

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)
Розроблення комп'ютеризованої системи центрування принтера з
циліндричною системою координат для друку без підтримок
(тема)

Виконав:
здобувач 2 року навчання,
групи КТРСм-23-2
Піхтерьов Андрій Дмитрович
(прізвище, ініціали)

Спеціальність 174 Автоматизація
комп'ютерно-інтегровані технології та
робототехніка
(код і повна назва спеціальності)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютеризовані та
роботизовані системи
(повна назва освітньої програми)

Керівник доц. каф. КІТАР Разумов-Фризюк
Є.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

Невлюдов І.Ш.
(прізвище, ініціали)
2025 р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ
 Кафедра _____ КІТАР
 Рівень вищої освіти _____ другий (магістерський)
 Спеціальність 174 Автоматизація комп'ютерно-інтегровані технології та
 робототехніка
 Тип програми _____ Освітньо-професійна
 Освітня програма _____ Комп'ютеризовані та роботизовані системи
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав.кафедри _____

(підпис)

«__» _____ 2024р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Піхтерьов А.Д.
 (шифр і назва)

1. Тема роботи: Розроблення комп'ютеризованої системи центрування
принтеру з циліндричною системою координат для друку без підтримок
 Затверджена наказом університету №1239Ст від 25.11.2024
 від _____

2. Термін подання студентом роботи до екзаменаційної комісії 21.01.2025р.

3. Вихідні дані до роботи: 3.1 Циліндрична система координат принтеру;
 3.2 Додаткова ось обертання екструдеру; 3.3 Діаметр робочої зони 300 мм

4. Перелік питань, що потрібно опрацювати в роботі: 4.1 Вступ; 4.2 Аналіз
 особливостей 3D друку полярного 3D принтеру; 4.3 Розробка методів прог-
 рамного коректування g-коду у випадках помилки центрування платформи 3d
 Принтеру; 4.4 Розробка системи автоматичного пошуку центру осі θ ;
 Висновки;

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Графічний демонстраційний матеріал в форматі PowerPoint(*.ppt) формату А4 –15 сторінок.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по-батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз особливостей полярного 3D принтеру	25.11.2024-01.12.2024	виконано
2	Аналіз систем координат та G-коду для них	02.12.2024-07.12.2024	виконано
3	Аналіз видів дефектів зборки полярних 3D принтерів	08.12.2024-10.12.2024	виконано
4	Розробка алгоритму корегування G-коду	11.12.2024-17.12.2024	виконано
5	Розробка алгоритму перетворення G-коду у циліндричну систему координат	18.12.2024-21.12.2024	виконано
6	Розробка конструкції системи автоматичного пошуку центру осі θ	22.12.2024-25.12.2024	виконано
7	Розробка алгоритму та програми автоматичного пошуку центру осі θ	26.12.2024-31.12.2024	виконано
8	Оформлення звіту	01.01.2025-04.01.2024	виконано

Дата видачі завдання 25 листопада 2024р.

Здобувач _____ Піхтерьов А.Д.
 (підпис) (прізвище, ініціали)

Керівник роботи _____ Разумов-Фризюк Є.А.
 (підпис) (прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав та не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

21.01.2025



Піхтерьов А.Д.

РЕФЕРАТ

Пояснювальна записка: 78 с., 37 рис., 1 дод., 17 джерел.

3D ПРИНТЕР, СИСТЕМИ КООРДИНАТ, САМ СИСТЕМИ, G-КОД.

Мета роботи – розроблення автоматизованої програмно-технічної системи центрування 3D принтеру з циліндричною системою координат.

Об'єкт розробки – технологія екструзійного 3D друку.

Предмет розробки – підвищення точності 3D друку.

Виконано розробку системи автоматичної корекції системи координат полярного принтеру, що працює в циліндричній системі координат. Що дозволить зменшити кількість браку 3D друку, який пов'язаний із помилками або неточністю зборки принтеру.

ABSTRACT

Explanatory note: 778 p., 37 fig., 1 add., 17 sources.

3D PRINTER, COORDINATE SYSTEMS, CAM SYSTEMS, G-CODE.

The purpose of the work is to develop an automated software and hardware centering system for a 3D printer with a cylindrical coordinate system.

The object of development is extrusion 3D printing technology.

The subject of development is to increase the accuracy of 3D printing.

A system for automatic correction of the coordinate system of a polar printer operating in a cylindrical coordinate system has been developed. This will reduce the number of 3D printing defects associated with errors or inaccuracies in printer assembly.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	9
ВСТУП	10
1 АНАЛІЗ ОСОБЛИВОСТЕЙ 3D ДРУКУ ПОЛЯРНОГО 3D ПРИНТЕРУ	12
1.1 Аналіз систем координат ЧПУ верстатів	12
1.2 Аналіз застосування циліндричної та декартової систем координат в 3D принтерах	17
1.3 Аналіз особливостей полярних 3D принтерів.....	24
1.4 Висновки до першого розділу	32
2 РОЗРОБКА МЕТОДІВ ПРОГРАМНОГО КОРЕКТУВАННЯ G-КОДУ У ВИПАДКАХ ПОМИЛКИ ЦЕНТРУВАННЯ ПЛАТФОРМИ 3D ПРИНТЕРУ	33
2.1 Генерація g-коду для полярних 3D принтерів	33
2.2 Аналіз прямого використання циліндричної системи координат у ЧПУ верстатах	35
2.3 Розробка моделі перетворення G-коду для полярного 3D принтеру	40
2.4 Розробка програмного модулю автоматичної заміни координат у G-кодi	46
2.5 Висновки до другого розділу.....	51
3 РОЗРОБКА СИСТЕМИ АВТОМАТИЧНОГО ПОШУКУ ЦЕНТРУ ОСІ θ .	52
3.1 Розробка конструкції системи пошуку центру платформи	52
3.3 Висновок до третього розділу.....	70
4 МОДЕЛЮВАННЯ ТОЧНОСТІ ПОПОВРОТУ КРОКОВОГО ВИГУНА 3D ПРИНТЕРУ	72
4.1 Чисельне моделювання	72
4.2 охорона праці.....	75

ВИСНОВКИ.....	80
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	81
ДОДАТОК А Апробація результатів дослідження	84
ДОДАТОК Б Демонстраційний матеріал у вигляді презентації.....	94

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення;

ЧПУ – числове програмне управління;

CAD – Computer-aided design;

CAM – Computer-aided manufacturing.

ВСТУП

Актуальність забезпечення точності друку на полярних 3D-принтерах полягає в їхній здатності оптимально працювати з нестандартними формами та забезпечувати ефективне використання матеріалу. Полярні 3D-принтери, на відміну від традиційних, застосовують циліндричну систему координат, що дозволяє друкувати складні геометричні структури з мінімальною кількістю рухів друкуючої головки. Це скорочує витрати на енергію та час друку, а також знижує механічне навантаження на обладнання. Але разом із тим вимоги до точності зборки таких принтерів значно вище.

Точність друку є критично важливою для створення деталей, що повинні відповідати суворим параметрам або ідеально поєднуватися з іншими компонентами. У разі використання полярних принтерів, навіть незначні похибки можуть призводити до порушення симетрії та геометрії кінцевого виробу. Тому забезпечення високої точності таких принтерів дозволяє досягти кращої якості та стабільності продукції, особливо в сферах, що потребують високої точності, таких як авіація, медицина та машинобудування.

Полярні 3D принтери не є дуже поширеними в наш час, особливо у порівнянні із тими, що працюють у декартовій системі координат. Вдосконалення технологій калібрування та контролю якості друку на полярних 3D-принтерах також сприяє зростанню попиту на них у промисловості, де точність є вирішальним фактором. Це відкриває нові можливості для застосування полярних принтерів у виробництві, знижуючи залежність від традиційних координатних систем та розширюючи можливості швидкого прототипування.

Тож розробка автоматизованої системи центрування принтеру з циліндричною системою координат є актуальною задачею.

Також, отримані результати роботи можна віднести до Цілі сталого розвитку 12 – "Відповідальне споживання та виробництво", а саме п. 12.4: "Зменшити обсяг утворення відходів". Система центрування принтера з циліндричною системою координат, що розроблена у рамках даної роботи, дозволяє позбавитися технологічних підтримок під час FFF 3D друку, що у свою чергу дозволяє суттєво зменшити вартість виробництва, мінімізувати утворення відходів пластику і повністю прибрати необхідність їх переробки. Завдяки інноваційній технології принтер сприяє підвищенню ефективності роботизованого виробництва та відповідає вимогам сталого розвитку.

Мета роботи – розроблення автоматизованої програмно-технічної системи центрування 3D принтеру з циліндричною системою координат.

Об'єкт розробки – технологія екструзійного 3D друку.

Предмет розробки – підвищення точності 3D друку.

Робота виконана згідно [1-3].

1 АНАЛІЗ ОСОБЛИВОСТЕЙ 3D ДРУКУ ПОЛЯРНОГО 3D ПРИНТЕРУ

1.1 Аналіз систем координат ЧПУ верстатів

У верстатах з числовим програмним управлінням (ЧПУ) використовуються різні системи координат для керування переміщенням інструментів відносно заготовки. Кожна система координат має свою специфіку, призначення і використовується для різних типів завдань. Загалом, існує кілька видів систем координат, які застосовуються в ЧПУ [4].

Декартова система координат – це найпоширеніша система координат, яку використовують практично в усіх верстатах з ЧПУ. Вона базується на трьох взаємно перпендикулярних осях: X, Y і Z, які визначають положення точки у тривимірному просторі (рисунок 1.1) [5]:

- X-вісь – горизонтальна вісь, що задає рух зліва направо;
- Y-вісь – горизонтальна вісь, що задає рух вперед і назад;
- Z-вісь – вертикальна вісь, що задає рух вгору або вниз.

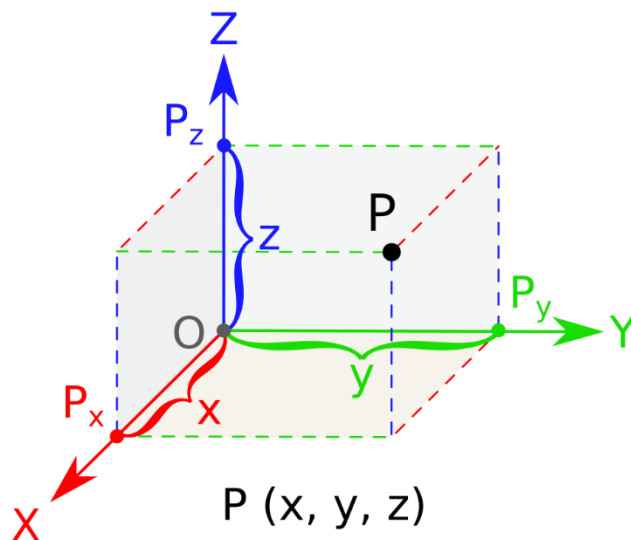


Рисунок 1.1 – Декартова система координат

Ця система координат дозволяє легко візуалізувати та програмувати переміщення інструмента або деталі у просторі.

Декартова система координат має наступні переваги:

- інтуїтивно зрозуміла та проста в програмуванні;
- використовується для багатьох типів верстатів (фрезерування, токарна обробка);
- придатна для обробки складних тривимірних форм.

Але їй притаманні і недолік. Може бути менш зручною для кругових або обертальних деталей, що обробляються по дугах або кривих.

Циліндрична система координат використовується для обробки кругових або обертальних деталей і базується на трьох координатах: радіус (R), кут (θ) і висота (Z) (рисунок 1.2):

- R – радіальна відстань від осі обертання до точки;
- θ – кут повороту навколо осі обертання;
- Z – висота точки вздовж осі обертання.

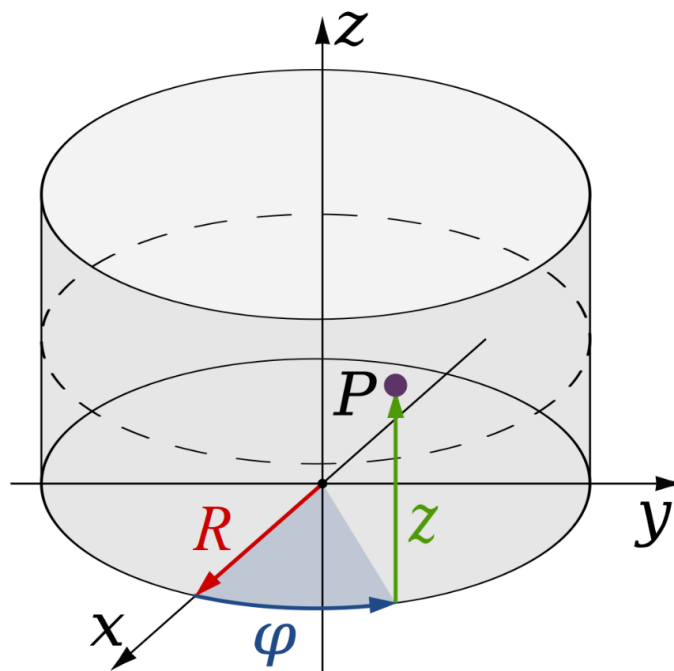


Рисунок 1.2 – Циліндрична система координат

Циліндрична система є зручною для верстатів, які обробляють деталі, що мають циліндричну симетрію, такі як осі, труби, диски тощо.

Циліндрична система координат має наступні переваги:

- спрощує обробку кругових і циліндричних деталей;
- зменшує обчислювальні навантаження при роботі з обертальними поверхнями.

Але їй притаманні і певні недоліки:

- менш ефективна для плоских або прямокутних деталей;
- вимагає додаткового розуміння і налаштування порівняно з декартовою системою.

Полярна система координат також використовується для опису кругових або обертальних рухів, але базується на двох основних параметрах: відстані від початку координат і куті відносно певного напрямку (рисунок 1.3):

- R – радіальна відстань від центру до точки;
- θ – кут повороту навколо початкової точки.

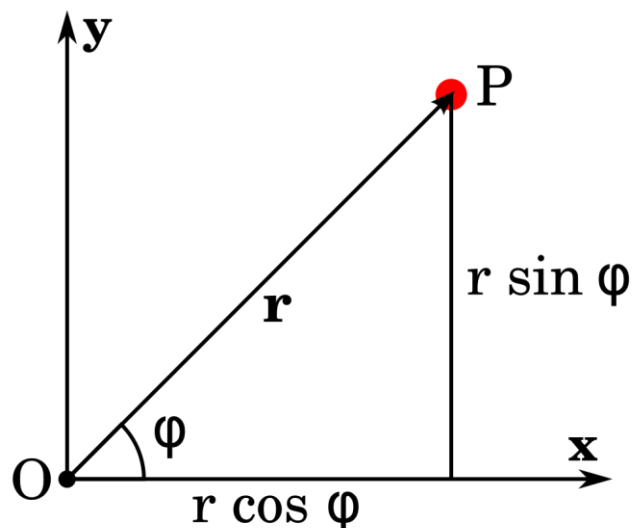


Рисунок 1.3 – Полярна система координат

Ця система зручна для обробки кругових деталей і часто використовується для свердлильних операцій по колу або обробки отворів у кругових заготовках. Але вона підтримує лише двовимірні операції.

Полярна система координат має наступні переваги:

- проста і ефективна для опису рухів по дугах і колах;
- використовується для свердління та різання кругових деталей.

Але їй притаманні і певні недоліки:

- обмежене використання для плоских або тривимірних деталей;
- не підходить для універсального програмування в багатьох типах завдань.

Сферична система координат використовується для визначення положення точок у сферичному просторі. Координати описуються трьома параметрами: радіусом (R), азимутальним кутом (θ) і полярним кутом (φ).

- R – відстань від центра до точки;
- θ – азимутальний кут, що визначає напрям руху в площині;
- φ – полярний кут, що визначає висоту точки відносно площини.

Сферична система менш поширена у верстатах з ЧПУ, але може бути корисною для специфічних задач, таких як обробка сферичних поверхонь або тривимірних форм із складною геометрією (рисунок 1.4).

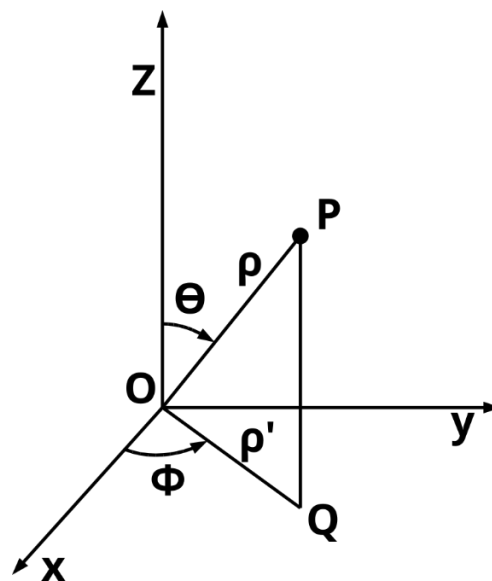


Рисунок 1.4 – Сферична система координат

Сферична система координат має наступні переваги:

- дозволяє працювати зі складними тривимірними формами;
- корисна для задач, де необхідно обробляти сферичні або купольні деталі.

Але їй притаманні і певні недоліки:

- вимагає складніших обчислень для точного програмування;
- рідше використовується в загальному виробництві.

Масштабована система координат є адаптованою до конкретних розмірів або форми заготовки. Вона використовується, коли необхідно масштабувати або адаптувати положення інструмента або деталі відносно нових координат. Наприклад, якщо деталь має інші розміри або положення у просторі, масштабована система координат допомагає скоригувати ці зміни.

Масштабована система координат має наступні переваги:

- гнучка і може використовуватися для роботи з деталями різних розмірів;
- дозволяє легко масштабувати програми для різних розмірів заготовок.

Але їй притаманні і недоліки: може бути складнішою для програмування у порівнянні зі стандартними системами координат.

Локальна система координат (G54 - G59) є частиною декартової системи і дозволяє визначати локальні початкові точки (нульові точки) для конкретних завдань. Наприклад, у ЧПУ часто використовують команди G54 - G59 для зміни нульової точки на різних частинах заготовки. G54 - G59 – коди, що визначають зміщення нульової точки для різних завдань.

Локальна система координат має наступні переваги:

- дозволяє легко змінювати початкові координати без необхідності перепрограмувати всю систему;
- використовується для точного позиціонування деталей при обробці.

Але їй притаманні і недоліки недоліки: не є окремою системою координат, а радше доповненням до декартової системи.

У верстатах з ЧПУ існує кілька видів систем координат, кожна з яких має свої переваги і застосування для різних виробничих завдань. Вибір системи координат залежить від форми заготовки, типу верстата і необхідної точності обробки. Декартова система є найпоширенішою і універсальною, тоді як циліндрична краще підходить для обробки кругових або обертальних деталей а у 3D принтерах дозволяє організувати друк буз підтримок.

1.2 Аналіз застосування циліндричної та декартової систем координат в 3D принтерах

У сучасному 3D-друці, так само як і в інших областях точного механічного обладнання, використовуються різні системи координат для керування переміщенням екструдера та платформи. Основними системами координат, що знаходять застосування у 3D-принтерах, є декартова та циліндрична системи координат. Кожна з цих систем має свої особливості, переваги та обмеження, що роблять їх більш або менш придатними для різних типів завдань. У даній статті ми детально розглянемо ці дві системи, їх переваги та недоліки, а також їх застосування у сучасних 3D принтерах.

Декартова система координат (також відома як прямокутна система координат) є найбільш поширеною в 3D-принтерах і базується на трьох взаємно перпендикулярних осях: X, Y і Z. Кожна з осей відповідає за рух екструдера або платформи у відповідному напрямку. Декартова система є найбільш інтуїтивно зрозумілою для більшості користувачів і широко використовується в різних типах обладнання для ЧПУ та 3D-друку. Ці осі дозволяють точно визначати положення екструдера відносно платформи для створення об'єкта шар за шаром.

Більшість сучасних 3D принтерів базуються на декартовій системі координат. Така конструкція зазвичай має кілька варіантів реалізації:

- класична конструкція з рухомим столом (Prusa-тип) – це стандартний варіант, де друкарська головка рухається по двох осях (X і Z), тоді як стіл переміщується по осі Y;

- гантрі-система (Gantry) – це коли друкарська головка може рухатися по всіх трьох осях, тоді як стіл залишається нерухомим. Така система використовується для великих промислових 3D принтерів, які друкують великі об'єкти;

- delta-принтери – хоча базуються на декартовій системі координат, Delta-принтери використовують три шарнірні важелі для управління рухом екструдера у просторі, що дозволяє досягти вищих швидкостей друку.

