ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. - 29 с.

2. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології освітньої програми «Системна інженерія» / Упоряд.: І.Ш. Невлюдов, О.М. Цимбал, О.В. Токарєва, А.І. Бронніков. Харків: ХНУРЕ, 2022. - 66 с.

3 Кафедра комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки. Про нас. Офіційний сайт кафедри КІТАМ ХНУРЕ . Режим доступу: https://tapr.nure.ua/golovna/pro-nas. Дата доступу: 24.04.2025.

4. Закон України "Про вищу освіту" від 01.07.2014 No 1556-VII. Режим доступу: https://zakon.rada.gov.ua/laws/show/1556-18#Text

5. Положення про організацію освітнього процесу у ХНУРЕ [електронний ресурс]: Наказ ХНУРЕ від 27.11.2020 р. No 400. Режим доступу: https://nure.ua/wp- content/uploads/Main_Docs_NURE/polozhennja-pro-organizaciju - osvitnogo-procesu- v-hnure.pdf

6 Положення про академічну доброчесність [Електронний ресурс]: Наказ ХНУРЕ від 02 лютого 2021 р. No 50. – Режим доступу: https://nure.ua/wpcontent/uploads/Main_Docs_NURE/polozhennja-pro-akademi chnu-dobrochesnist.pdf.

7 Іванов Ю.С. Система автоматизованого контролю та підтримки оптимального рівня освітленості у приміщеннях / Ю.С. Іванов, К.В. Силчук // Автоматизація та приладобудування («Automation and Development of Electronic Devices» ADED-2023): збірник студентських наукових статей / Харківський національний університет радіоелектроніки; Харків: ХНУРЕ, 2023. – Вип. 1. С.287-290.

8 Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» : Навчальний посібник / І. Ш. Невлюдов, В.А. Андрусевич, О. В. Токарєва, С. П. Новоселов, О. В. Сичова. – Харків : Видавництво Іванченка І. С., 2022. 151 с.

9 Комерційне інтегроване середовище розробки IntelliJ IDEA [Електронний ресурс] / Режим доступу: https://uk.wikipedia.org/wiki/IntelliJ_IDEA

10 Невлюдов І. Ш. BEAM робототехніка / І. Ш. Невлюдов, В. В. Євсєєв, С.С. Максимова. – 2024. – 276 с.

11 Кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки (КІТАР). nure. URL: https://nure.ua/department/kafedra-komp-yuterno- integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitam (дата звернення: 24.04.2025).

12 Верстатний профіль 30х30 T-track без покриття (3842990720). alu-met.com.ua. URL: https://alu-met.com.ua/stanochnij-profil-30h30-t-track-bez-pokritija--3842990720-/?gclid=Cj0KCQjw5azABhD1ARIsAA0WFUHXl93JzwMcY9szTlyZahvUp31GyIQZ65cUY_U04Fw22qQLYEaIBOIaArIlEALw_wcB (дата звернення: 24.04.2025).

13 Re-D-Bot v1 - бюджетний CoreXY 3D-принтер. red-bot.dipteam.com URL: https://red-bot.dipteam.com/pages/ru/printers/v1/assembly/parts_frames/ (date of access: 24.04.2025).

14 An Alternative Parallel Mechanism for Horizontal Positioning of a Nozzle in an FDM 3D Printer URL: https://www.researchgate.net/figure/a-Schematic-of-a-printer-with-Cartesian-XZ-head-architecture-b-The-same-for-XY_fig2_361758640 (date of access: 24.04.2025).

15 An Alternative Parallel Mechanism for Horizontal Positioning of a Nozzle in an FDM 3D Printer. researchgate URL:https://www.researchgate.net/figure/Schematic-of-an-H-bot-routing-as-seen-from-the-top-of-the-printer-The-pulleys_fig3_361758640 (date of

access: 24.04.2025).

16 3D Printers with Different Kinematics: Comparison, Advantages and Disadvantages. top3dshop. URL: https://top3dshop.com/blog/3d-printer-kinematics-explained (дата звернення: 24.04.2025).

17 MKS Gen V1.4 плата керування для 3D принтера. mini-tech.com.ua. URL: https://www.mini-tech.com.ua/ua/mks-gen (дата звернення: 24.04.2025).

18 MKS Robin Nano V1.2. grabcad.com. URL: https://grabcad.com/library/mks-robin-nano-v1-2-1 (дата звернення: 24.04.2025).

19 Creality Ender6 Board V4.3.1. grabcad.com. URL: https://grabcad.com/library/creality-ender6-board-v4-3-1-1 (дата звернення: 09.02.2024).

20 Basic features of Orange Pi Zero 3 www.orangepi.org. URL: http://www.orangepi.org/orangepiwiki/index.php/Orange_Pi_Zero_3#Basic_features_of_Orange_Pi_Zero_3 (дата звернення: 24.04.2025).

21 Raspberry Pi 4 Model B grabcad.com. URL: https://grabcad.com/library/raspberry-pi-4-model-b-1 (дата звернення: 24.04.2025).

