

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет магістерської підготовки

Кафедра інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Моделі автоматизованого проектування округлених профілів обробки матеріалів

Виконав: студент 2 курсу групи МІС-19
спеціальності 122 Комп'ютерні науки

Хамраєв Езіз

Керівник: професор каф. АСМНС, д. т. н.,
доцент Великодний Станіслав Сергійович

Консультант _____

Рецензент: к. ф.-м. н., доцент
Буяджи Василь Володимирович

Одеса 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет магістерської підготовки

Кафедра інформаційних технологій

Рівень вищої освіти магістр

спеціальність 122 – комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

“26” жовтня 2020 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Хамраєву Езізу

(прізвище, ім'я, по батькові)

1. Тема роботи Моделі автоматизованого проектування округлених профілів обробки матеріалів

керівник роботи Великодний Станіслав Сергійович, д. т. н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «16» жовтня 2020 р. № 194 «С»

2. Строк подання студентом роботи 07 грудня 2020 р.

3. Вихідні дані до роботи: CAD / CAM / CAE-системи; системи програмного відтворення рухів; методи інтерполяції; методи апроксимації; методи і принципи автоматизованого проектування; методи організації графічного діалогу і побудови інтерфейсу користувача із використанням можливостей програмної системи MATLAB; діаграмна методологія UML 2.5; CASE-інструментарій Rational Rose 2013

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

вступ; аналіз вихідних даних та технологій; аналіз спеціалізованих САПР; порівняльний техніко-економічний аналіз запропонованих проектів і вибраного аналога; моделювання архітектури CAD-модуля за допомогою методології UML; практична реалізація CAD-модуля, що проектується; алгоритм інтерполяції дуги окружності для СПВР; програмна реалізація CAD-модуля; вказівки користувачеві програмного засобу; організація виводу результатів розрахунків; висновки; додаток у вигляді програмного коду

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
графічний матеріал виконаний у вигляді презентації

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв


7. Дата видачі завдання "26" жовтня 2020 р.

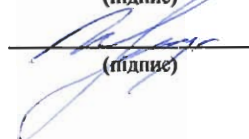
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Отримання завдання на роботу	26.10.2020	60	Задовільно
2	Аналіз вихідних даних	30.10.2020	60	Задовільно
3	Виконання аналітичного розділу	08.11.2020	60	Задовільно
4	Моделювання архітектури CAD-модуля за допомогою методології UML	14.11.2020	60	Задовільно
5	Розробка діаграм у CASE-засобі	18.11.2020	60	Задовільно
6	Проходження рубіжної атестації	19.11.2020	60	Задовільно
7	Практична реалізація CAD-модуля	28.11.2020	60	Задовільно
8	Розробка програмного засобу	02.12.2020	60	Задовільно
9	Розробка інтерфейсу програмного засобу	04.12.2020	60	Задовільно
10	Тестування програмного забезпечення	05.12.2020	60	Задовільно
11	Формування висновків до роботи	06.12.2020	60	Задовільно
12	Подання роботи на кафедру	07.12.2020	—	
13	Перевірка роботи на плагіат	08.12.2020	—	
14	Складання графічної частини	12.12.2020	—	
15	Підготовка доповіді на захист	15.12.2020	—	
16	Проходження рецензування	16.12.2020	—	
Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)				Задовільно

Студент

Керівник роботи


(підпис)


(підпис)

Хамраєв Е.

(прізвище та ініціали)

Великодний С. С.

(прізвище та ініціали)

АНОТАЦІЯ

«Моделі автоматизованого проектування округлених профілів обробки матеріалів». Хамраєв Езіз.

Мета роботи – розробка CAD-модуля, що дозволяє виконувати чітко поставлені технологічно орієнтовані виробничі завдання проектування траєкторії руху робочого органу СПВР. Об'єкт дослідження – системи автоматизованого проектування машинобудівного профілю. Предмет дослідження – моделі та методи проектування складних геометричних контурів.

Методи дослідження: методи інтерполяції, методи апроксимації, методи і принципи автоматизованого проектування, методи організації графічного діалогу і побудови інтерфейсу користувача із використанням можливостей програмної системи MATLAB.

Результати, отримані автором, полягають у такому: обрано метод інтерполяції дугами кола за рахунок однокритеріального аналізу і встановлення конкретного вигляду такої дуги, яка найбільш точно відтворює задану форму оброблюваної ділянки, що відрізняється від існуючих зменшенням часу опису складних контурів обробки, завдяки скороченню декомпозиції профілю на графічні примітиви; набув подальшого розвитку метод формування вихідних даних для пристрою керування СПВР, що дозволило скоротити кількість кадрів керуючої програми та, відповідно, зменшити ймовірність виникнення помилок зчитування.

Результатом розробки – став програмний продукт (у вигляді вихідного коду), готовий для установки та підключення до широкого спектру сучасних комп'ютерних операційних систем.

Магістерська кваліфікаційна робота містить: 78 с.; 14 рис., 3 табл., 1 додаток, 28 використаних літературних джерел.

КЛЮЧОВІ СЛОВА: система автоматизованого проектування, система програмного відтворення рухів, каркасне моделювання, програмний засіб, керуюча програма, графічний інтерфейс користувача, діаграма, модуль.

SUMMARY

“Models for Computer-Aided Design of Rounded Metalworking Profiles”.
Hamrayev Eziz.

Purpose – development of CAD-module, which allows a clear set of technologically oriented production planning task trajectory CNC working body. Object of study – computer aided design engineering profile. The subject of research – models and methods of design of complex geometric contours.

Methods: methods of interpolation methods for approximation methods of analytic geometry and calculus mathematics to model outlines complicated spatial forms, methods and principles of computer-aided design, methods of dialogue and building graphical user interface capabilities using the program system MATLAB.

Results obtained by the author are as follows: choose a method of interpolation circle arcs through criteria analysis and establish a specific form of such curves that best reproduces a given form of cultivated land, which is different from the current decrease time description of complex circuits processing by reducing the decomposition profile of graphics primitives; was developed a method for forming output control device CNC, thereby reducing the number of staff administering the program and thus reduce the likelihood of errors reading.

The result of the development – has become the software (in source code form), ready for installation and connection to a wide range of modern computer operating systems.

Master's thesis contains: 78 p., 14 fig., 3 tab., 1 applications, 28 used literature sources.

KEYWORDS: computer-aided design, system software playing motions, frame simulation, software, tool control program, graphical user interface, diagram, module

ЗМІСТ

Перелік умовних познач	8
Вступ	10
1 Аналіз CAD / CAM / CAE технологій, що застосовуються для проектування технологічної підготовки виробництва	15
1.1 Аналіз спеціалізованих електронних ресурсів	15
1.2 Аналіз спеціалізованих САПР	16
1.3 Порівняльний техніко-економічний аналіз запропонованих проектів і вибраного аналога	24
1.4 Виявлені проблеми, визначені як завдання магістерської роботи	27
2 Моделювання архітектури CAD-модуля за допомогою методології UML	30
2.1 Побудова діаграми варіантів використання	31
2.2 Побудова діаграми послідовності	33
2.3 Побудова кооперативної діаграми	36
2.4 Побудова діаграми активності	38
2.5 Побудова діаграми компонентів	40
2.6 Побудова діаграми розміщення	42
2.7 Висновки за розділом	44
3 Практична реалізація CAD-модуля, що проектується	46
3.1 Алгоритм інтерполяції дуги окружності для СПВР	46
3.1.1 Структура алгоритму	46
3.1.2 Опис роботи алгоритму	48
3.1.3 Складання ідентифікаторів ПЗ	51
3.2 Програмна реалізація CAD-модуля	56
3.2.1 Загальні принципи роботи програмного засобу	56
3.2.2 Написання програмного засобу «ARC»	56
3.2.3 Створення графічного інтерфейсу користувача ПЗ «ARC»	58
3.3 Вказівки користувачеві програмного засобу «ARC»	59

3.4 Організація виводу результатів розрахунків.....	61
3.5 Висновки до третього розділу	63
Висновки	66
Перелік джерел посилання	69
Додаток А. Фрагменти вихідного тексту (лістінга коду) CAD-модуля	72

ПЕРЕЛІК УМОВНИХ ПОЗНАК

АСК – абсолютна система координат.

ВР – вид руху.

ВСК – відносна система координат.

ДА – діаграми активності.

ДВВ – діаграма варіантів використання.

ДК – діаграма класів.

ДКм – діаграма компонентів.

ДП – діаграма послідовності.

ДР – діаграма розміщення.

ДС – діаграма станів.

ІД – ідентифікатор.

КД – кооперативна діаграма.

КП – керуюча програма.

ММ – математична модель.

ОПЗ – область припустимих значень.

ОС – операційна система.

ПЕОМ – персональна електронна обчислювальна машина.

ПК – персональний комп'ютер.

ПЗ – програмний засіб.

ППП – пакет прикладних програм.

ПР – промисловий робот.

РО – робочий орган.

САПР – система автоматизованого проектування.

СК – система координат.

СПВР – система програмного відтворення рухів.

СЧПК – система числового програмного керування.

ЧПК – числове програмне керування.

CAD – Computer-Aided Designing.

CAE – Computer-Aided Engineering.

CAM – Computer-Aided Manufacturing.

CAPP – Computer-Aided Process Planning.

CNC – Computer Numerical Control.

UG – UniGraphics.

UML – Unified Modeling Language

ВСТУП

На сучасному етапі становлення України, важливим напрямком науково-технічного розвитку – є розробка нових технологій автоматизованого проектування із підвищеними вимогами відносно ступеня складності конструкцій, скорочення термінів виконання замовлень, підсилення конкурентної боротьби на ринку. Ці складові примушують здійснювати роботи на найсучаснішому технічному рівні, тому сьогодні конструювання, аналіз і технологічна підготовка виробництва виконується із використанням, так званих, «важких» систем автоматизованого проектування (САПР), вибір та організація роботи яких, проходить у рамках єдиної концепції розвитку засобів проектування та технологічної підготовки виробництв, що спроможні вирішувати задачі, які були поставлені замовником перед виконавцем.

У більшості випадків, елементи, що проектуються, та для яких необхідно вирішувати задачу формоутворення – це тонкостінні конструкції, що складаються із великого набору криволінійних поверхонь, з'єднання яких повинно виконуватись дуже гладко та мати плавну зміну другої похідної. У місцях цих з'єднань доводиться вирішувати досить нетипові задачі сполучення за визначеним законом поверхонь, тому до процесу проектування висунуті найвищі вимоги відносно САПР.

Значні теоретичні досягнення та суттєві практичні результати, в галузі моделювання кривих ліній та складних геометричних поверхонь, отримані завдяки зусиллям провідних вітчизняних вчених: Ю.І. Бадаєва, С.М. Ковальова, Л.М. Куценка, В.Є. Михайленка, В.М. Найдиша, В.О. Надолінного, А.В. Павлова, О.Л. Підгірного, А.М. Подкоритова, М.М. Рижова, І.А. Скидана, П.В. Філіппова та їх наукових шкіл; а також закордонних вчених: І. Адамса (I. Adams), П. Без'є (P. Bézier), В. Гілоя (W. Gilloy), У. Ньюмена (W. Newmen), Д. Роджерса (D. Rogers), Р. Різенфорда (R. Rizenford), А. Сазерленда (I. Sutherland), І. Фергюссона (E. Fergusson), А. Фореста (A. Forest), Д. Швейкертта (J. Schweickert) та ін.

Розробці й дослідженню методів і засобів розв'язання задач формоутворення присвячені роботи А.А. Воронова, В.Д. Байкова, С.М. Вашкевича, І.А. Вульфсона, В.В. Карибського, А.А. Левіна, В.А. Ратмірова, В.Л. Сосонкіна, О.М. Калачьова, К.В. Филиповича та деяких інших вітчизняних та іноземних авторів.

Технологічний процес формоутворення складного геометричного контуру, що виконується системою програмного відтворення рухів (СПВР) (серед яких промислові роботи (ПР), автоматичні маніпулятори, верстати із ЧПК, графобудівники та інші системи програмного відтворення рухів й складні динамічні об'єкти, що виконують зварювальні, фарбувальні, механообробні роботи, а також різання, паяння, нанесення покриттів, розкрій та інші технологічні операції, які виконуються електричною дугою, лазерним променем, плазмою, пульверизатором, різцем тощо), передбачає проходження робочим органом (РО) заданої просторової (чи площинної) траєкторії з необхідною точністю.

Досить часто виникає потреба у здійсненні програмного руху РО СПВР за контурами, котрі найдоцільніше описати пласкою дугою кола, тому, що лінійна інтерполяція не витримує підвищених вимог складного формоутворення щодо зменшення точності проведених розрахунків, значної кількості похибок різного роду, великої кількості розбивань на графічні примітиви та інших факторів, які унеможливають виконання конкретного високоточного технологічного завдання, що пов'язане із якістю відтворення контуру.

Для реалізації найвищої точності опису округлених профілів елементів, що проектуються, програмне забезпечення сучасних САПР повинно містити ефективні алгоритми автоматичної інтерполяції, що легко стикуються із пристроями керування СПВР, котрі, відповідно, реалізують просторовий рух РО за заданими траєкторіями. Це призводить до необхідності вдосконалення існуючих методів інтерполяції та формування нових методів проектування складних геометричних контурів рухів РО СПВР, які б забезпечили високу

швидкодію та надали б простий аналітичний підхід до побудови програмних траєкторій у рамках цих методів.

Мета роботи і завдання дослідження. Із розгляду поставленої проблеми, можна сформулювати мету магістерської роботи, яка полягає в розробці САПР-модуля, що дозволяє виконувати чітко поставлені технологічно орієнтовані виробничі завдання проектування траєкторії руху робочого органу СПВР. Стосовно до методології CAD / CAM / CAE систем, подібний модуль можна класифікувати як CAD (Computer-Aided Design – комп'ютерне автоматизоване проектування), відповідно, модуль буде мати назву CAD-модуль.

Виконання означеної мети роботи може бути досягнуто після вирішення конкретно встановлених завдань:

- виконати огляд існуючих CAD / CAM / CAE технологій, для визначення кола аналогів, та провести їхній техніко-економічний аналіз;
- змоделювати архітектуру CAD-модуля за допомогою методології каркасного моделювання UML;
- створити інтерактивну графічну систему одержання зображення контуру, інтерфейс якої повною мірою відповідав би сучасним пристроям уведення інформації до СПВР та до якого звикли конструктори й технологи.

Результатом розробки буде – програмний продукт (у вигляді вихідного коду), готовий для установки та підключення до широкого спектру сучасних комп'ютерних операційних систем.

Об'єкт дослідження – системи автоматизованого проектування машинобудівного профілю.

Предмет дослідження – моделі та методи проектування округлених геометричних контурів.

Методи дослідження: методи інтерполяції, методи апроксимації, методи аналітичної геометрії та обчислювальної математики для моделювання контурів складної просторової форми, методи і принципи автоматизованого

проектування, методи організації графічного діалогу і побудови інтерфейсу користувача із використанням можливостей програмної системи MATLAB.

Елементи наукової новизни отриманих результатів. Розроблено і сформовано моделі проектування складних геометричних контурів рухів РО СПВР, які забезпечують високу швидкодію та надають простий аналітичний підхід до побудови програмних траєкторій у рамках цих методів.

Наукові результати, що отримані автором, полягають у такому:

- обрано метод інтерполяції дугами кола за рахунок однокритеріального аналізу і встановлення конкретного вигляду такої дуги, яка найбільш точно відтворює задану форму оброблюваної ділянки, що відрізняється від існуючих зменшенням часу опису складних контурів обробки, завдяки скороченню декомпозиції профілю на графічні примітиви;
- набув подальшого розвитку метод формування вихідних даних для пристрою керування СПВР, що дозволило скоротити кількість кадрів керуючої програми та, відповідно, зменшити ймовірність виникнення помилок зчитування.

Практичне значення отриманих результатів. Результати досліджень реалізовано у вигляді розробленого програмно-методичного комплексу, який створює основу для розробки інструментальних засобів автоматизованого проектування заданого технологічного процесу, що виконується СПВР разом із САПР.

Методи та моделі розв'язання завдань проектування реалізовані програмно. У процесі тестування та випробувань вони показали свою працездатність і ефективність на конкретних прикладах розв'язання завдань автоматизованого проектування та інтерполяції заданих складних геометричних контурів.

Створено інтерактивну графічну систему одержання зображення формоутворення, яка відрізняється від існуючих інтегрованих САПР-додатків побудови геометричних моделей контурів тим, що траєкторія руху РО інтер-

полюється відразу дугою кола (без наближень), що гарантує підвищену точність розрахунків та скорочення термінів проектування.

Розроблено інструкції та рекомендації щодо застосування необхідного математичного та програмного забезпечення. При практичному застосуванні програмних засобів (ПЗ), вихідні дані можуть бути задані в ході розв'язання зворотної задачі кінематики або визначені інженером-технологом аналітично. Додатково у ПЗ передбачено необхідні коментарі відносно їх роботи, перевірки коректності завдання вихідних даних та умов фізичної реалізації сформованої траєкторії руху РО СПВР.

1 АНАЛІЗ CAD / CAM / CAE ТЕХНОЛОГІЙ, ЩО ЗАСТОСОВУЮТЬСЯ ДЛЯ ПРОЕКТУВАННЯ ТЕХНОЛОГІЧНОЇ ПІДГОТОВКИ ВИРОБНИЦТВА

1.1 Аналіз спеціалізованих електронних ресурсів

В Інтернеті існує чимало сайтів, що належать всесвітньо відомим журналам. Частина з них цілком публікує обрані статті своїх випусків і, звичайно, анонси нових номерів. Крім того, існують і електронні журнали (тобто не мають друкованої форми), такі журнали вимагають оплати за передплату та доступ до повноформатним версіями статей. Нарешті, є третя група сайтів – вони розміщують статті повністю, але їх публікацію оплачує рекламодавець.

У будь-якому випадку, фахівцеві в області CAD/CAM/CAE навряд чи доведеться потримати в руках свіжий номер будь-якого іноземного журналу. Тому найбільш допитлива «просунута» частина проектувальників і користувачів САПР тримає руку на пульсі прогресу, видряпуючи по крихтах freeware-статті з зарубіжних сайтів.

Зробимо короткий огляд найбільш популярних електронних ресурсів в області CAD/CAM, з якими було проведено знайомство в період переддипломної практики.

Сайт MetalWorking Digest Online виправдовує назву – дайджест матеріалів про обробку металів. Є вільні для читання матеріали, потрібна реєстрація, є безкоштовна розсилка. На сайті солідна бібліотека. Об'єднані традиційні функції – пошук роботи, пошук CAD/CAM-софта за категоріями, CAD/CAM-лінки, відомості про CAD/CAM-асоціаціях. Є презентація CAD/CAM/CAE-виставок на поточний рік. Сайт має непоганий дизайн.

Електронна версія журналу Cutting Tool Engineering на якому доступні анонси, вибрані статті, архів статей. Все інше традиційно – лінки, класифікатори, події в галузі, підписка (розсилка новин).