Переваги декартової системи координат:

- простота та універсальність. Декартова система є найбільш інтуїтивно зрозумілою і легкою для програмування та налагодження. Вона широко застосовується в 3D-принтерах різного типу, як у промислових, так і у домашніх пристроях;

- висока точність. Точність декартової системи залежить від якості лінійних направляючих і двигунів, але в цілому вона дозволяє досягти високої деталізації друку. Лінійні переміщення по кожній осі забезпечують точне позиціонування екструдера або платформи;

- широка підтримка програмного забезпечення. Декартова система координат є стандартом для більшості програм для підготовки моделей до друку, таких як Cura, PrusaSlicer та інші. Це забезпечує простоту інтеграції та сумісність з багатьма пристроями;

- доступність матеріалів та технологій. Велика кількість моделей і брендів 3D принтерів використовують декартову систему, що робить її найпоширенішою у виробництві і на ринку доступних принтерів.

Недоліки декартової системи координат:

- обмежена ефективність для великих об'єктів. У випадку з великими об'єктами або принтерами з великою робочою зоною можуть виникати проблеми з точністю через вагу компонентів і необхідність компенсувати рухи при великому масштабі;

- повільність при друку великих моделей. Оскільки всі три осі повинні рухатися окремо, час на переміщення може бути більшим у порівнянні з іншими системами, що призводить до зниження швидкості друку при великих об'ємах;

- складність механіки. Для великих і точних 3D принтерів з декартовою системою координат необхідні додаткові елементи стабілізації, компенсації та жорсткості конструкції для підтримання високої точності.

Циліндрична система координат базується на іншому підході до управління рухом екструдера та платформи. У цій системі положення точки в просторі визначається за допомогою трьох параметрів: R – радіус (відстань від центральної осі до точки), θ – кут обертання навколо центральної осі, Z – висота точки над основою.

Циліндрична система координат часто використовується в 3D принтерах, які спеціалізуються на друці кругових або обертальних об'єктів. У таких принтерах платформа або екструдер обертається навколо центральної осі, дозволяючи друкувати об'єкти з обертальною симетрією.

Циліндричні 3D принтери мають іншу механіку у порівнянні з декартовими:

- рухома платформа. Платформа обертається навколо осі Z , тоді як екструдер рухається радіально по осі R і вгору-вниз по осі Z ;

- обертальний екструдер. У деяких конструкціях екструдер може обертатися навколо заготовки, що дозволяє точніше контролювати подачу матеріалу;

- такі конструкції часто використовуються для друку об'єктів з круговою симетрією, таких як труби, вази, циліндри, і можуть мати перевагу над декартовими системами в таких завданнях.

Переваги циліндричної системи координат:

- оптимізована для кругових об'єктів. Циліндрична система значно спрощує друк об'єктів з обертальною симетрією, таких як циліндри або кругові компоненти. Вона дозволяє уникнути складних переміщень екструдера, зосередившись на обертальних рухах;
- компактність конструкції. Принтери з циліндричною системою координат зазвичай компактніші за своїми розмірами, оскільки не потребують великих направляючих для осей X і Y. Це може бути перевагою для домашніх або офісних принтерів;
- менше матеріальних витрат на підтримки. Завдяки круговій симетрії об'єктів, друк часто не потребує додаткових підтримуючих структур, що дозволяє економити матеріал і зменшити час на постобробку;
- ефективність у певних випадках. Для виготовлення конкретних типів об'єктів, таких як труби, кругові резервуари або навіть елементи архітектури, циліндрична система може бути значно ефективнішою, ніж декартова.

Але застосування циліндричної системи координат має і велику кількість недоліків.

Обмеженість у складних формах, так як циліндрична система найкраще підходить для друку об'єктів із круговою симетрією, таких як циліндри, вази або трубки. Однак вона має серйозні обмеження при роботі з об'єктами, які не мають обертальної симетрії або мають складну тривимірну форму, яку важко описати радіусом і кутом.

Проблема з друком плоских і прямокутних об'єктів: циліндрична система координат не підходить для друку плоских, прямокутних або неправильної форми деталей. Для таких об'єктів потрібні додаткові обчислення, а ефективність системи знижується.

Складність у програмуванні складних геометрій: оскільки циліндрична система краще працює з круглими поверхнями, друк деталей зі складною

геометрією може вимагати додаткових програмних інструментів або спеціальних алгоритмів для корекції переміщень.

Обмежена підтримка програмного забезпечення, хоча більшість програм для 3D-друку підтримують декартову систему координат, циліндрична система не настільки широко підтримується. Для підготовки моделей та управління циліндричними принтерами потрібне спеціалізоване ПЗ або модифікації у звичайних програмах.

Мала кількість готових профілів друку, програми для slicing (нарізки моделі на шари) зазвичай орієнтовані на декартові принтери, і тому профілі для циліндричних принтерів рідко зустрічаються у стандартних налаштуваннях.

Ускладнення при конверсії моделей. Стандартні 3D-моделі, створені для декартової системи, можуть вимагати перетворення координат для роботи з циліндричною системою, що ускладнює процес підготовки до друку.

Менша гнучкість у налаштуванні, циліндрична система має обмежені можливості для змін під час друку, особливо коли потрібно змінити геометрію друкованого об'єкта в процесі. У декартовій системі можна легко керувати лінійними переміщеннями у трьох вимірах, а у циліндричній системі зміна радіуса чи кута може вимагати складніших алгоритмів.

Важче адаптуватися до змін у процесі друку. У циліндричній системі складно коригувати модель у режимі реального часу, що робить її менш гнучкою для об'єктів зі змінною геометрією.

Залежність від центральної осі. Вся система координат базується на центральній осі обертання, тому друк об'єктів, які знаходяться далеко від цієї осі, може призводити до зниження точності через неможливість ефективно компенсувати зміни відстані.

Можливі обмеження у швидкості друку, оскільки циліндричні 3D-принтери часто використовують поворотні рухи для переміщення платформи або екструдера, це може призводити до додаткових затримок або обмежень у

швидкості друку. Такі механізми можуть бути менш ефективними для високошвидкісного друку у порівнянні з лінійними системами.

Обмеження швидкості обертання платформи. Платформа, що обертається, може мати обмеження по швидкості через фізичні обмеження механізму або можливі вібрації на високих швидкостях.

Необхідність точного контролю обертальних рухів. Системи з обертальним управлінням вимагають точного контролю кутових переміщень, що може бути більш складним, ніж контроль лінійних переміщень у декартовій системі.

Точність друку може бути нижчою, циліндричні 3D-принтери можуть мати нижчу точність у порівнянні з декартовими через обмеження у переміщеннях по кутових траєкторіях. Лінійні рухи у декартовій системі легше контролювати та калібрувати для забезпечення високої точності друку.

Кутові похибки. Похибки у кутових обчисленнях можуть призводити до неточностей у відтворенні деталей, особливо якщо модель має складну форму, яка виходить за межі простої кругової симетрії.

Проблеми зі стабільністю об'єкта: у процесі друку великі або нестабільні об'єкти можуть виявлятися менш стабільними на обертовій платформі, що може призводити до дефектів друку.

Складна механіка і конструкція, циліндричні принтери мають більш складну конструкцію у порівнянні з декартовими через необхідність у обертових компонентах і механізмах для керування кутовими переміщеннями. Це може призвести до підвищених витрат на виготовлення та обслуговування таких пристроїв.

Більша кількість рухомих частин: складніша механіка може потребувати частішого обслуговування і калібрування. Це також робить принтери дорожчими і менш надійними при тривалому використанні.

Важчий доступ до запчастин і технічного обслуговування: через меншу поширеність циліндричних 3D-принтерів, доступ до запчастин або спеціалізованих сервісів для ремонту таких пристроїв може бути обмеженим.

Вузька спеціалізація. Циліндричні 3D-принтери зазвичай використовуються для друку об'єктів з обертальною симетрією, тому вони є вузькоспеціалізованими. Вони не настільки універсальні, як декартові принтери, і менш придатні для створення різноманітних об'єктів.

Мала універсальність. Принтери з циліндричною системою координат часто призначені для виконання обмеженого кола завдань, що робить їх менш універсальними для загального використання.

Обмежений асортимент моделей для друку. Через специфіку роботи циліндричних принтерів вони можуть бути менш придатними для друку широкого спектра моделей, що обмежує їх використання у загальному 3D-друці.

Не дивлячись на те що на перший погляд циліндрична система має набагато більше недоліків у порівнянні із декартовою, вона знаходить своє застосування у 3D принтерах. Одним із яскравих прикладів такого застосування є 3D друк без підтримок конічними перетинами під кутом 45° (рисунок 1.5).

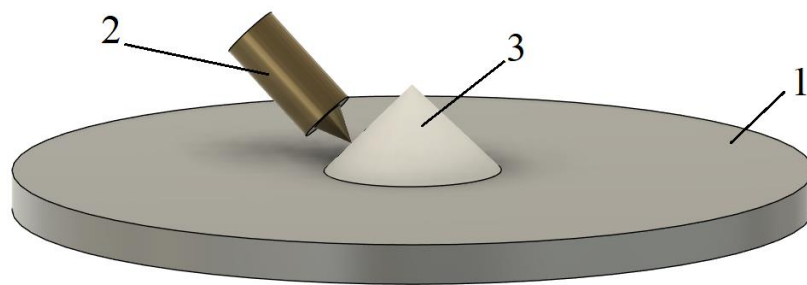


Рисунок 1.5 – Друк конічними перетинами без підтримок

У випадку відсутності додаткової поворотної осі екструдера такого роду друк можливий лише на полярних 3D принтерах.

1.3 Аналіз особливостей полярних 3D принтерів

Полярні 3D-принтери – це типи адитивних виробничих систем, які використовують циліндричну систему координат для переміщення екструдеру та платформи під час створення об'єктів. На відміну від традиційних 3D-принтерів, які працюють у декартовій системі координат (з осями X, Y, Z), полярні 3D-принтери застосовують кутове обертання та радіальне переміщення, що відкриває нові можливості для конструкції та роботи таких пристроїв.

Полярні 3D-принтери використовують дві основні координати для визначення положення друкуючої головки: радіус (R) і кут (θ). Третя координата – висота Z залишається такою ж, як і в інших принтерах. Основні елементи конструкції таких принтерів можна розбити на кілька складових.

Полярні 3D-принтери мають наступну кінематичну структуру. Вони працюють на основі кругових рухів, що істотно відрізняється від традиційної лінійної кінематики. Це досягається через два основні типи руху. Обертальний рух (кутове переміщення θ). У більшості конструкцій цей рух реалізується за допомогою повороту платформи, на якій розміщений об'єкт, що друкується. Екструдер може залишатися нерухомим або також рухатися уздовж радіуса платформи.

Радіальний рух (R) це переміщення екструдеру від центру до краю платформи або навпаки. Зазвичай воно забезпечується рейковим механізмом або системою напрямних. Ці дві координати дозволяють досягати тієї ж точності, що й у декартових принтерах, але з деякими особливостями.

У полярних принтерах платформа зазвичай має круглу форму, оскільки обертання відбувається навколо центра. Для обертання використовуються крокові двигуни. Крім того, поверхня платформи має можливість нагріву для забезпечити краще прилипання матеріалу під час друку.

Друкуюча головка або екструдер в полярному 3D-принтері може бути схожа на екструдери традиційних принтерів і мати функцію руху вгору-вниз уздовж осі Z. При цьому важливо, щоб механізм подачі матеріалу був адаптований для плавного радіального руху, оскільки швидкість переміщення головки в полярних координатах може відрізнятись залежно від поточного радіусу.

Управління полярними 3D-принтерами здійснюється за допомогою спеціальних прошивок і алгоритмів для роботи з полярною системою координат. Програмне забезпечення повинно правильно обробляти геометрію моделей, щоб перетворити декартові координати в полярні. Більшість сучасних прошивок підтримують цей тип принтерів, але для точного налаштування може знадобитися додаткова оптимізація.

Як і в традиційних принтерах, рама полярного 3D-принтера повинна бути жорсткою та стабільною, щоб уникнути вібрацій під час друку. Механічна частина повинна бути адаптована для обертання платформи, тому часто використовуються міцні підшипники та напрямні з низьким коефіцієнтом тертя.

Полярні принтери поки що не мають такого широкого поширення як ті, що працюють у декартовій системі координат. Частина з цих принтерів є результатом проектування DIY майстрів, але і існують промислові 3D принтери.

Так до DIY проектів можна віднести полярний 3D принтер Pimaker 3D [7]. Його конструкцію наведено на рисунку 1.6. Pimaker 3D обрано у якості прикладу для аналізу полярних 3D принтерів через прозорість матеріалів його конструкції, що дає змогу детально розглянути основні компоненти.

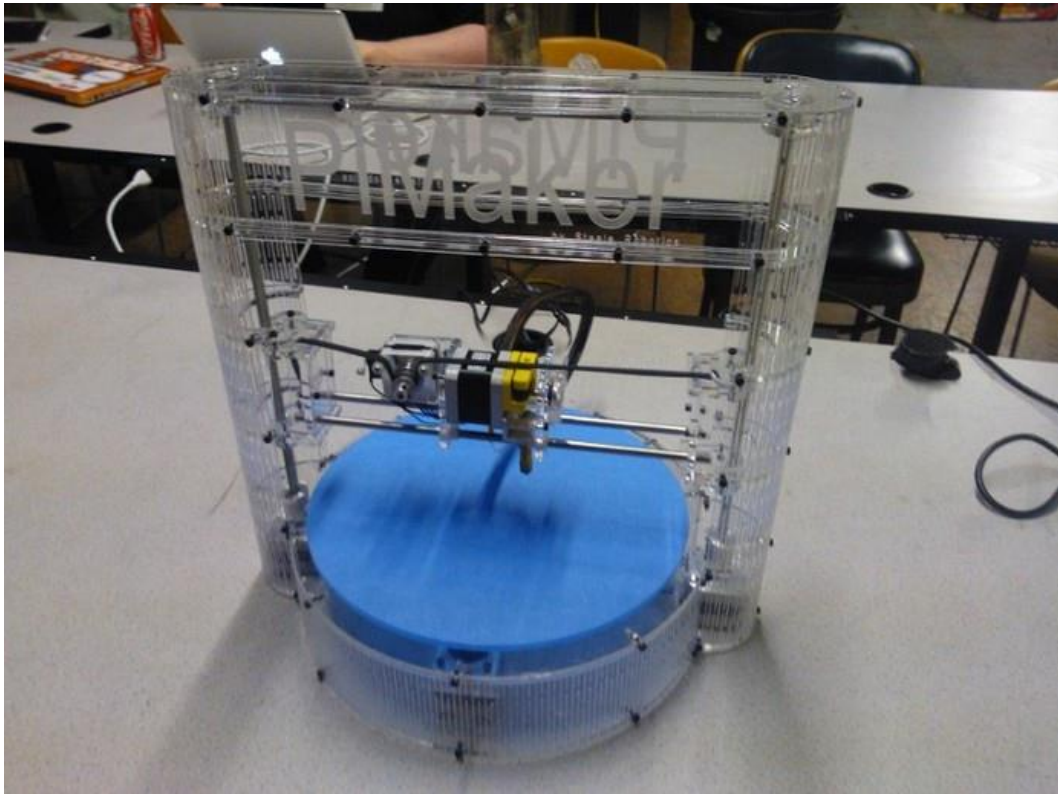


Рисунок 1.6 – Полярний 3D принтер Pimaker 3D

За рух вздовж осі θ в даному принтері відповідає платформа, що обертається [8]. Ця особливість притаманна всім полярним 3D принтерам, що було проаналізовано. З одного боку обертання столу дозволяє забезпечити швидкий рух за координатою θ , що є перевагою даної конструкції. Але з іншого боку така конструкція передбачає переважно велику масу самої платформи, поверхні друку, нагрівачів елементів і як наслідок цей рух має достатньо високу інерцію. Вплив ваги стає більш значущим при збільшенні розмірів робочої платформи. Що у свою чергу ускладнює рух платформи на високих прискореннях та швидкостях і вимагає збільшення часу друку. По-друге розмір кроку переміщення по осі θ значно збільшується при віддаленні по осі R від центру обертання платформи (рисунок 1.7).

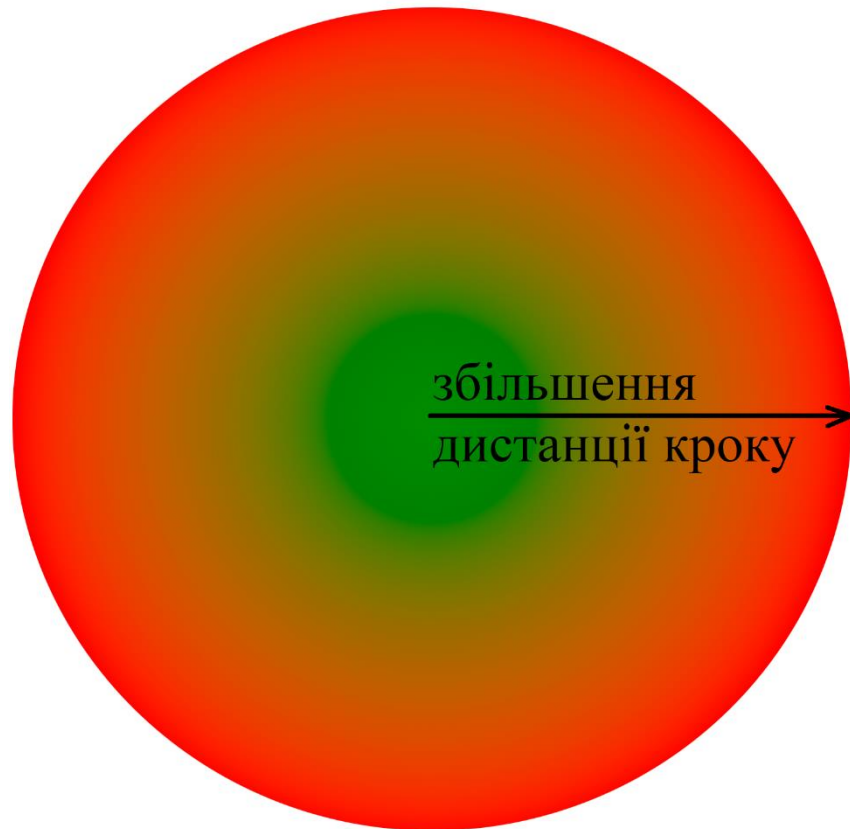


Рисунок 1.7 – Візуалізація зменшення точності переміщення по осі θ

Ця особливість полярного 3D принтеру пов'язана з тим що рух платформи забезпечується кроковим двигуном, іноді із редуктором. Тож при повороті платформи принтеру коли екструдер по осі R знаходиться в зоні, що на рисунку 1.7 зображено зеленим кольором, мінімальна дистанція переміщення при одному кроці двигуна по осі θ буде значно менша ніж при розташуванні екструдера у «червоній» зоні. Таким чином робоча платформа має різну дискретність і як наслідок точність переміщення у різних частинах робочої зони.

Переміщення вздовж осі R забезпечується лінійним переміщенням екструдера вздовж осі, що має проходити через центр обертання платформи. Конструкційне виконання даної осі може бути різним: з опорою з двох боків або у вигляді консолі. Даний тип лінійного переміщення може відбуватися на високій швидкості та прискоренні, в залежності від типу направляючих та

приводу, маси екструдеру та жорсткості конструкції. Найбільший вплив на точність друку має значення відхилення реального положення осі R відносно від ідеальної осі, що проходить через центр обертання платформи (рисунок 1.8). Навіть при максимально точному проектуванні і зборці принтеру ці відхилення будуть присутні, вони будуть змінюватися навіть при заміні сопла екструдеру. Ці зміщення можуть бути достатньо невеликими, до 1 мм, але вони призведуть до значної деформації виробу, що друкується.