22 NVIDIA Jetson Nano. grabcad.com.URL: https://grabcad.com/library/nvidia-jetson-nano-2 (date of access: 24.04.2025).

23 Крокoвий двигун NEMA17 17HS4401 4кг. uamper.com. URL: https://uamper.com/index.php?route=product/product&path=380&product_id=267&gad_source=1&gbraid=0AAAADk5dvqjvj47Fbs21XCZNbl47GYri&gclid=Cj0KCQjw5azABhD1ARIsAA0WFUGs7dK8EkzQIZwlM_3ASPZHqzC7o_8cFGy4_CL_u1LLx1aGk5RBB4EaAgriEALw_wcB (дата звернення: 24.04.2025).

24 Крокoвий двигун Nema23 23HS8430. uamper.com. URL: https://uamper.com/index.php?route=product/product&path=380&product_id=1163&gad_source=1&gbraid=0AAAADk5dvqjvj47Fbs21XCZNbl47GYri&gclid=Cj0KCQjw5azABhD1ARIsAA0WFUF1lDHVq0EdA6siz-FACI3RxUfN9at6jFCKBkf312NPGIjF7yB0UDwaAj3bEALw_wcB (дата звернення: 24.04.2025).

25 Крокoвий двигун Nema 14 OK35STH28-0754A. voron.ua. URL:

https://voron.ua/catalog/040722--shagovyy-dvigatel-nema-14-ok35sth28-0754a?lang=ua (дата звернення: 04.02.2025).

26 Акселерометр ADXL345 USB Mellow Mellow ADXL345 USB Mellow. screwmaker.com.ua. URL: https://screwmaker.com.ua/akselerometr-adxl345-usb-mellow (дата звернення: 24.04.2025).

27 3-осний акселерометр ADXL345 DFRobot. miniboard.com.ua. URL: https://miniboard.com.ua/sensors/664-3-kh-osevoj-akselerometr-adxl345-dfrobot.html (дата звернення: 24.04.2024).

28 Measuring Resonances Klipper https klipper3d.org URL: https://www.klipper3d.org/Measuring_Resonances.html. (дата звернення: 24.04.2024)

29 CoreXY reprap.org URL: https://reprap.org/wiki/CoreXY(дата звернення: 24.04.2024)

30 MKS GEN reprap.org URL: https://reprap.org/wiki/MKS_GEN(дата звернення: 24.04.2024)

31 Pressure advance klipper3d.org URL: https://www.klipper3d.org/Pressure_Advance.html (дата звернення: 24.04.2024)

32 Linear Advance marlinfw.org URL: https://marlinfw.org/docs/features/lin_advance.html (дата звернення: 24.04.2024)

33 Hutchinson S., Vidyasagar M., Spong M. W. Robot Modeling and Control. Wiley & Sons, Incorporated, John, 2020. 608 с. дата звернення: 24.04.2024)

34 Investigating pressure advance algorithms for filament-based melt extrusion additive manufacturing: theory, practice and simulations / S. Arnts та ін. 2019. 839 с. дата звернення: 24.04.2024)

35 Комплекс навчально-методичного забезпечення навчальної дисципліни «Безпека праці в індустрії ІТ-технологій» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [http://catalogue.nure.ua/knmz] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 122 с

ДОДАТОК А

Код програми:

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <sstream>
#include <regex>
#include <cmath>
#include <iomanip>

// For UTF-8 support on Windows
#ifdef _WIN32
#include <windows.h>
#endif

// PI constant
const double M_PI = 3.14159265358979323846;

// Structure to store coordinates
struct Point {
    double x, y, z;
    Point(double x_ = 0.0, double y_ = 0.0, double z_ = 0.0) : x(x_), y(y_), z(z_)
{}
};


Point correctPointSingleAngle(const Point& p, double theta) {
    // CF – ідеальна позиція по y
    double CF = p.y;
```

```
    // Обчислюємо горизонтальний зсув GF з трикутника FGD:
    double GF = CF * std::sin(theta * M_PI / 180.0);
    // Обчислюємо GD – смежний катет трикутника FGD:
    double GD = CF * std::cos(theta * M_PI / 180.0);
    // Вертикальний зсув GC = CF - GD:
    double GC = CF - GD;

    // Нові координати обчислюються як:
    // x коригується на GF, а y – на GC.
    return Point(p.x - GF, p.y - GC, p.z);
}


