

Донбаська державна машинобудівна академія
Кафедра автоматизації виробничих процесів

МЕТОДИЧНІ ВКАЗІВКИ

ДО КУРСОВОГО ПРОЕКТУВАННЯ ПО ДИСЦИПЛІНІ

«ЦИФРОВІ СИСТЕМИ КЕРУВАННЯ І ОБРОБКИ ІНФОРМАЦІЇ»

(для студентів спеціальності 174

**«Автоматизація, комп'ютерно-інтегровані технології та
робототехніка» другого (магістерського) рівня освіти за ОПП
«Автоматизація та комп'ютерно-інтегровані технології»)**

Затверджено
на засіданні методичної ради
кафедри АВП
Протокол № 8 від 20.05.2024 р.

Краматорськ-Тернопіль,
ДДМА
2024

УДК 681.5 (075.8)
ББК 32.965
Р –17

Разживін О.В.

- Р-17 Методичні вказівки до виконання курсового проекту з дисципліни «Цифрові системи керування та обробки інформації» для студентів 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» другого (магістерського) рівня освіти за ОПІ «Автоматизація та комп'ютерно-інтегровані технології»/ О. В. Разживін, В. Є. Циганаш. – Краматорськ : ДДМА, 2024. – 80 с.

Приводяться структура й склад курсового проекту. Викладаються методики рішення найбільш важливих завдань проектування систем керування й обробки інформації – аналізу об'єкта керування, математичного опису й моделювання системи, а також рекомендації з розробки програмного забезпечення й створення документації до проекту.

У додатках представлені необхідні довідкові матеріали.

© О. В. Разживін, В. Є. Циганаш, 2024

ВСТУП

Курсовий проект по дисципліні «Цифрові системи керування й обробки інформації» виконується з метою придбання вмінь і навичок системного сприйняття завдань проектування та розкриття взаємозв'язку між концепцією керування й конструкторською концепцією в рамках конкретного завдання автоматизації виробництва.

Результати захисту курсового проекту визначають рівні теоретичної й практичної підготовки студента до виконання *науково-дослідних та інженерних завдань*, установлених кваліфікаційною характеристикою спеціальності стосовно до області проектування систем автоматизації.

Предметом проектування є один з наступних варіантів:

- *Перший варіант* – цифрова система керування обладнанням на технологічному рівні, що працює в **режимі реального часу**.
- *Другий варіант* – автоматизована виконавча система виробництва MES-рівня, що забезпечує оперативну обробку інформації в **режимі машинного часу**.

Зміст курсового проекту повинен повністю розкривати сутність розв'язку завдань, поставлених в індивідуальній темі проекту.

Як методологічний напрямок повинен бути прийнятий **системний підхід**, який дозволяє розкрити цілісність процесу в умовах різноманіття типів зв'язків, а також об'єднати цифрові й аналогові канали, механічні й електронні пристрої, матеріальні та інформаційні потоки, програмні й апаратні засоби.

Курсовий проект з дисципліни «Цифрові системи керування та обробки інформації» повинна сформулювати наступні програмні **результати навчання**, що передбачені освітньо-професійною програмою підготовки магістрів:

- Створювати системи автоматизації, кіберфізичні виробництва на основі використання інтелектуальних методів управління, баз даних та баз знань, цифрових та мережевих технологій, робототехнічних та інтелектуальних мехатронних пристроїв.

- Створювати високонадійні системи автоматизації з високим рівнем функціональної та інформаційної безпеки програмних та технічних засобів.

- Аналізувати виробничо-технічні системи у певній галузі діяльності як об'єкти автоматизації і визначати стратегію їх автоматизації та цифрової трансформації

- Застосовувати сучасні математичні методи, методи теорії автоматичного керування, теорії надійності та системного аналізу для дослідження та створення систем автоматизації складними технологічними та організаційно-технічними об'єктами, кіберфізичних виробництв.

- Розробляти функціональну, організаційну, технічну та інформаційну структури систем автоматизації складними технологічними та організаційно-технічними об'єктами, розробляти програмно-технічні керуючі комплекси із

застосуванням мережевих та інформаційних технологій, промислових контролерів, робототехнічних пристроїв, засобів людино-машинного інтерфейсу та з урахуванням технологічних умов та вимог до управління виробництвом.

- Розробляти і використовувати спеціалізоване програмне забезпечення та цифрові технології для створення систем автоматизації складними організаційно-технічними об'єктами, професійно володіти спеціальними програмними засобами.

- Дотримуватись норм академічної доброчесності, знати основні правові норми щодо захисту інтелектуальної власності, комерціалізації результатів науково-дослідної, винахідницької та проектної діяльності.

У результаті виконання курсової роботи з дисципліни «Цифрові системи керування та обробки інформації» студент повинен продемонструвати достатній рівень сформованості певних результатів навчання через здобуття наступних загальних та фахових компетентностей:

- Здатність проведення досліджень на відповідному рівні.

- Здатність генерувати нові ідеї (креативність).

- Здатність проектувати та впроваджувати високонадійні системи автоматизації та їх прикладне програмне забезпечення, для реалізації функцій управління та опрацювання інформації, здійснювати захист прав інтелектуальної власності на нові проектні та інженерні рішення

- Здатність здійснювати автоматизацію складних технологічних об'єктів та комплексів, створювати кіберфізичні системи на основі інтелектуальних методів управління та цифрових технологій з використанням баз даних, баз знань, методів штучного інтелекту, робототехнічних та інтелектуальних мехатронних пристроїв;

- Здатність проектувати та впроваджувати високонадійні системи автоматизації та їх прикладне програмне забезпечення, для реалізації функцій управління та опрацювання інформації, здійснювати захист прав інтелектуальної власності на нові проектні та інженерні рішення;

- Здатність аналізувати виробничо-технологічні системи і комплекси як об'єкти автоматизації, визначати способи та стратегії їх автоматизації та цифрової трансформації

- Здатність застосовувати спеціалізоване програмне забезпечення та цифрові технології для розв'язання складних задач і проблем автоматизації та комп'ютерно-інтегрованих технологій.

- Здатність розробляти функціональну, технічну та інформаційну структуру комп'ютерно-інтегрованих систем управління організаційно-технологічними комплексами із застосуванням мережевих та інформаційних технологій, програмно-технічних керуючих комплексів, промислових контролерів, мехатронних компонентів, робототехнічних пристроїв та засобів людино-машинного інтерфейсу.

Відповідно до цього студенти повинні виконати наступне:

1. На основі системного аналізу технологічного процесу визначити проблемні ситуації, які стримують вирішення завдань підвищення

ефективності процесу.

2. На основі огляду літературних джерел розробити пропозиції щодо удосконалення існуючої системи автоматизації.

3. Виконати моделювання системи керування або предметної області.

4. Для системи керування обладнанням розробити схеми з'єднань апаратних компонентів системи. Для автоматизованої виконавчої системи розробити модель програмного додатку та модель бази даних.

5. Розробити програмне забезпечення системи автоматизації.

6. Створити документацію проекту відповідно до вимог стандартів.

На всіх етапах проектування студенти повинні використовувати в якості інструментів проектування спеціальні програмні засоби, що забезпечують сучасні вимоги до якості проекту й скорочення часу на його розробку.

Тематика курсових проектів формується кафедрою й затверджується деканом факультету. При формуванні тематики враховуються теми дипломних робіт, визначені при проходженні практики. Зразкова тематика курсових проектів наведена в додатку А.

Завдання на курсове проектування розробляється керівником курсового проекту. У завданні визначаються тема проекту, вихідні дані для розробки проекту, зміст пояснювальної записки й графічної частини проекту, строк виконання. Зразок бланка завдання наведений у додатку Б.

Результати проектування повинні бути представлені в розрахунково-пояснювальній записці обсягом до 40 сторінок тексту на аркушах формату А4 і в графічній частині обсягом 4-5 аркушів формату А3 або А4.

Розрахунково-пояснювальна записка повинна бути оформлена відповідно до ГОСТ 2.105-85 «Общие требования к текстовым документам», а також ДСТУ 3008-95 «Документація. Звіти в сфері науки й техніки». Розрахунково-пояснювальна записка повинна містити титульний лист, завдання, реферат, зміст, основну частину, перелік посилань і додатки. Зразок титульного листа розрахунково-пояснювальної записки представлений у додатку В.

Рекомендується наступна структура й обсяг кожного елемента основної частини розрахунково-пояснювальної записки:

Для проекту цифрової системи керування обладнанням

Вступ (1 стор.).

1. Аналіз технологічного процесу й постановка завдань проектування (5 стор.).

2. Визначення основних параметрів системи (8-10 стор.).

3. Проектування системи (10-12 стор.).

4. Моделювання системи (5 стор.)

5. Розробка програмного забезпечення (5-8 стор.).

Висновки (1 стор.).

Графічна частина складається із схем та плакатів. Розрахунково-

пояснювальна записка повинна бути оформлена відповідно ДСТУ 3008-95 «Документація. Звіти у сфері науки і техніки ». Розрахунково-пояснювальна записка повинна містити титульний аркуш, завдання, реферат, зміст, основну частину, перелік посилань і додатки. Плакати використовуються для представлення результатів теоретичних та експериментальних досліджень. Плакат повинен мати назву (зверху), текстові та графічні елементи на плакаті не повинні викликати напруження у сприйманні.

Для проекту автоматизованої системи виробництва (розробка програмного додатка і бази даних)

Вступ (1 стор.).

1. Аналіз предметної області, літературний огляд й постановка завдань проектування (5 стор.).
2. Розробка моделі нового бізнес-процесу (IDEF0-модель, 5-7 стор.).
3. Моделювання програмного додатка – діаграма варіантів використання, діаграма класів, діаграма поведінки (UML-моделі, 10-15 стор.) .
4. Моделювання бази даних (ER-моделі, 6-8 стор.)
5. Розробка схеми розгортання системи (сервер, станція, мережа)

Висновки (1 стор.).

Графічна частина такого проекту складається із діаграм і плакатів. Діаграми являються програмними документами і тому оснащуються основним підписом як теоретичне креслення.

Графічні зображення на діаграмах повинні відповідати загально прийнятій нотації діаграм IDEF0, UML та ER.

1 АНАЛІЗ БАЗОВОГО ПРОЦЕСУ ТА ПОСТАНОВКА ЗАВДАНЬ ПРОЕКТУВАННЯ

Рекомендований перелік питань, які повинні бути розглянуті у першому розділі, наведено в таблиці 1.1.

Таблиця 1.1

Варіант предмету проектування	Перелік питань першого розділу
Система керування обладнанням технологічного рівня	1.1 Аналіз існуючого технологічного процесу і його недоліків 1.2 Аналіз сучасних підходів і технологічних рішень (літературний огляд) 1.3 Аналіз вимог до нової системи керування й розробка завдань проектування
Автоматизована виконавча система виробництва (інформаційна система)	1.1 Аналіз предметної області (існуючого процесу організації й керування виробництвом) 1.2 Аналіз сучасних підходів до створення інформаційних систем (літературний огляд) 1.3 Аналіз вимог до нової інформаційної системи й розробка завдань проектування

Пристаюючи до проектування автоматизованої системи, необхідно поставити **цілі проектування**.

Цілі проектування зазвичай перебувають у площині **економічних показників**, які, у свою чергу, визначаються однією із двох складових економічної ефективності – прямим ефектом або непрямим ефектом.

Для систем автоматичного керування обладнанням показниками ефективності є, головним чином, трудові й вартісні витрати, що ставляться до прямого ефекту. На величину цих витрат впливають наступні виробничі показники:

1. **Точність контролю, обліку й підтримки технологічних параметрів продукції.** Автоматизовані системи керування дозволяють зменшити трудові й вартісні витрати, збільшивши при цьому прибуток. Так, наприклад, якщо на прокатному стані підтримувати товщину смуги в межах нижнього поля допуску, то можна суттєво знизити витрати металу на погонний метр смуги. Економічно вигідне підвищення точності систем виміру й обліку споживання енергетичних ресурсів, тому що це сприяє зменшенню упущеної вигоди. Завдання підвищення точності повинні вирішуватися шляхом

ретельного аналізу всіх погрешностей і наступної розробки комплексних заходів.

2. *Витрата енергетичних ресурсів (електроенергія, газ, вода і т.д.), сировини й матеріалів на технологічні потреби.* Автоматизовані системи керування дозволяють застосувати найбільш ефективні алгоритми керування, які оптимізують споживання енергетичних ресурсів, сировини й матеріалів, що сприяє зменшенню їх витрат.

3. *Продуктивність процесу.* Продуктивність – це показник, який визначається як величина, зворотна сумі втрат часу на здійснення робочих рухів і виконання допоміжних операцій. Тому для підвищення продуктивності необхідно провести ретельний аналіз втрат часу й виключити або зменшити причини цих втрат.

4. *Коефіцієнт технічного використання.* Коефіцієнт технічного використання відбиває частку часу знаходження встаткування в працездатному стані щодо деякого часу експлуатації, наприклад, одного року. Він характеризує прості устаткування, пов'язані з його обслуговуванням і ремонтом. Системи автоматичного керування дозволяють підвищити *надійність* роботи устаткування й скоротити втрати часу на відновлення працездатності, що сприяє збільшенню дійсного фонду часу роботи устаткування, а також зменшенню матеріальних витрат, пов'язаних з обслуговуванням і ремонтом.

Ефективними засобами підвищення надійності є:

- вибір елементів з найменшою ймовірністю відмови;
- проектування засобів захисту від аварій;
- розробка розвинутої системи діагностики.

Для автоматизованої виконавчої системи виробництва (MES-системи) економічна ефективність визначається непрямим ефектом, інтегральним показником якого є *рентабельність*. Рентабельність відбиває ступінь ефективності використання матеріальних, трудових і грошових ресурсів виробництва. MES-системи поліпшують фінансові показники підприємства шляхом підвищення віддачі основних фондів, зменшення відсотка браку продукції, прискорення обігу коштів, забезпечення своєчасності поставок, розширення мережі клієнтів.

Передпроектний аналіз

Проектування системи автоматизації повинне базуватися на результатах системного аналізу технологічного процесу або предметної області програмного додатка. При виконанні цієї роботи потрібно визначити мету, об'єкт, предмет і зміст аналізу.

Метою аналізу технологічного процесу (предметної області) є конкретизація вимог до автоматизованої системи керування, а також

знаходження способів досягнення бажаного результату при мінімальних витратах.

Предметом аналізу повинен бути:

- **об'єкт автоматизації** (виробничий процес або предметна область організаційної діяльності);
- **засоби автоматизації** (апаратура керування, програмне забезпечення, канали передачі й т.п.).

Зміст аналізу – це перелік питань або завдань, які повинні бути вирішені. При аналізі технологічного процесу потрібно встановити недоліки існуючих засобів керування, а також визначити можливі способи їх усунення з урахуванням сучасних досягнень науково-технічного прогресу. При виконанні аналізу предметної області необхідно визначити недоліки в організації виконуваних робіт, процеси, не охоплені оперативним контролем, а також створити специфікацію функцій інформаційної системи.

Особливо уважно слід вивчити вимоги до технологічного процесу й особливості керування процесом.

Вимоги до технологічного процесу можна знайти в технічній документації – технічних умовах, галузевих нормативних документах і стандартах, в описах технологічних процесів, кресленнях, керівних матеріалах і т.п. Для обліку особливостей керування слід з'ясувати думки фахівців.

При розробці способів усунення виявлених недоліків не можна покладатися тільки на власні знання, а слід використати передовий досвід. Слід також враховувати, що в інвестиційному проекті найбільша увага приділяється його життєвому циклу – інвестори завжди вимагають надійного захисту своїх інвестицій на тривалий час (принцип «вкластися й не думати про модернізацію десять-п'ятнадцять років»). Тому розроблювач проекту повинен застосовувати в проекті перспективні рішення в організації апаратних і програмних засобів системи керування.

Саме ці обов'язки ставлять перед будь-яким проектувальником завдання аналізу наукової літератури й інтернет-джерел.

Підсумком виконання першого розділу є:

1. Специфікація вимог до проектованої системи (як повинна бути влаштована система і які функції вона повинна виконувати?);
2. Перелік завдань проектування (що потрібно зробити для забезпечення вимог до нової системи керування?)

Формулюючи завдання проектування, розроблювач проекту, по суті, повинен логічно зв'язати висновки аналізу технологічного процесу зі змістом наступних розділів пояснювальної записки.

Завдання проектування рекомендується формулювати так, як показано в таблиці 1.2.

Таблиця 1.2

Предмет проектування	Перелік завдань
Система керування устаткуванням технологічного рівня	1 Розрахунки основних параметрів системи керування й устаткування 2 Розробка структурної схеми системи 3 Конфігурування системи керування 4 Моделювання процесу керування виконавчим пристроєм 5 Розробка принципальних схем 6 Розробка структури програмного забезпечення й програмного коду (для однієї із задач керування)
Автоматизована виконавча система керування виробництвом (із проектуванням програмного додатка)	1 Розробка IDEF0-моделей предметної області «як є» і «як повинно бути». 2 Розробка UML-моделей для програми: <ul style="list-style-type: none"> • діаграми варіантів використання програми ; • діаграми послідовності для одного із варіантів використання; • діаграми класів програми. 3 Розробка ER-діаграми бази даних. 4 Розробка схеми розгортання додатка (апаратні засоби та їх зв'язки)

При розробці проекту слід **забезпечити логічну зв'язність** у розв'язанні завдань проектування. Рекомендується наступний порядок виконання проекту (залежно від теми).

При проектуванні системи керування устаткуванням необхідно спочатку розрахувати час робочого циклу, а також параметри периферійних пристроїв, що забезпечують необхідну точність і динаміку.

Після цих розрахунків можна приступити до вибору апаратури керування, вимірювальних перетворювачів і виконавчих засобів, а також до розробки структурної схеми.

Для підтвердження правильності компоновальних рішень необхідно виконати конфігурування системи. Конфігурування дозволяє визначити також адреси всіх каналів сигнальних модулів. Перевірка правильності конфігурації проводиться командою *Consistency Check*.

Якість керування оцінюється шляхом моделювання каналу керування, який включає регулятори, перетворювач енергії, двигун, а також пристрої

зворотного зв'язку. Параметрами оцінки є стійкість процесу, швидкодія й точність системи керування.

Якщо результат моделювання позитивний, виконується розробка схем з'єднань засобів керування з виконавчими пристроями, а також схем підключень каналів уведення-виводу.

На завершення проводиться розробка структури програми користувача й створення програмного коду для однієї із задач керування.

При *проектуванні програмного додатку*, що створюється для розв'язання деяких виробничих завдань, в якості парадигми програмування слід прийняти об'єктно-орієнтоване програмування, базовими концепціями якого є *об'єкти* й *класи*.

На першому етапі необхідно побудувати *функціональну модель* предметної області. Функціональне моделювання виконується із застосуванням технології IDEF0.

Спочатку треба розробити модель функцій існуючої організації бізнес-процесів – модель «як є» (по англ. «AS IS»). Ця модель піддається ретельному аналізу з метою встановлення «вузьких» місць, які можна усунути за допомогою комп'ютерних засобів, точніше програмного додатка. Для уточнення функцій програмного додатка треба розробити другу модель – модель – «як повинно бути» (по англ. «TO BE»).

В основу цієї моделі повинні бути покладені класи, які, по суті, становлять словник предметної області. Кожний клас повинен реалізувати певну поведінку, а сукупність класів створити бажану поведінку системи. Завдання полягає в тому, щоб бажана поведінка, яка на початковому етапі ще не зовсім зрозуміла, була реалізована в повному обсязі відповідно до функцій системи. Тому після побудови моделі TO BE необхідно створити ще ряд моделей, які дозволять уточнити функції системи, доки вони не задовольнять користувачів.

Після ідентифікації сутностей можна перейти до побудови інфологічної моделі бази даних і створенню даталогічних (логічної й фізичної) моделей, які «розуміє» система керування базою даних (СУБД). Для забезпечення функціонування бази даних необхідно розробити інтерфейси користувачів, а також створити відповідні сценарії (командні файли).

Уточнення поведінки об'єктів здійснюється за допомогою діаграм послідовностей. Після їх побудови слід перейти до моделювання статичної структури додатка – до діаграми класів. На діаграмах класів повинні бути вказані всі необхідні для роботи додатка відомості – імена змінних, типи даних, операції й методи, типи взаємодій об'єктів.

Моделювання програми здійснюється з використанням універсальної мови моделювання UML і CASE-інструментів. Застосування CASE-інструментів, наприклад, програми Rational Rose, дозволяє прискорити створення працюючого прототипу додатка й одержати необхідну програмну документацію. При цьому, якщо код буде неповним, його можна буде дописати звичайним шляхом, а потім знову регенерувати в модель.

Відповідно до вимог стандартів OPC (OLE Process Control – відкриті системи процесів керування) програмний додаток повинен функціонувати в інформаційному просторі, тобто він повинен бути інтегрованим в систему інформаційного забезпечення підприємства. Тому на завершальному етапі необхідно передбачити побудову діаграми розгортання програмної системи, яка являє собою модель взаємодії додатка із зовнішнім середовищем.

Розгортання системи в клієнт-серверній архітектурі потребує відображення наступних компонентів:

- *серверна частина* (для зберігання й обробки інформації);
- *клієнтська частина* (робочий інструмент користувача);
- *мережа*, яка забезпечує взаємодію між клієнтом і сервером відповідно до заданих специфікацій.

2 ВИЗНАЧЕННЯ ОСНОВНИХ ПАРАМЕТРІВ СИСТЕМИ УПРАВЛІННЯ ОБЛАДНАННЯМ

Процес проектування системи управління обладнанням починається з визначення основних параметрів майбутньої системи:

1. Визначення тривалості робочого циклу програми управління.
2. Розрахунки потрібної точності й швидкодії вимірювально-перетворювальних каналів
3. Розрахунки динамічних параметрів виконавчих пристроїв.

При розробці проекту слід враховувати, що системи автоматичного управління являються ієрархічними і мають дві важливі властивості:

1 *Властивість пріоритетів* – в ієрархічній системі команди верхнього рівня обов'язкові для виконання нижчестоящим рівнем.

2 *Властивість взаємозалежності* – функціональні можливості й характеристики нижчестоящого рівня повинні бути узгоджені із завданнями й командами вищого рівня.

Облік цих властивостей вимагає зосередити увагу, у першу чергу, на нижньому рівні, тому що саме тут формуються динамічні характеристики системи і її точність. На цьому рівні, перш за все, потрібно ретельно проаналізувати та визначити період дискретності системи управління, тривалість якого пов'язана із часом виконання програми.