Електронна версія журналу «Сучасні верстати». На сайті є можливість пошуку по введенню ключового слова та отримання списку необхідних статей, яких дійсно багато, вони повноформатні і добре ілюстровано. На жаль, більшість статей замовні, тобто містять приховану рекламу. Тому висновок в кожній статті один – софт автора завжди найкращий. Тим не менш, я вважаю цей сайт одним з найкращих за якістю і рівнем статей з CAD/CAM-тематики.

Електронна версія журналу «Технології автоматизованого проектування» – це повноформатні ілюстровані статті у великій кількості й асортименті, підписка на розсилку новин, крім того, на сайті маса новин, анонсів, прес-релізів. Вважаю сайт кращим CAM/CAE-порталом, в тому числі містить масу корисної інформації для читання.

Електронний журнал CADENCE CHANNEL – безкоштовний щомісячний електронний журнал. Повноформатні ілюстровані статті з тематики CAD/CAM. Сайт дуже корисний і рекомендований для читання проектувальникам будь-якого рівня.

1.2 Аналіз спеціалізованих САПР

У ході аналізу спеціалізованих САПР, схожих за реалізованими функціями з модулем, що розробляється (САПР-аналоги), то серед них виконувалося знайомство з поданими зразками.

САПР-ЧПК – універсальна система автоматизованого проектування, що охоплює підготовку керуючих програм (КП) для 2 і 2.5 координатної обробки для всіх моделей верстатів з ЧПК і обробних центрів (включаючи токарну і фрезерну обробку, плазморезку і електроеррозію) на базі ПЕОМ [1]¹⁾.

САПР-ЧПК забезпечує 100% сумісність з попередніми версіями – САП-ЄС, САП-СМ4, САП-ПК. Дані про геометрію і технології виготовлення

¹⁾ [1] Филиппович К.В. Некоторые аспекты настройки пользовательских предпочтений в САПР-ЧПУ/2005. Пермь: ООО «Евразия Лимитед», 2005. 42 с.

деталі вводяться в ПЕОМ за допомогою мовного опису або інтерактивного графічного введення, а також у режимі «наскрізного проектування», з використанням файлів-креслень, підготовлених в САД/САМ і САМ-системах [2] ¹⁾.

Наявний макроапарат препроцесора [3] ²⁾ і бібліотека макропроцедури дозволяють програмувати гравірування шрифтів (пуансон, матриця, трафарет), виготовлення інструменту, зубчастих коліс, а також елементи тривимірної обробки, що складаються з математично заданих поверхонь (сфера, циліндр, конус, еліпсоїд тощо). Також реалізований автоматичний розрахунок еквідістанти за схемами «Спіраль», «Зигзаг» при фрезеруванні кишень будь-якої форми з перешкодами, «Петля» – при точінні, підрізуванні, розточенні, включаючи тіньові зони. Забезпечено графічний контроль програмованої геометрії і траєкторії руху інструменту, тривимірне твердотільне моделювання процесу обробки деталі.

Багатокоординатних (до 5-ти координат) інваріантний постпроцесор САПР-ЧПК налаштовується на інформаційний код комплексу «верстат-система ЧПК» за допомогою паспорта (анкети), що описує індивідуальні характеристики конкретного комплексу [4] ³⁾.

Паспорт складається протягом 2 – 8 годин технологом, вивчив відповідний комплекс і методику підготовки КП. Інваріантний постпроцесор дозволяє на 100% задовольнити вимогам методики підготовки КП, що додається до конкретного комплексу, включаючи моделювання стандартних і індивідуальних верстатних підпрограм, автоціклов з графічним відображенням на дисплеї (принтері / плотері) траєкторії руху інструменту і розрахунком машин-

¹⁾ [2] Великодний С. С. Методологические основы реинжиниринга систем автоматизированного проектирования. *Междун. науч. журн. «Управляющие системы и машины»*. 2014. № 2. С. 39–43.

²⁾ [3] Schultz D.F. What's Wrong With Postprocessors? *Numerical Control Computer Sciences*. 1998. №1. P. 11–15.

³⁾ [4] Великодний С. С. Методи реінжинірингу програмних систем. *Технологии приборостроения*. 2014. Спец. вып. С. 65–68.

ного часу. Макромова постпроцесора дозволяє технологу самостійно нарощувати і змінювати функції інваріантного постпроцесора без звернення до розробників або програмістам [5]¹⁾.

Система САПР-ЧПК різних версій в період 1989 – 2010 впроваджена більш ніж на 370 підприємствах СНД (близько 1800 інсталяцій) [6]²⁾.

Модуль GRAFCAM призначений для проектування КП до верстатів з ЧПК, використовуючи принцип графічної побудови і редагування контуру деталі, траєкторії руху інструменту і технологічних команд [7], [8]³⁾.

Модуль GRAFCAM орієнтований на користувачів-технологів і може бути налаштований на будь-який рівень їх кваліфікації. Модуль GRAFCAM дозволяє будувати на екрані дисплея елементи роз'єднаної геометрії і об'єднувати їх в контури (рис. 1).

На цій основі технолог проектує траєкторію руху інструменту [9]⁴⁾. Кожен новий елемент шляху руху інструменту негайно відмальовується на екрані дисплея. Для проектування роз'єднаної геометрії використовуються 60 способів завдання точок, прямих, кіл. Після того, як користувач повідомить параметри нового елемента геометрії, він малюється на екрані дисплея. Можливий режим малювання поруч з геометричним елементом його ідентифікатора.

¹⁾ [5] Великодний С. С., Тимофєєва О. С., Зайцева-Великодна С. С., Нямцу К. Є. Порівняльний аналіз властивостей відкритого, вільного та комерційного програмного забезпечення. *Інформаційні технології та комп'ютерна інженерія*. 2018. № 1 (41). С. 21–27.

²⁾ [6] Werner J. The Case For Verifying And Optimizing Tool Paths. Irvine: CGTech, 2003. 58 p.

³⁾ [7] Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання поведінкової частини. *Вісник Кременчуцького національного університету ім. Михайла Остроградського*. 2019. Вип. 2 (115). С. 117–126. (кат. «Б») DOI: 10.30929/1995-0519.2019.2.117-126.

[8] Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання структурної частини. *Вісник Кременчуцького національного університету ім. Михайла Остроградського*. 2019. Вип. 3 (116). С. 130–139. (кат. «Б») DOI: 10.30929/1995-0519.2019.3.130-139.

⁴⁾ [9] Калачев О. Н. Диалоговая система KON7 для расчета размерных цепей при механообработке. Ярославль: ЯГТУ, 2005. 93 с.

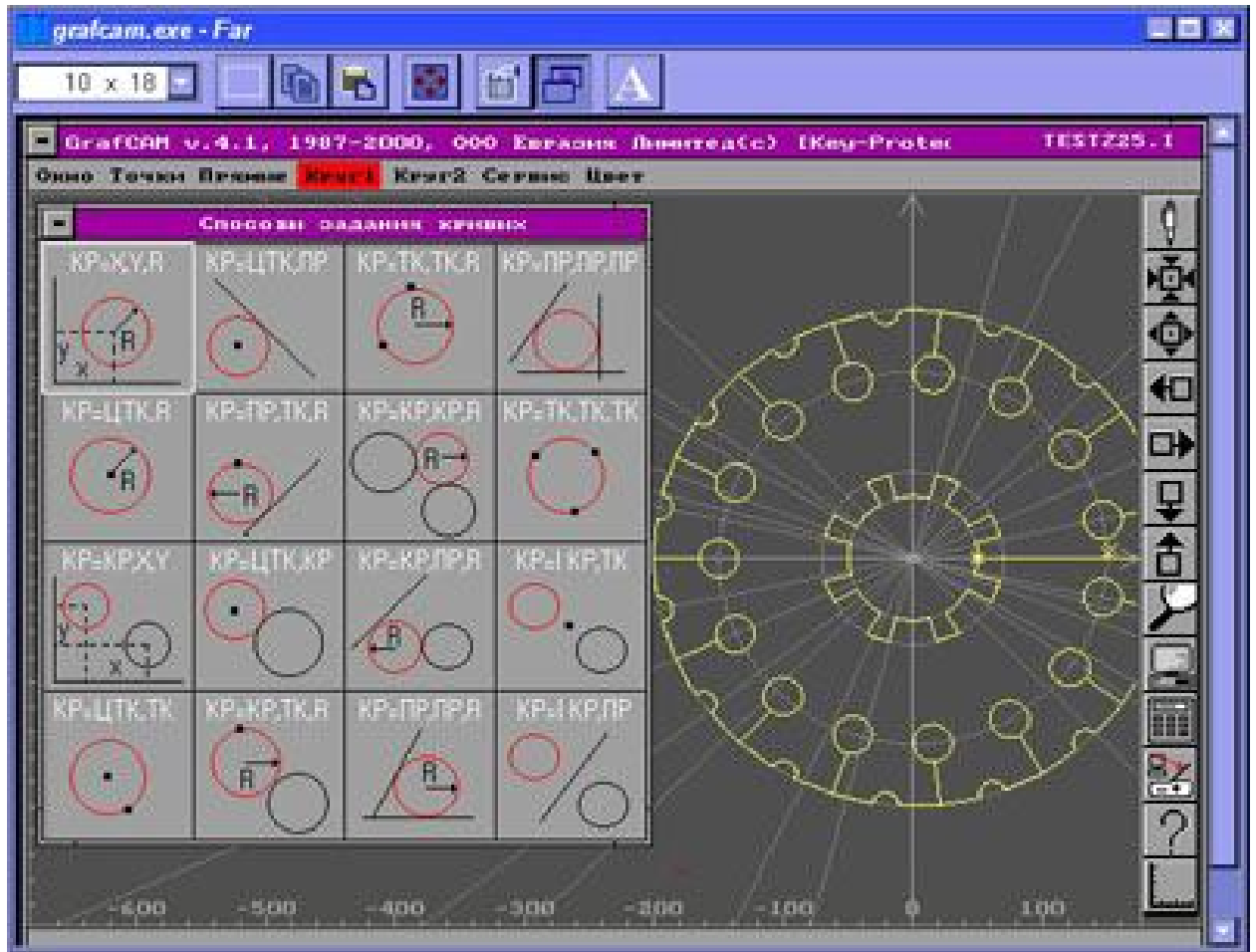


Рисунок 1 – Елементи роз'єднаної геометрії, об'єднані в контури за допомогою модуля GRAFCAM

Кожен геометричний елемент повинен мати статус (основний або допоміжний) [10]¹⁾. Статус призначається самим користувачем. GRAFCAM може малювати на екрані дисплея тільки основні елементи, що зручно при проектуванні траєкторії руху інструменту. Модуль GRAFCAM включає механізм «м'якого» видалення геометричних елементів, надаючи кошти для швидкого відновлення у разі помилки. Користувач може робити вимірювання

¹⁾ [10] Калачёв О.Н., Рехтер А.Д. Моделирование размеров механообработки в среде AutoCAD 200x на основе использования приложения GRAKON7. САПР и графика. 2002. №2. С. 100–104.

відстаней між геометричними елементами, а також вимірювати кути між прямими або центральні кути між точками. Для цього потрібно вибрати на екрані дисплея два елементи, використовуючи маніпулятор «миша» або клавіатуру.

GRAFCAM ідеальний для графічного формування траєкторії руху інструменту. Для цього користувач вказує мишею елементи роз'єднаної геометрії, пов'язуючи їх у ланцюг. Прямі перетворюються у відрізки, криві – в дуги (рис. 2).

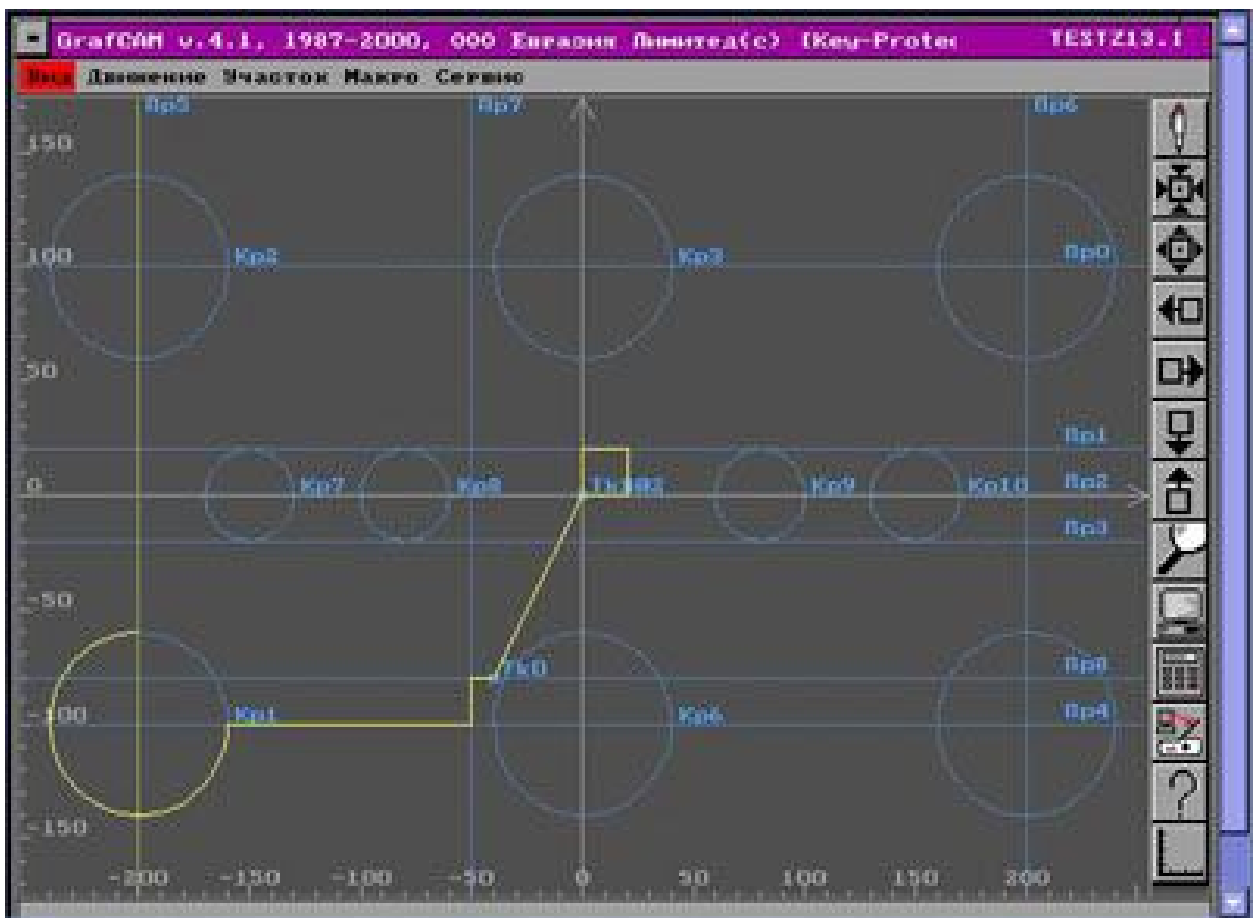


Рисунок 2 – Формування графічних примітивів траєкторії руху інструменту в модулі GRAFCAM

У будь-який момент часу можна ввести технологічну команду управління верстатом. GRAFCAM підтримує формування контурів (ділянок) і аф-

фiні операції над ними (зсув, поворот та ін.). Забезпечується промальовування, видалення, зміна номера і коректування конкретного контуру [11] ¹⁾.

Модуль GRAFCAM підтримує бібліотеки параметризованих елементів (макропроцедури). У будь-який момент часу користувач може доповнити контур деталі або траєкторію руху інструменту за допомогою звернення до підходящої макропроцедури. GRAFCAM допускає поповнення як бібліотеки, так і меню макропроцедури самим користувачем [12] ²⁾.

У будь-який момент користувачеві доступні різноманітні видові операції: (взяття вікна, відкат назад, зміна габаритів креслення, зменшення/збільшення масштабу, стиснення креслення по одній з осей, зсув креслення в одному з 8 напрямків).

Відмітною принципом GRAFCAM є автоматичне формування тексту програм на вхідній мові своєї базової системи, тобто САПР-ЧПК. Крім того, GRAFCAM є 100% графічним інтерпретатором і відладчиком текстів будь-яких програм, створених за допомогою систем САП-32, САП-ЄС, САП-СМ4, САП-ПК, САПР-ЧПК.

Досвідчені користувачі можуть редагувати і доповнювати свої програми, тобто елементи роз'єднаної геометрії, контури, траєкторію руху інструменту і технологічні команди, використовуючи вбудований в GRAFCAM текстовий редактор. Після редагування тексту програми всі зміни негайно графічно відображаються на екрані дисплея. Після закінчення сеансу роботи GRAFCAM записує автоматично сформований або відредагований текст програми (на вхідній мові САПР-ЧПК) в обрану користувачем директорію. Після цього подальшого налагодження можлива або за допомогою GRAFCAM, або

¹⁾ [11] Velykodniy S. Reengineering of open software system of 3D modeling BRL-CAD. *Innovative Technologies and Scientific Solutions for Industries*. 2019. No 3 (9), P. 62–71. (кат. «Б») DOI: <https://doi.org/10.30837/2522-9818.2019.9.062>.

²⁾ [12] Velykodniy S. S. Analysis and synthesis of the results of complex experimental research on reengineering of open CAD systems. *Applied Aspects of Information Technology*. 2019. Vol. 2. No 3. P. 186–205. (кат. «Б») DOI: 10.15276/aait.03.2019.2.

в інтегрованій оболонці САПР-ЧПК. В останньому випадку генерується текст КП для конкретного комплексу «верстат – система ЧПК» [13] ¹⁾.

GRAFCAM взаємопов'язаний з модулем ACAD-SAP для забезпечення режиму наскрізного проектування. Інтерфейсом виступає текст програми, сформована модулем ACAD-SAP і містить елементи роз'єднаної геометрії і контури (ділянки) [14] ²⁾.

Модуль GRAFCAM впроваджений у 1992 – 2010 р.р. на 260 підприємствах Росії та СНД (більше 310 інсталяцій).

Пакет макропроцедури «евольвенти» призначений для розрахунку КП для верстатів з ЧПК токарної, фрезерної, електроерозійної груп для виготовлення прямозубих циліндричних зубчастих коліс з евольвентним профілем зуба [15] ³⁾.

Пакет макропроцедури дозволяє вирізати прямозубі циліндричні зубчасті колеса зовнішнього і внутрішнього зачеплення з евольвентним профілем зуба на електро-ерозійних верстатах, верстатах з лазерною різкою, фрезерних верстатах, з використанням корекції на радіус інструменту. На верстатах токарної групи можлива обробка кінцевого фасонного ріжучого інструменту з евольвентним профілем для подальшого виготовлення конкретного зубчастого колеса на універсальному обладнанні оснащеному УДГ. У токарній обробці передбачена можливість автоматичної вибірки металу за схемою «Петля» з програмованим припуском під чистову обробку [16] ⁴⁾.

