословский С.В., Видин Б.В., Жаринов И.О., Жаринов О.О., Парамонов П.П., Сабо Ю.И. Формализация вектора наблюдений измерительного комплекса беспилотных летательных аппаратов // Изв. вузов. Приборостроение. – 2009. – Т. 52. – № 11. – С. 23–27.
	С. 139.	**С. 112.**
	Основне завдання комплексування апаратури полягає в об'єднанні методами структуризації різнорідних приладів і пристроїв (виріб 1, виріб 2 і т.д.) в єдиний комплекс бортового обладнання. Об'єднання здійснюється на апаратному та програмному рівні через аналіз інформаційних потоків аеродинамічних, пілотажно-навігаційних та ін. параметрів руху повітряного судна і синтез структур БРЕО. Результатом етапу комплексування апаратури є комплект документації на комплекс БРЕО. Відповідно до прийнятої термінології комплекс БРЕО являє собою систему більш високого порядку по відношенню до окремих її компонентів.	Основная задача комплексирования аппаратуры заключается в объединении методами структуризации разнородных приборов и устройств (изделие 1, изделие 2 и т.д.) в единый комплекс бортового оборудования. Объединение осуществляется на аппаратном и программном уровне через анализ информационных потоков аэродинамических, пилотажно-навигационных и др. параметров движения летательного аппарата и синтез структур БРЭО. Результатом этапа комплексирования аппаратуры является комплект документации на комплекс БРЭО. В соответствии с принятой терминологией комплекс БРЭО представляет собой систему более высокого порядка по отношению к отдельным ее компонентам.
	С. 139.	**С. 112–113.**
	Таким чином, завдання автоматизації етапів «проектування-виробництво-експлуатація» БРЕО на основі єдиного проектного середовища галузевої САПР авіаційного приладобудування слід розглядати як завдання створення такої інтегрованої САПР, в складі якої:	Таким образом, задачу автоматизации этапов «проектирование–производство–эксплуатация» БРЭО на основе единой проектной среды отраслевой САПР авиационного приборостроения следует рассматривать как задачу создания такой интегрированной САПР, в составе которой:

	– автоматизувалися би всі основні проектні процедури проектування БРЕО; – взаємодія (обмін даними між додатками) різнорідних засобів САПР здійснювалася б на основі стандартизованих протоколів передачі інформації, наприклад, за стандартами групи STEP (Standard for the Exchange of Product). **Коваленко пише «Таким чином», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Плагіат.**	– автоматизировались бы все основные проектные процедуры проектирования БРЭО; – взаимодействие (обмен данными между приложениями) разнородных средств САПР осуществлялось бы на основе стандартизованных протоколов передачи информации, например, по стандартам группы STEP (Standard for the Exchange of Product).
	С. 139–140.	**С. 113–114.**
	Схема взаємодії основних функціональних компонентів, що автоматизують проектні процедури створення БРЕО на основі галузевої САПР, представлена на рис. 3.3. Автоматизація етапу структурного проектування виробу здійснюється засобами автоматизації процедур вибору оптимальної (по заздалегідь введеному критерію) структури виробу і процедур розподілу функцій між апаратними та програмними компонентами виробу. Методи вибору оптимальної структури виробу в класі бортового приладового обладнання розглянуті в роботі [4], рішення задачі «про призначення» функціональних завдань на доступні обчислювальні ресурси БРЕО розглянуто в [5]. Автоматизація етапу розробки апаратної реалізації виробу здійснюється послідовно на основі автоматизації процедур схемотехнічного проектування та процедур розробки конструкції виробу. Методи автоматизації процедур схемотехнічного проектування і генерації проектних рішень розглянуті в роботах [6] і [7] відповідно. **Коваленко переписала чужий текст разом із номерами покликань №4, 5, 6 і 7, під якими в її дисертації знаходяться зовсім інші джерела – різні ДСТУ 1995–2000 років.** **Фальсифікація джерел.** **Плагіат.**	Схема взаимодействия основных функциональных компонентов, автоматизирующих проектные процедуры создания БРЭО на основе отраслевой САПР, представлена на рис. 3. Как следует из рис. 3, автоматизация этапа структурного проектирования изделия осуществляется средствами автоматизации процедур выбора оптимальной (по заранее введенному критерию) структуры изделия и процедур распределения функций между аппаратными и программными компонентами изделия. Методы выбора оптимальной структуры изделия в классе бортового приборного оборудования рассмотрены в работе [4], решение задачи «о назначении» функциональных задач на доступные вычислительные ресурсы БРЭО рассмотрено в [5]. Автоматизация этапа разработки аппаратной реализации изделия осуществляется последовательно на основе автоматизации процедур схемотехнического проектирования и процедур разработки конструкции изделия. Методы автоматизации процедур схемотехнического проектирования и генерации проектных решений рассмотрены в работах [6] и [7] соответственно.
	С. 140.	**С. 113.**
	 Рис. 3.3 Схема взаємодії основних функціональних компонентів, що автоматизують проектні процедури створення апаратури комплексів БРЕО	 Рис. 3. Схема взаимодействия основных функциональных компонентов, автоматизирующих проектные процедуры создания аппаратуры комплексов БРЭО
	С. 140.	**С. 114.**
	Автоматизація етапу розробки програмної реа-	Автоматизация этапа разработки программной

	лізації виробу здійснюється послідовно на основі автоматизації процедур проектування алгоритмічного забезпечення і процедур автоматизації програмування. Методи автоматизації процедур проектування алгоритмів і програм розглянуті в роботі [8]. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело.** **Фальсифікація джерел.** **Плагіат.**	реализации изделия осуществляется последовательно на основе автоматизации процедур проектирования алгоритмического обеспечения и процедур автоматизации программирования. Методы автоматизации процедур проектирования алгоритмов и программ рассмотрены в работе [8].
	С. 140–141.	**С. 114.**
	Результатом автоматизації етапів розробки апаратної і програмної реалізації виробу є комплекти програмної та конструкторської документації, що представляють собою основу (вхідні дані) для етапу технологічної підготовки виробництва. Автоматизація технологічної підготовки виробництва реалізується на основі принципів стандартизації і уніфікації технологічних операцій і технологічних процесів із застосуванням універсального верстатного обладнання. При цьому комплект КД на виріб трансформується в формат 3-Dimension (3D) - моделей. 3D-моделі є основою для виготовлення механічних деталей виробу автоматизованим способом на верстатах з числовим програмним управлінням. Складальне виробництво (поверхневий монтаж електрорадіоелементів, виготовлення багатошарових друкованих плат) також реалізується автоматизованим способом на основі маршрутних карт, що перетворюють конструкторські документи в керуючі програми для теплових печей і автоматизованих установників компонентів.	Результатом автоматизации этапов разработки аппаратной и программной реализации изделия являются комплекты программной и конструкторской документации, представляющие собой основу (входные данные) для этапа технологической подготовки производства. Автоматизация технологической подготовки производства реализуется на основе принципов стандартизации и унификации технологических операций и технологических процессов с применением универсального станочного оборудования. При этом комплект КД на изделие трансформируется в формат 3-Dimension (3D)-моделей. 3D-модели служат основой для изготовления механических деталей изделия автоматизированным способом на станках с числовым программным управлением. Сборочное производство (поверхностный монтаж электрорадиоэлементов, изготовление многослойных печатных плат) также реализуется автоматизированным способом на основе маршрутных карт, преобразующих конструкторские документы в управляющие программы для тепловых печей и автоматизированных установщиков компонентов.
	С. 141.	**С. 114.**
	Автоматизація процедур контролю і випробувань виробу здійснюється обчислювальними засобами діагностики, розробляються автоматизованим способом. До складу засобів діагностики входять програмні компоненти діагностичних тестів справності апаратури, що входять до складу комплекту програмного забезпечення (ПЗ) виробу, апаратні компоненти автоматизованих робочих місць (АРМ) з перевірки, налаштування і регулювання апаратури та випробувальне обладнання підприємства. Таким чином, автоматизація основних процедур проектування, виготовлення і випробувань апаратури може бути здійснена на основі інтеграції спеціалізованих компонентів САПР, взаємодіючих між собою за допомогою стандартизованих протоколів передачі інформації. **Коваленко пише «Таким чином», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Плагіат.**	Автоматизация процедур контроля и испытаний изделия осуществляется вычислительными средствами диагностики, разрабатываемыми автоматизированным способом. В состав средств диагностики входят программные компоненты диагностических тестов исправности аппаратуры, входящие в состав комплекта программного обеспечения (ПО) изделия, аппаратные компоненты автоматизированных рабочих мест (АРМ) по проверке, настройке и регулировке аппаратуры и испытательное оборудование предприятия. Такие образом, автоматизация основных процедур проектирования, изготовления и испытаний аппаратуры может быть осуществлена на основе интеграции специализированных компонентов САПР, взаимодействующих между собой посредством стандартизованных протоколов передачи информации.
	С. 141.	**С. 114.**
	Очевидною перевагою створення галузевої САПР є можливість її навчання в процесі експлуатації. В результаті одноразового застосування формуються галузеві бібліотеки елементів автома-	Очевидным преимуществом создания отраслевой САПР является возможность ее обучения в процессе эксплуатации. В результате однократного применения формируются отраслевые библиотеки

	тизації, придатні для повторного використання на підприємствах приладобудівного профілю, уточнюються принципи стандартизації і уніфікації в розробці, виробництві і експлуатації авіаційного приладового обладнання, розробляються засоби, що забезпечують автоматизацію розробки самих засобів автоматизації проектування.Функціональна схема ПЗ типової САПР АРМ розробника БРЕО з різними формами взаємодії «САПРі - файл результату проектування» приведена на рисунок 3.4.	элементов автоматизации, пригодные для повторного использования на предприятиях приборостроительного профиля, уточняются принципы стандартизации и унификации в разработке, производстве и эксплуатации авиационного приборного оборудования, разрабатываются средства, обеспечивающие автоматизацию разработки самих средств автоматизации проектирования. **Функциональная схема программного обеспечения отраслевой САПР и особенности ее применения** Функциональная схема ПО типовой САПР АРМ разработчика БРЭО с различными формами взаимодействия «САПРi – файл результата проектирования» приведена на рис. 4.
	С. 142.	**С. 115.**
	 Рис. 3.4. Функціональна схема ПЗ типової САПР автоматизованого робочого місця розробника БРЕО з різними формами взаємодії «САПР – файл результату проектування».	 Рис. 4. Функциональная схема ПО типовой САПР автоматизированного рабочего места разработчика БРЭО с различными формами взаимодействия «САПР$_i$ – файл результата проектирования»
	С. 142.	**С. 114.**
	До складу ПЗ галузевої САПР входять компоненти взаємодії ПЗ САПР з операційною системою інструментальної ЕОМ, на основі якої організовано АРМ розробника апаратури БРЕО, і спеціалізовані компоненти. Спеціалізовані компоненти представлені сукупністю: – інструментальної складової, що здійснює математичну підтримку вирішення основних проектних завдань і реалізації основних проектних операцій; – робочою складовою, що здійснює діалогову взаємодію розробника апаратури і робочого середовища проектування; – компонентів загального призначення, що реалізують невидимі для користувача операції; – програми-монітора, яка взаємодіє з компонентами операційної системи інструментальної ЕОМ для функціонування універсальних процедур і функцій ПЗ САПР.	В состав ПО отраслевой САПР входят компоненты взаимодействия ПО САПР с операционной системой инструментальной ЭВМ, на основе которой организовано АРМ разработчика аппаратуры БРЭО, и специализированные компоненты. Специализированные компоненты представлены совокупностью: – инструментальной составляющей, осуществляющей математическую поддержку решения основных проектных задач и реализации основных проектных операций; – рабочей составляющей, осуществляющей диалоговое взаимодействие разработчика аппаратуры и рабочей среды проектирования; – компонентов общего назначения, реализующих невидимые для пользователя операции; – программы-монитора, взаимодействующей с компонентами операционной системы инструментальной ЭВМ для функционирования универсальных процедур и функций ПО САПР.
	С. 142–143.	**С. 114.**
	Файл результатів проектування галузевої САПР формується в робочій оболонці САПР і доступний для передачі по локальній мережі підприємства всім спеціалізованим САПРі. Як випливає з рис. 3.5, можливі два різних режими роботи ПЗ САПР з файлами результатів проектування: безпосередній, коли кожне прикладне ПЗ САПРі може обробляти тільки файли, розроблені в даному середовищі проектування, і універсальний, коли файли результатів проектування, підготовлені в різних САПР,	Файл результатов проектирования отраслевой САПР формируется в рабочей оболочке САПР и доступен для передачи по локальной сети предприятия всем специализированным САПРi. Как следует из рис. 4, возможны два различных режима работы ПО САПР с файлами результатов проектирования: непосредственный, когда каждое прикладное ПО САПРi может обрабатывать (читать, редактировать) только файлы, разработанные в данной среде проектирования, и универсальный,

	доступні для обробки прикладного ПЗ будь-якої САПР на основі уніфікованих методів доступу. Для побудови галузевої САПР БРЕО, очевидно, придатний другий (універсальний) варіант. Схема взаємодії різнорідних САПР$_i$ в складі єдиної галузевої САПР авіаційного приладобудування приведена на рисунку 3.6.	когда файлы результатов проектирования, подготовленные в различных САПР, доступны для обработки прикладному ПО любой САПР на основе унифицированных методов доступа. Для построения отраслевой САПР БРЭО, очевидно, пригоден второй (универсальный) вариант. Схема взаимодействия разнородных САПР$_i$ в составе единой отраслевой САПР авиационного приборостроения приведена на рис. 5.
	С. 143.	**С. 115.**
	 Рис. 3.6. Схема організації процесу наскрізного проектування виробів БРЕО на базі локальної обчислювальної мережі підприємства і центрального банку даних проектів підприємства	 Рис. 5. Схема организации процесса сквозного проектирования изделий БРЭО на базе локальной вычислительной сети предприятия и центрального банка данных проектов предприятия
	С. 143–144.	**С. 115.**
	Взаємодія різнорідних САПРі в складі єдиної галузевої САПР БРЕО здійснюється програмним способом на основі інтерфейсів перетворення даних (ІПД). Функція інтерфейсу полягає в нестандартизованих перетвореннях результатів проектування САПР$_i$, що зберігаються в локальному банку даних (ЛБД) цієї САПР, у вид, придатний для введення цих даних в САПР$_{i+1}$ і для подальшої обробки. Функцію перетворення (технологічний перехід) даних q_i в q_{i+1} здійснює оператор f_i (оператор перетворення форматів). Фізичний обмін даними між різнорідними САПР здійснюється через технічні засоби шини локальної обчислювальної мережі підприємства і спеціалізоване мережеве програмне забезпечення (МПЗ).	Взаимодействие разнородных САПРі в составе единой отраслевой САПР БРЭО осуществляется программным способом на основе интерфейсов преобразования данных (ИПД). Функция интерфейса заключается в нестандартизованных преобразованиях результатов проектирования САПР$_i$, хранящихся в локальном банке данных (ЛБД) этой САПР, в вид, пригодный для ввода этих данных в САПР$_{i+1}$ и для последующей обработки. Функцию преобразования (технологический переход) данных q_i в q_{i+1} осуществляет оператор f_i (оператор преобразования форматов). Физический обмен данными между разнородными САПР осуществляется через технические средства шины локальной вычислительной сети предприятия и специализированное сетевое программное обеспечение (СПО).
	С. 144.	**С. 115–116.**
	Отже, всі технічні відділи, які беруть участь в розробці конструкторської, технологічної та програмної документації, мають доступ до локальної обчислювальної мережі підприємства. Доступ здійснюється через підключені уніфіковані АРМ розробників БРЕО. Результати проектування бортового авіаційного обладнання з використанням галузевої САПР зберігаються в центральному банку даних (ЦБД) підприємства, доступ до якого також здійснюється з АРМ розробників через ПС. В ЦБД зберігаються алгоритми і програми функціонування (АПФ) БРЕО, що розробляються підприємством; документація на вироби; нормативно-довідкова інформація (НДІ) – опис виробів, опис матеріалів, опис технологічних процесів; довідкова науково-технічна інформація (ДНТІ) - опис авторських винаходів, патентів, результатів виконання науково-дослідних і дослідно-конструкторських робіт (НДДКР); програмне забезпечення розробки конструкторської та технологічної документації (ПЗ РКТД). **Коваленко пише «Отже», ніби це вона робить якийсь висновок, а насправді переписує чужий**	Как следует из рис. 5, все технические отделы, принимающие участие в разработке конструкторской, технологической и программной документации, имеют доступ к локальной вычислительной сети предприятия. Доступ осуществляется через подключенные унифицированные АРМ разработчиков БРЭО. Результаты проектирования бортового авиационного оборудования с использованием отраслевой САПР хранятся в центральном банке данных (ЦБД) предприятия, доступ к которому также осуществляется с АРМ разработчиков через ИПД. В ЦБД хранятся разрабатываемые предприятием алгоритмы и программы функционирования (АПФ) БРЭО; документация на изделия; нормативно-справочная информация (НСИ) – описание изделий, описание материалов, описание технологических процессов; справочная научно-техническая информация (СНТИ) – описание авторских изобретений, патентов, результатов выполнения научно-исследовательских и опытно-конструкторских работ (НИОКР); программное обеспечение разработки конструкторской и технологической документации (ПО РКТД).

	текст. **Плагіат.**	
	С. 144–145.	**С. 116.**
	Дослідження проблеми побудови галузевих САПР авіаційного приладобудування проводяться сьогодні як у нашій країні, так і за кордоном. Зокрема, відомі роботи [9] в області розробки технологій наскрізного проектування авіаційного обладнання в США, які отримали назву ATIP (Avionics Technology Integrated and Prototyping). Технологія ATIP була вперше використана в якості елемента САПР дослідних стадій проектування компаніями Boeing і Lockheed Mar- tin в процесі виконання НДДКР AVSEP (Avionics Virtual Systems Engineering and Prototyping) при розробці архітектури БРЕО багатоцільового ударного винищувача-бомбардувальника F35 за програмою Joint Strike Fighter [10] . Проведення НДДКР AVSEP дозволило американським розробникам БРЕО визначити основні процедури ітераційного процесу проектування, за допомогою якого проектні організації повинні здійснювати перехід від затвердженої концепції побудови БРЕО до конкретного складу блоків (модулів), що є матеріалізацією технічного вигляду комплексу. **Коваленко переписала чужий текст разом із номерами покликань №9 і 10, під якими в її дисертації знаходяться зовсім інші джерела.** **Фальсифікація джерел.** **Плагіат.**	**Заключение** Исследования проблемы построения отраслевых САПР авиационного приборостроения проводятся сегодня как в нашей стране, так и за рубежом. В частности, известны работы [9] в области разработки технологий сквозного проектирования авиационного оборудования в США, которые получили название ATIP (Avionics Technology Integrated and Prototyping). Технология ATIP была впервые использована в качестве элемента САПР исследовательских стадий проектирования компаниями Boeing и Lockheed Martin в процессе выполнения НИОКР AVSEP (Avionics Virtual Systems Engineering and Prototyping) при разработке архитектуры БРЭО многоцелевого ударного истребителя-бомбардировщика F35 по программе Joint Strike Fighter [10]. Проведение НИОКР AVSEP позволило американским разработчикам БРЭО определить основные процедуры итерационного процесса проектирования, посредством которого проектные организации должны осуществлять переход от утвержденной концепции построения БРЭО к конкретному составу блоков (модулей), являющихся материализацией технического облика комплекса.
	С. 145.	**С. 116.**
	У нашій країні, в сучасних ринкових умовах проблема автоматизації рішення проектних завдань має ключове значення для успішного функціонування підприємства приладобудівного профілю. Створення галузевої САПР авіаційного приладобудування має стратегічну спрямованість і сприяє технічному і програмному переоснащення проектного та виробничого обладнання промислових підприємств. Застосування галузевої САПР авіаційного приладобудування жорстко не пов'язано зі структурою підприємства і можливо практично на будь-якому підприємстві авіаційної галузі. Розглянуті елементи вітчизняної галузевої САПР були апробовані на практиці. Експериментально отримані оцінки тривалості проведення НДДКР з використанням елементів галузевої САПР і без них представлені на рис. 3.6. Не важко бачити, що тривалість етапів проектування комплексу БРЕО з використанням засобів САПР становить 7 років (крива 1), без використання засобів САПР - 9 років (крива 2), із застосуванням САПР і принципів наступності, уніфікації та стандартизації розробок - 4,5 року (крива 3). Таким чином, ефект від автоматизації проектних процедур виражається в скороченні тривалості етапів проектування БРЕО на 2-5 років. **Коваленко пише «Таким чином», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.**	В нашей стране, в современных рыночных условиях проблема автоматизации решения проектных задач имеет ключевое значение для успешного функционирования предприятия приборостроительного профиля. Создание отраслевой САПР авиационного приборостроения имеет стратегическую направленность и способствует техническому и программному переоснащению проектного и производственного оборудования промышленных предприятий. Применение отраслевой САПР авиационного приборостроения жестко не связано со структурой предприятия и возможно практически на любом предприятии авиационной отрасли. Рассмотренные элементы отечественной отраслевой САПР были апробированы на практике. Экспериментально полученные оценки длительности проведения НИОКР с использованием элементов отраслевой САПР и без них представлены на рис. 6. Нетрудно видеть, что длительность этапов проектирования комплекса БРЭО с использованием средств САПР составляет 7 лет (кривая 1), без использования средств САПР – 9 лет (кривая 2), с применением САПР и принципов преемственности, унификации и стандартизации разработок – 4,5 года (кривая 3). Таким образом, эффект от автоматизации проектных процедур выражается в сокращении продолжительности этапов проектирования БРЭО на 2–4,5 года.

	Плагіат.	
6	**Коваленко Ю. Б.** **Інформаційна підтримка процесу проектування та експлуатації комплексу бортового обладнання інтегрованої модульної авіоніки.** – Дис. … доктора технічних наук. – Київ, 2025. (https://duikt.edu.ua/uploads/p_86_93515477.pdf)	**Zieja, M.; Szelmanowski, A.; Pazur, A.; Kowalczyk, G.** **Computer Life-Cycle Management System for Avionics Software as a Tool for Supporting the Sustainable Development of Air Transport** // Sustainability, 2021, 13, 1547. (https://doi.org/10.3390/su13031547) **Нумерація сторінок – за файлом pdf.**
	С. 27.	**С. 1.**
	Сучасні повітряні судна, оснащені різними радіоелектронними бортовими пристроями, які допомагають пілоту виконувати складні повітряні завдання. Стрімкий розвиток цифрової електроніки та інформаційних технологій, що стає все більш виразним протягом останніх років, вимагає від пілота сучасного літака підтримки в польоті бортового обладнання, яке, по суті, вже є «розумними» комп'ютерами зі складною та комплексною авіонікою програмного забезпечення. Така комп'ютеризація сучасного ПС має забезпечити необхідну його експлуатаційну надійність і безпеку як екіпажу, так і пасажирів [1–5]. **Покликання [1–5] – фальшиві, указані в дисертації джерела – це ДСТУ 1995-го та 1998-років, в яких немає тексту про «розумні» комп'ютери.** **Фальсифікація джерел.** **Плагіат.**	Contemporary aircraft, both civilian and military, are equipped with various radio–electronic onboard devices, which support a pilot in executing complex air missions. The rapid development of digital electronics and IT, increasingly distinct over the recent years, requires a pilot of a modern aircraft to have the in-flight support of onboard equipment, which, in fact, are already "smart" computers with sophisticated and comprehensive avionics software. Such computerization of a modern aircraft should ensure its required operating reliability and the safety of both the crew and the passengers [1–4].
	С. 27–28.	**С. 1–2.**
	Процес виробництва цифрових бортових пристроїв і систем забезпечує їх надійну роботу в усіх критичних для них умовах польоту, зокрема через зміни тиску і температури навколишнього повітря та виникнення лінійних перевантажень, наприклад, під час маневрового польоту. Збій програмного забезпечення системи посадки ILS під час заходу літака на посадку в аеропорт у складних метеоумовах .	The process of producing digital onboard devices and systems enables their reliable operation under all flight conditions that are critical to them, among others, due to changes in the pressure and temperature of ambient air and the occurrence of linear overloads, e.g., during a maneuvering flight. It is quite easy to imagine the consequences of, e.g., ILS landing-system software failure during an aircraft's approach to an airport in difficult weather conditions or damage to a helmet-mounted weaponry control system when indicating a target, selecting weapon type, or using it in the course of a combat flight.
	С. 28.	**С. 2.**
	Щоб задовольнити суворі вимоги щодо надійності в процесі розробки електронного повітряного обладнання та спеціального програмного забезпечення для авіоніки, спеціалісти склали відповідні стандарти, такі як стандарт DO-178C, що містить вимоги до програмного забезпечення, та стандарт DO-254, що містить апаратне забезпечення. вимоги, підкріплені додатковими документами стандартизації, включаючи ARP 4761 і ARP 4754A (рис. 1.1).	In order to satisfy the stringent reliability-related requirements within the process of developing electronic air equipment and their dedicated avionics software, specialists have drawn up relevant standards, such as standard DO-178C, containing software requirements, and standard DO-254, containing hardware requirements, supported by additional standardizing documents, including ARP 4761 and ARP 4754A (Figure 1).

