Technical Report: Design and Simulation of a Speed-Controlled Two-Quadrant DC Motor Drive (Motor Control) by Chibudom Melvin Amadi-Duru

Subject: Electrical Machine Control Systems

Based on: Design and Simulation of a Speed-Controlled Two-Quadrant DC Motor Drive Using MATLAB

1. Executive Summary

This project focuses on the development of a closed-loop speed control system for a separately excited DC motor. The primary objective is to maintain constant field flux while regulating the motor's speed through armature voltage control[1]. The system employs a **cascaded control structure**, featuring an inner current control loop for torque regulation and an outer speed control loop for velocity tracking. The power stage utilizes a three-phase bridge converter, enabling **two-quadrant operation**, which allows for both motoring and regenerative braking. Verification via MATLAB simulation confirms that the designed Proportional-Integral (PI) controllers meet safety constraints, such as a maximum armature current of 20 A, while achieving stable dynamic response.

2. System Architecture and Modeling

The drive system integrates power electronics, electromechanical components, and feedback instrumentation. The core topology consists of a transformer-fed bridge converter driving the DC motor armature, with feedback provided by a tachogenerator and current transducer.

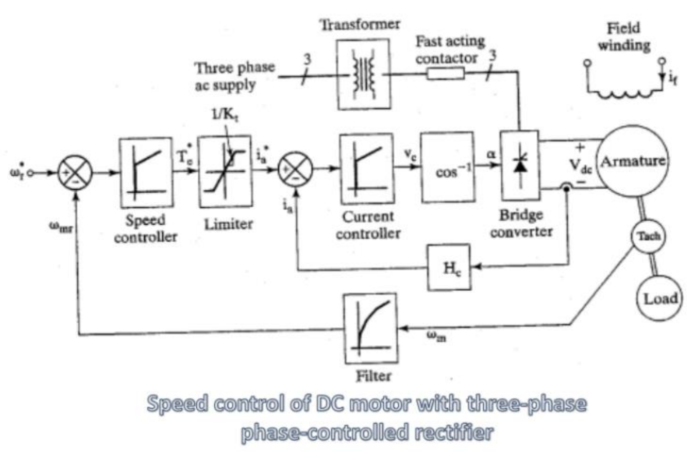

Figure 1: Speed control of DC motor with three-phase phase-controlled rectifier

2.1 Electromechanical Plant (DC Motor)

The plant is a 220 V, 1470 rpm DC motor. Its dynamic behavior is governed by electrical (armature) and mechanical equations.

• Electrical Time Constants: The motor's response is defined by two primary time constants derived from its differential equations: $T_1 = 0.1077$s and $T_2 = 0.0208$s4.

• Transfer Function: The relationship between armature voltage (V_s) and armature current (I_a) is modeled as: $\frac{I_a(s)}{V_s(s)} = \frac{K_1(1+sT_m)}{(1+sT_1)(1+sT_2)}$Where $K_1 = 0.0449$is the system gain and $T_m = 0.7$s is the mechanical time constant5. 2.2 Power Conversion Stage (Bridge Converter) A three-phase phase-controlled rectifier converts the 230 V, 60 Hz AC supply into a controlled DC voltage.

• Gain (K_r): The converter acts as a linear amplifier with a gain of 31.05 V/V, translating a ± 10V control signal into a maximum DC output of 310.5 V.

• Dynamics: The converter introduces a small delay modeled by a time constant T_r, resulting in the transfer function: $G_r(s) = \frac{31.05}{1+0.00138s}$.

2.3 Instrumentation and Feedback

• Current Transducer (H_c): To ensure the motor current stays within the safety limit of 20 A, the current is scaled to a control voltage range (7.09 V max), yielding a gain of 0.355 V/A9.

• Tachometer: A tachogenerator provides real-time angular velocity feedback (ω_m) to the speed controller.

Control System Synthesis

The control strategy employs a Cascade Control structure. This design separates the fast dynamics of the current loop from the slower dynamics of the speed loop, enhancing stability and allowing for precise current limiting

3.1 Inner Loop: Current Controller Design The current controller regulates the armature current (i_a) to track the reference command (i_a^*). A PI controller is tuned to cancel the motor's dominant electrical pole.

• **Pole Cancellation**: The controller's time constant (T_c) is set equal to the motor's electrical time constant ($T_2 = 0.0208$ s) to simplify the open-loop dynamics.

• **Gain Calculation**: To achieve a fast response, the proportional gain (K_c) is calculated as 2.3313.

• **Loop Approximation**: For the design of the outer loop, the closed current loop is approximated as a first-order system with a time constant $T_i \approx 0.0027$s14. This approximation is valid because the higher-order dynamics are negligible within the operational bandwidth.

3.2 Outer Loop: Speed Controller Design The speed controller generates the current reference (i_a^*) based on the speed error.

• **Tuning Strategy**: The controller is designed using the "Symmetrical Optimum" principle or similar bandwidth considerations, balancing speed of response with damping (stability).

• **Parameters**: o Gain (K_s): Calculated as 28.7316. o Time Constant (T_s): Set to 0.0188 s (derived as 4 ×the total delay T_4)17.

• **Limiter**: A saturation block is placed after the speed controller to clamp the current reference, ensuring the motor current never exceeds the safety rating of 20 A.

Operational Analysis and Results

4.1 Two-Quadrant Operation The drive is designed for two-quadrant operation, enabling it to operate in:

Forward Motoring: Converting electrical energy to mechanical torque to drive the load.

Regenerative Braking (Forward Braking): The motor acts as a generator, converting kinetic energy back into electrical energy which is fed back to the supply or dissipated.

Mechanism: The speed controller commands a negative torque, and the converter manages the bidirectional power flow to decelerate the load efficiently.

4.2 Simulation Verification: The system design was validated via frequency response (Bode plots) and time-domain step response simulations.

• **Stability**: The frequency response analysis confirmed adequate phase margins, validating the use of the first-order approximation for the current loop.

• **Transient Response**: Time-domain simulations demonstrated that the system tracks reference speed steps with minimal overshoot while strictly adhering to the current limits during startup and braking.

Conclusion

This project successfully designed a robust DC motor drive capable of precise speed tracking and regenerative braking. By utilizing a cascaded PI control structure, the system effectively decouples the mechanical and electrical dynamics. The simulation results validate the calculated controller gains ($K_c = 2.33$, $K_s = 28.73$), proving that the system can maintain high performance under varying loads while ensuring the safety of the motor and power electronics.

References

[1] *Design and Simulation of a Speed-Controlled Two-Quadrant DC Motor Drive Using MATLAB*.