Point correctPointDualAngle(const Point& p, double alpha, double beta) {
    // alpha  – кут зміщення осі Y (відносно ідеальної вертикалі)
    // beta   – кут зміщення осі X (відносно ідеальної горизонталі)
    //
    // В ідеальній моделі точка має координати (x, y)
    // Реальна координата визначається як:
    //   x' = x*cos(beta) + y*sin(alpha)
    //   y' = x*sin(beta) + y*cos(alpha)
    // Тобто, вектор зміщення:
    //   Δx = x*(cos(beta) - 1) + y*sin(alpha)
    //   Δy = x*sin(beta) + y*(cos(alpha) - 1)
    //
    // Перекладаємо кутові значення з градусів у радіани.
    double radAlpha = alpha * M_PI / 180.0;
    double radBeta = beta * M_PI / 180.0;

    double deltaX = p.x * (std::cos(radBeta) - 1.0) + p.y * std::sin(radAlpha);
```

```
    double deltaY = p.x * std::sin(radBeta) + p.y * (std::cos(radAlpha) - 1.0);

    return Point(p.x - deltaX, p.y - deltaY, p.z);
}

// Форматуємо рядок G-коду без параметру E
std::string formatGCodeLine(const std::string& command, const Point& p) {
    std::stringstream ss;
    ss << command << " X" << std::fixed << std::setprecision(3) << p.x
       << " Y" << p.y << " Z" << p.z;
    return ss.str();
}

// Calculate distance in XY plane
double distanceXY(const Point& p1, const Point& p2) {
    return std::sqrt(std::pow(p2.x - p1.x, 2) + std::pow(p2.y - p1.y, 2));
}

// Generate visualization code for GNU Octave
void generateOctaveVisualization(const std::vector<Point>& pointsBefore,
const std::vector<Point>& pointsAfter) {
    std::ofstream beforeFile("before.txt");
    std::ofstream afterFile("after.txt");

    if (!beforeFile.is_open() || !afterFile.is_open()) {
        std::cerr << "Error: Unable to open visualization files" << std::endl;
        return;
    }

    beforeFile << "x = [";
```

```
for (size_t i = 0; i < pointsBefore.size(); ++i) {
   beforeFile << pointsBefore[i].x << (i < pointsBefore.size() - 1 ? "; " : "");
}
beforeFile << "];\n";
beforeFile << "y = [";
for (size_t i = 0; i < pointsBefore.size(); ++i) {
   beforeFile << pointsBefore[i].y << (i < pointsBefore.size() - 1 ? "; " : "");
}
beforeFile << "];\n";
beforeFile << "z = [";
for (size_t i = 0; i < pointsBefore.size(); ++i) {
   beforeFile << pointsBefore[i].z << (i < pointsBefore.size() - 1 ? "; " : "");
}
beforeFile << "];\n";
beforeFile << "plot3(x, y, z, 'b.-', 'LineWidth', 1, 'MarkerSize', 10);\n";
beforeFile << "title('Points before correction');\n";
beforeFile << "xlabel('X axis (mm)');\n";
beforeFile << "ylabel('Y axis (mm)');\n";
beforeFile << "zlabel('Z axis (mm)');\n";
beforeFile << "grid on;\n";
beforeFile << "view(3);\n";
beforeFile << "axis equal;\n";

afterFile << "x = [";
for (size_t i = 0; i < pointsAfter.size(); ++i) {
   afterFile << pointsAfter[i].x << (i < pointsAfter.size() - 1 ? "; " : "");
}
afterFile << "];\n";
afterFile << "y = [";
for (size_t i = 0; i < pointsAfter.size(); ++i) {
```

```
            afterFile << pointsAfter[i].y << (i < pointsAfter.size() - 1 ? "; " : "");
        }
        afterFile << "];\n";
        afterFile << "z = [";
        for (size_t i = 0; i < pointsAfter.size(); ++i) {
            afterFile << pointsAfter[i].z << (i < pointsAfter.size() - 1 ? "; " : "");
        }
        afterFile << "];\n";
        afterFile << "plot3(x, y, z, 'r.-', 'LineWidth', 1, 'MarkerSize', 10);\n";
        afterFile << "title('Points after correction');\n";
        afterFile << "xlabel('X axis (mm)');\n";
        afterFile << "ylabel('Y axis (mm)');\n";
        afterFile << "zlabel('Z axis (mm)');\n";
        afterFile << "grid on;\n";
        afterFile << "view(3);\n";
        afterFile << "axis equal;\n";

        beforeFile.close();
        afterFile.close();
        std::cout << "Visualization code saved to before.txt and after.txt" << std::endl;
    }

    // Process G-code with angle correction
    // Під час обробки парсимо лише координати X, Y, Z
    void processGCode(const std::string& inputFilename, const std::string& outputFilename,
        double theta, double phi, bool twoAngles) {

        std::ifstream inFile(inputFilename);
        std::ofstream outFile(outputFilename);
```

```
        if (!inFile.is_open() || !outFile.is_open()) {
            std::cerr << "Error: Unable to open files" << std::endl;
            return;
        }

        std::vector<Point> beforePoints, afterPoints;
        double currentX = 0.0, currentY = 0.0, currentZ = 0.0;
        Point previousPoint(0.0, 0.0, 0.0);
        double previousZ = 0.0;

