DC Motor Position Control using Arduino

Aim

The aim of this experiment is to design and implement a PID position controller using an Arduino Mega for controlling the position of a DC motor.

Objective

- Rotate the DC motor by 180 degrees from any initial position.
- Ensure that the task adheres to specific design constraints:
  - Rise time of 0.5 seconds
  - Settling time of 1 second
  - Maximum overshoot of 10%

Control Algorithms

The system utilizes a Proportional-Integral-Derivative (PID) control algorithm to achieve precise motor position control. The PID controller computes the error between the desired motor position and the actual position, and adjusts the motor's position accordingly by varying the pulse width modulation (PWM) signals.

- Proportional Control (P): Responds proportionally to the current error, aiming to correct it by moving the motor closer to the target position.
- Integral Control (I): Accumulates the error over time and corrects for any persistent errors that the proportional control may not address fully.
- Derivative Control (D): Predicts future errors by calculating the rate of change of the error, allowing the controller to dampen the system and reduce overshoot.

Code Implemented

The code is written in Arduino C++ and implements the following steps:

- Initialization: Set up the motor control pins, potentiometer (for feedback), and PID parameters.

- Setup Function: Initialize the system, read initial potentiometer values, and compute the target angle.

- Loop Function:

  - Continuously read the current motor position via the potentiometer.

  - Compute the PID control errors (proportional, integral, and derivative).

  - Adjust the PWM signals to control motor direction and speed based on the PID output.

  - Log the relevant parameters for monitoring and analysis.

Challenges Faced and Solutions

- Nonlinear Response of the Motor:

  - Challenge: The motor's response to PWM signals was not perfectly linear, especially at lower angles.

  - Solution: Implemented nonlinear mapping of the motor angles and adjusted the PID parameters to compensate for these non-linearities.

- System Stability:

  - Challenge: Initial PID tuning led to overshoot beyond the acceptable limit.

  - Solution: Adjusted the derivative gain (Kd) to improve damping and minimize overshoot, and fine-tuned the proportional gain (Kp) to achieve the desired rise time and settling time.

- Time Delays:

  - Challenge: Ensuring that the control loop runs in real-time while maintaining the desired time constraints (rise time, settling time).

  - Solution: Carefully selected the delay in the loop function and optimized the PID calculations to ensure real-time performance.

Results

- The motor was successfully controlled to rotate by 180 degrees from any given starting position.

- The system adhered to the specified rise time of 0.5 seconds and settling time of 1 second.

- The observed overshoot was within the acceptable 10% range, demonstrating the effectiveness of the PID controller.

Observations and Inferences

- The PID control system provided precise control over the motor position, allowing it to reach the target angle within the specified constraints.

- The fine-tuning of PID parameters is crucial for achieving a balance between response speed and stability.

- Nonlinear characteristics of the motor and potentiometer need careful consideration during the design phase, as they can significantly affect system performance.

- The control logic implemented was effective in overcoming challenges related to motor nonlinearity and system stability.