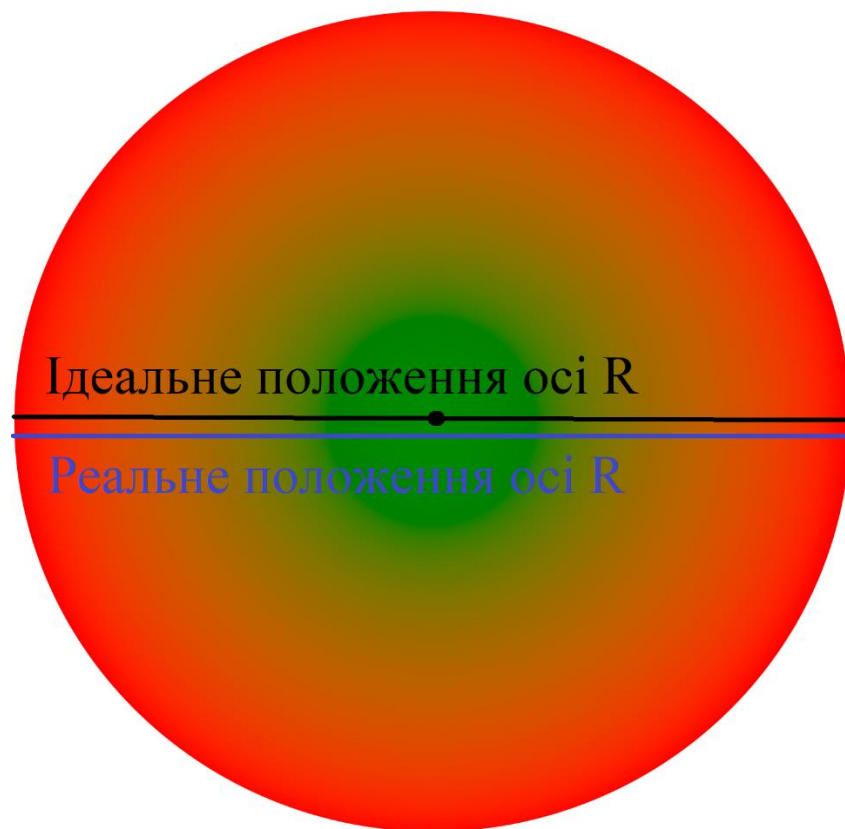


Рисунок 1.8 – Зміщення реального положення осі R

Деформації моделі будуть тим більше чим далі від нульової координати осі R знаходиться екструдер. Якщо у декартових 3D принтерів це призводить лише до зміщення моделі в робочому просторі, то у полярних 3D принтерів це призведе до зміни розмірів і форми моделі.

Окремим випадком такого типу виродження моделі, що друкується є невірне визначення нульової координати осі R (рисунок 1.9), це також приведе до деформацій як у попередньому випадку.

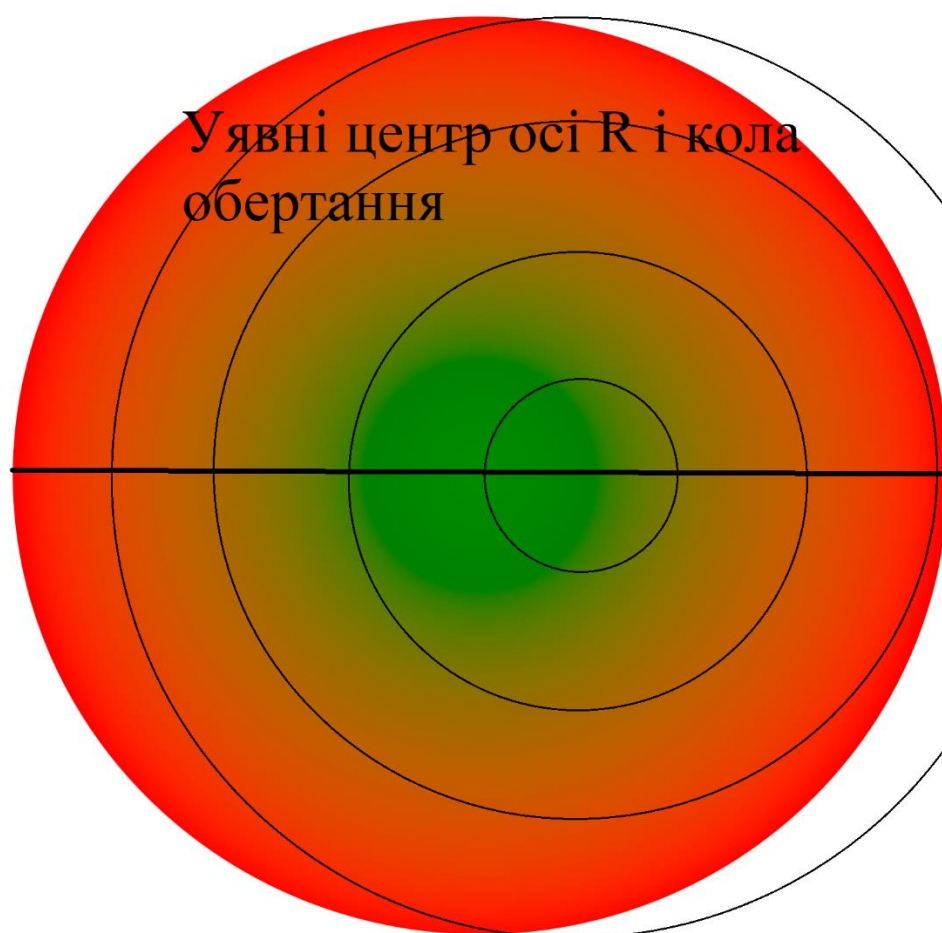


Рисунок 1.9 – Зміщення нульової координати осі R

Це порушення циліндричної системи координат може бути викликане неточністю встановлення датчиків початку координат 3D принтеру – кінцевих вимикачів. Перед початком друку більшість принтерів виконує команду g-коду G28 – рух у «домашнє» положення, яке визначається положенням кінцевих вимикачів із урахуванням оффсетів – зміщень «нуля» відносно положення датчиків. У більшості випадків це питання вирішується в процесі калібровки принтеру після його виготовлення.

Ось Z забезпечує вертикальне переміщення екструдера 3D принтеру. У випадку Primer 3D даний рух забезпечується двома гвинтовими передачами та двома кроковими двигунами. Даний тип вертикального переміщення не є унікальним, притаманним лише полярним 3D принтерам. Але через те що крокових двигуна використовується два, у випадку перекосу лівої і правої кареток осі Z виникає втрата паралельності між рухом сопла екструдера і поверхнею платформи друку, рисунок 1.10.

Якщо у випадку декартових 3D принтерів даний дефект вирішується або механічним налаштуванням рівня платформи, або створенням калібровочної сітки платформи, тобто програмним корегуванням осі Z в процесі переміщення сопла в горизонтальній площині. То у випадку з полярними принтерами необхідно або робити автоматичне вирівнювання лівого и правого крокових двигунів, або створювати сітку столу.

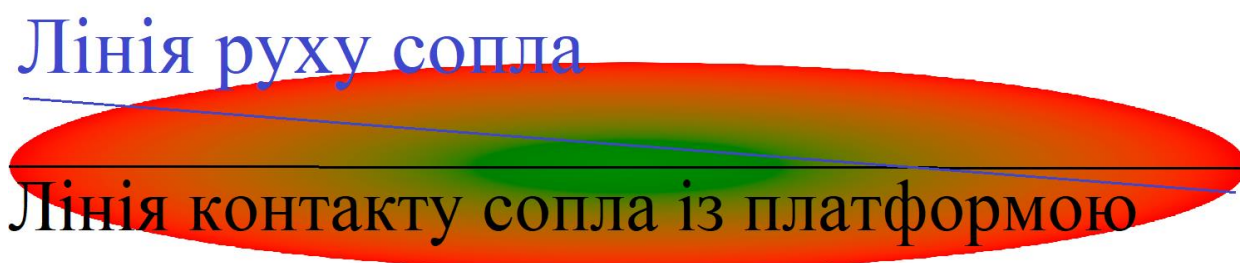


Рисунок 1.10 – Вертикальний перекося осі Z

Якщо даний тип дефекту принтеру не усунути, це може призвести до гальмування оборотів столу навколо осі θ , що у свою чергу може призвести до пропуску кроків крокового двигуна і браку друку.

Іншим конструкційні рішення полярних принтерів, які працюють в циліндричній системі координат є принтер Polar 3D [7]. Не дивлячись на

роботу в одній системі координат даний принтер має принципові відмінності в своїй конструкції, рисунок 1.11. Так екструдер рухається лише за координатою Z , а стіл забезпечує рух за координатами θ – обертання та R – лінійне переміщення.

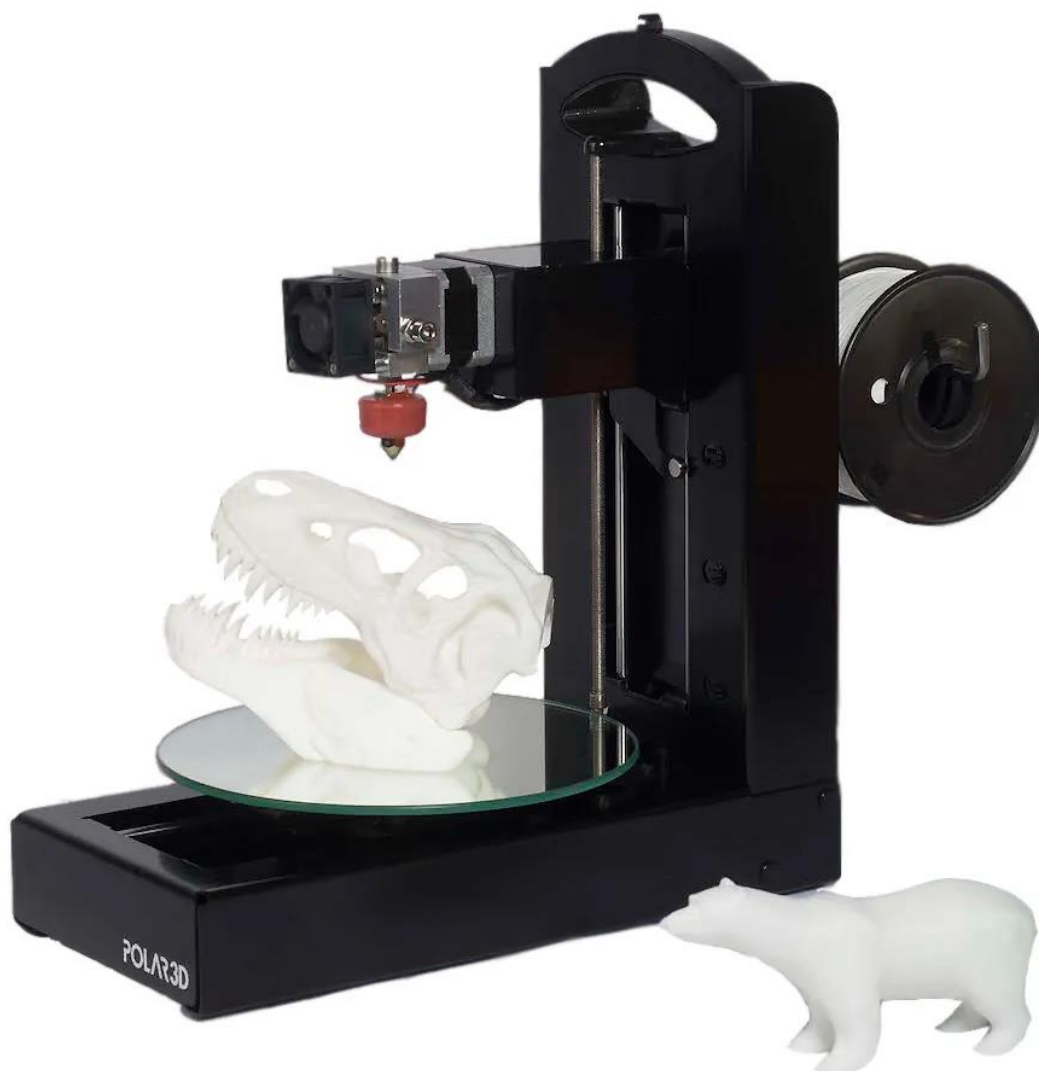


Рисунок 1.11 – 3D принтер Polar 3D

Комбіноване переміщення платформи вимагає високої точності та стабільності її рухів та малої ваги самої платформи. Таким чином виготовлення принтерів з великим розміром робочої зони, що працюють за такою кінематикою є вкрай складним, так як разом із розміром платформи

збільшиться її вага та ускладниться забезпечення рівномірності її переміщення.

За переміщення вздовж осі Z відповідає один кроковий двигун, екструдер переміщується лише вгору та вниз. Тож дефект, наведений на рисунку 1.6 не притаманний даній конструкції. Але все інші, вже наведені, дефекти в повній мірі виникають в даному принтері.

Інші види полярних принтерів мають схожі конструкційні реалізації, або їх комбінації.

1.4 Висновки до першого розділу

Надивлячись на велику кількість недоліків циліндрична система координат, у якій працюють сучасні полярні 3D принтери, має значні переваги. Ці переваги повною мірою можна оцінити у випадку реалізації 3D друку кінчними перетинами без підтримок.

Головні проблеми реалізації полярних 3D принтерів є забезпечення точного і коректного переміщення навколо осі θ циліндричної системи координат, тобто обертання платформи друку. Умовно ці проблеми можна поділити на ті, що можна виправити за допомогою електронного модулю системи центрування принтеру та на програмну корекцію G-коду. Саме цим питанням присвячені наступні розділи роботи.

2 РОЗРОБКА МЕТОДІВ ПРОГРАМНОГО КОРЕКТУВАННЯ G-КОДУ У ВИПАДКАХ ПОМИЛКИ ЦЕНТРУВАННЯ ПЛАТФОРМИ 3D ПРИНТЕРУ

2.1 Генерація g-коду для полярних 3D принтерів

Сучасні САМ системи для 3D друку – «слайсери» генерують g-коду у декартовій системі координат. Для прикладу розглянемо слайсер Ultimaker Cura, як один із найбільш поширених [9].

Даний слайсер дозволяє використовувати як стандартні принтери, що занесено у його базу даних так і налаштовувати власні кастомні рішення. У даній САМ-системі є можливість створення 3D принтерів із круглою формою робочої платформи та центром координат у центрі платформи (рисунок 2.1), але g-код генерується все одно у декартовій системі координат, про що свідчать осі координат. Всі перерахунки систем координат забезпечуються вже безпосередньо в прошивці принтеру.

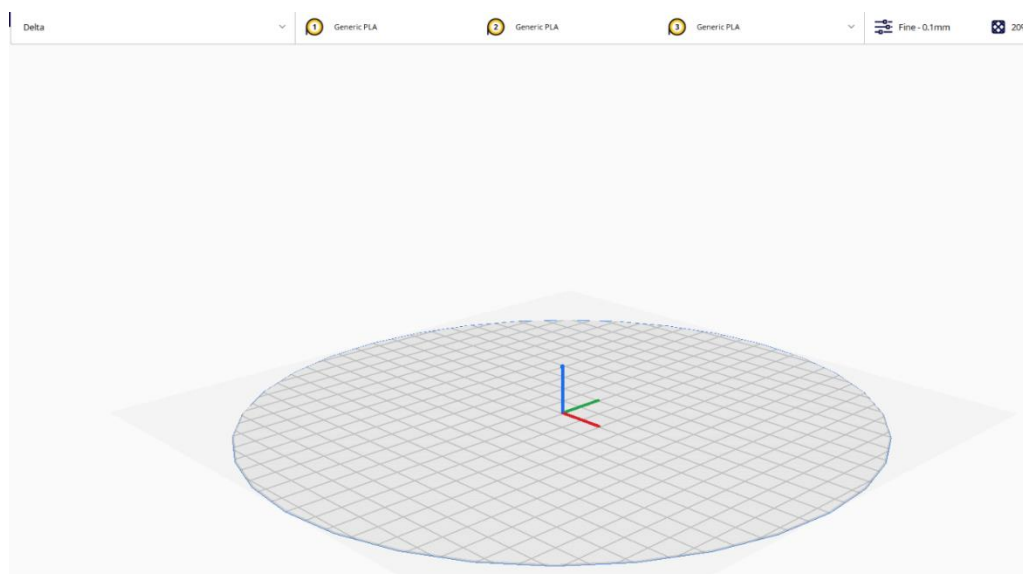


Рисунок 2.1 – Налаштування САМ системи для полярного принтеру

Таким чином в слайсері генерується g-код для звичайного декартового 3D принтеру (рисунок 2.2).

```

1 ;FLAVOR:Marlin
2 ;TIME:4608
3 ;Filament used: 11.2164m
4 ;Layer height: 0.3
5 ;MINX:46.3
6 ;MINY:49.3
7 ;MINZ:0.2
8 ;MAXX:153.7
9 ;MAXY:150.7
10 ;MAXZ:17.4
11 ;TARGET_MACHINE.NAME:Unknown
12 ;Generated with Cura_SteamEngine 5.6.0
13 M140 S50
14 M105
15 M190 S50
16 M104 S235
17 M105
18 M109 S235
19 M82 ;absolute extrusion mode
20 G28 ;Home
21 G1 Z15.0 F6000 ;Move the platform down 15mm
22 ;Prime the extruder
23 G92 E0
24 G1 F200 E3
25 G92 E0
26 G92 E0
27 G92 E0
28 G1 F2700 E-4.5
29 ;LAYER_COUNT:58
30 ;LAYER:0
31 M107
32 G0 F4500 X48.604 Y51.545 Z0.3
33 ;TYPE:SKIRT
34 G1 F2700 E0
35 G1 F3600 X49.768 Y50.594 E0.07499
36 G1 X51.143 Y49.866 E0.15261
37 G1 X52.627 Y49.425 E0.22985
38 G1 X54.067 Y49.301 E0.30196
39 G1 X145.905 Y49.3 E4.88377
40 G1 X147.402 Y49.432 E4.95875

```

Рисунок 2.2 – Лістинг G-коду згенерованого для полярного 3D принтеру

Якщо перші 31 строки команд з коментарями є універсальними для будь-якого 3D принтеру, за виключенням строки 21:

G1 Z15.0 F6000 ;Move the platform down 15mm,

яка рухає платформу вертикально вгору на 15 мм, але дані вісі у декартовій та циліндричній системах координат мають однакову назву та напрямок.

Починаючи із 32 строки можна чітко побачити що наступні команди робочих G1 та холостих G0 ходів використовують декартову систему координат. Наприклад:

G0 F4500 X48.604 Y51.545 Z0.3,

При виконанні якої екструдер виконує холостий рух у точку з координатами (X 48,604мм, Y 51,545мм, Z 0,3мм).

То ж фактично всі перетворення декартової системи координат у циліндричну виконуються безпосередньо платою керування 3D принтера.

2.2 Аналіз прямого використання циліндричної системи координат у ЧПУ верстатах

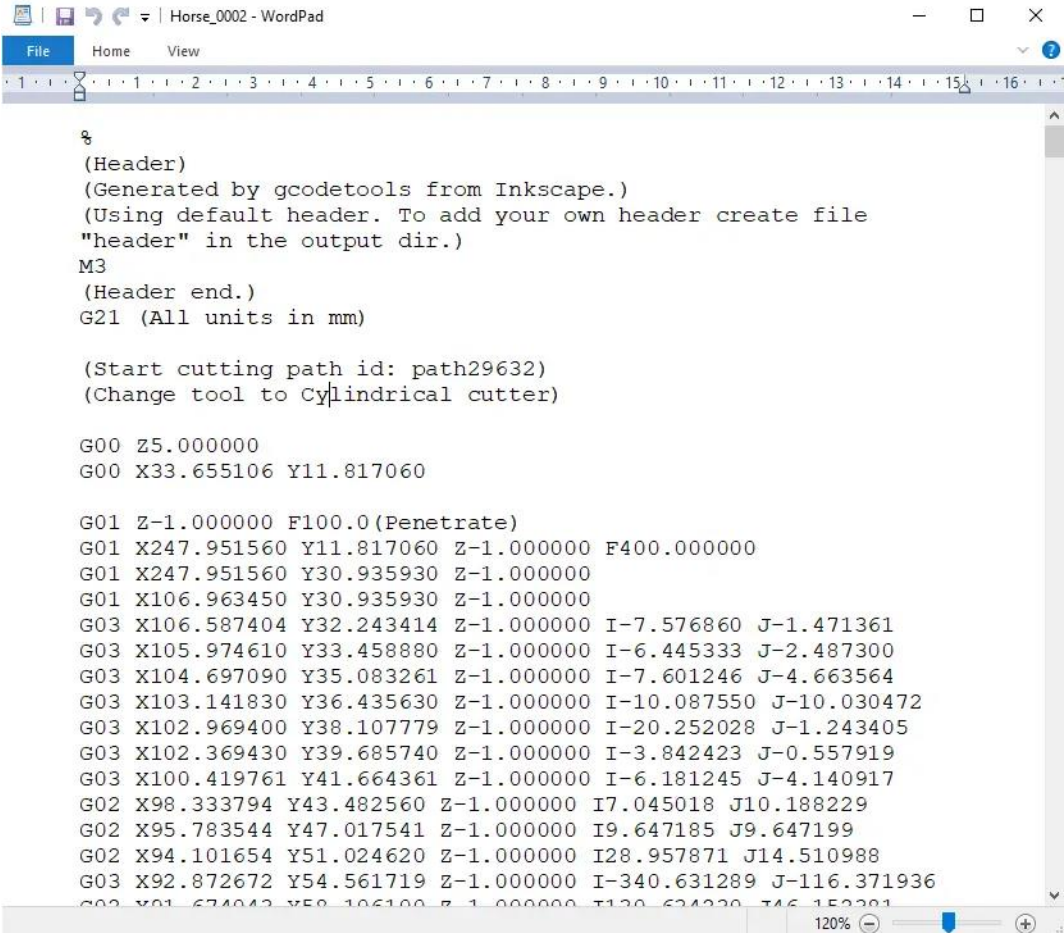
На відміну від 3D принтерів багатоосьові фрезерні і токарні ЧПУ верстати використовують циліндричні системи координат і G-код для них може генеруватися безпосередньо у ній.