        // Парсимо лише X, Y, Z;
        std::regex coordRegex("(X-?\\d*\\.?\\d*)|(Y-?\\d*\\.?\\d*)|(Z-?\\d*\\.?\\d*)");
        std::regex commandRegex("^(G0|G1)\\s");
        std::string line;

        while (std::getline(inFile, line)) {
            if (line.empty() || line[0] == ';' || line[0] == '(' ||
                (line.find("G0 ") == std::string::npos && line.find("G1 ") ==
std::string::npos)) {
                outFile << line << "\n";
                continue;
            }

            std::smatch commandMatch;
            std::string command = "G0";
            if (std::regex_search(line, commandMatch, commandRegex)) {
                command = commandMatch[1].str();
            }
```

```
double newX = currentX, newY = currentY, newZ = currentZ;
std::smatch match;
std::string::const_iterator searchStart(line.cbegin());
while (std::regex_search(searchStart, line.cend(), match, coordRegex)) {
   for (size_t i = 1; i < match.size(); ++i) {
      if (match[i].matched) {
         std::string coord = match[i].str();
         if (coord[0] == 'X')
            newX = std::stod(coord.substr(1));
         else if (coord[0] == 'Y')
            newY = std::stod(coord.substr(1));
         else if (coord[0] == 'Z')
            newZ = std::stod(coord.substr(1));
      }
   }
   searchStart = match.suffix().first;
}

Point idealPoint(newX, newY, newZ);
beforePoints.push_back(idealPoint);
currentX = newX;
currentY = newY;
currentZ = newZ;

Point correctedPoint;
if (twoAngles)
   correctedPoint = correctPointDualAngle(idealPoint, theta, phi);
else
   correctedPoint = correctPointSingleAngle(idealPoint, theta);
```

```
            afterPoints.push_back(correctedPoint);

            outFile << formatGCodeLine(command, correctedPoint) << "\n";

            previousPoint = correctedPoint;
            previousZ = currentZ;
        }

        inFile.close();
        outFile.close();
        std::cout << "New G-code saved to " << outputFilename << std::endl;

        generateOctaveVisualization(beforePoints, afterPoints);
    }

    int main() {
    #ifdef _WIN32
        SetConsoleOutputCP(CP_UTF8);
        SetConsoleCP(CP_UTF8);
    #endif
        std::cout.imbue(std::locale(""));

        int choice;
        std::cout << "Choose correction type: 1 - one angle, 2 - two angles: ";
        std::cin >> choice;

        double theta, phi = 0.0;
        std::cout << "Enter angle theta in degrees: ";
        std::cin >> theta;
        if (choice == 2) {
```

```
            std::cout << "Enter angle phi in degrees: ";
            std::cin >> phi;
        }

        processGCode("input.txt", "NewGcode.txt", theta, phi, choice == 2);

        return 0;
}
```

Код розробленого макросу :

```
        # ------------------------------------------------------
        # Файл: filament_sensor.cfg
        # Призначення: Налаштування оптичного енкодера FC-03 LM393
        # та макросу перевірки подачі філаменту з діалоговим вікном
        # ------------------------------------------------------

        ##########################################################################
  #
        # 1) Налаштування оптичного енкодера як датчика швидкості
        ##########################################################################
  #
        [filament_motion_sensor filament]
        switch_pin: ^PA6
        pulses_per_rotation: 20
        rotation_distance: 88.91
filter_count: 2
        ##########################################################################
  #
        # 2) Макрос порівняння швидкостей із діалоговим вікном у UI
        ##########################################################################
  #
```

```
[gcode_macro CHECK_FILAMENT]
# Принцип роботи:
#   • З G-code (F-параметр) передаємо бажану лінійну швидкість подачі філаменту в мм/с.
#   • Сенсор FC-03 (LM393) зчитує фактичну швидкість руху філаменту,
#    обчислену як кількість імпульсів за секунду, помножену на відстань за оберт.
#   • Всередині макроса порівнюються бажана та фактична швидкості.
#   • Результат (OK або  Недоподача) виводиться у вебморду Mainsail через Macro Prompts.
description:
   """
   Перевірка подачі філаменту:
    • DESIRED – бажана швидкість (мм/с) з G-code
    • THRESHOLD – мінімальна допускова частка (0.9 = 90%)
    • Принцип: якщо фактична < DESIRED × THRESHOLD → попередження і пауза
   """
default_parameter_THRESHOLD: 0.9

gcode:
   {% set desired = params.DESIRED|float %}
   {% set threshold = params.THRESHOLD|float %}
   {% set actual  = printer.filament_motion_sensor.filament.speed %}

   # action:prompt_begin Перевірка подачі філаменту
   # action:prompt_text Бажано: {{'{:.1f}'.format(desired)}} мм/с
   # action:prompt_text Фактично: {{'{:.1f}'.format(actual)}} мм/с

   {% if actual < desired * threshold %}
```

```
            # action:prompt_text Статус:  Недоподача!
            PAUSE
          {% else %}
            # action:prompt_text Статус:  Подача в нормі
          {% endif %}