¹⁾ [13] Великодний С. С. Проблема реінжиніринга видів забезпечення систем автоматизованого проектування. *Управляющие системы и машины*. 2014. № 1. С. 57–61, 76.

²⁾ [14] Трухнин Н. М., Филиппович К. В. Макрорасширения в GrafCAM – путь к наращиванию функциональности. Пермь: ООО «Евразия Лимитед», 2006. 72 с.

³⁾ [15] Филиппович К. В., Попович И. ToolStore – среда для ведения библиотеки инструментов в верификаторе CNC-Verify системы САПР-ЧПУ/2005. Пермь: ООО «Евразия Лимитед», 2005. 36 с.

⁴⁾ [16] Грунина Е.В., Калачёв О.Н. Методика проектирования в CAD/CAM Cimatron UP для гравирования профиля на юбилейной медали. Ярославль: ЯГТУ, 2008. 58 с.

У процесі розрахунку КП на етапі постпроцесування передбачена можливість розрахунку розміру між роликками для контролю евольвентного профілю зуба, виготовленого за розрахованої КП, також передбачена діагностика помилкових ситуацій у виборі діаметра ролика.

Після розрахунку КП виконується графічна верифікація контуру – червоний колір і траєкторії руху інструменту – зелений колір (рис. 3).

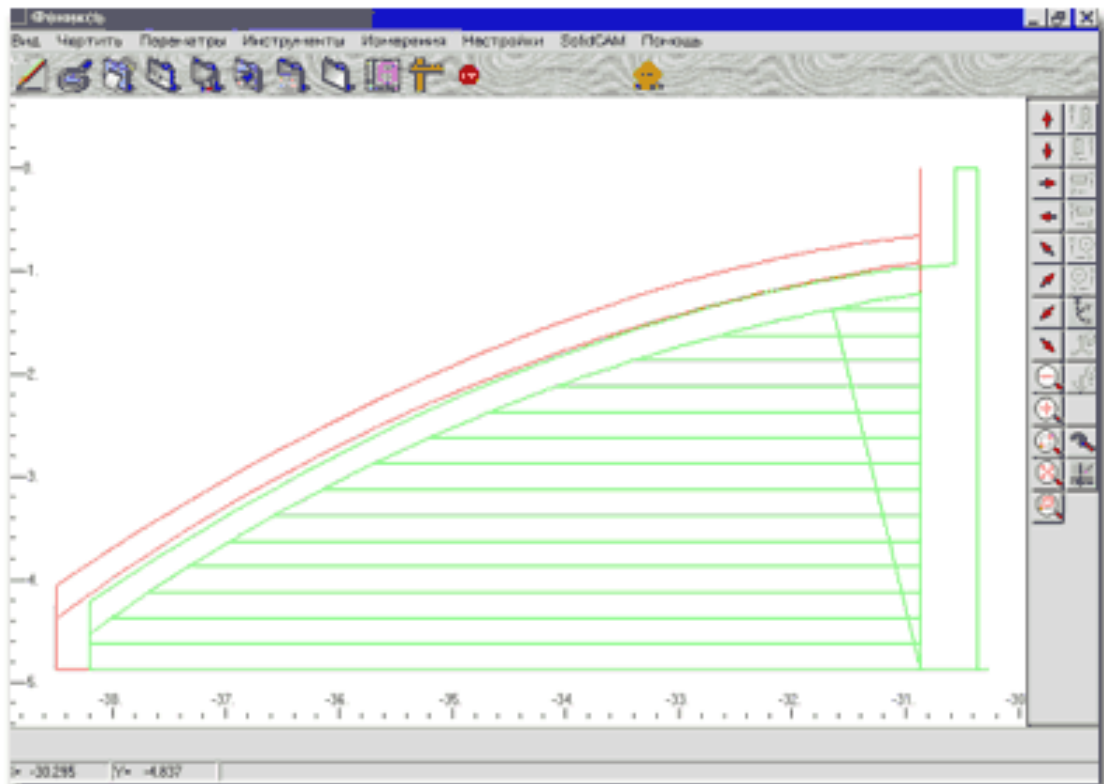


Рисунок 3 – Графічне представлення контуру обробки за допомогою пакету макропроцедури «евольвенти»

У результаті розрахунку КП із застосуванням сплайна з лінійно-кругової інтерполяцією отримуємо кадри КП з дуг кіл, плавно сполучених між собою [17]¹⁾.

¹⁾ [17] Великодний С. С. Метод представлення оцінки реінжинірингу програмних систем за допомогою проектних коефіцієнтів. *Сучасний стан наукових досліджень та технологій в промисловості*. 2019. №1 (7). С. 34–42. (кат. «Б») DOI: <https://doi.org/10.30837/2522-9818.2019.7.034>.

В якості переваг слід відзначити зафіксований факт того, що використання пакету «евольвенти» на заводі спеціалізованого інструменту і технологічної оснастки «Гомсільмаш» (Білорусь) при виготовленні кілець-калібрів для контролю зубчастих коліс показало, відхилення профілю зуба від аналогічних, виготовлених за КП, розрахованих із застосуванням систем PEPS-2 і PRO / Engineer не перевищує 0,001 мм, а розбіжність розрахованого розміру між роликами з фактичним не перевищує 0,003 мм.

1.3 Порівняльний техніко-економічний аналіз запропонованих проектів і вибраного аналога

При проведенні даного аналізу, перш за все, були проаналізовані техніко-економічні показники аналогів розроблюваної системи, в першу чергу ціна ліцензії на існуючі САД-системи, здатні виконувати ті ж обов'язкові функції, наявність яких – є головною вимогою замовника до виконавців (розробників) [18]¹⁾.

Спочатку, замовник мав ряд специфічних вимог (можливість генерації ISO-коду, підтримка G-функцій, редагування керуючих програм з підтримкою еквідистантно візуалізації відредагованого фрагмента та ін), проте, головною вимогою, яку в один момент відкинуло можливості використання взагалі всіх імпортованих САПР – стало: сполучення САПР з ЧПК-верстатами вітчизняного (радянського) «парку».

Тут необхідно зазначити, той факт, що хоча сфера інформаційних технологій дуже динамічна і швидко розвивається, але сучасна промисловість України, яка перебувала під постійним тиском кризових складових, використовує (на машинобудівних заводах) обладнання (верстати, оброблювальні

¹⁾ [18] Великодний С. С., Тимофеева О. С., Зайцева-Великодна С. С. Метод розрахунку показників оцінки проекту при виконанні реінжинірингу програмних систем. *Радіоелектроніка, інформатика, управління*. 2018. № 4. С. 135–142. DOI: 10.15588/1607-3274-2018-4-13.

центри та ін) ще часів СРСР. Не вдаючись у глибокий економічний аналіз цього чинника, скажемо лише наступне: ЧПК-обладнання 20-річної давності цілком може конкурувати за своїми точностними і якісними характеристикам, однак вимагає кілька специфічні стандарти введення технологічних команд (просто кажучи керуючі програми – КП), які, якщо і відомі західним розробникам, то вельми поверхнево. Більш того, жодна імпортна САПР (CAD/CAM/CAE) не підтримує ні трансляцію, ні редагування даного коду.

Напрошується питання: чому ж під наявні західні CAD/CAM/CAE не використовувати західні верстати та обладнання. Відповідь досить прозаїчний: сучасний обробний центр Siemens (Німеччина) або Fanuc (Японія) має вартість понад 1 млн. € – це не кажучи вже про те, що до нього ж необхідно придбати ліцензію на програмне забезпечення. Висновок очевидний: сучасний стан машинобудівної промисловості Україна – не дозволяє оновити існуючий парк верстатів з ЧПК.

За тієї умови, що динаміка вартості обладнання не втішна – зниження не планується ні в якому варіанті, тут чітких варіантів вирішення цього питання взагалі не передбачає. Тому, що точнісні та якісні характеристики в металообробці за допомогою вітчизняного обладнання – ані трохи не гірша за західні аналоги, а значить – немає необхідності в кардинальній заміні, що веде до гігантським фінансовим вкладенням, які все одно не підйомні вітчизняному виробнику, не дивлячись ні на яку дотацію держави і підтримку вітчизняного виробника.

Для довідки: переобладнання одного середнього заводу, що складається з приблизно 50 (статистичні дані) обробних центрів обійдеться в 50 – 60 млн. € (!) І це не враховуючи витрати на монтаж і перенавчання операторів і наладчиків (ще 20 млн. €) і вимушений простий устаткування і, отже, неможливості отримання прибутку від виконаних замовлень.

У зв'язку з вищевикладеним економічним обґрунтуванням, вибір виробника САПР редукували до масштабів СНД, а якщо точніше, то тільки декілька розробників САПР розробляють програмні продукти, що реалізують

вище зазначені функції для верстатів з ЧПК. Серед них: ТЕХТРАН (м. Уфа, РФ), АДЕМ (м. Єкатеринбург, РФ), ІНТЕР-МЕХ (м. Мінськ, Білорусь), West Labs (м. Харків, Україна). Інших аналогів просто не існує. Табл. 1 відображає фінансово-економічні показники даних аналогів, які вдалося визначити за допомогою маркетингового дослідження.

Таблиця 1 – Фінансово-економічні показники САПР-аналогів (ціни наведені у доларах США)

Розробник САПР	Продукт	Вартість продукту (1 роб. м.)	Мін. обсяг ліцензійних місць	Термін ліцензії	Підсумкова вартість
ТЕХТРАН	САПР-ЧПК	400	10	2 р.	4000
АДЕМ	АДЕМ	299	8	18 міс. (1,5 р.)	2392
ІНТЕР-МЕХ	CNC-IM	360	12	3 р.	4320
West Labs	West Labs 5.2	480	5	15 міс.	2400

Аналізуючи цю таблицю, до однозначних висновків прийти досить складно. По-перше: у другого і четвертого розробника модуль вартості найбільш низький, однак і терміни ліцензії найбільш короткі, в той же час третій виробник, не дивлячись на найвищу вартість – надає найбільший термін ліцензії. Тут ще цікавий той факт, що жоден розробник не надає інформацію щодо вартості продовження ліцензії, мотивуючи це неможливістю розрахунку трудомісткості на забезпечення сумісності з з'явилася (після закінчення терміну ліцензії) новою операційною системою. По-друге: виходячи з того, що нашому замовнику достатньо одного (максимум двох) ліцензійних копій (за кількістю робочих місць), а розробники жорстко обмежують мінімальний

обсяг ліцензійної покупки, гостро постає питання про недовикористання інших копій, а це навіть при самому лояльному ставленні четвертого розробника (див. табл. 1) – більше половини.

З наведеного короткого техніко-економічного аналізу відразу напрашується висновок, що в умовах і можливостях, у яких існує наш реальний замовник – єдиною альтернативою залишається розробка нашого програмного продукту, і що особливо важливо – без мінімального обмеження закупівельної ліцензії та чіткими ціновими показниками продовження.

1.4 Виявлені проблеми, визначені як завдання магістерської роботи

У процесі виконання науково-дослідницької роботи студента, складовою частиною якої було знайомство з передовими САПР-аналогами. Як показав практичний досвід, більшість технологів-користувачів таких систем зазнають об'єктивні проблеми з процесом генерації КП, особливо на вітчизняне обладнання. До числа основних проблем відносяться:

- відсутність в CAD/CAM-системі готового постпроцесора на конкретний комплекс верстат-система з ЧПК, а також непомірно висока вартість його індивідуальної розробки (400 - 1000 €);
- вартість універсальних постпроцесорів зарубіжних CAD/CAM-систем лежить в межах 3000 – 10000 €, що порівняно із ціною самої CAD/CAM-системи – це змушує підприємства замовляти розробку або купувати ліцензії на використання тільки кілька постпроцесорів; КП на решті верстатний парк розраховуються більше дешевими вітчизняними САМ-системами;
- універсальні постпроцесори зарубіжних CAD/CAM-систем дуже важко адаптуються до морально застарілому вітчизняному устаткуванню з ЧПК випуску 70 – 80-х років – як правило, це адаптація виконується дилерами при інтенсивній допомозі зарубіжних партнерів;

- кожна нова CAD/CAM-система вимагає індивідуальний постпроцесор на одне і теж обладнання з ЧПК, тим самим підприємство невинувато платить двічі за одне і теж;
- важко уявити середньостатистичного технолога-програміста, який розробляє постпроцесор за допомогою закордонного універсального постпроцесора – сьогодні локалізована тільки UG / Post. На решту – немає ні російської документації, ні перекладу help-файлів, остаточна настройка постпроцесора на обладнання з ЧПК передбачає програмування на спеціалізованому мовою високого рівня, що під силу тільки досвідченому програмісту-проектувальнику;
- замовні постпроцесори, як правило, важко піддаються самостійній подальшій коригуванню, у результаті багато технологи досі застосовують порочну практику ручного коригування КП.

Проаналізувавши можливості вирішення перелічених проблем і зорієнтувавши вектор їх рішення у бік з'єднання будь-яких CAD/CAM-систем з єдиним універсальним інваріантним постпроцесорів (CAPP), була визначена стратегічна мета магістерської роботи проектування, а саме: розробити CAD-модуль, що дозволяє замінити переважну більшість постпроцесорів для різних CAD/CAM-систем, які експлуатуються на будь-якому машинобудівному підприємстві, тобто один і той же паспорт на конкретний комплекс «верстат-система з ЧПК» можна буде застосувати для різних CAD / CAM-систем, наприклад, одночасно для Pro / Engineer і Duct.

Плановані переваги впровадження в машинобудівне виробництво розроблюваного модуля полягають у наступному:

- у разі, якщо знадобиться тонка настройка на особливості обладнання з ЧПК – використовується потужний апарат макромови складання паспортів ARTIPR;
- розробка нового постпроцесора можлива силами самого технолога приблизно за 4 – 8 годин, причому без послуг як заводського про-

граміста, так і тим більше без необхідності оплати програмісту-дилеру зарубіжної CAD/CAM-системи;

- ARTIPR має інтерфейси до більшості популярних CAD/CAM-систем, а також легко та швидко адаптує його налагодниками до нових CAD/CAM-продуктів.

Організаційне обґрунтування розроблюваного проекту, полягає у наданні класифікаційної оцінки різновидам проекту, використовуючи критерії:

- клас – мультипроект – дана розробка може підійти до різних ЧПК-верстатів, різних підприємств, однак використовують один і той же формат управління та кодування;
- тип – технічний – розробка відноситься до області технічного машинобудування;
- вид – комбінований – розробка, що включає навчальну (підготовка фахівців безпосередньо на підприємстві або за цільовим договором у навчальному закладі) та інноваційну (нова САПР, запроваджена на існуючому виробництві) компоненту;
- тривалість – середньостроковий – проект орієнтований на 2 – 3 роки (життєвий цикл операційної системи – ОС), а далі необхідна доробка під сумісність з новою ОС);
- складність проектної розробки – близько 2-х місяців, розміри (разом з додатками) – близько 100 Мб;
- рівень – корпоративний – працює в масштабах одного підприємства з локальної виробничої мережею, без мережевої підтримки оновлень.

Основними етапами виконання робіт по впровадженню CAD-модуля, будуть:

- планування;
- документування;
- проектування;
- програмування;
- тестування.

2 МОДЕЛЮВАННЯ АРХІТЕКТУРИ САД-МОДУЛЯ ЗА ДОПОМОГОЮ МЕТОДОЛОГІЇ UML

Методологія UML призначена для створення «каркасу» майбутнього програмного засобу, тобто всебічного аналізу як архітектури ПЗ, так і дослідження різноманітних зв'язків, що встановлюються та структур, що проектуються [19] ¹⁾.

Загальна перевага, що притаманна нотації UML – це всебічний аналіз проектуємого ПЗ, починаючи від завантаження окремих об'єктів до компонентної взаємодії [20] ²⁾.

В нотації UML використовуються декілька різновидів діаграм, а саме:

- діаграми варіантів використання (ДВВ);
- діаграми класів (ДК);
- діаграми послідовності (ДП);
- кооперативні діаграми (КД);
- діаграми станів (ДС);
- діаграми активності (ДА);
- діаграми компонентів (ДКм)
- діаграми розміщення (ДР).

Взагалі, при проектуванні архітектури ПЗ дуже рідко використовуються всі перелічені діаграми в рамках одного проекту тому, що спектр їх використання дуже широкий і, як кажуть, підійде на всі випадки життя. В поданій пояснювальній записці розглянуто найбільш типові діаграми нотації UML, які найбільш вичерпно описують усі можливі зв'язки та компоненти САД-модуля, що розроблюється.

¹⁾ [19] Боггс У., Боггс М., UML. Rational Rose: Секреты эффективного проектирования сопровод. объектно-ориентиров. прилож. Москва: «Лори», 2000. 570 с.

²⁾ [20] Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя (Сер. программистов). Москва: ДМК, 2000. 432 с.

2.1 Побудова діаграми варіантів використання

ДВВ модуля, що розроблюється являє собою розгалужену структуру із множинними зв'язками, які визначають [21]¹⁾:

- обладнання, на якому може бути встановлено цей модуль;
- виробничі процеси, які можуть виконуватись на наведеному обладнанні;
- інструменти, за допомогою яких виконуються виробничі процеси.

Причому, для нашої ДВВ характерні множинні відношення, тобто, наприклад: на одному обладнанні можна виконувати різні виробничі процеси та навпаки: один виробничий процес може виконуватись різним обладнанням, а якщо ще й прийняти до уваги, що виробничі процеси можуть виконуватись різними інструментами, які можуть встановлюватись на різному обладнанні, то без проектування наглядної структури ДВВ САД-модуля – дуже складно врахувати усі можливі зв'язки, які повинні бути описані в програмному засобі.

Розглянемо детальніше спроектовану ДВВ (рис. 4).

Інтерпретація поданої ДВВ виглядає наступним чином:

У якості актора – виступає спроектована система (САД-модуль), яка може бути використана системою програмного відтворення рухів (СПВР), що становить на поданій ДВВ – абстрактну діючу особу, яка через зв'язки «узагальнення», може бути умовно розділена на:

- верстати із ЧПК;
- промислові роботи;
- динамічні об'єкти.

¹⁾ [21] Гома Х., UML. Проектирование систем реального времени, параллельных и распределенных приложений. Москва: ДМК Пресс, 2002. 704 с.