Необхідно розглянути основну структуру G-коду.

G-код – це мова програмування для верстатів із числовим програмним керуванням. G-код означає геометричний код. Команди G-коду вказують машині, куди рухатися, як швидко рухатися і яким шляхом слідувати.

У разі верстата, такого як, наприклад, токарний верстат або фрезерний верстат, різальний інструмент приводиться в дію цими командами, щоб дотримуватися певної траєкторії інструменту, вирізуючи матеріал для отримання бажаної форми. Так само у разі використання 3D-принтерів команди G-коду наказують машині наносити матеріал шар за шаром, формуючи точну геометричну форму [11].

G-код для фрезерного верстата, що згенеровано в Inkscape, наведений на рисунку 2.3.



```

%
(Header)
(Generated by gcodetools from Inkscape.)
(Using default header. To add your own header create file
"header" in the output dir.)
M3
(Header end.)
G21 (All units in mm)

(Start cutting path id: path29632)
(Change tool to Cylindrical cutter)

G00 Z5.000000
G00 X33.655106 Y11.817060

G01 Z-1.000000 F100.0(Penstrate)
G01 X247.951560 Y11.817060 Z-1.000000 F400.000000
G01 X247.951560 Y30.935930 Z-1.000000
G01 X106.963450 Y30.935930 Z-1.000000
G03 X106.587404 Y32.243414 Z-1.000000 I-7.576860 J-1.471361
G03 X105.974610 Y33.458880 Z-1.000000 I-6.445333 J-2.487300
G03 X104.697090 Y35.083261 Z-1.000000 I-7.601246 J-4.663564
G03 X103.141830 Y36.435630 Z-1.000000 I-10.087550 J-10.030472
G03 X102.969400 Y38.107779 Z-1.000000 I-20.252028 J-1.243405
G03 X102.369430 Y39.685740 Z-1.000000 I-3.842423 J-0.557919
G03 X100.419761 Y41.664361 Z-1.000000 I-6.181245 J-4.140917
G02 X98.333794 Y43.482560 Z-1.000000 I7.045018 J10.188229
G02 X95.783544 Y47.017541 Z-1.000000 I9.647185 J9.647199
G02 X94.101654 Y51.024620 Z-1.000000 I28.957871 J14.510988
G03 X92.872672 Y54.561719 Z-1.000000 I-340.631289 J-116.371936
G03 X91.674043 Y58.106180 Z-1.000000 I-130.624230 J46.152281

```

Рисунок 2.3 – G-код для фрезерного верстата

Так команда G01 X247.951560 Y11.817060 Z-1.000000 F400.000000 має наступну структуру:

– G01 означає «переміститися по прямій лінії у певне положення»;

- далі оголошується координата зі значеннями X, Y та Z;
- за допомогою значення F встановлюється швидкість подачі або швидкість, з якою виконуватиметься переміщення виконавчого механізму.

Підсумовуючи можна сказати, що рядок G01 X247.951560 Y11.817060 Z-1.000000 F400 вказує фрезеру рухатися по прямій від поточного положення до координат X247.951560, Y11.817060 і Z00.

Необхідно розглянути найбільш поширені команди G-коду. G00 – швидке позиціонування. Команда G00 (рисунок 2.4) переміщує виконавчий механізм з максимальною швидкістю переміщення з поточного положення у вказану точку або координати, вказані командою. Верстат переміщатиме всі осі одночасно, тому вони здійснюють переміщення одночасно. Це призводить до прямолінійного руху до нової точки положення.

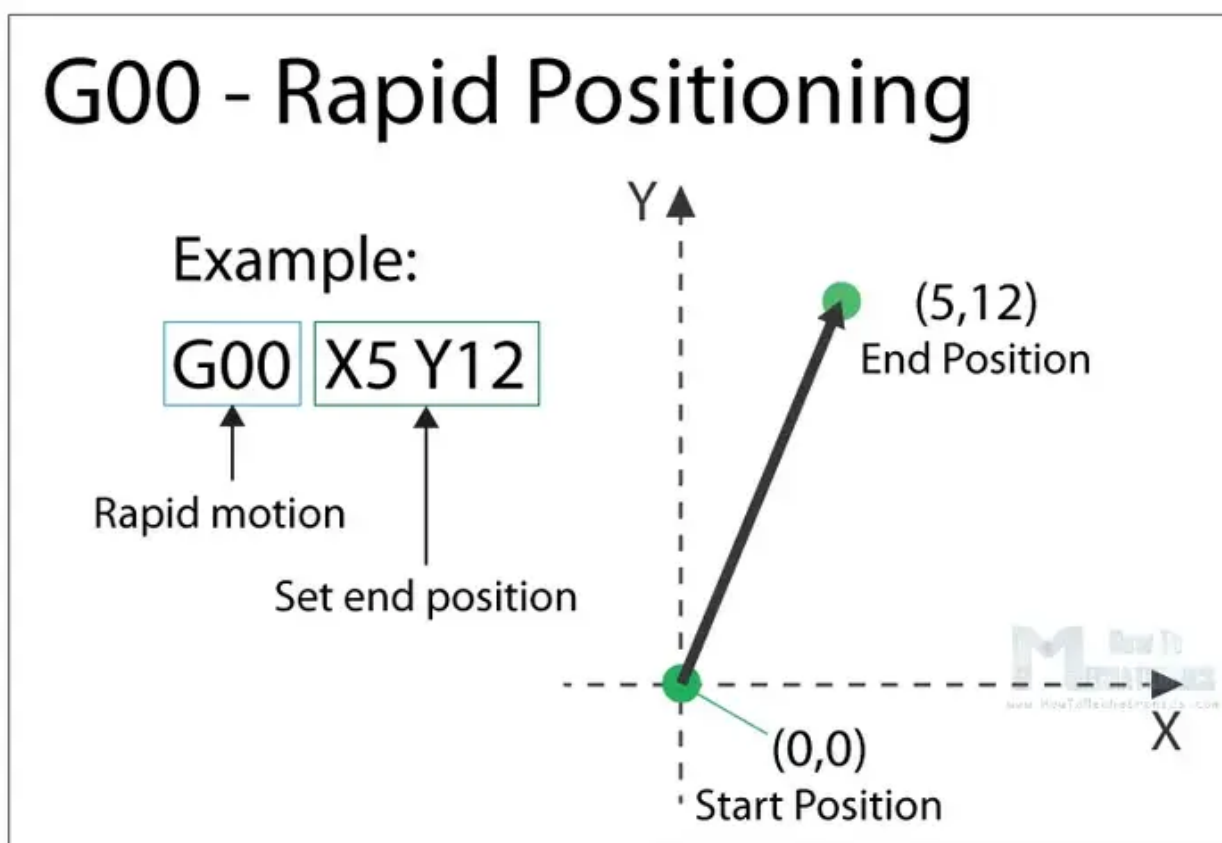


Рисунок 2.4 – Команда G00

G01 – лінійна інтерполяція. Команда G-коду G01 дає вказівку верстату рухатися по прямій лінії із заданою швидкістю подачі або швидкістю різання. Вказується кінцева позиція із значеннями X, Y і Z, а швидкість – значенням F. Контролер машини обчислює (інтерполює) проміжні точки, через які потрібно пройти, щоб отримати цю пряму лінію.

На відміну від команди G00, яка використовується тільки для позиціонування, команда G01 (рисунок 2.5) використовується, коли верстат виконує свою основну роботу. У разі токарного верстата або фрезерного верстата різання матеріалу по прямій лінії, а у разі 3D-принтера екструдування матеріалу по прямій лінії.

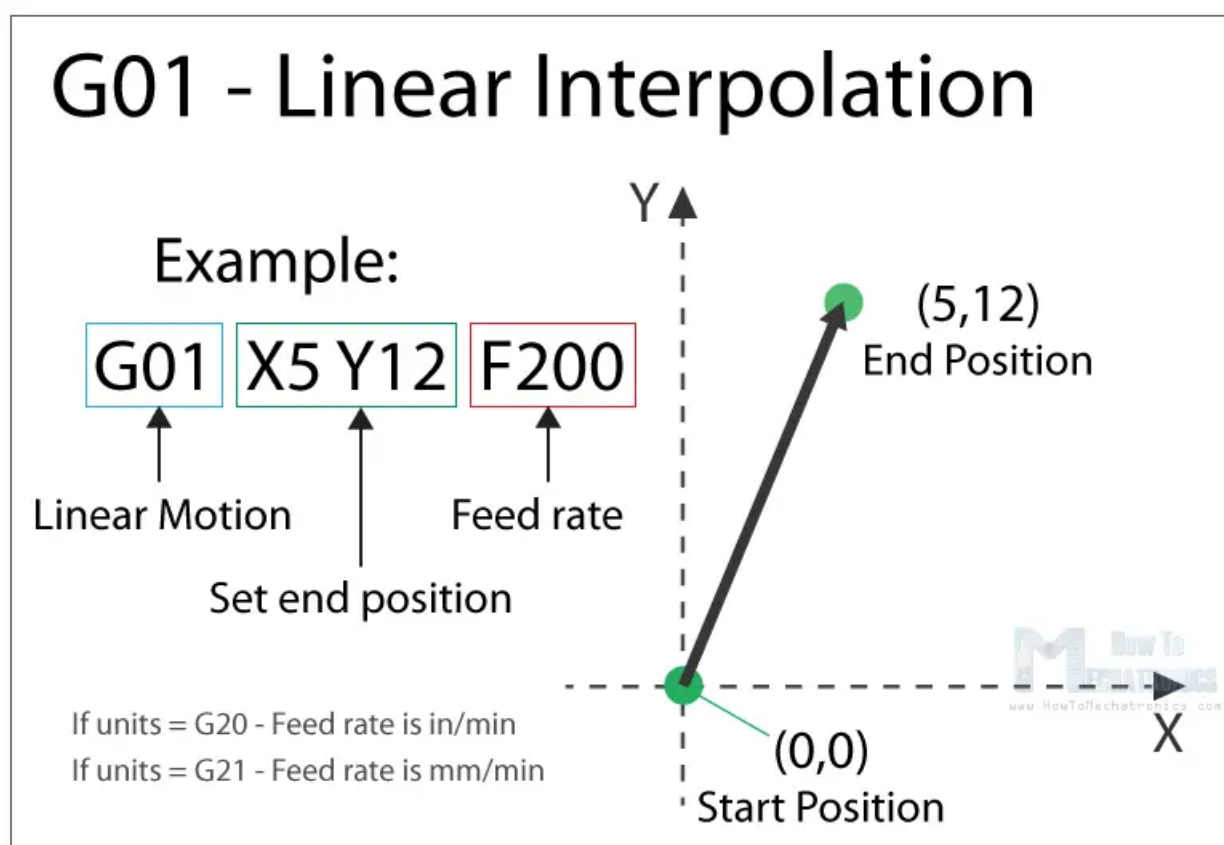


Рисунок 2.5 – Команда G01

G02 – кругова інтерполяція за годинниковою стрілкою. Команда G02 вказує верстату рухатися за годинниковою стрілкою по колу. Це та ж концепція, що й у команди G01 і вона використовується при виконанні

відповідного процесу обробки. Крім параметрів кінцевої точки, тут також необхідно визначити центр обертання або відстань початкової точки дуги від центральної точки дуги. Початкова точка є кінцевою точкою попередньої команди або поточною точкою. Ця команда як раз і імітує рух в циліндричній системі координат.

У першому прикладі наведено команду G01, яка переміщає верстат у точку X5, Y12. Це буде відправною точкою для G02. За допомогою параметрів X та Y команди G02 встановлюється кінцева точка. Щоб дістатися цієї кінцевої точки, використовуючи круговий рух або дугу, потрібно визначити її центральну точку. Використовуються параметри I та J. Значення I та J відносяться до початкової або кінцевої точки попередньої команди. Отже, щоб отримати центральну точку X5 та Y7, потрібно зробити зміщення 0 по осі X та зміщення -5 по осі Y. Роботу команди G02 представлено на рисунку 2.6.

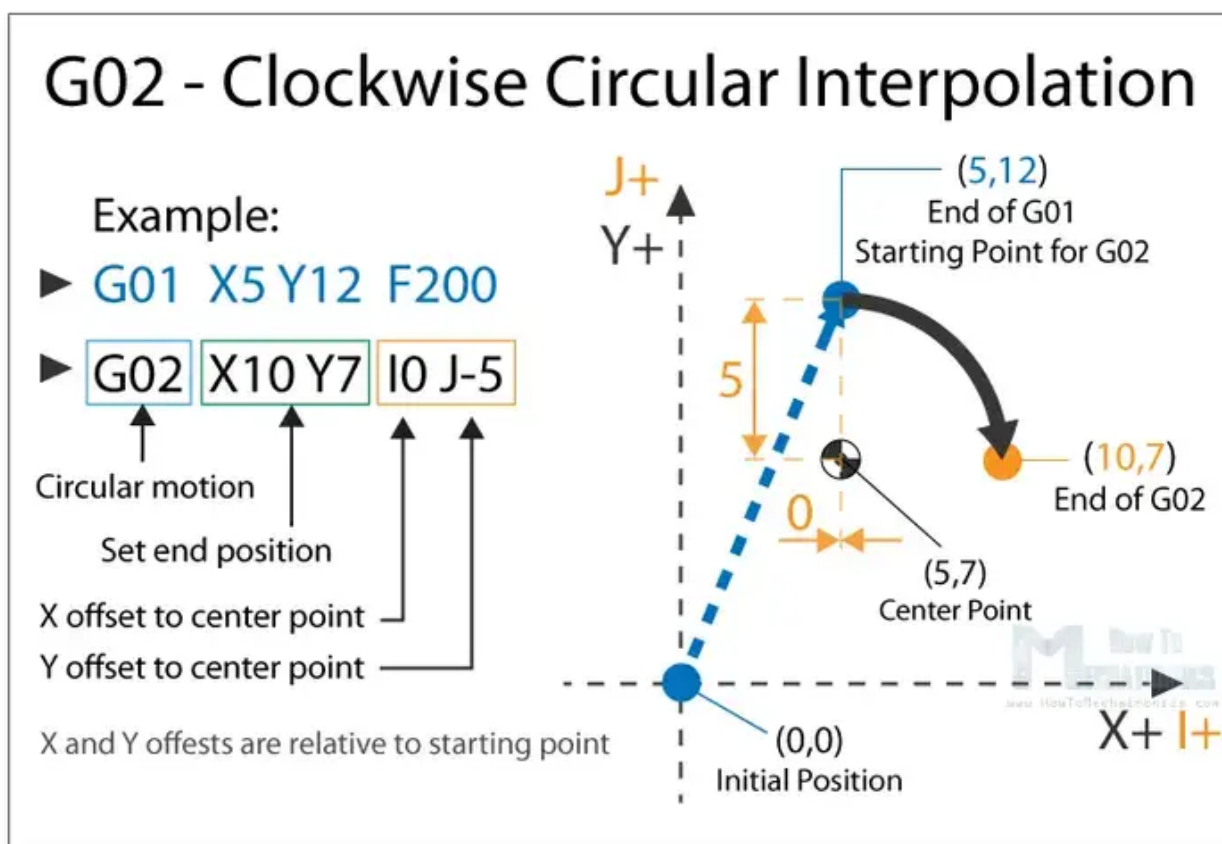


Рисунок 2.6 – Команда G02

2.3 Розробка моделі перетворення G-коду для полярного 3D принтеру

В результаті аналізу, що проведено у попередньому розділі можна зробити висновок що 3D принтери також можна адаптувати для більш повного використання команд G-коду. Спробуємо перевести G-коди, що згенеровані стандартними слайсерами у циліндричну систему координат.

Перехід від декартової системи координат до циліндричної є одним з основних способів зміни системи координат для опису фізичних явищ і геометричних об'єктів. Цей підхід часто використовується в тих випадках, коли форма або симетрія об'єкта підказує, що використання декартової системи є неефективним або складним. У багатьох практичних завданнях, особливо в тривимірному моделюванні, технічних розрахунках та роботі з G-кодом (особливо в контексті обробки або друку циліндричних або обертових об'єктів), перетворення декартових координат в циліндричні значно спрощує опис процесів.

Для введення циліндричної системи координат у просторі вибирається площина (основна площина) і на ній визначається полярна система координат з полюсом O і полярною віссю Ox . Через точку O перпендикулярно до основної площини проведемо вісь Oz (вісь аплікат) і виберемо її напрямком так, щоб зростання полярного кута, що спостерігається з боку позитивного напрямку осі Oz , відбувалося проти годинникової стрілки (рисунок 2.7) [10].

У циліндричній системі координат положення точки M , що не належить осі аплікат, характеризується полярними координатами r , θ точки M_0 – ортогональної проекції точки M на основну площину, і аплікати z – координатою точки M_z – ортогональної проекції точки M на вісь аплікат. Таким чином, циліндричні координати точки M – це впорядкована трійка чисел r , θ , z – полярний радіус ($r \geq 0$), полярний кут ($-\pi < \theta \leq \pi$) і аплікату ($-\infty$

$z < \infty$). У точок, що належать осі аплікату, не визначено полярний кут, вони задаються вказівкою нульового полярного радіусу і аплікати.

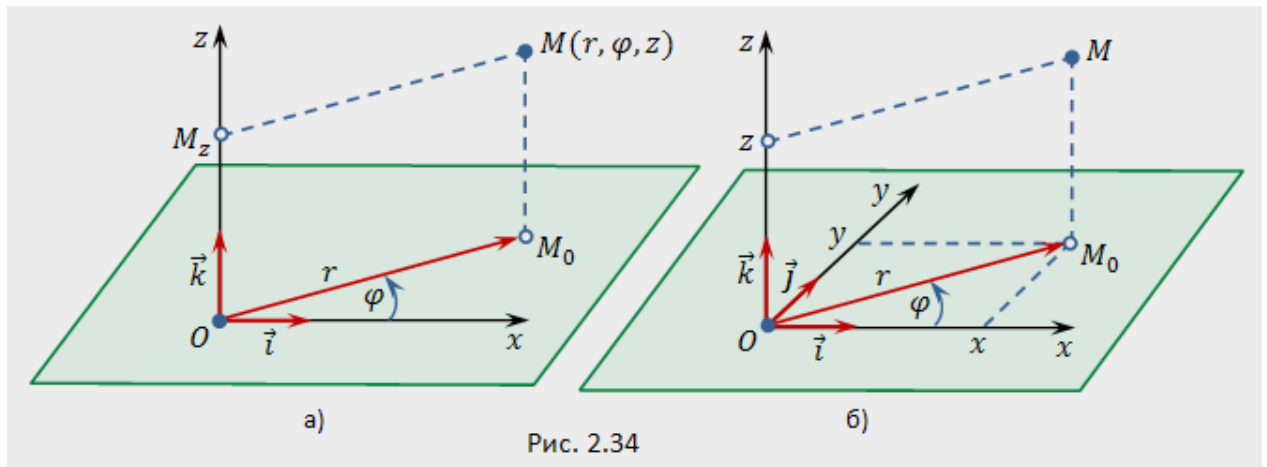


Рис. 2.34

Рисунок 2.7 – Перехід від декартової системи координат до циліндричної

З циліндричною системою координат $O_{r\theta z}$ можна пов'язати прямокутну систему координат O_{ijk} , у якої початок і базисні вектори \vec{i} , \vec{k} збігаються з початком циліндричної системи координат та одиничними векторами на полярній осі та осі аплікат відповідно, а базисний вектор \vec{j} вибирається так, щоб трійка \vec{i} , \vec{k} , \vec{j} була правою (при цьому базис виявляється стандартним).

