ECE 7440: Modern Control of Power Electronics

Three-Phase Motor Speed Control in an EV Drive

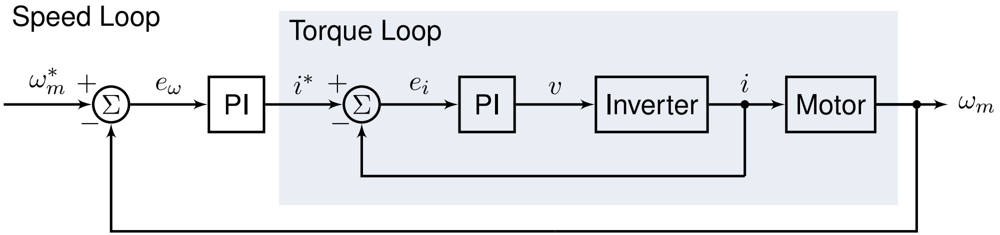
Final Presentation and Report

Manveer Aujla Gavin Feher Douglas Gray

13 December 2025

Background

- Speed Control of a three-phase (3ϕ) motor needs:
  - Torque control for accelerating/decelerating
  - Current control for torque
- 3ϕ **inverter** commonly used for actuation
- Simply applying 3ϕ voltages won't work
- Need timing with rotor position
- Answer: **Field-Oriented Control (FOC)**
- Two control loops:
  - Outer loop: Speed control
  - Inner loop: Torque (current) control



Control Objectives

- Perform FOC to drive an EV at a constant speed
- Compensate for undesired motor effects
  - Back EMF
  - Cross-coupling
- Compensate for external disturbances
  - Rolling resistance
  - Drag force (aerodynamics)

Parameter	**Value**
Electrical	
Inverter DC Voltage (V_{dc})	800 V
Switching Frequency (f_{sw})	10 kHz
Machine	
Motor Pole Pairs (p)	2
Motor Flux Linkage (λ)	0.04 Wb
Stator Resistance (R_s)	15 mΩ
d-axis Inductance (L_d)	250 µH
q-axis Inductance (L_q)	250 µH
Saturation Torque	150 N m
Rotor Inertia (J_m)	1.125 kg m^2
Vehicle	
Wheel Radius (r_w)	0.3 m
Gearbox Ratio (k_g)	12
Vehicle Mass (m)	1800 kg
Effective Frontal Area (A_f)	2.2 m^2
Rolling Resistance Coeff. (C_{rr})	0.1
Drag Constant (k_d)	9.26×10^{-6} kg m^2

Plant Modelling for Speed Control

- Outer control loop
- Nonlinear model:

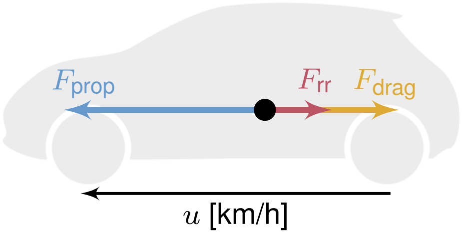
$$\frac{d\omega_m}{dt} = \frac{1}{J_m}(\tau_e - \tau_{\text{rr}} - k_d\omega_m^2)$$

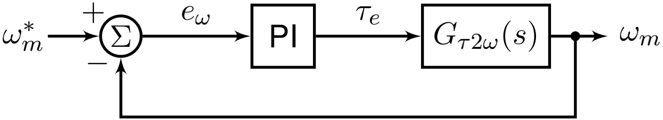
- Linearized model:

$$\frac{\tilde{\omega}_m}{\tilde{\tau}_e} = \frac{1}{sJ_m + 2k_d\Omega_m}$$

- Least stable case ($\Omega_m = 0$):

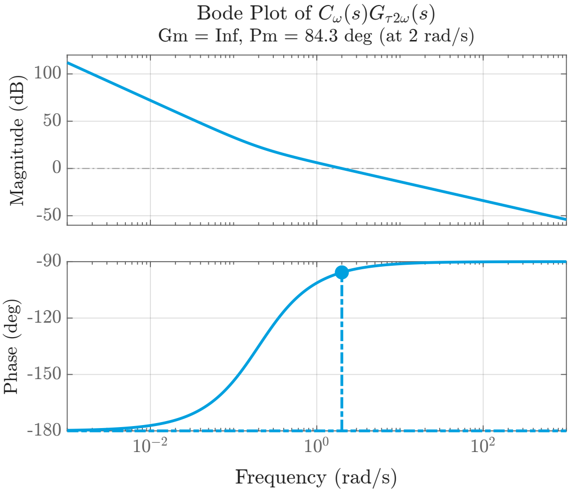
$$G_{\tau 2\omega}(s) = \frac{\tilde{\omega}_m}{\tilde{\tau}_e} = \frac{1}{sJ_m}$$