2.1 Розрахунки тривалості робочого циклу програми

Тривалість робочого циклу користувацької програми системи керування залежить від швидкості зміни технологічного параметра, яким управляє система, і припустимої помилки керування. Чим вище швидкість зміни технологічного параметра, тим менше повинне бути значення тривалості робочого циклу виконання користувацької програми. Тому при керуванні, наприклад, тепловим процесом, де регулюється температура нагрівання, час робочого циклу може становити одиниці секунд, а при керуванні процесом з високою динамікою, наприклад, приводом подачі верстата з ЧПУ, робочий цикл зазвичай має тривалість 0,1-10 мс.

Якщо від виконавчого пристрою, наприклад, приводу лінійних переміщень верстата, не потрібна висока динаміка, то розрахунки тривалості робочого циклу T_0 ведуться по формулі:

$$T_0 \leq \sqrt{\frac{2\varepsilon_c}{a_{\text{доп}}}} \quad (2.1)$$

де ε_c – задана величина відхилення поточної координати при максимальній швидкості (V_{\max}) у мм; $a_{\text{дон}}$ – допустима величина прискорення у м/с².

Якщо ж виконавчий пристрій працює в умовах високих вимог до динаміки, то розрахунки тривалості робочого циклу програми керування слід вести з урахуванням необхідної *смуги частот системи керування*.

На рисунку 2.1 зображена типова форма бажаної логарифмічної амплітудно-частотної характеристики (ЛАЧХ) цифрової системи керування швидкістю руху. Така характеристика властива для двухкоординатних рухів робочих вузлів верстата із ЧПУ (супорту або столу) при виконанні кругових переміщень, при яких швидкості по координатах змінюються по синусоїдальному закону.

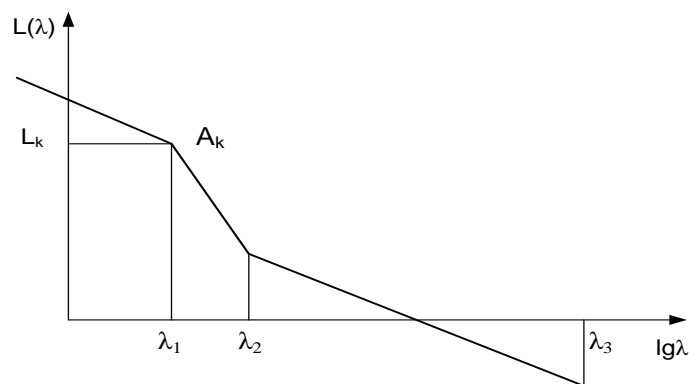


Рисунок 2.1 – Бажана форма ЛАЧХ цифрової системи управління швидкістю руху

Наведена форма ЛАЧХ має наступні особливості:

- можливість усунути позиційну помилку координатного руху – перша асимптота має нахил до осі частот 20 дБ/дек;
- обмеження швидкісної помилки – перша асимптота повинна зайняти певне положення на осі відносної амплітуди $L(\lambda)$;
- забезпечення стійкої роботи системи управління – ЛАЧХ системи перетинає вісь частот асимптотототою з нахилом 20 дБ/дек;
- забезпечення необхідної смуги частот і показника коливальності M – задається певна довжина асимптоти в частотному діапазоні $\lambda_2 \cdots \lambda_3$.

Для наведеної вище форми бажана ЛАЧХ описується дискретною частотною характеристикою (ДЧХ):

$$W_{\text{жс}}(j\lambda) = \frac{K \cdot (1 + \tau_2 \cdot j\lambda)}{j\lambda \cdot (1 + \tau_1 \cdot j\lambda)} \cdot \frac{\left(1 - j\lambda \frac{T_0}{2}\right)^{q_1}}{\left(1 + j\lambda \frac{T_0}{2}\right)^{q_2}}, \quad (2.2)$$

де λ – псевдочастота; $K = K_1$; $\tau_1 = \frac{1}{\lambda_1}$; $\tau_2 = \frac{1}{\lambda_2}$;
 λ_3 – основні параметри, обумовлені вимогами до системи керування;

$\frac{\left(1 - j\lambda \frac{T_0}{2}\right)^{q_1}}{\left(1 + j\lambda \frac{T_0}{2}\right)^{q_2}}$ – характеристика запізнювання вхідних і вихідних сигналів.

Для визначення основних параметрів ДЧХ необхідно перетворити задані параметри технологічного процесу (максимальну швидкість V_{max} та допустиме прискорення $a_{\text{дон}}$) в еквівалентні параметри гармонійного сигналу (амплітуду a_m та частоту ω), а потім визначити положення A_k критичної точки заборонної області ЛАЧХ.

Таке перетворення параметрів можливе у випадку, коли траєкторію руху інструмента, наприклад, фрези, представити функцією:

$$\alpha(t) = \alpha_m \sin \omega t, \quad (2.3)$$

де $a(t)$ – поточне значення координати; a_m – амплітуда (радіус кругового контуру); ω – кутова швидкість руху.

Перша й друга похідні (швидкість і прискорення) гармонійного сигналу визначаються відомими вираженнями:

$$\dot{\alpha}_m = \alpha_m \omega, \quad \ddot{\alpha}_m = \alpha_m \omega^2,$$

де індекси m позначають максимальні значення.

Звідси можна визначити еквівалентні параметри гармонійного руху – частоту $\omega_{\text{э}}$ й амплітуду $a_{\text{эм}}$:

$$\omega_{\text{э}} = \frac{\ddot{\alpha}_m}{\dot{\alpha}_m}, \quad a_{\text{эм}} = \frac{\dot{\alpha}_m^2}{\ddot{\alpha}_m}.$$

Для низькочастотної ділянки ЛАЧХ слушне допущення $\lambda \approx \omega$.

Тоді:

$$W(j\lambda) \approx W(j\omega). \quad (2.4)$$

Якщо відоме значення допустимого відхилення $\theta_m[n]$, то повинна бути виконана умова:

$$|W(j\lambda)| > \frac{\alpha_{\text{эм}}[n]}{\theta_m[n]}. \quad (2.5)$$

Для відносної амплітуди $L(\lambda_k)$ ця умова запишеться в наступному виді:

$$L(\lambda_k) = 20 \lg |W(j\lambda_k)| > 20 \lg \frac{\dot{\alpha}_m^2}{\ddot{\alpha}_m \cdot \theta_m[n]}, \quad (2.6)$$

У системах керування значення максимальної швидкості $\dot{\alpha}_m = V_{\text{max}}$, допустиме прискорення $\ddot{\alpha}_m = a_{\text{доп}}$, а також допустима величина відхилення траєкторії $\theta_m[n] = \varepsilon_c$ повинні бути відомі.

Тоді, враховуючи умови перетворення, для забезпечення необхідної точності у сталому режимі бажана ЛАЧХ повинна проходити вище критичної точки A_k з координатами:

$$\lambda_k = \frac{a_{\text{доп}}}{V_{\text{max}}}; \quad (2.7)$$

$$L(\lambda_k) = 20 \lg \frac{V_{\text{max}}^2}{a_{\text{доп}} \varepsilon_c}. \quad (2.8)$$

При цьому заборонна область обмежується по швидкості першою асимптотою, яка проводиться вліво від точки A_k з нахилом -20 дБ/дек, а по прискоренню – другою асимптотою, яка проводиться вправо від точки A_k із нахилом -40 дБ/дек.

Положення заборонної області показано на рисунку 2.2.

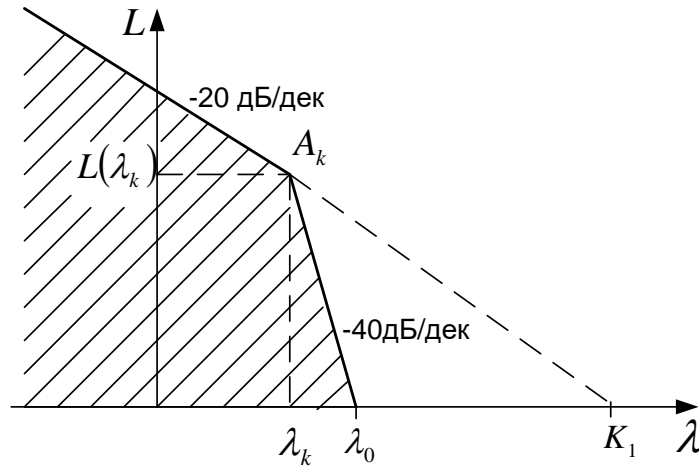


Рисунок 2.2 – Побудова заборонної області за критерієм точності

Швидкісна помилка ε_c й відповідна їй максимальна швидкість V_{\max} визначають необхідну добротність системи по швидкості K_1 :

$$K_1 = \frac{V_{\max}}{\varepsilon_c}. \quad (2.9)$$

Значення K_1 відповідає точці перетинання з віссю λ лінії, яка продовжує першу низькочастотну асимптоту (рис. 2.2).

При побудові слід дотримуватися наступного порядку.

1. Перша низькочастотна асимптота бажаної ЛАЧХ проводиться з нахилом -20 дБ/дек вище точки A_k на 3 дБ, щоб забезпечити запас стійкості. Підйом характеристики приводить до збільшення коефіцієнта добротності по швидкості на величину $\sqrt{2}$:

$$K_1 = \sqrt{2} \frac{V_{\max}}{\varepsilon_c}. \quad (2.10)$$

2. Друга асимптота проводиться з нахилом -40 дБ/дек від точки сполучення з координатами $(\lambda_k; L(\lambda_k) + 3)$ до точки перетинання з віссю λ , яка визначає базову частоту λ_0 заборонної області:

$$\lambda_0 = \sqrt{\frac{a_{\text{доп}}}{\varepsilon_c}} \sqrt{2}. \quad (2.11)$$

3. По заданому показникові коливальності M (зазвичай $M=1,2\dots,7$) визначається частота сполучення другої і третьої асимптот:

$$\lambda_2 = \lambda_0 \sqrt{\frac{M-1}{M}}. \quad (2.12)$$

4. Третя асимптота з нахилом -20 дБ/дек проводиться від точки λ_2 до точки λ_3 , яка також залежить від показника коливальності M :

$$\lambda_3 = \lambda_2 \frac{M+1}{M-1}. \quad (2.13)$$

Четверта асимптота у цифрових системах керування проводиться паралельно осі частот.

Остаточний вид логарифмічної частотної характеристики представлений на рисунку 2.3.

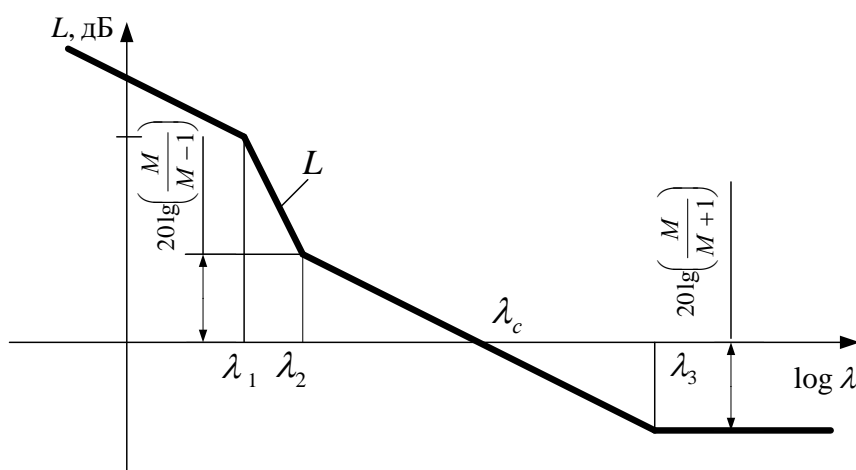


Рисунок 2.3 – Остаточний вид ЛАЧХ для проектованої СУ

У верхньому діапазоні частота λ_3 визначається значенням періоду дискретності T_0 :

$$\lambda_3 = \frac{2}{T_0}.$$

При виборі періоду дискретності слід керуватися вираженням:

$$T_0 \leq \frac{2}{\lambda_3}. \quad (2.14)$$

У наступних розрахунках повинне бути прийняте таке значення T_0 , яке задовольняє умові (2.14).

2.2 Розрахунки точності й швидкодії вимірювальних каналів

Визначення розрядності АЦП або ЦАП проводиться з умови забезпечення точності перетворення.

Величина похибки перетворення залежить від швидкості зміни сигналу на вході АЦП або на виході ЦАП.

Для гармонійного сигналу $a(t)$:

$$\alpha(t) = \alpha_m \sin \omega t \quad (2.15)$$

значення похибки визначається з нерівності:

$$|\Delta_g(T_0)| \leq \alpha_m \omega T_0. \quad (2.16)$$

Приклад. Нехай амплітуда $\alpha_m = 1$, швидкість $\omega = 20 \text{ с}^{-1}$, а період дискретності $T = 0,01 \text{ с}$. Тоді відносна величина помилки складе:

$$\Delta_g = 1 \cdot 20 \cdot 0,01 = 0,2.$$

Як видно із прикладу, навіть при невеликих значеннях частоти сигналу значення похибки становить 0,2, тобто 20% від рівня сигналу, що в багатьох випадках неприпустимо. При збільшенні частоти сигналу це значення буде збільшуватися пропорційно. Єдиним способом зменшення помилки є зменшення періоду дискретизації безперервного сигналу, тобто тривалості робочого циклу системи керування. Тому, наприклад, у системі регулювання вихідного струму частотного перетворювача для забезпечення необхідної точності дискретність повинна бути на рівні 2 мкс при частоті вихідної напруги перетворювача 400-600 Гц (кутова швидкість $\omega \approx 3000 \text{ с}^{-1}$).

Однак похибка дискретизації – це не єдина складова погрішності перетворення. Друга складова цієї погрішності – розв'язна здатність, або абсолютна помилка перетворювача. Зазвичай вона ухвалюється рівній ціні однієї дискрети, тобто 1 у молодшому розряді коду.

Для вибору розрядності коду використовується така характеристика, як динамічний діапазон.

Приклад. Необхідно визначити розрядність ЦАП для подачі сигналу на аналоговий регулятор сили струму, який здійснює регулювання струму від 0 до 600 А з точністю 2 А. Тоді динамічний діапазон буде рівний:

$$D = \frac{600}{2} = 300.$$

Необхідна розрядність (число розрядів) ЦАП визначається цілим числом *ent* (фр. *antъe*):

$$b = ent[\log_2 D] + 1,$$

де додаткова 1 означає заміну дробової частини цілим.

Для наведеного вище прикладу необхідна розрядність ЦАП рівна:

$$b = ent[\log_2 300] + 1 = 9 \text{ (розрядів).}$$

У такий же спосіб визначається й розрядність АЦП.

Приклад. Нехай аналоговий сигнал деякого вимірювального перетворювача змінюється в межах ± 5 В. Необхідна точність його вистави повинна становити не більш 1 мВ. Тоді динамічний діапазон буде дорівнювати відношенню подвійної амплітуди до точності вистави:

$$D = \frac{2 \cdot 5}{0,001} = 10000.$$

Кількість розрядів АЦП, необхідна для такої вистави, дорівнює:

$$b = ent[\log_2 10000] + 1 = 14 \text{ (розрядів).}$$

Параметри швидкодії АЦП і ЦАП важливо враховувати завжди, особливо, коли в системі існує кілька каналів аналогових уведень і аналогових виводів. У більшій мірі це важливо для каналів уведення, оскільки час перетворення в АЦП значно більше часу перетворення в ЦАП.

Проблема полягає в тому, що канали (групи каналів) включаються послідовно й загальний час обслуговування каналів (час циклу) залежить від кількості каналів і груп каналів.

На рисунку 2.4 показана тимчасова діаграма обробки сигналів для модуля SM 331 AI 8×14 bit High Speed, який має найбільш високу швидкодію – час перетворення одного каналу становить 52 мкс, а дозвіл модуля (час його обробки) – 625 мкс.

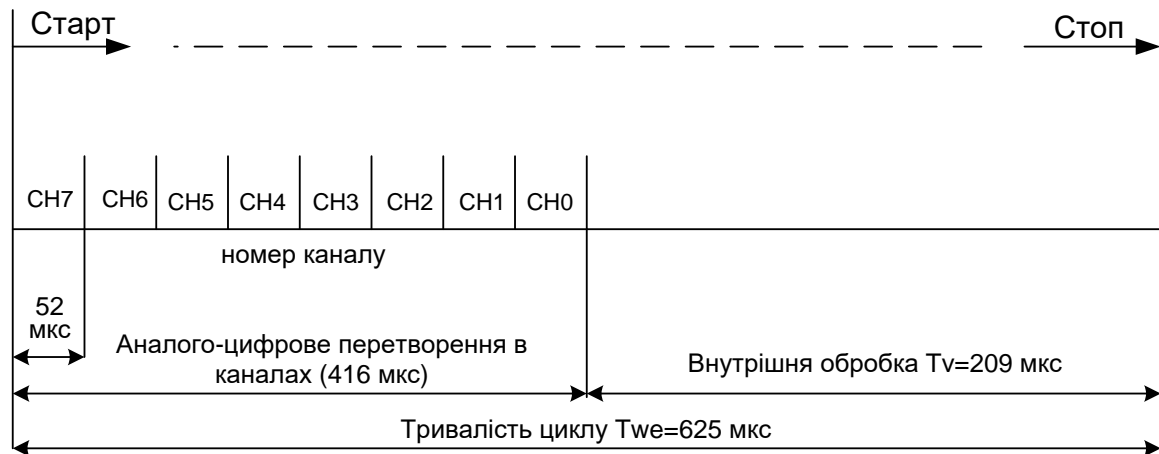


Рисунок 2.4 – Тимчасова діаграма обробки даних у модулі аналогового уведення SM 331 AI 8×14 bit High Speed

Для менш динамічних процесів слід застосовувати звичайний модуль уведення, наприклад, SM 331 AI 8×12 bit, у якого час перетворення каналу становить 100 мс, а час обслуговування модуля становить близько 1200 мс.

Із викладеного слідує, що для обслуговування модулів аналогового уведення необхідно організувати відповідний час циклу або вводити апаратне переривання при готовності модуля передати нові дані.

2.3 Розрахунки динамічних параметрів виконавчих пристроїв

Виконавчі пристрої системи керування, по суті, являють собою фільтр нижніх частот. Параметри такого фільтра слід вибирати так, щоб вони не «обрізали» бажану частотну характеристику системи керування.

Динамічні параметри виконавчих пристроїв багато в чому залежать від їхньої потужності – зі збільшенням потужності зазвичай зростають масогабаритні характеристики й моменти інерції. Крім того, динамічні характеристики залежать від конструктивних особливостей виконавчого пристрою.

Ці обставини створюють проектувальникові певні проблеми, для вирішення яких розробляються комплексні заходи. Так, наприклад, для задоволення високих вимог до динаміки в системах керування приводами подач верстатів із ЧПУ необхідно прагнути до мінімальної довжини кінематичного ланцюга, виключати пружні елементи, застосовувати двигуни з найменшими масогабаритними характеристиками, високими показниками перевантажувальної здатності, мінімальним моментом інерції якоря, а також мінімальними значеннями постійних часу.

У процесі проектування слід застосовувати ті методики розрахунків, які прийняті, як стандартні. Так, наприклад, для розрахунків електроприводів верстатів із ЧПУ слід застосовувати стандартну методику,

наведену в [4].

Виконання силових розрахунків повинне здійснюватися з обов'язковою виставою *розрахункової схеми*, на якій необхідно показати кінематичну схему привода, схему діючих навантажень, розмірні й кінематичні характеристики.

Динамічні властивості виконавчих пристроїв характеризуються постійними часу окремих ланок. Загальне значення постійної часу виконавчого пристрою можна одержати або шляхом розрахунків, використовуючи математичний опис виконавчого пристрою, або шляхом моделювання у математичному програмнім середовищі.

3 РОЗРОБКА -МОДЕЛЕЙ БІЗНЕС-ПРОЦЕСІВ

3.1 Розробка моделей IDEF0

Зазвичай спочатку будується модель існуючої організації роботи AS-IS (як є). На основі моделі AS-IS досягається консенсус між різними одиницями бізнесу по тому, "хто що зробив" і що кожна одиниця бізнесу додає в процес. Модель AS-IS дозволяє з'ясувати, "що ми робимо сьогодні" перед тем, як перестрибнути на те, "що ми будемо робити завтра". Аналіз функціональної моделі дозволяє зрозуміти, де перебувають найбільш слабкі місця, у чому будуть полягати переваги нових бізнес-процесів і наскільки глибоким змінам піддається існуюча структура організації бізнесу.

Деталізація бізнес-процесів дозволяє виявити недоліки організації навіть там, де функціональність на перший погляд видається очевидною. Ознаками неефективної діяльності можуть бути довгі пошуки необхідної інформації, некеровані роботи, роботи, які дублюються, неефективний документообіг (потрібний документ не виявляється в потрібному місці в потрібний час), відсутність зворотних зв'язків по керуванню (на проведення роботи впливає її результат), вхідна інформація використовуються нерационально і т.д.

Знайдені в моделі AS-IS недоліки повинні бути виправлені при створенні моделі TO-BE (як буде) – моделі нової організації бізнес-процесів. Модель TO-BE потрібна для аналізу альтернативних (кращих) шляхів виконання роботи й документування того, як буде змінено бізнес-процес у майбутньому.

Слід указати на розповсюджену помилку при створенні моделі AS-IS - це створення ідеалізованої моделі. Прикладом може служити створення моделі на основі знань керівника, а не конкретного виконавця робіт. Керівник знає, як передбачається виконання роботи згідно посадовим інструкціям і часто не знає, як насправді підлеглі виконують рутинні роботи. У результаті виходить прикрашена модель, яка несе неправильну інформацію і яку неможливо надалі використовувати для аналізу. Така модель називається SHOULD_BE (як повинно б бути).

Технологія проектування інформаційної системи має на увазі спочатку створення моделі AS-IS, її аналіз і поліпшення бізнес-процесів, тобто створення наступної моделі TO-BE, і тільки на основі моделі TO-BE будується модель даних, прототип і потім остаточний варіант інформаційної системи.

Побудова системи лише на основі моделі AS-IS приводить до автоматизації підприємства за принципом "усе залишити як є, тільки щоб комп'ютери стояли", тобто інформаційна система автоматизує недосконалі бізнес-процеси й дублює, а не заміняє існуючий документообіг. У результаті

впровадження така система приводить лише до додаткових витрат на закупівлю встаткування, створення програмного забезпечення й супровід того й іншого.

Іноді поточна AS-IS і майбутня TO-BE моделі різняться дуже сильно, так що перехід від початкового до кінцевого стану стає неочевидним. У цьому випадку необхідна третя модель, що описує процес переходу від початкового до кінцевого стану системи, оскільки такий перехід – це теж бізнес-процес.