# action:prompt_button Закрити||primary
```

ДОДАТОК Б

Демонстраційний матеріал

Рисунок Б.1–Титульний аркуш демонстраційного матеріалу

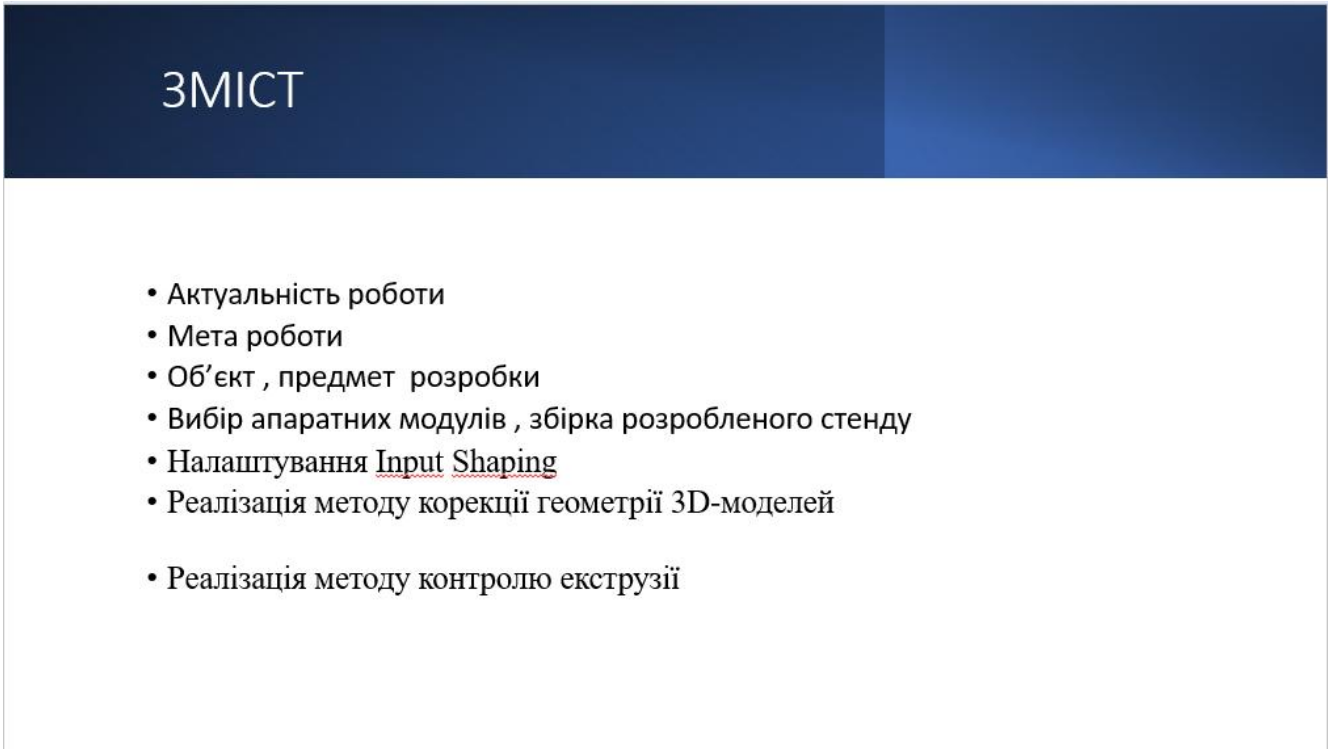

Рисунок Б.2–Зміст демонстраційного матеріалу

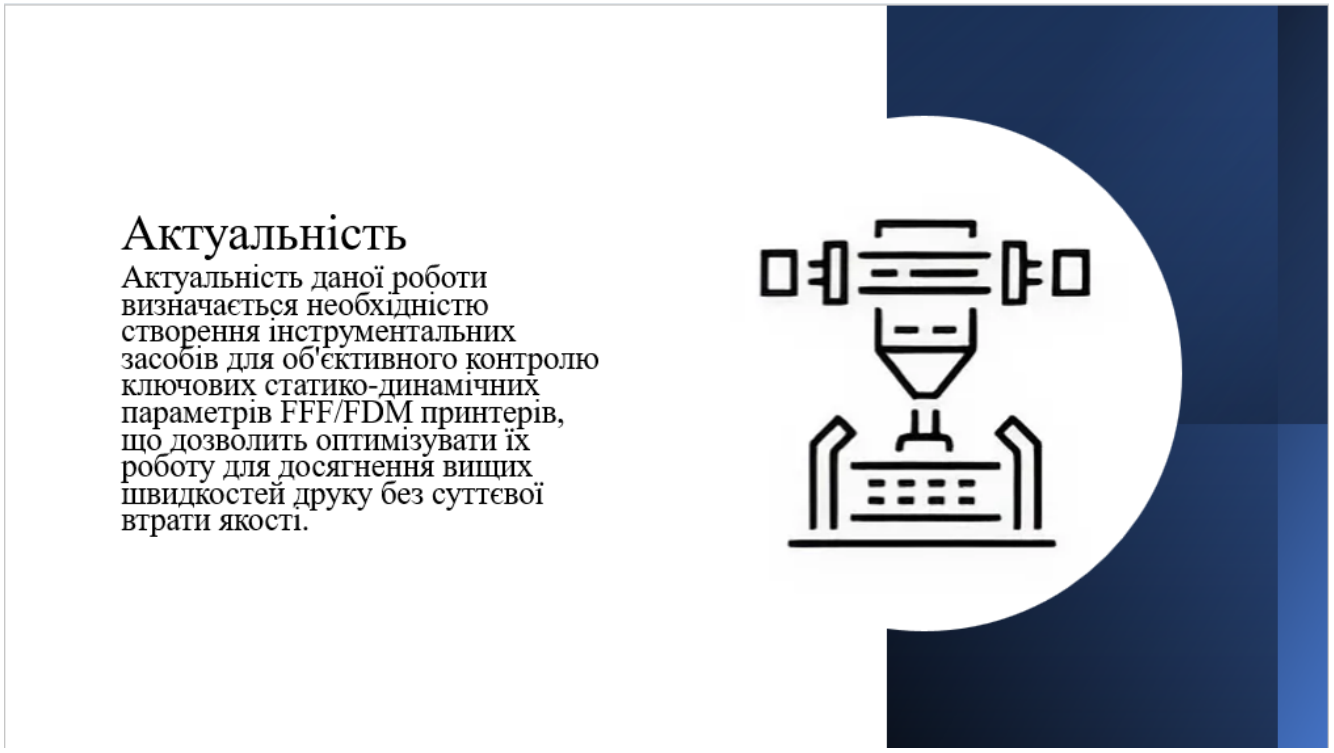

Рисунок Б.3–Слайд 3 демонстраційного матеріалу

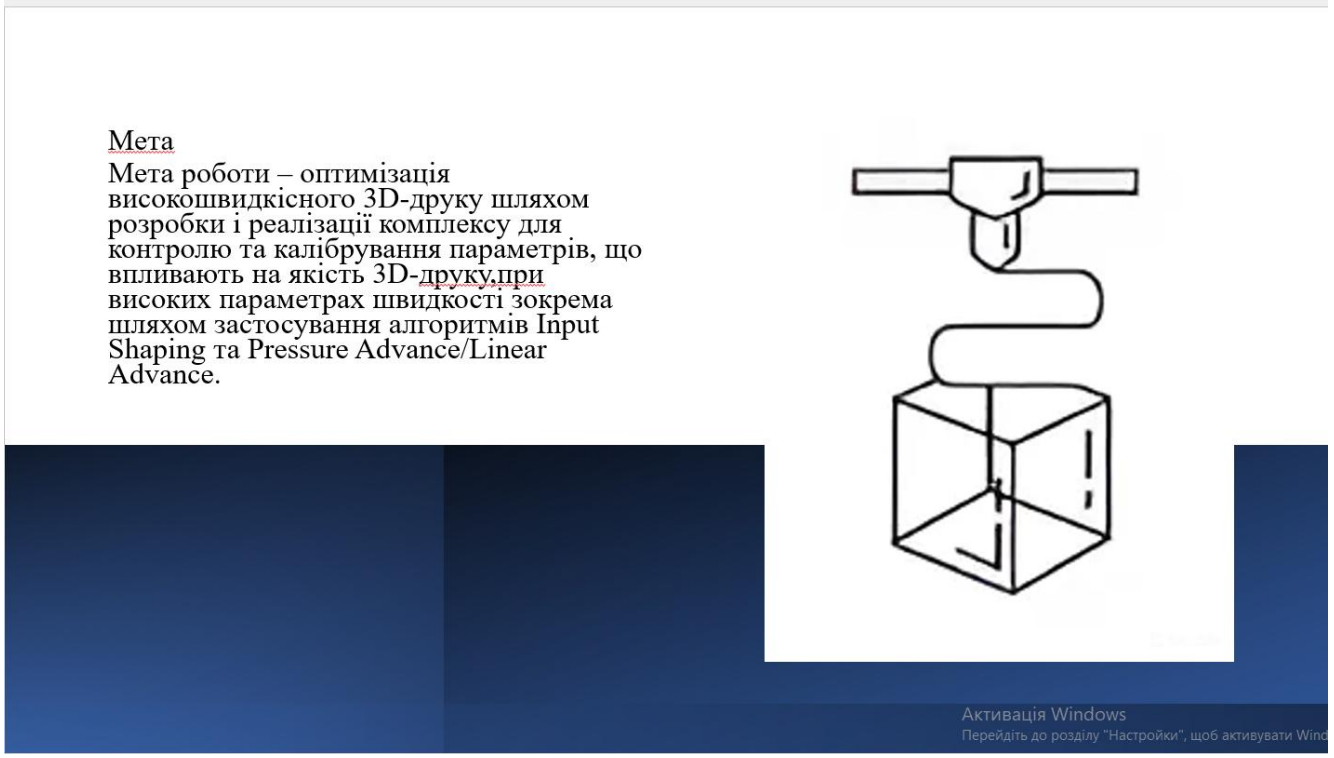

Рисунок Б.4– Слайд 4 демонстраційного матеріалу

Об’єкт, предмет розробки

• Об’єкт розробки – технологічний процес екструзійного FFF/FDM 3D-друку.
• Предмет дослідження – методи та алгоритми компенсації програмних та конструкційних відхилень 3D принтеру, що впливають на якість друку.

Рисунок Б.5– Слайд 5 демонстраційного матеріалу

Вибір апаратних модулів

• Обрані апаратні модулі та кінематика для реалізації стенду:
• - Кінематика: CoreXY – забезпечує високошвидкісний, точний рух друкуючої головки з мінімальною інерцією.
• - Плата управління: MKS Gen 1.4 – стабільна платформа, сумісна з сучасними прошивками (Klipper, Marlin).
• - Крокові двигуни: NEMA 17 – забезпечують оптимальний баланс між крутним моментом, точністю та компактністю.