Speech Controller Design

- Use a PI to track constant speeds
- Assume inner controller is much faster; model as a unity gain
- Crossover frequency $\omega_c = 2$ rad/s
  - Saturation torque limits dynamic response time
- Place zero at $\omega_c/10$
  - Maintains stability by not lowering phase near ω_c



Park Transformations for FOC

- To use PI current controllers, convert 3ϕ currents to dq frame ($\theta_e = \omega_e t$):

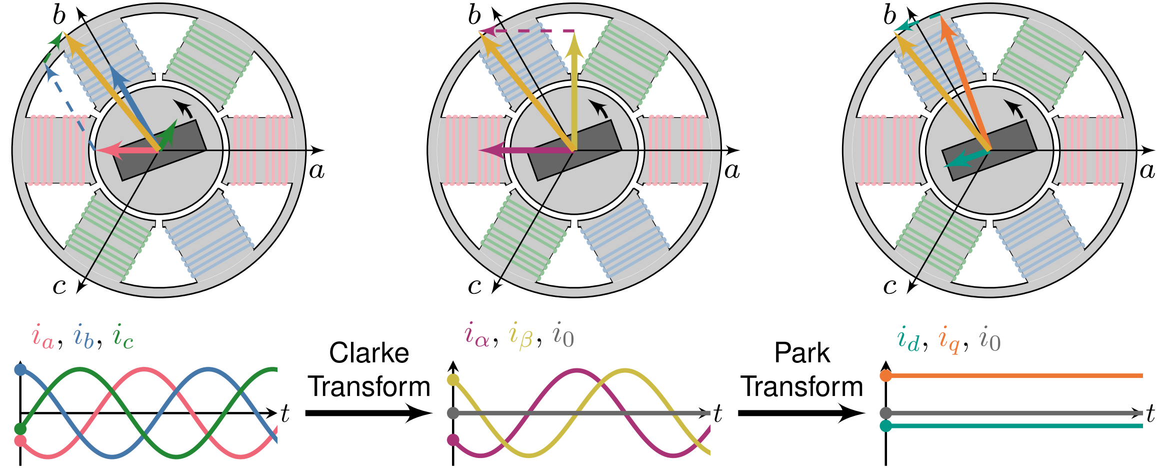
$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta_e) & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin(\theta_e) & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

- To apply 3ϕ inverter voltages, convert back to abc frame:

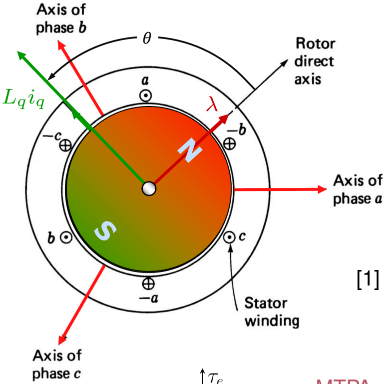
$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) & 1/2 \\ \cos(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e - \frac{2\pi}{3}) & 1/2 \\ \cos(\theta_e + \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) & 1/2 \end{bmatrix} \begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix}$$

- PLECS abc-dq blocks omit zero-sequence component; in balanced 3ϕ, $i_0 = v_0 = 0$

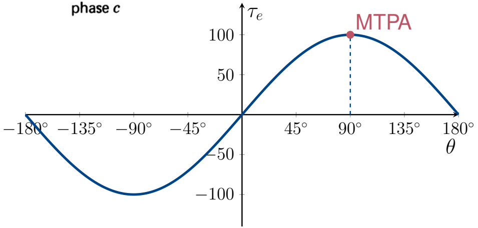
$$\vec{i}_{\text{net}} = \vec{i}_a + \vec{i}_b + \vec{i}_c = \vec{i}_\alpha + \vec{i}_\beta = \vec{i}_d + \vec{i}_q$$



- The net flux vector of all three phases **rotates**, and the rotor flux **follows**
- In non-salient pole machines ($L_d = L_q$), i_q produces a torque, i_d does not
- Induced torque is $\tau_e \approx \frac{3}{2} p \lambda i_q$
- Control electrical angle $\theta = 90^\circ$ between fluxes for maximum torque per amp
- Knowing rotor angle and currents, can go to dq and control for desired i_q (torque!)



