ДОДАТОК А

Лістинг коду програми

Основний цикл моделювання

```
current_time = 0; % Початковий час
for t = 1:Nt
   % Перевірка, чи триває експонування шару
ifmod(current_time, exposure_time + pause_time) <exposure_time
      % Тепло від екрану до фотополімеру
Q_transfer_screen_polymer = k_screen * (T_screen – mean(T_polymer, 'all')) / d_screen;

      % Теплообмін ванни із середовищем
Q_transfer_vat_env = h_vat * (T_vat – T_env);

      % Додаткове тепло для ванни від LED
Q_vat_effective = Q_vat – Q_transfer_vat_env;
      % Оновлення температури фотополімеру
      for i = 2:Nx-1
         for j = 2:Nz-1
            d2T_dx2 = (T_polymer(i+1, j) – 2*T_polymer(i, j) + T_polymer(i-1, j)) / dx^2;
            d2T_dz2 = (T_polymer(i, j+1) – 2*T_polymer(i, j) + T_polymer(i, j-1)) / dz^2;
T_polymer(i, j) = T_polymer(i, j) + dt * (k_polymer * (d2T_dx2 + d2T_dz2) / (rho_polymer * cp_polymer) + Q_transfer_screen_polymer / (rho_polymer * cp_polymer));
end
end
      % Оновлення температури екрану
T_screen = T_screen + dt * (Q_screen – Q_transfer_screen_polymer) / (rho_screen * cp_screen * d_screen);
      % Оновлення температури ванни
T_vat = T_vat + dt * Q_vat_effective / (rho_vat * cp_vat * d_vat);
end
   % Оновлення часу
current_time = current_time + dt;
end
```

Код розподілу температури ванни, екрану та фотополімеру в процесі друку

```
%% Візуалізація результатів
[X, Z] = meshgrid(1:Nx, 1:Nz);

figure;
surf(X, Z, T_polymer);
shadinginterp;
colorbar;
title('Розподіл температури у фотополімері');
xlabel('X (комірки)');
ylabel('Z (комірки)');
zlabel('Температура, K');

figure;
plot(1:Nt, T_screen * ones(1, Nt), 'r', 'LineWidth', 2);
hold on;
plot(1:Nt, T_vat * ones(1, Nt), 'b', 'LineWidth', 2);
title('Температура екрану та ванни');
legend('Екран', 'Ванна');
xlabel('Час (кроки)');
ylabel('Температура, K');
grid on;
```

Код для автоматизованого керування в ПЗ MatLab

```
% Параметри системи крокового двигуна
step_angle = 1.8; % Крок двигуна, градуси
gear_ratio = 1; % Передавальне відношення
steps_per_revolution = 360 / step_angle; % Кількість кроків на оберт
screw_pitch = 0.005; % Крок гвинта, м (переміщення платформи за один оберт)
motor_torque = 0.5; % Максимальний крутний момент двигуна, Н·м
inertia = 0.01; % Момент інерції системи, кг·м²
friction = 0.05; % Коефіцієнт тертя, Н·м·с
K_motor = 1; % Підсилення двигуна
T_motor = 0.1; % Постійна часу двигуна, с
J = 0.01; % Момент інерції, кг·м²
b = 0.1; % Коефіцієнт тертя, Н·м·с

% Передавальна функція двигуна
numerator = K_motor; % Чисельник передавальної функції
denominator = [J, b, 0]; % Знаменник передавальної функції
motor_tf = tf(numerator, denominator);

% Параметри ПІД-регулятора
Kp = 50; % Пропорційне підсилення
Ki = 10; % Інтегральне підсилення
Kd = 5; % Диференціальне підсилення

% Задане переміщення
desired_position = 0.05; % Задане положення, м
time_step = 0.01; % Крок моделювання, с
simulation_time = 5; % Час моделювання, с

% Замкнена система управління
open_loop_system = pid_tf * motor_tf; % Відкрита система
closed_loop_system = feedback(open_loop_system, 1); % Замкнена система

% Ініціалізація змінних
current_position = 0; % Початкове положення платформи, м
current_velocity = 0; % Початкова швидкість, м/с
current_angle = 0; % Початковий кут двигуна, градуси
error_integral = 0; % Інтегральна похибка
previous_error = 0; % Попередня похибка

% Моделювання
```

```
time = 0:time_step:simulation_time;
positions = zeros(size(time));
errors = zeros(size(time));
control_signals = zeros(size(time));

for i = 1:length(time)
    % Обчислення похибки
error = desired_position – current_position;
error_integral = error_integral + error * time_step;
error_derivative = (error – previous_error) / time_step;

    % ПІД-регулятор
control_signal = Kp * error + Ki * error_integral + Kd * error_derivative;
    % Обмеження сигналу для двигуна
control_signal = max(min(control_signal, motor_torque), -motor_torque);

    % Обчислення кутового прискорення
angular_acceleration = (control_signal – friction * current_velocity) / inertia;

    % Оновлення кутової швидкості та положення
current_velocity = current_velocity + angular_acceleration * time_step;
current_angle = current_angle + current_velocity * time_step;
    % Оновлення лінійного положення платформи
current_position = current_angle / 360 * screw_pitch * gear_ratio;
    % Збереження даних
positions(i) = current_position;
errors(i) = error;
control_signals(i) = control_signal;
previous_error = error;
end

% Візуалізація результатів
figure;
subplot(3, 1, 1);
plot(time, positions, 'b', 'LineWidth', 2);
hold on;
yline(desired_position, 'r--', 'LineWidth', 1.5);
xlabel('Час, с');
ylabel('Положення, м');
title('Рух платформи');
legend('Фактичне положення', 'Задане положення');
grid on;
subplot(3, 1, 2);
plot(time, errors, 'k', 'LineWidth', 2);
```

```
xlabel('Час, с');
ylabel('Похибка, м');
title('Похибка регулятора');

grid on;
subplot(3, 1, 3);
plot(time, control_signals, 'g', 'LineWidth', 2);

xlabel('Час, с');
ylabel('Сигнал керування, Н·м');
title('Сигнал керування двигуном');
grid on;
% Візуалізація характеристик
% 1. Годограф Ніквіста
figure;
nyquist(closed_loop_system);
title('Годограф Ніквіста');
grid on;
% 2. Перехідна характеристика
figure;
step(closed_loop_system);
title('Перехідна характеристика');
xlabel('Час, с');
ylabel('Вихід сигналу');
grid on;
% 3. Імпульсна характеристика
figure;
impulse(closed_loop_system);
title('Імпульсна характеристика');
xlabel('Час, с');
ylabel('Вихід сигналу');
grid on;
```

Додаток Б

Демонстраційний матеріал

Формат	Позиція.	Позначення	Найменування	Кільк.	Примітки
			Текстові документи		
А4	1	ГЮИК.502840.001ПЗ	Пояснювальна записка	64	с.
			Інші документи		
А4	2		Лістинг програми	5	с.
А4	3		Демонстраційний матеріал	16	с.

					ГЮИК.502840.001ВД
Змін	А	№ докум.	Підпис	Дата	
Розроб.		Карпов В.О.			Розробка моделі контролю температури фотополімерного 3D-друку
Перевір		Замірець М.В.			
Н.конт.		Теслюк С.І.			
Затв.		Невлюдов І.Ш.			

Літера	Аркуш	Аркушів
н	1	1

Кафедра КІТАР ХНУРЕ