Навпаки, якщо просторі задана права прямокутна система координат, то, прийнявши позитивну піввісь абсцис за полярну вісь, отримаємо циліндричну систему координат (пов'язану з даною прямокутною).

Оскільки аплікату z точки M у прямокутній системі координат і аплікату z у циліндричній системі координат збігаються, то формули, що зв'язують між собою прямокутні координати x, y, z точки M та її циліндричні координати r, θ, z мають вигляд, наступний:

$$\begin{cases} x = r \times \cos \theta \\ y = r \times \sin \theta \\ z = z. \end{cases} \quad (2.1)$$

Ці формули переходу дозволяють знайти прямокутні координати за відомими циліндричними. Зворотний перехід виконується за формулами:

$$r = \sqrt{x^2 + y^2}, \quad (2.2)$$

$$\cos \theta = \frac{x}{\sqrt{x^2 + y^2}}, \quad (2.3)$$

$$\sin \theta = \frac{y}{\sqrt{x^2 + y^2}}, \quad (2.4)$$

$$z = z. \quad (2.5)$$

Головне значення полярного кута ($-\pi < \theta \leq \pi$) знаходиться за формулами
 рисунок 2.8.

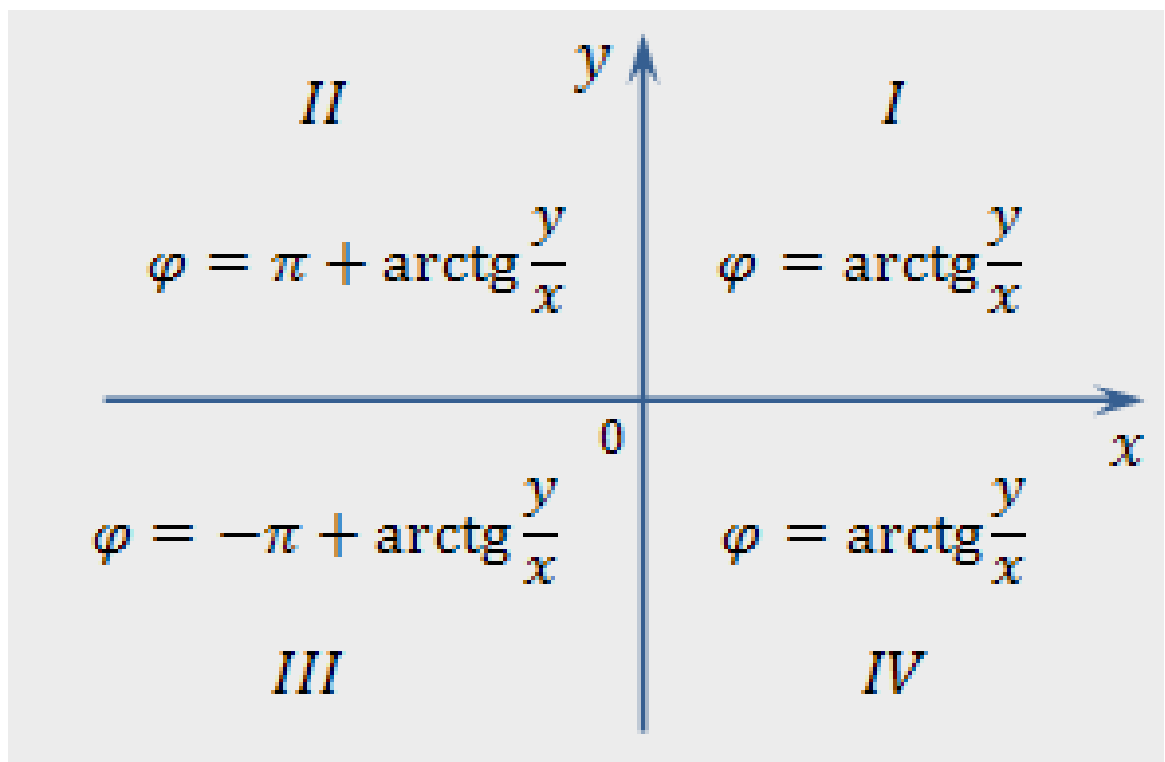


Рисунок 2.8 – Знаходження кута θ

Для автоматизації переводу декартового G-коду у циліндричний розроблено програмний код на Python [12];

```
import math
```

Ця бібліотека надає математичні функції, такі як обчислення квадратного кореня та арктангенса, які необхідні для перетворення координат [13].

```
def cartesian_to_cylindrical(x, y):
    # Перетворення декартових координат в циліндричні
    r = math.sqrt(x**2 + y**2) # Радіальна координата
    theta = math.degrees(math.atan2(y, x)) # Кутова координата в градусах
    return r, theta
```

Ця функція перетворює декартові координати (X, Y) у циліндричні (R, θ). Для цього:

- r (радіус) обчислюється як відстань від початку координат до точки (X, Y);
- θ (кут) обчислюється через арктангенс співвідношення y/x і перетворюється з радіан в градуси.

```
def convert_gcode_line(line):
    # Конвертуємо одну стрічку G-коду
    if 'X' in line and 'Y' in line:
        # Витягуємо X і Y з рядка
        x_start = line.find('X') + 1
        y_start = line.find('Y') + 1
        z_start = line.find('Z') if 'Z' in line else -1

        # Зчитуємо значення X і Y
        x_val = float(line[x_start:line.find(' ', x_start)])
        y_val = float(line[y_start:line.find(' ', y_start)])
```

```

# Перетворення в циліндричні координати
r, theta = cartesian_to_cylindrical(x_val, y_val)

# Формуємо нову строку з радіальними і кутовими координатами
new_line = line.replace(f'X{x_val}', f'R{r:.3f}').replace(f'Y{y_val}',
f'T{theta:.3f}')

# Якщо є Z, додаємо його
if z_start != -1:
    z_val = float(line[z_start + 1:line.find(' ', z_start)]) if ' ' in
line[z_start:] else float(line[z_start + 1:])
    new_line = new_line.replace(f'Z{z_val}', f'Z{z_val:.3f}')

return new_line
return line

```

Ця функція виконує такі дії:

- перевіряє, чи присутні координати X та Y у рядку;
- витягує значення X та Y за допомогою методу `find()`;
- викликає функцію `cartesian_to_cylindrical()`, щоб перетворити X та Y у координати R та θ ;
- замінює X і Y на R і θ відповідно (R – радіус, θ – кут);
- якщо є координата Z, то додається її значення без змін;
- повертає модифікований рядок із новими координатами.

```
def convert_gcode_file(input_file, output_file):
```

```

# Конвертація файлу G-коду
with open(input_file, 'r') as infile, open(output_file, 'w') as outfile:
    for line in infile:

```

```

new_line = convert_gcode_line(line.strip())
outfile.write(new_line + '\n')

```

Ця функція:

- відкриває вхідний файл G-коду для читання і вихідний файл для запису;
- для кожного рядка вхідного файлу викликає функцію `convert_gcode_line()`, щоб конвертувати координати;
- записує модифікований рядок у вихідний файл.

Використання

```

input_gcode = 'input_gcode_file.txt' # Вхідний файл G-коду
output_gcode = 'output_cylindrical_gcode.txt' # Вихідний файл для
циліндричного G-коду

convert_gcode_file(input_gcode, output_gcode)

print(f"G-код конвертовано та збережено у файл {output_gcode}")

```

У даній частині програми задаються імена файлів для входу та виходу, після чого програма викликає функцію `convert_gcode_file()`, щоб виконати конвертацію всього файлу. Завершення операції підтверджується виведенням повідомлення.

Функція `cartesian_to_cylindrical`: перетворює декартові координати X і Y в радіальну і кутову координати (RRR і TTT).

Функція `convert_gcode_line`: перетворює одну лінію G-коду. Вона шукає координати X і Y , потім обчислює нові радіальні й кутові координати та формує нову лінію G-коду. Також зберігає координату Z без змін.

Функція `convert_gcode_file`: читає G-код з вхідного файлу і записує перетворені рядки у новий файл. Приклад використання G-коду з координатами в декартовій системі, програма зчитує його і створює новий файл з відповідними координатами в циліндричній системі.

Важливо зазначити, що в кодї перетворюються лише команди, що містять X та Y координати, але його можна розширити під ваші конкретні потреби.

2.4 Розробка програмного модулю автоматичної заміни координат у G-кодї

Згідно аналізу, що проведено у першому розділі у деяких випадках неможливо виправити друк полярного 3D принтеру лише за допомогою системи пошуку центру обертання платформи. Наприклад коли ось R зміщено відносно центру обертання платформи. У такому разі необхідно розробити програмний модуль для автоматичної корекції G-коду.

Щоб розрахувати зміни координат у циліндричній системі координат після зміщення нуля осі R, можна використовувати геометричні або аналітичні методи на основі векторної геометрії. Нижче наведено кроки, як це зробити.

Припустимо, що початок координат осі R зміщено на вектор $\vec{d}=(d_x,d_y)$ у площині xy. Це означає, що новий центр для вимірювання радіальної координати знаходиться в точці $(d_x,d_y,0)$.

Нехай точка має координати в старій (початковій) циліндричній системі:

- R_{old} – радіальна координата (відстань від старого центру);
- ϕ_{old} – кутова координата;
- z – координата по осі z (вона не змінюється при зміні осі R).

Перейдемо до декартових координат:

$$x_{old}=R_{old}\cos(\varphi_{old}), \quad (2.6)$$

$$y_{old}=R_{old}\sin(\varphi_{old}). \quad (2.7)$$

У новій системі координат, зміщеній на вектор \vec{d} нові координати в декартовій системі будуть визначені як:

$$x_{new}=x_{old}-d_x, \quad (2.8)$$

$$y_{new}=y_{old}-d_y. \quad (2.9)$$

Тепер ці нові x_{new} та y_{new} використовуються для обчислення нових координат у циліндричній системі.

Радіальна координата R_{new} :

$$R_{new} = \sqrt{x_{new}^2 + y_{new}^2}. \quad (2.10)$$

Це відстань від нового центру до точки.

Кутова координата φ_{new} :

$$\Phi_{new}=\text{atan2}(y_{new},x_{new}). \quad (2.11)$$

Функція atan2 дозволяє обчислити кут правильно у всіх чотирьох квадрантах.

Координата z залишається незмінною, оскільки зміщення осі R відбувається тільки у площині xy .

Розглянемо приклад розрахунку. Припустимо, що початкові координати точки були $R_{old}=5$, $\varphi_{old}=\pi/4$, $z=3$, а зміщення центру обертання дорівнює $\vec{d}=(2,1)$.

Спочатку знайдемо початкові декартові координати точки:

$$x_{old} = 5 \cos\left(\frac{\pi}{4}\right) = 5 \frac{\sqrt{2}}{2} = 3,54,$$

$$y_{old} = 5 \sin\left(\frac{\pi}{4}\right) = 3,54.$$

Тепер віднімаємо зміщення:

$$x_{new} = 3,54 - 2 = 1,54,$$

$$y_{new} = 3,54 - 1 = 2,54.$$

Обчислюємо новий радіус і кут:

$$R_{new} = \sqrt{1,54^2 + 2,54^2} = \sqrt{2,37 + 6,45} = \sqrt{8,82} = 2,97,$$

$$\varphi_{new} = \text{atan2}(2,54, 1,54) \approx 1,03 \text{ радіан.}$$

Отже, нові координати точки: $R_{new} \approx 2,97$ радіан, $\varphi_{new} \approx 1,03$ радіан, $z = 3$.

Таким чином, після зміщення осі координати змінюються на основі нової відстані від центру та нового кута. Це дозволяє ефективно виконувати перетворення в циліндричних координатах.

Наведений метод корекції координат реалізовано у вигляді програми на мові програмування Python [8]:

```
import math
```

```
# Функція для перерахунку координат у нову циліндричну систему
def cylindrical_to_new_cylindrical(R_old, phi_old, z, d_x, d_y):
```



```
"""
```

Перерахунок координат в нову циліндричну систему після зміщення центру на (d_x, d_y) .

```
:param R_old: Початкова радіальна координата
```

```
:param phi_old: Початковий кут у радіанах
```

```
:param z: Координата по осі z
```

```
:param d_x: Зміщення по осі x
```

```
:param d_y: Зміщення по осі y
```

```
:return: Нові координати (R_new, phi_new, z)
```

```
"""
```

```
# Перетворення циліндричних координат у декартові
```

```
x_old = R_old * math.cos(phi_old)
```

```
y_old = R_old * math.sin(phi_old)
```

```
# Зміщення по осі x та y
```

```
x_new = x_old - d_x
```

```
y_new = y_old - d_y
```

```
# Нові циліндричні координати
```

```
R_new = math.sqrt(x_new**2 + y_new**2)
```

```
phi_new = math.atan2(y_new, x_new) # atan2 повертає кут у всіх  
квadrантах
```

```
# Координата z залишається тією самою
```

```
return R_new, phi_new, z
```

```
# Вхідні дані
```

```

R_old = 5      # Початкова радіальна координата
phi_old = math.pi / 4 # Початковий кут у радіанах (45 градусів)
z = 3         # Координата по осі z
d_x = 2       # Зміщення по осі x
d_y = 1       # Зміщення по осі y
# Виклик функції

R_new, phi_new, z_new = cylindrical_to_new_cylindrical(R_old, phi_old,
z, d_x, d_y)

# Виведення результатів
print(f"Нова радіальна координата: R_new = {R_new:.2f}")
print(f"Новий кут: phi_new = {phi_new:.2f} радіан")
print(f"Координата по осі z: z = {z_new}")

```

Програма, що розроблено працює наступним чином.

Для її роботи необхідні наступні вхідні параметри:

- R_old: початкова радіальна координата;
- phi_old: початковий кут у радіанах;
- z: початкова координата по осі z (незмінна);
- d_x, d_y: зміщення по осях x та y відповідно.

Алгоритм роботи програми наступний:

- спочатку циліндричні координати перетворюються у декартові x та y;
- змінюються координати x та y з урахуванням зміщення нуля осі R;
- розраховуються нові радіальні та кутові координати;
- координата z залишається незмінною.

В результаті виконання алгоритму програма виводить нові значення радіальної координати Rnew, кутової координати φnew та незмінної координати Z.

2.5 Висновки до другого розділу

У сучасних САМ системах для 3D друку використовується декартові системи координат і G-код генерується з командами, що переміщують екструдер і платформу саме у декартовій системі координат. Але згідно аналізу, що проведено у першому розділі в реальних полярних 3D принтерах можливі зміщення окремих вузлів відносно осей координат, що призводить до виникнення браку виробів в процесі 3D друку. Корегувати такі помилки зборки 3D принтерів набагато простіше у циліндричній системі координат.

Тож у даному розділі запропоновано математичну модель переходу від декартової системи до циліндричної, а також розроблено програму – перетворювач декартового G-коду у циліндричний, якій використовується у багатьох фрезерних та токарних ЧПУ верстатах.

Також у розділі реалізовано програмну корекцію G-коду для випадків зсуву нуля координати R відносно центру обертання платформи.

3 РОЗРОБКА СИСТЕМИ АВТОМАТИЧНОГО ПОШУКУ ЦЕНТРУ ОСІ θ

3.1 Розробка конструкції системи пошуку центру платформи

Для автоматичного пошуку центра осі θ пропонується система, що працює наступним чином. По периметру робочої платформи, що має форму кола розташовано 16 неодимових магнітів, ця кількість може бути зміненою. Менша кількість магнітів пришвидшить процес пошуку центру платформи та зробить платформу трохи легше, більша – підвищить точність системи. Обчислення буде провести легше і точніше при парній кількості магнітів. На каретці осі R розташовується датчик Холла таким чином щоб його чутливий елемент знаходився чітко у на одній вертикальній осі із отвором екструдеру.

Для розуміння алгоритму роботи системи необхідно навести візуалізацію алгоритму. Позначення частин системи умовне, так екструдер та датчик Холла зображене як єдине ціле, а за чутливий елемент датчику приймається вершина конусу екструдеру, умовне сопло. На рисунку 3.1: 1 – екструдер із датчиком Холлу, 2 – платформа друку, 3 – система магнітів.

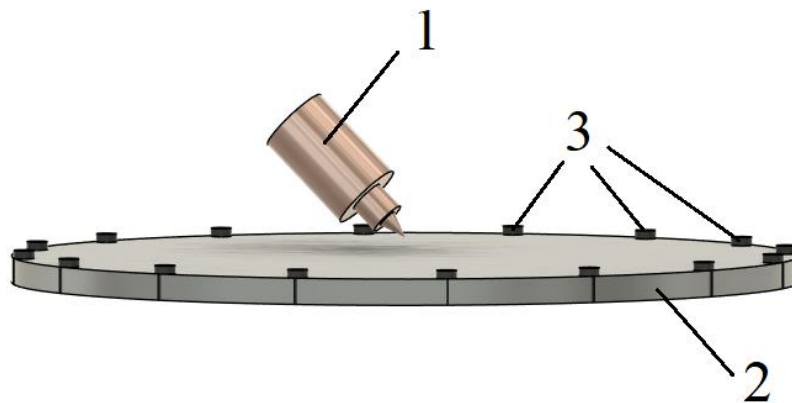


Рисунок 3.1 – Частини системи

На початку процесу пошуку каретка осі R рухається до спрацювання кінцевого вимикача, який знаходиться поза межами робочої платформи (рисунок 3.2).

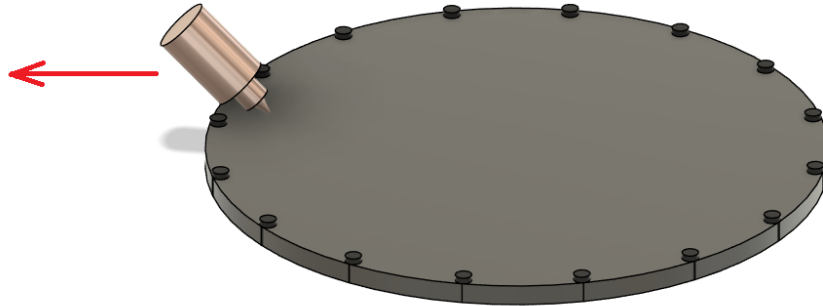


Рисунок 3.2 – Рух екструдера до кінцевого вимикачу осі R

На другому кроці екструдер рухається на один крок по напрямку до платформи (рисунок 3.3). Розмір кроку руху визначається конструкцією принтеру, розмірами платформи, налаштуваннями чутливості датчику Холла, магнітним полем магнітів.

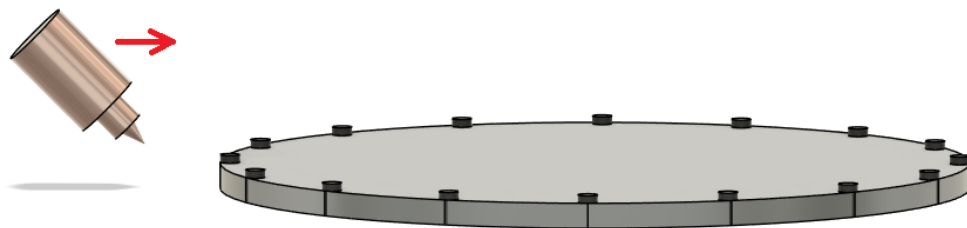


Рисунок 3.3 – Рух екструдера до платформи на один крок

Через те, що положення платформи, а саме магнітів, може бути не визначеним, на наступному кроці алгоритму платформа починає обертатися

покроково на кут $22,5^\circ$, це відстань між сусідніми магнітами платформи. Обертання відбувається або до спрацювання датчику Холла, тобто коли датчик фіксує наявність магніту, або на повний кут – $22,5^\circ$. У першому випадку алгоритм переходить до виконання наступного кроку, у другому – до попереднього. Що можна спостерігати на рисунку 3.4.