На рисунку 3.1 показаний приклад застосування технології IDEF0 для моделювання процесу приймання замовлень у клієнтів Call-центру. Принциповою особливістю даного процесу є однакова кількість входів та виходів процесу (скільки замовлень поступило в систему, стільки ж замовлень повинно бути оброблено).

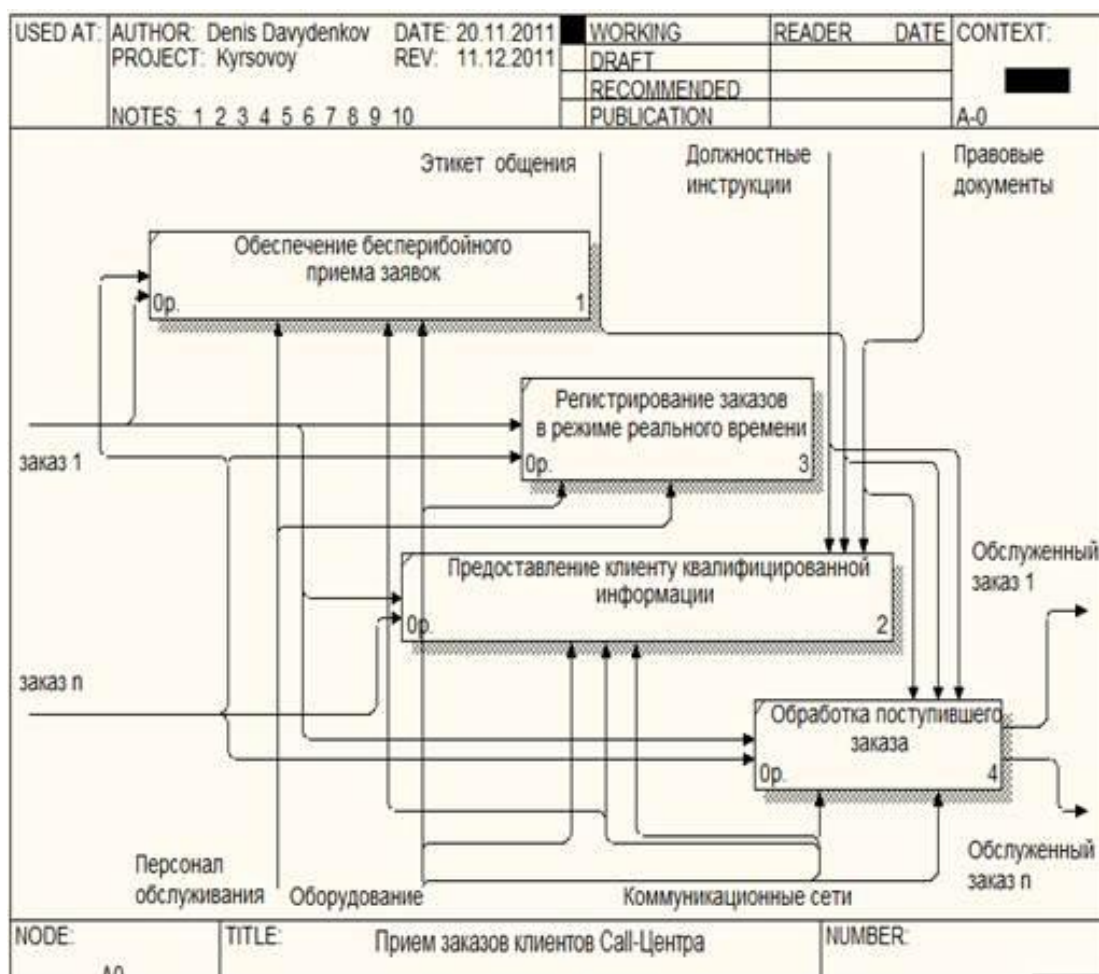


Рисунок 3.1 – Діаграма IDEF0 для моделювання процесу обслуговування клієнтів у Call-центрі

Для повноцінного функціонування Call-центру необхідно забезпечити надійне виконання кожного функціонального блоку.

3.2 Розробка інфологічної моделі предметної області

Ціль інфологічного моделювання полягає у забезпеченні найбільш природних для людини способів відбору й вистави тієї інформації, яку передбачається зберігати в базі даних. Основними конструктивними елементами інфологічних моделей є *сутності*, *зв'язки* між ними й *властивості* (атрибути) сутностей.

Сутність – це будь-який помітний об'єкт, тобто об'єкт, який можна відрізнити від іншого об'єкта по його *атрибутах*. Якщо об'єкти мають однакові атрибути, то вони не різняться по типу й належать одній сутності. Таким чином, атрибути необхідні, тому що вони є поійменованими характеристиками сутностей.

Зв'язок – це асоціювання двох або більш сутностей. Зв'язок дозволяє знайти в базі даних одні сутності за відомими значенням інших, а в програмі зв'язки визначають припустимі взаємодії між об'єктами. При проектуванні баз даних створення зв'язків є найбільш складним завданням.

При побудові інфологічної моделі бази даних можна використовувати мову ER-діаграм (від англ. Entity-Relationship – сутність-зв'язок), універсальну мову моделювання UML, а також мову інфологічного моделювання. Приклад концептуальної ER-діаграми бази даних, яка *не враховує* особливості конкретної СУБД, наведено на рисунку 3.2.

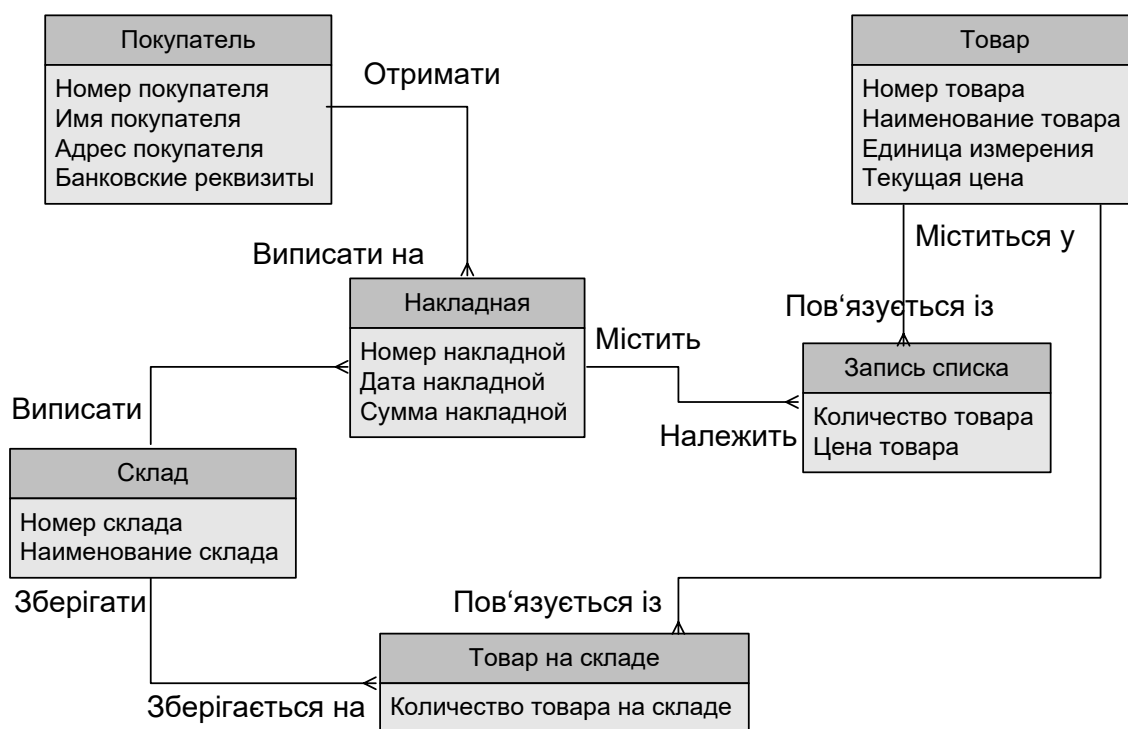


Рисунок 3.2 – Концептуальна ER-діаграма бази даних

У процесі інфологічного моделювання бази даних необхідно не тільки скласти перелік сутностей предметної області, їх атрибутів і зв'язків, але й зібрати інформацію про користувачів і особливості існуючих додатків, що використовують базу даних. Це потрібно зробити для того, щоб урахувати зовсім протилежні вимоги до бази даних з боку прикладних програмістів, що створюють бази даних, вимоги користувачів, що роблять запит на дані, а також вимоги адміністраторів баз даних, що опікуються про виключення можливих викривлень даних при введенні нової інформації або видаленні існуючої.

Слід урахувати, що погано продуману структуру бази даних неможливо буде виправити ніякими додатками.

Особливості моделювання предметної області

Модель предметної області повинна являти собою словник термінів (об'єктів і класів), необхідних для моделювання програмного додатка.

В основу моделі повинні бути покладені діаграми класів. Клас – це місце для розміщення атрибутів (даних про об'єкти) і операцій (функцій, виконуваних об'єктами). Однак при моделюванні предметної області у визначенні атрибутів і операцій немає необхідності – ці процедури слід виконати пізніше. Насамперед необхідно сконцентрувати увагу на виявленні об'єктів і відносин між ними.

Побудова моделі предметної області додатка починається з **виявлення абстракцій, таких сутностей**, які існують у реальному світі і являють собою основні концептуальні об'єкти системи.

При цьому програмне забезпечення повинне бути структуроване так, щоб у центрі виявилися об'єкти із простору задачі.

Однією з головних цілей розробки програми на основі абстракцій реального світу є створення можливості повторного використання того, або іншого програмного модуля. Ця можливість створюється саме на етапі, коли формуються класи, тобто абстракції реального світу. Від вибору класів, тобто від якості абстрагування, залежить також, якою буде спостережувана поведінка об'єкта. Реалізація цієї поведінки забезпечується внутрішнім устроєм (проведеною інкапсуляцією), однак цей устрій не повинен містити надмірностей, які ускладнюють поведінку, або не повинен бути занадто простим, що обмежує поведінку.

При моделюванні предметної області слід рухатися *зсередини назовні*. Це означає, що спочатку в системі виділяються ключові об'єкти, а потім, у процесі вивчення з якими ще об'єктами вони взаємодіють, до ключових об'єктів додаються інші з убутним ступенем значимості, поки не буде створена повна картина взаємодії об'єктів і не буде розроблений інтерфейс, що забезпечує керування цією взаємодією.

Необхідно врахувати, що при виконанні наступного етапу, тобто при виявленні прецедентів і описі поведінки системи, слід застосовувати рух ззовні усередину, поки не буде зрозуміло, за допомогою чого реалізується кожний елемент необхідної поведінки.

Отже, моделювання предметної області – це погляд на систему зсередини назовні й перше, що потрібно зробити при побудові статичної моделі системи, це знайти класи, які адекватно відбивають абстракції предметної області.

Кращими джерелами для виявлення класів є:

- високорівневий опис завдання;
- низькорівневі вимоги до системи.

Практично слід використовувати опис завдання й формулювання вимог, зроблені в першому розділі проекту, і в цих описах і формулюваннях обвести або підкреслити все іменники й іменні групи. Таким способом можна знайти майже всі важливі доменні об'єкти (класи).

У міру уточнення цього переліку повинне відбуватися наступне:

- іменники й іменні групи можуть стати об'єктами або атрибутами;
- дієслова й дієслівні групи можуть стати операціями й асоціаціями;
- родовий відмінок показує, що іменник повинен бути атрибутом, а не об'єктом.

Далі з отриманого списку класів треба вилучити непотрібні (надлишкові або несуттєві) і непридатні (занадто розпливчасті) елементи.

При побудові діаграм класів можна також розглянути можливі узагальнення (відносини виду «є», коли окремі об'єкти можна об'єднати одним іменем). Етап моделювання предметної області – це саме той момент, коли слід також прийняти рішення про агрегацію класів.

Так, наприклад, клас «Привод» містить у собі агреговані класи (об'єкти) – перетворювач, двигун і датчики, що здійснюють задану поведінку привода.

У підсумку модель предметної області, доповнена асоціаціями (статичними відносинами між парами класів), повинна адекватно описувати статичні аспекти задачі, які не залежать від часу, і в цьому нагадувати діаграму сутність-зв'язок, яка застосовується в моделюванні бази даних.

Приклад моделі предметної області для програмного забезпечення банкомата наведений на рисунку 3.3. У моделі показані класи програмного забезпечення і їх взаємозв'язки.

Для розміщення програми використовується *абстрактний* клас «Контролер», який не має чіткої поведінки, а тому не описується атрибутами й операціями. Чітка поведінка реалізується класом «Транзакція», що

забезпечує з'єднання контролера з банком за допомогою класу-інтерфейсу «ІБанк».

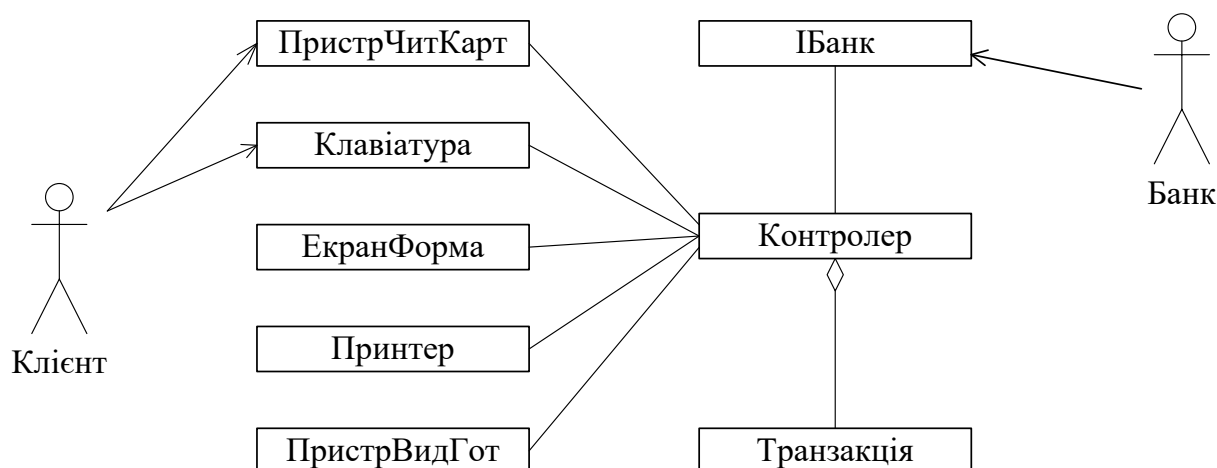


Рисунок 3.3 – Модель предметної області “Банкомат”

Клієнт взаємодіє із програмним забезпеченням банкомата за допомогою пристрою читання пластикової карти, який обслуговується класом «ПристрЧитКарт», а також за допомогою введення інформації із клавіатури, яка обслуговується класом «Клавіатура».

Сервісні функції банкомата реалізують класи «ЕкранФорма» (діалог із клієнтом), «Принтер» (печатка квитанції) і «ПристрВидГот» (видача готівки).

Розповсюджені помилки моделювання предметної області

При моделюванні предметної області студенти допускають наступні помилки:

1. Занадто велику увагу приділяють аналізу всіх іменників і дієслів, упускаючи головні сутності (класи).

При намаганні скласти великий словник є ризик спуститися на низький рівень абстракції, що призведе до надмірного ускладнення програми.

2. Включають у класи операції, не вивчивши деталі прецедентів.

Не слід приділяти занадто багато уваги визначенню операцій на етапі моделювання предметної області. У цей момент ще мало інформації для прийняття обґрунтованих розв'язків про деталі поведінки.

3. При аналізі асоціації виду «є частиною» зазнають труднощів, що використовувати - агрегацію чи композицію.

На стадії моделювання предметної області краще говорити просто про агрегацію. Більш точний вибір слід відкласти на етап детального проектування.

5. Студенти дають класам малозрозумілі імена, наприклад,

sportmgr.Intf *замість* Portfoliomanager (фахівець, відповідальний за аналіз інвестицій). Чим очевидніше імена класів, тим простіше взаєморозуміння учасників процесу розробки програми. Про акроніми й інші види скорочень можна подумати на етапі реалізації.

3.3 Розробка специфікації функцій системи

Програмні додатки різняться між собою набором функцій, які вони реалізують. Набір функцій визначається конкретним завданням автоматизації.

Однак конкретність завдання на початковому етапі проектування зазвичай відсутня. Тому для формування переліку функцій програмного додатка доводиться застосовувати моделювання.

Візуальне моделювання можна представити як деякий процес порівневого спуску від найбільш загальних вистав про завдання автоматизації до уточнених процедур і операцій програмної системи. Для цього спочатку будується так звана діаграма варіантів використання (use case diagram), яка описує функціональне призначення системи або, інакше кажучи, те, що система буде робити в процесі свого функціонування.

Вершинами в діаграмі використання є *актори* й *елементи Use Case*. Актори представляють зовнішній світ, що потребує результатів роботи системи, а елементи Use Case представляють дії, виконувані системою в інтересах акторів. Один актор може використовувати кілька елементів Use Case, і навпаки, кожний елемент може бути використаним декількома акторами.

Між актором і елементом Use Case може бути *тільки один вид відносини* – *асоціація*, яка відображає їхню взаємодію. Асоціація може бути позначена іменем, ролями та потужністю.

Між елементами Use Case можуть існувати два види відносин – *розширення* (extend) і *включення* (include). Ці відносини показано на рисунку 3.4.

Відношення розширення (*extend*) між елементами Use Case означає, що поведінка базового елемента *може бути розширена* поведінкою іншого елемента Use Case.

Відношення включення (*include*) між двома елементами відповідає делегації обов'язків. При цьому елемент, що включається, *реалізує обов'язкову* поведінку.

Елемент Use Case описує, *що повинна робити система*, але не визначає, *як* вона повинна це робити. Це правило дозволяє відокремлювати інтерфейс від способу реалізації системи.

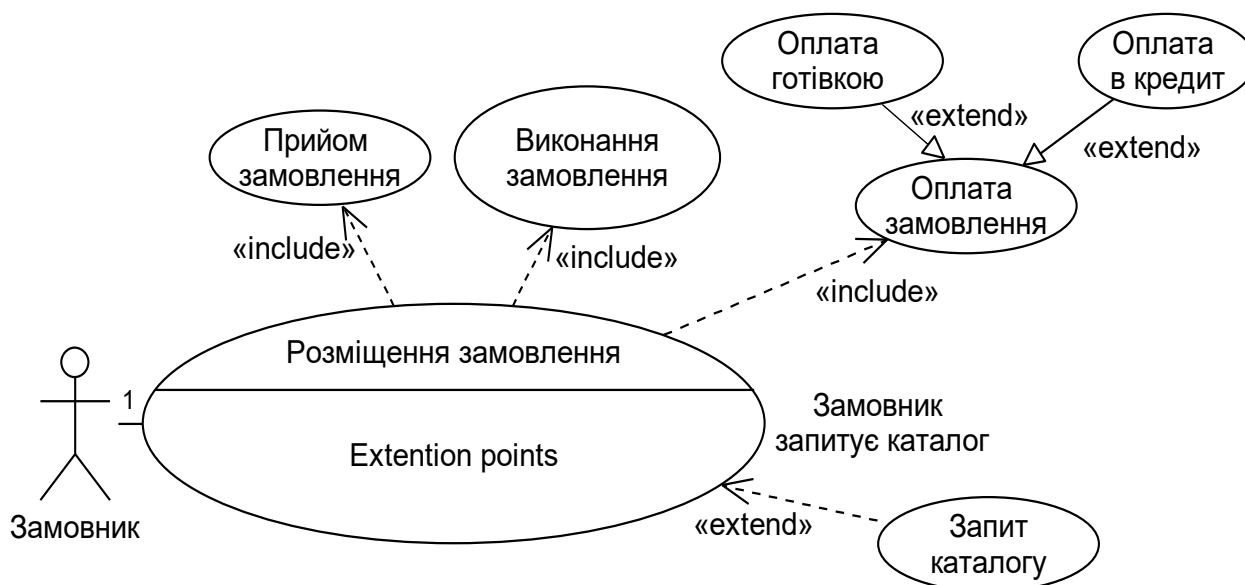


Рисунок 3.4 – Діаграма варіанта використання «Розміщення замовлення» програмної системи інтернет-магазину

Поведінка елемента Use Case описується потоком подій, у якому виділяють:

- основний потік і альтернативні потоки поведінки;
- умови для запуску й зупинки елемента Use Case;
- умови взаємодії елемента Use Case з акторами;
- дані, якими обмінюються актор і система.

Для специфікації прецедентів потрібно використовувати прототипи графічного інтерфейсу, за допомогою яких можна представити, що буде робити користувач і як на цю дію користувача буде реагувати система.

Текст прецеденту слід формулювати в ясній і короткій формі. Кожна пропозиція повинна мати структуру «іменник-дієслово» і повинні бути відразу видні актори й потенційні доменні об'єкти. У міру виявлення нових об'єктів або їх уточнення потрібно відразу оновлювати модель предметної області. Важливо також уважно стежити за можливими альтернативними послідовностями дій (виключеннями) для кожного прецеденту.

Рекомендується наступна послідовність моделювання прецедентів:

1. Створити шаблон опису прецеденту, у якому треба передбачити розділи «Головна послідовність» і «Альтернативні послідовності». Інші розділи не потрібні, вони будуть тільки відволікати увагу.

2. Задати собі питання: «Що повинне відбуватися?». З відповіді на нього починається головна послідовність.

3. *Задати собі питання: «Що ще може відбутися?»*. Важливо врахувати все, що може зробити користувач і зафіксувати всі альтернативні варіанти, тобто виключення. Якщо буде упущений якийсь варіант, ситуація стане невизначеною й система зависне.

Типові помилки при моделюванні прецедентів

1. *Замість словесного сценарію використання студенти пишуть функціональні вимоги.*

Слід чітко відрізнити опис порядку використання від вимог до системи. Вимоги зазвичай формулюються щодо того, ***що повинна робити система***. Сценарій же описує дії, які здійснюються користувачем і системою, при цьому дії системи є її *реакцією* на дії користувача.

2. *Описуються атрибути й методи, а не порядок використання.*

Тексти прецедентів не повинні зайво докладно описувати деталі вистави. Слід також утриматися від посилань на поля в екранних формах. У тексті прецеденту не слід ні йменувати, не описувати методи (операції), тому що вони говорять про те, ***як система робить щось***.

3. *Прецеденти записуються занадто лаконічно.*

При написанні текстів прецедентів, багатослівність більш переважна, чим лаконічність. Переходячи до наступного етапу моделювання (моделювання поведінки), добре б мати частину деталей з опису прецедентів.