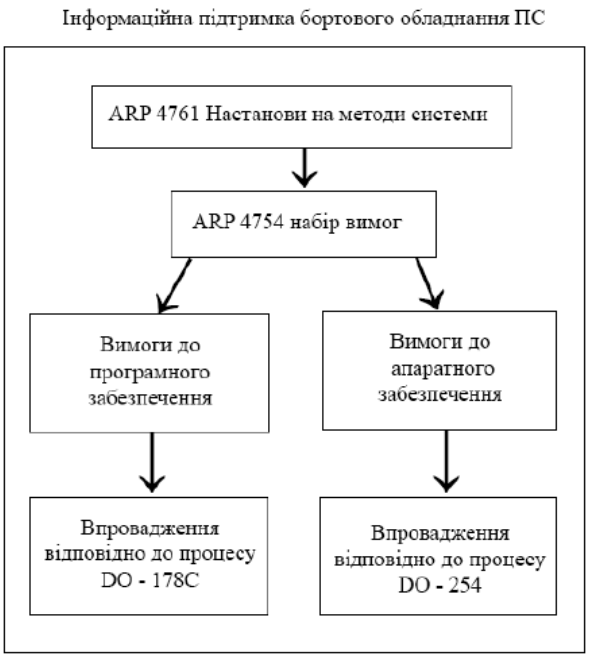
<table>
<tr>
<td></td>
<td> 

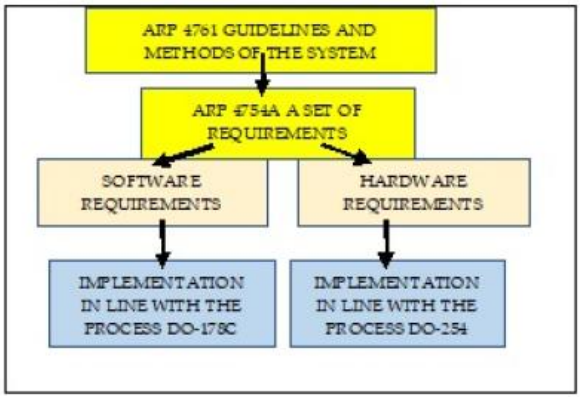
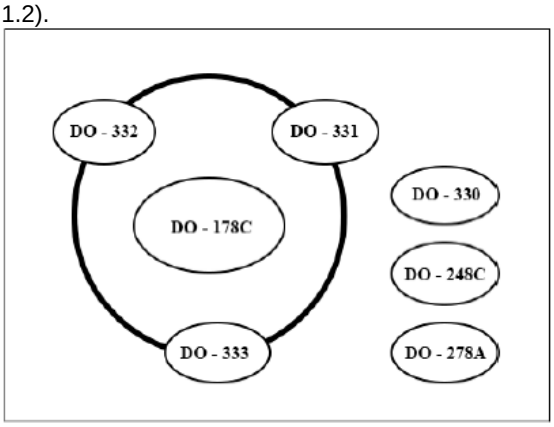
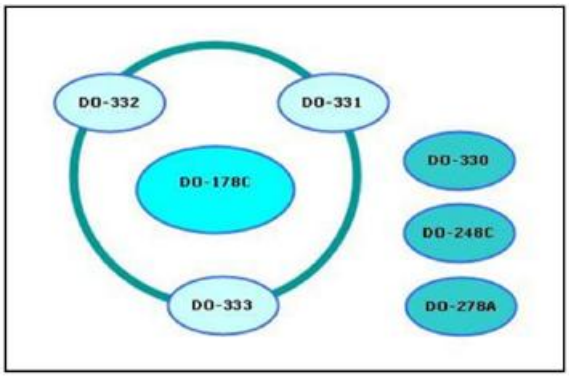
Рис. 1.1 Інформаційна підтримка процесу проектування авіоніки та програмного забезпечення, критичного для безпеки польотів, відповідно до стандарту DO-178C.</td>
<td> 

Figure 1. Manufacturing process diagram for avionic devices and software critical to flight safety, as per standard DO-178C.</td>
</tr>
<tr>
<td></td>
<td>С. 28–29.</td>
<td>С. 2.</td>
</tr>
<tr>
<td></td>
<td>Стандарт ARP 4761 — це набір процесів і методів, які використовуються для забезпечення необхідного рівня надійності розроблених пристроїв і систем; призначений для встановлення на борту ПС. Цей стандарт визначає п'ять рівнів пошкодження бортового обладнання та їх наслідки [6]. Рівень А охоплює збитки, виникнення яких може призвести до серйозних жертв і матеріальних втрат. Рівень В покриває значні збитки, які можуть призвести до серйозних травм. Рівень С охоплює значні пошкодження нижчого ступеня опромінення, що означає відмову основних бортових систем, але дозволяється продовження польоту . Рівень D покриває незначні пошкодження, наприклад, коли дозволяється продовжувати політ без певної функції. Рівень Е охоплює пошкодження, виникнення яких є незначним для процесу продовження польоту та дій пілота. Стандарт ARP 4754A — це набір функціональних системних вимог щодо програмного та апаратного забезпечення.

Коваленко переписала текст із англійської статті 2021-го року.
Фальшиве покликання [6] – це ДСТУ 1999 р.
Фальсифікація джерел.
Плагіат.</td>
<td>Standard ARP 4761 is a set of processes and techniques used to ensure the required assurance level of designed devices and systems; it is intended for installation on board aircraft. This standard defines five damage levels for onboard equipment and their consequences [5]. Level A covers damage, the occurrence of which may lead to severe casualties and material losses (e.g., air crash). Level B covers major damage that may lead to severe injuries (e.g., injury of a pilot or passengers). Level C covers major damage of a lower exposure degree, meaning the failure of main onboard systems but the flight being allowed to continue (e.g., through manual steering). Level D covers minor damage, e.g., when the flight is allowed to continue without a specific functionality (e.g., main radio damage). Level E covers damage, the occurrence of which is insignificant to the flight continuation process and pilot’s actions (e.g., mechanical wear of steering element buttons). Standard ARP 4754A is a set of functional system requirements in terms of software and hardware.</td>
</tr>
<tr>
<td></td>
<td>С. 29.</td>
<td>С. 2.</td>
</tr>
<tr>
<td></td>
<td>Стандарт DO-178C є основним документом, що застосовується на міжнародному авіаційному ринку до процесу виробництва та сертифікації програмного забезпечення бортових електронних пристроїв. Він складається з базової частини та додаткових стандартів DO-331, DO-332 та DO-333, а також DO-330, DO-248C та DO-278A, які визначають додаткові вимоги та дії, які мають бути передбачені програмним забезпеченням. процесу розробки в за-</td>
<td>Standard DO-178C is the basic document applied within the international aviation market to the process of manufacturing and certifying the software of onboard electronic devices. It consists of a basic part and supplementary standards DO-331, DO-332, and DO-333 as well as DO-330, DO-248C, and DO-278A, which define additional requirements and activities to be provided for in the software development process, depending on the adopted implementation technique</td>
</tr>
</table>

	лежності від прийнятої методики реалізації (рис. 1.2).  Рис. 1.2. Модель взаємозв'язків між додатковими стандартами, які визначають допоміжні вимоги та види діяльності відповідно до стандарту DO-178C.	(Figure 2).  Figure 2. Relationships between supplementary standards that define additional requirements and activities under standard DO-178C.
	С. 30.	**С. 3.**
	Стандарт DO-331 описує розробки програмного та апаратного забезпечення з використанням техніки моделювання (проектування на основі моделі). Обчислювальні алгоритми або їх фрагменти реалізуються за допомогою зовнішніх засобів, а процес проектування виконується шляхом об'єднання блоків, які виконують різноманітні математичні операції. Реалізована таким чином обчислювальна діаграма підлягає тестуванню та перевірці. Заключним етапом реалізації алгоритму є автоматичне формування діаграми на основі розробленого вихідного коду програмного забезпечення. При використанні цього методу обчислювальні алгоритми реалізуються візуально з автоматичною генерацією результуючого програмного коду. Стандарт DO-332 застосовується при реалізації проектів з використанням об'єктно-орієнтованих мов програмування (включаючи C++, Java, Ada). Фундаментальним питанням у випадку об'єктно-орієнтованих мов є перевірка такого програмного забезпечення, яка дотримується трьох основних методів, а саме: успадкування, поліморфізму та динамічного зв'язування. Втіленням такого стандарту може бути наявність ще одного методу з такою ж назвою в межах тієї самої ієрархії класів, що гарантує, що під час виклику відповідного класу виконується відповідний метод.	Standard DO-331 describes software and hardware designs using a modeling technique (model-based designs). Computational algorithms or their fragments are implemented using external tools, and the designing process is executed by combining blocks that execute various mathematical operations. A computation diagram implemented this way is subject to testing and verification. The final stage of algorithm implementation is the automatic generation of a diagram based on the developed software source code. When using this method, computational algorithms are implemented visually, with automatic generation of the resultant software code. Standard DO-332 is applied when projects are implemented with the use of objectoriented programming languages (including C++, Java, Ada). The fundamental issue in the case of object-oriented languages is the verification of such software, which follows three basic techniques, namely, inheritance, polymorphism, and dynamic linking. An embodiment of such a standard can be the occurrence of yet another method of the same name within the same class hierarchy, which ensures that when a relevant class is called, the appropriate method is executed.
	С. 30–31.	**С. 3.**
	Стандарт DO-333 застосовується, коли в процесі проектування використовуються формальні методи. Формальні методи – це математичні прийоми, які застосовуються в процесі розробки та перевірки програмного забезпечення. Завдання формальних методів полягає в тому, щоб розробити математичну систему для побудованої системи та перевірити її поведінку, щоб довести, що система, що будується, є функціональною та задовольняє передбачуваним вимогам безпеки. Стандарт DO-333 описує процес кваліфікації програмних засобів, які використовуються в процесі розробки, тестування, запуску та модифікації програмного забезпечення. Процес кваліфікації розуміється як оцінка програмних засобів, результати яких не перевіряються і які спрощують, прискорюють і автоматизують стандартний процес DO-178C.	Standard DO-333 is applied when formal methods are used within the designing process. Formal methods are mathematical techniques, which are applied within the software development and verification process. The task of formal methods is to develop a mathematical system for the constructed system and to verify its behavior in order to prove that the system under construction is functional and satisfies the assumed safety requirements. Standard DO-333 describes the qualification process for software tools used within the process of software development, testing, launching, and modification. A qualification process is understood as an evaluation of software tools, the output of which is not verified and which simplify, accelerate, and automate a DO-178C standard process.

	С. 31.	**С. 3.**
	Стандарт DO-248C є додатковим і додатковим документом до стандарту DO-178C, що пояснює його незрозумілі та проблемні аспекти. Вищезазначені вимоги стандарту DO-178C стали основою для побудови комп'ютерної системи, що підтримує управління життєвим циклом програмного забезпечення авіоніки.	Standard DO-248C is an additional and supplementary document for the DO-178C standard, explaining its incomprehensible and problematic aspects. The above requirements of the DO-178C standard became the basis for the ITWL to build a computer system supporting the lifecycle management of avionic software, developed and certified for electronic devices, where the main goal is to develop software and implement it into the SWPL-1 CYKLOP helmet flight-parameter display system [6–8].
	С. 31.	**С. 3.**
	Життєвий цикл програмного забезпечення авіоніки пов'язаний із виконанням численних різноманітних типів взаємопов'язаних завдань, метою яких є проектування та розробка програмного забезпечення відповідної якості, що відповідає вимогам замовника, а також забезпечення необхідного рівня безпеки розроблені авіонічні системи та пристрої [1–9]. Стандарт DO-178C визначає три основні проекти, а саме планування, розробку та інтеграцію програмного забезпечення, які є взаємно паралельними у вибраних областях. Це означає, що початок одного процесу необов'язково пов'язувати із завершенням попереднього. Прикладом може бути початок тестування закодованого програмного фрагмента до етапу повного завершення кодування . **Покликання [1–7] – це ДСТУ 1995–2000 рр.** **Покликання [8] – це:** Durak U., Rausch A., et al. Digital twin of iron bird test rig with flight simulator for testing avionics and system integration // DLR (German Aerospace Center), 2022. URL: https://elib.dlr.de/195403/ **Покликання [9] – це:** Li L., Digital Twin in Aerospace Industry: A Gentle Introduction, IEEE Access, 2021. DOI: 10.1109/ACCESS.2021.3136458. **Усі ці покликання фальшиві, насправді Коваленко переписала текст із англійської статті 2021-го року.** **Фальсифікація джерел.** **Плагіат.**	**2. Materials and Methods** The lifecycle of avionics software is associated with the implementation of numerous, various types of interlinked tasks, the objective of which is to design and develop software of appropriate quality that meets the requirements of the ordering party as well as to ensure the required security level of the designed avionic systems and devices [9–11]. Standard DO-178C defines three primary projects, namely, software planning, development, and integration, which are mutually concurrent within selected areas. This means that the commencement of one process does not have to be linked with the completion of a previous one. An example would be starting the tests of an encoded software fragment prior to the stage of total encoding completion [12–17]. **Покликання [12] – це:** Anton, P.S.; Anderson, R.H.; Mesic, R.; Scheiern, M. Finding and Fixing Vulnerabilities in Information Systems: The Vulnerability Assessment and Mitigation Methodology; RAND: Pittsburgh, PA, USA, 2003. **Покликання [13] – це:** Kasprzyk, R.; Stachurski, A. A concept of standard-based vulnerability management automation for IT systems. Comput. Sci. Math. Model. 2016, 3, 33–38. **Покликання [14] – це:** The MITRE Corporation. Common Weakness Scoring System (CWSS). 2014. **Покликання [15] – це:** Alhazmi, O.H.; Malaiya, Y.K.; Ray, I. Measuring, analyzing and predicting security vulnerabilities in software systems. Comput. Secur. 2007, 26, 219–228. **Покликання [16] – це:** Alhazmi, O.H.; Malaiya, Y.K. Application of Vulnerability Discovery Models to Major Operating Systems. IEEE Trans. Reliab. 2008, 57, 14–22. **Покликання [17] – це:** Moreno, J.A. AQAP 2105 NATO Requirements Required a Quality Plan for a Product Constituting a Contract Subject); NSO: Brussels, Belgium, 2009.
	С. 31.	**С. 3.**
	Існує багато моделей життєвого циклу програмного забезпечення, які включають каскадну модель. У цій моделі розрізняють наступні основні кроки, які необхідно виконати, щоб відповідати вимогам стандарту DO178C. Вони передбачають збір вимог і їх аналіз, а також проектування, розробку, тестування та впровадження програмного забезпечення. Кожен із цих кроків дозволяє нам повернутися до попереднього стану, забезпечуючи взаємодію між виконуваними завданнями, що особливо важливо у разі виявлення програмних помилок, які не-	There are many software lifecycle models, which include the cascade model. The following major steps, to be executed in order to meet the requirements of standard DO-178C, are distinguished within this model. They involve collecting the requirements and their analyses as well as software design, development, testing, and implementation. Each of these steps allows us to go back to the previous state, enabling interaction between the executed tasks, which is of particular importance in the case of detecting software bugs that have to be fixed. The implementation of the developed avionics soft-

	обхідно виправити. Впровадження розробленого програмного забезпечення авіоніки також охоплює його експлуатацію до моменту утилізації.	ware also covers its operation until its disposal.
	С. 31–32.	**С. 3–4.**
	Планування програмного забезпечення включає процеси, пов'язані з плануванням і стандартами, після яких слідує розробка програмного забезпечення. На цьому етапі стандарт DO-178C визначає п'ять основних етапів і три стандарти програмного забезпечення. Він також виділяє інші документи, які необхідні для сертифікації будь-якого розробленого програмного забезпечення [18–20]. Основні етапи програмного забезпечення авіоніки такі: – Етап сертифікації аспектів програмного забезпечення (PSAC): свого роду «контракт» між підрядником і органом сертифікації. – Етап розробки програмного забезпечення (SDP): містить вимоги щодо планування програмного забезпечення, кодування та етапів інтеграції. SDP був написаний для розробників програмного забезпечення і є свого роду керівництвом щодо розробки програмного забезпечення, щоб воно відповідало прийнятим вимогам. **Коваленко переписала англійську статтю 2021-го р. разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела:** **Покликання [18] – це:** Ghanjaoui Y., Fuchs M., Biedermann J., Nagel B.«Model-based design and multidisciplinary optimization of complex system architectures in the aircraft cabin», CEAS Aeronautical Journal, 2023, Vol. 14(4), pp. 887–905. **Покликання [19] – це:** Konakhovych, G. "Research of variants of construction of a system of technical maintenance and repair in aircraft engineering", Bulletin of the Engineering Academy of Ukraine, Issue 1, pp. 52-56, 2012. **Покликання [20] – це:** Міляєв Ю., Нечипоренко О. , Основи надійності технічних систем: навч. посіб. К.: Видавн.-полігр. центр Акад. муніцип. управління, 2008, 246 с. **У цьому фрагменті її дисертації <u>всі покликання фальшиві.</u>** **Фальсифікація джерел.** **Плагіат.**	Software planning involves processes associated with planning and standards, after which software development will follow. At this stage, standard DO-178C defines five major plans and three software standards. It also distinguishes other documents that are required for the certification of any developed software [18–20]. The major avionics software plans are as follows: ● Plan for Software Aspects of Certification (PSAC): a kind of "contract" between the contractor and the certification body. ● Software Development Plan (SDP): contains the requirements concerning software planning, coding, and integration stages. The SDP was written for software developers and is, a kind of a guide on how to develop the software in order for it to satisfy the adopted requirements. **Покликання [18] – це:** Maj, J. AQAP 2210 (NATO Supplementary Requirements for AQAP 2110 on Ensuring Software Quality); NSO: Brussels, Belgium, 2006. **Покликання [19] – це:** Borowski, J. Quality Plan for an IT Project. SWPL-1 System Software; AFIT: Warsaw, Poland, 2015. **Покликання [20] – це:** Michalak, S. Software Documentation for SWPL-1 Flight Parameter Display System; AFIT: Warsaw, Poland, 2015.
	С. 32.	**С. 4.**
	– Етап перевірки програмного забезпечення (SVP): містить аспекти, пов'язані з перевіркою функціональності програмного забезпечення, і призначений для тестувальників програмного забезпечення. SVP асоціюється з SDP, тому що на цьому етапі перевіряються припущення, які були продумані на етапі розробки програмного забезпечення. – Етап керування конфігурацією програмного забезпечення (SCMP): визначає процедури, інструменти та методи, спрямовані на досягнення цілей, пов'язаних із керуванням вимогами протягом усього життєвого циклу програмного забезпечення. Він охоплює процедури з точки зору визначення базової версії та ідентифікації версій програмного забезпечення, звітування про проблеми, контролю та перегляду модифікацій, архівування, контролю завантаження та відновлення програмного забезпечення.	● Software Verification Plan (SVP): contains aspects associated with verifying software functionality and is intended for software testers. The SVP is associated with the SDP because the assumptions that were thought out at the software development stage are verified at this stage. ● Software Configuration Management Plan (SCMP): defines procedures, tools, and methods aimed at achieving the objectives associated with managing the requirements throughout the entire software lifecycle. It covers the procedures in terms of defining the baseline version and identifying software versions, reporting issues, controlling and reviewing modifications, archiving, controlling software loading and recovery.
	С. 32.	**С. 4.**
	– Етап забезпечення якості програмного забез-	● Software Quality Assurance Plan (SQAP): de-

	печення (SQAP): визначає процедури та методи, які мають застосовуватися, щоб задовольнити вимоги до якості, пов'язані зі стандартом DO-178C. Він визначає процедури з точки зору управління якістю, виконання аудиту, дії, пов'язані зі звітуванням про проблему, і методологію коригувальних дій. – Основні стандарти розробки програмного забезпечення БРЕО: **Коваленко переписує чужий текст, не розуміючи його. Вступне речення перед переліком перетворюється в неї в один із пунктів цього переліку: «**– Основні стандарти розробки програмного забезпечення БРЕО:**».** **Недолугий плагіат.**	fines the procedures and methods to be applied in order to satisfy the quality requirements associated with standard DO-178C. It determines the procedures in terms of quality management, audit execution, actions associated with issue reporting, and corrective action methodology. The major software development standards are as follows:
	С. 33.	**С. 4.**
	– Стандарти вимог до програмного забезпечення (SRS), які визначають принципи, методи та інструменти, що застосовуються для розробки вимог високого рівня. Вони включають методи, що використовуються для розробки програмного забезпечення, позначення для реалізації вимог (алгоритми, блок-схеми), обмеження інструментів проекту, які будуть використовуватися для розробки програмного забезпечення, а також критерії, пов'язані з вимогами. – Стандарти розробки програмного забезпечення (SDS): вони визначають методи, інструменти та обмеження в процесі розробки програмного забезпечення. Вони включають вимоги низького рівня та архітектуру програмного забезпечення. Вони призначені для команди розробників програмного забезпечення та пояснюють, як ефективно реалізувати проект. Вони охоплюють, серед іншого, методи номенклатури, обмеження інструментів проектування та складність програмного забезпечення (наприклад, тривалість процедури).	● Software Requirements Standards (SRS), which define principles, methods, and tools to be applied for developing high-level requirements. They include methods used for software development, notations for requirement implementation (algorithms, flow diagrams), and project tool limitations, which will be used for software development, as well as criteria related to the requirements. ● Software Design Standards (SDSs): They define methods, tools, and limitations within the software design process. They include low-level requirements and software architecture. They are intended for a software development team and explain how to implement a design effectively. They cover, among other things, nomenclature methods, design tool limitations, and software complexity (e.g., procedure length).
	С. 33.	**С. 4.**
	– Стандарти програмного кодування (SCS); Вони визначають методи, інструменти та обмеження в процесі кодування програмного забезпечення. Вони включають, серед іншого, стандарти кодування, використовувані мови програмування, стандарти представлення коду, стандарти номенклатури, обмеження компілятора та обмеження, що випливають із стандартів програмування. – Вищезазначені документи, як еталонні шаблони, реалізовані в рамках побудованої комп'ютерної системи, яка підтримує процес управління розробка та сертифікація програмного забезпечення авіоніки. – Процес розробки програмного забезпечення авіоніки складається з чотирьох детальних процесів: – Процес вимог до програмного забезпечення, який призводить до розробки вимог високого рівня (HLR); **Коваленко переписує чужий текст, не розуміючи його. Вступне речення перед переліком перетворюється в неї в один із пунктів цього переліку: «**– Процес розробки програмного забезпечення авіоніки складається з чотирьох детальних процесів:**».** **Недолугий плагіат.**	● Software Coding Standards (SCSs); They define methods, tools, and limitations within the software coding process. They include, among other things, coding standards, programming languages used, code presentation standards, nomenclature standards, compilator limitations, and restrictions arising from programming standards. ● The aforementioned documents, as reference templates, have been implemented within the constructed computer system, which supports the process of managing avionics software development and certification. 2.2. Avionics Software Development Process and Possible Computer-Aided Support The avionics software development process consists of four detailed processes: ● The software requirement process, which leads to the development of high-level requirements (HLRs);