• - Одноплатний ПК: Orange Pi Zero 3 – ефективний сервер для керування процесами друку.
• - Датчик: Акселерометр Mellow Fly ADXL345 – для вимірювання вібрацій та реалізації алгоритмів Input Shaping.

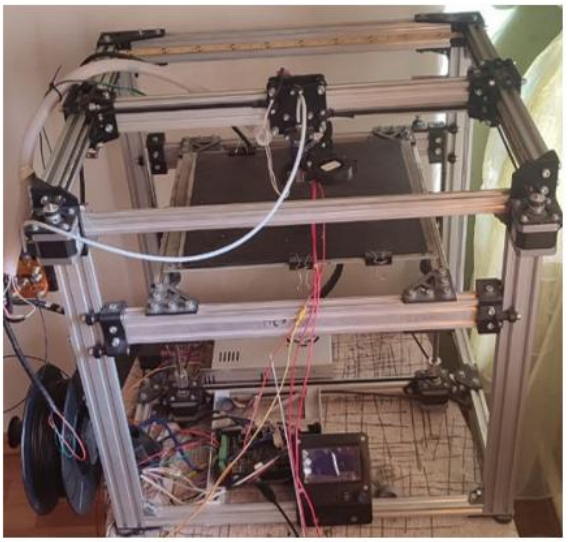

Рисунок Б.6– Слайд 6 демонстраційного матеріалу

Налаштування Input Shaping

- Для компенсації вібрацій друкуючої головки застосовано алгоритм Input Shaping, який формує послідовність імпульсів для гасіння резонансних коливань.
- За результатами розрахунків основна резонансна частота осі X становить приблизно 35 Гц.
- ZV shaper (Zero Vibration):
- Два імпульси з амплітудами 0.5;
- Перший імпульс подається в момент часу t = 0, другий – із затримкою ~14,32мс.
- ZVD shaper (Zero Vibration Derivative):
- Три імпульси з амплітудами 0.25, 0.5 і 0.25;
- Затримки: t = 0, t ≈ 14,32мс мста t ≈ 28,64 мс.

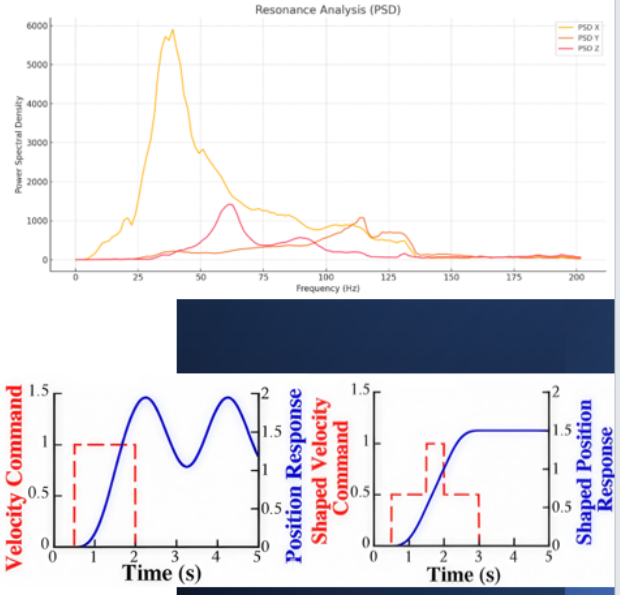

Рисунок Б.7– Слайд 7 демонстраційного матеріалу

Реалізація методу корекції геометрії 3D-моделей

- -Однокутова корекція:
- - Використовується для компенсації глобальних помилок через відхилення осей.