4. *Прецеденти повністю абстрагуються від інтерфейсу користувача.*

Один з фундаментальних принципів, заснований на прецедентах, полягає в тому, що при проектуванні системи розроблювачі повинні враховувати дії, які користувачі виконують за допомогою екрана й клавіатури, тобто пам'ятати про інтерфейс.

5. *Не привласнюються явні імена граничним об'єктам.*

Граничними називаються об'єкти, з якими взаємодіють актори. До них належать екранні форми, діалогові вікна й меню. Граничні об'єкти слід іменувати в текстах прецедентів явно.

6. *Прецеденти описуються не з погляду користувача.*

З погляду користувача прецедент найбільше ефективно формулюється із застосуванням *дієслів теперішнього часу в дійсній заставі*, наприклад, «*Користувач вводить пароль*». Прецеденти повинні описувати дії користувача й реакцію на них системи, а такого роду текст найкраще звучить у дійсній заставі.

7. *Описуються тільки дії користувача, а реакція системи ігнорується.*

Текст прецеденту повинен відображати як подію, так і відгук на цю подію, наприклад: «*Коли користувач робить те-те, система робить те-те*». Прецедент повинен описувати те, що відбувається «під капотом» у відповідь

на дії користувача (створення нових об'єктів, контроль даних, які вводяться, або вивід повідомлень про помилки). У тексті прецеденту повинні розглядатися обидві сторони діалогу користувача із системою. Ігнорування опису реакції системи рівносильне ігноруванню поведінки програми.

8. *Опускаються описи альтернативних послідовностей дій.*

Головні потоки зазвичай описувати простіше. Але це не означає, що роботу з альтернативними потоками можна відкласти до етапу детального проектування. Досвід показує, що, коли залишаються нерозкритими важливі альтернативні послідовності, виникають проблеми з доведенням проекту, тому що непередбачувані дії користувачів приводять до «зависання» додатка.

Переходити до подальших етапів процесу розробки треба після того, як досягнуті наступні цілі моделювання прецедентів:

- прецеденти описують усю необхідну функціональність системи;
- для кожного прецеденту чітко й коротко описана головна послідовність дій, а також усі альтернативні послідовності;
- виділені сценарії, загальні для декількох прецедентів.

Завдяки моделі прецедентів, розроблювач додатка може одержати уявлення про динамічний аспект об'єктної моделі.

3.4 Моделювання програмного додатка

Під вираженням «моделювання програмного додатка» розуміють розробку або створення *схеми перетворення специфікації функцій у готовий додаток*.

Для того, щоб моделювати окремі потоки керування в складі прецеденту, застосовуються діаграми взаємодій (interaction diagrams).

Діаграми взаємодій використовуються *для моделювання динамічних аспектів системи*. Вони важливі також для здійснення прямого й зворотного проектування, тобто по моделі згенерувати програмний код і, навпаки, – по програмному коду одержати модель.

До діаграм взаємодії ставляться два типи:

- 1 **Діаграма послідовностей** (Sequence diagram), що акцентує увагу на тимчасовій упорядкованості повідомлень.
- 2 **Діаграма кооперації** (Collaboration diagram), основна увага в якій приділяється структурній організації об'єктів.

Для створення **діаграми послідовності** необхідно, насамперед, розташувати об'єкти, що беруть участь у потоці керування, у верхній її частині уздовж горизонтальної осі. Зазвичай ініціюючий керування об'єкт розміщають ліворуч, а інші – праворуч, за принципом підпорядкованості об'єктів.

Від кожного об'єкта вниз проводиться лінія його життя, на якій розміщуються маленькі прямокутники – фокуси керування. Фокус керування призначений для відображення активного стану об'єкта. Потім на вертикальній осі часу розміщуються горизонтальні лінії повідомлень, які об'єкти посилають і ухвалюють. Це дозволяє одержати наочну картину розвитку потоку керування в часі (рис. 3.5).

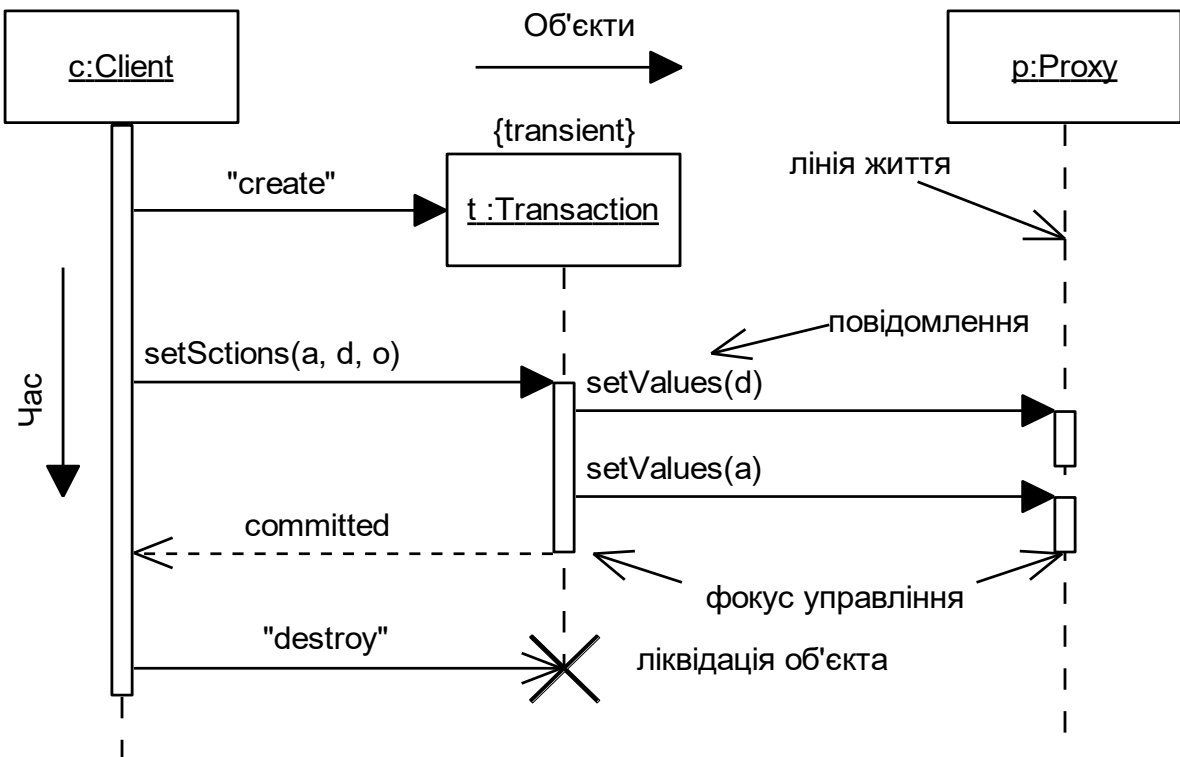


Рисунок 3.5 – Приклад оформлення діаграми послідовностей

UML дозволяє моделювати дії декількох видів:

- **call** (викликати) – викликає операцію, застосовувану до об'єкта;
- **return** (повернути) – повертає значення об'єкту;
- **send** (послати) – посилає об'єкту сигнал;
- **create** (створити) – створює новий об'єкт;
- **destroy** (знищити) – видаляє об'єкт.

Створюючи діаграми послідовності, слід урахувувати, що об'єктам призначаються певні зобов'язання.

Для первісного виявлення деяких об'єктів слід вивчити іменники в потоці подій (керування). Іменники допоможуть визначити, що може бути об'єктом. Якщо не зрозуміло, що описує іменник – об'єкт або атрибут, слід визначити, чи є в нього поведінка. Якщо поведінка відсутня, то це, імовірно, атрибут. Якщо для іменника поведінка визначається, то, швидше за все, це об'єкт.

Окремі об'єкти, виконавши свою роль у системі, можуть бути знищені, щоб звільнити займані ними ресурси. Для таких об'єктів лінія життя

обривається в момент їх знищення. Для позначення моменту знищення об'єкта в мові UML використовується спеціальний символ у формі латинської букви "X" (рис. 3.5). Нижче цього символу пунктирна лінія не зображується, оскільки відповідного об'єкта в системі вже немає і цей об'єкт повинен бути виключений із усіх наступних взаємодій.

Зовсім не обов'язково створювати всі об'єкти в початковий момент часу. Окремі об'єкти в системі можуть створюватися в міру необхідності, суттєво заощаджуючи ресурси системи й підвищуючи її продуктивність. У цьому випадку прямокутник такого об'єкта зображується не у верхній частині діаграми послідовності, а в тій її частині, яка відповідає моменту створення об'єкта, наприклад, об'єкт `t` класу `Transaction` на рисунку 3.5. При цьому прямокутник об'єкта розташовується в тому місці діаграми, яке по осі часу збігається з моментом його виникнення в системі. Очевидно, що об'єкт обов'язково створюється зі своєю лінією життя й, можливо, з фокусом керування.

Фокус керування зображується у формі витягнутого вузького прямокутника, верхня сторона якого позначає початок активності, а її нижня сторона – закінчення активності. Цей прямокутник може замінити лінію життя об'єкта, якщо на всій її протязі він є активним.

Періоди активності об'єкта можуть чергуватися з періодами його пасивності або очікування. У цьому випадку в такого об'єкта є кілька фокусів керування. Важливо розуміти, що одержати фокус керування може тільки існуючий об'єкт, у якого в цей момент є лінія життя. Якщо ж деякий об'єкт був знищений, то знову виникнути в системі він уже не може. Замість нього може бути створений інший екземпляр цього ж класу, який, строго говорячи, буде іншим об'єктом.

Ініціатором потоку керування може бути актор (зовнішній користувач), який зображується у вигляді дротового чоловічка. Актор розміщується на діаграмі послідовності найпершим об'єктом ліворуч зі своїм фокусом керування. При цьому сам актор може мати власне ім'я або залишатися анонімним.

Взаємодія об'єктів представляється повідомленнями, які є специфікацією передачі інформації й показують, що один об'єкт викликає функцію іншого. Далі, коли будуть визначені операції й методи класів, кожне повідомлення стане операцією або викликом методу.

Приклад побудови діаграми послідовності для потоку керування, який ставиться до моделювання процесу керування регуляторами струму й швидкості електропривода від керуючої програми системи ЧПУ, наведений на рисунку 3.6.

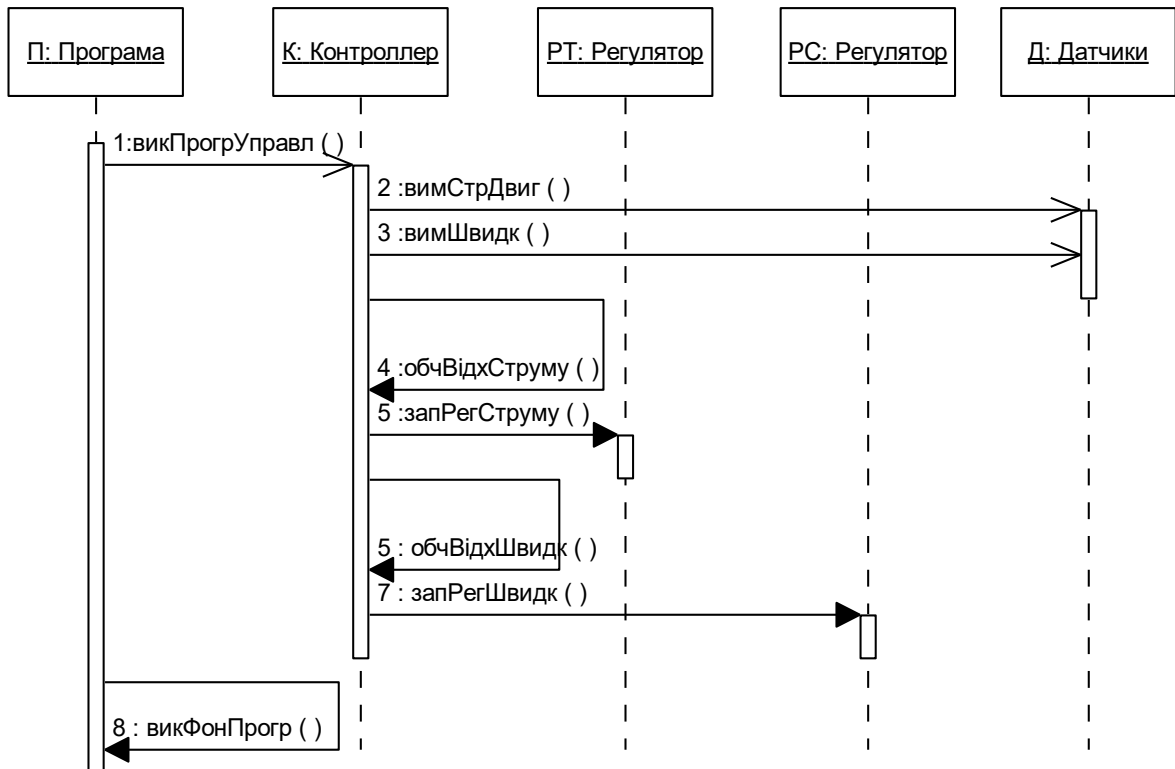


Рисунок 3.6 – Діаграма послідовності системи керування електроприводом

На цій діаграмі показана взаємодія об'єктів, що становлять предметні сутності процесу керування – програма ЧПУ (П), контролер (К), регулятор струму (РТ), регулятор швидкості (РС) і датчики (Д), що включають у себе датчики струму й швидкості.

Послідовність виконання операцій позначена порядковими номерами:

1 Керуюча програма П посилає контролеру К просте повідомлення *викПрогрУправл()* – виконати програму управління електроприводом.

2 Контролер К одержує фокус керування й направляє повідомлення *вимСтрДвиг()* – виміряти поточне значення струму двигуна.

3 Датчик струму повертає конкретне значення струму двигуна, після чого контролер К направляє об'єкту Д повідомлення *вимШвидк()* – виміряти поточне значення швидкості двигуна.

4 Датчик швидкості повертає контролеру конкретне значення швидкості й котроллер переходить на виконання рефлексивного повідомлення *обчВідхилСтруму()* – обчислити відхилення струму.

5 Відхилення струму від заданого значення записується в регістр регулятора струму.

6 Контролер виконує рефлексивне повідомлення *обчВідхШвидк()* – обчислює відхилення швидкості від заданого значення.

7 Відхилення швидкості записується в регістр регулятора швидкості.

8 Керуюча програма викликає диспетчера фоновому режиму для

обслуговування пульта оператора.

Після розробки діаграм послідовності, що моделюють динамічні процеси, слід перейти до розробки діаграм класів, що визначають статичний стан програмної системи. Логіка такої послідовності моделювання полягає в тому, що діаграми послідовності дозволяють уточнити об'єкти і їх поведінку. Об'єкти з однаковою поведінкою можна співвіднести із одним класом і в такий спосіб одержати уточнений перелік класів і уточнену картину їх поведінки.

Побудова діаграм класів – це найважливіший і трудомісткий етап у створенні моделі програми. Кожний клас має набір *методів* (operations) і *змінних* (attributes).

Зазвичай для опису системи створюють кілька діаграм класів. На одних показують деякі підмножини класів і відносини між класами підмножини. На інших відображають ті самі підмножини, але разом з атрибутами й операціями класів. Треті відповідають тільки пакетам класів і відносинам між ними. Для зображення повної картини системи можна розробити стільки діаграм класів, скільки потрібно.

При наявності великої кількості класів вони можуть поєднуватися в пакети (packages). Поєднувати класи можна як завгодно, однак існує декілька *найпоширеніших підходів*.

По-перше, можна групувати класи за стереотипом. У такому випадку виходить один пакет із класами-сутностями, один із прикордонними класами, один з керуючими класами і т.д. Цей підхід може бути корисним з погляду розміщення готової системи, оскільки всі прикордонні класи, які перебувають на клієнтських машинах, уже виявляються в одному пакеті.

Другий підхід полягає в об'єднанні класів по функціональності. Наприклад, у пакеті *Security* (безпека) будуть утримуватися всі класи, які відповідають за безпеку додатка.

Перевага цього методу полягає в можливості повторного використання пакетів. Якщо уважно підійти до групування класів, можна одержати практично незалежні один від іншого пакети.

Третій підхід – це комбінація двох зазначених підходів. На високому рівні можна згрупувати класи по функціональності, наприклад, створити пакет *Security*, відповідальний за безпеку, а на нижньому рівні згрупувати класи, які відповідальні за безпеку, по стереотипах.

Діаграми класів є гарним інструментом проектування. З їхньою допомогою розроблювачі можуть бачити й планувати структуру програмної системи ще до фактичного написання коду, завдяки чому на самому початку можна зрозуміти, чи добре спроектована система.

Повний опис класу включає ім'я класу, стереотип, список атрибутів і

операцій (методів). Усе компоненти повинні мати специфікацію, у якій фіксуються всі властивості й коментарі.

Завдяки тому, що атрибути втримуються усередині класу, вони сховані від інших класів. Але деякі класи мають право читати й змінювати атрибути. Така можливість може бути задана *видимістю атрибута* (attribute visibility).

Найчастіше використовуються наступні значення видимості:

- **public** (загальний, відкритий). Атрибут видно всім іншим класам. Будь-який клас може переглянути або змінити значення атрибута. У нотації UML загальному атрибуту відповідає знак "+";
- **private** (закритий, секретний). Атрибут невидимий ніяким іншим класам. У нотації UML закритий атрибут позначається знаком "-";
- **protected** (захищений). Атрибут доступний тільки самому класу і його спадкоємцям. Нотація UML для захищеного атрибута має знак "#".

У загальному випадку атрибути рекомендується робити закритими або захищеними. Це дозволяє краще контролювати сам атрибут і код, а також уникати ситуації, коли значення атрибута змінюється всіма класами.

У деяких випадках у класи включаються атрибути або методи, які ставляться до всього класу в цілому, а не до окремого екземпляра (об'єкту) цього класу. Для вказівки цього атрибут треба зробити статичним (static).

Ім'я операції визначається її типом. Зазвичай застосовуються чотири типи операцій.

Операції реалізації (implementor operations) реалізують деяку бізнес-функціональність.

Операції доступу (access operations). Атрибути зазвичай бувають закритими або захищеними. Проте, інші класи іноді повинні переглядати або змінювати їхнє значення. Такий підхід дає можливість безпечно інкапсулювати атрибути усередині класу, захистивши їх від інших класів, але все-таки дозволяє здійснювати контрольований доступ. Створення операцій *Get* і *Set* (одержання й зміни значення для кожного атрибута класу) є промисловим стандартом.

Допоміжними операціями (helper operations) називаються такі операції класу, які необхідні йому для виконання його відповідальностей, але про які інші класи не повинні нічого знати. Це закриті й захищені операції класу.

Для ідентифікації операцій необхідно виконати такі дії:

1 Вивчити повідомлення на діаграмах послідовності. Більша частина повідомлень на цих діаграмах являє собою операції реалізації. Рефлексивні повідомлення будуть допоміжними операціями.

2 Розглянути керуючі операції. Можливо, потрібно додати конструктори й деструктори.

3 Розглянути операції доступу. Для кожного атрибуту класу, з яким будуть працювати інші класи, необхідно створити операції *Get* і *Set*.

Однією з особливостей добре спроектованого додатка є порівняно невелика кількість зв'язків у системі. Клас, у якого багато зв'язків, повинен знати про велике число інших класів системи. У результаті його важко буде використовувати в інших додатках. Крім того, складно буде вносити зміни в готовий програмний продукт. Внесення змін у клас може вплинути на поведінку багатьох класів.

На рисунку 3.7 наведений приклад діаграми класів системи керування приводом подачі металорізального верстата. На діаграмі представлений активний клас *УправлПрограма*, який має атрибут (властивість) *траекторіяОбробДеталі* та операції: *викПрограму()*, *аналізСтану()*, *прогнозЗакінчУправл()*.

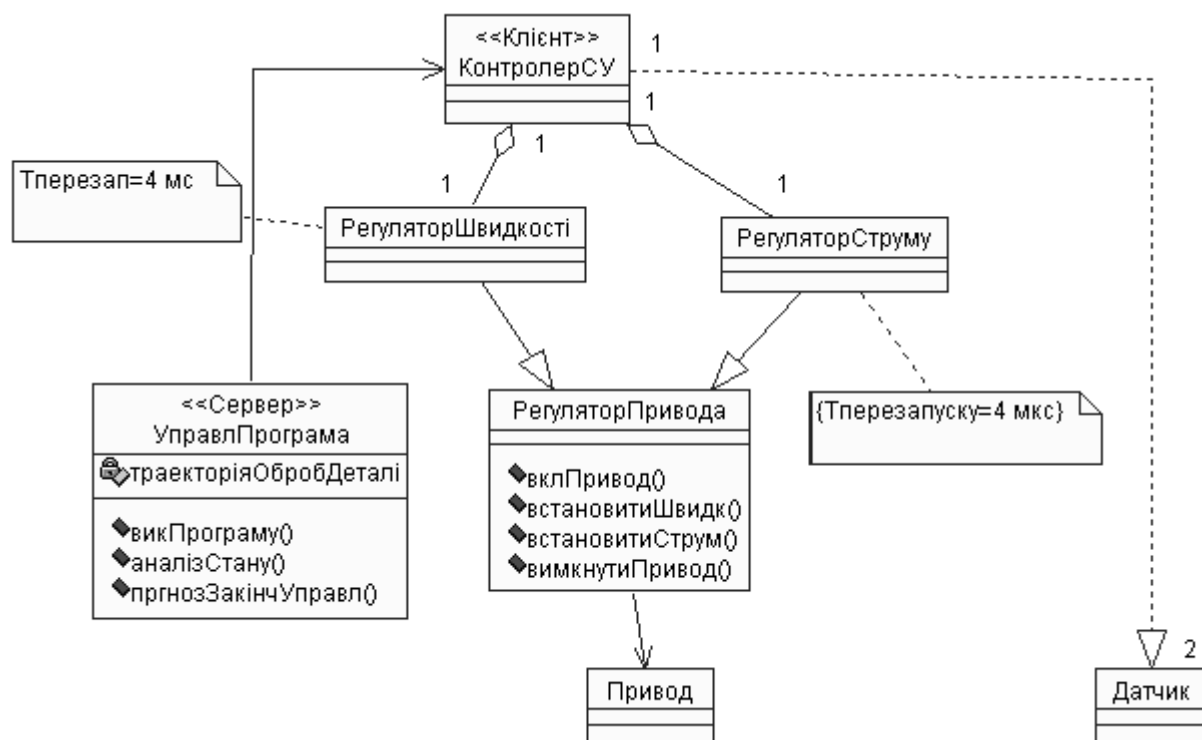


Рисунок 3.7 – Діаграма класів системи керування приводом

Між цим класом і класом *КонтролерСУ* є асоціація у вигляді клієнт-серверного відношення – екземпляри програми задають параметри руху, які повинні забезпечувати екземпляри контролера.