	С. 34.	**С. 4.**
	– Процес проектування програмного забезпечення, який розробляє вимоги низького рівня (LLR) і архітектуру програмного забезпечення на основі HLR; – Процес кодування, який призводить до створення вихідного коду та неінтегрованого об'єктного коду; – Процес інтеграції програмного забезпечення, який передбачає консолідацію програмного забезпечення у формі виконуваних програм та його інтеграцію із зовнішніми пристроями.	● The software design process, which develops low-level requirements (LLRs) and software architecture based on HLRs; ● The coding process, which leads to the creation of a source code and a nonintegrated object code; ● The software integration process, which involves consolidating the software into the form of executable programs and its integration with external devices.
	С. 34.	**С. 4–5.**
	Вимоги високого рівня (HLR) реалізуються на основі архітектури системи та системних вимог. Вони включають часові сигнали, керування пам'яттю, заплановані зв'язки із зовнішніми пристроями, методи реагування та виявлення помилок, моніторинг роботи системи та розділення програмного забезпечення. HLR є основою для розробки вимог низького рівня, які використовуються в процесі проектування програмного забезпечення, які включають описи зв’язків із зовнішніми пристроями, визначення та спосіб потоку даних, механізми зв’язку та компоненти програмного забезпечення. Процес кодування включає переклад LLR у вихідний код і попередньо скомпільований об’єктний код. Це пов'язано з процесом верифікації, оскільки на цьому етапі відбувається попереднє виконання частково розробленого коду. Процес інтеграції передбачає компіляцію та об’єднання скомпільованого коду у виконувані програми і вбудовування цього програмного забезпечення на цільову платформу.	High-level requirements (HLRs) are implemented based on system architecture and system requirements. They involve time waveforms, memory management, planned links with external devices, methods of responding and detecting errors, system operation monitoring, and software partitioning. HLRs constitute a base to develop low-level requirements used in the software design process, which include descriptions of connections with external devices, definitions and manner of data flow, communication mechanisms, and software components. The coding process involves translating LLRs into source code and a precompiled object code. It is associated with the verification process because the preliminary execution of a partially developed code is conducted at this stage. The integration process involves compiling and combining the compiled code into executable applications (one or more) and embedding this software on the target platform (the onboard device).
	С. 34–35.	**С. 5.**
	Процеси розробки програмного забезпечення визначають один або багато рівнів системних вимог. Вимоги високого рівня визначаються безпосередньо на основі архітектури системи та системних вимог. Вони розробляються в процесі проектування програмного забезпечення, таким чином створюючи взаємопов’язані вимоги низького рівня. Однак, коли вихідний код створюється безпосередньо на основі вимог високого рівня, він також відповідає вимогам низького рівня та підлягає рекомендаціям щодо вимог низького рівня. Процеси вимог до програмного забезпечення використовують вихідні дані процесу життєвого циклу програмного забезпечення для створення вимог високого рівня. Основним результатом цього процесу є дані про вимоги до програмного забезпечення.	Software development processes determine one or many system requirement levels. High-level requirements are determined based directly on system architecture and system requirements. They are developed within the software design process, thus creating interrelated low-level requirements. However, when a source code is generated based directly on high-level requirements, it also corresponds to low-level requirements and is subject to recommendations on low-level requirements. Software requirement processes utilize software lifecycle process outputs to create high-level requirements. The basic result of this process is software requirement data.
	С. 35.	**С. 5.**
	Дані про вимоги до програмного забезпечення визначають вимоги високого рівня, включаючи вимоги, надані замовником. Дані повинні включати опис розподілу вимог до програмної системи, враховуючи вимоги безпеки та потенційні умови помилок, функціональні та робочі вимоги для кожного режиму роботи, критерії продуктивності (наприклад, точність і точність), вимоги та обмеження, пов’язані з часом, обмеження розміру пам’яті, апаратні та програмні інтерфейси (наприклад, протоколи, формати, частота вводу/виводу), виявлення помилок, моніторинг безпеки, а також вимоги до розділення програмного забезпечення (як окремі компоненти програмного забезпечення взаємодіють) і рівні програмного забезпечення для кожного	Software requirement data define high-level requirements, including the requirements provided by the ordering party. The data should include a description of the software system requirement allocation, taking into account the safety requirements and potential error conditions, functional and operational requirements for each operating mode, performance criteria (e.g., precision and accuracy), time-related requirements and limitations, memory size limitations, hardware and software interfaces (e.g., protocols, formats, input/output frequency), error detection, safety monitoring, as well as the requirements of software partitioning (how separated software components cooperate) and software levels for each component. The input into the avionics software design process

	компонента. Вхідні дані в процес розробки програмного забезпечення авіоніки включають вимоги до програмного забезпечення, план розробки програмного забезпечення та стандарти розробки програмного забезпечення. Якщо заплановані критерії переходу виконуються, вимоги високого рівня використовуються в процесі виробництва для створення архітектури програмного забезпечення та вимоги низького рівня. Вони можуть включати один або декілька рівнів вимог.	includes software requirements, a software development plan, and software design standards. If the planned transition criteria are met, the high-level requirements are used within the manufacturing process for creating software architecture and low-level requirements. They may include one or more requirement levels.
	С. 35–36.	**С. 5.**
	Основним результатом цих процесів є опис проекту, який містить архітектуру програмного забезпечення та вимоги низького рівня. Дані повинні включати детальний опис того, як програмне забезпечення задовольняє вимоги високого рівня, включаючи алгоритми та структуру даних, і як вимоги до програмного забезпечення відповідають процесам і завданням. Він також повинен надавати опис архітектури програмного забезпечення, що визначає структуру програмного забезпечення, із реалізованими вимогами, описами вводу/виводу (наприклад, словник даних, дані та потік керування в рамках проекту), обмеження ресурсів, стратегію управління ресурсами та їх обмеженнями, а також запаси як а також методи вимірювання цих запасів. Наприклад, час і пам'ять, процедури послідовності, опис має включати внутрішньопроцесорні та внутрішньозадачні механізми зв'язку, включаючи фіксовані часові послідовності переривання, методи проектування та деталі їх реалізації, наприклад завантаження програмного забезпечення. Важливим елементом опису є програмне забезпечення, модифіковане користувачем, методи розділення та заходи щодо запобігання злому розділу, а також опис компонентів програмного забезпечення незалежно від того, чи є вони новими чи раніше виготовленими і посилання на базову версію, з якої вони були створені. завантажено. Цей опис має також включати похідні вимоги, що випливають із процесу розробки програмного забезпечення. Якщо система містить неактивний код, опис заходів безпеки для активації коду на цільовому комп'ютері та обґрунтування проектних рішень безпосередньо включені до системних вимог, пов'язаних із її безпекою.	The basic output in these processes is the design description, which contains software architecture and low-level requirements. The data should include a detailed description of how the software satisfies high-level requirements, including algorithms and data structure, and how the software requirements correspond to the processes and tasks. It should also provide the software architecture description defining software structure, with implemented requirements, input/output descriptions (e.g., data dictionary, data and control flow within the design), resource limitations, a management strategy for resources and their limitations, and margins as well as methods for measuring these margins (e.g., time and memory, sequencing procedures). The description should include intraprocessors and intratask communication mechanisms, including fixed interruption time sequences, design methods, and details on their implementation (e.g., software loading). An important element of the description is user-modified software, partitioning methods, and measures to prevent partition breach as well a description of software components (regardless of whether they are new or previously manufactured) and a reference to the baseline version from which they were downloaded. This description should also include derivative requirements arising from the software design process. If a system contains an inactive code, a description of security measures for the activation of the code on the target computer and a justification of design decisions are directly included in the system requirements associated with its security.
	С. 274.	**С. 4.**
	5.2. Процес розробки програмного забезпечення та автоматизована інформаційна підтримка IMA. Процес розробки програмного забезпечення авіоніки складається з чотирьох детальних процесів: – Процес вимог до програмного забезпечення, який призводить до розробки вимог високого рівня (HLR); – Процес проектування програмного забезпечення, який розробляє вимоги низького рівня (LLR) і архітектуру програмного забезпечення на основі HLR; – Процес кодування, який призводить до створення вихідного коду та неінтегрованого об'єктного коду; – Процес інтеграції програмного забезпечення, який передбачає консолідацію програмного забезпечення у формі виконуваних програм та його інтеграцію із зовнішніми пристроями.	*2.2. Avionics Software Development Process and Possible Computer-Aided Support* The avionics software development process consists of four detailed processes: ● The software requirement process, which leads to the development of high-level requirements (HLRs); ● The software design process, which develops low-level requirements (LLRs) and software architecture based on HLRs; ● The coding process, which leads to the creation of a source code and a nonintegrated object code; ● The software integration process, which involves consolidating the software into the form of executable programs and its integration with external devices.

	Коваленко знову повторює текст, який уже був переписаний на с. 33–34. **Повторний плагіат.**	
	С. 274.	**С. 4–5.**
	Вимоги високого рівня (HLR) реалізуються на основі архітектури системи та системних вимог. Вони включають часові сигнали, керування пам'яттю, заплановані зв'язки із зовнішніми пристроями, методи реагування та виявлення помилок, моніторинг роботи системи та розділення програмного забезпечення [90-92]. HLR є основою для розробки вимог низького рівня, які використовуються в процесі проектування програмного забезпечення, які включають описи зв'язків із зовнішніми пристроями, визначення та спосіб потоку даних, механізми зв'язку та компоненти програмного забезпечення. **Коваленко знову повторює текст, який уже був переписаний на с. 34, однак на цей раз додала фальшиві покликання [90–92], яких немає в оригінальній статті.** **Фальсифікація джерел.** **Повторний плагіат.**	High-level requirements (HLRs) are implemented based on system architecture and system requirements. They involve time waveforms, memory management, planned links with external devices, methods of responding and detecting errors, system operation monitoring, and software partitioning. HLRs constitute a base to develop low-level requirements used in the software design process, which include descriptions of connections with external devices, definitions and manner of data flow, communication mechanisms, and software components.
	С. 275.	**С. 5.**
	Процес кодування включає переклад LLR у вихідний код і попередньо скомпільований об'єктний код. Це пов'язано з процесом верифікації, оскільки на цьому етапі відбувається попереднє виконання частково розробленого коду. Процес інтеграції передбачає компіляцію та об'єднання скомпільованого коду у виконувані програми (одну або декілька) і вбудовування цього програмного забезпечення на цільову платформу (бортовий пристрій). Процеси розробки програмного забезпечення визначають один або багато рівнів системних вимог. Вимоги високого рівня визначаються безпосередньо на основі архітектури системи та системних вимог. Вони розробляються в процесі проектування програмного забезпечення, таким чином створюючи взаємопов'язані вимоги низького рівня. Однак, коли вихідний код створюється безпосередньо на основі вимог високого рівня, він також відповідає вимогам низького рівня та підлягає рекомендаціям щодо вимог низького рівня. Процеси вимог до програмного забезпечення використовують вихідні дані процесу життєвого циклу програмного забезпечення для створення вимог високого рівня. Основним результатом цього процесу є дані про вимоги до програмного забезпечення **Коваленко знову повторює текст, який уже був переписаний на с. 34–35.** **Повторний плагіат.**	The coding process involves translating LLRs into source code and a precompiled object code. It is associated with the verification process because the preliminary execution of a partially developed code is conducted at this stage. The integration process involves compiling and combining the compiled code into executable applications (one or more) and embedding this software on the target platform (the onboard device). Software development processes determine one or many system requirement levels. High-level requirements are determined based directly on system architecture and system requirements. They are developed within the software design process, thus creating interrelated low-level requirements. However, when a source code is generated based directly on high-level requirements, it also corresponds to low-level requirements and is subject to recommendations on low-level requirements. Software requirement processes utilize software lifecycle process outputs to create high-level requirements. The basic result of this process is software requirement data.
	С. 275–276.	**С. 5.**
	Дані про вимоги до програмного забезпечення визначають вимоги високого рівня, включаючи вимоги, надані замовником. Дані повинні включати опис розподілу вимог до програмної системи, враховуючи вимоги безпеки та потенційні умови помилок, функціональні та робочі вимоги для кожного режиму роботи, критерії продуктивності, вимоги та обмеження, пов'язані з часом, обмеження розміру	Software requirement data define high-level requirements, including the requirements provided by the ordering party. The data should include a description of the software system requirement allocation, taking into account the safety requirements and potential error conditions, functional and operational requirements for each operating mode, performance criteria (e.g., precision and accuracy), time-related

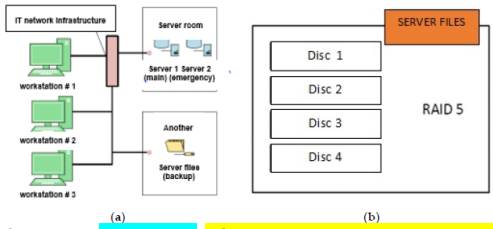
	пам'яті, апаратні та програмні інтерфейси, виявлення помилок, моніторинг безпеки, а також вимоги до розділення програмного забезпечення рівні програмного забезпечення для кожного компонента. Вхідні дані в процес розробки програмного забезпечення авіоніки включають вимоги до програмного забезпечення, план розробки програмного забезпечення та стандарти розробки програмного забезпечення. Якщо заплановані критерії переходу виконуються, вимоги високого рівня використовуються в процесі виробництва для створення архітектури програмного забезпечення та вимоги низького рівня. Вони можуть включати один або декілька рівнів вимог. **Коваленко знову повторює текст, який уже був переписаний на с. 35.** **Повторний плагіат.**	requirements and limitations, memory size limitations, hardware and software interfaces (e.g., protocols, formats, input/output frequency), error detection, safety monitoring, as well as the requirements of software partitioning (how separated software components cooperate) and software levels for each component. The input into the avionics software design process includes software requirements, a software development plan, and software design standards. If the planned transition criteria are met, the high-level requirements are used within the manufacturing process for creating software architecture and low-level requirements. They may include one or more requirement levels.
	С. 276.	**С. 5.**
	Основним результатом цих процесів є опис проекту, який містить архітектуру програмного забезпечення та вимоги низького рівня. Дані повинні включати детальний опис того, як програмне забезпечення задовольняє вимоги високого рівня, включаючи алгоритми та структуру даних, і як вимоги до програмного забезпечення відповідають процесам і завданням. Він також повинен надавати опис архітектури програмного забезпечення, що визначає структуру програмного забезпечення, із реалізованими вимогами, описами вводу/виводу (наприклад, словник даних, дані та потік керування в рамках проекту), обмеження ресурсів, стратегію управління ресурсами та їх обмеженнями, а також запаси як а також методи вимірювання цих запасів (наприклад, час і пам'ять, процедури послідовності). Опис має включати внутрішньопроцесорні та внутрішньозадачні механізми зв'язку, включаючи фіксовані часові послідовності переривання, методи проектування та деталі їх реалізації (наприклад, завантаження програмного забезпечення). Важливим елементом опису є програмне забезпечення, модифіковане користувачем, методи розділення та заходи щодо запобігання злому розділу, а також опис компонентів програмного забезпечення (незалежно від того, чи є вони новими чи раніше виготовленими) і посилання на базову версію, з якої вони були створені. завантажено. Цей опис має також включати похідні вимоги, що випливають із процесу розробки програмного забезпечення. Якщо система містить неактивний код, опис заходів безпеки для активації коду на цільовому комп'ютері та обґрунтування проектних рішень безпосередньо включені до системних вимог, пов'язаних із її безпекою. **Коваленко знову повторює текст, який уже був переписаний на с. 35–36.** **Повторний плагіат.**	The basic output in these processes is the design description, which contains software architecture and low-level requirements. The data should include a detailed description of how the software satisfies high-level requirements, including algorithms and data structure, and how the software requirements correspond to the processes and tasks. It should also provide the software architecture description defining software structure, with implemented requirements, input/output descriptions (e.g., data dictionary, data and control flow within the design), resource limitations, a management strategy for resources and their limitations, and margins as well as methods for measuring these margins (e.g., time and memory, sequencing procedures). The description should include intraprocessors and intratask communication mechanisms, including fixed interruption time sequences, design methods, and details on their implementation (e.g., software loading). An important element of the description is user-modified software, partitioning methods, and measures to prevent partition breach as well a description of software components (regardless of whether they are new or previously manufactured) and a reference to the baseline version from which they were downloaded. This description should also include derivative requirements arising from the software design process. If a system contains an inactive code, a description of security measures for the activation of the code on the target computer and a justification of design decisions are directly included in the system requirements associated with its security.
	С. 276–277.	**С. 5–6.**
	У процесі кодування програмного забезпечення вихідний код реалізується на основі архітектури програмного забезпечення та вимог низького рівня. Вхідні дані процесу кодування – це вимоги низького рівня та архітектура програмного забезпечення з процесів проектування програмного забезпечення, плану розробки програмного забезпечення та стан-	The software design process is complete when its objectives and the goals of the associated integration processes are achieved. Within the software coding process, the source code is implemented based on software architecture and low-level requirements. Coding process inputs are low-level requirements and software architecture from the software design pro-

	дартів кодування програмного забезпечення [97]. Процес програмного кодування може бути розпочато, коли будуть виконані заплановані критерії переходу. Вихідний код розробляється в ході цього процесу і базується на системній архітектурі та вимогах низького рівня. Цільовий комп'ютер і вихідний код процесів кодування програмного забезпечення використовуються для компіляції, об'єднання та завантаження даних у процесі інтеграції; це спрямовано на інтеграцію системи авіоніки або її складових обладнання [98]. **Коваленко до чужого переписаного тексту додала фальшиві покликання [97] і [98], яких немає в оригінальній статті.** **Фальсифікація джерел.** **Плагіат.**	cesses, the software development plan, and software coding standards. The software coding process can be commenced when planned transition criteria are met. The source code is developed in the course of this process and is based on system architecture and low-level requirements. The target computer and the source code from software coding processes are used in order to compile, combine, and load data within the integration process; this is aimed at integrating an avionics system or its equipment constituents.
	С. 277–278.	**С. 6.**
	Процес інтеграції програмного забезпечення авіоніки складається з чотирьох детальних процесів: – Комунікація в рамках сертифікації програмного забезпечення; – Управління вимогами в рамках сертифікації ПЗ; – Перевірка в рамках сертифікації ПЗ; – Оцінка якості в рамках сертифікації ПЗ.	The avionics software integration process consists of four detailed processes: ● Communication within software certification; ● Requirement management within software certification; ● Verification within software certification; ● Quality assessment within software certification.
	С. 278.	**С. 6.**
	Спілкування в рамках сертифікації є важливим процесом, завданням якого є успішне завершення сертифікації програмного забезпечення. Це передбачає постійну співпрацю та спілкування між заявником та органом сертифікації. Заявник є організацією, яка бажає отримати сертифікацію. Цей процес поширюється на весь життєвий цикл програмного забезпечення, який починається з планування та закінчується його утилізацією. Завдання заявника полягає у визначенні заходів відповідності, які визначають спосіб, у який програмне забезпечення буде задовольняти основним вимогам сертифікації. Процес управління вимогами, як і комунікація в рамках сертифікації, поширюється на весь життєвий цикл програмного забезпечення. Він охоплює всі дані та документацію, що використовуються для розробки та перевірки програмного забезпечення. Управління вимогами — це «мистецтво» визначення, організації та контролю змін на етапі розробки програмного забезпечення. Основним завданням цього процесу є досягнення максимально можливої ефективності при мінімізації помилок. Метод управління вимогами пов'язаний з рівнем пошкодження бортового обладнання. Стандарт DO-178C визначає два рівні управління програмним забезпеченням: рівні CC1 і CC2. Рівень CC1 має відповідати всім вимогам DO-178C, тоді як рівень CC2 задовольняє лише деякі з них (пов'язані з рівнями гарантії C і D).	Communication within certification is an essential process, the task of which is the successful completion of software certification. It involves constant cooperation and communication between the applicant and the certification body. The applicant is the entity seeking certification. This process spreads over the entire software lifecycle, which starts with planning and ends with its disposal. The task of the applicant is to determine compliance measures, which define the manner in which the software will satisfy basic certification requirements. The process of requirement management, just like communication within certification, spreads over the entire software lifecycle. It covers all data and documentation used for software development and verification. Managing the requirements is the “art” of identifying, organizing, and controlling changes at the software development stage. The main task of this process is to achieve the highest possible efficiency while minimizing errors. The requirement management method is associated with the onboard equipment damage level. Standard DO-178C defines two software control levels: Levels CC1 and CC2. Level CC1 must satisfy all DO-178C requirements, whereas Level CC2 only satisfies some of them (related to Assurance Levels C and D).
	С. 278–279.	**С. 6.**
	Процес верифікації програмного забезпечення передбачає виявлення та опис помилок, які виникли від етапу планування програмного забезпечення до етапу розробки. Стандарт DO-178C визначає не методи, які використовуються для верифікації, а скоріше цілі, які мають бути досягнуті. Завдання процесу оцінки якості полягає в тому, щоб продемонструвати, що розроблене програмне	The software verification process involves the detection and description of errors introduced from the software planning stage to the development stage. Standard DO-178C does not define the techniques utilized for verification but rather the objectives that must be achieved. The task of the quality assessment process is to demonstrate that the developed software is compliant

	забезпечення відповідає передбачуваним вимогам і стандартам, що, як наслідок, має призвести до того, що продукт відповідає очікуванням замовника (або демонструє розбіжності щодо прийнятих вимог). Оцінка якості програмного забезпечення – це безперервний процес, який починається на етапі планування та продовжується через етапи розробки та тестування до кінцевого продукту [119, 118]. **Коваленко до чужого переписаного тексту додала фальшиві покликання [119] і [118], яких немає в оригінальній статті.** **Фальсифікація джерел.** **Плагіат.**	with the assumed requirements and standards, which, in consequence, should result in a product meeting the expectations of the ordering party (or show discrepancies relative to the adopted requirements). Software quality assessment is an ongoing process, which starts at the planning stage and continues through the development and testing stages until the final product.
	С. 279.	**С. 7.**
	Початковою точкою розробленого методу визначено, згідно безпеки програмного забезпечення, можна визначити наступним чином: R(t) = 1 − Q(t) (5.5) де R(t) – стан безпеки програмного забезпечення (уразливостей не виявлено); Q(t) – вразливий стан (уразливості, виявлені в досліджуваному ПЗ); λ(t) – інтенсивність переходу зі стану R(t) у Q(t). Інтенсивність появи вразливості програмного забезпечення визначається як щільність ймовірності появи вразливості в часі за умови, що протягом цього часу не виявлено жодної уразливості програмного забезпечення [120–122]. **Коваленко переписала англійську статтю, видаливши покликання в першому абзаці і замінивши джерела на фальшиві в останньому абзаці цього фрагмента.** **Фальсифікація джерел.** **Плагіат.**	The starting point of the developed method is determined according to which software security can be defined as follows [24,25]: R(t) = 1 – Q(t) (1) where R(t) is the software security state (no vulnerabilities discovered); Q(t) is the vulnerable state (vulnerabilities discovered in the examined software); l(t) is the intensity of transition from state R(t) to Q(t). Software vulnerability occurrence intensity is defined as the density of vulnerability occurrence probability in time, provided that no software vulnerabilities are discovered during that time [26–30].
	С. 279.	**С. 7.**
	Інтенсивність появи вразливості програмного забезпечення : $\lambda(t) = lim_{\Delta t \to 0} \frac{P\{t < T \le t + \Delta t | t < T\}}{\Delta t}$ (5.6) Імовірність виявлення вразливості в одному випадку програмного забезпечення виражається формулою: $q = 1 - e^{-\lambda t}$ (5.7) де q – ймовірність виявлення вразливості в програмному забезпеченні; t – час роботи в інтервалі часу (0, t).	Software vulnerability occurrence intensity is described by the following formula: $\lambda(t) = \lim_{\Delta t \to 0} \frac{P\{t < T \le t + \Delta t | t < T\}}{\Delta t}$ (2) The probability of discovering vulnerabilities in a single software occurrence is expressed by the formula: $q = 1 - e^{-\lambda t}$ (3) where q is the probability of vulnerability discovery in a single software instance; t is the time of a single software instance operation in time interval (0, t).
	С. 279.	**С. 7.**
	Подальші перетворення дозволили визначити оцінку (λˆ) невідомої інтенсивності виявлених вразливостей — параметр λ. Для цього використовуємо метод максимальної правдоподібності, який дає результат, який можна записати так: $\lambda^{\wedge} = \frac{n1+n2+\cdots+ni}{T_1+T_2+\cdots+T_i}$ (5.8)	Further transformations allowed to determine estimator (λˆ) of the unknown intensity of discovered vulnerabilities—parameter l. For this purpose, the maximum likelihood method was used, which gives the result that can be written as follows: $\hat{\lambda} = \frac{n_1 + n_2 + \ldots + n_i}{T_1 + T_2 + \ldots + T_i}$ (4)
	С. 279.	**С. 7.**
	Запропонований метод передбачає розрахунок ймовірності усунення вразливостей програмного забезпечення та середнього значення усунутих уразливостей (у відповідний момент часу). Запропонований підхід базується на використанні розгалужених процесів [123].	The proposed method involves calculating the probability of eliminating software vulnerabilities and the mean value of eliminated vulnerabilities (at a relevant time). The proposed approach is based on using branching processes [31].

