%% Mini Project for Advanced Electrical Drives Documentation

Mini Project: Advanced Electrical Drives

Overview

This document describes the MATLAB mini project for Advanced Electrical Drives, with key equations, code snippets, and workflow explanations. The workflow is split based on the value of kappa (κ).

Machine Parameters

```
Psi_f = 90e-3;  % Field flux linkage [Vs]
L_sd = 200e-6;   % d-axis inductance [H]
L_sq = 500e-6;   % q-axis inductance [H]
i_max = 500;     % Maximum stator current [A]
U_dc = 350;      % DC link voltage [V]
u_smax = U_dc/sqrt(3);  % Maximum stator voltage [V]
p = 4;           % Number of pole pairs
```

Calculated Parameters

```
i_sc = Psi_f / L_sd;      % Short-circuit current [A]
kappa = i_sc / i_max;    % Normalized short-circuit current
chi = (L_sq-L_sd)/(2*L_sd);  % Saliency
```

Key Equations

1. **Short-circuit current:** $i_{sc} = \frac{\Psi_f}{L_{sd}}$
2. **Kappa (normalized short-circuit current):** $\kappa = \frac{i_{sc}}{i_{max}}$
3. **Saliency:** $\chi = \frac{L_{sq} - L_{sd}}{2L_{sd}}$
4. **Torque (general reference):** $T_e = \frac{3}{2}p(\Psi_f i_{sq} + (L_{sd} - L_{sq})i_{sd}i_{sq}))$
5. **MTPA current calculation:** $i_{sd}^{MTPA}, i_{sq}^{MTPA}$ (via `calc_i_ref_MTPA`)
6. **Constant Torque locus:** $i_{sd}^{LCT}, i_{sq}^{LCT}$ (via `calc_i_s_ref_LCT`)
7. **MA circle trajectory:** i_{sd}^{MA}, i_{sq}^{MA} (via `calc_i_s_ref_MA`)
8. **MTPF trajectory (field weakening):** $i_{sd}^{MPTF}, i_{sq}^{MPTF}$ (via `calc_i_s_ref_MPTF`)
9. **Dynamic speed thresholds:** $\omega_A = f(T_e), \omega_B = f(T_e), \omega_C = f(T_e), \omega_{MA\_max} = f(T_e)$

Workflow Split Based on Kappa

Case 1: κ >= 1 (High Short-Circuit)

Follow Tasks 1.1 and 1.2 (Reference Currents + Torque-Speed Characteristics):

```
if kappa >= 1
    % Task 1.1 & 1.2: Calculate reference currents and speed thresholds
    [i_sd_ref, i_sq_ref] = calc_reference_currents(T_e_ref, omega_s_ref, kappa);
    draw_contours;
    plot_current_trajectory(T_e_ref, omega_s_ref);
    plot_torque_over_speed_map(T_e_ref, omega_s_ref);
end
```

Case 2: κ < 1 (Low Short-Circuit)

Follow Task 1.3 (Field Weakening / MTPF adjustments):

```
if kappa < 1
    i_max = 400;  % Adjusted max current
    kappa = i_sc / i_max;
    [i_sd_ref, i_sq_ref] = calc_reference_currents(T_e_ref, omega_s_ref, kappa);
    plot_current_trajectory(T_e_ref, omega_s_ref);
    plot_torque_over_speed_map(T_e_ref, omega_s_ref);
end
```

Helper Functions with Dynamic Speed Thresholds

Calculate Reference Currents

```
function [i_sd_ref, i_sq_ref] = calc_reference_currents(T_e_ref, omega_s_ref,
kappa)
    % Determine max torque and dynamic speed thresholds
    T_e_max = calc_maximum_torque(T_e_ref);
    omega_A = get_max_omega_A(T_e_ref);
    omega_B = get_max_omega_B(T_e_ref);

    if kappa < 1
        omega_C = get_max_omega_C(T_e_ref);
    else
        omega_MA_max = get_omega_max(T_e_ref);
    end
```

```
    % Select operating region based on kappa and speed thresholds
    if kappa >= 1
        if omega_s_ref >= omega_B && omega_s_ref <= omega_MA_max
            [i_sd_ref, i_sq_ref] = calc_i_s_ref_MA(omega_s_ref);
        elseif omega_s_ref >= omega_A
            [i_sd_ref, i_sq_ref] = calc_i_s_ref_LCT(T_e_ref, omega_s_ref);
        else
            [i_sd_ref, i_sq_ref] = calc_i_ref_MTPA(T_e_ref);
        end
    else
        if omega_s_ref >= omega_C
            [i_sd_ref, i_sq_ref] = calc_i_s_ref_MPTF(omega_s_ref);
        elseif omega_s_ref >= omega_B
            [i_sd_ref, i_sq_ref] = calc_i_s_ref_MA(omega_s_ref);
        elseif omega_s_ref >= omega_A
            [i_sd_ref, i_sq_ref] = calc_i_s_ref_LCT(T_e_ref, omega_s_ref);
        else
            [i_sd_ref, i_sq_ref] = calc_i_ref_MTPA(T_e_ref);
        end
    end
end
```

Maximum Torque & Dynamic Thresholds

```
function T_e_max = calc_maximum_torque(T_e_ref)
end

function omega_A = get_max_omega_A(T_e_ref)
end

function omega_B = get_max_omega_B(T_e_ref)
end

function omega_C = get_max_omega_C(T_e_ref)
end

function omega_MA_max = get_omega_max(T_e_ref)
end
```

Reference Current Calculations

- `calc_i_ref_MTPA(T_e)`
- `calc_i_s_ref_LCT(T_e, omega_s)`
- `calc_i_s_ref_MA(omega_s)`
- `calc_i_s_ref_MPTF(omega_s)`

Flowchart for Kappa-Based Decision (Eraser.io Compatible)

```
flowchart TD
    A[Torque Request T_e_ref + Speed omega_s_ref] --> B{Check kappa}
    B -->|kappa >= 1| C[Tasks 1.1 & 1.2]
    B -->|kappa < 1| D[Task 1.3]
    C --> E[calc_reference_currents() with dynamic omega_A/B/MA_max]
    D --> F[calc_reference_currents() with dynamic omega_A/B/C]
    E --> G[Select operating region based on thresholds (MTPA, LCT, MA)]
    F --> H[Select operating region based on thresholds (MTPA, LCT, MA, MTPF)]
    G --> I[Calculate i_sd_ref, i_sq_ref]
    H --> J[Calculate i_sd_ref, i_sq_ref]
```