Клас *КонтролерСУ* включає по одному екземпляру класів *РегуляторШвидкості* та *РегуляторСтруму* та має зв'язок реалізації із двома екземплярами класу *Датчик*.

Екземпляри *РегуляторШвидкості* і *РегуляторСтруму* включені в *КонтролерСУ* фізично – тут присутнє відношення *композиція*. Екземпляри *Датчик* включені по посиланню, тобто екземпляр *КонтролерСУ* має лише

показчики (адреси або реєстри) на об'єкти-датчики.

Одночасно *РегуляторШвидкості* і *РегуляторСтруму* – це підкласи абстрактного суперкласу *РегуляторПривода*, який передає їм у спадщину абстрактні операції: *вклПривод()*, *вимкнутиПривод()*, *встановитиШвидкість()* та *встановитиСтрум()*. У свою чергу, клас *РегуляторПривода* використовує конкретний клас *Привод*.

Для класу *РегуляторСтруму* задане обмеження часу на повторне включення 4 мкс, для класу *РегуляторШвидкості* – 4 мс.

На діаграмі класів об'єкти не існують ізольовано друг від друга – вони або зазнають впливу, або самі впливають на інші об'єкти. При цьому поведінка об'єкта є функцією, як його стану, так і виконуваних їм операцій.

Операції позначають обслуговування, яке об'єкт пропонує своїм клієнтам. Можливі п'ять видів операцій *клієнта над об'єктом*:

- 1) *модифікатор* – по цій операції змінюється стан об'єкта, наприклад, вага;
- 2) *селектор* – об'єкт дозволяє доступ до стану, але не змінює його, наприклад, надає інформацію;
- 3) *ітератор* – зміст об'єкта доступний в певному порядку, наприклад, по номенклатурі;
- 4) *конструктор* – операцією створюється об'єкт і ініціалізується його стан, наприклад, включається об'єкт, який забезпечує заміну інструмента;
- 5) *деструктор* – операція руйнування об'єкта й звільнення пам'яті.

У наведеному на рисунку 3.7 прикладі операція *викПрограму()* є модифікатором, операція *аналізСтану()* – ітератором, а операція *прогнозЗакінчУправл()* – селектором.

4 ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ОБЛАДНАННЯМ

У цьому розділі слід розробити «каркас» проекту, тобто розв'язати основні завдання проектування системи. Перелік цих завдань зводиться до наступного:

- 1 Розробка структурної схеми системи керування.
- 2 Вибір засобів та конфігурування системи керування.
- 3 Розробка схем з'єднань і підключень.

Предметом проектування є цифрова система керування, яка містить у собі наступні компоненти:

- засоби програмного керування;
- засоби інтерфейсу;
- інформаційні пристрої;
- виконавчі пристрої;
- перетворювальні пристрої;
- апаратура з'єднання;
- щити й пульти керування;
- алгоритмічне й програмне забезпечення.

Розробка проектної документації проводиться в наступній послідовності.

На першому етапі об'єкт керування треба декомпонувати на окремі вузли й механізми так, щоб можна було визначити:

- 1) де розташовані зони концентрації інформаційних, перетворювальних і виконавчих пристроїв і які відстані між цими зонами;
- 2) де повинні бути встановлені шафи й пульти керування;
- 3) які засоби необхідно застосувати для з'єднання територіально розподілених пристроїв;
- 4) яка інформація повинна відображатися на панелі оператора.

Для вирішення цих завдань корисно скласти план розміщення апаратних засобів системи.

На наступному етапі розробляється спрощена структурна схема системи.

Структурна схема – це графічне зображення структури системи. Під структурою розуміється сукупність частин, на які може бути розділена система за певною ознакою, а також шляхи передачі впливів між ними.

Створюючи структурну схему, слід відображати такі структурні

елементи (приводяться для орієнтування):

- Апаратура керування верхнього рівня.
- Програмувальний контролер.
- Панель оператора.
- Пульти ручного керування.
- Перетворювачі енергії.
- Основні датчики.
- Виконавчі пристрої.
- Шини й лінії зв'язку.

У теперішній час для контролю й керування широко застосовуються багатофункціональні агрегатні системи. Такі системи спрощують виконання монтажних робіт та поліпшують умови експлуатації систем управління.

Практично всі сучасні системи керування технологічного рівня забезпечують можливість інтеграції в системи верхніх рівнів керування за допомогою їх підключення до промислової мережі Industrial Ethernet.

При виборі апаратури керування важливо врахувати, яку інформаційну, апаратну й програмну підтримку забезпечує виробник цієї апаратури. Слід віддати перевагу тим виробникам, які вдосконалюють якість своєї продукції, надають широкий спектр послуг, застосовують гнучкі механізми знижок на ціну продукції, користуються довірою на ринку.

Приклад структурної схеми наведений на рисунку 4.1.

Слід урахувати, що засоби керування повинні мати можливість вибору автоматичного або ручного режиму роботи.

Автоматичний режим є основним (робочим) режимом. Ручний режим необхідний для проведення ремонтно-налагоджувальних робіт, випробовування після завершення налагоджувальних робіт, ліквідації аварійних і позаштатних ситуацій, а також для роботи без локальної системи керування у випадку її відмови. Відмінною рисою цього режиму є те, що всі сигнали керування формуються оператором за допомогою пульта й впливають на перетворювачі, які управляють виконавчими пристроями.

У зв'язку із цим при виборі перетворювача слід ураховувати, що для нормальної роботи виконавчого пристрою в режимі ручного керування в перетворювачі повинні бути інтегровані всі функції керування й захисту виконавчих пристроїв.

При виборі перетворювача для керування електродвигуном необхідно переконатися, що він забезпечує наступні (основні) функції захисту:

- від короткого замикання;
- від перевантаження;
- від підвищення й зниження напруги в ланцюзі постійного струму;

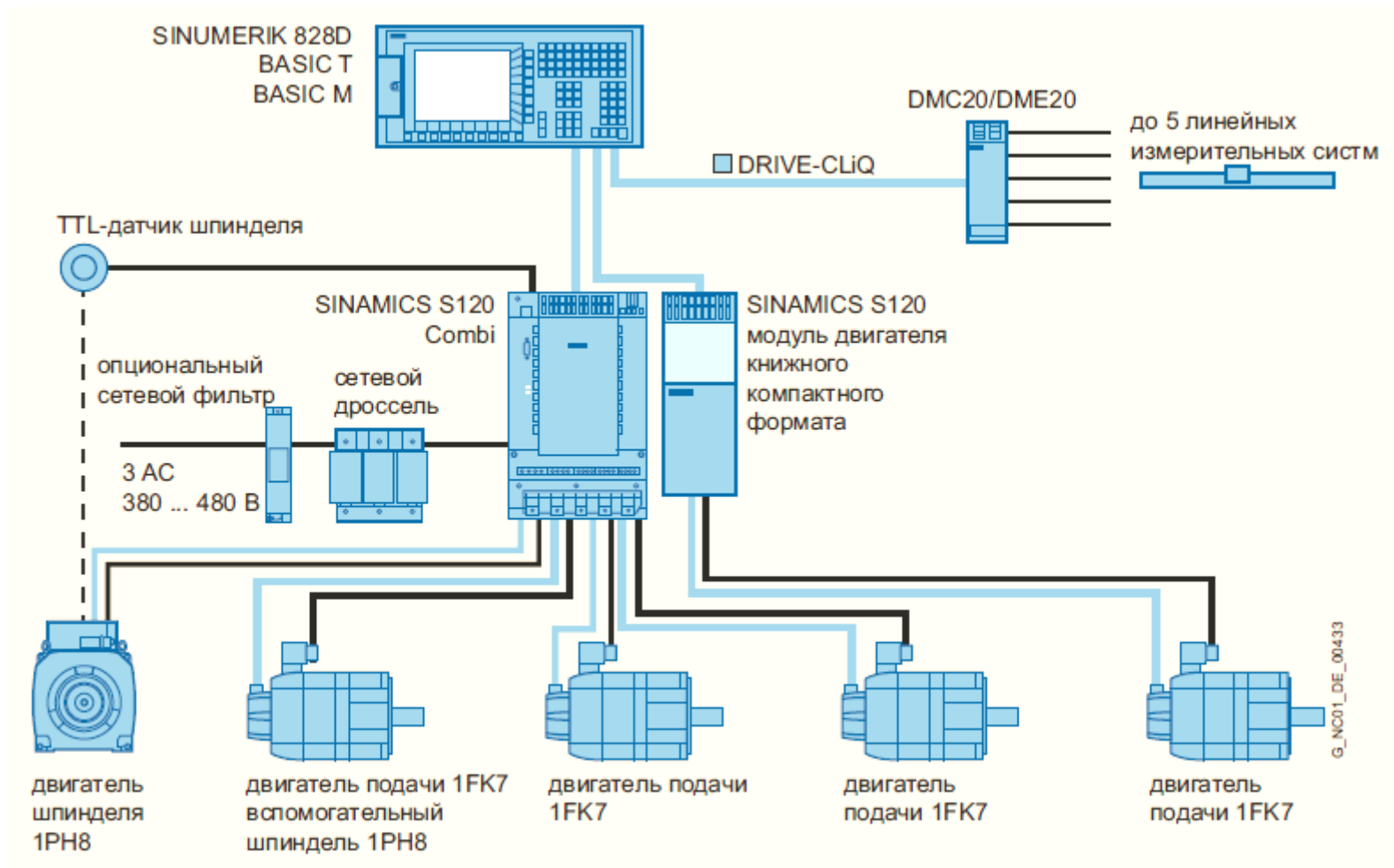


Рисунок 4.1 – Приклад структурної схеми приводної системи SINAMICS верстата із ЧПУ

- від обриву фази живлячої напруги;
- від перегріву радіаторів силових елементів;
- від перегріву двигуна;
- від замикань на землю силових ланцюгів.

Спрацьовування будь-якого вузла захисту повинне приводити до відключення електропривода. Інформація про спрацьовування захисту повинна передаватися на панель керування.

Усі периферійні засоби повинні бути розраховані на підключення до портів уведення-виводу і, у той же час, мати достатню гнучкість та підтримувати нормальне функціонування у випадку заміни програмувального контролера на іншій, більш досконалий, або побудований по іншій архітектурі.

Після вибору основних структурних елементів можна перейти до конфігурування системи керування.

Конфігурування системи автоматизації необхідно для того, щоб виключити помилки в створенні комунікаційного середовища. Для розв'язку завдання конфігурування застосовуються спеціальні механізми й програмні засоби, які забезпечують ретельний аналіз усіх аспектів функціонування майбутньої системи.

Якщо проєктована система керування устаткуванням будується на базі апаратури фірми Siemens, то для її конфігурування необхідно застосовувати програмні додатки Configuring Networks (Netpro) і Hardware Configuration (HW Config) програмного середовища STEP 7.

Фірма Siemens розробила також інструмент швидкого проєктування систем ЧПУ верстатів – програму NCD-Configurator. Ця програма призначена для конфігурування систем типу SINUMERIK.

При використанні апаратури інших виробників конфігурування можна виконати за допомогою програмної системи CoDeSys або SCADA-системи TRACE MODE.

Результати конфігурування представляються в проєкті у вигляді файлу конфігурації, а також скріншотів у двох представленнях – конфігурація контролера і конфігурація розподіленої периферії системи (функціональна архітектура системи).

Після розробки структурної або функціональної схеми слід розробити більш детальне представлення проєкту – скласти принципиальні схеми, схеми з'єднань (монтажні) і схеми підключень (зовнішніх з'єднань).

Принципіальні схеми (електричні, гідравлічні й пневматичні) служать для визначення повного складу приладів, апаратів і пристроїв, а також зв'язків між ними. Принципіальні схеми є основою для розробки монтажних

схем.

Схеми з'єднань (монтажні) призначені для виконання монтажу щитів і пультів, а також для з'єднання частин певної електроустановки. Прикладом такої схеми може служити схема з'єднань перетворювача з електродвигуном і пусковою апаратурою.

Схеми підключення (зовнішніх з'єднань) служать для графічного відображення з'єднань апаратури керування як між собою (для цього використовуються шини, кабелі й дроти), так і із зовнішніми пристроями – вхідними й вихідними. При цьому передбачається, що це електричне устаткування територіально «розкидане».

При виконанні схем підключення необхідно враховувати вимоги, наведені в ГОСТ 2.701-84 і ГОСТ 2.702-75 "Правила выполнения электрических схем".

Схеми підключень пристроїв уведення сигналів будуються в такий спосіб. У верхній частині аркушу розташовуються підключення до лінії напруги живлення (точки найвищого потенціалу). Нижче розташовуються пристрої, що підключаються до модуля уведення. У нижній частині аркушу розташовується умовне зображення модуля. Пристрої й клеми модуля забезпечуються сполучними лініями.

Схеми підключень пристроїв виводу сигналів керування або індикації будуються інакше. Модуль виводу, виходи якого мають найбільший потенціал, зображується у верхній частині аркуша, а зовнішні пристрої, що підключаються до нього і мають нижчий потенціал, – у нижній частині аркуша.

5 РЕКОМЕНДАЦІЇ З МОДЕЛЮВАННЯ СИСТЕМИ

У системах керування технологічного рівня зазвичай моделюють роботу виконавчих пристроїв.

При моделюванні привода постійного струму, для якого розроблений досить повний математичний опис, доцільно застосовувати функціональні моделі пакету MATLAB Simulink.

Для моделювання приводів змінного струму, що відрізняються громіздкими математичними перетвореннями й переходами від однієї системи координат до іншої, більш ефективно застосування віртуальних блоків, наявних у бібліотеках додатка SimPowerSystems пакета MATLAB/Simulink.

Віртуальні блоки дозволяють моделювати перетворювачі, включені в мережу однофазної й трифазної напруги. Типи віртуальних блоків наведені на рисунку 5.1.

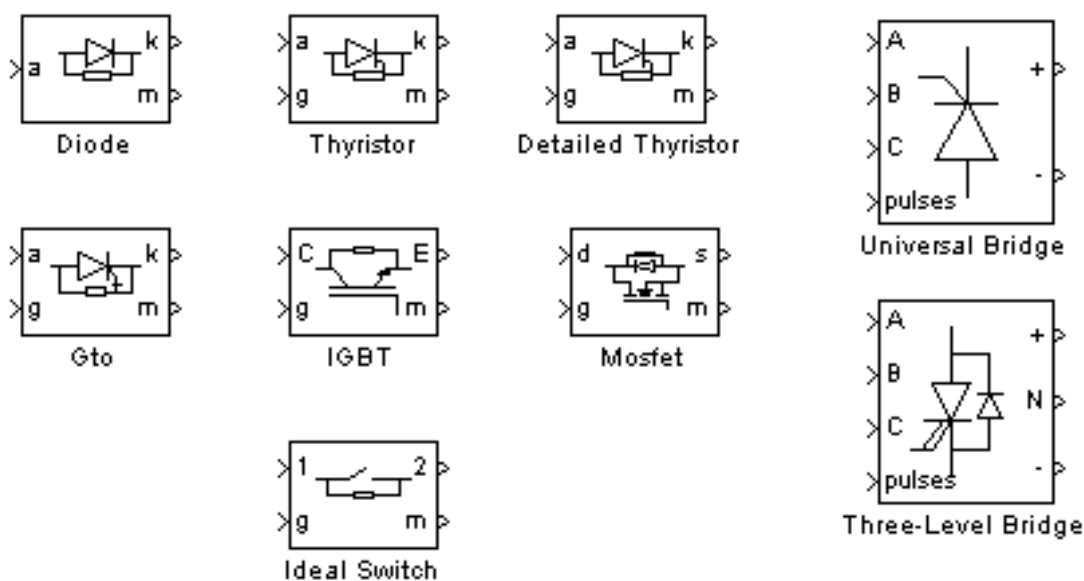
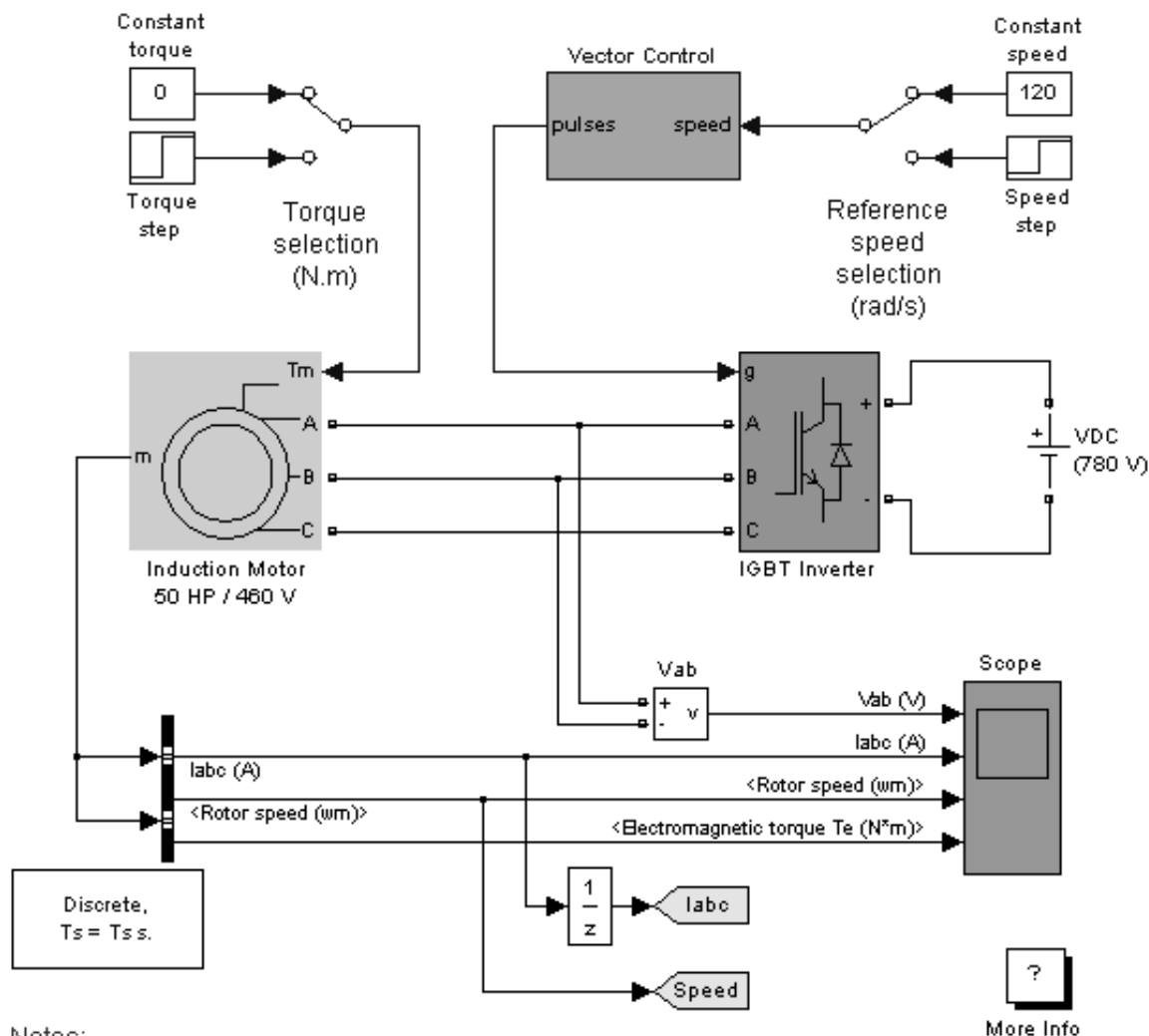


Рисунок 5.1 – Віртуальні блоки напівпровідникових перетворювачів в SimPowerSystems

Слід урахувати, що тиристорні мостові схеми моделюються з використанням блоку Universal Bridge.

Повна віртуальна модель системи векторного керування асинхронним короткозамкненим двигуном з регулятором моменту (power_acdrive.mdl) наведена на рисунку 5.2.

Vector Control of AC Motor Drive



Notes:

Рисунок 5.2 – Структурна схема моделі системи частотного керування двигуном змінного струму

Модель дозволяє досліджувати струм статора, момент і швидкість обертання в перехідних процесах при зміні початкових умов або параметрів налаштування регуляторів.

Модель містить у собі три основні компоненти:

- 1 Віртуальний асинхронний двигун (Induction Motor), завдання параметрів якого проводиться у вікні налаштування.
- 2 Трифазний автономний інвертор на IGBT-транзисторах, параметри якого задаються у вікні його налаштування.
- 3 Система векторного керування (Vector Control).

У свою чергу, модель системи векторного керування, наведена на рис. 5.3 (її потрібно розкрити подвійним клацанням миші), складається з наступних компонентів:

При необхідності дослідження *оптимальних* значень коефіцієнтів регулятора в нелінійних системах керування слід застосовувати пакет прикладних програм Nonlinear Control Design (NCD) Blockset. Цей пакет реалізує *метод динамічної оптимізації Зіглера-Ніколса*.

Набір блоків пакета автоматично набудовує параметри моделюємих систем, ґрунтуючись на заданих користувачем обмеженнях тимчасових характеристик. Враховуючи те, що в цифрових системах керування розрахункові процедури затримують сигнали керування в часі, введення в систему елементів затримки Transport Delay цілком логічно.

У роботі з пакетом використовується метод *click and drag* («клацни й тягни»), який прийнятий в Simulink.

За допомогою пакета (NCD) Blockset можна виконати:

- інтерактивну оптимізацію регуляторів;
- моделювання об'єктів із запізнюванням;
- моделювання схем придушення перешкод;
- налаштування регуляторів в умовах невизначеності параметрів.

Основним блоком пакета є блок оптимізації NCD Output. Приклад його підключення для оптимізації параметрів ПІД-регулятора показаний на рисунку 5.4.

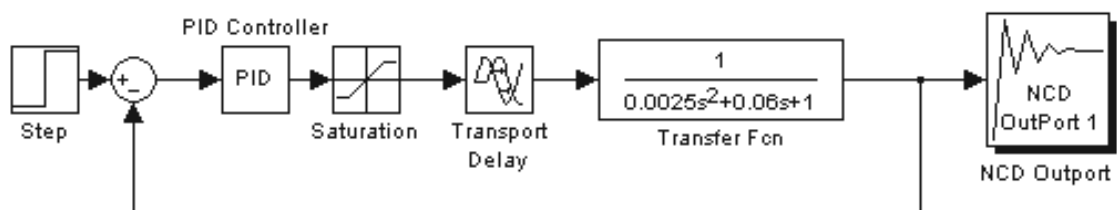


Рисунок 5.4 – Приклад підключення блоку оптимізації NCD Output

Початкові значення параметрів, що підлягають оптимізації, задаються в режимі командного рядка MATLAB, інші параметри вводяться у вікнах налаштування блоку NCD Output.