	Покликання [123] це: Parmee I C & Watson A H, Preliminary airframe design using coevolutionary multi objective genetic algorithms, in GECCO–99: Proc Gen Evol Computat Conf (Banzhaf W et al. eds), Orlando, Florida, USA, Morgan Kaufmann, July 1999, 1657-1665. **Це покликання фальшиве.** **Фальсифікація джерел.** **Плагіат.**	**Покликання [31] це:** Zieja, M.; Stachurski, A. An outline of the method for predicting IT vulnerabilities. In Proceedings of the MATEC Web of Conferences 210, 22nd International Conference on Circuits, Systems, Communications and Computers, Majorca, Spain, 14–17 July 2018.
	С. 279–280.	**С. 7–8.**
	Отже, розглядаємо сценарій із застосуванням заходів захисту від уразливості. У такому випадку передбачається, що після одного періоду тестування всі вразливості програмного забезпечення виявлені та виправлені. Як наслідок, ці попередні вразливості або не існують у наступний період, або їхній вплив падає до безпечного рівня; однак нові вразливості з'являються через виправлення попередніх. Крім того, завжди існує принаймні одна вразливість програмного забезпечення, тому для спрощення позначення передбачається, що нульовий період має принаймні одну вразливість.: $Q(n+1) = \sum_{k=Q(1)}^{Q(n)} Y_k^{(n+1)}$ (5.9) де $(Y_k)_{k\geq 1}$ утворює послідовність незалежних, однаково розподілених, невід'ємних випадкових величин.	The method concept involves patching vulnerable software. Firstly, a scenario with applied antivulnerability measures is considered. In such a case, it is assumed that after one testing period, all software vulnerabilities are detected and patched. As a result, these former vulnerabilities are either nonexistent in the following period or their impact falls to a safe level; however, new vulnerabilities appear due to patching the previous ones. Furthermore, there is always at least one software vulnerability, so for the sake of simplifying the notation, it is assumed that the zero period has at least one vulnerability. Let it denote the number of sensitivity descendants for each of them. This means that a new population size appears from time to time, which can be obtained as follows: $Q(n+1) = \sum_{k=Q(1)}^{Q(n)} Y_k^{(n+1)}$ (5) where $(Y_k)_{k\geq 1}$ forms a sequence of independent, identically distributed, non-negative (and almost surely finite) random variables.
	С. 280.	**С. 9.**
	Пошук вразливостей програмного забезпечення безпеки авіоніки, які призводять до помилок або хакерських атак, вважається корисною діяльністю, яка може сильно вплинути на безпеку та надійність польотів. Здатність передбачити появу вразливостей програмного забезпечення або кількісно виміряти їх вплив дає змогу прогнозувати тенденції безпеки програмного забезпечення та планувати широко зрозумілий процес управління його безпекою [124]. **Коваленко переписала англійську статтю, додавши фальшиве покликання.** **Фальсифікація джерел.** **Плагіат.**	Finding avionics software security vulnerabilities that lead to errors or hacker attacks is considered a useful activity that can highly impact flight safety and reliability. The ability to predict the occurrence of software vulnerabilities or to quantitatively measure their impact enables the forecasting of software security trends and the planning of a widely understood process of managing its safety.
	С. 280.	**С. 9.**
	Розроблений метод спрямований на покращення можливості прогнозування вразливостей у тестованому програмному забезпеченні. Перевірка та подальше підвищення точності запропонованого методу вимагає подальших досліджень з подальшим емпіричним аналізом з використанням даних із баз даних уразливостей або інших типів ресурсів щодо вразливостей. Отже, одним із рішень, запроваджених на AFIT у сфері обмеження помилок у розробленому програмному забезпеченні авіоніки, є автоматизована система керування відповідно до вимог стандарту DO-178C та впровадження цих вимог у формі процедури в ISO -9001 Система забезпечення якості. Створена комп'ютерна система дозволяє здійснювати перевірки та створювати документи, необхідні	The developed method is aimed at improving the ability to predict vulnerabilities in the tested software. Verifying and then improving the accuracy of the proposed method requires further research, followed by empirical analysis using the data from vulnerability databases or other types of resources on vulnerability. **3. Results** One of the solutions introduced at AFIT in the field of limiting errors within developed avionics software is a computer-aided management system, as per the requirements of standard DO-178C, and the implementation of these requirements in the form of a procedure in the ISO-9001 Quality Assurance System. The constructed computer system enables the implementation of verifications and the creation of docu-

	за стандартом DO-178C (тобто плани, звіти, звіти) безпосередньо з IT-мережі [125]. **Коваленко пише «Отже», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Коваленко переписала англійську статтю, додавши фальшиве покликання.** **Фальсифікація джерел.** **Плагіат.**	ments required by standard DO-178C (i.e., plans, reports, statements) directly from the AFIT IT network.
	С. 280–281.	**С. 9–10.**
	Завданням, прийнятим для реалізації, було створення засобу автоматизованого керування розробкою програмного забезпечення для параметрів польоту, яка розроблятиметься та створюватиметься в AFIT за стандартом DO-178C. Основні функції побудованої IT системи комп'ютерної підтримки включали: – Налаштування нового проекту, що передбачає введення інформації про назву проекту, персональні дані керівника проекту та окремих підрядників, їх повноваження та рівні доступу системи до системи; – Внесення даних до бази знань щодо реалізації проекту (деталі, фінанси, обмеження); – Автоматична генерація шаблонів документів, необхідних у стандарті DO-178C (тобто планів, стандартів, процедур і методів перевірки, звітів та інших записів);	*3.1. Basic Tasks and Functions of a Computer System Supporting the Management Process* The task adopted for the purposes of implementation was constructing a tool for computer-aided management of software development for the purposes of a helmetmounted flight-parameter display system, to be developed and constructed at AFIT as per standard DO-178C. The main functions of the constructed computer support IT system included: ● Setting up a new project, which involves entering information on, among other things, project title, personal data of the project manager and individual contractors, their authorizations, and system accessibility levels to the system; ● Entering data into the knowledge base regarding project implementation (details, finances, and limitations); ● Automatic generation of document templates required in standard DO-178C (i.e., plans, standards, verification procedures and methods, reports, and other entries);
	С. 281.	**С. 10.**
	– Автоматична генерація тестів для розробленого програмного забезпечення та архівація результатів тестування; – Архівування листування між виконавцями проекту, програмних файлів і результатів їх тестування; – Автоматичне резервне копіювання файлів на сервер, розташований в іншій будівлі на території АФІТ (захист від втрати даних); – Надання даних про реалізацію проекту згідно введеної авторизації користувачів системи; – Звітування про статус проекту для цілей аудиту чи перевірки відповідно до введених інструкцій.	● Automatic generation of tests for the developed software and archiving the test results; ● Archiving correspondence between project contractors, programfiles, and their test results; ● Automatic backup of the files to a server located in another building within AFIT premises (protection against data loss); ● Providing project implementation data as per entered authorization of system users; ● Reporting project status for the purposes of an audit or inspection, as per the entered guidelines.
	С. 281–282.	**С. 10.**
	Комп'ютерна система, що підтримує управління керуванням програмним забезпеченням авіоніки, після інсталяції спеціалізованого програмного забезпечення (включаючи статичні та динамічні аналізи, адаптовані до вразливості програмного забезпечення та виявлення помилок, для визначення його вразливості до структурних пошкоджень системи авіоніки та хакерських атак), забезпечує прямі, електронні співпраця між учасниками проекту та її контроль з боку керівника проекту, який відповідає за правильне виконання проекту.	A computer system supporting the management of avionics software management, after installing specialized software (including static and dynamic analyses, adapted to software vulnerability and error detection, to determine its vulnerabilities to structural damage of the avionics system and hacker attacks), enables direct, electronic cooperation between project participants and its supervision by the project manager, who is responsible for correct project implementation.
	С. 282.	**С. 10.**
		3.2. Structural Diagram of a Computer System Supporting the Management Process

	Основним структурним елементом комп'ютерної системи, що забезпечує управління розробкою програмного забезпечення авіоніки, є спеціальний сервер, вбудований у мережу AFIT IT. Сервер взаємодіє з робочими станціями окремих користувачів системи. Файловий сервер, який називається резервним, використовується для захисту зібраної інформації. Його завдання – архівувати файли. Поточний стан виконання проекту зберігається в його пам'яті після кожного «робочого дня». Він також співпрацює з аварійним сервером, який включається в разі збою основного сервера (захист поточного прогресу проекту) [126]. **Коваленко переписала англійську статтю, додавши фальшиве покликання.** **Фальсифікація джерел.** **Плагіат.**	The main structural element of the computer system supporting the management of avionics software development is a special server built into the AFIT IT network (Figure 3). The server cooperates with the workstations of individual system users. A file server, called a backup, is used to protect the gathered information. Its task is to archive the files. The current state of project progress is saved in its memory after each "working day". It also cooperates with an emergency server, which is turned on in the case of main server failure (protecting the current project's progress).
	С. 282.	**С. 10.**
	Операційне програмне забезпечення основного сервера включає операційну систему Windows Server, базу даних Windows SQL Server і пакет редагування та розрахунків MS Office як спеціалізоване програмне забезпечення для управління проектами з використанням інструкцій згідно стандарту DO-178C.	The main server operating software includes a Windows Server operating system, a Windows SQL Server database, and an MS Office editing and calculation suite as specialized software for project management using guidelines as per standard DO-178C.
	С. 282.	**С. 10.**
	Модель комп'ютерної системи, що підтримує розробку програмного забезпечення авіоніки, критичного для безпеки польотів; (1) діаграма архітектури комп'ютерної системи, інтегрованої з мережею IT; (2) діаграма архітектури файлового сервера [127]. **Коваленко переписала англійську статтю, включивши в основний текст підпис до рисунку без самого рисунку!** **До чужого тексту Коваленко додала фальшиве джерело.** **Фальсифікація джерел.** **Смішний плагіат.**	 Figure 3. Structure of a computer system supporting the development of avionics software critical to flight safety; (a) Architecture diagram for a computer system integrated with an IT network; (b) Architecture diagram of a file server.
	С. 282.	**С. 10.**
	Спеціалізоване програмне забезпечення, встановлене на сервері, використовує обчислювальні модулі, до складу яких входять: – Базовий аналіз для попереднього тестування та верифікації розробленого програмного забезпечення (Static Analysis, Dynamic Analysis, TBvision, TBrun, TBmisra, TBsafe) [128]; – Розширений аналіз для попереднього тестування та верифікації розробленого програмного забезпечення (Modified Condition/Decision Coverage, Information Flow Analysis, Dynamic Data Flow Coverage, Extract Semantic Analysis) [127]; **Коваленко переписала англійську статтю, додавши фальшиві покликання.**	The specialized software installed on the server uses computational modules, which include: • Basic analysis for preliminary testing and verification of the developed software (Static Analysis, Dynamic Analysis, TBvision, TBrun, TBmisra, TBsafe); • Advanced analysis for preliminary testing and verification of the developed software (Modified Condition/Decision Coverage, Information Flow Analysis, Dynamic Data Flow Coverage, Extract Semantic Analysis);

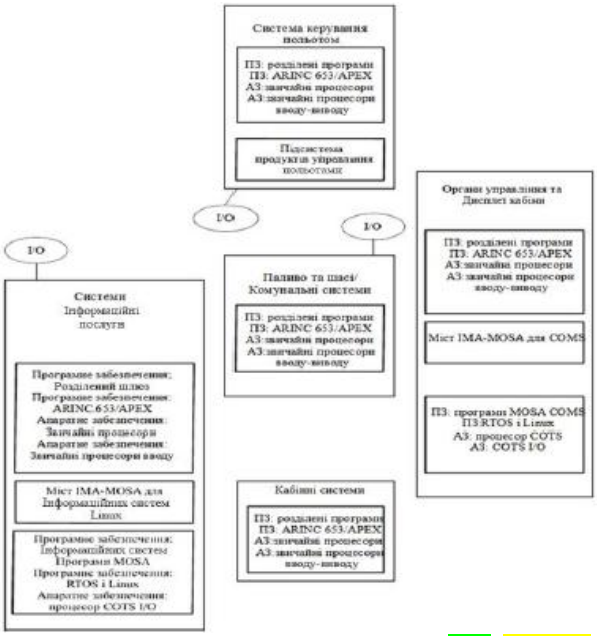
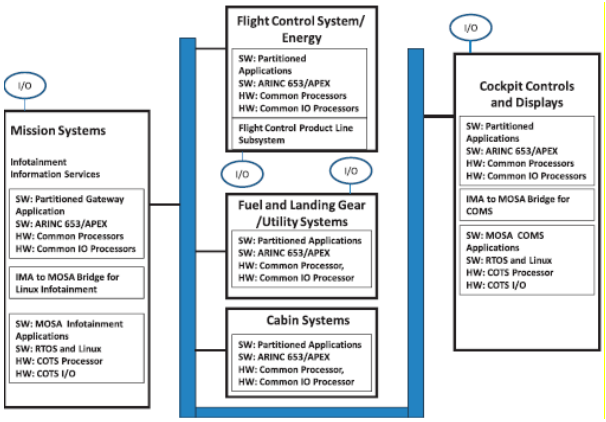
	Фальсифікація джерел. **Плагіат.**	
	С. 283.	**С. 11.**
	– Додатковий аналіз для попереднього тестування та верифікації розробленого програмного забезпечення (Test Vector Generation, TBeXtreme, TBmanager, Support for Target Testing, Tool Qualification); – Додаткове програмне забезпечення для тестування, яке забезпечує постійну підтримку процесу розробки окремих програмних компонентів. Представлене спеціалізоване програмне забезпечення, інтегроване з комп'ютерною системою підтримки процесу розробки програмного забезпечення авіоніки [128]. **Коваленко переписала англійську статтю, додавши фальшиве покликання.** **Фальсифікація джерел.** **Плагіат.**	– Complementary analysis for preliminary testing and verification of the developed software (Test Vector Generation, TBeXtreme, TBmanager, Support for Target Testing, Tool Qualification); – Additional testing software, which provides ongoing support for the process of developing individual software components. The presented specialized software, integrated with the computer system supporting avionics software development process management, was used to test the software developed for a helmet-mounted flight-parameter display system.
	С. 283.	**С. 11.**
	У технічній реалізації комп'ютерної системи підтримки процесу управління розробкою програмного забезпечення авіоніки використовувалися спеціалізовані модульні комп'ютери, які функціонували як сервери та вбудовані в IT-шафу.	*3.3. Technical Implementation of a Computer System Supporting the Management Process* The technical implementation of the computer system supporting the avionics software development management process utilized specialized modular computers (Figure 4), operating as servers and built into an IT cabinet.
	С. 283.	**С. 11.**
	У мережу AFIT IT також входять модулі живлення, комутатори, з'єднувальні панелі, кабельна розводка та додаткові елементи. Модульна структура серверів і застосування панелей підключення дозволяє вибрати конфігурацію IT-мережі, оптимальну для системного адміністратора та користувачів.	The AFIT IT network also includes power supply modules, switches, connection panels, cabling, and additional elements (including thermometers and hygrometers to monitor the condition of the air in the server room area). A modular structure of the servers and the application of connection panels (Figure 4b) enables the selection of an IT network configuration that is optimal for the system administrator and users.
	С. 283.	**С. 13.**
	Вимоги до безпеки польотів є результатом оцінки рівня безпеки, який містить функціональні вимоги, вимоги до інтеграції та надійності для даної системи. Вимоги до рівня помилок визначаються в ході процесу оцінки безпеки, щоб гарантувати цілісність системи шляхом визначення засобів захисту системи та відповідей у разі таких помилок. Ці вимоги визначені для програмного та апаратного забезпечення з метою усунення або обмеження ефектів помилок, а також забезпечення виявлення помилок, допуску, видалення та уникнення. Системні процеси, відповідальні за вдосконалення та призначення системних вимог апаратному та/або програмному забезпеченню, призводять до розробки відповідної архітектури для системи відображення параметрів польоту.	Flight safety requirements result from assessing the security level, which contains functional, integration, and reliability requirements for a given system. Requirements on the error level are defined in the course of the security assessment process in order to guarantee system integrity through specifying the system protections and responses in the event of such errors. These requirements are defined for software and hardware with the aim of eliminating or limiting error effects as well as ensuring error detection, tolerance, removal, and avoidance. System processes responsible for improving and assigning system requirements to hardware and/or software lead to the development of the appropriate architecture for the helmet-mounted flight parameter display system.
	С. 283–284.	**С. 13.**
	Конфігурація BIOS розуміється як процес налаштування параметрів BIOS (двійкова система введення/виведення), доступних програмісту, що призводить до покращеної взаємодії між апаратним і програмним рівнями.	BIOS configuration is understood as the process of setting the BIOS (Binary Input/Output System) parameters available to a programmer, which leads to improved cooperation between the hardware and software layers.
	С. 284.	**С. 13.**

	Програмне забезпечення, нестандартна поведінка якого, як зазначено в процесі оцінки безпеки, може призвести або сприяти суттєвій помилці, що призведе до умов, що обмежують функціональні можливості вертольота або додаткове навантаження для пілота, підлягає спеціальному аналізу [129]. **Покликання [129] – це:** Marta A, Parametric study of a genetic algorithm: using an aircraft design optimization problem, Report Stanford University, Department of Aeronautics and Astronautics, 2008. **Це джерело відсутнє в Інтернеті.** **Коваленко переписала англійську статтю, додавши фальшиве покликання.** **Фальсифікація джерел.** **Плагіат.**	The software, the nonstandard behavior of which, as indicated in the security assessment process, can lead or contribute to a significant error, resulting in conditions limiting the helicopter's functionalities or an additional burden for the pilot, is subject to a special analysis. If a given component is awarded such a level, it will not be approved by the certification body.
	С. 284.	**С. 13.**
	Процес планування із залученням програмного забезпечення для встановлення системи відображення параметрів польоту визначається таким чином, щоб вимоги були виконані, а рівень достовірності був адекватним рівню надійності прийнятого програмного забезпечення.	*4.1. Meeting the Software Planning Requirements* The planning process involving the software for the helmet-mounted flight-parameter display system is defined in a way that the requirements are met and the confidence level is adequate to the adopted software's assurance level.
	С. 284.	**С. 14.**
	Відповідно до вимог стандарту DO-178C процеси розробки програмного забезпечення, відображення параметрів польоту містяться в процесі планування програмного забезпечення та процесі розробки програмного забезпечення. Процеси, пов'язані з розробкою програмного забезпечення, включають процеси визначення вимог до програмного забезпечення; процеси, пов'язані з кодуванням програмного забезпечення, і процеси, пов'язані з інтеграцією [130]. **Коваленко переписала англійську статтю, додавши фальшиве покликання.** **Фальсифікація джерел.** **Плагіат.**	*4.2. Meeting the Software Development Requirements* According to the requirements of standard DO-178C, software development processes for a helmet-mounted flight-parameter display system is contained in the software planning process and the software development process. Processes associated with software development include software requirement definition processes, software design processes, processes associated with software coding, and integration-related processes.
	С. 284.	**С. 14.**
	Процес інтеграції програмного забезпечення для системи відображення параметрів польоту, включає інтеграцію програмного забезпечення та інтеграцію апаратного/програмного забезпечення. Процеси інтеграції можуть бути виконані, коли виконано заплановані вимоги переходу. Входами процесу інтеграції є архітектура програмного забезпечення з процесів проектування програмного забезпечення та вихідний код із процесів кодування програмного забезпечення, тоді як виходами процесу інтеграції є файли об'єктного коду з компіляцією. Інтеграційні процеси є завершеними, коли досягнуті їхні цілі та цілі пов'язаних інтегральних процесів. Об'єктний код має бути згенерований із вихідного коду, а потім скомпільований.	*4.3. Meeting the Software Integration Requirements* The integration process of the software for a helmet-mounted flight-parameter display system involves software integration and hardware/software integration. Integration processes can be executed when planned transition requirements are met. Integration process inputs are software architecture from software design processes and the source code from software coding processes, whereas the integration process outputs are the object code files with compilation. Integration processes are complete when their objectives and the goals of associated integral processes are met. The object code should be generated from the source code and then compiled.
	С. 284–285.	**С. 14.**
	Усі файли з параметрами даних мають бути	All files with data parameters should be generated,

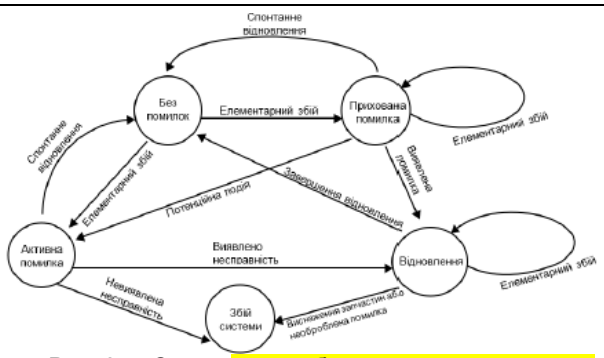
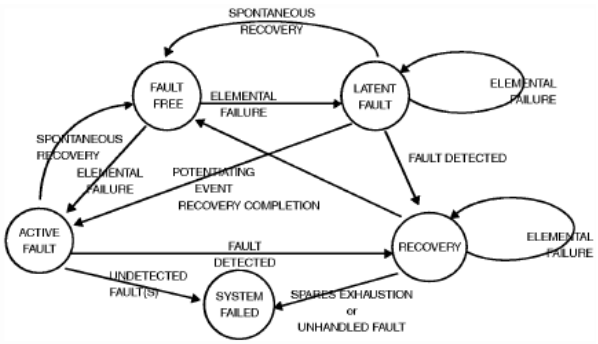
	згенеровані, а програмне забезпечення має бути інтегровано в головний комп'ютер, емулятор цільового пристрою або цільовий пристрій. Програмне забезпечення має бути реалізовано на цільовому комп'ютері з метою інтеграції апаратного/програмного забезпечення. Невідповідні або помилкові вхідні дані, виявлені під час процесу інтеграції, мають бути направлені до процесів вимог до програмного забезпечення, процесів розробки програмного забезпечення, процесів кодування або процесів планування програмного забезпечення як зворотний зв'язок, який потребує перевірки.	and software should be integrated into the main computer, target device emulator, or the target device. The software should be implemented in the target computer for the purposes of hardware/software integration. Inappropriate or erroneous inputs detected during the integration process should be forwarded to software requirement processes, software design processes, coding processes, or software planning processes as feedback that requires verification.
	С. 285.	**С. 14.**
	Графічний комп'ютер системи відображення параметрів польоту передає інформацію про параметри польоту на денний дисплей або нічний дисплей. Інформація, отримана від системи узгодження сигналів, приймача супутникової навігації GPS і блоку аеродинамічних даних ADU, подається у вигляді графічних символів або в цифровому вигляді. Структура графічного програмного забезпечення складається з таких елементів: – Безпосереднє налаштування операційної системи процесора BIOS; – Конфігурація вбудованої операційної системи WINDOWS XP; – Конфігурація програмного забезпечення користувача графічного комп'ютера; – Архітектура програмного забезпечення користувача графічного комп'ютера	*4.4. Selected Test Results Involving Software Developed for a Helmet-Mounted Flight-Data Display System* The graphic computer of the helmet-mounted flight-parameter display system conveys flight parameter information on the DWN-1 daytime helmet-mounted display or the NWN-1 night-time helmet-mounted display. The information received from the signal matching system, the GPS satellite navigation receiver, and the ADU aerodynamic data unit is presented as graphic symbols or in digital form. The graphic computer software structure comprises the following elements: ● BIOS processor direct operation system configuration; ● WINDOWS XP Embedded operating system configuration; ● Graphic computer user software configuration; ● Graphic computer user software architecture (Mi17sys main module, Mi17konfigurator configuration module, and Mi17hud display module).
	С. 285–286.	**С. 14.**
	Графічне програмне забезпечення комп'ютера вбудовано в постійну пам'ять материнської плати ЦП. Результати калібрування для окремих каналів вимірювання зберігаються у зовнішній пам'яті комп'ютера у FLASH-пакеті. Графічна ідентифікація комп'ютерного програмного забезпечення містить таку інформацію, як назва програмного забезпечення, ідентифікаційний номер програмного забезпечення, ідентифікатор версії програмного забезпечення, модулі компонентів програмного забезпечення та надану ліцензію.Системне програмне забезпечення проходить тестування з метою підтвердження його відповідності встановленим вимогам стандартів AQAP 2210 та DO-178C [103]. Водночас забезпечується високий рівень упевненості у тому, що потенційні помилки, здатні спричинити неприйнятні умови відмови, ідентифіковані в процесі оцінки безпеки відповідно до стандарту ARP 4761, своєчасно виявляються та усуваються. **Покликання [103] – це:** Y. Kovalenko, «Methodology For Testing Languages For Embedded Avionics Systems». Problems of Informatizati on and Management, no. 63, pp. 44 – 52, 2022. **Коваленко переписала англійську статтю 2021-го року, додавши фальшиве покликання на свою власну статтю 2022-го року.** **Фальсифікація джерел.** **Плагіат.**	Graphic computer software is embedded in the permanent memory of the CPU motherboard. The calibration results for individual measurement channels are saved in the computer's external memory on the FLASH packet. Graphic computer software identification contains such information as software name, software ID-number, software version ID, software component-modules, and the granted license. The system software is tested in order to demonstrate that it satisfies the basic requirements set out in AQAP 2210 and DO-178C and to demonstrate, with a high degree of confidence, that the errors, which can lead to the unacceptable failure conditions defined in the ARP 4761 security assessment process, have been removed.