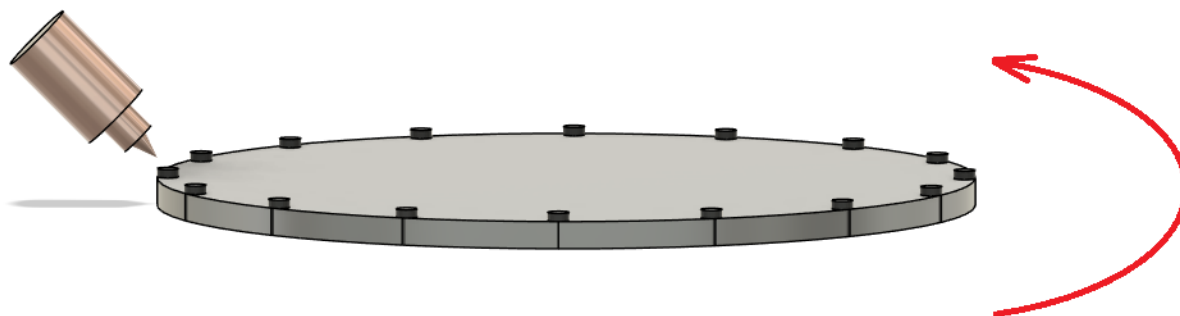


Рисунок 3.4 – Обертання платформи

На четвертому кроці відбувається налаштування датчику Хола на максимальне магнітне поле від магнітів (рисунок 3.5). Тобто відбувається переміщення по осі R та осі θ доки сигнал на датчику не досягне необхідного рівня. Рівень визначається чутливістю датчику та типом магніту. Координати центру магніту фіксуються системою.

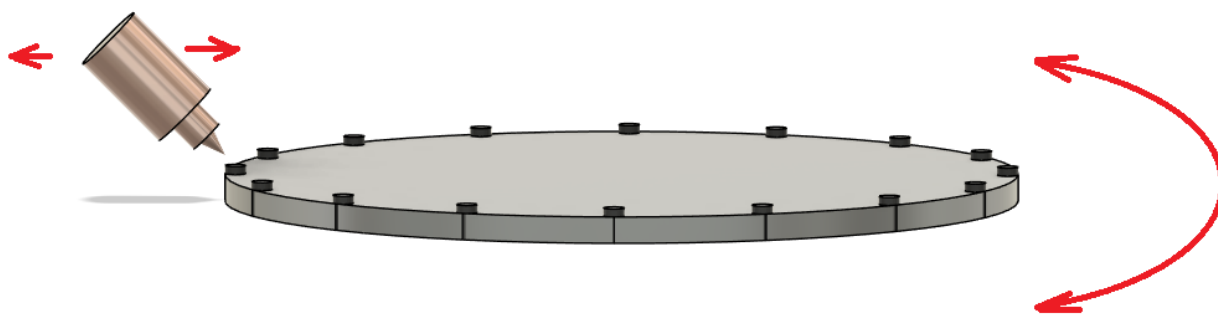


Рисунок 3.5 – Налаштування датчику на центр магніту

Після чого кроки три та чотири алгоритму повторюються для всіх магнітів платформи. З результатами пошуку центру магнітів будується карта платформи (рисунок 3.6). І знаходиться центр обертання.

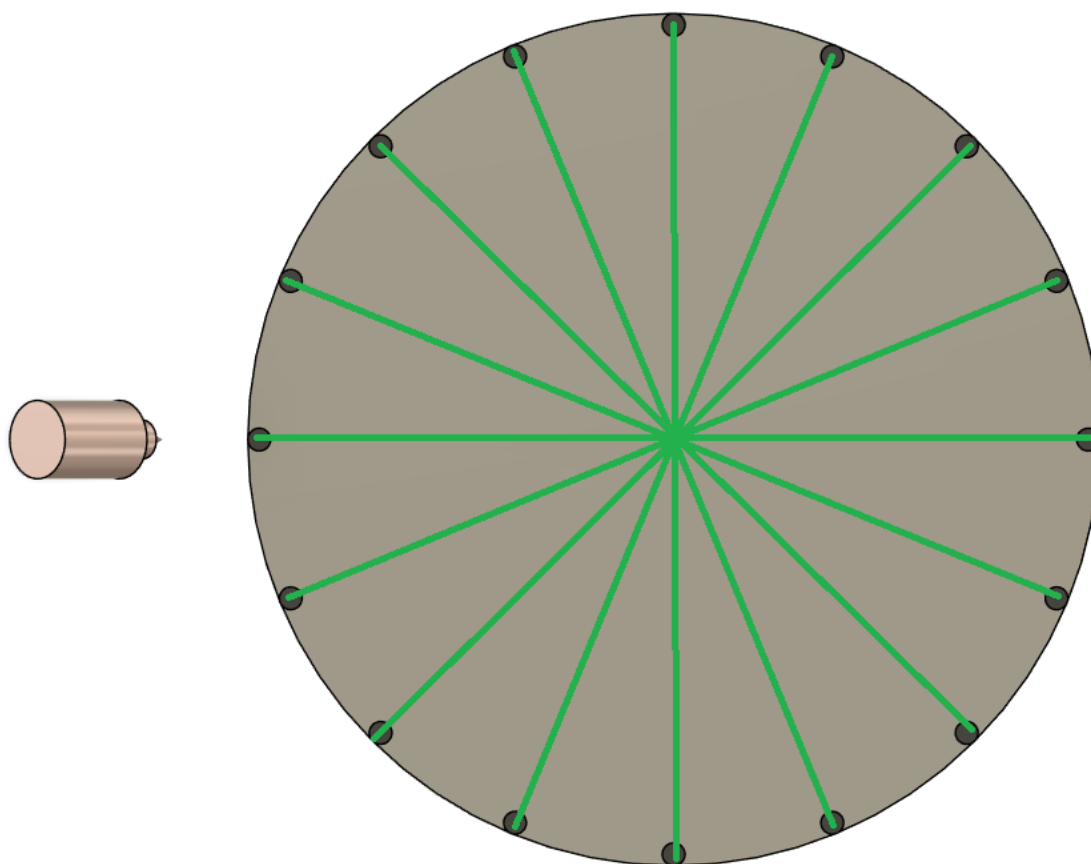


Рисунок 3.6 – Побудова карти платформи для знаходження центру

Кількість магнітів не може бути занадто великою для запобігання їх взаємному впливу на магнітне поле. Датчик Холла в конструкції є зйомним, тобто використовують для визначення так званих офсетів осі R, Офсет – відтань від датчику кінцевого положення до нуля системи координат.

Система, що розроблюється є універсальною до будь-якого полярного 3D принтеру. Але для візуалізації та кращого розуміння необхідно навести зовнішній вигляд механічних та електричних компонентів системи. У якості принтера, до якого адоптується система пошуку центру осі θ обрано принтер, що розробляється в рамках лабораторії 3D прототипування кафедри КІТАР

Харківського національного університету радіоелектроніки студентом Кальченко А.С. в рамках виконання магістерської атестаційної роботи за темою «Розробка полярного 3D принтеру з можливістю друку без технологічних підтримок» (рисунок 3.7 - 3.8).



Рисунок 3.7 – Зовнішній вигляд 3D принтеру

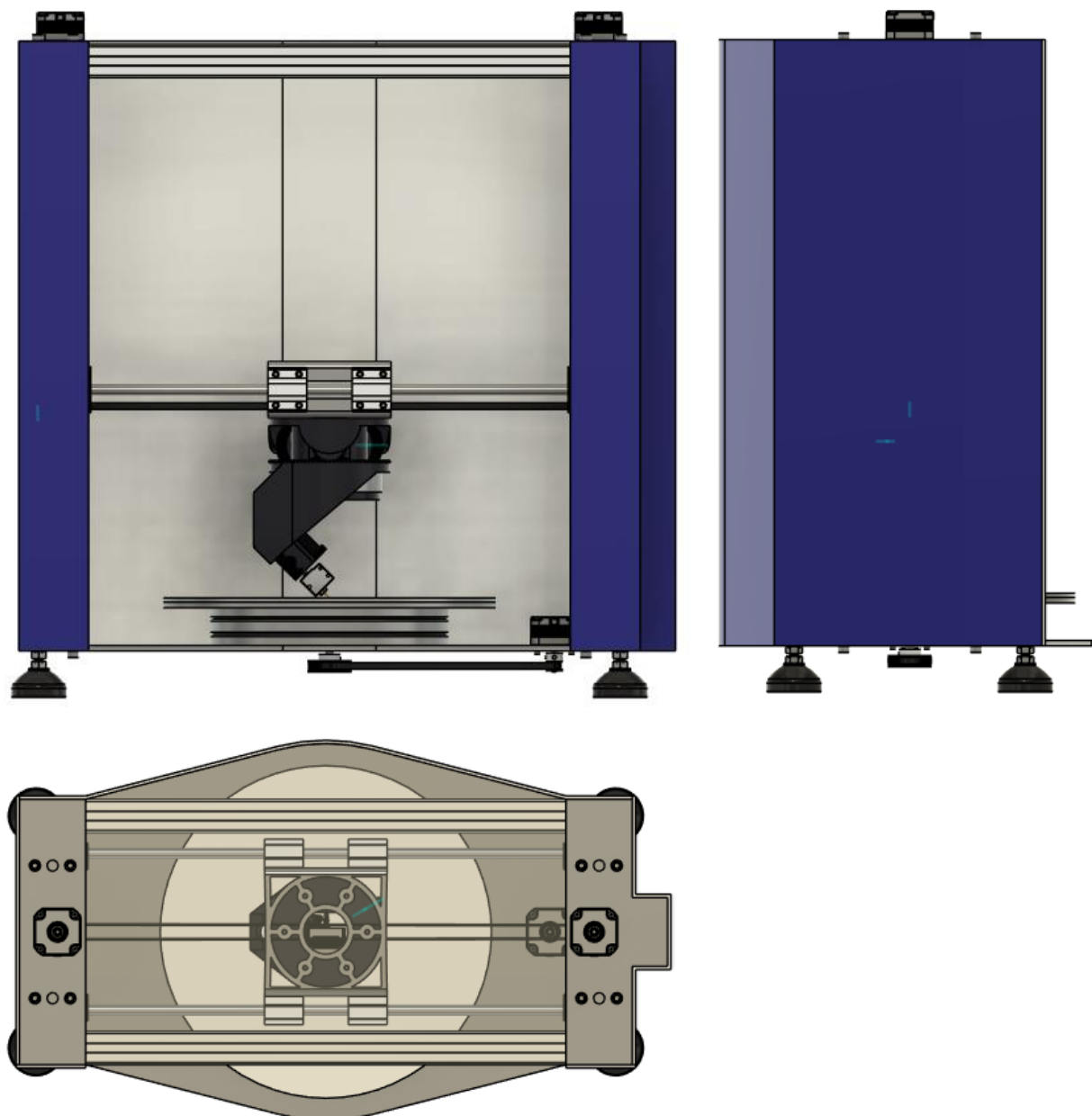


Рисунок 3.8 – Ескіз 3D принтеру

Дану конструкцію допрацьовано наступним чином: додані магнітів по периметру поверхні платформи, на екструдері зафіксовано кронштейн датчику Холла та сам датчик.

На рисунку 3.9 наведено зовнішній вигляд платформи з урахуванням змін. 3D моделі створено у Autodesk Fusion [14].

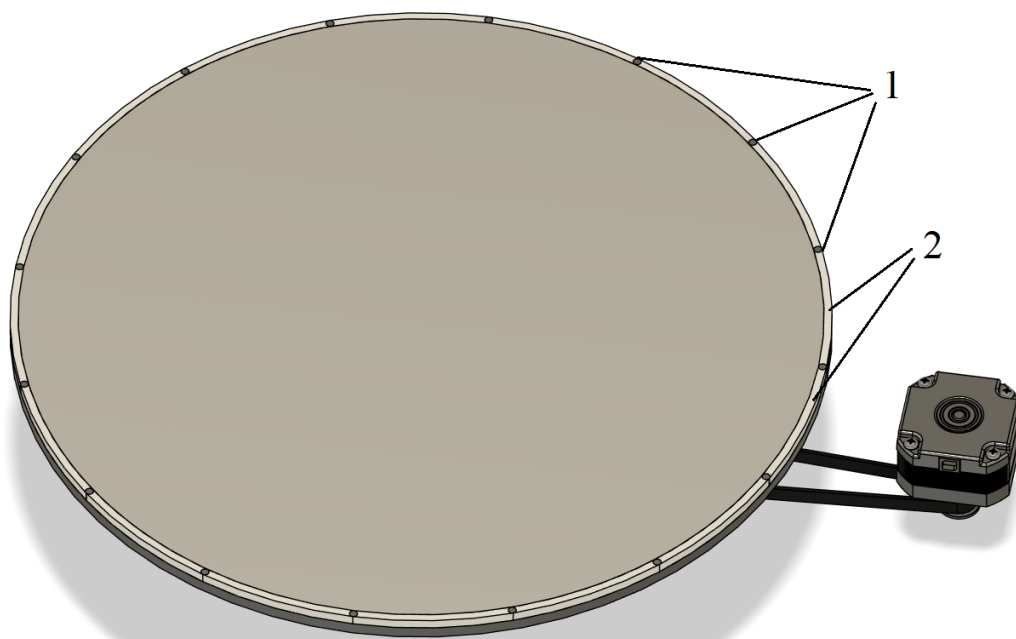


Рисунок 3.9 – Розташування магнітів на платформі 3D принтеру

На рисунку 3.9 зображено: 1 – неодимові магніти, 2 – пластиковий спейсер. Пластиковий спейсер використовується для позиціонування магнітів і забезпечення рівномірності їх розташування (рисунок 3.10).

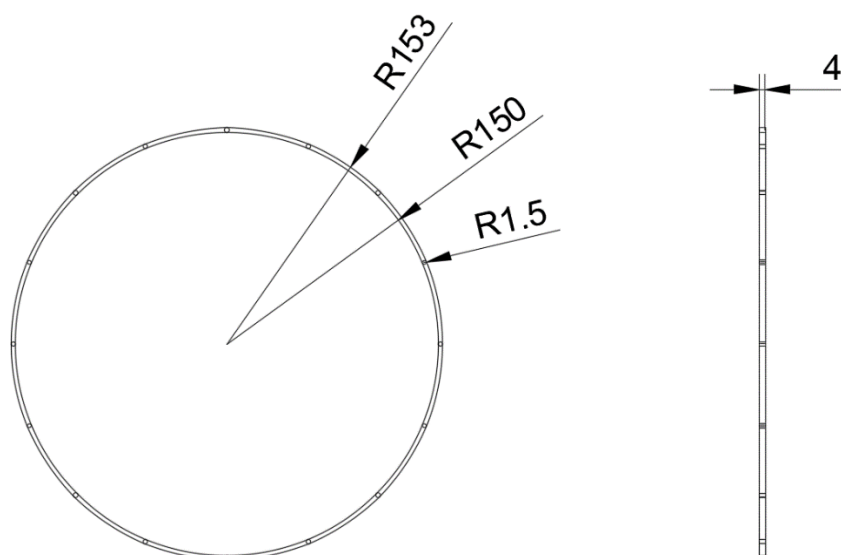


Рисунок 3.10 – Спейсер магнітів

Магніти передбачається використовувати циліндричної форми із розмірами: діаметр 3 мм, висота 5 мм.

На рисунку 3.11 наведено систему фіксації датчику Холла на каретці екструдера [15].

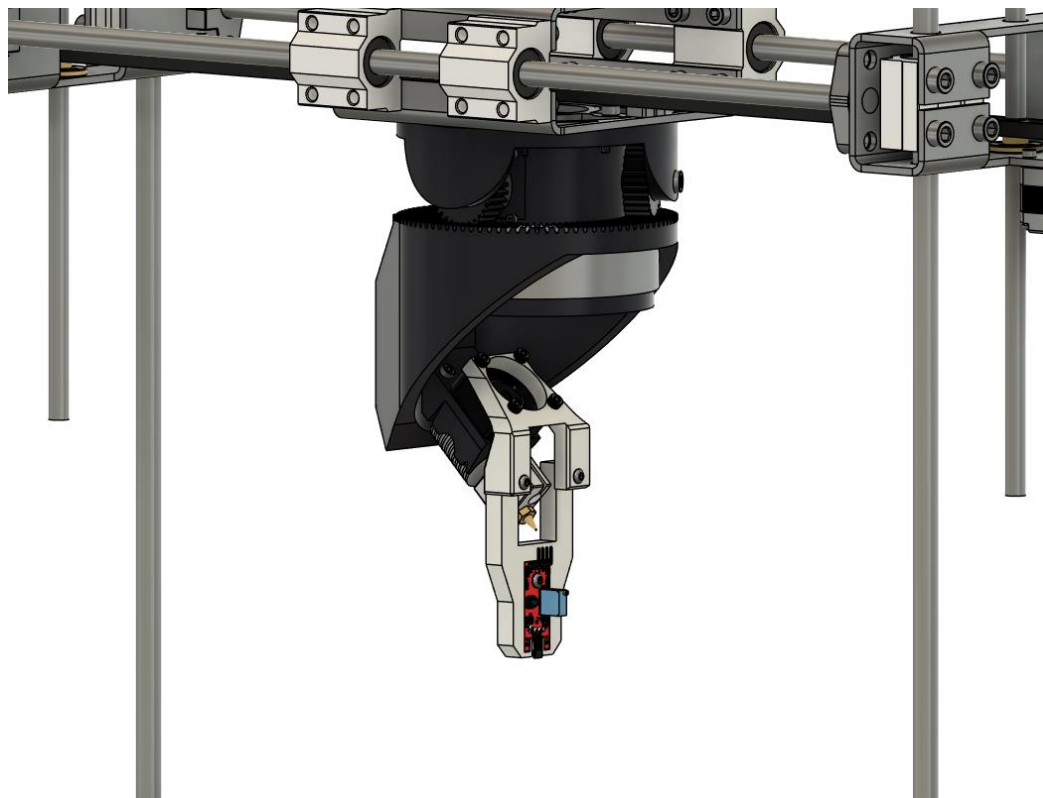


Рисунок 3.11 – Фіксація датчику Холла на каретці екструдера

Датчик, що використано в даній системі це КУ-024. Його фіксатор наведено на рисунку 3.12 - 3.14. Фіксатор складається із двох окремих частин, які з'єднуються між собою за допомогою двох гвинтових з'єднань гвинтами м3. Стационарна частина (рисунок 3.15 та 3.16) знаходиться на каретці постійно. А корпус датчику (рисунок 3.17 та 3.18) приєднується або для первинного налаштування офсетів осі R, або для періодичного калібрування принтеру. Датчик кріпиться в корпусі за допомогою гвинта та гайки м3.