6 РЕКОМЕНДАЦІЇ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЕКТУ

У сучасних умовах відбувається постійне підвищення складності завдань. Ця складність позначається на можливості застосовувати доступні (відомі) схеми алгоритмів. Щоб зменшити складність, потрібно декомпонувати програму. Декомпозиція дозволяє представити програму у вигляді підпрограм, модулів, компонентів або блоків.

Перевагою такого розподілення програми є можливість переносу й повторного використання готових компонентів програми в інших розробках, що дозволяє прискорити процеси розробки програм. Це особливо важливо для систем керування устаткуванням, де апаратні засоби та виконавчі пристрої, з погляду керування, практично однакові для багатьох процесів. Крім того, програмні компоненти можуть бути створені з використанням тієї мови програмування, яка буде для них найбільш виразною.

В інформаційних задачах створення й обробки документації алгоритми програм не є постійними у зв'язку з мінливими умовами організації виробництва. У таких задачах доводиться враховувати специфічні та перемінливі умови підприємства, а тому для створення програмних додатків доцільно застосовувати об'єктно-орієнтовану методологію розробки програм. Ця методологія дозволяє порівняно легко модернізувати окремі компоненти програми, пристосовуючи їх до нових умов виробництва.

У системах керування технологічного рівня для програмування логічних контролерів застосовуються стандартні мови програмування й стандартні підходи в організації програм. Ці підходи базуються на принципі модульної організації програм, парадигмою якого (по Страуструпу) є: *«Розбий програму так, щоб сховати дані в модулях»*.

У програмних комплексах STEP 7, CoDeSys, TRACE MODE і інших складна програма керування структурується шляхом її розбивки на функціональні блоки. Процес створення блоків закінчується тоді, коли здійснювати подальшу розбивку програми недоцільно або неможливо. При цьому в кожному блоці можна застосувати одну із стандартних мов програмування.

Якщо завдання програми зводиться до керування автоматичним циклом, у якому задіяно ряд виконавчих пристроїв, то програмування слід виконувати мовою S7-HiGraph.

Мова програмування S7-HiGraph заснована на використанні графів станів. Процес розділяється на індивідуальні графи стану з певною функціональною областю дії. Така вистава зручна не тільки для

програмістів, але й для інженерів-механіків, а також інженерів по експлуатації.

Програма структурується в такий спосіб:

1. Завдання автоматизації розділяються на функціональні матеріальні одиниці. Цими одиницями в системі, наприклад, металорізального верстата можуть бути "Затискний пристрій", "Двигун привода головного руху", "Пристрій подачі" і т.п.

2. Поведінка кожної функціональної одиниці описується за допомогою графа станів. Дії, які робить автомат у цих станах, можуть бути зроблені при вході в стан, під час стану й при виході зі стану.

3. Для переходу від одного стану до іншого аналізуються умови, задані при програмуванні. При досягненні встановлених умов автомат здійснює транзакцію – перехід у новий стан.

4. Для програмування умов і дій використовується мова програмування STEP 7 STL.

5. Графи станів вкладаються в групові граfi й можуть зв'язуватися один з одним за допомогою повідомлень. Групові граfi стану можуть використовуватися як координатори (диспетчери).

6. Для групового графа створюється функція ініціалізації (FC) і блок даних (DB), що забезпечують підготовку автомата при включенні системи. Блок даних містить також дані для індивідуальних графів стани.

Таким чином, процес програмування складається з наступних етапів:

1. Створення графів станів.
2. Оголошення змінних.
3. Програмування станів.
4. Програмування транзакцій.
5. Програмування постійних інструкцій.
6. Створення групового графа.
7. Установка послідовності виконання графів.
8. Призначення фактичних параметрів.
9. Компіляція проекту і його збереження.

На рисунку 6.1 показаний граф станів, створений у редакторі програми S7-NiGraph для керування гідروциліндром привода вертикальної подачі інструмента. Тут у початковому стані 0 проводиться ініціалізація й визначається, у який стан повинна перейти функціональна одиниця після включення живлення – або продовжити операцію, перервану відключенням живлення, або встановити початковий стан графа. Із цією метою аналізується значення змінної INIT_SD. Далі перехід з одного стану в інший визначається політикою групового графа.

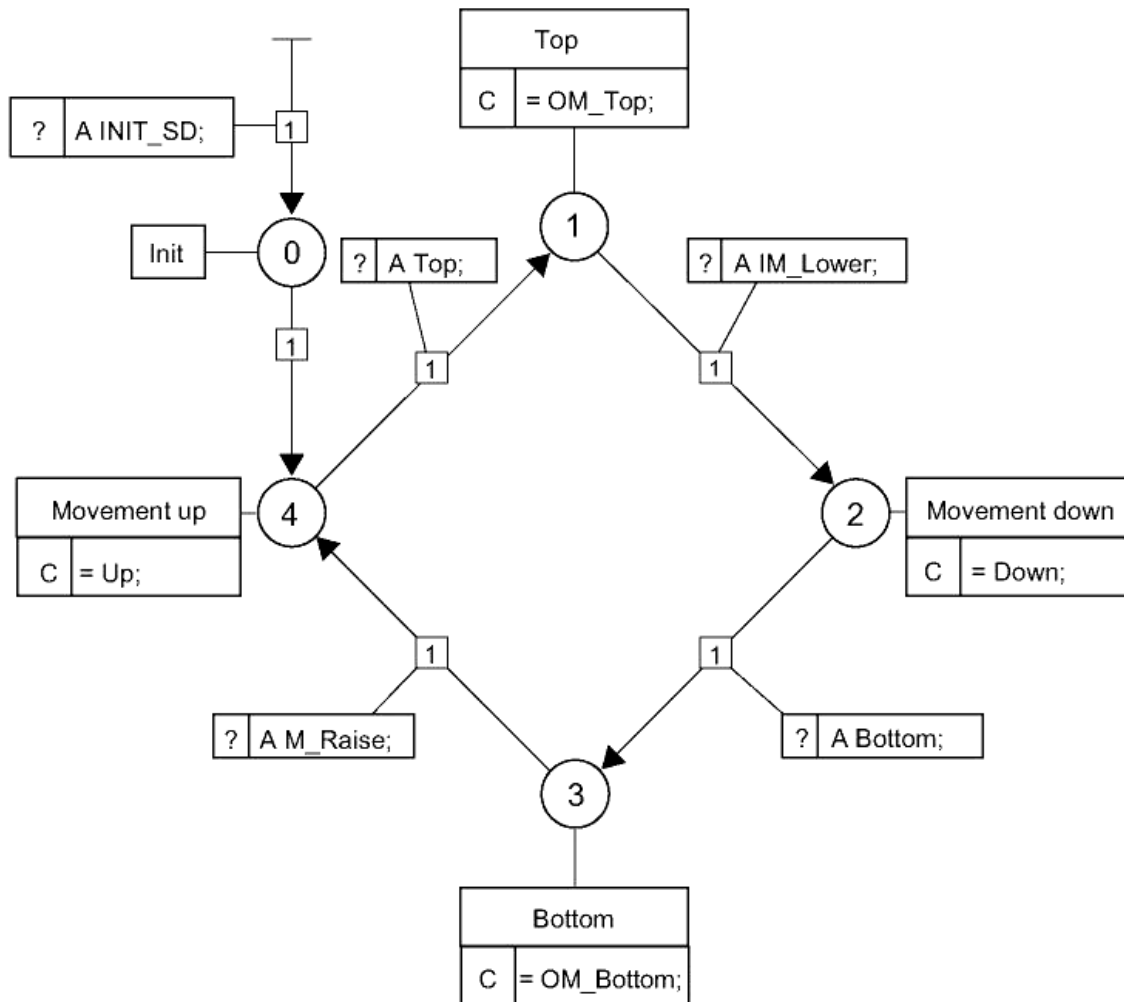


Рисунок 6.1 – Граф станів для програми S7-HiGraph керування гідроциліндром

Підключення станів друг до друга здійснюється командою Transition. У кожному стані програмуються три види дій: дія при вході в стан, циклічні дії в самому стані й дія при виході зі стану, тобто при виконанні наступної транзакції. Пріоритет транзакції задається й відображається в маленькому квадраті, від якого відходить стрілка, що відображає транзакцію.

При додаванні нового стану графа редактор автоматично створює для нього набір змінних (інтерфейс стану), що спрощує процес програмування.

Створена програма може бути налагоджена в цьому ж редакторі.

7 ОФОРМЛЕННЯ ДОКУМЕНТАЦІЇ ПРОЕКТУ

7.1 Вимоги до оформлення текстових документів

Текстові документи слід оформляти з дотриманням вимог стандарту України ДСТУ 3008-95.

У прийнятій на кафедрі АВП системі позначення документів пояснювальна записка курсового проекту в 2-м триместрі студента з номером студентського квитка (залікової книжки), наприклад, 4257 буде мати позначення:

КП02.004257.001ПЗ

На текстових документах основний напис виконується *на аркуші 4* (зміст) за ГОСТ 2.104 – 68 форма 2 (рис 7.1).

Зм	Лист	№ докум.	Підп.	Дата				
Розроб.						Літ.	Лист	Листів
Перев.								
Н.контр.								
Затв.								

Рисунок 7.1 – Форма 2 основного напису текстового документа

Загальні дані про склад проекту виконують за формою, наведеною на рисунку 7.2.

№ п/п	Позначення	Найменування	Кіл.	Примеч.
	<u>Текстові документи</u>			
1	КП02.004257.001.ПЗ	Пояснювальна записка	1	35с.
	<u>Графічні документи</u>			
2	КП02.004257.001ЕЗ	Модуль керування	2	
3	✓	✓		

Рисунок 7.2 – Форма відомості проекту

Відомість проекту забезпечується основним написом за формою 1 (рис. 7.3). Вище від основного напису будується таблиця відомості з наступними розмірами стовпців (зліва направо): 15; 60; 70; 15; 25.

Рядок заголовка виконується висотою 15 мм, інші рядки – 10 мм.

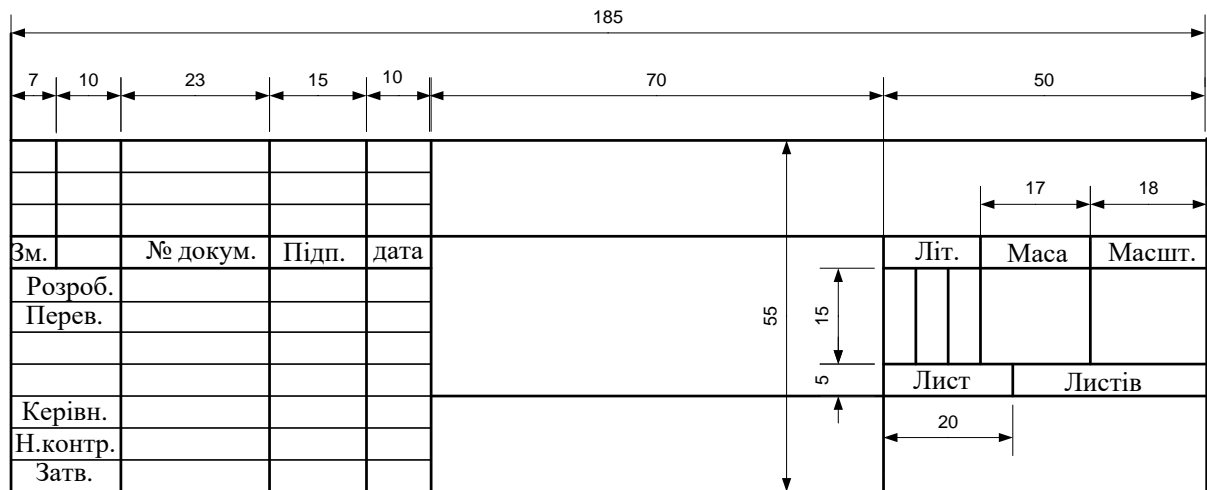


Рисунок 7.3 – Форма 1 основного напису

Пояснювальну записку умовно розділяють на вступну частину, основну частину й додатки.

Вступна частина містить наступні структурні елементи:

- титульний аркуш;
- завдання;
- реферат;
- зміст (лист 4).

Основна частина містить наступні структурні елементи:

- вступ;
- суть звіту;
- висновки;
- перелік посилань.

Додатки розміщують після основної частини звіту.

Реферат призначений для ознайомлення зі звітом. Він повинен бути коротким, інформативним і містити відомості, що дозволяють приймати рішення щодо доцільності прочитання всього звіту.

Реферат повинен містити:

- відомості про обсяг звіту, кількості ілюстрацій, таблиць, додатків, кількості джерел у переліку посилань;
- текст реферату;
- перелік ключових слів.

Текст реферату повинен відображати інформацію, представлену у звіті, у такій послідовності:

- об'єкт дослідження або розробки;
- ціль (мета) роботи;
- результати розробок і їх новизна;
- рекомендації з використання результатів роботи;
- значимість роботи й висновки.

Реферат необхідно виконувати обсягом не більш 500 слів *на одній сторінці формату А4*. Ключові слова, істотні для розкриття суті звіту, поміщають після тексту реферату. Перелік ключових слів включає від 5 до 15 слів (словосполучень), надрукованих прописними буквами в називному відмінку в рядок через коми.

Вступ повинен містити:

- оцінку існуючого стану процесу;
- актуальність даної роботи й підстава для її проведення;
- ціль роботи і перелік завдань;
- об'єкт і предмет дослідження;
- досягнені результати і їх новизна.

Перелік посилань на джерела інформації приводять у порядку, у якому вони вперше згадуються в тексті. Бібліографічні описи посилань у переліку повинні відповідати діючому стандарту по бібліотечній і видавничій справі. Зразки оформлення посилань приводяться в додатку Г.

Додатки містять матеріал, який:

- є необхідним для повноти звіту, але включення його в основну частину звіту може змінити впорядковане й логічне уявлення про роботу;
- не може бути послідовно розміщений в основній частині звіту через великий обсяг або способи відтворення;
- може бути виключений для широкого кола читачів, але є необхідним для фахівців у даній області.

Текст звіту слід друкувати шрифтом Times New Roman розмір 14, дотримуючи наступні розмірів полів: *верхнє, лівє й нижнє – не менш 20 мм, правє – не менш 10 мм*.

При виконанні звіту необхідно дотримувати рівномірної щільності, контрастності й чіткості зображення по всьому звіту.

Структурні елементи "РЕФЕРАТ", "ЗМІСТ", "УВЕДЕННЯ", "ВИВОДИ", "ПЕРЕЛІК ПОСИЛАНЬ" *не нумерують*, а їх найменування служать заголовками структурних елементів.

Розділи й підрозділи повинні мати заголовки. Заголовки структурних елементів звіту й заголовки розділів слід розташовувати в середині рядка й друкувати прописними буквами без крапки наприкінці, не підкреслюючи. *Переноси слів у заголовку розділу не допускаються*.

Заголовки підрозділів, пунктів і підпунктів звіту слід починати з абзацного відступу й *друкувати малими літерами*, крім першої прописний, не підкреслюючи, без крапки наприкінці.

Абзацний відступ повинен бути однаковим по всьому тексту звіту й

рівним *п'ятьом знакам*.

Відстань між заголовком і наступним або попереднім текстом повинна бути не менш трьох інтервалів.

Відстань між рядками заголовка, а також між двома заголовками приймається такою ж, як у тексті.

Не допускається розміщати найменування розділу, підрозділу, а також пункту й підпункту в нижній частині сторінки, якщо, після нього розташований тільки один рядок тексту.

Сторінки звіту слід нумерувати арабськими цифрами, дотримуючи наскрізну нумерації по всьому тексту звіту. Номер сторінки проставляють у правому верхньому або нижньому куті сторінки без крапки наприкінці.

Титульний аркуш включають у загальну нумерацію сторінок звіту. Номер сторінки на титульному аркуші не проставляють.

Розділи підрозділи, пункти, підпункти звіту слід нумерувати арабськими цифрами.

Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Номер підрозділу складається з номера розділу й порядкового номера підрозділу, розділених крапкою. Після номера підрозділу крапку не ставлять.

Ілюстрації (креслення, рисунки, графіки, схеми, діаграми, фотознімки) слід розташовувати у звіті *безпосередньо після тексту*, у якому вони згадуються вперше, або на наступній сторінці. На всі ілюстрації повинні бути дані посилання у звіті.

Ілюстрації слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу. Номер ілюстрації складається з номера розділу й порядкового номера ілюстрації, розділених крапкою, наприклад, рисунок 3.2 – другий рисунок третього розділу.

Ілюстрація позначається словом "Рисунок", яке разом з номером і назвою ілюстрації поміщають *після пояснюючих даних*.

Таблиці застосовують для оформлення цифрового матеріалу або класифікацій. Горизонтальні й вертикальні лінії, які розмежовують рядки таблиці, а також лінії, що обмежують таблицю ліворуч, праворуч і знизу, можна не проводити, якщо їх відсутність не утрудняє користування таблицею.

Таблицю слід розташовувати безпосередньо після тексту, у якому вона згадується вперше, або на наступній сторінці. На всі таблиці повинні бути посилання в тексті звіту.

Таблиці слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу, за винятком таблиць, що приводяться в

додатках. Номер таблиці складається з номера розділу й порядкового номера таблиці, розділених крапкою, наприклад, «Таблиця 2.1» – перша таблиця другого розділу. Таблиця *може мати назву*, яку друкують малими літерами (крім першої прописної) і поміщають над таблицею з абзацним відступом. Назва повинна бути короткою і відбивати зміст таблиці.

Формули й рівняння розташовують безпосередньо після тексту, у якому вони згадуються, посередині сторінки. Вище й нижче кожної формули або рівняння повинне бути залишене не менш одного вільного рядка.

Формули й рівняння у звіті (за винятком формул і рівнянь, наведених у додатках) слід нумерувати порядковою нумерацією в межах розділу. Номер формули або рівняння складається з номера розділу й порядкового номера формули або рівняння, розділених крапкою. Номер формули або рівняння вказують на рівні формули або рівняння в дужках у крайньому правому положенні на рядку.

Формули виконуються в редакторі Equation у *математичному* стилі.

Додатки слід оформляти як продовження звіту на його наступних сторінках. Кожний додаток повинен починатися з нової сторінки. Додаток повинен мати заголовок, надрукований угорі малими літерами з першої прописної симетрично щодо тексту сторінки. Над заголовком у центрі рядка малими літерами з першої прописної повинне бути надруковане слово "Додаток" і далі прописна буква, що позначає додаток.

Додатки слід позначати послідовно прописними буквами (для українського алфавіту виключаються букви Є, З, І, Ї, І, О, Ч, Ї), наприклад, додаток А, додаток Б і т. д.

Один додаток позначається як додаток А.

7.2 Вимоги до оформлення графічної частини

Графічна частина проекту може включати креслення й плакати. Графічна частина оформляється на аркушах формату А4 або на А3. Кількість аркушів визначається завданням на проектування, однак зазвичай достатньо 4-5 аркушів.

Структурні схеми систем керування повинні мати засоби інтеграції у відповідності зі стандартом ОРС. На структурних схемах повинне бути показане, які саме технічні засоби і їх модифікації застосовані в системі автоматизації.

На функціональних схемах технологічне устаткування повинне показуватися спрощено, однак давати ясне уявлення про принцип роботи й взаємодії. Технологічні апарати, трубопроводи, датчики, прилади й засоби автоматизації показуються умовними зображеннями відповідно з вимогами стандартів і повинні мати відповідні написи й позначення.

Принципiальнi електричнi схеми виконують за правилами, установленим дiючими стандартами: При виконаннi електричних схем цифрової обчислювальної технiки користуються правилами за ДСТУ 2.709-72, лiтерно-цифровi позначення в електричних схемах - за ДСТУ 2.710-81. Загальнi вимоги до виконання схем

На принципiальнiй схемi зображують усi електричнi елементи, необхіднi для здiйснення й контролю у виробi заданих електричних процесiв, а також електричнi елементи (рознiмання, затискачi й т.п.), якими закiнчуються вхiднi й вихiднi ланцюги.

Для принципiальної схеми повинен бути складений перелiк елементiв, який може бути оформлений окремим документом.

Зв'язок перелiку елементiв зi схемою здiйснюється через позицiйнi позначення, що складаються з лiтерного позначення й порядкового номера елемента. Запис елементiв у таблицю проводиться в порядку латинського алфавiту, починаючи з лiтерного позначення А и закiнчуючи Z.

Лiнii зв'язку (горизонтальнi й вертикальнi вiдрiзки) повиннi мати мiнiмальне число зламiв, показуються повнiстю, однак якщо це утрудняє читання схем, допускається їх обривати, закiнчуючи стрiлкою з оцiнкою, куди пiдключаються, або номером ланцюга, наприклад: «+12В», «402».

На схемах з'єднання й пiдключення повиннi бути показанi ланцюги живлення, керування, вимiру (контролю) i сигналiзацiї. Елементи схем (модулi, пристрої, шини) повиннi мати позицiйнi позначення й позначення виводiв.

Плакати повиннi бути постаченi заголовками. Основний напис на плакатах не ставиться, однак у нижньому правому кутi листа повинен бути напис «До курсового проекту студ. _____».

8 ПОРЯДОК ВИКОНАННЯ ТА ЗАХИСТУ КУРСОВОГО ПРОЄКТУ

8.1 Етапи виконання та захисту

До основних етапів виконання курсового проєкту належать.

1. Вибір напрямку дослідження.

Визначення актуальності обраної предметної області дослідження, визначення структури роботи й об'єкта дослідження.

Здобувач має право самостійно здійснювати вибір теми курсового проєкту, виходячи із власного професійного інтересу, а також з огляду на актуальність питання (проблеми) для підприємства.

Здобувачу слід враховувати специфіку підприємства, його вимоги щодо збереження корпоративної таємниці, доступність технологічної та техніко-економічної інформації та інші специфічні умови та обмеження в наслідок бойових дій в країні.

2 Здійснення огляду джерел

Аналіз предметної області тематики курсового проєкту виконується на основі огляду інформації, опублікованої в навчальній і науково-технічній літературі, в науково-технічних статтях, вебресурсах, інших доступних джерелах інформації.

3. Формулювання теми курсового проєкту.

Неприпустимим є вибір однієї теми декількома здобувачами вищої освіти в межах однієї академічної групи. Можливим є виконання комплексних робіт в межах одного технологічного об'єкту. Формулювання теми курсового проєкту має бути стислим та вказувати на головний результат дослідження. Теми робіт розглядаються і погоджуються на засіданні кафедри автоматизації виробничих процесів.

4. Виконання курсового проєкту.