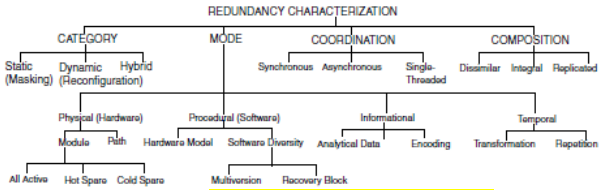
7	**Коваленко Ю. Б.** **Інформаційна підтримка процесу проектування та експлуатації комплексу бортового обладнання інтегрованої модульної авіоніки.** – Дис. … доктора технічних наук. – Київ, 2025. (https://duikt.edu.ua/uploads/p_86_93515477.pdf)	**Gaska T., Watkin C., Chen Y.** **Integrated Modular Avionics—Past, Present, and Future** // IEEE Aerospace and Electronic Systems Magazine 2015, 30(9), p. 12–23. (https://doi.org/10.1109/MAES.2015.150014)
	С. 62.	**С. 14.**
	Boeing 777 став першим цивільним транспортним літаком що використовував множинний доступ (ARINC 629) на цивільному транспортному ПС для підтримки мережі IMA, рисунок 2.1.	The Boeing 777 was the first civil transport aircraft to use a multiple access bus (ARINC 629) on a civil transport aircraft to support IMA networking.
	С. 63.	**С. 14.**
	ARINC 629 детермінований і кожен термінал має високу цілісність над асоційованим доступом пристрою до шини, правильним використанням смуги пропускання та відбором переданих і отриманих даних. Хоча це було критичне покращення на 2 Мбіт/с порівняно з мережами ARINC 429 точка-точка, які раніше працювали на 12,5 і 1000 Кбіт/с, ARINC 629 необхідні промислові ASIC і з'єднувачі. Хоча в IMA був досягнутий прогрес завдяки таким зусиллям, як AIMS, впровадження було оптимізовано для конкретних доменів і досягли лише частини намічених галузевих цілей IMA.	ARINC 629 is very deterministic in operation and each terminal has high integrity over the associated unit's access to the bus, the correct use of bandwidth, and the selection of transmitted and received data. While it was a critical improvement at 2 Mbps over ARINC 429 point-to-point networks previously operating at 12.5 and 1000 kbps, ARINC 629 required custom industry ASICs and couplers. The Avionics Handbook published in 2001 suggested the following challenges for IMA going forward [7]. ► While there was progress in IMA with efforts like AIMS, the implementations were optimized for specific domains and only met part of the intended industry IMA objectives,
	С. 65–66.	**С. 15.**
	Аппаратне забезпечення. Уніфікований набір CPM використовується в підсистемах комерційного транспорту в повній конфігурації IMA з використанням стандарту «Керівництво з проектування інтегрованої модульної авіоніки» (ARINC 651) для визначення загальної архітектури апаратного забезпечення, змішаного з апаратним забезпеченням на основі спеціальної стійки або шафи [11] . В Військові IMA, як правило, базуються на серії COTS і надійних готових (ROTS) OpenVPX або VMEbus CPM на основі ATR або апаратного забезпечення на основі стійки. Через поступові спіралі військових і потребу в розширеній обробці для нових функцій у вибраних підсистемах розгортається кілька поколінь CPM в даній системній архітектурі платформи. Також у військовій авіоніці деякі підсистеми базуються на IMA, деякі використовують спільні кластери обробки IMA, а деякі зберігають об'єднані рішення. Однією з областей небажання використовувати IMA як у цивільних, так і у військових системах є системи керування польотом. Занепокоєння викликає додаткова потреба в більш жорсткій синхронізації каналів і складність керування резервуванням для більш розподілених конфігурацій.	HARDWARE A unified set of CPMs are typically used across commercial transport subsystems in a full IMA configuration using the "Design Guidance For Integrated Modular Avionics" (ARINC 651) standard for definition of the generic hardware architecture mixed with custom rack or cabinet based hardware [11]. The military IMAs are typically based on a range of COTS and rugged off-the-shelf (ROTS) OpenVPX or VMEbus CPMs based on ATR or rack based hardware. Due to the incremental spirals of the military and need for increased processing for new functions in selected subsystems, multiple generations of CPMs are deployed in a given platform system architecture. Also in military avionics, some subsystems are IMA based, some use shared IMA processing clusters, and some retain federated solutions. One area of reluctance in both civil and military systems for the use of IMA is in flight control systems. The concern is the additional need for tighter channel synchronization and the complexity of redundancy management for more distributed configurations.
	С. 66–67.	**С. 15.**
	Програмне забезпечення. Додатки авіоніки складаються з набору функцій або завдань, які виконуються на одному або кількох розподілених CPM. ARINC 653 і його стандарти програмного забезпечення інтерфейсу APEX використовуються як на комерційних, так і на військових платформах, щоб уможливити просторовий і часовий розподіл функцій авіоніки в архітектурі платформи. Стандарт ARINC 653 був представлений у 2005 році. APEX використовується для встановлення тимчасового розділення шляхом введення фіксованих періодич-	SOFTWARE Avionics applications are composed of a set of functions or tasks that are executed on a single or multiple distributed CPMs. ARINC 653 and its APEX interface software standards are used in both commercial and military platforms to enable spatial and temporal partitioning of the avionics functions of platform architecture. The ARINC 653 standard was introduced in 2005. APEX is used to establish temporal partitioning by introduction of fixed periodically scheduled partitions with limited time-windows where applications are allowed to execute. A major frame repre-

	но запланованих розділів з обмеженими часовими вікнами, у яких програми можуть виконуватися. Основний кадр представляє основну періодичну послідовність виконання розділу. Основний кадр розбивається на цілу кількість другорядних кадрів однакового періоду та тривалості. Просторове розділення реалізується шляхом визначення віртуального адресного простору для кожного розділу статично під час запуску. Це логічне розділення дозволяє програмам зі змішаним рівнем критичності (як визначено в ARP-4754) працювати на одному CPM. У військових системах для критичних функцій, що не стосуються безпеки, також використовуються інші операційні системи реального часу на основі міркувань переносимості застарілих версій, а також драйвер OSA для використання Linux як основного інтерфейсу.	sents the principal periodic sequence of partition executions. The major frame is broken down into an integral number of minor frames of equal period and duration. Spatial partitioning is realized by the definition of the virtual address space for each partition statically at startup. This logical separation allows mixed criticality level applications (as defined in ARP-4754) to run on the same CPM. In military systems, for nonsafety critical functions, other real-time operating systems are also used based on legacy portability considerations and also the OSA driver to use Linux as a key interface.
	С. 72.	**С. 15.**
	Уніфікація мережі залежить від архітектури IMA, але стандарт ARINC 664, частина 7, використовується в комерційному просторі IMA для забезпечення якості обслуговування (QoS) між розподіленими CPM. Важливою перевагою ARINC 664 Part 7 є те, що він може сформувати безперервну мережу, яка охоплює як рівень літака (зв'язок між LRU, центрами IMA та іншими інтелектуальними периферійними пристроями), так і рівень системного домену (зв'язок між модулями, пов'язаними з певною групою систем).	Network unification varies in IMA architectures but the ARINC 664 Part 7 standard is used in the commercial IMA space to enable quality-of-service (QoS) between distributed CPMs. An important advantage of ARINC 664 Part 7 is that it can form a continuous network covering both the aircraft level (communication between LRUs, IMA centers, and other smart peripherals) and the system domain level (communication between modules associated with a particular group of systems).
	С. 73–74.	**С. 15.**
	Стандарт ARINC 664, частина 7, був представлений у 2005 році. Віддалені завдання спілкуються через порти APEX на каналі APEX. Дві розподілені комунікаційні задачі належать до розділу джерела та розділу призначення відповідно. На стороні передачі швидкість передачі даних пов'язана з одним віртуальним багатоадресним односпрямованим каналом зв'язку, який називається «Віртуальний канал». Налаштований комутатор ARINC 664 Part 7 виконує переадресацію пакетів, а також забезпечує контроль трафіку на своїх вхідних портах і працює на швидкості 10 або 100 Мбіт/с. У військових архітектурах IMA широко використовується стандартне з'єднання Ethernet у системах місій, де адекватного керування смугою пропускання достатньо для задоволення вимог QoS датчиків і каналів передачі даних без потреби в спеціалізованому ARINC 664 Частина 7 перемикачі.	The ARINC 664 Part 7 standard was introduced in 2005. Remote tasks communicate via APEX ports on an APEX channel. Two distributed communicating tasks belong to a source partition and a destination partition, respectively. On the transmitting side, a data transmission rate is associated to one virtual multicast unidirectional communication channel called a “Virtual Link.” A customized ARINC 664 Part 7 switch performs packet forwarding as well as enforcing traffic policing at its input ports and operates at 10 or 100 Mbps. In military IMA architectures, there is significant use of standard Ethernet interconnection in mission systems, where adequate bandwidth management is sufficient to meet sensor and data link QoS requirements without the need for specialized ARINC 664 Part 7 switches.
	С. 74.	**С. 15.**
	Мережі IMA є надзвичайно гнучкими, інтелектуальними магістральними мережами, які з'єднують зв'язок між різними форматами та структурами даних. Типова структура мережі IMA показана на рисинку 2.4.	IMA networks are extremely flexible, intelligent backbone networks that bridge communications between nonsimilar data formats and structures. A representative IMA network structure is shown in Figure 2.
	С. 74.	**С. 16.**

	 Рис. 2.4. Інформаційна підтримка IMA уніфікованої мережі взаємозвязку з резервною топологією ARINC 664	 **Unified Interconnection Network with Redundant Topology ARINC 664 Part 7** Figure 2. Sample IMA for civil aviation.
	С. 75–76.	**С. 15.**
	Ключовим елементом IMA є відокремлення прямого вводу-виводу для кожної функції та створення елемента даних, доступного в мережі. У сучасних реалізаціях деякий тип вводу/виводу або віддаленого терміналу є другим найважливішим апаратним елементом для CPM, з третім — мережевий комутатор. Крім того, діаграма показує похідну від чистої IMA, де такі функції, як керування польотом, інформаційно-розважальна система та зв'язок, можуть використовувати лінійку продуктів OSA, а не звичайні будівельні блоки IMA. Органи управління польотом є можливі в чистому IMA, але зазвичай їх уникають через проблеми загального режиму (за винятком випадків, коли архітектура IMA включає різний дизайн, який не є стандартним для цивільних ринків). Інформаційно-розважальні системи та засоби зв'язку – це той випадок, коли може знадобитися мережа IP більш широка підтримка IP і більше стандартів щодо інфраструктури комутатора. Однак навіть у цих випадках потрібен якийсь міст до мережі ARINC 664 Part 7.	A key element of IMA is to decouple direct I/O to each function and make it a network accessible data element. In today's implementations, some type of I/O bridge or remote terminal is the second most critical hardware element to the CPM, with the third being the network switch. Also, the diagram shows a derivative from pure IMA where functions like flight controls, infotainment, and communications might use an OSA product line rather than common IMA building blocks. Flight controls are possible in pure IMA but are normally avoided due to common mode concerns (unless the IMA architecture includes dissimilar design, which is not standard in civil markets). Infotainment and communications are the case where IP networking may require more extensive IP support and more standards with regard to the switch infrastructure. However, even in these cases, some bridge to the ARINC 664 Part 7 network is required.
	С. 76.	**С. 16.**
	Трафік ARINC 664, частина 7, обмежений затримкою та має витримувати значне тремтіння, яке вимагає асинхронного архітектурного інтерфейсу між системами. На відміну від цього, Ethernet із синхронізованим часом забезпечує критичний до часу потік даних для синхронних архітектур. Це забезпечує детерміновану мережу Ethernet, яка може бути більш придатною для критичних за часом функцій літака, таких як управління польотом. Синхронна природа Ethernet, що запускається за часом, також підходить для архітектур розподіленої обробки, оскільки забезпечує жорсткий контроль тремтіння даних і суворий детермінізм між елементами обробки. Завдання обробки синхронізуються з періодичними мережевими передачами даних із множинним доступом із розділенням часу (TDMA) [13]. Подібно до того, як ARINC 653 забезпечує розподіл за допомогою розподілу часу на процесорі, протокол Ethernet із синхронізацією за	ARINC 664 Part7 traffic is latency bounded and must tolerate appreciable jitter, which requires an asynchronous architectural interface between systems. In contrast, time-triggered Ethernet provides a time-critical dataflow for synchronous architectures. This provides a deterministic Ethernet network that can be more suitable for time-critical aircraft functions such as flight controls. The synchronous nature of time-triggered Ethernet also lends itself to distributed processing architectures since it provides tight control of data jitter and strict determinism between processing elements. Processing tasks are synchronized with periodic time division multiple access (TDMA) network data transfers [13]. Just like ARINC 653 provides partitioning through time-sharing on a processing unit, the timetriggered Ethernet protocol provides similar partitioning between critical and noncritical communications through predefined time slots on the network. This temporal partitioning allows multiple traffic clas-

	часом забезпечує аналогічний розподіл між критичними та некритичними зв'язками через попередньо визначені часові інтервали в мережі. Це тимчасове розділення дозволяє кільком класам трафіку співіснувати в одній мережі без перешкод. Багатоцільовий транспортний засіб екіпажу NASA, що розробляється, Orion, використовує технологію Ethernet із синхронізованим часом як магістраль даних, що дозволяє співіснувати трьом класам трафіку в одній мережі [14]. У військових системах авіоніки FC-AEASM використовувався для реалізації запланованих мереж як з періодичними, так і з асинхронними операціями як попередник повної можливості SAE AS6802, але з використанням Fibre Channel, а не Ethernet. Там, де плановані мережі не потрібні, у військових системах місії OSA широко використовується стандартна мережа з комутацією GigE разом із додаванням шлюзів безпеки та маршрутизаторів. **Коваленко переписала чужий текст разом із номерами покликань, під якими в її дисертації знаходяться зовсім інші джерела.** **Фальсифікація джерел.** **Плагіат.**	ses to coexist on the same network without interference. NASA's multipurpose crew vehicle under development, Orion, utilizes time-triggered Ethernet technology as its data backbone, allowing three traffic classes to coexist on the same network [14]. In military avionics systems, FC-AE-ASM has been used to implement schedulable networks with both periodic and asynchronous operations as a precursor to full SAE AS6802 capability but using Fibre Channel rather than Ethernet. Where schedulable networks are not required, there is significant use of standard GigE switched networking in military OSA mission systems along with the addition of security gateways and routers.
8	**Коваленко Ю. Б.** **Інформаційна підтримка процесу проектування та експлуатації комплексу бортового обладнання інтегрованої модульної авіоніки.** – Дис. … доктора технічних наук. – Київ, 2025. (https://duikt.edu.ua/uploads/p_86_93515477.pdf)	**Spitzer C. R. (Ed.)** **The Avionics Handbook.** – CRC Press LLC, 2001. (https://www.twirpx.com/file/2830309/)
	С. 146.	**С. 500.**
	Як обговорювалося в другому розділі, збій системи — це втрата системних служб або очікуваної функціональності. За відсутності відмовостійкості система може вийти з ладу лише після однієї критичної помилки. Така система, яка фактично є безвідмовною працездатністю, була б допустимою для некритичних функцій, крім того, характеризує цей тип системи, оскільки безперервне обслуговування залежить від спонтанної ремісії активної несправності або несправності, наслідки якої не є настільки серйозними, щоб призвести до відмови системи. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	**28.2.1 General Mechanization** As discussed in Section 28.1.2, system failure is the loss of system services or expected functionality. In the absence of fault tolerance, a system may fail after just a single crucial fault. This kind of system, which in effect is zero fail-operational, would be permissible for non-critical functions. Figure 28.2, moreover, characterizes this kind of system in that continued service depends on spontaneous remission of an active fault or a fault whose consequences are not serious enough to yield a system failure.
	С. 146.	**С. 500.**
	Якщо ймовірність безперервного обслуговування має бути високою, можна включити резервування, щоб забезпечити працездатність системи за наявності будь-яких постійних збоїв. Такі відмовостійкі системи містять додатковий стан несправності, а саме стан відновлення, як показано на рисунку 3.7.	Where the likelihood of continued service must be high, redundancy can be incorporated to ensure system operability in the presence of any permanent fault(s). Such fault-tolerant systems incorporate an additional fault status state, namely that of recovery, as shown in Figure 28.4.
	С. 146.	**С. 501.**

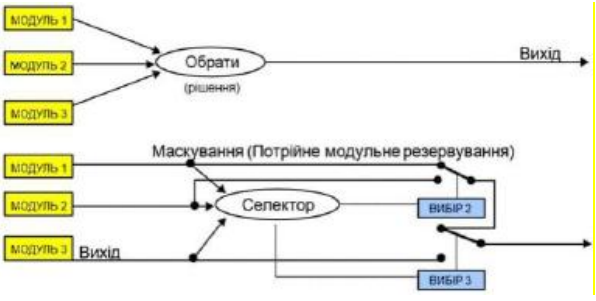
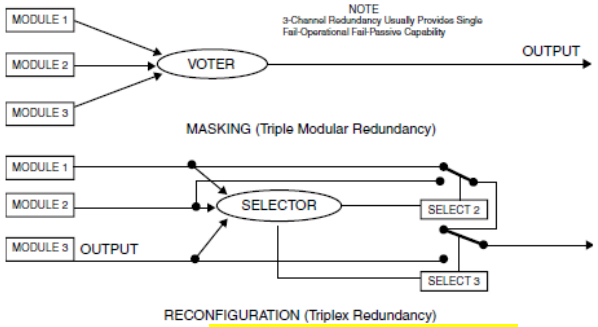
	 Рис. 3.7. Схема стану обладнання з коригуючою дією.	 FIGURE 28.4 Hardware states (with corrective action).
	С. 146.	**С. 500.**
	Збій системи виникає лише після вичерпання запасних частин або невирішеної серйозної несправності. Вищезазначений рівень резервування може зробити вкрай малоймовірним, що запасні частини будуть вичерпані лише в результаті несправності обладнання. Необроблена помилка може виникнути лише як наслідок помилки проектування, як-от вчинення загальної помилки, коли наявність помилки навіть не буде виявлено.	Here, system failure occurs only after the exhaustion of spares or an unhandled severe fault. The aforementioned level of redundancy can render it extremely unlikely that the spares will be exhausted as a result of hardware faults alone. An unhandled fault could occur only as a consequence of a design error, like the commission of a generic error wherein the presence of a fault would not even be detected.
	С. 147.	**С. 500.**
	У цьому розділі розглядається перспектива системного рівня та досліджуються відмовостійкі системні архітектури та їх приклади. Тим не менш, ці приклади втілюють і висвітлюють загальні принципи відмовостійкості системного рівня. Особлива увага приділяється системам керування польотом, оскільки вони мотивували та запровадили багато технологій відмовостійкості у застосуванні до цифрових систем польоту. У минулому такі системи були функціонально виділеними, таким чином забезпечуючи значний захист від несправностей через сторонні причини. Проте зі збільшенням поширення інтегрованої авіоніки ступінь розподілу функцій безповоротно зменшується. Насправді це не зовсім доцільно, більше функцій авіоніки, ніж будь-коли, є критично важливими, і ряд переваг отримується від інтегрованої обробки. Крім того, розробник системи може використовувати прерогативи, які забезпечують захист від сторонніх помилок.	This section assumes a system-level perspective, and undertakes to examine fault-tolerant system architectures and examples thereof. Still, these examples embody and illuminate general system-level principles of fault tolerance. Particular prominence is directed toward flight control systems, for they have motivated and pioneered much of the fault tolerance technology as applied to digital flight systems. In the past, such systems have been functionally dedicated, thereby providing a considerable safeguard against malfunction due to extraneous causes. With the increasing prevalence of integrated avionics, however, the degree of function separation is irretrievably reduced. This is actually not altogether detrimental, more avionics functions than ever are critical, and a number of benefits accrue from integrated processing. Furthermore, the system developer can exercise prerogatives that afford safeguards against extraneous faults.
	С. 147.	**С. 501.**
	Відмовостійкість зазвичай базується на певній формі резервування для підвищення надійності системи за допомогою виклику альтернативних ресурсів. Надмірність може полягати в апаратному забезпеченні, програмному забезпеченні, часі або їх комбінаціях. Існує три основних типи резервування апаратного та програмного забезпечення: статичне, динамічне та гібридне. Статичне резервування маскує помилки, беручи більшість результатів від повторюваних завдань. Динамічне резервування передбачає двоетапну процедуру для виявлення та відновлення збоїв. Гібридна надлишковість — це комбінація статичної та динамічної надлишковості. **Коваленко переписала чужий текст, видаливши заголовок і покликання.** **Плагіат.**	**28.2.2 Redundancy Options** Fault tolerance is usually based on some form of redundancy to extend system reliability through the invocation of alternative resources. The redundancy may be in hardware, software, time, or combinations thereof. There are three basic types of redundancy in hardware and software: static, dynamic, and hybrid. Static redundancy masks faults by taking a majority of the results from replicated tasks. Dynamic redundancy takes a two-step procedure for detection of, and recovery from faults. Hybrid redundancy is a combination of static and dynamic redundancy [Shin and Hagbae, 1994].

	С. 147–148.	**С. 501.**
	Загалом, велика частина цієї надмірності міститься в додаткових апаратних компонентах. Додавання компонентів скорочує середній час між діями з технічного обслуговування, оскільки є більше електроніки, яка може, і в якийсь момент, відмовить. Оскільки у відмовостійких системах можуть виникати численні різні несправності, існує багато додаткових режимів відмов, які необхідно оцінити при встановленні льотної придатності всієї системи. Вага, енергоспоживання, охолодження тощо є іншими покараннями за надмірність компонентів. Інші форми резервування також викликають накладні витрати на керування системою, наприклад, обчислювальну потужність для виконання програмно реалізованих завдань відмовостійкості. Отже, проектування, реалізація відмовостійкості передбачає компроміси та необхідність оптимізації дизайну. Зрештою, слід шукати збалансовану, мінімальну та придатну для валідації конструкцію, яка б наочно забезпечувала гарантії та межі стійкості до несправностей, що відповідають даному застосуванню. **Коваленко пише «Отже», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Плагіат.**	In general, much of this redundancy resides in additional hardware components. The addition of components reduces the mean-time-between-maintenance actions, because there are more electronics that can, and at some point will, fail. Since multiple, distinct faults can occur in fault-tolerant systems, there are many additional failure modes that have to be evaluated in establishing the airworthiness of the total system. Weight, power consumption, cooling, etc. are other penalties for component redundancy. Other forms of redundancy also present system management overhead demands, like computational capacity to perform software-implemented fault tolerance tasks. Like all design undertakings, the realization of fault tolerance presents trade-offs and the necessity for design optimization. Ultimately, a balanced, minimal, and validatable design must be sought that demonstrably provides the safeguards and fault survival margins appropriate to the subject application.
	С. 148.	**С. 501.**
	Існує широкий діапазон варіантів реалізації резервування для механізації бажаних рівнів і типів відмовостійкості. На рисунку 3.8 представлено класифікацію варіантів резервування, які можуть бути використані у відповідних комбінаціях, щоб отримати всеохоплюючу відмовостійку архітектуру.	A broad range of redundancy implementation options exist to mechanize desired levels and types of fault tolerance. Figure 28.5 presents a taxonomy of redundancy options that may be invoked in appropriate combinations to yield an encompassing fault tolerance architecture.
	С. 148.	**С. 502.**
	 Рис. 3.8.Структура класифікації резервування.	 FIGURE 28.5 Redundancy classification.
	С. 148–149.	**С. 501.**
	Ця таксономія вказує на широкий спектр можливостей резервування, які можуть бути використані при проектуванні системи. Хоча більшість із цих варіантів наведено в якості прикладів або описано в наступних параграфах, можна коротко зазначити, що резервування характеризується його категорією, режимом, координацією та аспектами складу. Архітектурні зобов'язання мають бути зроблені в кожному аспекті. Таким чином, класична система може, наприклад, використовувати маскування несправностей за допомогою дубльованих апаратних модулів, які працюють синхронно. На нижчому рівні пов'язані шини даних можуть використовувати надлишкове кодування для виявлення та виправлення помилок. Для захисту від загальної помилки проектування можна додати можливість резервного копіювання за допомогою іншої схеми резервування.	This taxonomy indicates the broad range of redundancy possibilities that may be invoked in system design. Although most of these options are exemplified or described in later paragraphs, it may be noted briefly that redundancy is characterized by its category, mode, coordination, and composition aspects. Architectural commitments have to be made in each aspect. Thus, a classical system might, for example, employ fault masking using replicated hardware modules that operate synchronously. At a lower level, the associated data buses might use redundancy encoding for error detection and correction. To safeguard against a generic design error, a backup capability might be added using a dissimilar redundancy scheme.
	С. 149.	**С. 502.**