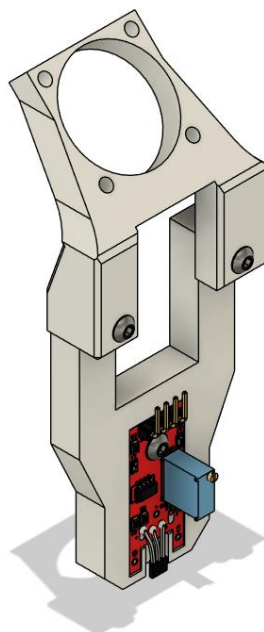


Рисунок 3.12 – Фіксатор датчику Холла

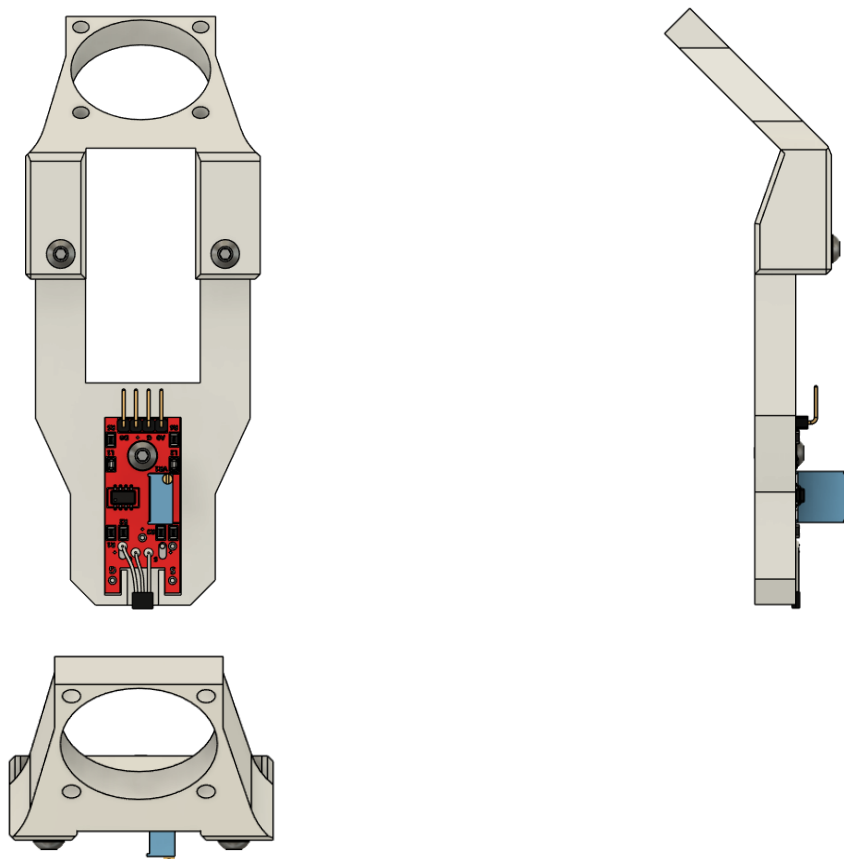


Рисунок 3.13 – Ескіз фіксатору датчику Холла

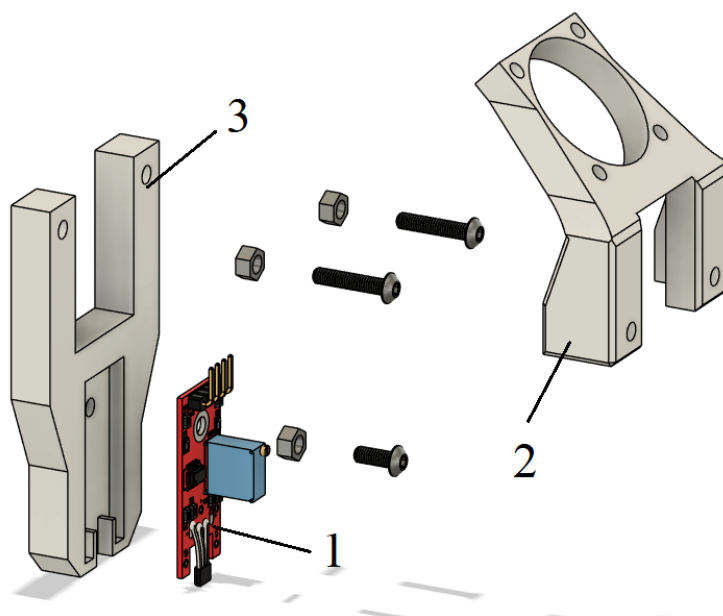


Рисунок 3.14 – «Вибух-схеми» фіксатору датчику Холла

На рисунку 3.14 зображено: 1 – датчик Холлу, 2 – стаціонарна частина, 3 – корпус датчику.

Стаціонарна частина (рисунок 3.15 та 3.16) фіксатору кріпиться чотирма гвинтами разом із вентилятором обдуву радіатору екструдеру да каретки екструдеру. Стаціонарна частина фіксатору виготовлюється із пластмаси PET за допомогою технології екструзійного FFF/FDM 3D друку.

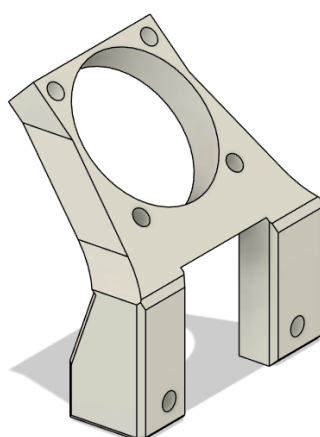


Рисунок 3.15 – Стаціонарна частина фіксатору

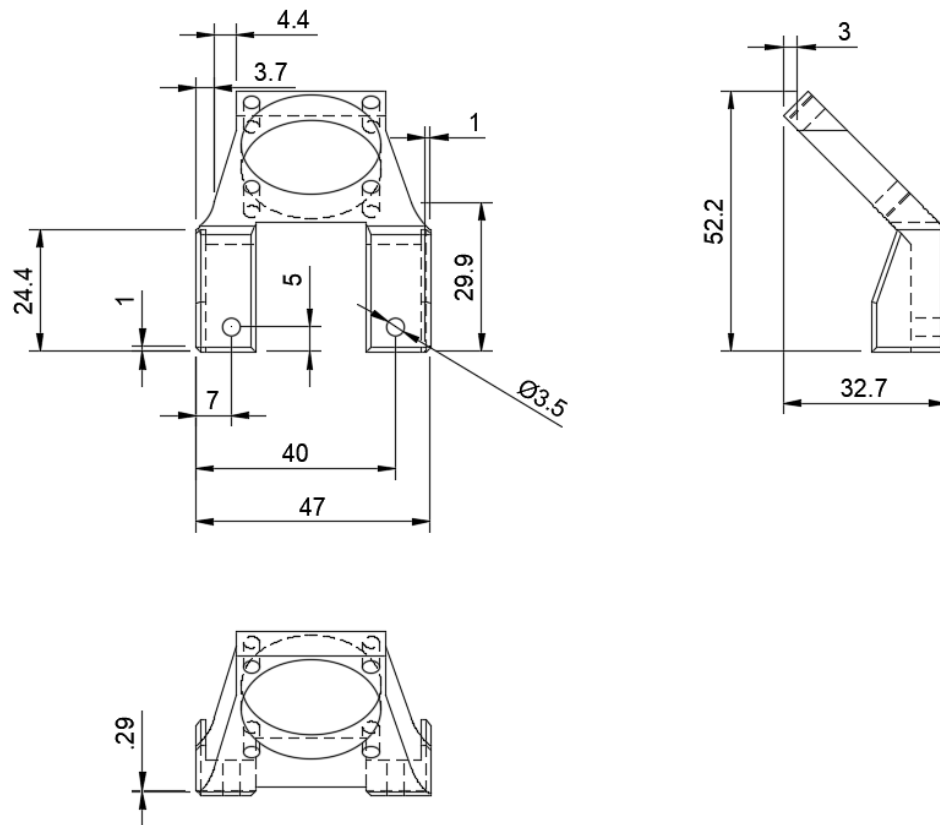


Рисунок 3.16 – Ескіз стаціонарної частини фіксатору

Корпус датчику зображено на рисунках 3.17 та 3.18.

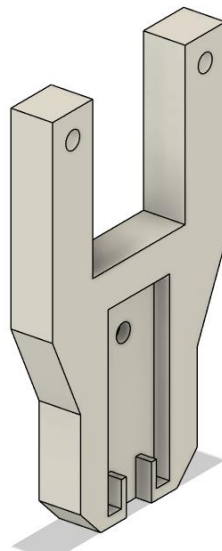


Рисунок 3.17 – Корпус датчику

Корпус датчику є зйомною частиною, яка не використовується під час друку. Корпус датчику виготовлюється із пластмаси PET за допомогою технології екструзійного FFF/FDM 3D друку.

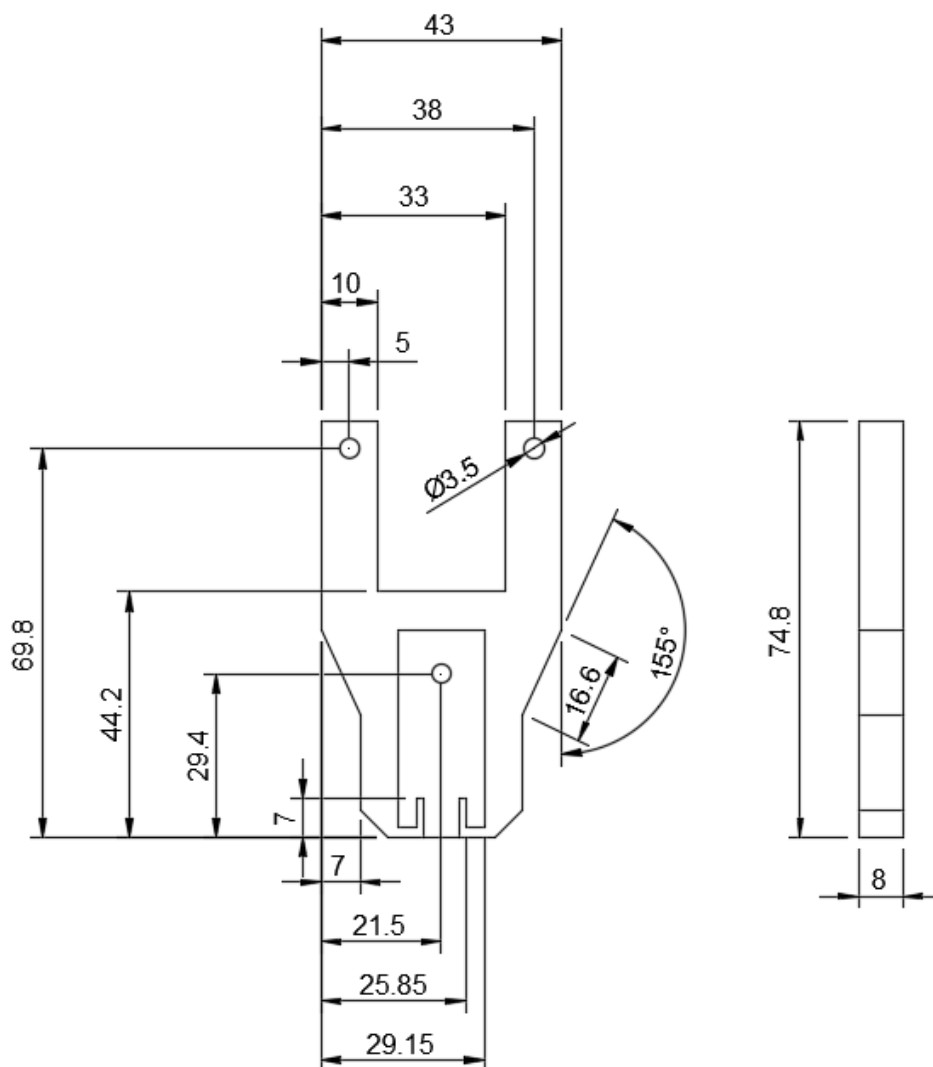


Рисунок 3.18 – Ескіз корпусу датчику

Для переміщення вздовж осі R та обертання платформи використовуються крокові двигуни Nema 17 та драйвери крокових двигунів TMC2209. Ці данні необхідно враховувати при розробці програмного забезпечення, що буде керувати процесом знаходження центру платформи.

3.2 Розробка програмного забезпечення

Для автоматизованого знаходження центру круглої платформи з 16 неодимовими магнітами по її периметру за допомогою датчика Холла та крокових двигунів (Nema 17 з драйверами TMC2209) створено Python-програму, яка керуватиме рухом платформи та лінійним рухом датчика Холла над платформою [16]. Програма повинна сканувати платформу та обчислювати центр на основі сигналів від датчика Холла, коли він виявляє магніти.

Структура програми:

- керування кроковими двигунами: Використовуємо бібліотеку для керування кроковими двигунами через драйвери TMC2209;
- читання сигналу датчика Холла: Читання сигналу датчика Холла для виявлення магнітів;
- знаходження центру платформи: Платформа обертається, а датчик рухається лінійно. Програма фіксує положення кожного магніту та обчислює центр платформи на основі їхнього розташування [17].

```
import time
import RPi.GPIO as GPIO
from RpiMotorLib import RpiMotorLib

# Налаштування GPIO для крокових двигунів та датчику Холла
DIR_PIN_R = 20 # Пін напрямлення для крокового двигуна (ось R)
STEP_PIN_R = 21 # Пін кроку для крокового двигуна осі R

DIR_PIN_PLATFORM = 19 # Пін напрямлення для платформи (ось  $\theta$ )
STEP_PIN_PLATFORM = 26 # Пін кроку для платформи

HALL_SENSOR_PIN = 17 # Пін для датчику Холла
```



```

STEPS_PER_REV = 200 # Кількість кроків для повного обороту (для
Nema 17)
DELAY = 0.001 # Затримка між кроками

# Ініціалізація GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(HALL_SENSOR_PIN, GPIO.IN)

# Налаштування крокового двигуна з використання драйверів TMC2209
motor_R = RpiMotorLib.A4988Nema(DIR_PIN_R, STEP_PIN_R, (-1, -1, -
1), "A4988")
motor_platform = RpiMotorLib.A4988Nema(DIR_PIN_PLATFORM,
STEP_PIN_PLATFORM, (-1, -1, -1), "A4988")

# Функція для обертання платформи
def rotate_platform(steps, direction, delay):
    GPIO.output(DIR_PIN_PLATFORM, direction)
    for _ in range(steps):
        GPIO.output(STEP_PIN_PLATFORM, GPIO.HIGH)
        time.sleep(delay)
        GPIO.output(STEP_PIN_PLATFORM, GPIO.LOW)
        time.sleep(delay)

# Функція для переміщення датчику по осі R (лінійний рух)
def move_sensor_r(steps, direction, delay):
    GPIO.output(DIR_PIN_R, direction)
    for _ in range(steps):
        GPIO.output(STEP_PIN_R, GPIO.HIGH)
        time.sleep(delay)

```

```
GPIO.output(STEP_PIN_R, GPIO.LOW)
time.sleep(delay)

# Функція сканування магнітів на платформі
def scan_magnets():
    magnet_positions = []

    # Платформа обертається та перевіряє сигнал з датчику Холла
    for step in range(STEPS_PER_REV):
        rotate_platform(1, True, DELAY)

        # Перевірка датчику Холла
        if GPIO.input(HALL_SENSOR_PIN) == GPIO.HIGH:
            magnet_positions.append(step)
            print(f"Магнит обнаружен на шаге: {step}")
            time.sleep(0.01)

    return magnet_positions

# Функція для знаходження центру платформи
def find_center():
    detected_magnets = []

    # Лінійне переміщення датчику Холла по осі R
    for x_step in range(STEPS_PER_REV):
        move_sensor_x(1, True, DELAY)

        # Сканування платформи на наявність магнітів
        magnets = scan_magnets()
```

```

if magnets:
    detected_magnets.append(magnets)
    print(f"Магніти знайдені на позиціях: {magnets}")

# Розрахунок центру платформи
if detected_magnets:
    total_steps = sum(sum(mag_pos) for mag_pos in detected_magnets)
    total_magnets = sum(len(mag_pos) for mag_pos in detected_magnets)
    center = total_steps / total_magnets
    print(f"Центр платформи знайдено на кроці: {center}")
    return center
else:
    print("Центр не знайдено.")
    return None

# Основа програма
try:
    center_position = find_center()
    if center_position is not None:
        print(f"Центр платформи знаходиться на кроці: {center_position}")
finally:
    GPIO.cleanup()

```

Програма, що розроблено, працює наступним чином:

Ініціалізація та налаштування:

- програма використовує бібліотеку RpiMotorLib для керування кроковими двигунами;
- на етапі ініціалізації задаються піни для управління напрямом та кроками для обох двигунів: DIR_PIN_X та STEP_PIN_R – піни для двигуна,

який переміщує датчик Холла лінійно; `DIR_PIN_PLATFORM` та `STEP_PIN_PLATFORM` – піни для двигуна, що обертає платформу;

- `HALL_SENSOR_PIN` – пін для підключення датчика Холла;
- змінна `STEPS_PER_REV` задає кількість кроків на повний оберт двигуна (200 кроків для Nema 17), а `DELAY` керує затримкою між кроками (час між перемиканням сигналу на кроковому двигуні).

Функції для керування кроковими двигунами:

- `rotate_platform(steps, direction, delay)` – ця функція обертає платформу на вказану кількість кроків.

Параметри:

- `steps`: кількість кроків, на які потрібно повернути платформу;
- `direction`: напрямок обертання (залежно від сигналу, що подається на пін `DIR_PIN_PLATFORM`);
- `delay`: затримка між кроками, що контролює швидкість обертання;
- `move_sensor_x(steps, direction, delay)` – функція для лінійного переміщення датчика Холла по осі R (над платформою);
- `steps`: кількість кроків, на які потрібно перемістити датчик;
- `direction`: напрямок руху (залежно від сигналу `DIR_PIN_R`);
- `delay`: затримка між кроками, що визначає швидкість лінійного руху.

Функція сканування магнітів: `scan_magnets()` – основна функція пошуку магнітів на платформі:

- платформа здійснює повний оборот (200 кроків, якщо задані параметри для Nema 17), датчик Холла перевіряє кожен крок на наявність магнітного поля;
- якщо датчик Холла фіксує магніт (сигнал `HALL_SENSOR_PIN` стає HIGH), програма записує поточний крок платформи як положення магніту;
- ці положення зберігаються у списку `magnet_positions`;
- функція повертає перелік кроків, на яких виявлено магніти.

Знаходження центру платформи: `find_center()` – це основна логіка програми, яка поєднує лінійний рух датчика та обертання платформи для

пошуку магнітів та обчислення їхнього середнього положення (центру платформи):

- датчик Холла переміщається по осі R за допомогою функції `move_sensor_x()` на кожному кроці;
- у кожній позиції датчика викликається функція `scan_magnets()`, яка обертає платформу та фіксує положення магнітів;
- коли виявлено магніти, їх позиції зберігаються у списку `detected_magnets`;
- після закінчення роботи функція обчислює середнє значення всіх положень магнітів, яке буде центром платформи;
- центр знаходиться як середнє арифметичне всіх кроків, на яких було виявлено магніти;
- якщо магніти були знайдені на кількох позиціях, програма підсумовує всі кроки та поділяє їх на кількість виявлених магнітів.

Основна логіка програми:

- в основній частині програми викликається функція `find_center()`, яка керує процесом знаходження центру;
- якщо центр знайдено (магніти виявлено), програма виводить повідомлення з координатами центру платформи (у кроках двигуна);
- наприкінці роботи програма очищає налаштування GPIO за допомогою функції `GPIO.cleanup()`, щоб коректно завершити виконання та звільнити ресурси.

Алгоритм роботи програми наступний: датчик Холла починає рухатися вздовж осі R за допомогою лінійного приводу. Двигун осі R робить один крок, датчик Холла рухається на невелику відстань, потім зупиняється для сканування платформи.

Починається обертання платформи та сканування магнітів. При кожному зупинці датчика платформа починає обертатися за допомогою двигуна осі θ .

На кожному кроці обертання перевіряється сигнал датчика Холла. Коли датчик фіксує магніт (сигнал HIGH на HALL_SENSOR_PIN), поточне положення зберігається. Процес триває доти, доки не буде зафіксовано кілька магнітів.

Висновок результату: коли програма знаходить кілька магнітів по периметру платформи, вона обчислює їхнє середнє положення та виводить координати центру платформи. Якщо магніти не виявлені, програма повідомляє, що центр не знайдено.

Важливими моментами є наступні. Програма розрахована використання двох крокових двигунів: один для обертання платформи, інший для лінійного переміщення датчика Холла. Датчик Холла використовується для виявлення магнітів, розміщених по периметру платформи.

Логіка програми побудована на обчисленні середнього становища всіх магнітів визначення центру платформи. Ця програма є базовим прикладом і для її роботи потрібно налаштувати правильні параметри для кроків двигуна, а також калібрувати рух датчика та платформи.

Програма може бути адаптована під інші платформи та компоненти, якщо використовуються різні драйвери або мікроконтролери.

3.3 Висновок до третього розділу

В даному розділі запропоновано програмно-технічний метод винайдення реального центру обертання платформи полярного 3D принтеру. Тобто виявлення нуля координат θ , який при виготовленні 3D принтеру може не співпадати з нулем координати осі R. У випадку коли ось R проходить через центр обертання платформи, корекція офсетів осі R. Коли ось R не проходить через центр обертання платформи, корекція відбувається за допомогою алгоритму та програми, що наведено у попередньому розділі.

Розроблений програмно-технічний метод використовується на етапі зборки чи поточної калібровки принтеру. І полягає у наступному: на платформі друку по периметру на однаковій відстані від центру розташовано шістнадцять неодимових магнітів, на каретку екструдера встановлюється датчик Холла. На початку алгоритму датчик Холла та каретка знаходяться за межами робочої платформи. Каретка покроково наближається до платформи, а платформа після кожного кроку обертається на $22,5^\circ$ – кут між сусідніми магнітами. Після виявлення магнітного поля, датчик Холла налаштовується на його максимум. Таким чином знаходяться координати всіх магнітів. Після чого відбувається розрахунок центру обертання платформи.