- - Горизонтальний зсув обчислюється як GF = CF · sin(θ), де CF – ідеальна координата по осі Y, а θ – виміряний кут відхилення.
- - Вертикальний зсув визначається як GC = CF – CF · cos(θ).
- - Нові координати розраховуються як: x' = x – GF, y' = y – GC.
- Двокутова корекція:
- - Забезпечує адаптивну трансформацію координат з урахуванням двох параметрів (кутів) відхилення осей.
- - Формули перетворення: x' = x·cos(β) + y·sin(α) та y' = x·sin(β) + y·cos(α),
- де α і β – відповідні кути відхилення для осей Y та X.
- - Обрані методи дозволяють для кожної точки моделі адаптивно визначити векторні зсуви та компенсувати як глобальні, так і локальні геометричні спотворення.

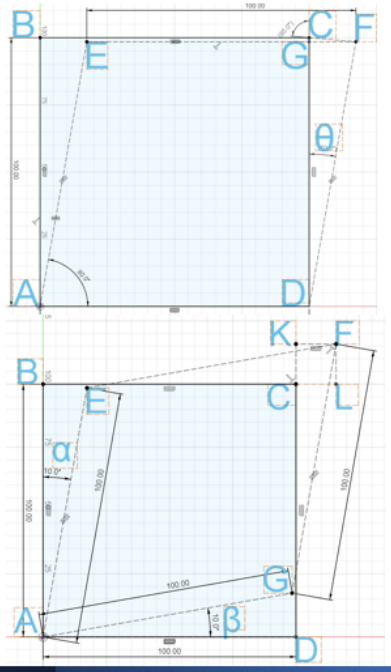

Рисунок Б.8– Слайд 8 демонстраційного матеріалу

Візуалізація реалізації корекції геометрії 3д-моделей

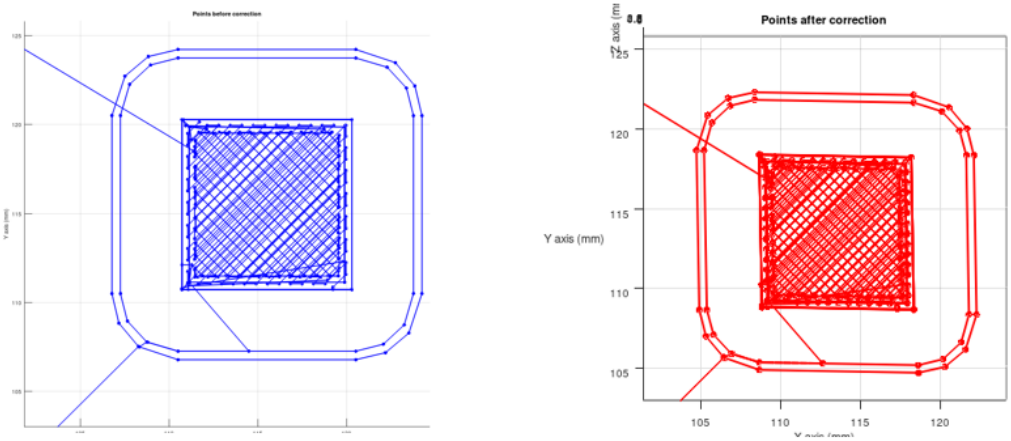

Рисунок Б.9– Слайд 9 демонстраційного матеріалу

Контроль екструзії

- - Впроваджено систему моніторингу фактичного подання філаменту з використанням оптичного енкодера (FC-03), встановленого на фідері.