	Зокрема, розглянуті елементи маскування проти реконфігурації, активних і резервних запасних частин, а також реплікованого та різнорідного резервування. Найбільш економічно ефективна фаза загального процесу проектування та розробки для зменшення ймовірності CMFs є найранішою частиною програми. У таблиці 3.1 представлені методи уникнення помилок та інструменти, які використовуються. Методи та інструменти усунення несправностей загального режиму включають перевірку дизайну, моделювання, тестування, впровадження несправностей і програму контролю якості. **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	<…> In particular, the elements of masking vs. reconfiguration, active vs. standby spares, and replicated vs. dissimilar redundancy are reviewed here. No unifying theory has been developed that can treat CMFs the same way the Byzantine resilience (BR) treats random hardware or physical operational faults. Three techniques, fault-avoidance, faultremoval, and fault-tolerance are the tools available to design a system tolerant of CMFs. The most cost effective phase of the total design and development process for reducing the likelihood of CMFs is the earliest part of the program. Table 28.4 presents fault avoidance techniques and tools that are being used [Lala and Harper, 1994]. Common-mode fault removal techniques and tools include design reviews, simulation, testing, fault injection, and a rigorous quality control program.
	С. 149.	**С. 502.**
	Таблиця 3.1 Методи та інструменти уникнення несправностей Техніка Використання зрілих і офіційно перевірених компонентів Відповідність стандартам Формальні методи Автоматизація проектування Інтегровані формальні методи та методологія проектування VHDL Спрощення абстракцій Уникнення збоїв загального режиму продуктивності Практика розробки програмного та апаратного забезпечення Різноманітність дизайну	TABLE 28.4 Fault Avoidance Techniques and Tools Technique Use of mature and formally verified components Conformance to standards Formal methods Design automation Integrated formal methods and VHDL design methodology Simplifying abstractions Performance common-mode failure avoidance Software and hardware engineering practice Design diversity
	С. 149–150.	**С. 502.**
	Відмовостійкість загального режиму вимагає виявлення та відновлення помилок. Необхідно підтвердити інформацію про помилку через резервні канали, щоб визначити, який механізм відновлення (тобто фізичне відновлення або відновлення збоїв у загальному режимі) використовувати. Відновлення з CMF у реальному часі вимагає відновлення системи до попередньо відомої правильної точки, з якої обчислювальна діяльність може відновитися. **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	Common-mode fault tolerance requires error detection and recovery. It is necessary to corroborate the error information across redundant channels to ascertain which recovery mechanism (i.e., physical fault recovery, or common-mode failure recovery) to use. Recovery from CMF in real time requires that the state of the system be restored to a previously known correct point from which the computational activity can resume [Lala and Harper, 1994].
	С. 150.	**С. 502.**
	Як показано на рис. 3.8, існує три категорії відмовостійких архітектур: маскування, реконфігурація та гібрид. Підхід маскування є класичним за концепцією потрійного модульного резервування (TMR) фон Неймана, яка була узагальнена для довільних рівнів резервування. Концепція TMR зосереджується на виборці, який у межах обмеження виснаження запасів запобігає проходженню сигналу з помилкою	**28.2.3 Architectural Categories** As indicated in Figure 28.5, the three categories of fault-tolerant architectures are masking, reconfiguration, and hybrid. **28.2.3.1 Fault Masking** The masking approach is classical per von Neumann’s triple modular redundancy (TMR) concept, which has been generalized for arbitrary levels of re-

	по шляху сигналу. Цей підхід є пасивним, оскільки не потрібно змінювати конфігурацію, щоб запобігти поширенню помилкового стану або ізолювати несправність. **Коваленко переписала чужий текст, видаливши заголовки.** **Плагіат.**	dundancy. The TMR concept centers on a voter that, within a spares exhaustion constraint, precludes a faulted signal from proceeding along a signal path. The approach is passive in that no reconfiguration is required to prevent the propagation of an erroneous state or to isolate a fault.
	С. 150–151.	**С. 503.**
	Модульні системи авіоніки, що складаються з кількох ідентичних модулів і вибірковий елемент, вимагають компромісу між надійністю та безпекою. «Модуль» не може бути апаратним модулем; модуль представляє сутність, здатну виробляти результат. Коли безпека розглядається разом з надійністю, конструкція модуля впливає як на безпеку, так і на надійність. Зазвичай очікується, що надійність і безпека підвищаться з додаванням резервування. Якщо модуль має вбудовану функцію виявлення помилок, можна підвищити як надійність, так і безпеку за допомогою додавання одного модуля, який надає дані виборцеві. Якщо на рівні модуля немає можливості виявлення помилок, для підвищення надійності та безпеки потрібні принаймні два додаткові модулі. Арбітражна стратегія контролю помилок — це функція, яку реалізує виборець, щоб вирішити, що є правильним виходом, і коли помилки у виходах модуля є надмірними, тому правильний вихід не може бути визначений, виборець може подати небезпечний сигнал. Надійність і безпека n-module safe modular redundant (nSMR) залежать від надійності окремого модуля та від конкретної стратегії арбітражу, що використовується. Жодна стратегія арбітражу не є оптимальною для підвищення як надійності, так і безпеки. Надійність визначається як ймовірність того, що вихідні дані вибірковий елемент правильні, і виборець не стверджує про небезпечний сигнал. Безпека надійність плюс ймовірність того, що виборець стверджує про небезпечний сигнал. Оскільки надійність і безпека системи взаємопов'язані, підвищення надійності системи може призвести до зниження безпеки системи, і навпаки. **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	Modular avionics systems consisting of multiple identical modules and a voter require a trade-off of reliability and safety. A “module” is not constrained to be a hardware module; a module represents an entity capable of producing an output. When safety is considered along with reliability, the module design affects both safety and reliability. It is usually expected that reliability and safety should improve with added redundancy. If a module has built-in error detection capability, it is possible to increase both reliability and safety with the addition of one module providing input to a voter. If no error detection capability exists at the module level, at least two additional modules are required to improve both reliability and safety. An error control arbitration strategy is the function implemented by the voter to decide what is the correct output, and when the errors in the module outputs are excessive so that the correct output cannot be determined, the voter may put out an unsafe signal. Reliability and safety of an n-module safe modular redundant (nSMR) depend on the individual module reliability and on the particular arbitration strategy used. No single arbitration strategy is optimal for improving both reliability and safety. Reliability is defined as the probability the voter’s data output is correct and the voter does not assert the unsafe signal. Safety = Reliability plus the probability the voter asserts the unsafe signal. As system reliability and safety are interrelated, increasing system reliability may result in a decrease in system safety, and vice versa [Vaidya and Pradhan, 1993].
	С. 151.	**С. 503.**
	Вибіркові елементи, які використовують порівняння біт за бітами, використовуються, коли помилки складаються з довільної поведінки з боку несправних компонентів, навіть до екстремальних проявів, здавалося б, розумної зловмисної поведінки. Такі розломи отримали назву візантійських. Вимоги, що висуваються до архітектури, стійкої до візантійських помилок (іменованої стійкою до візантійської архітектури [BR]), включають нижню межу кількості областей локалізації помилок, їх підключення, синхронності та використання певних простих протоколів обміну інформацією. Жодних апріорних припущень щодо поведінки компонентів не потрібно при використанні порівняння біт за біт. Домінуючим фактором відмови правильно розробленої архітектури системи BR є збій синфазного режиму.	Voters that use bit-for-bit comparison have been employed when faults consist of arbitrary behavior on the part of failed components, even to the extreme of displaying seemingly intelligent malicious behavior [Lala and Harper, 1994]. Such faults have been called Byzantine faults. Requirements levied on an architecture tolerant of Byzantine faults (referred to as Byzantine-resilient [BR]) comprise a lower bound on the number of fault containment regions, their connectivity, their synchrony, and the utilization of certain simple information exchange protocols. No a priori assumptions about component behavior are required when using bit-for-bit comparison. The dominant contributor to failure of correctly designed BR system architecture is the common-mode failure.

	Коваленко переписала чужий текст, видаливши покликання. **Плагіат.**	
	С. 151–152.	**С. 503.**
	Наслідки несправності необхідно маскувати, доки не можна буде вжити заходів щодо відновлення. Необхідно керувати резервною системою, щоб продовжувати правильну роботу за наявності несправності. Одним з підходів є розділення надлишкових елементів на окремі зони локалізації несправності (FCR). FCR — це набір компонентів, який працює належним чином незалежно від будь-якої логічної чи електричної несправності за межами регіону. Межа локалізації несправностей вимагає, щоб апаратні компоненти були забезпечені незалежними джерелами живлення та синхронізації. Інтерфейси між FCR повинні бути електрично ізольовані. Стійкість до такого фізичного пошкодження, як попадання зброї, вимагає фізичного розділення FCR, наприклад, різних відсіків авіоніки. У системах керування польотом канал може бути природним FCR. Наслідки несправності, що проявляються як помилкові дані, можуть поширюватися через межі FCR. Таким чином, система також повинна забезпечувати стримування помилок. Це робиться за допомогою виборців у різних точках обробки, включаючи голосування за надлишковими входами, голосування за результат обчислень закону керування та голосування на вході приводу. Маскування несправностей і помилок забезпечує коректну роботу системи з необхідністю негайної оцінки пошкоджень, ізоляції несправностей і реконфігурації системи. **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	Fault effects must be masked until recovery measures can be taken. A redundant system must be managed to continue correct operation in the presence of a fault. One approach is to partition the redundant elements into individual fault containment regions (FCRs). An FCR is a collection of components that operates correctly regardless of any arbitrary logical or electrical fault outside the region. A fault containment boundary requires the hardware components be provided with independent power and clock sources. Interfaces between FCRs must be electrically isolated. Tolerance to such physical damage as a weapons hit necessitates a physical separation of FCRs such as different avionics bays. In flight control systems, a channel may be a natural FCR. Fault effects manifested as erroneous data can propagate across FCR boundaries. Therefore, the system must provide error containment as well. This is done using voters at various points in the processing including voting on redundant inputs, voting the result of control law computations, and voting at the input to the actuator. Masking faults and errors provides correct operation of the system with the need for immediate damage assessment, fault isolation, and system reconfiguration [Lala and Harper, 1994].
	С. 152.	**С. 503.**
	Апаратні блокування забезпечують перший рівень захисту перед реконфігурацією або використанням решти несправних каналів. У триплексній або вищій системі резервування більшість каналів можуть вимкнути вихід несправного каналу. Перш ніж виконати цю дію, система визначить, чи є збій постійним чи тимчасовим. Після того, як система визначить, що збій є постійним або постійним, наступним кроком є визначення того, які функції потрібні для решти місії та чи потрібно системі викликати оцінку збитків, ізоляцію збоїв і реконфігурацію решти системних активів. Розробник системи, необхідної для довготривалих місій, може взяти на себе реалізацію проекту, що має можливість реконфігурації. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	**28.2.3.2 Reconfiguration** Hardware interlocks provide the first level of defense prior to reconfiguration or the use of the remaining non-faulty channels. In a triplex or higher redundancy system, the majority of channels can disable the output of a failed channel. Prior to taking this action, the system will determine whether the failure is permanent or transient. Once the system determines a fault is permanent or persistent, the next step is to ascertain what functions are required for the remainder of the mission and whether the system needs to invoke damage assessment, fault isolation, and reconfiguration of the remaining system assets. The designer of a system required for long-duration missions may undertake to implement a design having reconfiguration capability.
	С. 152.	**С. 503–504.**
	Гібридна відмовостійкість використовує гібридне резервування, яке є комбінацією статичної та	**28.2.3.3 Hybrid Fault Tolerance** Hybrid fault tolerance uses hybrid redundancy, which is a combination of static and dynamic redun-

	динамічної надмірності, тобто маскування, виявлення та відновлення, що може передбачати реконфігурацію. Система, яка використовує гібридне резервування, матиме N-активні резервні модулі, а також запасні (S) модулі. Детектор розбіжностей визначає, чи вихідні дані будь-якого з активних модулів відрізняються від вихідних даних вибірковий елемент. Якщо вихідний сигнал модуля не відповідає виборцю, схема комутації замінює несправний модуль на запасний. Гібридна (N, S) система не може мати більше (N-1)/2 несправних модулів одночасно в ядрі, інакше система неправильно вимкне справний модуль, якщо два з трьох вийшли з ладу. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	dancy, i.e., masking, detection, and recovery that may involve reconfiguration. A system using hybrid redundancy will have N-active redundant modules, as well as spare (S) modules. A disagreement detector detects if the output of any of the active modules is different from the voter output. If a module output disagrees with the voter, the switching circuit replaces the failed module with a spare. A hybrid (N,S) system cannot have more than (N-1)/2 failed modules at a time in the core, or the system will incorrectly switch out the good module when two out of three have failed.
	С. 152–153.	**С. 504.**
	Гібридна відмовостійкість використовує комбінацію маскування та реконфігурації, як зазначено в другому розділі. Намір полягає в тому, щоб використати сильні сторони обох підходів для досягнення чудової відмовостійкості. Маскування запобігає впливу помилкового стану на роботу системи, таким чином усуваючи потребу у виправленні помилок. Реконфігурація видаляє помилкові входи для вибірковий елемент, щоб численні помилки не могли перемогти вибірковий елемент. Дії маскування та реконфігурації зазвичай реалізуються в механізмі порівняння виборців, який обговорюється в розділі На рисунку 3.9 зображено гібридне розташування TMR із резервним каналом, що забезпечує подвійну можливість роботи при збоях. При виході з ладу першого активного каналу, він вимикається з конфігурації введення вибірковий елемент, і вмикається резервний канал. При виході з ладу другого каналу, невідповідний вхід для вибірковий елемент вимикається. Тоді залишаються лише два входи, тому наступний (третій) збій каналу може бути виявлений, але не ідентифікований виборцем як таким. З виборцем, який вибирає нижчий із двох сигналів, що залишилися, і, отже, виключає вихідний сигнал, постійне неправильне порівняння призводить до втрати функції системи при відмові. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	**28.2.3.4 Hybrid Fault Tolerance** Hybrid fault tolerance employs a combination of masking and reconfiguration, as noted in Section 28.2.3. The intent is to draw on strengths of both approaches to achieve superior fault tolerance. Masking precludes an erroneous state from affecting system operation and thus obviating the need for error recovery. Reconfiguration removes faulted inputs to the voter so that multiple faults cannot defeat the voter. Masking and reconfiguration actions are typically implemented in a voter-comparator mechanism, which is discussed in Section 28.3.1. Figure 28.6 depicts a hybrid TMR arrangement with a standby spare channel to yield double failoperational capability. Upon the first active channel failure, it is switched out of the voter-input configuration, and the standby channel is switched in. Upon a second channel failure, the discrepant input to the voter is switched out. Only two inputs remain then, so a succeeding (third) channel failure can be detected but not properly be identified by the voter per se. With a voter that selects the lower of two remaining signals, and hence precludes a hardover output, a persistent miscomparison results in a failpassive loss of system function.
	С. 153.	**С. 504.**
	 Рис. 3.9. Маскування проти реконфігурації.	 FIGURE 28.6 Masking vs. Reconfiguration.

	С. 153–154.	**C. 504.**
	Альтернативна робоча конфігурація з подвійною невдачею відмовилася б від перемикання резервних каналів і просто використовувала б квадруплексний виборець. Ця архітектура насправді досить поширена в спеціальних критично важливих системах, таких як системи управління польотом Fly-by-wire (FBW). Ця архітектура все ще використовує реконфігурацію для видалення помилкових вхідних даних для вибіркового елемента. Елементи проектування відмовостійкості, описані в розділі нижче, відображені у відмовостійкій архітектурі на рис. 45. Наприклад, виявлення помилок забезпечується компараторами; потім оцінка пошкоджень виконується логікою реконфігурації з використанням різних станів компаратора. усунення несправностей і продовження служби реалізуються через вибір, що також усуває потребу у виправленні помилок. І, нарешті, усунення несправності виконується шляхом перемикання несправного шляху, запропонованого логікою реконфігурації. Таким чином, цей простий приклад ілюструє на високому рівні, як різні аспекти відмовостійкості можуть бути включені в інтегрований дизайн.	An alternative double-fail operational configuration would forego the standby channel switching and simply employ a quadruplex voter. This architecture is actually rather prevalent in dedicated flight-critical systems like fly-by-wire (FBW) flight control systems. This architecture still employs reconfiguration to remove faulty inputs to the voter. The fault tolerance design elements described in Section 28.1.6 are reflected in the fault-tolerant architecture in Figure 28.6 by way of annotations. For example, error detection is provided by the comparators; damage assessment is then accomplished by the reconfiguration logic using the various comparator states. fault containment and service continuation are both realized through the voter, which also obviates the need for error recovery. Last, fault treatment is accomplished by the faulty path switching prompted be the reconfiguration logic. Thus, this simple example illustrates at a high level how the various aspects of fault tolerance can be incorporated into an integrated design.
	С. 154.	**C. 505.**
	Отже, розробники повинні враховувати ці вразливості при проектуванні відмовостійкої системи. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	**28.2.4 Integrated Mission Avionics** In military applications, redundant installations in some form will be made on opposite sides of the aircraft to avoid loss of functionality from battle damage to a single installation. Vulnerability to physical damage also exists in the integrated rack installations being used on commercial aircraft. Designers must take these vulnerabilities into account in the design of a fault-tolerant system.
	С. 154–155.	**C. 505–506.**
	Аналіз надійності системи авіоніки залежить від припущень про готовність системи до руху. Для систем з меншою критичністю певне зменшення надлишковості іноді може бути прийнятним під час руху. Однак для постійно важливих для польоту систем, як правило, є повністю працездатна система з усім резервуванням. Передбачається в прогнозі надійності системи. Це припущення висуває значні вимоги до самоперевірки системи перед польотом щодо охоплення та достовірних значень. Такий тест, як правило, є наскрізним тестом, який виконує всі елементи поетапно, що було б неможливо під час польоту. Положення про відмовостійкість вимагають особливої уваги. Наприклад, таке тестування змушує рідко використовувані відключення компаратора, щоб переконатися у відсутності прихованих несправностей, таких як пасивні збої обладнання. Аналіз пов'язаних схем тестування та їхнього охоплення обов'язково є поточним завданням аналізу проекту під час розробки. Ці схеми також повинні включати відповідні логічні блокування для забезпечення безпечного виконання попередніх випробування, наприклад, блокування ваги на колесах, щоб ви-	**28.2.5 System Self Tests** Avionics system reliability analyses are conditional on assumptions of system readiness at dispatch. For lower-criticality systems, certain reductions in redundancy may sometimes be tolerable at dispatch. For full-time flight-critical systems, however, a fully operable system with all redundancy intact is generally assumed in a system reliability prediction. This assumption places appreciable demands on system preflight self test in terms of coverage and confidence values. Such a test is typically an end-to-end test that exercises all elements in a phased manner that would not be possible during flight. The fault tolerance provisions demand particular emphasis. For example, such testing deliberately seeks to force seldom-used comparator trips to ensure the absence of latent faults, like passive hardware failures. Analysis of associated testing schemes and their scope of coverage is necessarily an ongoing design analysis task during development. These schemes must also include appropriate logic interlocks to ensure safe execution of the preflight test, e.g., a weight-on-wheels interlock to preclude testing except on the ground. Fortunately, the programming of system self-tests can be accomplished in a relatively complete and high-fidelity man-

	ключити тестування, окрім як на землі. Оскільки, програмування системних самоперевірок може бути виконано відносно повним і високоточним способом. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	ner.
	С. 155.	**С. 506.**
	Через характер дискретного часу цифрових систем не вся ємність використовується для прикладних функцій. Таким чином, під час польоту можливі періодичні самоперевірки цифрових компонентів, таких як процесори. Також процесори можуть періодично перевіряти стан справності інших компонентів системи. Такі тести забезпечують самоконтроль, який може виявити наявність невідповідності до появи станів помилки або перевищення порогів виявлення. Час виконання самотестування може бути значним, оскільки постійний політ може не імітувати поїздки компаратора через сигнали низької амплітуди. Крім того, чим довше несправність залишається латентною, тим більша ймовірність появи другої несправності. Таким чином, періодичні самоперевірки можуть значно підвищити надійність і безпеку системи шляхом зменшення впливу множинних одночасних проявів несправностей. Самоконтроль може використовуватися на ще нижчих рівнях, але існує компроміс щодо деталізації виявлення несправностей. Цей компроміс залежить від відповіді на виявлення/відновлення помилок і вибраного рівня локалізації помилок. Загалом стримування несправностей визначає деталізацію виявлення несправностей, якщо тільки час реакції відновлення не вимагає швидшого виявлення несправностей, що найкраще досягається на нижчих рівнях.	Because of the discrete-time nature of digital systems, all capacity is not used for application functions. Hence, periodic self tests are possible for digital components like processors during flight. Also, the processors can periodically examine the health status of other system components. Such tests provide a self-monitoring that can reveal the presence of a discrepancy before error states are introduced or exceed detection thresholds. The lead time afforded by self tests can be substantial because steady flight may not simulate comparator trips due to low-amplitude signals. Moreover, the longer a fault remains latent, the greater the possibility that a second fault can occur. Hence, periodic self-tests can significantly enhance system reliability and safety by reducing exposure to coincident multiple fault manifestations. Self-monitoring may be employed at still lower levels, but there is a trade-off as to the granularity of fault detection. This trade-off keys on fault detection/recovery response and on the level of fault containment selected. In general, fault containment delineates the granularity of fault detection unless recovery response times dictate faster fault detection that is best achieved at lower levels.
	С. 155–156.	**С. 506.**
	Компаратори обрання дуже широко використовуються в стійких до відмов системах авіоніки, і вони, як правило, життєво важливі для цілісності та безпеки пов’язаних систем. Через вирішальну роль компараторів для виборців необхідно приділяти особливу увагу їх розвитку. Ці динамічні елементи системи, які можуть бути реалізовані як програмним, так і апаратним забезпеченням, не такі прості, як може здатися. Зокрема, цілісність пристрою та налаштування порогових параметрів можуть бути проблематичними. Певні основні елементи та проблеми застосовуються до діапазону варіантів порівняння виборців. Концептуальний вигляд триплексного вибірковий елемент-компаратора зображено на рисунку 3.10. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	**28.3.1 Voter Comparators** Voter comparators are very widely used in fault-tolerant avionics systems, and they are generally vital to the integrity and safety of the associated systems. Because of the crucial role of voter comparators, special care must be exercised in their development. These dynamic system elements, which can be implemented in software as well as hardware, are not as simple as they might seem. In particular, device integrity and threshold parameter settings can be problematic. Certain basic elements and concerns apply over the range of voter-comparator variants. A conceptual view of a triplex voter-comparator is depicted in Figure 28.7.
	С. 156.	**С. 505.**

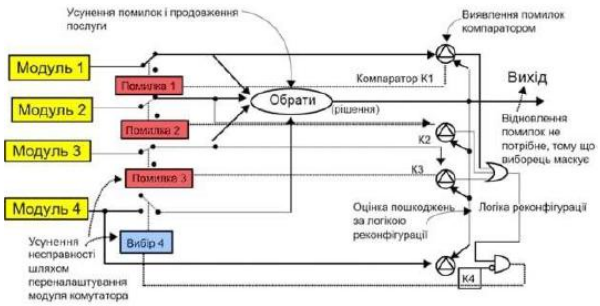


Рис. 3.10. Гібридна схема TMR.

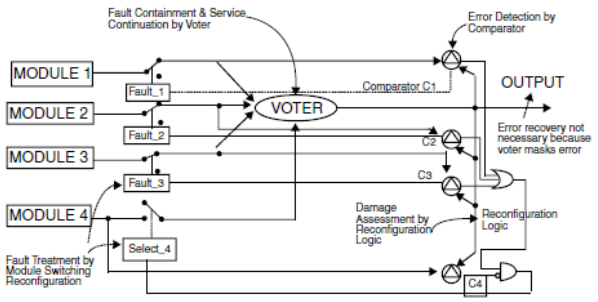


FIGURE 28.7 Hybrid TMR arrangement.

С. 156–157.

С. 506.

Вибірковий елемент у цьому випадку реалізований як середній селектор сигналу, тобто як вихід обирається середнє (проміжне) значення з трьох вхідних сигналів. Блок вибіркового елементу розташований перед компараторами, оскільки саме його вихід подається на вхід кожного компаратора. Вихід вибіркового елемента розглядається як еталонний (опорний) сигнал, а будь-який вхідний сигнал, що суттєво відхиляється від цього еталона, вважається помилковим.

На рис. 3.10 показано, що відповідні входи кожного з сигналових каналів є амплітудно-модульованими серіями імпульсів, як це типово для цифрової обробки сигналів. Кожна ітерація роботи вибірковий елемент є незалежним актом вибору, тому вихідний сигнал може бути сформований на основі будь-якого з трьох вхідних каналів.

Це видно на рисунку 3.11, де компоненти вихідної серії імпульсів пронумеровані для вхідного шляху, вибраного в кожен момент часу. З кожним кроком часу вихід вибірковий елемент подається на кожен із компараторів, а різниця з кожним вхідним сигналом подається на відповідний пороговий детектор амплітуди. Поріг амплітуди встановлюється таким чином, щоб накопичені допуски не спровокували спрацьовування детектора. Як показано, детектор амплітуди видає встановлений вихід, коли вперше спостерігається надмірна різниця. Коли різниця знову падає в межах порогу, видається вихід скидання.

The voter here is taken to be a middle signal selector, which means that the intermediate level of three inputs is selected as the output. The voter section precedes the comparators because the output of the voter is an input to each comparator. Basically, the voter output is considered the standard of correctness, and any input signal that persists in varying too much from the standard is adjudged to be erroneous.

