

---

```
function index

% Desired Value = 45px -- Starting Value = 500px

% Define the PID controller function
pidControllerFunction = pidController(45, 0.1, 0.5, 0.1);

% Initialize arrays to store time and output data
time = 0:0.1:12; % Time vector from 0 to 12 milliseconds
pidOutput = zeros(size(time)); % Initialize PID output array

% Calculate initial PID output
pidOutput(1) = pidControllerFunction(500);

% Iterate over time steps
for i = 2:numel(time)
    % Calculate PID output for current time step
    pidOutput(i) = pidControllerFunction(pidOutput(i-1));
end

% Plot the PID output
plot(time, pidOutput);

% Add title and labels
title('Output from the PID');
xlabel('Time (ms)');
ylabel('Output (px)');
end

% PID Controller code
function pidController = pidController(desiredValue, P, I, D)
    % Initialize variables
    currentError = 0;
    integral = 0;
    previousError = 0;

    % Define the nested function
    function output = controller(currentValue)
        % Calculate the error
        currentError = desiredValue - currentValue;

        % Update integral term
        integral = integral + currentError;

        % Calculate derivative term
        derivative = currentError - previousError;

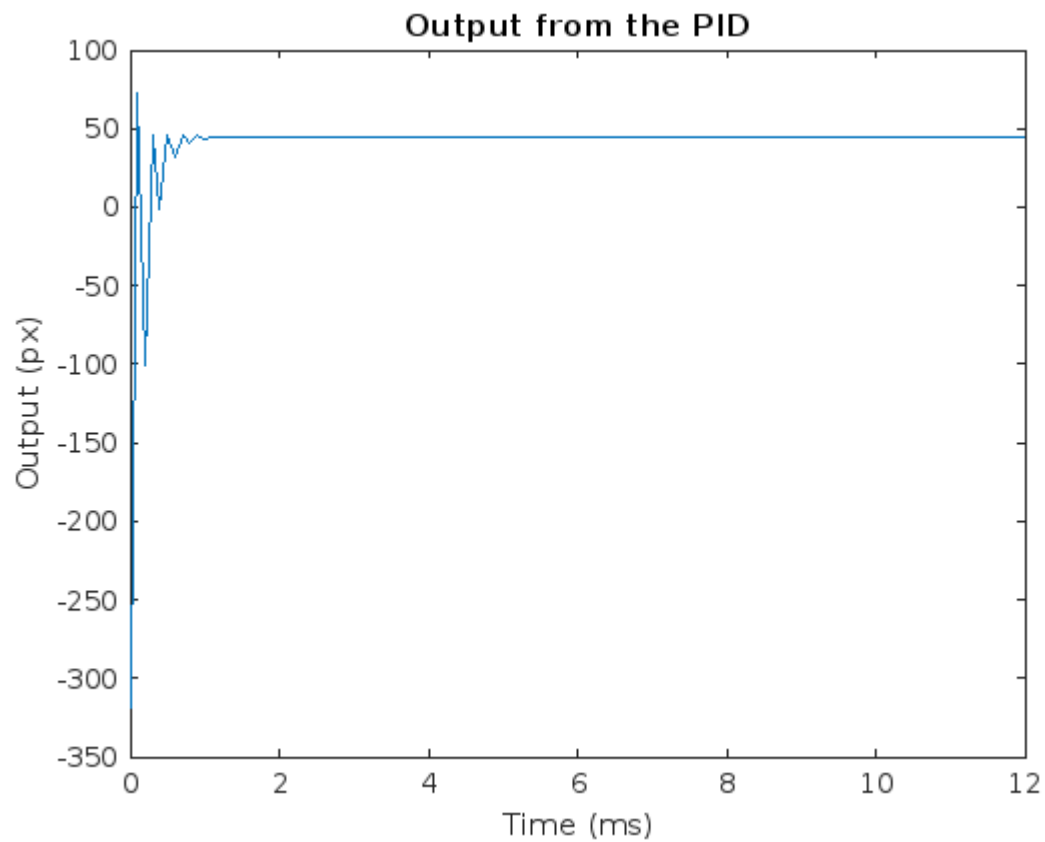
        % Calculate PID output
        output = P * currentError + I * integral + D * derivative;
```

---

---

```
    % Update previous error
    previousError = currentError;
end

% Assign the nested function to the output
pidController = @controller;
end
```



*Published with MATLAB® R2023b*