- - Запланована екструзія визначається параметром F у G-коді, що задає бажану швидкість подачі матеріалу.
- - Реальна швидкість подачі розраховується на основі імпульсів від оптичного енкодера та порівнюється із запланованою; якщо вона нижча за (DESIRED × THRESHOLD, де THRESHOLD за замовчуванням 0.9), система генерує попередження «Недоподача» і приймає заходи, наприклад, зупиняє друк.

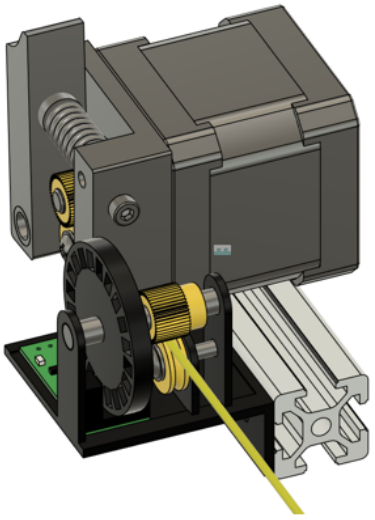

Рисунок Б.10– Слайд 10 демонстраційного матеріалу

Формат	Позиція.	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кільк.	Примітки
			Текстові документи		
А4	1	ГЮІК 416616.002 ПЗ	Пояснювальна записка		75с.
			Інші документи		
	2		Демонстраційний матеріал		10с.

Змін.	Арк.	Номер докум.	Підпис	Дата	ГЮІК 416616.002 ВД			
Розроб.		Бурлей М. П.			Розробка стенду контролю статико-динамічних параметрів FFF/FDM 3D принтерів Відомість кваліфікаційної роботи	Літера	Аркуш	Аркушів
Перевір.		Разумов-Фризюк Є. А.				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.контр.		Демська Н. П.						
Затв.		Невлюдов І. Ш.						