In Figure 28.7, the respective inputs to each of the signal paths is an amplitude-modulated pulse train, as is normal in digital processing. Each iteration of the voter is a separate selection, so each voter output is apt to derive from any input path. This is seen in Figure 28.8, where the output pulse train components are numbered per the input path selected at each point in time. At each increment of time, the voter output is applied to each of the comparators, and the difference with each input signal is fed to a corresponding amplitude threshold detector. The amplitude threshold is set so that accumulated tolerances are not apt to trip the detector. As shown here, the amplitude detector issues a set output when an excessive difference is first observed. When the difference falls back within the threshold, a reset output is issued.

С. 157.

С. 505.

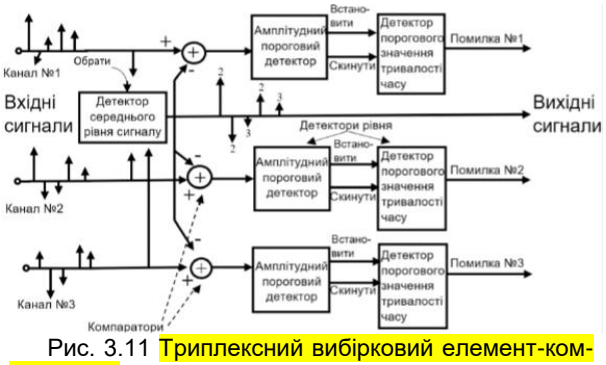


Рис. 3.11 Триплексний вибірковий елемент-компаратора.

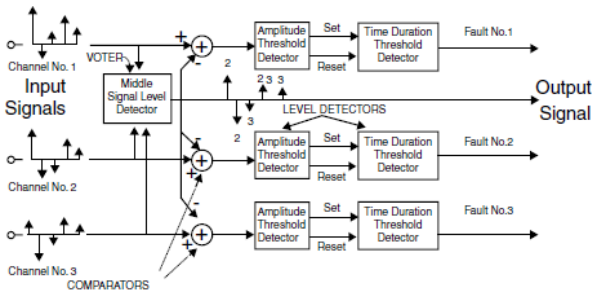


FIGURE 28.8 Triplex voter-comparator.

С. 157.

С. 507.

Оскільки перехідні процеси можуть призвести до короткочасних відключень амплітудного детектора, до виходу кожного амплітудного детектора застосовується поріг синхронізації. Як правило, необхідна певна кількість послідовних порогових відключень амплітуди за межами допуску, щоб оголосити несправний сигнал. Таким чином, детектор порогового значення тривалості часу починає під-

Because transient effects may produce short-term amplitude detector trips, a timing threshold is applied to the output of each amplitude detector. Typically, a given number of consecutive out-of-tolerance amplitude threshold trips are necessary to declare a faulty signal. Hence, a time duration threshold detector begins a count whenever a set signal is received, and in the absence of further inputs, increments the count for

	рахунок щоразу, коли надходить встановлений сигнал, і за відсутності подальших вхідних даних збільшує відлік для кожного інтервалу вибірки після цього. Якщо задана кількість циклів перевищена, оголошується помилковий стан і для уражених каналів встановлюється логічний сигнал несправності. В іншому випадку підрахунок повертається до нуля, коли надходить сигнал скидання.	each sample interval thereafter. If a given cycle count is exceeded, a erroneous state is declared and a fault logic signal is set for the affected channels. Otherwise, the count is returned to zero when a reset signal is received.
	С. 157–158.	**С. 507.**
	Встановлення порогів часу та амплітуди має вирішальне значення через компроміс між логічними відключеннями неприємної несправності та повільною реакцією на фактичні несправності. Неприємні відключення підривають довіру користувачів до системи, їх невиправдані відключення потенційно можуть призвести до виснаження ресурсів. З іншого боку, запізніла реакція на помилку може призвести до виникнення небезпечних умов або катастрофічної події. Дозволений час для відновлення після даного типу несправності, який залежить від програми, є ключем до правильного встановлення порогових значень. Ступінь синхронізації каналів і спотворення даних також впливають на порогові налаштування, оскільки вони повинні враховувати будь-яку похибку. Компроміс може стати досить проблематичним, якщо потрібне швидке відновлення несправності. Оскільки цілісність і функціональність системи поставлені під загрозу, детальний проект компаратора виборців має бути предметом ретельної оцінки на всіх етапах розробки. У випадку з апаратно реалізованим пристроєм необхідно ретельно перевірити його аспекти виявлення несправностей. Пасивні збої в схемах, які зазвичай не використовуються, викликають основне занепокоєння. Вбудований тест, самоконтроль або симптоми збоїв є звичайними підходами до цілісності пристрою. У випадку компараторів виборців, реалізованих програмним забезпеченням, їхню надійність можна посилити за допомогою формальних методів перевірки та коду, перевіреного під час експлуатації.	The setting of both the timing and amplitude thresholds is of crucial importance because of the tradeoff between nuisance fault logic trips and slow response to actual faults. Nuisance trips erode user confidence in a system, their unwarranted trips can potentially cause resource depletion. On the other hand, a belated fault response may permit an unsafe condition or catastrophic event to occur. The allowable time to recover from a given type of fault, which is application-dependent, is the key to setting the thresholds properly. The degree of channel synchronization and data skewing also affect the threshold settings, because they must accommodate any looseness. The trade-off can become quite problematic where fast fault recovery is required. Because the integrity and functionality of the system is at stake, the detailed design of a voter comparator must be subject to careful assessment at all stages of development. In the case of a hardware-implemented device, its fault detection aspects must be thoroughly examined. Passive failures in circuitry that is not normally used are the main concern. Built-in test, self-monitoring, or fail-hard symptoms are customary approaches to device integrity. In the case of software-implemented voter comparators, their dependability can be reenforced through formal proof methods and in-service verified code.
	С. 158–159.	**С. 507.**
	Сторожові таймери можна використовувати, щоб уловлювати як апаратне, так і програмне забезпечення, що переходять у небажані стани. Перевірки часу є формою перевірки тверджень. Цей вид перевірки корисний, оскільки багато помилок програмного та апаратного забезпечення проявляються у надмірному часу, який витрачається на певну операцію. У синхронних архітектурах потоку даних дані мають надходити в певний час. Помилки передачі даних такого типу можна виявити за допомогою таймера. Програмне забезпечення відіграє вирішальну роль у цифрових системах. Термін «програмно реалізована відмовостійкість», який використовується в цьому розділі, використовується в ширшому сенсі, вказуючи на роль програмного забезпечення в реалізації відмовостійкості. **Коваленко переписала чужий текст, видаливши заголовки та покликання.** **Плагіат.**	**28.3.2 Watchdog Timers** Watchdog timers can be used to catch both hardware and software wandering into undesirable states [Lala and Harper, 1994]. Timing checks are a form of assertion checking. This kind of check is useful because many software and hardware errors are manifested in excessive time taken for some operation. In synchronous data flow architectures, data are to arrive at a specific time. Data transmission errors of this type can be detected using a timer. **28.4 Software-Implemented Fault Tolerance—State Consistency** Software performs a critical role in digital systems. The term ‘‘software implemented fault tolerance’’ as used in this chapter is used in the broader sense indicating the role software plays in the implementation of fault tolerance, and not as a reference to the SRI International project performed for NASA in the late 1970s and referred to as SIFT.

	С. 159.	С. 507.
	Програмне забезпечення відіграє важливу роль у виявленні помилок. Виявлення помилок на системному рівні має базуватися на специфікації поведінки системи. Виходи системи слід перевіряти, щоб переконатися, що виходи відповідають специфікаціям. Ці перевірки мають бути незалежними від системи. Оскільки вони реалізовані в програмному забезпеченні, перевірки вимагають доступу до інформації, що перевіряється, і тому можуть потенційно зіпсувати цю інформацію. Отже, незалежність між системою та її перевіркою не може бути абсолютною. Надання ідеальних перевірок для виявлення помилок рідко є практичним, і більшість систем використовують перевірки на прийнятність . **Коваленко переписала чужий текст, видаливши заголовок і покликання.** **Коваленко пише «Отже», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Плагіат.**	**28.4.1 Error Detection** Software plays a major role in error detection. Error detection at the system level should be based on the specification of system behavior. The outputs of the system should be checked to assure that the outputs conform to the specification. These checks should be independent of the system. Since they are implemented in software, the checks require access to the information to be checked, and therefore may have the potential of corrupting that information. Hence, the independence between a system and its check cannot be absolute. The provision of ideal checks for error detection is rarely practical, and most systems employ checks for acceptability [Anderson and Lee, 1981].
	С. 159.	**С. 507–508.**
	Вирішити, де використовувати виявлення системних помилок, непросто. Перевірки на ранніх стадіях не слід розглядати як заміну перевіркам в останній момент. Перевірки на ранніх стадіях обов'язково базуватиметься на знанні внутрішньої роботи системи і, отже, не буде незалежною від системи. Перевірки на ранніх стадіях може виявити помилку на самих ранніх стадіях і таким чином мінімізувати поширення пошкоджень. Перевірка в останній момент гарантує, що жоден вихід системи не залишиться неперевіреним. Тому в системі повинні бути передбачені як перевірки в останній момент, так і перевірки на ранніх стадіях. Щоб виявити помилки програмного забезпечення, необхідно, щоб резервні версії програмного забезпечення були незалежними одна від одної, тобто мали різний проект. **Коваленко пише «Отже», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	Deciding where to employ system error detection is not a straightforward matter. Early checks should not be regarded as substitute for last-moment checks. An early check will of necessity be based on a knowledge of the internal workings of the system and hence will lack independence from the system. An early check could detect an error at the earliest possible stages and hence minimize the spread of damage. A last moment check ensures that none of the output of the system remains unchecked. Therefore, both last-moment and early checks should be provided in a system [Anderson and Lee, 1981]. In order to detect software faults, it is necessary that the redundant versions of the software be independent of each other, that is, of diverse design [Avizenis and Kelley, 1982] (See Section 28.5).
	С. 159–160.	**С. 508.**
	Якщо очікуються помилки проекту, необхідно забезпечити реплікацію з використанням версій системи з різними проектами. Перевірки реплікації порівнюють два набори результатів, отриманих як вихідні дані репліковних модулів. Перевірка реплікації піднімає позначку помилки та ініціює запуск інших процесів, щоб визначити, який компонент або канал є помилковим. Перевірки часу використовуються для виявлення наявності несправностей у системі, але не їх відсутності. У синхронних системах жорсткого реального часу повідомлення, що містять дані, пере-	**28.4.1.1 Replication Checks** If design faults are expected, replication must be provided using versions of the system with different designs. Replication checks compare the two sets of results produced as outputs of the replicated modules. The replication check raises an error flag and intiates the start of other processes to determine which component or channel is in error [Anderson and Lee, 1981]. **28.4.1.2 Timing Checks** Timing checks are used to reveal the presence of

	даються через шини даних за певним розкладом. Неотримання повідомлення в запланований час є помилкою. Помилка може бути спричинена несправністю датчика, шини даних тощо. У цьому випадку, якщо дані були критичними, методом допущення несправності може бути використання прямої екстраполяції стану. **Коваленко переписала чужий текст, видаливши заголовки та покликання.** **Плагіат.**	faults in a system, but not their absence [Anderson and Lee, 1981] In synchronous hard real-time systems, messages containing data are transmitted over data buses at a specific schedule. Failure to receive a message at the scheduled time is an error. The error could be caused by faults in a sensor, data bus, etc. In this case, if the data were critical, a method of tolerating the fault may be to use a forward state extrapolation.
	С. 160.	**С. 508.**
	Скасована перевірка бере виходи з системи та обчислює, якими мали бути вхідні дані, щоб отримати цей вихід. Потім розраховані вхідні дані можна порівняти з фактичними вхідними даними, щоб перевірити, чи є помилка. Системи, що забезпечують математичні функції, часто піддаються сторнованій перевірці. Аналітична надлишковість з використанням будь-якого з двох загальних методів виявлення помилок, множинної моделі (ММ) або узагальненого відношення правдоподібності, є формою зворотної перевірки. Обидва методи використовують модель системи, представлену фільтрами Калмана. ММ намагається обчислити міру того, наскільки добре відстежує кожен із фільтрів Калмана, дивлячись на помилки передбачення. Реальні системи мають нелінійність, і модель передбачає лінійну систему. Питання полягає в тому, чи відповідає помилка відстеження від розширеного фільтра Калмана лінеаризованій моделі, «найближчій до» справжньої нелінійної системи, і чи є вона помітно меншою, ніж помилки від фільтрів, заснованих на «більш віддалених» моделях. Відмови приводів і датчиків можна моделювати різними способами за допомогою цієї методології. **Коваленко переписала чужий текст, видаливши заголовок і покликання.** **Плагіат.**	**28.4.1.3 Reversal Check (Analytical Redundancy)** A reversal check takes the outputs from a system and calculates what the inputs should have been to produce that output. The calculated inputs can then be compared with the actual inputs to check whether there is an error. Systems providing mathematical functions often lend themselves to reversal checks [Anderson and Lee, 1981]. Analytic redundancy using either of two general error detection methods, multiple model (MM) or generalized likelihood ratio (GLR), is a form of reversal check. Both methods make use of a model of the system represented by Kalman filters. The MM attempts to calculate a measure of how well each of the Kalman filters is tracking by looking at the prediction errors. Real systems possess nonlinearity and the model assumes a linear system. The issue is whether the tracking error from the extended Kalman filter corresponds to the linearized model “closest to” the true, nonlinear system and is markedly smaller than the errors from the filters based on “more distant” models. Actuator and sensor failures can be modeled in different ways using this methodology [Willsky, 1980].
	С. 160–161.	**С. 508.**
	Узагальнене співвідношення правдоподібності (GLR) використовує формулювання, подібне до формулювання для ММ, але настільки інше, що структура рішення є зовсім іншою. Відправною точкою GLR є модель, що описує нормальну роботу спостережуваних сигналів або системи, з якої вони надходять. Оскільки GLR безпосередньо спрямований на виявлення різких змін, його застосування обмежене проблемами, пов’язаними з такими змінами, такими як виявлення збоїв. GLR, на відміну від ММ, вимагає одного фільтра Калмана. Будь-який виявлений збій демонструватиме систематичне відхилення між тим, що спостерігається, і тим, що передбачається спостерігати. Якщо ефект параметричного збою «достатньо близький» до ефекту адитивного, система працюватиме.	The Generalized Likelihood Ratio (GLR) uses a formulation similar to that for MM, but different enough that the structure of the solution is quite different. The starting point for GLR is a model describing normal operation of the observed signals or of the system from which they come. Since GLR is directly aimed at detecting abrupt changes, its applications are restricted to problems involving such changes, such as failure detection. GLR, in contrast to MM, requires a single Kalman filter. Any detectable failure will exhibit a systematic deviation between what is observed and what is predicted to be observed. If the effect of the parametric failure is “close enough” to that of the additive one, the system will work.
	С. 161.	**С. 508.**
	В основі обох методів GLR і ММ лежить проблема використання надмірності системи для генерації сигналів порівняння, які можна використову-	Underlying both the GLR and MM methods is the issue of using system redundancy to generate comparison signals that can be used for the detection of

	вати для виявлення збоїв. Фундаментальна ідея пошуку сигналів порівняння полягає у використанні надмірності системи, тобто відомих взаємозв'язків між виміряними змінними, щоб генерувати сигнали, які є малими за нормальної роботи та які відображають передбачувані моделі, коли розвиваються певні аномалії. Усі виявлення несправностей базуються на аналітичних взаємозв'язках між вимірюваними змінними, включаючи методи голосування, які припускають, що датчики вимірюють ту саму змінну. Компроміс із використанням аналітичних зв'язків полягає в тому, що ми можемо зменшити надмірність апаратного забезпечення та підтримувати той самий рівень працездатності від збоїв. Крім того, аналітична надлишковість дозволяє отримувати більше інформації з даних, дозволяючи виявити тонкі зміни в характеристиках компонентів системи. З іншого боку, використання цієї інформації може спричинити проблеми, якщо існують великі невизначеності в параметрах, що визначають аналітичні зв'язки. **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	failures. The fundamental idea involved in finding comparison signals is to use system redundancy, i.e., known relationships among measured variables to generate signals that are small under normal operation and which display predictable patterns when particular anomalies develop. All failure detection is based on analytical relationships between sensed variables, including voting methods, which assume that sensors measure precisely the same variable. The trade-off using analytical relationships is that we can reduce hardware redundancy and maintain the same level of fail-operability. In addition, analytical redundancy allows extracting more information from the data, permitting detection of subtle changes in system component characteristics. On the other hand, the use of this information can cause problems if there are large uncertainties in the parameters specifying the analytical relationships [Willsky, 1980].
	С. 161–162.	**С. 509.**
	Другою частиною алгоритму виявлення відмов є правило прийняття рішень, яке використовує доступні сигнали порівняння для прийняття рішень щодо переривання нормальної роботи шляхом оголошення відмов. Однією з переваг цих методів є те, що правило прийняття рішень — максимізація та порівняння з порогом — є простим, тоді як основним недоліком є те, що правило явно не відображає бажаних компромісів. Підхід байєсівського послідовного рішення, в якому алгоритм для розрахунку приблизного рішення Байєса, має прямо протилежні властивості, тобто дозволяє пряме включення компромісів продуктивності, але є надзвичайно складним. Проблема послідовного прийняття рішень Байєса полягає у виборі правила зупинки та правила кінцевого рішення, щоб мінімізувати загальну очікувану вартість та очікувану вартість, яка накопичується до зупинки. **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	The second part of a failure detection algorithm is the decision rule that uses the available comparison signals to make decisions on the interruption of normal operation by the declaration of failures. One advantage of these methods is that the decision rule — maximize and compare to a threshold — are simple, while the main disadvantage is that the rule does not explicitly reflect the desired trade-offs. The Bayesian Sequential Decision approach, in which an algorithm for the calculation of the approximate Bayes decision, has exactly the opposite properties, i.e., it allows for a direct incorporation of performance trade-offs, but is extremely complex. The Bayes Sequential Decision Problem is to choose a stopping rule and terminal decision rule to minimize the total expected cost, and the expected cost that is accrued before stopping [Willsky, 1980].
	С. 162.	**С. 509.**
	Перевірки кодування базуються на надмірності у представленні об'єкта, що використовується в системі. Всередині об'єкта надлишкові дані підтримуються в певному фіксованому зв'язку з (не надлишковими) даними, що представляють значення об'єкта. Перевірка на парність є добре відомим прикладом перевірки кодування, як і коди виявлення та виправлення помилок, такі як код Хеммінга, циклічна перевірка надмірності та арифметичні коди. Ці перевірки базуються на знанні мінімальних і максимальних значень вхідних даних, а також обмежень швидкості зміни вхідних даних. Ці перевірки базуються на знанні фізичної роботи датчиків і використовують моделі цієї роботи.	**28.4.1.4 Coding Checks** Coding checks are based on redundancy in the representation of an object in use in a system. Within an object, redundant data are maintained in some fixed relationship with the (nonredundant) data representing the value of the object. Parity checks are a well-known example of a coding check, as are error detection and correction codes such as the Hamming, cyclic redundancy check, and arithmetic codes [Anderson and Lee, 1981]. **28.4.1.5 Reasonableness Checks** These checks are based on knowledge of the minimun and maximum values of input data, as well as the limits on rate of change of input data. These checks

	Коваленко переписала чужий текст, видаливши заголовки та покликання. **Плагіат.**	are based on knowledge of the physical operation of sensors, and employ models of this operation.
	С. 162.	**С. 509.**
	До структур даних в обчислювальній системі можна застосовувати дві форми перевірок. Перевірки семантичної цілісності даних стосуватимуться узгодженості інформації, що міститься в структурі даних. Перевірки структурної цілісності стосуватимуться того, чи є сама структура узгодженою. Наприклад, зовнішні дані з підсистем передаються з цифрових шин даних, таких як MIL-STD-1553, ARINC 429 або ARINC 629. Вміст повідомлення (кількість слів у повідомленні, вміст кожного слова) з підсистеми зберігається а вхідні дані перевіряються на відповідність. Діагностичні перевірки створюють вхідні дані для апаратних елементів системи, які мають видавати відомий вихід. Ці перевірки рідко використовуються як основний засіб виявлення помилок. Зазвичай вони запускаються під час запуску та можуть бути ініційовані оператором як частина вбудованого тесту. Вони також можуть працювати безперервно у фоновому режимі, коли процесор може простоювати. Також проводяться діагностичні перевірки, щоб виявити певні несправності. **Коваленко переписала чужий текст, видаливши заголовки.** **Плагіат.**	**28.4.1.6 Structural Checks** Two forms of checks can be applied to the data structures in a computing system. Checks on the semantic integrity of the data will be concerned with the consistency of the information contained in a data structure. Checks on the structural integrity will be concerned with whether the structure itself is consistent. For example, external data from subsystems is transmitted from digital data buses such as MIL-STD-1553, ARINC 429 or ARINC 629. The contents of a message (number of words in the message, contents of each word) from a subsystem are stored and the incoming data checked for consistency. **28.4.1.7 Diagnostic Checks** Diagnostic checks create inputs to the hardware elements of a system, which should produce a known output. These checks are rarely used as the primary error detection measure. They are normally run at startup, and may be initiated by an operator as part of a built-in test. They may also run continously in a background mode when the processor might be idle. Diagnostic checks are also run to isolate certain faults.
	С. 162–163.	**С. 509.**
	При виявленні помилки, необхідно визначити ступінь пошкодження, завданої нею, перш ніж можна буде виконати відновлення помилки. Оцінка ступеня збитку зазвичай пов'язана зі структурою системи. За умови своєчасного виявлення помилок оцінка збитку зазвичай обмежується поточним обчисленням або процесом. Припускається, що стан є відповідним при вході. Перед виходом із поточного обчислення виконується перевірка виявлення помилок. Припускається, що будь-які виявлені помилки викликані помилками в поточному обчисленні. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	**28.4.2 Damage Confinement and Assessment** When an error has been discovered, it is necessary to determine the extent of the damage done by the fault before error recovery can be accomplished. Assessing the extent of the damage is usually related to the structure of the system. Assuming timely detection of errors, the assessment of damage is usually determined to be limited to the current computation or process. The state is assumed consistent on entry. An error detection test is performed before exiting the current computation. Any errors detected are assumed to be caused by faults in the current computation.
	С. 163.	**С. 509–510.**
	Після визначення масштабу пошкодження важливо відновити систему до нормального стану. Існує два основні підходи — зворотне та пряме відновлення помилок. При зворотному відновленні помилок система повертається до попереднього узгодженого стану. Потім поточне обчислення можна повторити з наявними компонентами, з альтернативними компонентами, або його можна проігно-	**28.4.3 Error Recovery** After the extent of the damage has been determined, it is important to restore the system to a consistent state. There are two primary approaches — backward and forward error recovery. In backward error recovery, the system is returned to a previous consistent state. The current computation can then be retried with existing components (retry)* with alternate components (reconfigure), or it can be ignored (skip

	рувати. Використання зворотного відновлення передбачає можливість збереження та відновлення стану. Зворотне відновлення помилок не залежить від оцінки збитку. Упереджене відновлення помилок намагається продовжити поточне обчислення, відновлюючи систему до узгодженого стану, компенсуючи невідповідності, виявлені в поточному стані. Попереднє відновлення помилок передбачає детальне знання масштабу завданої шкоди та стратегію усунення невідповідностей. Пряме відновлення помилок важче реалізувати, ніж зворотне відновлення помилок. **Коваленко переписала чужий текст, видаливши заголовок і покликання.** **Плагіат.**	frame).** The use of backward recovery implies the ability to save and restore the state. Backward error recovery is independent of damage assessment. Forward error recovery attempts to continue the current computation by restoring the system to a consistent state, compensating for the inconsistencies found in the current state. Forward error recovery implies detailed knowledge of the extent of the damage done, and a strategy for repairing the inconsistencies. Forward error recovery is more difficult to implement than backward error recovery [Hitt et al., 1984].
	С. 163–164.	**С. 510.**
	Після відновлення системи після помилки може бути бажаним ізолювати та/або виправити компонент, який спричинив помилку. Усунення несправностей не завжди є необхідним через тимчасовий характер деяких несправностей або через те, що процедур виявлення та відновлення достатньо для вирішення інших повторюваних помилок. Для постійних несправностей усунення несправностей стає важливим, оскільки маскування постійних несправностей знижує здатність системи справлятися з наступними несправностями. Деякі технології відмовостійкого програмного забезпечення намагаються виділити помилки в поточних обчисленнях шляхом своєчасного виявлення помилок. Після виділення несправності лікування несправності можна виконати шляхом переналаштування обчислень для використання альтернативних форм обчислень, щоб забезпечити безперервне обслуговування. (Це можна робити послідовно, як у блоках відновлення, або паралельно, як у програмуванні NVersion.) Припущення полягає в тому, що пошкодження через помилки належним чином інкапсульовано до поточного обчислення, а саме виявлення помилок є бездоганним (тобто виявляє всі помилок і не викликає власних причин). **Коваленко переписала чужий текст, видаливши заголовок і покликання.** **Плагіат.**	**28.4.4 Fault Treatment** Once the system has recovered from an error, it may be desirable to isolate and/or correct the component that caused the error. Fault treatment is not always necessary because of the transient nature of some faults or because the detection and recovery procedures are sufficient to cope with other recurring errors. For permanent faults, fault treatment becomes important because the masking of permanent faults reduces the ability of the system to deal with subsequent faults. Some fault-tolerant software techniques attempt to isolate faults to the current computation by timely error detection. Having isolated the fault, fault treatment can be done by reconfiguring the computation to use alternate forms of the computation to allow for continued service. (This can be done serially, as in recovery blocks, or in parallel, as in NVersion programming.) The assumption is that the damage due to faults is properly encapsulated to the current computation and that error detection itself is faultless (i.e., detects all errors and causes none of its own) [Hitt et al., 1984].
	С. 164.	**С. 510.**
	Багатопроцесорні архітектури, що складаються з обчислювальних ресурсів, пов'язаних між собою зовнішніми шинами даних, повинні проектуватися як розподілена відмовостійка система. Обчислювальні ресурси можуть бути встановлені в корпусі за допомогою паралельної об'єднавчої шини для реалізації багатопроцесорної обробки всередині корпусу. Кожен корпус можна вважати віртуальним вузлом у загальній мережі. Мережева операційна система в поєднанні з шинами даних і їх протоколом завершує відмовостійку розподілену систему. Архітектура може бути асинхронною, слабко синхронною або жорстко синхронною. Підтримувати узгодженість даних через резервні канали асин-	**28.4.5 Distributed Fault Tolerance** Multiprocessing architectures consisting of computing resources interconnected by external data buses should be designed as a distributed fault-tolerant system. The computing resources may be installed in an enclosure using a parallel backplane bus to implement multiprocessing within the enclosure. Each enclosure can be considered a virtual node in the overall network. A network operating system, coupled with the data buses and their protocol, completes the fault-tolerant distributed system. The architecture can be asynchronous, loosely synchronous, or tightly synchronous. Maintaining consistency of data across redundant channels of asynchronous systems is difficult [Papadopoulos, 1985].