Після погодження та затвердження плану роботи здобувач вищої освіти починає написання роботи. Вимоги до структури й оформлення окремих розділів наведені в цих методичних рекомендаціях. У процесі написання окремих розділів здобувач вищої освіти подає їх керівнику на перевірку, виправляє та вносить доповнення у разі потреби, звітує керівнику про готовність роботи. Обговорення проблемних питань з викладачем – керівником здійснюється під час індивідуально-консультативних зустрічей з підготовки та захисту курсового проєкту або на консультаціях викладача відповідно до затвердженого розкладу.

5 Подання курсового проєкту на перевірку.

Контроль виконання, подання на перевірку і представлення закінченої курсового проєкту здійснюється керівником роботи.

Керівник курсового проєкту надає здобувачу освіти свої зауваження, коментарі, рекомендації, на підставі яких він виправляє роботу.

Завершену, остаточно оформлену роботу, підписану здобувачем, керівником здобувач надає для проведення перевірки роботи на наявність плагіату у роботі.

8.2 Права та обов'язки керівника, здобувача вищої освіти

Керівництво та консультування курсовим проектом здійснюється з метою надання здобувачу вищої освіти необхідних консультацій, контролю термінів виконання та якості роботи.

Обов'язки керівника роботи є:

- пояснення основних вимог до курсового проекту;
- узгодження розроблення разом зі здобувачем вищої освіти предметного напрямку дослідження;
- надання консультацій з питань, що виникають у здобувача під час проведення дослідження;
- надання рекомендацій щодо завершального етапу підготовки та захисту курсового проекту.

Обов'язки здобувача освіти:

- ознайомитись із цими методичними рекомендаціями;
- проявляти ініціативність та сумлінність при виконанні курсового проекту;
- своєчасно відвідувати консультації керівника;
- дотримуватися термінів виконання роботи;
- дотримуватися вимог академічної доброчесності при виконанні та захисті курсового проекту.

Права здобувача освіти:

- отримувати консультації та організаційно-методичні консультації з приводу виконання курсового проекту;
- отримувати роз'яснення від керівника щодо вирішення задач курсового проекту, підготовки тексту, підготовки захисту роботи;
- отримувати поради від керівника щодо літературних джерел та інших інформаційних ресурсів, які можна використати при виконанні курсового проекту;
- вимагати дотримання умов об'єктивності та дотримання процедури оцінювання курсового проекту;
- оскаржувати оцінку керівника та комісії з захисту роботи в установленому порядку.

8.3 Застереження щодо академічної доброчесності

Як член студентської спільноти ДДМА здобувач має дотримуватися певних стандартів та академічної політики відповідно «Положення про академічну доброчесність науково-педагогічних, наукових, педагогічних працівників та здобувачів вищої освіти Донбаської державної машинобудівної академії» (<http://surl.li/laufq>):

- шахрайство та плагіат заборонені.
- методичні та інші матеріали, які отримані здобувачами в рамках

процедур організації виконання роботи, захищені авторським правом, можуть бути використані лише тільки здобувачами освіти, яким призначено даний курс. зарахованих на курс для цілей, пов'язаних з цим курсом і не можуть поширюватися.

– спілкування з однокурсниками та викладачем має бути професійним та ввічливим.

– очікується, що здобувач освіти перевірятиме всі власні письмові повідомлення, включаючи поштові повідомлення, на коректність змісту та мови.

– Академія прагне підтримувати середовище, вільне від дискримінації або дискримінаційних домагань, спрямованих на будь-яку людину або групу в межах своєї спільноти – здобувачів вищої освіти, співробітників або відвідувачів.

Виконання роботи має здійснюватися з урахуванням **вимог щодо академічної доброчесності**. Відповідно до статті 42 Закону України «Про освіту»: «**Академічна доброчесність** – це сукупність етичних принципів та визначених законом правил, якими мають керуватися учасники освітнього процесу під час навчання, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання та/або наукових (творчих) досягнень». Головним проявом академічної недоброчесності вважається академічний плагіат. **Академічний плагіат** – оприлюднення (частково або повністю) наукових (творчих) результатів, отриманих іншими особами, як результатів власного дослідження (творчості) та/або відтворення опублікованих текстів (оприлюднених творів мистецтва) інших авторів без зазначення авторства, а саме:

а) відтворення в тексті роботи (повний текст роботи, з коментарями, примітками, бібліографією, переліком джерел та всіма додатками до основного тексту) без змін, з незначними змінами, або в перекладі тексту іншого автора (інших авторів), обсягом від речення і більше, без посилання на автора (авторів) відтвореного тексту;

б) відтворення в тексті роботи, повністю або частково, тексту іншого автора (інших авторів) через його перефразування чи довільний переказ без посилання на автора (авторів) відтвореного тексту;

в) відтворення в тексті роботи наведених в іншому джерелі цитат з третіх джерел без вказування, за яким саме безпосереднім джерелом наведена цитата.

г) відтворення в тексті роботи наведеної в іншому джерелі науково-технічної інформації (крім загальновідомої) без вказування на те, з якого джерела взята ця інформація.

д) перефразування тексту джерела у формі, що є близькою до оригінального тексту, або наведення узагальнення ідей, інтерпретацій чи висновків з певного джерела без посилання на це джерело;

е) подання як власних робіт, виконаних на замовлення іншими особами, у тому числі робіт, стосовно яких справжні автори надали згоду на таке використання.

Запобігання академічному плагіату у роботах, наукових та науково-методичних працях, публікаціях Учасників освітнього процесу полягає у здійсненні технічної перевірки за допомогою спеціалізованих програмних засобів, що використовуються в Академії, та експертної оцінки щодо відсутності/наявності академічного плагіату. Процедура та особливості здійснення автоматизованої перевірки робіт, наукових та науково-методичних праць, публікацій Учасників освітнього процесу наведено в Тимчасовому положенні «Про запобігання та виявлення академічного плагіату у навчальній та науково-дослідній роботі учасників освітнього процесу у ДДМА» та в «Порядку проведення перевірки робіт студентів на наявність запозичень з інших документів»

Рекомендації щодо запобігання академічному плагіату в роботі:

а) робота має виконуватися самостійно, без видання за власний результат чужих робіт і результатів;

б) будь-який текстовий фрагмент обсягом від речення і більше, відтворений в тексті роботи без змін, з незначними змінами, або в перекладі з іншого джерела, обов'язково має супроводжуватися посиланням на це джерело (у формі підрядкового посилання, наприклад як це зроблено щодо Закону «Про освіту» на попередній сторінці); винятки допускаються лише для стандартних текстових кліше, які не мають авторства та/чи є загальноновживаними;

в) якщо перефразування чи довільний переказ в тексті роботи тексту іншого автора (інших авторів) займає більше одного абзацу, посилання (бібліографічне та/або текстуальне) на відповідний текст та/або його автора (авторів) має міститися щонайменше один раз у кожному абзаці роботи, крім абзаців, що повністю складаються з формул, а також нумерованих та маркованих списків (в останньому разі допускається подати одне посилання наприкінці списку);

г) якщо цитата з певного джерела наводиться за першоджерелом, в тексті роботи має бути наведено посилання на першоджерело; якщо цитата наводиться не за першоджерелом, в тексті роботи має бути наведено посилання на безпосереднє джерело цитування («цитується за ХХХХХХХ») і посилання на відповідний пункт списку використаних джерел;

д) будь-яка наведена в тексті роботи науково-технічна інформація має супроводжуватися чітким вказуванням на джерело, з якого взята ця інформація із посиланням на відповідний пункт списку використаних джерел; винятки припускаються лише для загальновідомої інформації, визнаної всією спільнотою фахівців відповідного профілю; у разі використання у роботі тексту нормативно-правового акту достатньо зазначити його назву, дату ухвалення та, за наявності, дату ухвалення останніх змін до нього або нової редакції, а також посилання на відповідний пункт списку використаних джерел.

е) для підтвердження власних аргументів посиланням на авторитетне джерело або для критичного аналізу того чи іншого друкованого твору слід наводити цитати; науковий етикет потребує точно відтворювати цитований

текст, бо найменше скорочення наведеного витягу може спотворити зміст, закладений автором.

Правила цитування та посилання на використані джерела є такими:

1. При написанні здобувач повинен давати посилання на джерела, матеріали з яких наводяться у роботі. Такі посилання дають змогу відшукати документи та перевірити достовірність відомостей про цитування документа, дають необхідну інформацію щодо нього, допомагають з'ясувати його зміст, мову тексту, обсяг. Посилатися бажано на останні видання публікацій. На більш ранні видання можна посилатися лише в тих випадках, коли в них є матеріал, який не включено до останнього видання.

2. Якщо використовують відомості, матеріали з монографій, оглядових статей, інших джерел з великою кількістю сторінок, тоді в посиланні необхідно точно вказати номери сторінок, ілюстрацій, таблиць, формул з джерела, на яке дано посилання в роботі.

3. Посилання додаються одразу після закінчення цитати у квадратних дужках, де вказується порядковий номер джерела у списку літератури та відповідна сторінка джерела (наприклад: [12, с. 172]), або під текстом цієї сторінки у вигляді зноски, в якій вказують прізвище та ініціали автора, назву джерела, видавництво, рік видання та сторінку. При цьому враховувати наступне:

- текст цитати починається і закінчується лапками і наводиться в тій граматичній формі, в якій він поданий у джерелі, із збереженням особливостей авторського написання; наукові терміни, запропоновані іншими авторами, не виділяються лапками, за винятком тих, що викликали загальну полеміку – у цих випадках використовується вираз «так званий»;

- цитування повинно бути повним, без довільного скорочення авторського тексту та без перекручень думок автора;

- пропуск слів, речень, абзаців при цитуванні допускається без перекручення авторського тексту і позначається трьома крапками, вони ставляться у будь-якому місці цитати (на початку, всередині, наприкінці); якщо перед випущеним текстом або за ним стояв розділовий знак, то він не зберігається;

- кожна цитата обов'язково супроводжується посиланням на джерело;

- при непрямому цитуванні (переказі, викладі думок інших авторів своїми словами), що дає значну економію тексту, слід бути гранично точним у викладенні думок автора, коректним щодо оцінювання його результатів і давати відповідні посилання на джерело;

- якщо необхідно виявити ставлення автора роботи до окремих слів або думок з цитованого тексту, то після них у круглих дужках ставлять знак оклику або знак питання;

- коли автор роботи, наводячи цитату, виділяє в ній деякі слова, то робиться спеціальне застереження, тобто після тексту, який пояснює виділення, ставиться крапка, потім дефіс і вказуються ініціали автора дисертації, а весь текст застереження вміщується у круглі дужки. Варіантами

таких застережень є: (курсив наш. – М.Х.), (підкреслено мною. – М.Х.), (розбивка моя. – М.Х.).

До числа **порушень академічної доброчесності**, класифікованих законодавством України, що можуть трапитися при виконанні роботи, належать:

– Академічний плагіат – оприлюднення (частково або повністю) наукових результатів, отриманих іншими особами, як результатів власного дослідження, та/або відтворення опублікованих текстів інших авторів без зазначення авторства;

– Самоплагіат – оприлюднення (частково або повністю) власних раніше опублікованих наукових результатів як нових наукових результатів;

– Академічне шахрайство – будь-які дії учасників освітньо-наукового процесу, сутність яких полягає у наданні або отриманні будь-якої несанкціонованої допомоги або нечесної переваги у будь-якій формі академічної роботи;

– Фабрикація – вигадкування даних чи фактів, що використовуються в освітньому процесі або наукових дослідженнях;

– Фальсифікація результатів досліджень, посилянь у власних публікаціях, будь-яких інших даних, у тому числі статистичних, що стосуються освітнього процесу та наукових досліджень;

– Списування – виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання, зокрема під час оцінювання результатів навчання;

– Обман – надання завідомо неправдивої інформації щодо власної освітньої (наукової, творчої) діяльності чи організації освітнього процесу; формами обману є, зокрема, академічний плагіат, самоплагіат, фабрикація, фальсифікація та списування;

– Хабарництво – надання (отримання) учасником освітнього процесу чи пропозиція щодо надання (отримання) коштів, майна чи послуг матеріального або нематеріального характеру з метою отримання неправомірної вигоди в освітньому процесі;

– Неправомірна вигода – грошові кошти або інше майно, переваги, пільги, послуги, нематеріальні активи, будь-які інші вигоди нематеріального чи не грошового характеру, що обіцяють, пропонують, надають або одержують без законних на те підстав;

– Необ'єктивне оцінювання – свідоме завищення або заниження оцінки результатів навчання здобувачів освіти.

Відповідальність за порушення академічної доброчесності під час здійснення освітньої та наукової діяльності покладається на Працівників та Здобувачів Академії.

Працівників та Здобувачів Академії несуть відповідальність за порушення академічної доброчесності відповідно до вимог законодавства України.

– За порушення норм академічної доброчесності Працівники Академії

можуть бути притягнуті до відповідальності відповідно до нормативних і розпорядчих документів ДДМА та норм законодавства України;

– До Здобувачів Академії, у випадку порушення правил академічної доброчесності, в т.ч. встановлення факту плагіату, може бути застосовано такі види заходів впливу: – академічні (попередження; незарахування роботи; повторне проходження оцінювання; повторне проходження відповідного освітнього компонента освітньої (освітньо-наукової) програми); – дисциплінарні (догана, письмове попередження, відрахування з ДДМА) та ін.

8.4 Регламенти і процедури виявлення порушень вимог академічної доброчесності та наслідки такого виявлення

Основні норми і процедури дотримання академічної доброчесності викладено у «Положення про запобігання та виявлення академічного плагіату у ДДМА (тимчасове)» та «Порядок перевірки на плагіат». З положеннями відповідно можливо ознайомиться за посиланням (<http://surl.li/stzpv>) та (<http://surl.li/tpgzh>).

На першому етапі особа, яка відповідальна на кафедрі за перевірку роботи на наявність плагіату, проводить перевірку електронної версії документу на наявність ознак академічного плагіату за допомогою систем StrikePlagiarism.com (<http://strikeplagiarism.com>) (далі – Системи), використання яких регламентується відповідними угодами Академії. Система формує Звіт подібності, що містить інформацію, яка вказує на наявність текстових та інших запозичень зі знайдених джерел.

Така відповідальна особа не дає оцінку змісту наукової роботи, а виконує виключно технічну перевірку. Подальший аналіз Звіту подібності здійснює науковий керівник.

Інтерпретація показників Звіту подібності системи StrikePlagiarism.com та формування Звіту подібності:

У разі відповідності друкованої та електронної версій роботи Системний оператор проводить перевірку її електронної версії на можливу наявність у тексті робіт чужих опублікованих результатів (текстів) без належного посилання на авторів. За результатами перевірки роботи Антиплагіатною інтернет-системою Strikeplagiarism.com Системний Оператор отримує Звіт подібності, що містить інформацію, яка вказує на ймовірність неправомірних запозичень з інших джерел та інші характеристики, а саме:

- Коефіцієнт Подібності 1. Коефіцієнт Подібності № 1 – це значення (у відсотках), що визначає рівень запозичень, знайдених у певних джерелах (базах даних та Інтернеті), які складаються з фрагментів тексту, що містять щонайменше п'ять слів (довідково: у більшості мов загальновикористаними є фрази з п'яти і більше слів; перевищення встановленого значення Коефіцієнту Подібності № 1 не є ознакою наявності неправомірних запозичень, але вказує на необхідність додаткової перевірки тексту роботи).

- Коефіцієнт Подібності 2. Коефіцієнт Подібності № 2 – це значення (у відсотках), що визначає рівень запозичень, знайдених у певних джерелах (базах даних та Інтернеті), які складаються з фрагментів тексту, що містять щонайменше 25 слів. Інша довжина фрагменту, відмінна від 25 слів, може індивідуально визначатися вченою радою факультету (навчально-наукового інституту) на основі керівних принципів, прийнятих університетом, та рекомендацій, представників компанії StrikePlagiarism.com. (довідково: перевищення встановленого значення Коефіцієнту Подібності № 2 не є ознакою наявності неправомірних запозичень, але вказує на необхідність додаткової перевірки тексту роботи).

- Сигнал „Тривога!“. Сигнал „Тривога!“ з являється, якщо є вірогідність прихованого запозичення. Сигнал „Тривога!“ – це повідомлення, що вказує на наявність у тексті знаків одного алфавіту, замінені схожими знаками іншого алфавіту. Сигнал „Тривога!“ привертає увагу Системного Оператора до можливої необґрунтованості використання зазначених символів, тобто на можливу спробу фальсифікувати результат перевірки з метою збільшення показників оригінальності роботи.

- При перевищенні допустимих значень Коефіцієнтів Подібності, а також при сигналі „Тривога!“ робота підлягають перевірці фахівцем із тематики роботи.

Рекомендовані показники оригінальності за Коефіцієнтом Подібності № 1 становлять: – не більше 20% – оригінальна робота (позитивний висновок), – від 21% до 50% – задовільна оригінальність, – більше 50% – умовно оригінальна робота.

Якщо в Звіті Подібності коефіцієнт подібності № 2 перевищує 5% або наявний сигнал «Тривога!», то робота вважається умовно оригінальною (при будь-якому значенні Коефіцієнту № 1) і підлягає перевірці згідно з цим Порядком.

На підставі Звіту Подібності Системний Оператор готує Протокол контролю оригінальності роботи (Додаток Е).

При отриманні позитивного висновку про відсутність неправомірних запозичень у роботі (робота визнана оригінальною), Протокол надається завідувачу кафедри для проведення попереднього захисту роботи. Якщо за результатами перевірки курсового проєкту робота оцінюється як задовільно оригінальна або умовно оригінальна, Системний Оператор протягом двох наступних робочих днів готує повний Звіт Подібності, який направляє науковому керівнику та завідувачу кафедри. Якщо робота класифікована як задовільно оригінальна, то в її тексті перевіряється наявність та правильне оформлення цитувань та посилань на першоджерела.

Висновок щодо можливості попереднього захисту задовільно оригінальної роботи приймають завідувач кафедри разом із науковим керівником роботи протягом трьох робочих днів із моменту отримання Звіту Подібності (Додаток Д). Висновок оформлюється у двох екземплярах. Один екземпляр Висновку видається студенту, другий залишається в документах кафедри.

Якщо прийнято рішення про неможливість допустити задовільно оригінальну роботу до попереднього захисту, студент має право протягом трьох робочих днів подати на кафедру доопрацьований текст роботи. Після належного оформлення проводиться попередній захист роботи.

Якщо робота визнана умовно оригінальною, то члени кафедри разом із науковим керівником роботи мають проаналізувати: – наявність у роботі великих фрагментів тексту, що ідентифіковані системою як подібні, – наявність детальної подібності роботи та джерела (джерел), розташованого в мережі Інтернет та/або базах даних, – можливість кваліфікувати особливості викладу тексту роботи як механічне переписування вже існуючого документа.

Висновок кафедри щодо можливості захисту умовно оригінальної роботи має бути зроблений протягом двох робочих днів із моменту отримання Звіту Подібності. Висновок кафедри оформлюється у двох екземплярах відповідно до форми, наведеної у Додатку Д. Один екземпляр Висновку видається студенту, другий залишається у документах кафедри.

Якщо прийнято рішення про неможливість допустити умовно оригінальну роботу до попереднього захисту, студенту дається п'ять робочих днів на корегування тексту роботи. Після отримання Системним Оператором оновленого варіанту роботи проводиться повторна перевірка відповідно до цього Порядку. Якщо за результатами повторної перевірки робота оцінюється як умовно оригінальна або неоригінальна, то ця робота не допускається до захисту в Екзаменаційній комісії та не завантажується до бази даних системи. Висновок кафедри про це повинен бути наданий студенту протягом двох робочих днів. Декану факультету подається витяг із протоколу засідання кафедри про недопуск курсового проекту до захисту. Студент підлягає відраховуванню з Академії як такий, що не виконав навчальний план.

Порядок подання апеляції та її розгляд проводиться відповідно «Положення про запобігання та виявлення академічного плагіату у ДДМА (тимчасове)» та «Порядок перевірки на плагіат». З положеннями відповідно можливо ознайомиться за посиланням (<http://surl.li/stzpv>) та (<http://surl.li/tpgzh>)

Порядок подання апеляції та її розгляд проводиться:

- У випадку незгоди з висновком щодо виявлення факту плагіату (компіляцій) у роботі, автор має право у триденний термін з моменту виявлення подати письмову апеляційну заяву на ім'я завідувача кафедрою.

- Для розгляду апеляційної заяви студента створюється апеляційна комісія, персональний склад якої формується розпорядженням завідувача кафедрою з найбільш досвідчених та авторитетних викладачів кафедри (загальний склад від 3 до 5 осіб). Секретарем комісії призначається працівник кафедри.

- Голова апеляційної комісії проводить засідання у тижневий термін з моменту виходу розпорядження завідувача кафедрою про створення апеляційної комісії. Про дату та час проведення засідання заявник

попереджається щонайменше за два дні. Якщо заявник не з'являється на засідання апеляційної комісії, питання розглядається за його відсутності.

- У випадку необхідності отримання додаткової уточнюючої інформації, засідання апеляційної комісії може проводитись у кілька етапів з розривом не більше трьох робочих днів.

- Сумніви, що виникають у членів апеляційної комісії, трактуються на користь особи, робота якої розглядається апеляційною комісією.

- За результатами засідання апеляційна комісія формує висновки, які підписує голова апеляційної комісії, її члени та заявник, зазначаючи «З висновками апеляційної комісії погоджуюсь». Висновки апеляційної комісії щодо академічного плагіату (копіляцій) у творах студентів зберігаються на кафедрі.

8.5 Критерії оцінювання курсового проєкту

Курсовий проєкт є самостійним теоретико-прикладним дослідженням здобувача вищої освіти, що виконується ним на етапі здобуття повної вищої освіти, що засвідчує професійну зрілість випускника, виявити його загальнотеоретичну та спеціальну підготовку, уміння застосовувати здобуті знання для розв'язання конкретних практичних завдань і, відповідно, готовність до самостійної професійної діяльності.

При виставлянні підсумкової оцінки враховується попередні оцінки, виставлені керівником. Підсумкова оцінка може не збігатися з попередніми оцінками роботи.

Підсумкову оцінку курсового проєкту комісія, з спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка», яка враховує такі чинники:

1. Достатній теоретичний рівень розробки проблеми.
2. Актуальність проведеного дослідження та впровадження системи автоматизації.
3. Зв'язок теоретичних положень, розглянутих у роботі, із практикою.
4. Наявність елементів самостійної творчості:
 - формулювання й обґрунтування власного підходу;
 - самостійність аналізу матеріалу;
 - повнота й системність пропозицій по розглянутій проблемній ситуації;
 - самостійний вибір і обґрунтування теоретичної бази досліджень;
 - самостійне формулювання висновків за результатами проведеного дослідження та практичної реалізації системи автоматизації.
5. Використання оригінальних джерел аналітичного й статистичного характеру.
6. Збалансована комбінація кількісних і якісних методів аналізу.
7. Повнота розв'язку поставлених у роботі завдань.
8. Грамотність і логічність у викладі матеріалу.