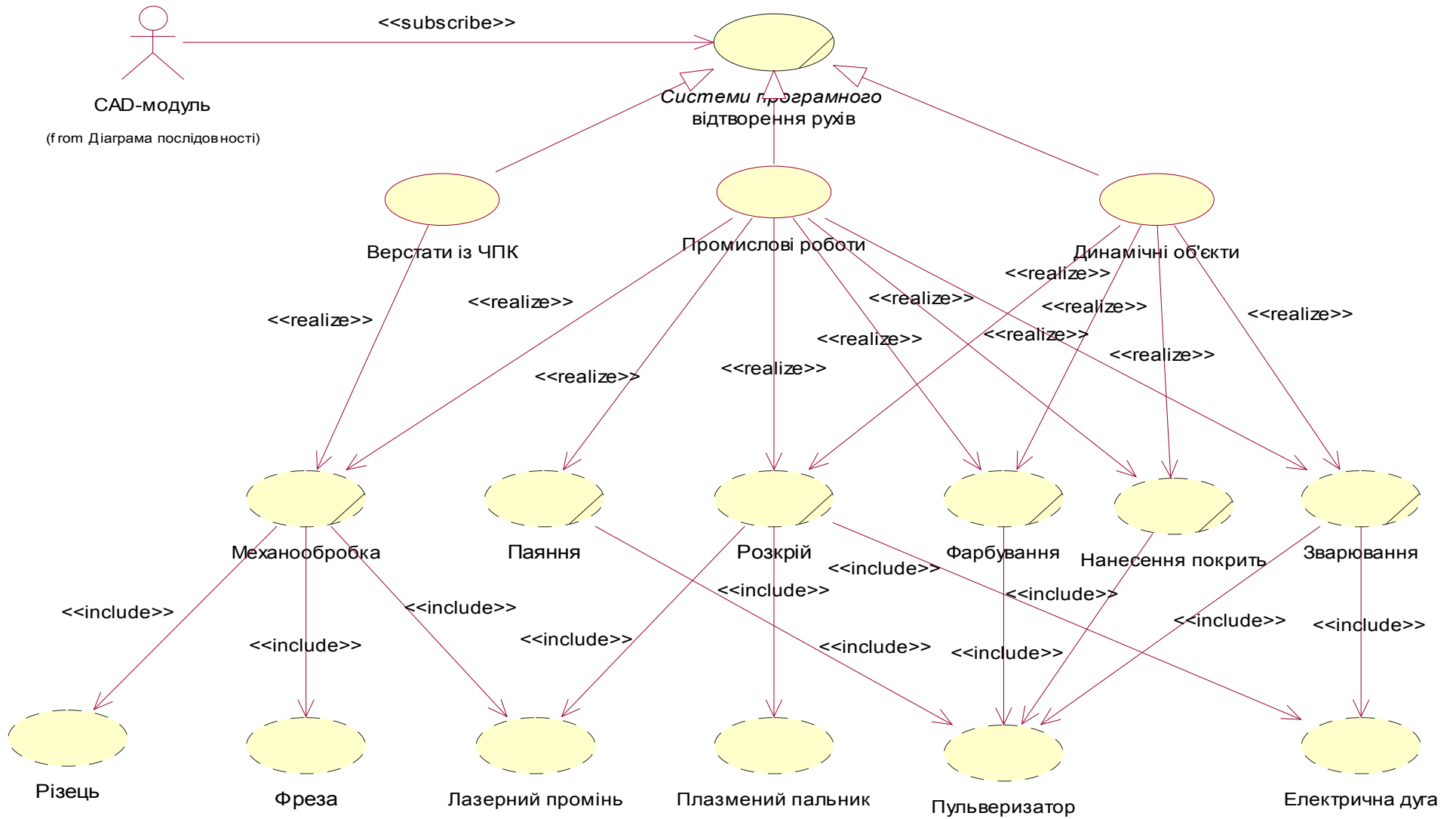


Рисунок 4 – ДВВ CAD-модуля

В свою чергу, кожні з них виконують наступні виробничі процеси:

- верстати із ЧПК – механообробку;
- промислові роботи – механообробку, паяння, розкрій, фарбування, нанесення покриттів, зварювання;
- динамічні об'єкти – розкрій, фарбування, нанесення покриттів, зварювання.

Кожен виробничий процес може виконуватись такими інструментами:

- механообробка – різцем, фрезою, лазерним променем;
- паяння, фарбування, нанесення покриттів – пульверизатором;
- розкрій – лазерним променем, плазменним пальником, електричною дугою;
- зварювання – пульверизатором, електричною дугою.

Таким чином, після погодження усіх перелічених зв'язків та варіантів використання, отримаємо спроектовану ДВВ із визначеною топологією, яка наведена на рис. 4.

2.2 Побудова діаграми послідовності

ДП використовуються для того, щоб показати послідовність дій у системі, що проектується, щоб розробникам була зрозуміла послідовність кроків, що виконуються [22]¹⁾.

Наша ДП відображає крім послідовності конкретного технологічного процесу також типи повідомлень, що передаються. Розроблену ДП CAD-модуля наведено нижче.

В якості об'єктів, між якими передаються повідомлення, які показують послідовність дій – виступають відомий нам з ДВВ CAD-модуль та нові актори, а саме:

¹⁾ [22] Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование. Москва: ДМК Пресс, 2001. 176 с.

- оператор;
- виробнича мережа;
- СЧПК.

Кожного об'єкта (актора) позначено лінією життя (штрихова риска на рис. 5), між якими встановлюються повідомлення визначеного типу, у випадку нашої ДП – це такі повідомлення:

- прості;
- синхронні (з обов'язковою відповіддю від об'єкта до якого це повідомлення направлене);
- асинхронні;
- повідомлення з лімітом часу;
- рефлексивні (повідомлення вказують на те, що даний процес відбувається в межах об'єкту, який його направив).

Стосовно до нашої ДП – відбувається така послідовність дій між об'єктами (стосовно до реальної роботи із майбутнім модулем CAD) (нумерація нижче – відповідає нумерації повідомлень на рис. 5):

- визначення технологічної задачі (повідомлення з лімітом часу);
- обробка задачі (рефлексивне повідомлення);
- увід координат (повідомлення з лімітом часу);
- перевірка умов фізичної реалізуємості (рефлексивне повідомлення);
- завдання швидкості (повідомлення з лімітом часу);
- перевірка припустимого діапазону швидкостей (рефлексивне повідомлення);
- завдання дискретності (повідомлення з лімітом часу);
- перевірка можливості відпрацювання (рефлексивне повідомлення);
- розрахунок керуючої програми (рефлексивне повідомлення);
- передача керуючої програми (асинхронне повідомлення);
- передача керуючої програми наступному об'єкту (синхронне повідомлення);



Рисунок 5 – ДП CAD-модуля

- відповідь про початок або відмову (асинхронне повідомлення);
- відповідь про початок або відмову наступному об'єкту (асинхронне повідомлення); обробка (рефлексивне повідомлення);
- сигналізація про завершення (асинхронне повідомлення);
- сигналізація про завершення наступному об'єкту (асинхронне повідомлення);
- сигналізація про завершення наступному об'єкту (асинхронне повідомлення).

Таким чином, після побудови усіх перелічених послідовних дій, отримуємо спроектовану ДП, яка наведена на рис. 5.

2.3 Побудова кооперативної діаграми

КД використовуються для аналізу топології завантаженості системи, що проектується [23]¹⁾, тобто визначаються об'єкти найбільш завантажені зв'язками з метою встановлення їх дублювання (резервування) у випадку виходу їх зі строю.

КД будується на основі побудованої ДП із збереженням зв'язків між об'єктами та, відповідно, номери – показують послідовність повідомлень (по аналогії з ДП) [24]²⁾.

Побудована КД приведена на рис. 6. Виходячи з її топології можна зробити висновки щодо найбільшого завантаження об'єкту, яким є САД-модуль – це є очікуваним тому, що саме модуль є предметом дипломного проектування.

¹⁾ [23] Леоненков А. В. Самоучитель UML. Санкт-Петербург: БХВ-Петербург, 2002. 304 с.

²⁾ [24] Ларман К. Применение UML и шаблонов проектирования: уч. пос. Москва: Вильямс, 2001. 496 с.

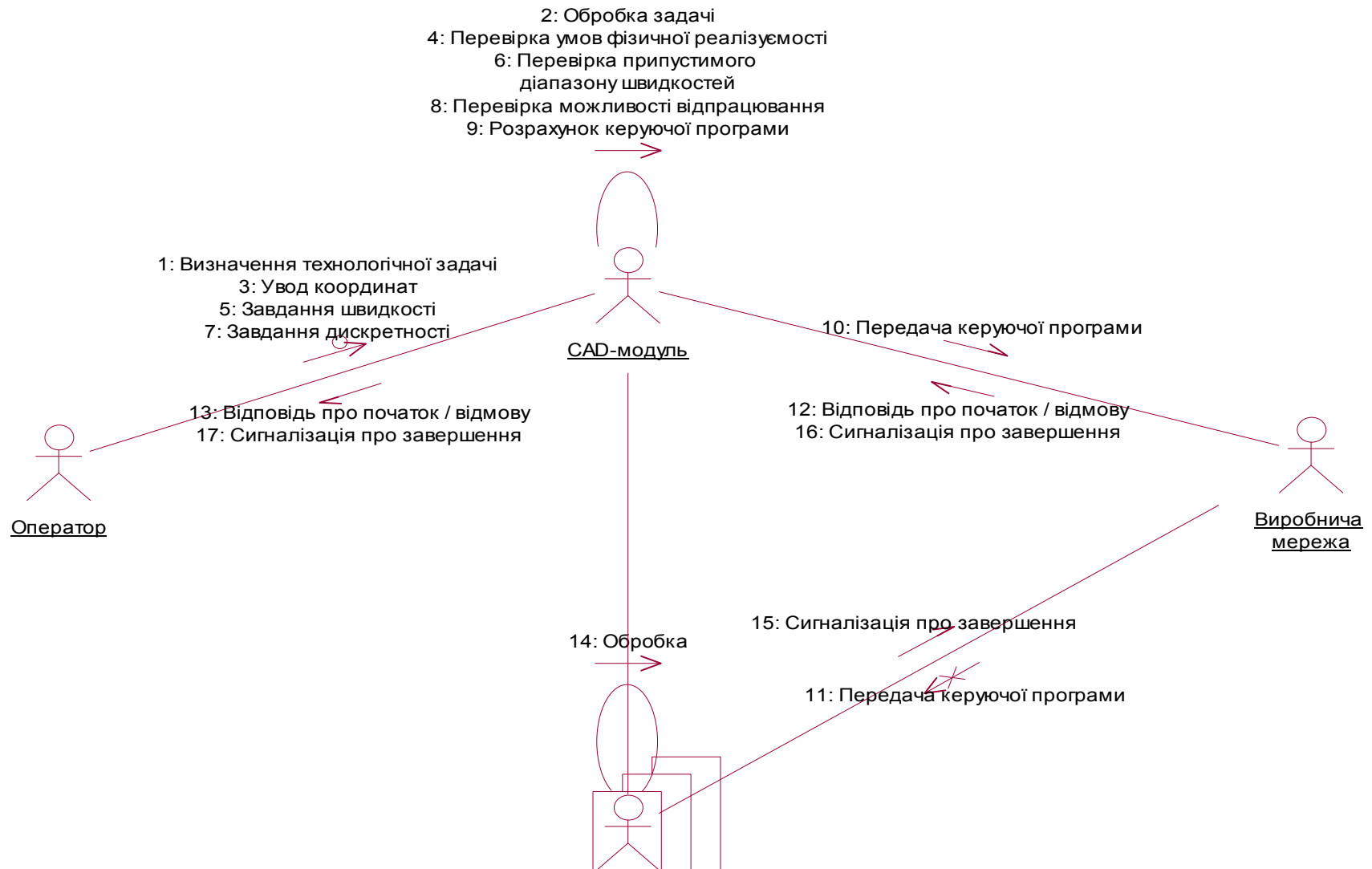


Рисунок 6 – КД CAD-модуля

2.4 Побудова діаграми активності

ДА визначає активність об'єктів, що проектуються, на конкретних періодах функціонування програмного засобу [25]¹⁾. В ДА існує поняття доріжка активності, яка відокремлює об'єкти між собою, крім того – існують різноманітні блоки, що вказують – у яких саме випадках даний об'єкт буде активним.

ДА, що розроблено для САД-модуля, який проектується наведено на рис. 7. З початку (позначено на діаграмі), активність, у будь-якому випадку, починається з оператора і містить у собі «завдання умов обробки»; далі – йде «передача даних» і в залежності від виконання умови «задано вірно?» (доріжка активності САД-модуля) відбувається перехід до «розрахунку КП» або повертання до оператора для повторного завдання.

Далі активність переходить на доріжку виробничої мережі («транслявання») і потім на доріжку СЧПК з метою «перевірка цілісності» та дотриманням умови «програма повна?» – якщо «так» – слідом йде «обробка» за допомогою СЧПК і завершення роботи (позначка на діаграмі), якщо «ні» – відбувається зворотнє «транслявання» (доріжка виробничої мережі) до САД-модуля з «індикацією проблем» до оператора та прийняття їм відповідного рішення.

Після чого, якщо відбулося «усунення проблеми» – настає повтор «розрахунку КП», а далі – вже відомий шлях: «транслявання» – «перевірка цілісності» – перевірка умови «програма повна?» – «обробка»; у випадку, якщо «усунення проблеми» – не відбулося – настає зупинка роботи до більш детального вивчення причин проблеми, що з'явилася.

Таким чином, було побудовану ДА, яка включає всі цикли проблем, що можуть з'явитися та шляхи їх вирішення із встановленням об'єктів активізації (рис. 7).

¹⁾ [25] Мацяшек Л. А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML. Москва: Вильямс, 2002. 432 с.

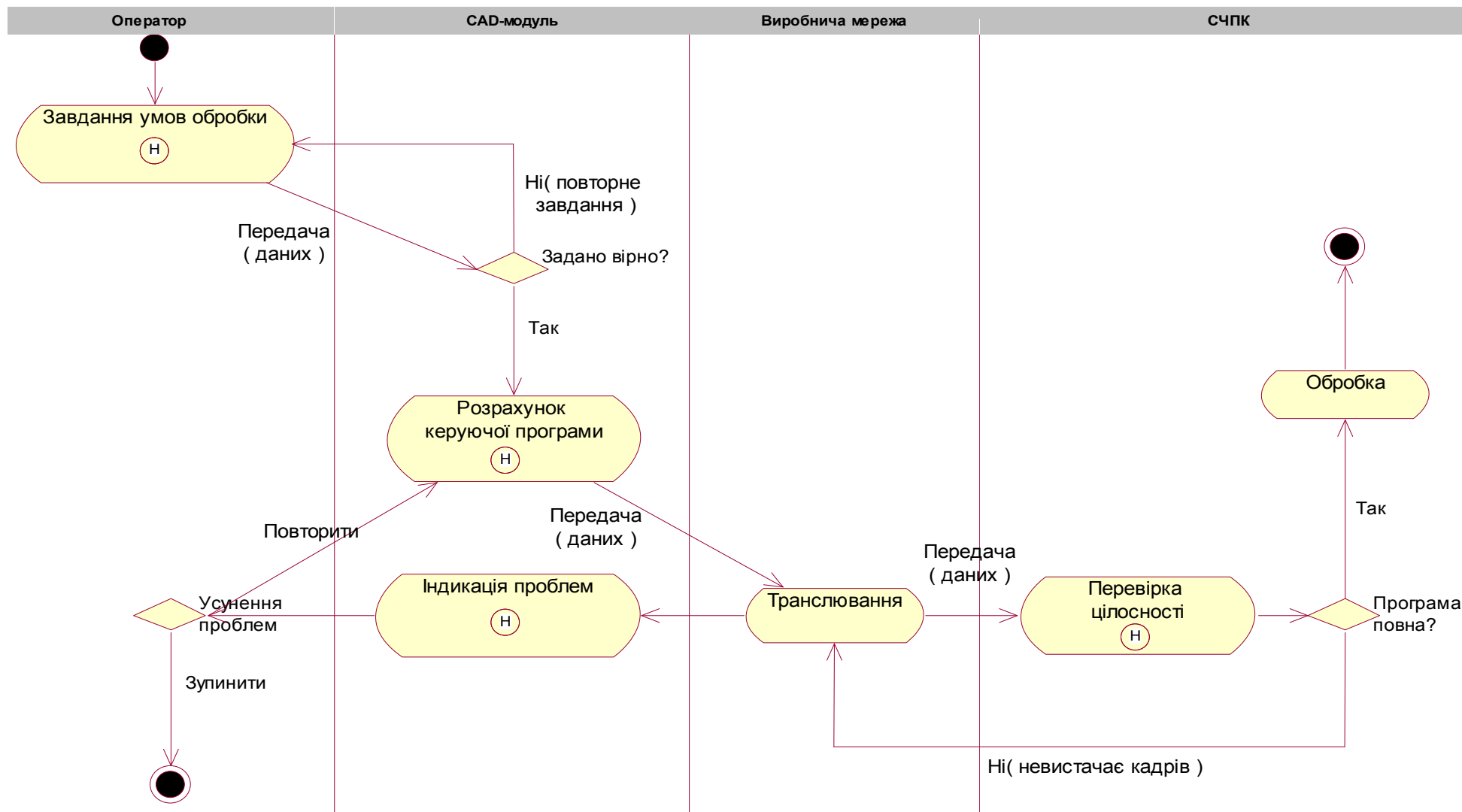


Рисунок 7 – ДА CAD-модуля

2.5 Побудова діаграми компонентів

ДКм показує структуру компонентів, які входять до проектуємої системи та які взаємодіють із іншими пакетами програм [26] ¹⁾ та операційними системами (ОС).

Компонент являє собою фізичний модуль коду, тобто окремий кодінг, для виконання заздалегідь визначеної функції, що входить до САД-модуля. На ДКм існує тільки один тип зв'язків – це залежності, які показують який з компонентів входить в склад іншого (залежить) [27] ²⁾. В такому випадку стрілка залежності направлена в сторону головного компонента (компонент «В» залежить від компонента «А», де «А» – головний).

Спроектвана ДКм наведена на рис. 8, до неї входять компоненти, що розділяються за такими стереотипами:

- тіло пакета;
- специфікація пакета;
- задача;
- головна програма;
- тіло підпрограми;
- специфікація підпрограми;
- графічний компонент та примітка.

Розглянемо детальніше структуру нашої ДКм. Почнемо з нижнього рівня, а саме з тих компонентів САД-модуля, які необхідно розробити. До них відносяться:

- Arg.exe – головна програма, яка запускає САД-модуль;
- Arg.m – тіло підпрограми, що містить весь вихідний кодінг, із математичним алгоритмом включно;

¹⁾ [26] Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник. Санкт-Петербург: Питер, 2002. 656 с.

²⁾ [27] Фаулер М., Скотт К. UML. Основы. Санкт-Петербург: Символ-Плюс, 2002. 192 с.