	хронних систем важко. **Коваленко переписала чужий текст, видаливши заголовок і покликання.** **Плагіат.**	
	С. 164.	**С. 510.**
	Помилки програмного забезпечення, які вважаються помилками проектування, можуть виникати під час розробки вимог, створення специфікацій, проектування архітектури програмного забезпечення, створення коду та інтеграції коду. Хоча під час системної інтеграції та тестування можна знайти й усунути багато несправностей, практично неможливо усунути всі можливі помилки розробки програмного забезпечення. Отже, використовується відмовостійкість програмного забезпечення. Двома основними методами, які використовуються для забезпечення відмовостійкості програмного забезпечення, є програмне забезпечення N-версії та блоки відновлення. **Коваленко переписала чужий текст, видаливши заголовок.** **Коваленко пише «Отже», ніби це вона робить якийсь висновок, а насправді переписує чужий текст.** **Плагіат.**	**28.5 Software Fault Tolerance** Software faults, considered design faults, may be created during the requirements development, specification creation, software architecture design, code creation, and code integration. While many faults may be found and removed during system integration and testing, it is virtually impossible to eliminate all possible software design faults. Consequently, software fault tolerance is used. Table 28.5 lists the major fault-tolerant software techniques in use today. The two main methods that have been used to provide software fault tolerance are N-version software and recovery blocks.
	С. 164–165.	**С. 510–511.**
	Багатоверсійне програмне забезпечення — це будь-яка відмовостійка програмна техніка, у якій реалізовано, виконано дві або більше альтернативних версій, а результати порівнюються за допомогою певної форми алгоритму прийняття рішень. Мета полягає в тому, щоб розробити ці альтернативні версії таким чином, щоб помилки програмного забезпечення, які можуть існувати в одній версії, були відсутні. Не повна відмовостійка техніка програмного забезпечення, виявляє лише помилки міститься в іншій версії, і алгоритм прийняття рішень визначає правильне значення серед альтернативних версій. Незалежно від того, які засоби використовуються для створення альтернативних версій, спільною метою є наявність різних версій програмного забезпечення, щоб імовірність одночасного виникнення помилок була невеликою і щоб помилки можна було розрізнити, коли результати виконання мультиверсій порівнюються один з одним. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	**28.5.1 Multiversion Software** Multiversion software is any fault-tolerant software technique in which two or more alternate versions are implemented, executed, and the results compared using some form of a decision algorithm. The goal is to develop these alternate versions such that software faults that may exist in one version are not contained in the other version(s) and the decision algorithm determines the correct value from among the alternate versions. Whatever means are used to produce the alternate versions, the common goal is to have distinct versions of software such that the probability of faults occuring simultaneously is small and that faults are distinguishable when the results of executing the multiversions are compared with each other.
	С. 165.	**С. 511.**
	Функція порівняння виконується як алгоритм прийняття рішень після отримання результатів від кожної версії. Алгоритм прийняття рішень вибирає відповідь або сигналізує, що він не може визначити відповідь. Цей алгоритм прийняття рішень і розробка альтернативних версій є основним методом виявлення помилок. Оцінка збитку передбачає, що	The comparison function executes as a decision algorithm once it has received results from each version. The decision algorithm selects an answer or signals that it cannot determine an answer. This decision algorithm and the development of the alternate versions constitute the primary error detection method. Damage assessment assumes the damage is limited

	збиток обмежується інкапсуляцією окремих версій програмного забезпечення. Несправні компоненти програмного забезпечення маскуються так, що несправності обмежуються модулем, у якому вони виникають. Відновлення несправного компонента може бути зроблено або не зроблено. N-версій програми незалежно створюють N-групи інженерів програмного забезпечення, які працюють на основі загальної специфікації. Кожна версія виконується незалежно від інших версій. Кожна версія повинна мати доступ до ідентичного набору вхідних значень, а результати порівнюються виконавцем, який вибирає використаний результат. Вибір точного чи неточного алгоритму перевірки голосування залежить від критичності функції та часу, пов'язаного з голосуванням.	to the encapsulation of the individual software versions. Faulted software components are masked so that faults are confined within the module in which they occur. Fault recovery of the faulted component may or may not be attempted. N-versions of a program are independently created by N-software engineering teams working from a (usually) common specification. Each version executes independently of the other versions. Each version must have access to an identical set of input values and the outputs are compared by an executive which selects the result used. The choice of an exact or inexact voting check algorithm is influenced by the criticality of the function and the timing associated with the voting.
	С. 165–166.	**С. 511.**
	Друга основна техніка, наведена в таблиці — це блок відновлення та його підкатегорії — механізм кінцевих термінів і різне програмне забезпечення резервного копіювання. Техніка блоку відновлення розпізнає ймовірність того, що в програмному забезпеченні можуть існувати залишкові помилки. Замість того, щоб розробляти незалежні надлишкові модулі, ця методика покладається на використання програмного модуля, який виконує приймальний тест на вихідних даних основного модуля. Приймальний тест викликає виключення, якщо стан системи неприйнятний. Наступним кроком є оцінка збитку та усунення несправності. Враховуючи, що помилка конструкції в основному модулі могла спричинити довільну шкоду стану системи, і що точний час, коли були згенеровані помилки, неможливо визначити, найбільш підходящим попереднім станом для відновлення є стан, який існував безпосередньо перед основним модулем було введено. **Коваленко переписала чужий текст, видаливши заголовок і покликання.** **Плагіат.**	**28.5.2 Recovery Blocks** The second major technique shown in Table 28.5 is the recovery block and its subcategories — deadline mechanism and dissimilar backup software. The recovery block technique recognizes the probability that residual faults may exist in software. Rather than develop independent redundant modules, this technique relies on the use of a software module which executes an acceptance test on the output of a primary module. The acceptance test raises an exception if the state of the system is not acceptable. The next step is to assess the damage and recover from the fault. Given that a design fault in the primary module could have caused arbitrary damage to the system state, and that the exact time at which errors were generated cannot be identified, the most suitable prior state for restoration is the state that existed just before the primary module was entered [Anderson and Lee, 1981].
	С. 166.	**С. 511–512.**
	Охоплення великої кількості несправностей має пов'язані накладні витрати з точки зору надмірності та обробки, пов'язаної з виявленням помилок. Розробник може використовувати моделювання та симуляцію, щоб визначити кількість резервування, необхідного для реалізації відмовостійкості, порівняно з ймовірністю та впливом різних типів несправностей. Якщо помилка має мінімальний вплив або взагалі не впливає на безпеку чи виконання місії, інвестиції в резервування для усунення цієї помилки можуть бути неефективними, навіть якщо ймовірність виникнення помилки є значною. Відмовостійкі системи повинні використовуватися щоразу, коли збій може призвести до втрати життя або втрати цінного активу. Фізичні збої апаратного забезпечення зменшуються, тоді як збої в дизайні практично неможливо повністю усунути. **Коваленко переписала чужий текст, видаливши**	**28.5.3 Trade-Offs** Coverage of a large number of faults has an associated overhead in terms of redundancy, and the processing associated with the error detection. The designer may use modeling and simulation to determine the amount of redundancy required to implement the fault tolerance vs. the probability and impact of the different types of faults. If a fault has minimal or no impact on safety, or mission completion, investing in redundancy to handle that fault may not be effective, even if the probability of the fault occuring is significant. **28.6 Summary** Fault-tolerant systems must be used whenever a failure can result in loss of life, or loss of a high-value asset. Physical failures of hardware are decreasing whereas design faults are virtually impossible to completely eliminate.

	заголовки. **Плагіат.**	
	С. .166–167	**С. 512.**
	При застосуванні відмовостійкості до складної системи існує небезпека того, що нові механізми можуть створити додаткові джерела збоїв через помилки проектування та реалізації. Тому важливо, щоб нові механізми були впроваджені таким чином, щоб зберегти цілісність конструкції з мінімальною додатковою складністю. Розробник повинен використовувати засоби моделювання та симуляції, щоб переконатися, що проект забезпечує необхідну відмовостійкість. Для спрощення процесу прийняття проектних рішень розроблено певні принципи проектування. Інкапсуляція та ієрархія пропонують способи досягнення простоти та загальності в реалізації окремих функцій відмовостійкості. Інкапсуляція забезпечує: – організація даних і програм як єдиних об'єктів із суворим контролем взаємодії об'єктів. – Організація наборів альтернативних версій програм у відмовостійкі програмні модулі (наприклад, блоки відновлення та набори програм N-версій). – Організація узгоджених наборів точок відновлення для кількох процесів. – Організація зв'язків між розподіленими процесами як атомарних (неподільних) дій. – Організація функцій операційної системи в відновлювані модулі. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	**28.6.1 Design Analyses** In applying fault-tolerance to a complex system, there is a danger that the new mechanisms may introduce additional sources of failure due to design and implementation errors. It is important, therefore, that the new mechanisms be introduced in a way that preserves the integrity of a design, with minimum added complexity. The designer must use modeling and simulation tools to assure that the design accomplishes the needed fault tolerance. Certain design principles have been developed to simplify the process of making design decisions. Encapsulation and hierarchy offer ways to achieve simplicity and generality in the realization of particular fault-tolerance functions. Encapsulation provides: • Organization of data and programs as uniform objects, with rigorous control of object interaction. • Organization of sets of alternate program versions into fault-tolerant program modules (e.g. recovery blocks and N-version program sets). • Organization of consistent sets of recovery points for multiple processes. • Organization of communications among distributed processes as atomic (indivisible) actions. • Organization of operating system functions into recoverable modules.
	С. 167–168.	**С. 512.**
	– Приклади принципу ієрархії, який використовується для підвищення надійності функцій відмовостійкості, включають: – Організація всього програмного забезпечення, як прикладного, так і системного типу, на рівні, з односпрямованими залежностями між шарами. – Інтеграція сервісних функцій і функцій відмовостійкості на кожному рівні. – Використання вкладених блоків відновлення для забезпечення можливості ієрархічного відновлення. – Організація функцій операційної системи таким чином, що лише мінімальний набір на найнижчому рівні («ядро») повинен бути звільнений від відмовостійкості. – Інтеграція глобальних і локальних даних і управління в розподілених процесорах. Тій частині ядра операційної системи, яка забезпечує базові механізми, які решта системи використовує для досягнення відмовостійкості, слід «довіряти». Це ядро має бути обмеженої складності, щоб можна було перевірити всі можливі шляхи для забезпечення правильної роботи за будь-якої логіки та умов даних. Це ядро не повинно бути відмовостійким, якщо можна гарантувати вищезазначене. **Коваленко переписує чужий текст, не розуміючи його. Вступне речення перед переліком пе-**	Examples of the hierarchy principle used to enhance reliability of fault-tolerance functions include: • Organization of all software, both application and system type, into layers, with unidirectional dependencies among layers. • Integration of service functions and fault-tolerance functions at each level. • Use of nested recovery blocks to provide hierarchical recovery capability. • Organization of operating system functions so that only a minimal set at the lowest level (a “kernel”) needs be exempted from fault tolerance. • Integration of global and local data and control in distributed processors. That portion of the operating system kernel that provides the basic mechanisms the rest of the system uses to achieve fault-tolerance should be “trusted.” This kernel should be of limited complexity so that all possible paths can be tested to assure correct operation under all logic and data conditions. This kernel need not be fault tolerant if the foregoing can be assured.

	ретворюється в неї в один із пунктів цього переліку: «– Приклади принципу ієрархії**».** **Недолугий плагіат.**	
	С. 168.	**С. 512.**
	Безпека визначається з точки зору небезпек і ризиків. Небезпека – це стан або сукупність умов, які можуть призвести до нещасного випадку за відповідних обставин. Рівень ризику, пов'язаного з небезпекою, залежить від імовірності того, що небезпека станеться, ймовірності нещасного випадку, якщо небезпека все ж відбудеться, і потенційних наслідків аварії. **Коваленко переписала чужий текст, видаливши заголовок і покликання.** **Плагіат.**	**28.6.2 Safety** Safety is defined in terms of hazards and risks. A hazard is a condition, or set of conditions that can produce an accident under the right circumstances. The level of risk associated with the hazard depends on the probability that the hazard will occur, the probability of an accident taking place if the hazard does occur, and the potential consequence of the accident [Williams, 1992].
	С. 168–169.	**С. 512–513.**
	Валідація — це процес, за допомогою якого системи демонструють роботу, яка відповідає специфікаціям. Процес перевірки починається, коли специфікація завершена. Визнали складність розробки точної специфікації, яка ніколи не зміниться. Ця реальність призвела до ітеративного процесу розробки та перевірки. Перевірка вимагає розробки тестових випадків і виконання цих тестових випадків на апаратному та програмному забезпеченні, що входить до складу системи, яка буде доставлена. Випробування повинні охоплювати 100% помилок, які система розроблена для терпимості, і дуже високий відсоток можливих проектних помилок, будь то апаратне забезпечення, програмне забезпечення або взаємодія апаратного та програмного забезпечення під час виконання всіх можливих даних і логічних шляхів. Після перевірки системи її можна вводити в експлуатацію. Щоб мінімізувати потребу перевіряти повну оперативну польотну програму кожного разу, коли вона змінюється, методи розробки намагаються розкласти велику систему на модулі, які незалежно визначаються, реалізуються та перевіряються. Лише ті модулі та їхні інтерфейси, які були змінені, повинні бути повторно перевірені за допомогою цього підходу. Швидке створення прототипів, моделювання та анімація – це всі методи, які допомагають перевірити систему. Формальні методи використовуються для розробки та перевірки цифрових систем авіоніки. Існують аргументи як на користь, так і проти використання формальних методів. **Коваленко переписала чужий текст, видаливши заголовок і покликання.** **Плагіат.**	**28.6.3 Validation** Validation is the process by which systems are shown through operation to satisfy the specifications. The validation process begins when the specification is complete. The difficulty of developing a precise specification that will never change has been recognized. This reality has resulted in an iterative development and validation process. Validation requires developing test cases and executing these test cases on the hardware and software comprising the system to be delivered. The tests must cover 100% of the faults the system is designed to tolerate, and a very high percentage of possible design faults, whether hardware, software, or the interaction of the hardware and software during execution of all possible data and logical paths. Once the system has been validated, it can be put in operation. In order to minimize the need to validate a complete Operational Flight Program (OFP) every time it is modified, development methods attempt to decompose a large system into modules that are independently specified, implemented, and validated. Only those modules and their interfaces that are modified must be revalidated using this approach (see Chapter 29). Rapid prototyping, simulation, and animation are all techniques that help validate the system. Formal methods are being used to develop and validate digital avionics systems. There are arguments both in favor of and against the use of formal methods [Rushby, 1993; Williams, 1992].
9	**Коваленко Ю. Б.** **Інформаційна підтримка процесу проектування та експлуатації комплексу бортового обладнання інтегрованої модульної авіоніки.** – Дис. … доктора технічних наук. – Київ, 2025. (https://duikt.edu.ua/uploads/p_86_93515477.pdf)	**Albert B., Usach H., Vila J., Crespo A.** **Development of Integrated Modular Avionics Applications based on Simulink and XtratuM** // Conference: Data Systems In Aerospace (DASIA) At: Porto (Portugal), 2013. (https://www.researchgate.net/publication/295391894_Development_of_Integrated_Modular_Avionics_Applications_based_on_Simulink_and_XtratuM)

	С. 171.	**С. 1.**
	Simulink та Embedded Coder підтримують ключові етапи розробки на основі методу моделювання (Model-Based Design). Ця методологія проектування дозволяє легко моделювати складні системи (в тому числі нелінійні) і забезпечує автоматичну генерацію коду. Системи автоматичного кодування іноді вважаються ненадійними, оскільки вони схильні до надійними, оскільки вони схильні до створення "спагеті-коду". Однак, це не стосується Simulink, оскільки він має ряд супутніх інструментів для автоматизації перегляду вихідного коду згенерованого коду, для автоматизації раннього відстеження вимог для забезпечення перевірки відповідності стандартам моделювання та документування моделей. Перевагами цього процесу проектування на основі моделей є процесу на основі моделей є прискорення розробки тестових стендів та допомога досягненню цілей DO-178B, пов'язаних із завданнями верифікації завдяки використанню належних інструментів.	Simulink and Embedded Coder support key development activities based on Model-Based Design. This design methodology allows easy modeling of complex systems (including non-linear systems) and enables automatic coding generation. Automatic coding systems are sometimes regarded as non reliable since they are prone to produce “spaghetti code”. However, this is not the case of Simulink since it has a number of associated tools to automate source code reviews of generated code, to automate early requirements tracing, to provide modeling standards compliance checking, and to document models. The benefits of this Model-Based design process are accelerating testbench development and helping satisfying DO-178B objectives related to verification tasks through the use of proper tools.
	С. 171–172.	**С. 1.**
	Іншим ключовим елементом на етапі розробки програмного забезпечення є операційна система RTOS, на якій будуть виконуватися додатки для авіоніки. Авіоніка - це, як правило, розподілені додатки, що працюють в реальному часі і включають декілька обчислювальних модулів. RTOS повинна забезпечувати середовище виконання для виконання різних обчислювальних модулів програми, які повинні бути критично важливими для безпеки (як визначено авіаційними правилами), працювати в реальному часі та бути детермінованими [4]. Ці цілі можуть бути досягнуті за допомогою двох різних підходів: федеративної архітектури або інтегрованої архітектури. **Коваленко переписала чужий текст разом із номером покликання, під яким в її дисертації знаходиться зовсім інше джерело – ДСТУ 1995-го року.** **Фальсифікація джерел.** **Плагіат.**	The other key element in our software development phase is the RTOS on top of which avionics applications will be executed. Avionics applications are usually real-time distributed applications involving several computational modules. The RTOS must provide an execution environment for executing the different computational modules of an application which is required to be safety-critical (as defined by aviation regulations), real-time, and deterministic [4]. These goals can be achieved using two different approaches: federated architectures or integrated architectures.
	С. 172.	**С. 1.**
	Федеративні архітектури базуються на спеціальних обчислювальних модулях або процесорних блоках, розподілених по всьому транспортному засобу. Як правило, кожен модуль містить одну програму в системі авіоніки. Основна перевага такої архітектури полягає в тому, що вона забезпечує відмовостійкість: збої в одному модулі не впливають на інші модулі. В інтегрованих архітектурах кожен модуль містить безліч додатків різного рівня критичності. Така архітектура дозволяє зменшити кількість обчислювальних блоків, але вимагає надійного механізму розбиття на блоки та механізму моніторингу стану для ізоляції виконання різних модулів, як у федеративній архітектурі. У цьому підході обчислювальний блок розбивається на кілька віртуальних комп'ютерів, на кожному з яких розміщується додаток або модуль додатку . Інтегрована модульна авіоніка є найважливішим прикладом такої архітектури, яка стандартизована ARINC 653. Ця специфікація також стандартизує інтерфейс прикладного програмування (API). API, визначений ARINC 653, називається APEX – APplication Executive. Наш процес розробки програмного забезпе-	Federated Architectures are based on dedicated computational modules or processing units distributed across a vehicle. Generally, each module hosts one single application within the avionics system. The major advantage of this architecture is that it provides inherently failure independence: failures on one module do not affect other modules. In Integrated Architectures, each module hosts numerous applications of different criticality levels. This architecture allows to reduce the number of computing units but it requires a robust partitioning mechanism and a health monitoring mechanism to isolate the execution of the different modules as in a federated architecture. In this approach a computing unit is splitted into multiple virtual computers, each one hosting an application or an application module [5]. Integrated Modular Avionics (IMA) is the most important example of such architecture which is standardized by ARINC 653. This specification also standardizes the Application Programming Interface (API). The API defined by ARINC 653 is called APEX – Application EXecutive. Our software development process is based on XtratuM, a real-time hypervisor which provides the IMA model under differ-

	чення базується на XtratuM, гіпервізорі реального часу, який забезпечує модель IMA за допомогою різних API. Одним з них є Lithos, який сумісний з ARINC 653. У статті описується, як Simulink проекти можуть бути націлені на виконання на XtratuM. **Коваленко переписала чужий текст, видаливши покликання.** **Плагіат.**	ent API's. One of them is Lithos which is ARINC 653 compliant. The paper describes how Simulink designs can be targeted to execute on XtratuM.
	С. 172–173.	**С. 2.**
	XtratuM - це гіпервізор для вбудованих систем реального часу, який значною мірою базується на концепції IMA . Архітектура IMA – це, по суті, інтегрована архітектура, заснована на концепції розділення. Ця концепція виникла для захисту та розділення додатків з просторової та часової точок зору. XtratuM віртуалізує основні апаратні пристрої (пам'ять, таймери та переривання) для одночасного виконання декількох ОС. Однією з таких гостьових ОС є Lithos, яка реалізує концепцію процесу, представлену в стандарті ARINC 653, яка відсутня в XtratuM. Іншим способом реалізації концепції процесу є використання Partikle, що реалізує POSIX-потоки. Сервіси, які надає XtratuM через Lithos або Partikle, є тими, що визначені специфікацією ARINC 653, і коротко описані далі. **Коваленко переписала чужий текст, видаливши заголовок.** **Плагіат.**	**II. THE IMA CONCEPT AND XTRATUM** XtratuM is a hypervisor for real-time embedded systems that is based to a large extent on the IMA (Integrated Modular Avionics) concept. The IMA architecture is essentially an integrated architecture based on the partitioning concept. This concept emerges for protection and separation among applications from the spatial and temporal point of views. In the ESA project (IMA for Space), the IMA concept has been extended with space applications requirements to cover the space applications needs. XtratuM virtualises the essential hardware devices (memory, timers and interrupts) to execute concurrently several OSes. One of these guest OS is Lithos that implements the process concept presented in the ARINC 653 standard which is not present in XtratuM. Another way of implementing the concept process is using Partikle that implements POSIX threads. The services provided by XtratuM either through Lithos or Partikle are the ones defined by the ARINC 653 specification and are summarized next.
	С. 173.	**С. 2.**
	Керування розділами. Розділи – це середовища виконання з незалежними областями пам'яті та суворо захищеним часом виконання. Розділи планується відповідно до циклічного планувальника, який вказується у файлі конфігурації. Всі ресурси, що використовуються розділом (процеси, дошки, семафори, порти) повинні бути визначені під час конфігурування системи, створені та ініціалізовані під час фази ініціалізації розділу.	**Partition management.** Partitions are execution environments with independent memory spaces and strictly protected execution time. Partitions are scheduled according to a cyclic scheduler which is specified in the configuration file. All resources used by a partition (processes, blackboards, semaphores, ports, ...) have to be defined at system configuration time and created and initialised during the initialisation phase of the partition.