4 МОДЕЛЮВАННЯ ТОЧНОСТІ ПОПОВРОТУ КРОКОВОГО ВИГУНА 3D ПРИНТЕРУ

4.1 Чисельне моделювання

При проектуванні 3D-принтерів одним з ключових питань стоїть розробка системи управління приводами його робочих органів. Управління кроковим двигуном реалізується мікроконтролером через драйвер. Для цих цілей існують визначення стандарту - це сигналь STEP, DIR. Сигнал STEP відповідає за крок двигуна, DIR – за напрямок обертання.

На рисунку 4.1 зображено схему драйвера, що реалізована в середовищі Simulink. Основою всієї схеми є логічний блок, який формує керуючі впливи для електричних транзисторних ключів. Також до складу включений модуль, який генерує ШІМ-сигнал. Блок ШІМ складається з генератора напруги пилоподібної форми з частотою близько 50 кГц та компаратора напруги. Пилоподібний сигнал порівнюється з напругою на виході завдання, в результаті чого формується ШІМ-сигнал, який надходить на ключі нижнього рівня.

Для реалізації регулювання та обмеження струму в котушках крокового двигуна драйвер повинен містити змінні елементи, які дозволять впливати на струм, у складі своєї структури. У нашому випадку цю функцію виконують два блоки регулювання струмів фаз.

У деяких випадках управління потрібно отримати електричне дроблення кроку. У наведеній структурі формування мікрокроку відповідає модуль PWM (рисунок 4.1).

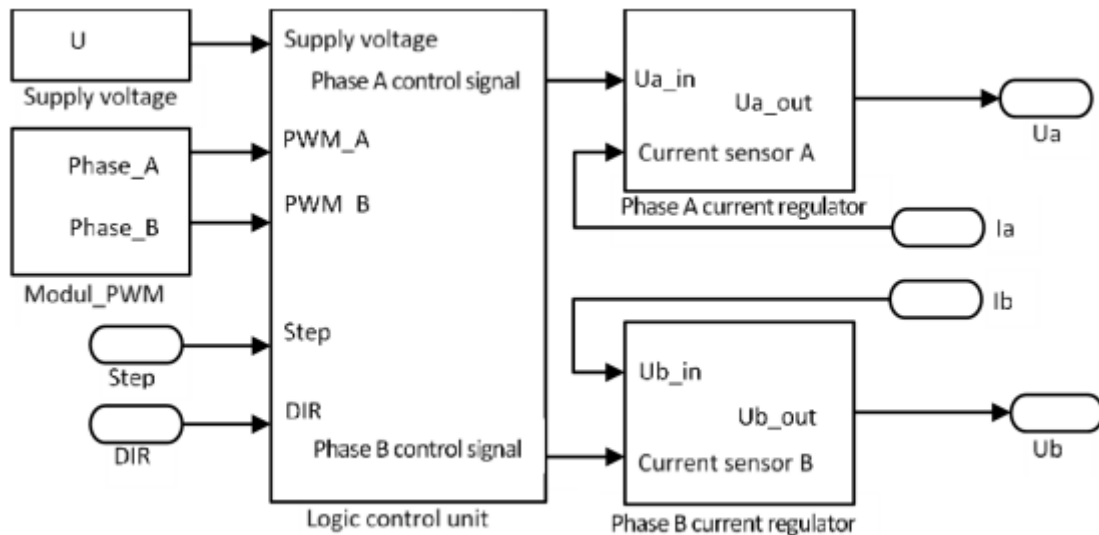


Рисунок 4.1 – Структурний блок драйверу крокового двигуна в Simulink

Для реалізації чисельного моделювання всієї системи в пакеті Matlab/Simulink була в результаті створена структура замкнутої системи управління з контролем і регулюванням положення ротора крокового двигуна (рисунок 4.2). Структура містить кроковий двигун із заданими паспортними значеннями; драйвер з напругою живлення 12 вольт і обмеженням струму до 2,5 А. Керування драйвером реалізується імпульсами каналами STEP і DIR. Як приклад у блоці завдання виставлено значення кута повороту в 10° .

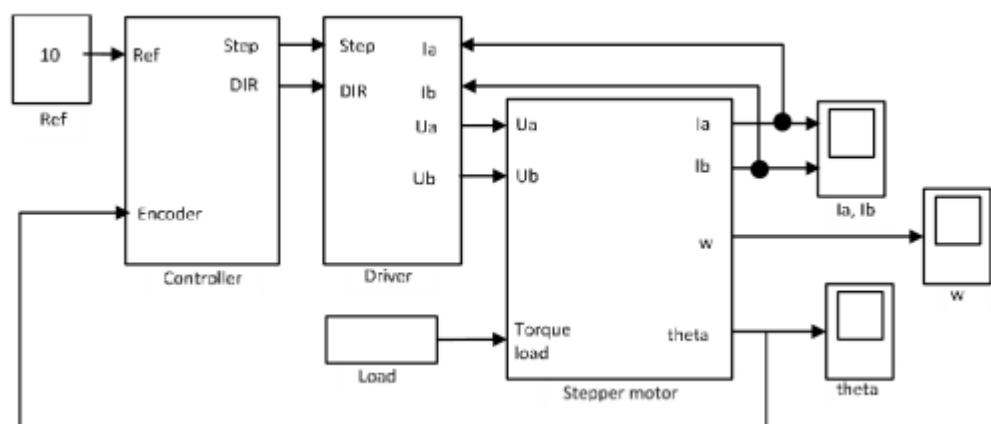


Рисунок 4.2 – Замкнена система керування кроковим двигуном

Результати моделювання представлені на рисунках 4.3-4.5.

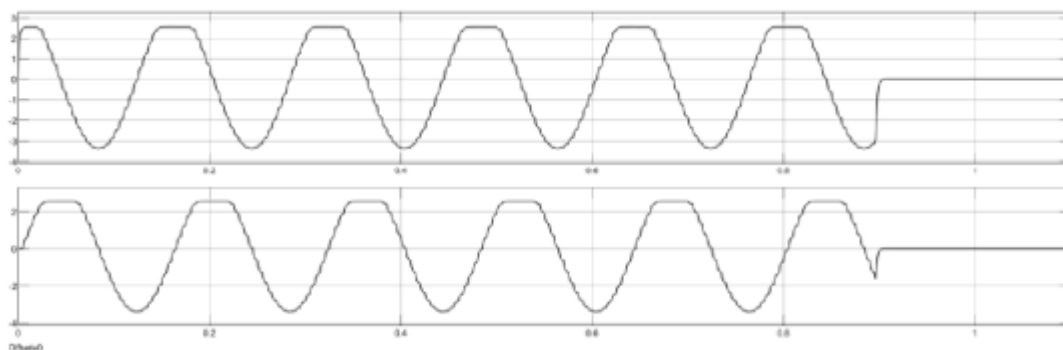


Рисунок 4.3 – Струми двигуна з обмеженням

На рисунку 4.3 представлені струми двигуна з обмеженням. Внутрішній контур керування струмами двигуна реалізований на позиційному регуляторі з обмеженням усередині драйвера.

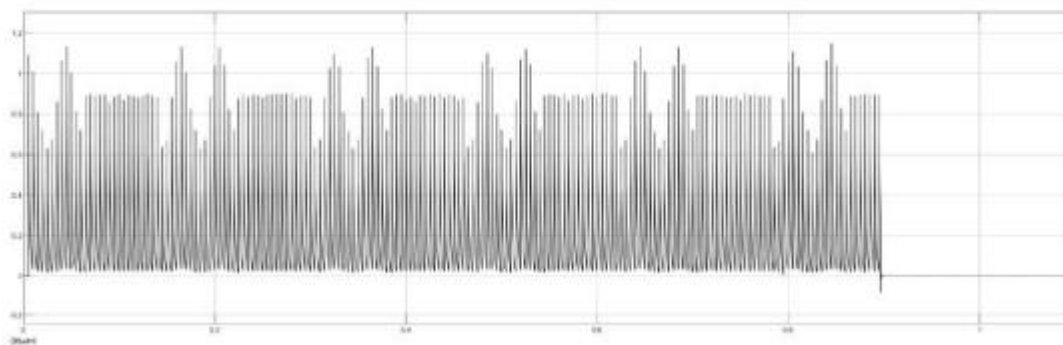


Рисунок 4.4 – Швидкість крокового двигуна

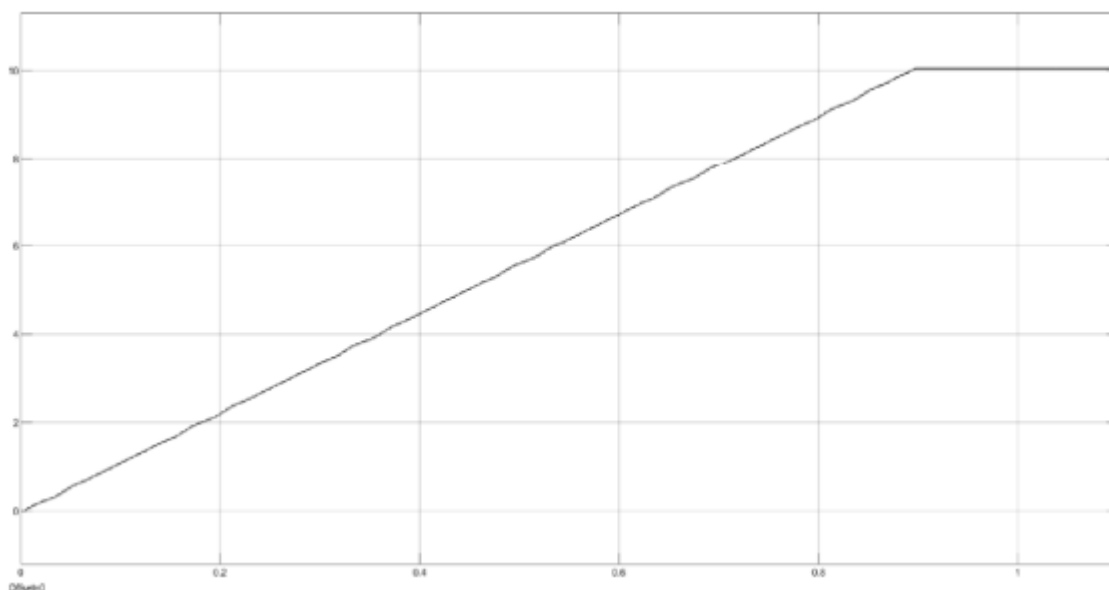


Рисунок 4.5 – Зміна кута повороту

Графіки перехідного процесу замкнутої системи позиціонування крокового двигуна представлений на малюнку 5. Похибка позиціонування моделі склала в абсолютному виразі менше $0,02^\circ$.

4.2 охорона праці

В даній роботі розроблено систему центрування 3D принтеру із використанням магнітів та датчику Холла. Тож доцільно розглянути вплив магнітного поля на організм людини. Магнітне поле є важливим фізичним фактором, з яким часто стикаються працівники в багатьох галузях, таких як енергетика, медицина (МРТ), промислове виробництво тощо. Вплив магнітного поля на організм людини залежить від його інтенсивності, частоти та тривалості дії. Знання про механізми впливу та методи захисту є необхідними для забезпечення здорових умов праці.

Магнітне поле має наступний вплив на організм людини. Магнітні поля, особливо змінні, можуть мати різний вплив на організм людини, викликаючи індукцію електричних струмів у тканинах. Ці струми можуть спричиняти:

- тепловий ефект: поля високої частоти викликають нагрівання тканин, що може призвести до підвищення температури тіла;
- електрофізіологічний вплив: низькочастотні магнітні поля (50-60 Гц) спричиняють індукцію струмів, які впливають на нервову і м'язову систему;
- вплив на центральну нервову систему: тривалий вплив магнітних полів може викликати порушення роботи ЦНС, головний біль, погіршення сну і концентрації.

Значні зміни можуть виникати при тривалому впливі полів промислової частоти (50 Гц), що є типовим для енергетичної та промислової сфери.

Існує кілька основних параметрів магнітних полів та методів їх нормування. Для оцінки впливу магнітного поля на працівника використовують такі основні параметри:

- магнітна індукція ВВВ, яка вимірюється в теслах (Тл);
- напруженість магнітного поля ННН, вимірюється в амперах на метр (А/м).

Залежно від рівня та тривалості дії, встановлені нормативи для магнітної індукції.

Нормативні значення магнітної індукції в робочій зоні:

- при перебуванні до 8 годин: $B \leq 0,01$ Тл;
- при перебуванні до 2 годин: $B \leq 0,1$ Тл;
- при короткочасній дії до 15 хвилин: $B \leq 0,5$ Тл.

Для розрахунків оцінки магнітного поля використовується кілька основних формул.

Магнітна індукція та напруженість поля. Загальна формула для магнітної індукції в однорідному середовищі:

$$B = \mu H, \quad (3.1)$$

де B – магнітна індукція, Тл;

μ – магнітна проникність середовища (для вакууму $\mu_0 = 4\pi \cdot 10^{-7}$ Гн/м);

H – напруженість магнітного поля, А/м.

Індуковане електричне поле. Для змінного магнітного поля індуковане електричне поле обчислюється за формулою:

$$E = - \frac{\partial B}{\partial t}, \quad (3.2)$$

де E – напруженість індукованого електричного поля, В/м;

$\frac{\partial B}{\partial t}$ – зміна магнітної індукції за часом.

Ця формула є основою для оцінки величини електричних струмів, що виникають у тілі під впливом змінного магнітного поля.

Також для оцінки впливу магнітного поля необхідно провести розрахунок потужності поглинання.

Потужність поглинання електромагнітної енергії в тілі людини може бути визначена за формулою:

$$P = \sigma E^2 V, \quad (3.3)$$

де P – потужність поглинання, Вт;

σ – електропровідність тканини, С/м;

E – напруженість індукованого електричного поля, В/м;

V – об'єм тіла, що знаходиться у магнітному полі, м³.

Для захисту від шкідливого впливу магнітного поля використовуються різні методи. Основні з них наступні.

Захисне екранування проводиться за допомогою матеріалів з високою магнітною проникністю, які здатні ефективно знижувати рівень магнітної індукції. Ефективність екрану визначається наступною формулою:

$$B_e = B_0 e^{-\frac{d}{\delta}}, \quad (3.4)$$

де B_e – індукція після проходження через екран;

B_0 – початкова індукція;

d – товщина екрана;

δ – глибина проникнення, яка залежить від магнітної проникності та частоти поля.

Другим методом захисту є обмеження часу впливу. Робота в умовах підвищеного магнітного поля регламентується обмеженням часу перебування у зоні дії. Наприклад, якщо поле має індукцію $B=0,1$ Тл, то дозволено перебувати в зоні його дії не більше 2 годин.

Також використовуються захисні відстані. Захист за допомогою дистанції базується на зниженні інтенсивності магнітного поля при збільшенні відстані від його джерела. Інтенсивність магнітного поля зменшується обернено пропорційно квадрату відстані від джерела:

$$B \propto \frac{1}{r^2}, \quad (3.5)$$

де B – магнітна індукція;

r – відстань від джерела поля.

Останнім методом є використання індивідуального захисту. Застосовуються також засоби індивідуального захисту, такі як спеціальний одяг з вставками зі струмопровідних матеріалів або матеріалів з високою магнітною проникністю, що допомагають захищати організм від впливу магнітних полів.

Робота в умовах підвищеного магнітного поля потребує ретельного дотримання заходів безпеки для мінімізації ризиків для здоров'я. Використання екранування, контроль за тривалістю впливу, організація робочих місць на безпечній відстані, а також засоби індивідуального захисту є ефективними методами забезпечення охорони праці. Дотримання цих рекомендацій дозволяє забезпечити безпеку та зберегти здоров'я працівників на виробництві.

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи є розробка автоматизованої системи центрування принтеру з циліндричною системою координат. В ході виконання роботи отримані наступні результати:

– проведено аналіз особливостей конструкції полярних 3D принтерів, що працюють в циліндричній системі координат. Виявлено їх основні недоліки та види дефектів друку, що можуть виникнути в процесі виробництва. Також виявлено помилки зборки 3D принтеру, що можуть призвести до цих дефектів;

– проведено аналіз існуючих систем генерації g-коду для полярних 3D принтерів. В результаті виявлено що все САМ системи «слайсери» генерують G-код у декартовій системі координат, а перетворення координат відбувається завдяки самому 3D принтеру. Разом із тим можливості G-коду дозволяють створювати команди для декартових координат;

– запропоновано алгоритм та розроблено програму для автоматичного перетворення G-коду з декартовими координатами у G-код з циліндричними координатами. Що дозволить простіше коригувати G-код при виявленні дефектів зборки принтеру;

– розроблено алгоритм та програмний модуль автоматичної заміни координат у G-кодi, який використовується при зміщенні нуля осі R відносно нуля координати обертання платформи θ , що може виникнути при зборці принтеру;

– розроблено програмно-технічний метод винайдення реального центру обертання платформи полярного 3D принтеру. Тобто виявлення нуля координат θ , який при виготовленні 3D принтеру може не співпадати з нулем координати осі R. У випадку коли ось R проходить через центр обертання платформи, корекція офсетів осі R. Коли ось R не проходить через центр обертання платформи, корекція відбувається за допомогою алгоритму та програми що наведено у попередньому пункті висновків.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2021. 55 с.
2. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. К.: ДП «УкрНДНЦ». 2016. 30 с. 3. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології»: довід. / І. Ш. Невлюдов, А. О. Андрусевич, О. В. Токарєва, Г. В. Пономарьова. Київ, 2018. 320 с.
3. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології»: довід. / І. Ш. Невлюдов, А. О. Андрусевич, О. В. Токарєва, Г. В. Пономарьова. Київ, 2018. 320 с.
4. Піхтерьов А.Д. Переваги та недоліки 3D-принтерів з полярною кінематикою / Automation and Development of Electronic Devices, Харків. 2023. №1. с. 61-64.
5. Alan Overby CNC Machining Handbook: Building, Programming, and Implementation 2nd Edition. McGraw Hill TAB 2019. p. 251.
6. Офіційний сайт Pressbooks // Сайт Manufacturing Processes. URL.: <https://openoregon.pressbooks.pub/manufacturingprocesses45/front-matterintroduction/> (дата звернення 22.11.2024).

7. Diy3dprinting // Сайт Pimaker 3D URL.: <https://diy3dprinting.blogspot.com/2013/08/pimaker-diy-3d-printer-with-circular.html> (дата звернення 23.11.2024).
8. Офіційний сайт Polar3d // Сайт Polar3d URL.: <https://polar3d.com> (дата звернення 23.11.2024).
9. Офіційний сайт UltimakerCura // Сайт UltimakerCura URL.: <https://ultimaker.com/software/ultimaker-cura/#downloads> (дата звернення 15.12.2024)
10. I. M. Gelfand, E. G. Glagoleva, A. A. Kirillov The Method of Coordinates. Dover Publications 2018. P.80.
11. Офіційний сайт International Organization for Standardization // Сайт G-code URL.: https://www.iso.org/?PROD_isoorg_en%5Bquery%5D=g-code (дата звернення 23.12.2024).
12. Офіційний сайт Python // Сайт Python. URL.: <https://www.python.org/> (дата звернення 20.06.2024).
13. Офіційний сайт Python // math — Mathematical functions. URL.: <https://docs.python.org/uk/3/library/math.html> (дата звернення 20.06.2024).
14. Fusion 360 // Сайт Autodesk. URL.: <https://www.autodesk.com/products/fusion-360/overview?term=1-YEAR&tab=subscription> (дата звернення 25.12.2024).
15. Hall Effect Sensors: A Comprehensive Guide // Сайт Monolithicpower. URL.: https://www.monolithicpower.com/en/learning/resources/hall-effect-sensors-a-comprehensive-guide?srsltid=AfmBOopQRcduXdaD-OrHRzCPbKOsUwflS76_XS23tAByD7K4L65bTYoo (дата звернення 20.11.2024)
16. Edward Ramsden Hall-Effect Sensors 2nd Edition. Newnes. 2016. p. 249.
17. Integrated computer solutions // Control Raspberry Pi GPIO Pins from Python URL.: <https://www.ics.com/blog/control-raspberry-pi-gpio-pins-python>

18. Піхтерьов А.Д. Корекція системи координат полярного 3D принтеру для підвищення якісних показників друку / Automation and Development of Electronic Devices, Харків. 2023. №2. с. 29-37.