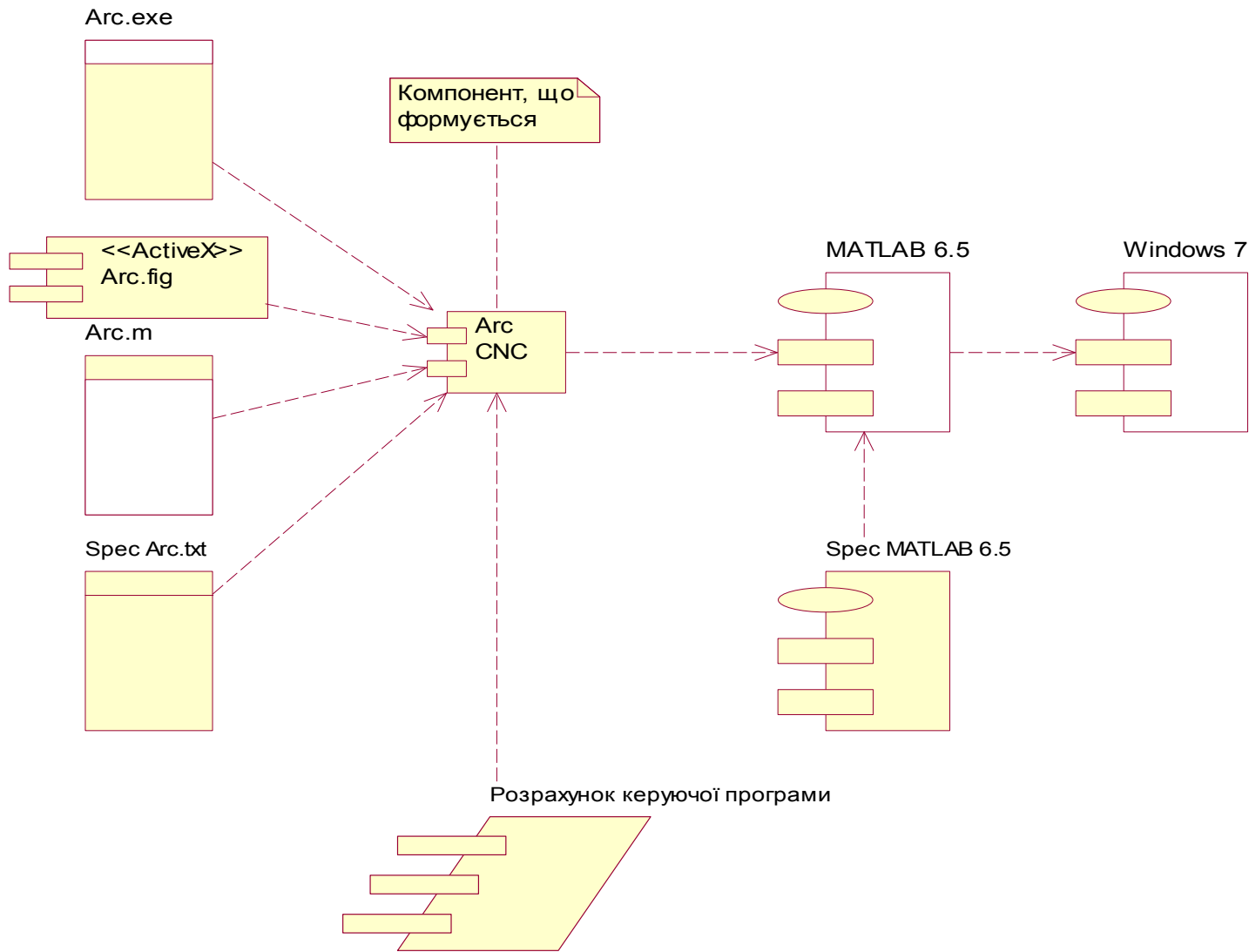


Рисунок 8 – ДКМ САД-модуля

- Arc.txt – специфікація підпрограми, тобто дані про версію, дату останньої редакції, середовище реалізації та ін.
- Arc.fig – графічний компонент, який виконує всі відображення за проведеними розрахунками.

Всі перелічені компоненти входять (залежать), до формуемого компонента Arc CNC (CAD-модуля), який, в свою чергу, залежить від пакету прикладних програм MATLAB 6.5 (із відповідним до MATLAB компонентом специфікації) , до якого він підключається і компонентом якого він – фактично стає. А вже MATLAB 6.5 підключається (інсталюється) до ОС Windows 7 (на ДКм вказана у вигляді пакету).

Отже, була спроектована ДКм, стосовно до розроблюваного CAD-модуля, яка включає як компоненти, що будуть спроектовані, так і компоненти від яких вони будуть залежати.

2.6 Побудова діаграми розміщення

ДР призначені для проектування розміщення фізичних пристроїв майбутньої системи (вузлів) із обліком усіх мережевих сполучень між ними [28] ¹⁾.

У ДР, яка розробляється стосовно до нашого CAD-модуля, будуть брати участь такі вузли:

- процесори – апаратура, яка спроможна виконувати розрахунки;
- пристрої – апаратура, яка не виконує розрахунків, а служить для вводу та виводу даних або підтримки інших будь-яких функцій, які виконуються процесорами.

Спроекована ДР наведена на рис. 9.

У ній у якості процесора виступають два вузли:

¹⁾ [28] Чмырь И. А., Ус М. Ф. Моделирование систем в среде UML (Unified Modeling Language): конспект лекций. Одесса: ОГАХ, 2001. 70 с.

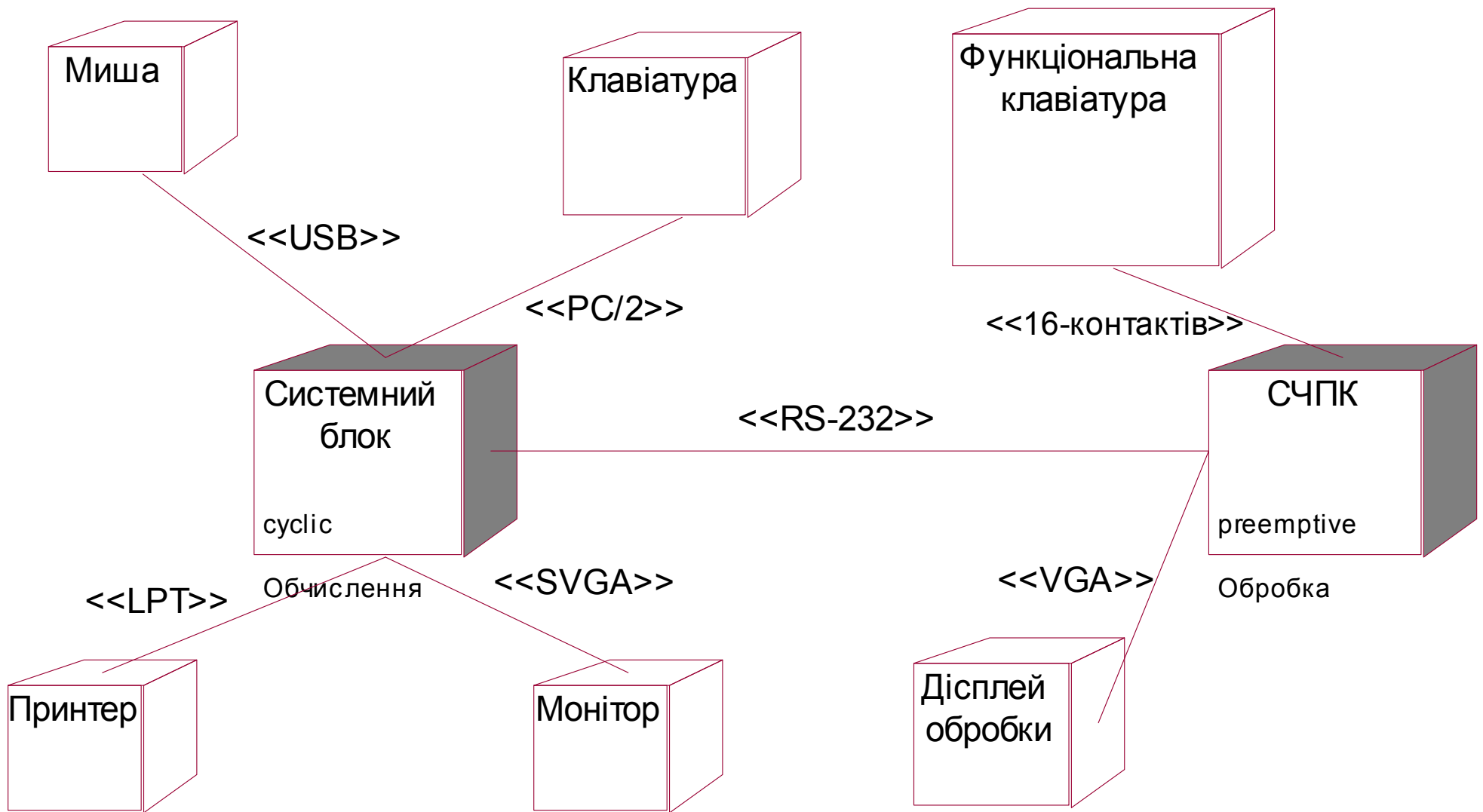


Рисунок 9 – ДР CAD-модуля

- системний блок (конструктора) із функцією виконання необхідних «обчислень» та циклічною роботою;
- СЧПК із функцією «обробки» та пріоритетним викликом, корегуванням або зупинкою її.

Ці два процесори пов'язує між собою виробнича мережа, яка реалізована відповідно до інтерфейсу передачі RS-485.

У якості пристроїв виступають такі пристрої вводу / виводу:

- клавіатура (виконує функцію уводу технологічної інформації до процесору «системний блок»);
- миша (виконує функцію уводу різноманітної допоміжної інформації до процесору «системний блок»);
- принтер (виконує функцію виводу конструкторської інформації з процесора «системний блок»);
- монітор (виконує функцію виводу різноманітної інформації з процесору «системний блок»);
- функціональна клавіатура (виконує функцію корегування технологічної інформації до процесору «СЧПК»);
- дисплей обробки (виконує функцію виводу конструкторсько-технологічної інформації з процесору «СЧПК»).

Отже, була спроектована ДР фізичних пристроїв, які будуть взаємодіяти із CAD-модулем, що проектується. Крім того, подана ДР враховує всі параметри передачі майбутньої виробничої мережі.

2.7 Висновки за розділом

В процесі виконання даного розділу магістерської атестаційної роботи, була спроектована архітектура майбутнього CAD-модуля за допомогою методології каркасного моделювання UML. Були розроблені наступні діаграми:

- діаграма варіантів використання (ДВВ);
- діаграма послідовності (ДП);

- кооперативна діаграма (КД);
- діаграма активності (ДА);
- діаграма компонентів (ДКм)
- діаграма розміщення (ДР).

На основі побудованих UML-діаграм, яки становлять основу для розробки самого САД-модуля, приступимо до виконання наступних етапів проектування, а саме: написання кодингу та формування інтерфейсу модуля.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ САД-МОДУЛЯ, ЩО ПРОЕКТУЄТЬСЯ

3.1 Алгоритм інтерполяції дуги окружності для СПВР

На підставі відповідних розроблених UML-діаграм, стосовно до завдань програмного керування рухами РО верстатів із ЧПК та робототехнічних комплексів, сформовано алгоритм інтерполяції дуги окружності для СПВР.

Алгоритм сформований стосовно до двох ПЗ:

- організувати рух РО СПВР по дузі окружності із заданої початкової точки \vec{r}_1 в задану кінцеву точку \vec{r}_3 , через задану точку \vec{r}_2 , з постійною кутовою швидкістю $\omega_{зад}$. Визначити необхідний час t переміщення РО за розрахованою траєкторією із точку \vec{r}_1 в точку \vec{r}_3 ;
- організувати рух РО СПВР по дузі окружності із заданої початкової точки \vec{r}_1 в задану кінцеву точку \vec{r}_3 , через задану точку \vec{r}_2 , за заданий час $t_{зад}$. Визначити необхідну постійну кутову швидкість ω переміщення РО з початкової точки \vec{r}_1 в кінцеву \vec{r}_3 .

Обидві постановки завдання припускають попередній розрахунок довжини l дуги окружності й визначення поточних координат положення кінця РО (у кожний заданий дискретний момент часу t_i), у процесі руху.

3.1.1 Структура алгоритму

В цілому, алгоритм формування траєкторії руху за дугою окружності зводиться до наступної послідовності дій.

Крок 1. Завдання координат трьох точок \vec{r}_1 , \vec{r}_2 та \vec{r}_3 в АСК СПВР – $OXYZ$.

Крок 2. Розрахунок в АСК абсолютних координат точки \vec{r}_0 – центра окружності, що проходить через три завдані (вихідні) точки.

Крок 3. Завдання ВСК СПВР – $\hat{O} \hat{X} \hat{Y} \hat{Z}$ та визначення напрямку руху РО за дугою окружності від точки \vec{r}_1 до точки \vec{r}_3 через точку \vec{r}_2 .

Крок 4. Розрахунок довжини дуги ℓ_{13} у кутових (рад, кут. град) та умовних лінійних (м, мм) одиницях.

Крок 5. Увід завданих технологічних параметрів руху РО за дугою окружності, у відповідності із варіантом постановки задачі: 1 або 2.

Крок 6. Розрахунок, у залежності від постановки задачі, або часу t переходу РО з початкової точки \vec{r}_1 в кінцеву \vec{r}_3 , або потрібної кутової швидкості ω руху РО за дугою.

Крок 7. Завдання кількості N поточних точок, що фіксуються, траєкторії руху РО (або, відповідно, інтервалів дискретизації руху за часом), виведених у розрахункові моменти часу t_i ($i = \overline{0, N}$).

Крок 8. Розрахунок матриць T та T^{-1} перетворення прямокутних СК (матриць стану РО), радіусу R дуги окружності, визначення відносних (у ВСК – $\hat{O} \hat{X} \hat{Y} \hat{Z}$) x_i, y_i, z_i та абсолютних (у АСК – $OXYZ$) x_i, y_i, z_i координат рухомої точки дуги окружності (зокрема, положення кінця РО), у кожен заданий дискретний момент часу t_i ($i = \overline{0, N}$). Перевірка виконання обмежень на область припустимих значень (ОПЗ) абсолютних координат, кожної розрахункової точки траєкторії.

Крок 9. Організація видачі результуючої інформації, що містить наступні параметри руху РО: час t , кутову ω та лінійну v швидкості, відстань ℓ (що пройдено від моменту початку руху t_0), поточні координати (у ВСК та АСК) положення точки РО на дузі кола, у дискретні моменти часу t_i ($i = \overline{0, N}$).

3.1.2 Опис роботи алгоритму

Графічно алгоритм сформований у вигляді блок-схеми, що містить 312 блоків і розміщується на 12 аркушах А4.

Робота алгоритму починається з уведення вихідних даних (блок 1): № варіанта завдання; координат вихідних точок \vec{r}_1 , \vec{r}_2 і \vec{r}_3 ; варіанта постановки завдання; варіанта виду руху (ВР) РО (за дугою або повною окружністю); кількості інтервалів дискретизації часу руху – N .

У блоках 3 – 6 виконується перевірка координат точок \vec{r}_1 , \vec{r}_2 і \vec{r}_3 на збіг. У випадку повного збігу координат хоча б двох точок, організується відповідне повідомлення та аварійна «зупинка» роботи алгоритму.

У блоці 7 виконується перевірка на приналежність трьох вихідних точок одній прямій. При підтвердженні цієї умови, також організовується повідомлення та аварійна «зупинка» роботи алгоритму.

У блоках 9 – 12 обчислюються координати середин відрізків $\overline{r_1 r_2}$ і $\overline{r_2 r_3}$ та відповідних нормалей $\vec{\beta}^*$ і $\vec{\beta}^{**}$ до них.

Блоки 13 – 88 призначені для визначення абсолютних координат точки \vec{r}_0 – центру окружності, що проходить через три задані точки \vec{r}_1 , \vec{r}_2 і \vec{r}_3 .

Крім того, точка \vec{r}_0 – є початком координат ВСК – $\hat{O} \hat{X} \hat{Y} \hat{Z}$.

Визначення спрямованих косинусів t_{ij} ($i, j = \overline{1,3}$) осей ВСК, стосовно осей АСК, виконується в блоках 89 – 99.

Одночасно, блоки 96 – 100, – є блоками формування матриці T – перетворення прямокутних СК: ВСК в АСК (вирішення зворотного завдання кінематики СПВР).

Формування матриці T^{-1} для вирішення прямого завдання кінематики СПВР, виробляється в блоках 101 – 109.

У блоках 110 – 115 здійснюється перевірка ВСК і АСК на односторонність осей. АСК задається правобічною при виборі опорної СК СПВР. Орієн-

тація осей ВСК може бути змінена заміною напрямку осі $\hat{O}\hat{Y}$ на протилежне, що дозволяє, при необхідності, домогтися правобічної орієнтації обох СК.

Блоки 116 – 118 призначені для визначення відносних координат у системі ВСК вихідних точок \vec{r}_1 , \vec{r}_2 , \vec{r}_3 і точки \vec{r}_0 – центра окружності.

Блоки 121 – 127 – для визначення відносних координат і довжин відрізків (векторів) $\overrightarrow{\hat{r}_0 \hat{r}_1}$, $\overrightarrow{\hat{r}_0 \hat{r}_2}$ і $\overrightarrow{\hat{r}_0 \hat{r}_3}$.

У блоках 128 і 129 виробляється обчислення косинусів центральних кутів: $\cos n_{13}$ для дуги $\overset{\cup}{r_1 r_3}$, і $\cos n_{12}$ – для дуги $\overset{\cup}{r_1 r_2}$.

Значне місце в алгоритмі займає процедура рішення принципового завдання – визначення значення центрального кута n_{13} і вибір «вірного» напрямку руху по шуканій дузі окружності $\overset{\cup}{r_1 r_3}$.

У блок-схемі алгоритму розглянуто 20 варіантів можливого розташування вихідних точок \vec{r}_1 , \vec{r}_2 і \vec{r}_3 на квадрантах СК $\hat{O}\hat{X}\hat{Y}$. Конкретне розташування точок \vec{r}_1 , \vec{r}_2 і \vec{r}_3 визначає необхідний напрямок руху РО за дугою (за годинною стрілкою або проти години стрілки) і «вірне» обчислення значення центрального кута n_{13} .

Процедура визначення центральних кутів n_{12} і n_{13} , і вибір напрямку руху за дугою (для 20 варіантів розташування вихідних точок) розміщується в блоках 130 – 271.

Варто помітити, що в блок-схемі алгоритму кути n_{12} і n_{13} визначаються через тригонометричну функцію «arctg». Формули для обчислення значень центральних кутів n_{1j} ($j = 2, 3$) через функцію «arctg», залежно від напрямку руху РО по дузі $\overset{\cup}{r_1 r_j}$ ($j = 2, 3$), наведені в табл. 2.