9. Якість оформлення роботи.

Комісія оцінює всі етапи захисту:

- презентацію результатів роботи;
- розуміння питань, що задаються членами комісії, і якість відповідей на запитання;
- уміння вести професійну дискусію;
- кваліфікацію й загальний рівень розуміння дослідженої проблеми, продемонстровані студентом у процесі захисту;
- загальний рівень культури спілкування з аудиторією.

Критерії оцінювання курсового проєкту та її захисту наведені у табл. 8.1. Роботи, за якими визначено, що вони виконані без дотримання вимог академічної доброчесності, не оцінюються і до захисту не допускаються.

Таблиця 8.1 – Критерії оцінювання курсового проєкту

Критерії оцінювання курсового проєкту	Оцінка
Текст проєкту свідчить про оволодіння навичками самостійного (під керівництвом викладача) проведення роботи: відбір і аналіз літератури, узагальнення і творче осмислення теоретичних основ вирішення проблеми, формулювання висновків. Всі завдання виконані, мета роботи досягнута. Методи дослідження відібрані і застосовані коректно. Сформульовані в роботі пропозиції обґрунтовані і достатні. Текст роботи викладений логічно, послідовно, науково-професійною державною мовою, з коректним використанням професійної термінології. Оформлення роботи цілком відповідає вимогам. Під час захисту роботи доповідь відображала усі її основні положення, висновки і рекомендації. Презентація під час доповіді повністю відповідала її змісту. Під час відповідей на питання здобувач показав повне володіння матеріалом роботи, аргументовано відстоював свої ідеї.	90-100 балів А «Відмінно»
Курсовий проєкт виконано на високому професійному рівні, вона містить усі необхідні елементи, має практичне значення. Всі завдання роботи виконані, мета досягнута. Висновки та пропозиції у роботі в цілому достатньо обґрунтовані й логічні. Технічні та програмні складові роботи відповідають сучасним вимогам. Доповідь на захисті стисла, логічна, проголошена вільно. Презентація доповіді повністю відповідає її змісту. При відповіді на запитання здобувач вищої освіти в цілому продемонстрував високий рівень володіння матеріалом, однак окремі відповіді не зміг чітко аргументувати.	81-89 балів В «Добре»
Тема проєкту в цілому розкрита, але мають місце окремі недоліки не принципового характеру (неповнота теоретичного огляду літературних джерел, запропоновані програмні	75-80 балів С «Добре»

Критерії оцінювання курсового проєкту	Оцінка
<p>рішення є неоптимальними, допущені незначні помилки у формулюванні висновків). Текст роботи свідчить про оволодіння навичками самостійного (під керівництвом викладача) виконання курсового проєкту: проаналізована предметна область, коректно реалізована сформульована задача автоматизації, причинно-наслідковий зв'язок між результатами аналізу, висновками і пропозиціями не порушений. На захисті доповідь логічна, проголошена вільно, але затягнута і містить несуттєві проміжні результати і подробиці. Презентація доповіді в цілому відповідає її змісту, але має недоліки оформлення. Відповіді здобувача на запитання членів комісії загалом вірні, але недостатньо конкретні та/або неповні.</p>	
<p>В цілому завдання проєкту виконані й мета досягнута. Текст роботи викладений логічно, послідовно, науково-професійною державною мовою з коректним використанням професійної термінології. В оформленні роботи допущені незначні помилки. Згідно із змістом теми курсового проєкту загалом розкрита, але є зауваження змістовного характеру (проаналізовано недостатньо джерел інформації, не в повному обсязі реалізовані поставлені задачі, недостатньо обґрунтовані запропоновані рішення; висновки і пропозиції не повністю відповідають завданням тощо). Здобувач освіти під час доповіді недостатньо розкрив усі суттєві положення роботи, презентація доповіді не повністю відповідає її змісту та/або має вади оформлення. Здобувач під час захисту не завжди міг відповісти на запитання по суті роботи, аргументувати свої відповіді.</p>	<p>61-74 бали D «Задовільно»</p>
<p>В основному завдання проєкту виконані й мета досягнута. Текст роботи свідчить про помилки в оволодінні навичками самостійного (під керівництвом викладача) проведення роботи: аналіз предметної області недостатньо повний для вирішення поставлених завдань; при реалізації задач автоматизації проблеми допущені помилки. Запропоновані рішення щодо технічного та програмного забезпечення системи автоматизації недостатньо обґрунтовані, при їх реалізації допущені помилки. Доповідь під час захисту не була достатньо чіткою, побудована недостатньо логічно і послідовно та/або не повністю відображала всі суттєві результати, висновки і пропозиції. Презентація до доповіді оформлена зі значними недоліками, неповна або містить матеріал, який не ілюструє тези доповіді. Здобувач демонструє суттєві труднощі з аргументацією власних ідей, недостатньо володіє професійною термінологією, на значну</p>	<p>55-60 балів E «Задовільно»</p>

Критерії оцінювання курсового проєкту	Оцінка
кількість запитань не може дати відповідь.	
Текст проєкту свідчить про значні прогалини в оволодінні навичками самостійного (під керівництвом викладача) виконання роботи: аналіз предметної області недостатньо повний для вирішення поставлених завдань; при постановці задач автоматизації проблеми та їхній реалізації допущені помилки. Сформульовані в роботі пропозиції щодо технічного та програмного забезпечення системи автоматизації є недостатньо обґрунтованими і неповними. Окремі завдання роботи виконані, але мета досягнута не повністю. Текст роботи викладений недостатньо логічно і послідовно, містить стилістичні помилки, використання професійної термінології не завжди коректне. В оформленні роботи є суттєві невідповідності вимогам. Оцінка рецензента негативна. Під час захисту здобувач освіти у доповіді не зміг розкрити результати аналізу, аргументувати висновки і пропозиції, погано знає матеріал роботи і погано володіє професійною термінологією. Презентація до доповіді оформлена зі значними недоліками, неповна або містить матеріал, який не ілюструє тези доповіді. Здобувач не зміг відповісти на переважну кількість запитань комісії.	35-55 балів FX «Незадовільно»
Текст проєкту свідчить про значні прогалини в оволодінні навичками самостійного (під керівництвом викладача) проведення дослідницької роботи: відбір і аналіз літератури недостатньо повний для вирішення поставлених завдань, аналіз предметної області недостатньо повний для вирішення поставлених завдань; при постановці задач автоматизації проблеми та їхній реалізації допущені помилки. Сформульовані в роботі пропозиції щодо технічного та програмного забезпечення системи автоматизації є недостатньо обґрунтованими і неповними. Окремі завдання роботи виконані, але мета досягнута не повністю. Текст роботи викладений недостатньо логічно і послідовно, містить стилістичні помилки, використання професійної термінології не завжди коректне. Відсутні логічна побудова роботи, її системність та глибина дослідження. Оформлення роботи не відповідає вимогам Оцінки наукового керівника і рецензента негативні. Робота до захисту не допускається.	0-34 бали F «Незадовільно»

8.6 Порядок оскарження результатів оцінювання курсового проєкту

У разі незгоди з оцінкою здобувач вищої освіти має право подати апеляцію відповідно п.5. «ПОЛОЖЕННЯ про екзаменаційні комісії

Донбаської державної машинобудівної академії з атестації здобувачів вищої освіти» (<http://surl.li/tppze>). Апеляція подається особисто здобувачем вищої освіти на ім'я ректора в день проведення захисту випускної роботи або оголошення результатів письмового екзамену, але не пізніше ніж на наступний робочий день після оголошення результатів.

За дорученням ректора у разі надходження апеляції, розпорядженням першого проректора за його головуванням створюється комісія для розгляду апеляції. Апеляція розглядається апеляційною комісією не пізніше ніж на наступний робочий день після її подання. Висновки апеляційної комісії оформлюються відповідним протоколом. У разі встановлення апеляційною комісією порушень під час проведенні атестації, які вплинули на результати оцінювання, рішенням екзаменаційної комісії на підставі висновків комісії з розгляду апеляції здійснюється зміна оцінки.

У випадку незгоди з висновком щодо виявлення факту плагіату (компіляцій) у роботі відповідно п.8 «Положення Про запобігання та виявлення академічного плагіату у навчальній та науково-дослідній роботі учасників освітнього процесу у ДДМА» (<http://surl.li/stzpv>), автор має право у триденний термін з моменту виявлення подати письмову апеляційну заяву на ім'я завідувача кафедрою.

Для розгляду апеляційної заяви студента створюється апеляційна комісія, персональний склад якої формується розпорядженням завідувача кафедрою з найбільш досвідчених та авторитетних викладачів кафедри (загальний склад від 3 до 5 осіб). Секретарем комісії призначається працівник кафедри.

Голова апеляційної комісії проводить засідання у тижневий термін з моменту виходу розпорядження завідувача кафедрою про створення апеляційної комісії. Про дату та час проведення засідання заявник попереджається щонайменше за два дні. Якщо заявник не з'являється на засідання апеляційної комісії, питання розглядається за його відсутності.

У випадку необхідності отримання додаткової уточнюючої інформації, засідання апеляційної комісії може проводитись у кілька етапів з розривом не більше трьох робочих днів.

Сумніви, що виникають у членів апеляційної комісії, трактуються на користь особи, робота якої розглядається апеляційною комісією.

За результатами засідання апеляційна комісія формує висновки, які підписує голова апеляційної комісії, її члени та заявник, зазначаючи «З висновками апеляційної комісії погоджуюсь». Висновки апеляційної комісії щодо академічного плагіату (компіляцій) у творах студентів зберігаються на кафедрі.

Процедура розгляду апеляції проводиться відповідно «Положення про апеляційну комісію Донбаської державної машинобудівної академії» (<http://surl.li/tpqdk>)

ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Технічні засоби автоматизації: методичні вказівки до самостійної роботи (для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології») / сост. О.В. Разживін - Краматорськ: ДДМА, 2022. - 52 с.
2. Технічні засоби автоматизації: методичні вказівки до виконання контрольної роботи (для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології») / сост. О.В. Разживін. - Краматорськ: ДДМА, 2022. - 52 с.
3. Механізми виконавчі електричні однооборотні МЕО [Електронний ресурс]. – Режим доступу: <http://ap.net.ua/meo.html>.
4. Механізми виконавчі електричні однооборотні ВЕО, VOGEZ [Електронний ресурс]. – Режим доступу: <http://www.et.kh.ua/products/vogez/VES.html>.
5. Звіти у сфері науки і техніки. Структура та правила оформлювання. ДСТУ 3008-2015. Національний стандарт України. Режим доступу: http://www.knmu.kharkov.ua/attachments/3659_3008-2015.PDF.
6. Робоча навчальна програма дисципліни «Проектування систем автоматизації на базі ПЛК» для студентів першого (бакалаврського) рівня за ОПІ «Автоматизація та комп'ютерно-інтегровані технології» спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка», 2024 - 30 с.
7. Сердюк О.О. Вимоги стандартів до оформлення проектів. Методичні вказівки. / Уклад. О.О. Сердюк. Краматорськ: ДДМА, 2021 - 44 с.
8. **Сердюк, О. О.** Проектування систем автоматизації Simatic : навчальний посібник для студентів вищих навчальних закладів / О. О. Сердюк, О. В. Разживін. – Краматорськ : ДДМА, 2012. – 208 с.
9. **Клименко, Б. В.** Електричні апарати. Електромеханічна апаратура комутації, керування та захисту. Загальний курс :навчальний посібник. – Харків : Точка, 2012. – 340 с.
10. «Про вищу освіту» : Закон України від 01.07.2014 р. № № 1556-УП. URL: <http://zakon5.rada.gov.ua/laws/show/1556-18> (дата звернення: 01.07.2023).
11. «Про авторське право та суміжні права» : Закон України від 23.12.1993 № 3792-ХІІ. URL: <http://zakon3.rada.gov.ua/laws/show/3792-12> (дата звернення: 01.07.2023).
12. **Бурштинський, М. В.** Давачі / М. В. Бурштинський, М. В. Хай, Б. М. Харчишин. – 2-е вид. доповн. – Львів : ТзОВ «Простір М», 2014. – 202 с.
13. Датчик температури [Електронний ресурс]. – Режим доступу: <http://www.aotera.com.ua/pages/ru/sensors.html?gclid=CP3FIJyO9NACFQKkGAodt3UKHA>.
14. Датчик температури [Електронний ресурс]. – Режим доступу: <http://www2.emersonprocess.com/ruRU/brands/Metran/products/Temperature/Pages/>

index.aspx

15. Датчики тиску. Датчик тиску газу, рідини. [Електронний ресурс]. – Режим доступу: <http://www.ukrteh.kiev.ua/page/text/category=pressure>.

16. Датчики тиску [Електронний ресурс]. – Режим доступу: <http://www2.emersonprocess.com/ua-ru/brands/Metran/products/Pressure/Pages/index.aspx>

17. Витратоміри. Лічильники [Електронний ресурс]. – Режим доступу: <http://www2.emersonprocess.com/ruRU/brands/Metran/products/Flow/Pages/index.aspx>

18. Технічні засоби для проектування систем автоматизації : навчальний посібник / О. В. Разживін, О. В. Суботін. – Краматорськ : ЦТРІ «Друкарський дім», 2017. – 129 с

19. Інтерактивний каталог продуктів Siemens IA&DT. Техніка автоматизації Siemens [Електронний ресурс]. – Режим доступу: <http://w5.siemens.com/web/ru/iadt/ia/FAPA/Pages/as.aspx>

20. Simatic. Програмування з допомогою Step 5 V5.5. Посібник. Редакція 01/2010, A5E00261405-1.

21. Програмовані контролери Simatic S7 [Електронний ресурс]. – Режим доступу: <http://w5.siemens.com/web/ua/ua/iadt/ia/FAPA/PLC/Pages/PLC.aspx>.

22. Автоматизація у промисловості. Каталог Siemens CA01 2022 [Електронний ресурс]. – Режим доступу: <http://w5.siemens.com/web/ua/ua/iadt/Pages/ca01.aspx..>

23. Методичні вказівки до комп'ютерного практикуму по дисципліні "Цифрові системи керування й обробки інформації" (для студентів спеціальності 151 "Автоматизація та комп'ютерно-інтегровані технології») / Уклад. О. О. Сердюк. - Краматорськ: ДДМА, 2018 – 87 с

ДОДАТОК А

Приблизна тематика курсових проєктів

- 1 Проєкт модернізації системи керування токарського верстата із ЧПУ типу 16Д20Ф3.
- 2 Проєкт модернізації системи керування натягом смуги для прокатного стану 700.
- 3 Проєкт модернізації локальної обчислювальної мережі для сервісного центру ПрАТ НКМЗ.
- 4 Проєкт автоматизованої виконавчої системи для термічного виробництва ПрАТ СКМЗ.
- 5 Проєкт інформаційної системи безпеки для заводу «Славтяжмаш»
- 6 Розробка автоматизованої системи керування приводом скіпового підйомника.
- 7 Розробка автоматизованої системи керування приводами подачі каретки й роликів бесцентрового токарського верстата.
- 8 Розробка системи автоматичного заправлення дроту для електроерозійного агрегату.
- 9 Проєкт автоматизації термічної печі для загартування й відпустки металовиробів в умовах Дружковського заводу металовиробів.
- 10 Розробка підсистеми керування приводом повороту платформи крокуючого екскаватора ЕШ 11/70.
- 11 Розробка інформаційної системи обліку енергетичних ресурсів заводу ПрАТ ЕМСС.
- 12 Проєкт автоматизованої системи діагностики вузлів редукторів для мостового крана.

Додаток Б
Зразок бланка завдання

Донбаська державна машинобудівна академія
Кафедра автоматизації виробничих процесів

ЗАВДАННЯ

на розробку курсового проекту по дисципліні
«Цифрові системи керування й обробки інформації»

Студент гр. _____

(прізвище, ім'я, по батькові)

Тема курсового проекту: _____

Вихідні дані: _____

Зміст пояснювальної записки

Зміст графічної частини

Завдання видав:

_____ (прізвище, ініціали)

Дата видачі завдання _____

Строк здачі проекту _____

Міністерство освіти й науки України
Донбаська державна машинобудівна академія
Кафедра автоматизації виробничих процесів

КУРСОВИЙ ПРОЕКТ

по дисципліні
«Цифрові системи керування й обробки інформації»

На тему: _____

Виконав студент гр. _____
(прізвище й ініціали)

Керівник _____
(прізвище й ініціали)

Краматорськ-Тернопіль 2024

Додаток Г
Зразок оформлення реферату

РЕФЕРАТ

Розрахунково-пояснювальна записка містить 39 с., 15 рис., 5 таблиць, 3 додатка, 19 джерел. Дослідницька частина містить 13 с.

Об'єкт проектування – інтегрована система керування термічною вертикальною піччю В-1 заводу НКМЗ.

Ціль роботи – підвищення ефективності термічного виробництва.

У проекті виконаний критичний аналіз існуючої системи інформаційного забезпечення ділянки термічного виробництва, сформульовані технічні вимоги до розроблюваної системи керування тепловим режимом печі, а також завдання проекту.

Для підвищення ефективності керування термічним виробництвом із застосуванням мови UML розроблені моделі програмного забезпечення. Для реалізації програми керування термічним режимом печі застосована мова РНР. Виконані розробки дозволяють інтегрувати окремі ділянки термічного виробництва в єдину інформаційну систему із централізованим керуванням.

Для реалізації проекту здійснені розрахунки локальної мережі й зроблений вибір мережного устаткування. Проведене моделювання мережі дозволяє зробити вивід про її працездатність.

ПІЧ ТЕРМІЧНА ВЕРТИКАЛЬНА, АВТОМАТИЗАЦІЯ ТЕРМІЧНОЇ ПЕЧІ, ВИКОНАВЧА СИСТЕМА ВИРОБНИЦТВА, ДІАГРАМА UML, СИСТЕМА ТЕПЛООВОГО РЕЖИМУ ПЕЧІ.

ДОДАТОК Д. ЗАЯВА ТА ПРОТОКОЛ ПЕРЕВІРКИ РОБОТИ НА
ПЛАГІАТ

ЗАЯВА

щодо самостійності виконання курсової роботи (проекту),
Я, _____, студента __ курсу, групи АВП**-*, ФМ,
Донбаської державної машинобудівної академії, заявляю:

моя курсова робота на тему:

« _____

_____ .»

представлена для захисту у комісію, виконана самостійно і в ній не міститься елементів плагіату. Всі запозичення з друкованих та електронних джерел, а також із захищених раніше дослідницьких робіт, кандидатських і докторських дисертацій мають відповідні посилання. Я ознайомлений(а) з діючим положенням ДДМА, згідно з яким виявлення плагіату є підставою для відмови в допуску письмової роботи до захисту та застосування дисциплінарних заходів.

Підписуючи цей документ, надаю згоду Власнику бази даних (кафедра АВП) на збір, обробку та використання моїх матеріалів кваліфікаційної роботи з метою перевірки матеріалу в системі та зберіганні роботи в репозитарії кафедри (академії).

Підтверджую, що з Положенням «Протидія плагіату» кафедри ознайомлений.

Матеріали автора також можуть бути використані Власником або третіми особами з метою та на умовах, що визначені відповідно до ст. 32 Закону України «Про вищу освіту» та до Закону України «Про захист персональних даних».

Власник зобов'язується використовувати матеріали справедливо і законно відповідно до законодавства. У разі зміни мети обробки матеріалів Власник має повідомити про це суб'єкта і отримати згоду на обробку його матеріалів у відповідності зі зміненою метою.

Дата:

Підпис:

ЕКСПЕРТНИЙ ВИСНОВОК
про перевірку на наявність академічного плагіату у випускній курсовій
роботі студента

Комісія з виявлення та запобігання академічного плагіату,
перевіривши курсову роботу студента _____

_____.

гр. АВПХХ-Х на тему:

« _____

_____»

прийшла до висновку:

*Рівень оригінальності твору станом на « ____ » _____ 202X р. є
задовільним (_____)*

Зав. кафедри _____ ПІБ.

Відповідальна особа по кафедрі _____ ПІБ.

Керівник роботи _____ ПІБ

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ БАЗОВОГО ПРОЦЕСУ ТА ПОСТАНОВКА ЗАВДАНЬ ПРОЕКТУВАННЯ.....	7
2 ВИЗНАЧЕННЯ ОСНОВНИХ ПАРАМЕТРІВ СИСТЕМИ УПРАВЛІННЯ ОБЛАДНАННЯМ.....	13
2.1 Розрахунки тривалості робочого циклу програми.....	13
2.2 Розрахунки точності й швидкодії вимірювальних каналів	19
2.3 Розрахунки динамічних параметрів виконавчих пристроїв.....	21
3 РОЗРОБКА -МОДЕЛЕЙ БІЗНЕС-ПРОЦЕСІВ	23
3.1 Розробка моделей IDEF0.....	24
3.2 Розробка інфологічної моделі предметної області	25
3.3 Розробка специфікації функцій системи.....	29
3.4 Моделювання програмного додатка.....	32
4 ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ОБЛАДНАННЯМ.....	40
5 РЕКОМЕНДАЦІЇ З МОДЕЛЮВАННЯ СИСТЕМИ.....	45
6 РЕКОМЕНДАЦІЇ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЕКТУ	49
7 ОФОРМЛЕННЯ ДОКУМЕНТАЦІЇ ПРОЕКТУ	52
7.1 Вимоги до оформлення текстових документів.....	52
7.2 Вимоги до оформлення графічної частини.....	56
8 ПОРЯДОК ВИКОНАННЯ ТА ЗАХИСТУ КУРСОВОГО ПРОЄКТУ	58
8.1 Етапи виконання та захисту	58
8.2 Права та обов'язки керівника, здобувача вищої освіти.....	59
8.3 Застереження щодо академічної доброчесності.....	59
8.4 Регламенти і процедури виявлення порушень вимог академічної доброчесності та наслідки такого виявлення	64
8.5 Критерії оцінювання курсового проєкту.....	67
8.6 Порядок оскарження результатів оцінювання курсового проєкту.....	70
ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ	72
Додаток А. Приблизна тематика курсових проєктів	74
Додаток Б. Зразок бланка завдання	75
Додаток В. Зразок титульного аркуша розрахунково-пояснювальної записки.....	76
Додаток Г. Зразок оформлення реферату.....	77
Додаток Д. Заява та протокол перевірки роботи на плагіат	79