Таблиця 2 – Обчислення значень центрального кута

Номер квадранту ВСК, у якому розташований кут $n_{1j}, j = 2, 3$		Формули для обчислення центрального кута $n_{1j}, j = 2, 3$
При русі РО проти годинної стрілки	При русі РО за годинною стрілкою	
I	IV	$n_{1j} = \arctg \sqrt{\frac{1 - \cos^2 n_{1j}}{\cos^2 n_{1j}}}$
II	III	$n_{1j} = \pi - \arctg \sqrt{\frac{1 - \cos^2 n_{1j}}{\cos^2 n_{1j}}}$
III	II	$n_{1j} = \pi + \arctg \sqrt{\frac{1 - \cos^2 n_{1j}}{\cos^2 n_{1j}}}$
IV	I	$n_{1j} = 2\pi - \arctg \sqrt{\frac{1 - \cos^2 n_{1j}}{\cos^2 n_{1j}}}$

У блоках 278 – 284, залежно від: постановки завдання (1 або 2) і види руху – $BP = 1$ (по дузі окружності) або $BP = 2$ (по повній окружності), виробляється розрахунок необхідних параметрів руху РО. Якщо $ПЗ = 1$, то рух відбувається із заданою кутовою швидкістю $\omega_{зад}$, а відсутнім параметром, є час t руху по дузі $r_1 r_3$, що і визначається – при $BP = 1$ по формулі:

$$t = \frac{n_{13}}{\omega_{зад}}, [c];$$

при $BP = 2$ – по формулі:

$$t = \frac{2\pi}{\omega_{зад}}, [c];$$

Для ПЗ = 2 – заданий час $t_{зад}$ руху по дузі окружності $r_1 r_3$, а відсутнім параметром, є необхідна кутова швидкість ω переміщення РО, що і визначається – при $BP = 1$:

$$\omega = \frac{n_{13}}{t_{зад}}, [рад / c];$$

відповідно, при $BD = 2$:

$$\omega = \frac{2\pi}{t_{зад}}, [рад / c];$$

Безпосереднє визначення координат положення кінця РО на дузі окружності, у кожний заданий дискретний момент часу t_i , в ВСК і АСК, виконується в блоках 285 – 309.

У блоках 310 – 312 формуються вихідні інформаційні повідомлення для відображення результатів роботи алгоритму.

3.1.3 Складання ідентифікаторів ПЗ

Після синтезу алгоритму інтерполяції дуги окружності й, безпосередньо, перед створенням ПЗ, що наочно відображає розраховану траєкторію руху СПВР, наведемо відповідності (табл. 3) між параметрами математичної моделі (ММ) руху СПВР (розглянутої в пунктах 3.1.1 – 3.1.2) і умовними по-

значками, що визначають відповідні параметри в ПЗ (Додаток А), тобто ідентифікаторами (ІД).

Таблиця 3 – Відповідність між параметрами ММ та ІД ПЗ

Позначення параметрів, що прийнято в ММ	Позначення ідентифікаторів, що прийнято в ПЗ
1. ПЗ, ВР	1. pz, dv
2. $t_{зад}, \omega_{зад}$ [радий/з], N	2. T0, ω_0 , n0
3. $\vec{r}_1 = \{x_1, y_1, z_1\}$	3. r1[1], r1[2], r1[3]
4. $\vec{r}_2 = \{x_2, y_2, z_2\}$	4. r2[1], r2[2], r2[3]
5. $\vec{r}_3 = \{x_3, y_3, z_3\}$	5. r3[1], r3[2], r3[3]
6. $\vec{\alpha}^* = \{x^*, y^*, z^*\}$	6. r4[1], r4[2], r4[3]
7. $\vec{\alpha}^{**} = \{x^{**}, y^{**}, z^{**}\}$	7. r5[1], r5[2], r5[3]
8. $\vec{\beta}^* = \{a^*, b^*, c^*\}$	8. r6[1], r6[2], r6[3]
9. $\vec{\beta}^{**} = \{a^{**}, b^{**}, c^{**}\}$	9. r7[1], r7[2], r7[3]
10. Визначники: А, В, С	10. r8[1], r8[2], r8[3]
11. Матриця $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$	11. Три масиви: a[1, 1], a[1, 2], a[1, 3]; a[2, 1], a[2, 2], a[2, 3]; a[3, 1], a[3, 2], a[3, 3]
12. det A	12. Oa
13. $\vec{b} = \{b_{11}, b_{12}, b_{13}\}^T$	13. r9[1], r9[2], r9[3]
14. Визначник: $-D \equiv b_{13}$	14. r9[3]

Продовження табл. 3

Позначення параметрів, що прийнято в ММ	Позначення ідентифікаторів, що прийнято в ПЗ
15. $\vec{r}_0 = \{x_0, y_0, z_0\}$	15. r0[1]=X1, r0[2]=Y1, r0[3]=Z1
16. Координати: A^*, B^*, C^*	16. c1[1], c1[2], c1[3]
17. Модуль вектора: $\sqrt{(A^*)^2 + (B^*)^2 + (C^*)^2}$	17. d1
18. $\vec{s} = \{m, n, p\}^T$	18. c3[1], z3[2], z3[3]
19. Модуль вектора \vec{s} : $\sqrt{m^2 + n^2 + p^2}$	19. d3
20. Модуль вектора: $\sqrt{A^2 + B^2 + C^2}$	20. d2
21. Матриця стану РО для ВСК $\hat{O} \hat{X} \hat{Y} \hat{Z}$ $\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & x_0 \\ t_{21} & t_{22} & t_{23} & y_0 \\ t_{31} & t_{32} & t_{33} & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	21. Для осі $\hat{O} \hat{X}$: $t[i, 1], i = 1, 2, 3, 4$; для осі $\hat{O} \hat{Y}$: $t[i, 2], i = 1, 2, 3, 4$; для осі $\hat{O} \hat{Z}$: $t[i, 3], i = 1, 2, 3, 4$; для точки \hat{O} – початку координат ВСК: $t[i, 4], i = 1, 2, 3, 4$
22. Матриця стану РО для АСК $OXYZ$ $\mathbf{T}^{-1} = \begin{bmatrix} (A^*)^T & - (A^*)^T \vec{r}_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	22. Для осі OX : $t1[i, 1], i = 1, 2, 3, 4$; для осі OY : $t1[i, 2], i = 1, 2, 3, 4$; для осі OZ : $t1[i, 3], i = 1, 2, 3, 4$; для точки O – початку координат АСК: $t1[i, 4], i = 1, 2, 3, 4$
23. $\det \mathbf{A}^* = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$	23. Oa

Продовження табл. 3

Позначення параметрів, що прийнято в ММ	Позначення ідентифікаторів, що прийнято в ПЗ
24. $\vec{\hat{r}}_0 = \{\hat{x}_0, \hat{y}_0, \hat{z}_0, 1\}$	24. r0_1[1], r0_1[2], r0_1[3], r0_1[4]
25. $\vec{\hat{r}}_1 = \{\hat{x}_1, \hat{y}_1, \hat{z}_1, 1\}$	25. r1_1[1], r1_1[2], r1_1[3], r1_1[4]
26. $\vec{\hat{r}}_2 = \{\hat{x}_2, \hat{y}_2, \hat{z}_2, 1\}$	26. r2_1[1], r2_1[2], r2_1[3], r2_1[4]
27. $\vec{\hat{r}}_3 = \{\hat{x}_3, \hat{y}_3, \hat{z}_3, 1\}$	27. r3_1[1], r3_1[2], r3_1[3], r3_1[4]
28. $\vec{\hat{r}}_0 \hat{r}_1 = \{\hat{x}_1 - \hat{x}_0, \hat{y}_1 - \hat{y}_0, \hat{z}_1 - \hat{z}_0\}$	28. r0r1[1], r0r1[2] r0r1[3],
29. $\vec{\hat{r}}_0 \hat{r}_2 = \{\hat{x}_2 - \hat{x}_0, \hat{y}_2 - \hat{y}_0, \hat{z}_2 - \hat{z}_0\}$	29. r0r2[1], r0r2[2], r0r2[3]
30. $\vec{\hat{r}}_0 \hat{r}_3 = \{\hat{x}_3 - \hat{x}_0, \hat{y}_3 - \hat{y}_0, \hat{z}_3 - \hat{z}_0\}$	30. r0r3[1], r0r3[2], r0r3[3]
31. Модуль вектора $\vec{\hat{r}}_0 \hat{r}_1$: $ \vec{\hat{r}}_0 \hat{r}_1 = R$ [ум. лін. од.]	31. d4
32. Модуль вектора $\vec{\hat{r}}_0 \hat{r}_2$: $ \vec{\hat{r}}_0 \hat{r}_2 = R$	32. d5

Продовження табл. 3

Позначення параметрів, що прийнято в ММ	Позначення ідентифікаторів, що прийнято в ПЗ
33. Модуль вектора $\overrightarrow{r_0 r_3}$: $ \overrightarrow{r_0 r_3} = R$	33. d6
34. $\cos n_{12}, \cos n_{13}$	34. cn12, cn13
35. n_{12}, n_{13} [рад]	35. n12, n13
36. n_{12}, n_{13} [кут. град]	36. n12g, n13g
37. l_{12}, l_{13} [кут. лін. од.]	37. L12, L13
38. Номер гілки	38. nv
39. Напрямок руху	39. nd
40. t, ω [рад / с]	40. T0, w0
41. $\Delta t, \Delta \varphi$ [рад]	41. delta, delta
42. Лінійна швидкість: $\vec{v}_{зад} = \frac{R \cdot \omega_{зад}}{2\pi}$ [ум. лін. од. / с] або $\vec{v} = \frac{R \cdot \omega}{2\pi}$ [ум. лін. од. / с]	42. speed
43. $t_i, \varphi(t_i) \forall i = \overline{0, N}$	43. Tnow[i], Dum[i], $i = \overline{1, n0+1}$
44. $\varphi(t_i)$ [кут. град] $\forall i = \overline{0, N}$	44. Fnowgrad[i], $i = \overline{1, n0+1}$
45. $\hat{x}_i = \hat{x}(t_i), \hat{y}_i = \hat{y}(t_i),$ $\hat{z}_i = \hat{z}(t_i) \forall i = \overline{0, N}$	45. X_now[i], Y_now[i], $Z_now[i], i = \overline{1, n0+1}$

Кінець табл. 2

Позначення параметрів, що прийнято в ММ	Позначення ідентифікаторів, що прийнято в ПЗ
46. Координати поточної точки в АСК: $x_i = x(t_i)$, $y_i = y(t_i)$, $z_i = z(t_i) \quad \forall i = \overline{0, N}$	46. $X_{now}[i]$, $Y_{now}[i]$, $Z_{now}[i]$, $i = \overline{1, n0 + 1}$

3.2 Програмна реалізація САД-модуля

3.2.1 Загальні принципи роботи програмного засобу

Одним з підходів до вирішення поставлених завдань програмної реалізації – є використання інтерактивних методів одержання векторної моделі, при яких технолог задає три точки, що належать досліджуваній дузі кола, та ряд інших параметрів, котрі дозволяють керувати процесом інтерполяції одержуваного геометричного обводу, а, значить, і впливати на точність результату.

Апроксимація контуру дугами кіл знадобиться у випадку, коли геометрична модель містить велику кількість округлених ділянок. Виникаюча необхідність у даній делінеаризації, викликана особливостями технологій (наприклад, різання металу) та проблемами із керуванням рухом інструмента на коротких відрізках, при яких вирізка складних обводів, котрі описуються тільки відрізками прямих, призводила до істотного погіршення якості поверхні розрізу.

3.2.2 Написання програмного засобу «ARC»

Розглянутий алгоритм формування траєкторії руху РО СПВР за дугою кола, реалізовано у вигляді машинної програми «ARC». Скрипт програми написано із використанням компіляції текстового режиму ППП MATLAB (рис. 10), у вигляді інтерактивного ПЗ для ПК (додаток А), що доповнено

необхідним сервісним математичним забезпеченням, яке працює в діалоговому режимі.

```

1 function varargout = arc(varargin)
2 % Arc M-file for arc.fig
3 % Arc, by itself, creates a new Arc or raises the existing
4 % singleton*.
5 %
6 % H = Arc returns the handle to a new Arc or the handle to
7 % the existing singleton*.
8 %
9 % Arc('Property','Value',...) creates a new Arc using the
10 % given property value pairs. Unrecognized properties are passed via
11 % varargin to arc_OpeningFcn. This calling syntax produces a
12 % warning when there is an existing singleton*.
13 %
14 % Arc('CALLBACK') and Arc('CALLBACK',hObject,...) call the
15 % local function named CALLBACK in Arc.M with the given input
16 % arguments.
17 %
18 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22 %
23 % Edit the above text to modify the response to help arc
24 %

```

Рисунок 10 – Початок скрипта «ARC», створеного за допомогою MATLAB

ПЗ «ARC» розроблено стосовно до реальних задач інтерполяції дуги окружності на різних ділянках зони обслуговування СПВР, які працюють у прямокутній СК. Дане ПЗ точно розраховує й наочно відображає в просторі траєкторію руху РО і її проекції на кожен робочу площину. Програма повністю готова до застосування, вона дозволяє виконувати інтерполяцію дуги кола та практично реалізовувати програмне керування рухом за кожною прямокутною координатою переміщення РО СПВР.

3.2.3 Створення графічного інтерфейсу користувача ПЗ «ARC»

За допомогою вбудованого редактора графічних інтерфейсів, було виконано графічну обробку розрахованих результатів роботи програмного засобу (ПЗ) «ARC». При створенні графічного вікна відображення результатів, було сформовано, за допомогою внутрішніх функцій, вікно вводу початкових даних (рис. 11).

ввод данных					
	r1	r2	r3	постановка задачи	1
x	15	10	10	вид движения	
y	16	11	12	1-по дуге	1
z	13	18	17	2-по окружности	
w=	0.5	T=	12.56	N=	50
					Запуск

Рисунок 11 – Сформоване вікно вводу початкових даних

Після заповнення полів вікна вводу, відбувається побудова аксонометричної проекції дуги кола, що інтерполюється, у абсолютній системі координат (АСК) (для неї сформовано основне вікно, позначене «plot_mov» на рис. 12); проекцій дуги на три Декартові площини (три вікна праворуч, позначені «ХОУ», «YOZ» та «ZOX») та вигляду дуги у відносній системі координат (ВСК) (найменше вікно на рис. 12 – «plot_o»).

Інтерфейс ПЗ повністю відповідає сучасним пристроям уведення інформації в СПВР: контур руху РО відображається за чотирма основними проекціями, надаючи всі загальноприйняті точки огляду, до яких звикли конструктори й технологи.

Наведену вище структуру інтерфейсу, з усіма необхідними зв'язками між вікнами та командами побудови зображень необхідних проекцій, було зведено до файлу «Arc.fig».

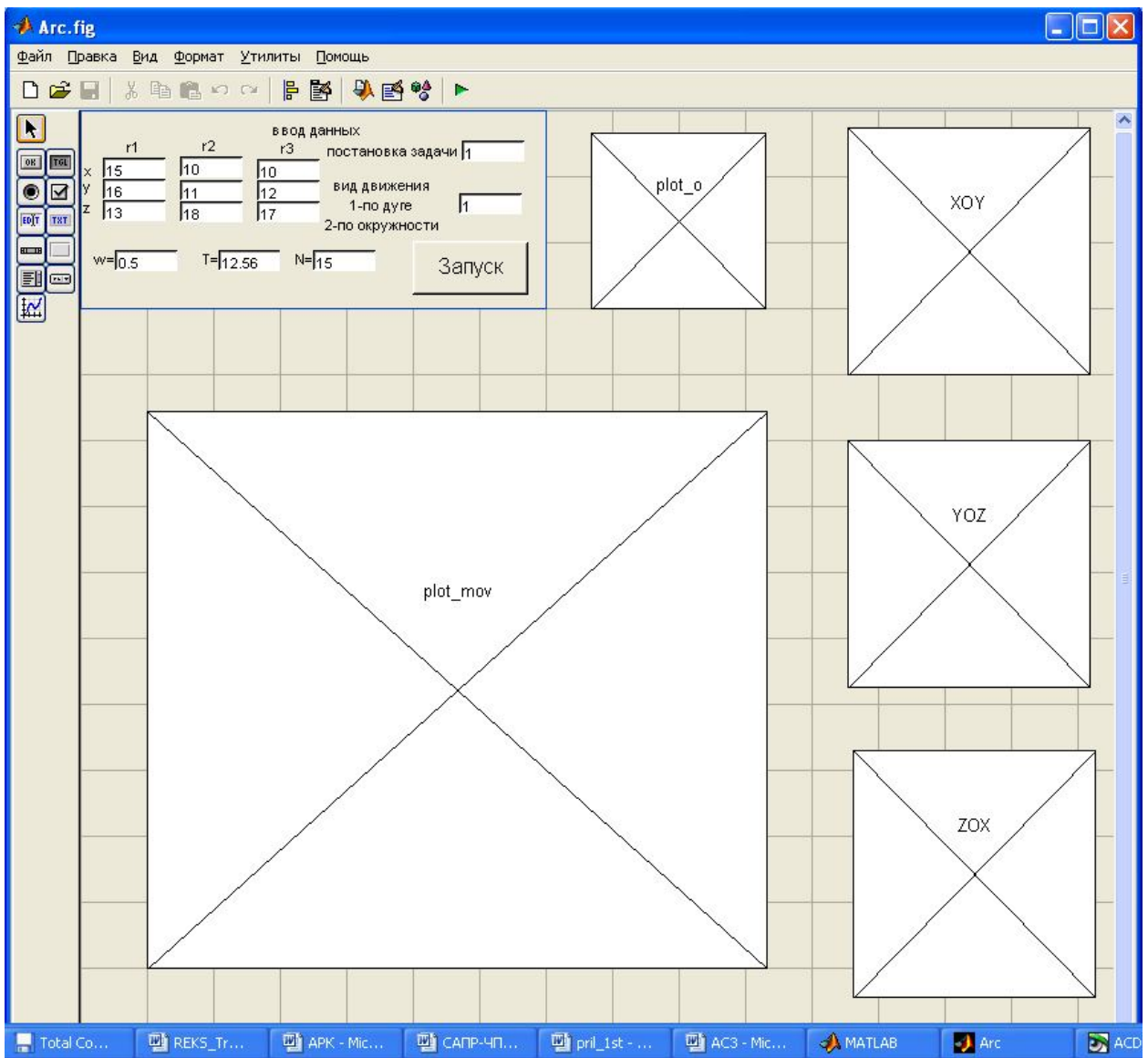


Рисунок 12 – Структура інтерфейсу виводу графічних результатів

3.3 Вказівки користувачеві програмного засобу «ARC»

Скрипт ПЗ «ARC» записано у вигляді файлу «Arc.m», що запускається командою «arc», у режимі командного рядка.

Безпосередня робота з ПЗ «ARC» починається із введення початкових даних. Для цього на екрані монітора з'являється вікно із запитом на введення початкових даних (рис. 12), до яких відносяться:

- номер варіанта постановки задачі (1 або 2);
- номер варіанта виду руху (1 або 2);

- абсолютні координати трьох вихідних точок дуги кола:
 $\vec{r}_1 = \{x_1, y_1, z_1\}$, $\vec{r}_2 = \{x_2, y_2, z_2\}$, $\vec{r}_3 = \{x_3, y_3, z_3\}$, що виражені в ум. лін. од., але на які існують кадрові обмеження відпрацювання:

$$1 \leq x_i \leq 5000,$$

$$1 \leq y_i \leq 5000,$$

$$1 \leq z_i \leq 5000, \quad i = 1, 2, 3;$$

- необхідна кутова швидкість руху за дугою кола – $\omega_{\text{зад}}$ (якщо постановка задачі: 1), виражена у рад / с, при обмеженні:

$$0 \leq \omega_{\text{зад}} \leq 10;$$

- необхідний час руху за дугою кола – $t_{\text{зад}}$ (якщо постановка задачі: 2), що виражений у секундах, із обмеженням тривалості відпрацювання:

$$0 \leq t_{\text{зад}} \leq 1000;$$

- кількість інтервалів дискретизації (що пов'язане із точністю інтерполяції) дуги кола – N , знаходиться у межах:

$$2 \leq N \leq 360.$$

Усі початкові дані можуть бути задані довільно (із урахуванням обмежень) або прийняті відповідно до технологічного креслення або САПР-моделі.

У залежності від варіанта постановки задачі, вводиться один з технологічних параметрів: $\omega_{\text{зад}}$ або $t_{\text{зад}}$. Значення параметра, що залишився, – уво-

дять рівним 0. На практиці, цілком достатньо вибрати кількість інтервалів дискретизації N у межах $10 \leq N \leq 50$.

Особливу увагу необхідно приділяти значенням координат вихідних точок \vec{r}_1 , \vec{r}_2 і \vec{r}_3 : точки не повинні належати одній прямій та координати навіть двох точок – не повинні збігатися! У протилежному випадку, програма не буде запускатися на розрахунок (бо не буде виконана умова кругової інтерполяції) – необхідно повторити уведення даних і увести припустимі значення координат початкових точок.

Після закінчення уведення початкових даних, необхідно у тому ж вікні (рис. 12), натиснути кнопку «ЗАПУСК», що пустить програму на розрахунок. Увесь розрахунок траєкторії виконується автоматично – без втручання користувача.

3.4 Організація виводу результатів розрахунків

Після виконання розрахунків на екрані з'явиться «вікно результатів», яке містить дані сформованої КП (командне вікно) (рис. 13).

У цьому вікні, якщо скористатися «прокруткою», то можна вивести всю інформацію, щодо проведених розрахунків, а саме:

- абсолютні координати трьох вихідних точок \vec{r}_1 , \vec{r}_2 і \vec{r}_3 – для контролю правильності уведення;
- абсолютні координати точки \vec{r}_0 – початку ВСК (центра кола);
- значення елементів матриці T – перетворення прямокутних координат ВСК у прямокутні координати АСК;
- значення елементів матриці T^{-1} – перетворення прямокутних координат АСК у прямокутні координати ВСК;
- відносні координати трьох вихідних точок \vec{r}'_1 , \vec{r}'_2 , \vec{r}'_3 , і \vec{r}'_0 ;

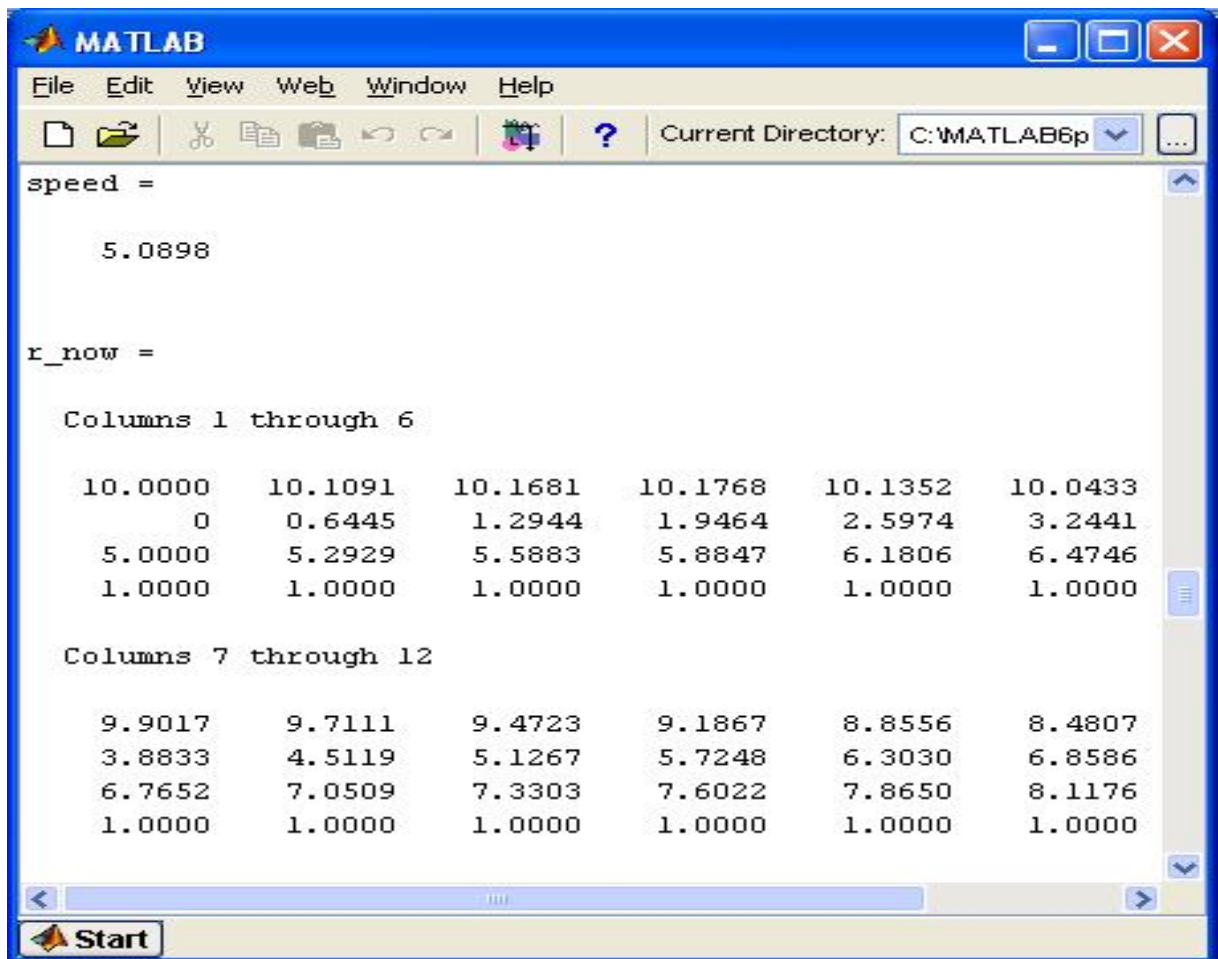


Рисунок 13 – Загальний вигляд вікна, що містить розраховані дані КП

- відносні координати трьох спрямованих відрізків $\overrightarrow{\hat{r}_0 \hat{r}_1}$, $\overrightarrow{\hat{r}_0 \hat{r}_2}$ і $\overrightarrow{\hat{r}_0 \hat{r}_3}$;
- значення модулів спрямованих відрізків $|\overrightarrow{\hat{r}_0 \hat{r}_1}|$, $|\overrightarrow{\hat{r}_0 \hat{r}_2}|$ і $|\overrightarrow{\hat{r}_0 \hat{r}_3}|$;
- значення косинусів центральних кутів: $\cos n_{12}$ і $\cos n_{13}$;
- напрямок руху РО за дугою кола, номер «гілки» алгоритму, а також значення центральних кутів n_{12} , n_{13} , обчислені за формулами та виражені у радіанах або кутових градусах;
- цей пункт результатів – є перевірочним: якщо побудувати ВСК $\hat{O}\hat{X}\hat{Y}$ та відкласти у ній точки $\vec{\hat{r}}_1$, $\vec{\hat{r}}_2$ і $\vec{\hat{r}}_3$, відповідно до початкових даних, а потім виміряти центральні кути n_{12} , і n_{13} , то отримані значення по-

винні збігатися із розрахунковими, що виконані за допомогою ПЗ «ARC»;

- приводяться значення розрахункових параметрів для визначення координат поточної рухливої точки дуги кола: кутова швидкість ω [рад / с] та час руху t [с], кількість інтервалів дискретизації N , лінійна швидкість \vec{V} [ум. лін. од. / с], дискретний крок за часом Δt [с] і збільшення кута $\Delta\varphi$ [рад];
- номери варіантів постановки задачі і виду руху, задані користувачем;
- таблиця із розрахунковими даними поточної точки дуги кола, у кожному дискретний момент часу t_i ($i = 0, N$) (координати поточної точки приведені послідовно у АСК та ВСК).

На рис. 14 наведено наочний приклад графічної побудови траєкторії руху РО СПВР за розрахованою КП, у вікні відображення результатів роботи ПЗ «ARC».

Слід додати, що під час перевірки та тестування даного програмного продукту, було розраховано та проаналізовано на вірність побудови більш ніж 80 різноманітних варіантів розташувань початкових точок.

3.5 Висновки до третього розділу

У третьому розділі магістерської кваліфікаційної роботи, у результаті практичної реалізації САД-модуля, було вирішено наступні задачі:

- виконано програмну реалізацію обчислювальних алгоритмів формування (додаток А);
- створено зручний інтерактивний ПЗ одержання й обробки графічної інформації – «ARC» (додаток А), за допомогою якого виконується розрахунок КП для обробки на СПВР дуги окружності, стосовно до завдань автоматизованого проектування складних геометричних контурів.

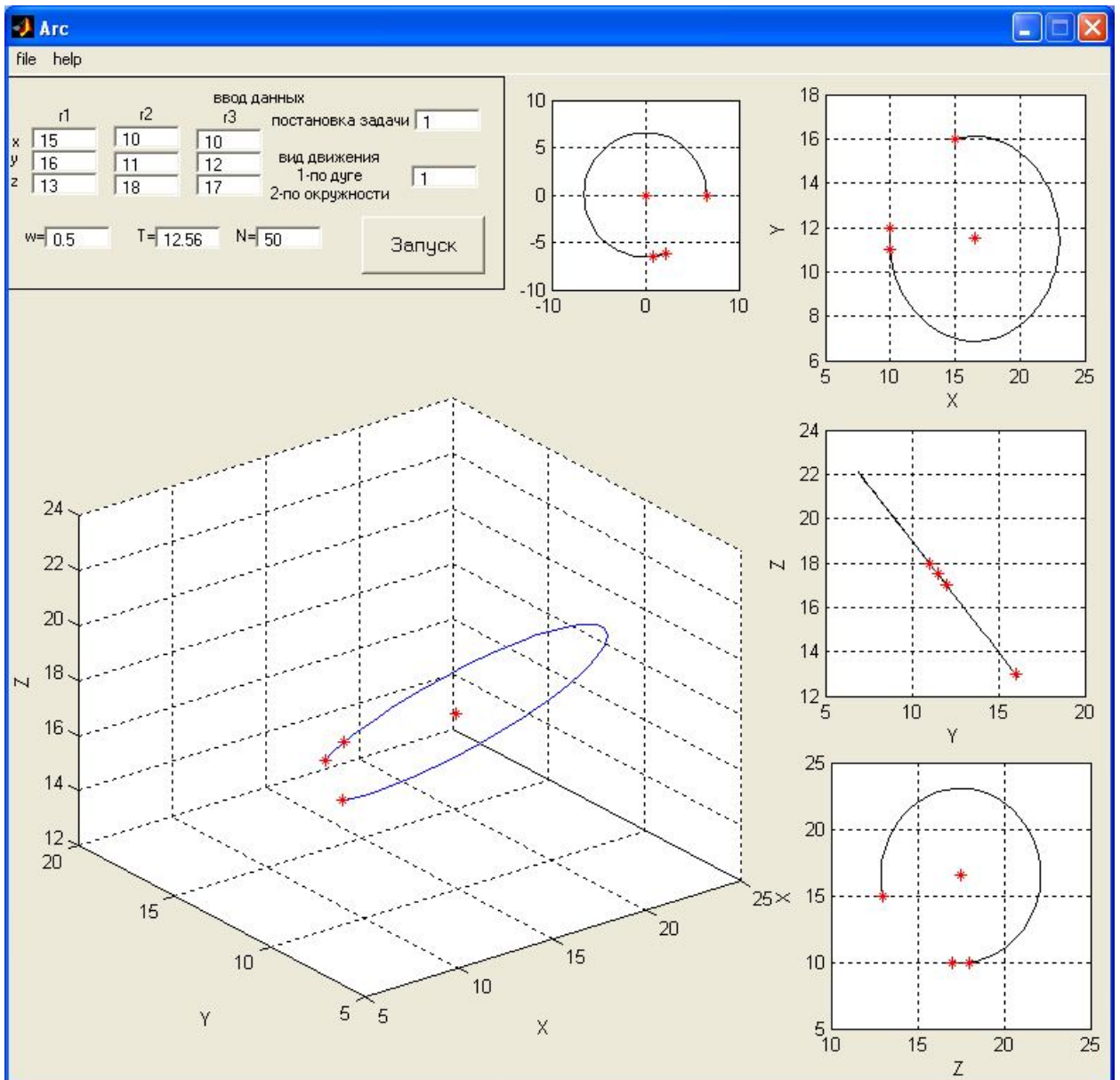


Рисунок 14 – Наочний приклад побудови траєкторії руху РО СПВР за розрахованою КП, у вікні графічного відображення результатів роботи ПЗ «ARC»

Сам процес вирішення практичного завдання роботи, тобто створення ПЗ, можна розділити на наступні етапи:

- компіляція сформованого алгоритму, на основі відповідного методу;
- створення графічного інтерфейсу користувача;
- організація виводу результатів розрахунку;
- формулювання необхідних коментарів для користувача.

Крім того, представлені результати третього розділу магістерської роботи, можуть бути застосовані при програмуванні різноманітних технологічних процесів обробки в цехах із ЧПК-верстатами та автоматичними маніпуляторами.

ВИСНОВКИ

В процесі виконання магістерської роботи, було досягнуто мету роботи, яка полягала в розробці CAD-модуля, що дозволяє виконувати чітко поставлені технологічно орієнтовані виробничі завдання проектування траєкторії руху робочого органу СПВР.

Подану мету роботи було досягнуто шляхом вирішення низки практичних завдань, що ставилися безпосередньо перед виконанням роботи, а саме:

- виконано огляд існуючих CAD / CAM / CAE технологій, для визначення кола аналогів, та проведено їхній техніко-економічний аналіз;
- змодельована архітектура CAD-модуля за допомогою методології каркасного моделювання UML;
- створено інтерактивну графічну систему одержання зображення контуру, інтерфейс якої повною мірою відповідав би сучасним пристроям уведення інформації до СПВР та до якого звикли конструктори й технологи.

Наукові результати полягають у такому:

- обрано метод інтерполяції дугами кола за рахунок однокритеріального аналізу і встановлення конкретного вигляду такої дуги, яка найбільш точно відтворює задану форму оброблюваної ділянки, що відрізняється від існуючих зменшенням часу опису складних контурів обробки, завдяки скороченню декомпозиції профілю на графічні примітиви;
- набув подальшого розвитку метод формування вихідних даних для пристрою керування СПВР, що дозволило скоротити кількість кадрів керуючої програми та, відповідно, зменшити ймовірність виникнення помилок зчитування.

Методи та моделі розв'язання завдань проектування реалізовані програмно. У процесі тестування та випробувань вони показали свою працездатність і ефективність на конкретних прикладах розв'язання завдань автомати-

зованого проектування та інтерполяції заданих складних геометричних контурів.

Результатом розробки став програмний продукт (у вигляді вихідного коду), готовий для установки та підключення до широкого спектру сучасних комп'ютерних операційних систем.

Результати досліджень реалізовано у вигляді розробленого програмно-методичного комплексу, який створює основу для розробки інструментальних засобів автоматизованого проектування заданого технологічного процесу, що виконується СПВР разом із САПР.

Сам САД-модуль, що створено на підставі методу й відповідного йому алгоритму, доповнено необхідним сервісним математичним забезпеченням та реалізується із застосуванням ППП MATLAB, у вигляді інтерактивного програмного засобу для ПК, що працює в діалоговому режимі. ПЗ «ARC» розроблено стосовно до реальних задач інтерполяції на різних ділянках зони обслуговування СПВР, що працюють у прямокутній системі координат. Засіб точно розраховує КП та наглядно відображує у просторі траєкторію руху РО та її проекції на кожну робочу площину. Також інтерфейс ПЗ повністю відповідає сучасним пристроям вводу інформації у СПВР.

Розроблено інструкції та рекомендації щодо застосування необхідного математичного та програмного забезпечення. При практичному застосуванні програмних засобів (ПЗ), вихідні дані можуть бути задані в ході розв'язання зворотної задачі кінематики або визначені інженером-технологом аналітично. Додатково у ПЗ передбачено необхідні коментарі відносно їх роботи, перевірки коректності завдання вихідних даних та умов фізичної реалізації сформованої траєкторії руху РО СПВР.

Практичне значення отриманих результатів полягає в тому, що розроблена інтерактивна графічна система, відрізняється від існуючих програмних засобів побудови геометричних моделей обводів тим, що траєкторія руху РО СПВР інтерполюється відразу колом (без наближень, які викликані лінійною інтерполяцією), що гарантує підвищену точність розрахунків, зменшення рі-

зноманітних похибок, скорочення довжини КП, а, відповідно, термінів її проектування та обробки.

Також вдосконалено процес формування вихідних даних для сполучення ПК із пристроєм керування СПВР: траєкторія руху формується у вигляді масивів точок, які задані в абсолютній та відносній системах координат, у залежності від дискретних можливостей конкретної СПВР та технологічних особливостей виготовлення, що враховуються у створеному програмному засобі.

Практичні результати спеціальної частини магістерської роботи можуть бути використано при викладанні на факультеті комп'ютерних наук, управління та адміністрування Одеського державного екологічного університету при проведенні лабораторних, практичних та окремих курсових робіт за дисциплінами, що відповідають тематиці магістерського наукового дослідження.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Филиппович К.В. Некоторые аспекты настройки пользовательских предпочтений в САПР-ЧПУ/2005. Пермь: ООО «Евразия Лимитед», 2005. 42 с.
2. Великодний С. С. Методологические основы реинжиниринга систем автоматизированного проектирования. *Междун. науч. журн. «Управляющие системы и машины»*. 2014. № 2. С. 39–43.
3. Schultz D.F. What's Wrong With Postprocessors? *Numerical Control Computer Sciences*. 1998. №1. P. 11–15.
4. Великодний С. С. Методи реінжинірингу програмних систем. *Технології приборостроєння*. 2014. Спец. вып. С. 65–68.
5. Великодний С. С., Тимофеева О. С., Зайцева-Великодна С. С., Нямцу К. С. Порівняльний аналіз властивостей відкритого, вільного та комерційного програмного забезпечення. *Інформаційні технології та комп'ютерна інженерія*. 2018. № 1 (41). С. 21–27.
6. Werner J. The Case For Verifying And Optimizing Tool Paths. Irvine: CGTech, 2003. 58 p.
7. Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання поведінкової частини. *Вісник Кременчуцького національного університету ім. Михайла Остроградського*. 2019. Вип. 2 (115). С. 117–126. (кат. «Б») DOI: 10.30929/1995-0519.2019.2.117-126.
8. Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання структурної частини. *Вісник Кременчуцького національного університету ім. Михайла Остроградського*. 2019. Вип. 3 (116). С. 130–139. (кат. «Б») DOI: 10.30929/1995-0519.2019.3.130-139.

9. Калачев О. Н. Диалоговая система KON7 для расчета размерных цепей при механообработке. Ярославль: ЯГТУ, 2005. 93 с.
10. Калачёв О.Н., Рехтер А.Д. Моделирование размеров механообработки в среде AutoCAD 200x на основе использования приложения GRAKON7. *САПР и графика*. 2002. №2. С. 100–104.
11. Velykodniy S. Reengineering of open software system of 3D modeling BRL-CAD. *Innovative Technologies and Scientific Solutions for Industries*. 2019. No 3 (9), P. 62–71. (кат. «Б») DOI: <https://doi.org/10.30837/2522-9818.2019.9.062>.
12. Velykodniy S. S. Analysis and synthesis of the results of complex experimental research on reengineering of open CAD systems. *Applied Aspects of Information Technology*. 2019. Vol. 2. No 3. P. 186–205. (кат. «Б») DOI: [10.15276/aait.03.2019.2](https://doi.org/10.15276/aait.03.2019.2).
13. Великодний С. С. Проблема реинжиниринга видов обеспечения систем автоматизированного проектирования. *Управляющие системы и машины*. 2014. № 1. С. 57–61, 76.
14. Трухнин Н. М., Филиппович К. В. Макрорасширения в GrafCAM – путь к наращиванию функциональности. Пермь: ООО «Евразия Лимитед», 2006. 72 с.
15. Филиппович К. В., Попович И. ToolStore – среда для ведения библиотеки инструментов в верификаторе CNC-Verify системы САПР-ЧПУ/2005. Пермь: ООО «Евразия Лимитед», 2005. 36 с.
16. Грунина Е.В., Калачёв О.Н. Методика проектирования в CAD/CAM Cimatron УП для гравирования профиля на юбилейной медали. Ярославль: ЯГТУ, 2008. 58 с.
17. Великодний С. С. Метод представлення оцінки реінжинірингу програмних систем за допомогою проектних коефіцієнтів. *Сучасний стан наукових досліджень та технологій в промисловості*. 2019. № 1 (7). С. 34–42. (кат. «Б») DOI: <https://doi.org/10.30837/2522-9818.2019.7.034>.

18. Великодний С. С., Тимофеева О. С., Зайцева-Великодна С. С. Метод розрахунку показників оцінки проекту при виконанні реінжинірингу програмних систем. *Радіоелектроніка, інформатика, управління*. 2018. № 4. С. 135–142. DOI: 10.15588/1607-3274-2018-4-13.
19. Боггс У., Боггс М., UML. Rational Rose: Секреты эффективного проектирования сопровод. объектно-ориентиров. прилож. Москва: «Лори», 2000. 570 с.
20. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя (Сер. программистов). Москва: ДМК, 2000. 432 с.
21. Гома Х., UML. Проектирование систем реального времени, параллельных и распределенных приложений. Москва: ДМК Пресс, 2002. 704 с.
22. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование. Москва: ДМК Пресс, 2001. 176 с.
23. Леоненков А. В. Самоучитель UML. Санкт-Петербург: БХВ-Петербург, 2002. 304 с.
24. Ларман К. Применение UML и шаблонов проектирования: уч. пос. Москва: Вильямс, 2001. 496 с.
25. Мацяшек Л. А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML. Москва: Вильямс, 2002. 432 с.
26. Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник. Санкт-Петербург: Питер, 2002. 656 с.
27. Фаулер М., Скотт К. UML. Основы. Санкт-Петербург: Символ-Плюс, 2002. 192 с.
28. Чмырь И. А., Ус М. Ф. Моделирование систем в среде UML (Unified Modeling Language): конспект лекций. Одесса: ОГАХ, 2001. 70 с.

ДОДАТОК А

Фрагменти вихідного тексту (лістинга коду) CAD-модуля

```

function varargout = arc(varargin)
% ARC M-file for arc.fig
%   ARC, by itself, creates a new ARC or raises the existing
%   singleton*.
%
%   H = ARC returns the handle to a new ARC or the handle to
%   the existing singleton*.
%
%   ARC('Property','value',...) creates a new ARC using the
%   given property value pairs. Unrecognized properties are
passed via
%   varargin to arc_OpeningFcn. This calling syntax produces
a
%   warning when there is an existing singleton*.
%
%   ARC('CALLBACK') and ARC('CALLBACK',hObject,...) call the
%   local function named CALLBACK in ARC.M with the given
input
%   arguments.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI
allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help arc
%
%   Copyright 2020 OSENU. ASMOS.
% Last Modified by GUIDE v3.2 06-Dec-2020 21:15:06
%
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @arc_OpeningFcn, ...
                  'gui_OutputFcn',  @arc_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before arc is made visible.
function arc_OpeningFcn(hObject, eventdata, handles, varargin)

```



```

% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from
the
%            command line (see VARARGIN)

% Choose default command line output for arc
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes arc wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command
line.
function varargout = arc_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object deletion, before destroying
properties.
function figure1_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on mouse press over figure background.
function figure1_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)
% hObject    handle to start (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
r1(1,1) = str2double(get(handles.r11,'String'));
r1(2,1) = str2double(get(handles.r12,'String'));
r1(3,1) = str2double(get(handles.r13,'String'));

```

```

r2(1,1) = str2double(get(handles.r21,'String'));
r2(2,1) = str2double(get(handles.r22,'String'));
r2(3,1) = str2double(get(handles.r23,'String'));
r3(1,1) = str2double(get(handles.r31,'String'));
r3(2,1) = str2double(get(handles.r32,'String'));
r3(3,1) = str2double(get(handles.r33,'String'));
w0 = str2double(get(handles.w0,'String'));
T0 = str2double(get(handles.T0,'String'));
n0 = str2double(get(handles.N0,'String'));
post_zap = str2double(get(handles.pz,'String'));
dvigenie = str2double(get(handles.vd,'String'));
%
% Шаг 1 задание ОСК
rr1=[r1; 1];
rr2=[r2; 1];
rr3=[r3; 1];
r4=0.5*(r1+r2);
r5=0.5*(r2+r3);
r6=r1-r2;
r7=r2-r3;
r9=[1; 1; 1];
A=det([[r1(2) r1(3); r2(2) r2(3); r3(2) r3(3)] r9]);
B=-det([[r1(1) r1(3); r2(1) r2(3); r3(1) r3(3)] r9]);
C=det([[r1(1) r1(2); r2(1) r2(2); r3(1) r3(2)] r9]);
D=-det([r1'; r2'; r3']);
G=[r6'; r7'; A B C];
if det(G)==0 error('detG=0'); end;
h=[r6'*r4; r7'*r5; -D];
r0=G^-1*h
rr0=[r0; 1];
%
c1=r1-r0; d1=(c1(1)^2+c1(2)^2+c1(3)^2)^.5;
c3(1)=c1(3)*B-c1(2)*C; c3(2)=c1(1)*C-c1(3)*A; c3(3)=c1(2)*A-
c1(1)*B;
d3=(c3(1)^2+c3(2)^2+c3(3)^2)^.5;
d2=(A^2+B^2+C^2)^.5;
tau=[c1(1)/d1 c3(1)/d3 A/d2;
      c1(2)/d1 c3(2)/d3 B/d2;
      c1(3)/d1 c3(3)/d3 C/d2];
if det(tau')<0 tau=[c1(1)/d1 -c3(1)/d3 A/d2;
                  c1(2)/d1 -c3(2)/d3 B/d2;
                  c1(3)/d1 -c3(3)/d3 C/d2];
elseif det(tau')==0 error('матрица косинусов вырождена'); end;
for i=1:3 t(i)=0; for j=1:3 t(i)=t(i)+tau(i,j)^2; end; end; t
% Шаг 2 определение матриц T и T^-1
T=[tau r0;
   0 0 0 1]
% находим заданные точки в ОСК
ro1=T^-1*rr1
ro2=T^-1*rr2
ro3=T^-1*rr3
ro0=T^-1*rr0
% координаты векторов заданных точек в ОСК
ro01=ro1-ro0
ro02=ro2-ro0
ro03=ro3-ro0
d4=(ro01(1)^2+ro01(2)^2+ro01(3)^2)^.5
d5=(ro02(1)^2+ro02(2)^2+ro02(3)^2)^.5
d6=(ro03(1)^2+ro03(2)^2+ro03(3)^2)^.5
% косинусы углов между векторами заданных точек в ОСК

```

```

cno12=(ro01'*ro02)/(d4*d5)
cno13=(ro01'*ro03)/(d4*d6)
% Шаг 3 выбор направления движения PO
if cno12>=0
    if ro2(2)>0
        if ro3(2)<0 % 1 четверть против часовой стрелки
            n12=atan(((1-cno12^2)/cno12^2)^.5);
n12g=n12*180*7/22;
            l12=d4*n12; ND=1;
            if cno13>0 % 4 четверть против часовой стрелки
                n13=44/7+atan(-((1-cno13^2)/cno13^2)^.5);
                n13g=n13*180*7/22; l13=d4*n13; NV=1;
            else % 3 четверть против часовой стрелки
                n13=22/7+atan(((1-cno13^2)/cno13^2)^.5);
                n13g=n13*180*7/22; l13=d4*n13; NV=2;
            end;
        else
            if ro3(1)>0
                if ro2(2)>ro3(2) % 1 четверть по часовой стрелке
                    n13=44/7+atan(-((1-cno13^2)/cno13^2)^.5);
                    n13g=n13*180*7/22; l13=d4*n13;
                    n12=44/7+atan(-((1-cno12^2)/cno12^2)^.5);
                    n12g=n12*180*7/22; l12=d4*n12; ND=-1; NV=3;
                else % 1 четверть против часовой стрелке
                    n13=atan(((1-cno13^2)/cno13^2)^.5);
                    n13g=n13*180*7/22; l13=d4*n13;
                    n12=atan(((1-cno12^2)/cno12^2)^.5);
                    n12g=n12*180*7/22; l12=d4*n12; ND=1; NV=4;
                end;
            else % 2 четверть против часовой стрелки
                n13=22/7-atan(((1-cno13^2)/cno13^2)^.5);
                n13g=n13*180*7/22; l13=d4*n13;
                n12=atan(((1-cno12^2)/cno12^2)^.5);
                n12g=n12*180*7/22; l12=d4*n12; ND=1; NV=45;
            end;
        end;
    else
        if ro3(2)>0 % 4 четверть по часовой стрелке
            n12=atan(((1-cno12^2)/cno12^2)^.5);
            n12g=n12*180*7/22; l12=d4*n12; ND=-1;
            if cno13>0 % 1 четверть по часовой стрелке
                n13=44/7+atan(-((1-cno13^2)/cno13^2)^.5);
                n13g=n13*180*7/22; l13=d4*n13; NV=5;
            else % 2 четверть по часовой стрелке
                n13=22/7+atan(((1-cno13^2)/cno13^2)^.5);
                n13g=n13*180*7/22; l13=d4*n13; NV=6;
            end;
        else
            if ro3(1)>0
                if ro3(2)>ro2(2) % 4 четверть против часовой
стрелки
                    n13=44/7+atan(-((1-cno13^2)/cno13^2)^.5);
                    n13g=n13*180*7/22; l13=d4*n13;
                    n12=44/7+atan(-((1-cno12^2)/cno12^2)^.5);
                    n12g=n12*180*7/22; l12=d4*n12; ND=1; NV=7;
                else % 4 четверть по часовой стрелке
                    n13=atan(((1-cno13^2)/cno13^2)^.5);
                    n13g=n13*180*7/22; l13=d4*n13;
                    n12=atan(((1-cno12^2)/cno12^2)^.5);
                    n12g=n12*180*7/22; l12=d4*n12; ND=-1; NV=8;
                end;
            end;
        end;
    end;
end;

```

```

end;
else % 3 четверть по часовой стрелке
n13=22/7-atan(((1-cno13^2)/cno13^2)^.5);
n13g=n13*180*7/22; l13=d4*n13;
% 4 четверть по часовой стрелке
n12=atan(((1-cno12^2)/cno12^2)^.5);
n12g=n12*180*7/22; l12=d4*n12; ND=-1; NV=85;
end;
end;
end;
else % выполнение цикла по правой ветви
if ro2(2)>0
if ro3(2)<0 % 2 четверть против часовой стрелки
n12=22/7-atan(((1-cno12^2)/cno12^2)^.5);
n12g=n12*180*7/22; l12=d4*n12; ND=1;
if cno13<=0 % 3 четверть против часовой стрелки
n13=22/7+atan(((1-cno13^2)/cno13^2)^.5);
n13g=n13*180*7/22; l13=d4*n13; NV=9;
else % 4 четверть против часовой стрелки
n13=44/7-atan(((1-cno13^2)/cno13^2)^.5);
n13g=n13*180*7/22; l13=d4*n13; NV=10;
end;
else
if ro3(1)<0
if ro2(2)<=ro3(2) % 2 четверть по часовой
стрелке
n13=22/7+atan(((1-cno13^2)/cno13^2)^.5);
n13g=n13*180*7/22; l13=d4*n13;
n12=22/7+atan(((1-cno12^2)/cno12^2)^.5);
n12g=n12*180*7/22; l12=d4*n12; ND=-1; NV=11;
else % 2 четверть против часовой стрелке
n13=22/7-atan(((1-cno13^2)/cno13^2)^.5);
n13g=n13*180*7/22; l13=d4*n13;
n12=22/7-atan(((1-cno12^2)/cno12^2)^.5);
n12g=n12*180*7/22; l12=d4*n12; ND=1; NV=12;
end;
else % 1 четверть по часовой стрелки
n13=44/7-atan(((1-cno13^2)/cno13^2)^.5);
n13g=n13*180*7/22; l13=d4*n13;
n12=22/7+atan(((1-cno12^2)/cno12^2)^.5);
n12g=n12*180*7/22; l12=d4*n12; ND=-1; NV=125;
end;
end;
else
if ro3(2)>0 % 3 четверть по часовой стрелке
n12=22/7-atan(((1-cno12^2)/cno12^2)^.5);
n12g=n12*180*7/22; l12=d4*n12; ND=-1;
if cno13>0 % 1 четверть по часовой стрелке
n13=44/7-atan(((1-cno13^2)/cno13^2)^.5);
n13g=n13*180*7/22; l13=d4*n13; NV=13;
else % 2 четверть по часовой стрелке
n13=22/7+atan(((1-cno13^2)/cno13^2)^.5);
n13g=n13*180*7/22; l13=d4*n13; NV=14;
end;
else
if ro3(1)<=0
if ro3(2)<ro2(2) % 3 четверть против часовой
стрелки
n13=22/7+atan(((1-cno13^2)/cno13^2)^.5);
n13g=n13*180*7/22; l13=d4*n13;

```

```

        n12=22/7+atan(((1-cno12^2)/cno12^2)^.5);
        n12g=n12*180*7/22; l12=d4*n12; ND=1; NV=15;
    else % 3 четверть по часовой стрелке
        n13=22/7-atan(((1-cno13^2)/cno13^2)^.5);
        n13g=n13*180*7/22; l13=d4*n13;
        n12=22/7-atan(((1-cno12^2)/cno12^2)^.5);
        n12g=n12*180*7/22; l12=d4*n12; ND=-1; NV=16;
    end;
else % 4 четверть против часовой стрелки
    n13=44/7-atan(((1-cno13^2)/cno13^2)^.5);
    n13g=n13*180*7/22; l13=d4*n13;
    % 3 четверть против часовой стрелки
    n12=22/7+atan(((1-cno12^2)/cno12^2)^.5);
    n12g=n12*180*7/22; l12=d4*n12; ND=1; NV=17;
end;
end;
end;
end;
ND
NV
n13
n13g
n12
n12g
% Шаг 4 определение ОК текущей точки в ОСК
if post_zap==1
    if dvigenie==1 T0=n13/w0;
        else T0=44/(7*w0); end;
else
    if dvigenie==1 w0=n13/T0;
        else w0=44/(7*T0); end;
end;
w0
n0
T0
deltaT=T0/n0
deltaF=w0*deltaT
speed=d4*w0
Tnow(1)=0;
Dum(1)=0;
Fnow_g(1)=0;
ro_now=zeros(4,n0+1); ro_now(:,1)=[d4; 0; 0; 1];
%
for i=2:n0+1
    Tnow(i)=Tnow(i-1)+deltaT;
    Dum(i)=Dum(i-1)+deltaF; Fnow_g(i)=Dum(i)*180*7/22;
    ro_now(1,i)=d4*cos(Dum(i));
    ro_now(2,i)=d4*sin(Dum(i));
    ro_now(3,i)=0; ro_now(4,i)=1;
end;
% Шаг 5 определение координат текущей точки в АСК
r_now=zeros(4,n0+1); r_now=T*ro_now
% шаг 6 построение диаграммы движения
axes(handles.plot_mov)
plot3(r_now(1,:),r_now(2,:),r_now(3,:),r1(1),r1(2),r1(3),'*r',r2
(1),r2(2),r2(3),'*r',...
r3(1),r3(2),r3(3),'*r',r0(1),r0(2),r0(3),'*r');
grid on; xlabel('X'); ylabel('Y'); zlabel('Z');

axes(handles.plot_o)

```

```

plot(ro_now(1,:),ro_now(2,:), '-
k',ro1(1),ro1(2),'*r',ro2(1),ro2(2),'*r',:...
    ro3(1),ro3(2),'*r',ro0(1),ro0(2),'*r')
grid on;

axes(handles.XOY)
plot(r_now(1,:),r_now(2,:), '-
k',r1(1),r1(2),'*r',r2(1),r2(2),'*r',:...
    r3(1),r3(2),'*r',r0(1),r0(2),'*r')
grid on; xlabel('X'); ylabel('Y');

axes(handles.YOZ)
plot(r_now(2,:),r_now(3,:), '-
k',r1(2),r1(3),'*r',r2(2),r2(3),'*r',:...
    r3(2),r3(3),'*r',r0(2),r0(3),'*r')
grid on; xlabel('Y'); ylabel('Z');

axes(handles.ZOX)
plot(r_now(3,:),r_now(1,:), '-
k',r1(3),r1(1),'*r',r2(3),r2(1),'*r',:...
    r3(3),r3(1),'*r',r0(3),r0(1),'*r')
grid on; xlabel('Z'); ylabel('X');

% -----
% -----
function m_above_callback(hObject, eventdata, handles)
% hObject    handle to m_above (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```