[1]



Plant Modelling for Torque Control

- Inner control loop
- Controlling torque means controlling $i_d = 0$ and i_q according to

$$i_q = \frac{2}{3p\lambda}\tau_e$$

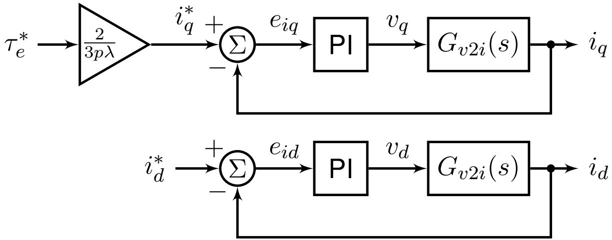
- Motor model in dq frame:

$$v_d = R_s i_d + L_d \frac{d}{dt} i_d - L_q p \omega_m i_q$$

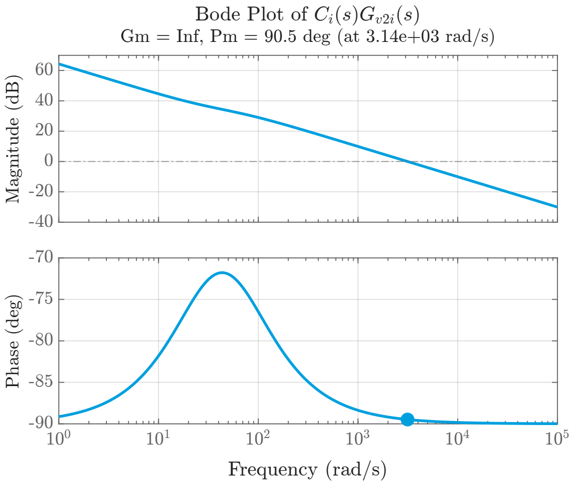
$$v_q = R_s i_q + L_q \frac{d}{dt} i_q + L_d p \omega_m i_d + \lambda p \omega_m$$

- Use feed-forward compensation to cancel cross-coupling and back EMF
  - Can design controller using the RL response!

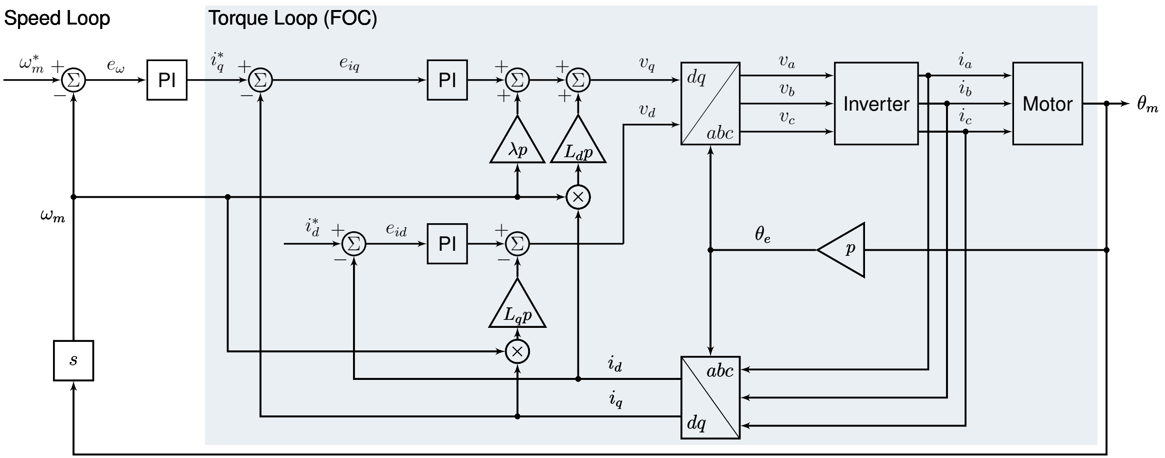
$$G_{v2i}(s) = \frac{\tilde{i}_d}{\tilde{v}_d} = \frac{1}{sL_d + R_s}$$



- Use a PI to track constant currents
  - Controller computes the required voltage
- $\omega_c = 2\pi \times 500$ rad/s
  - 20 times slower than $f_{\text{sw}} = 10$ kHz
  - Much faster than outer loop
- Place zero at $\omega_c/100$
- Can use the same controller for i_d and i_q since $L_d = L_q$



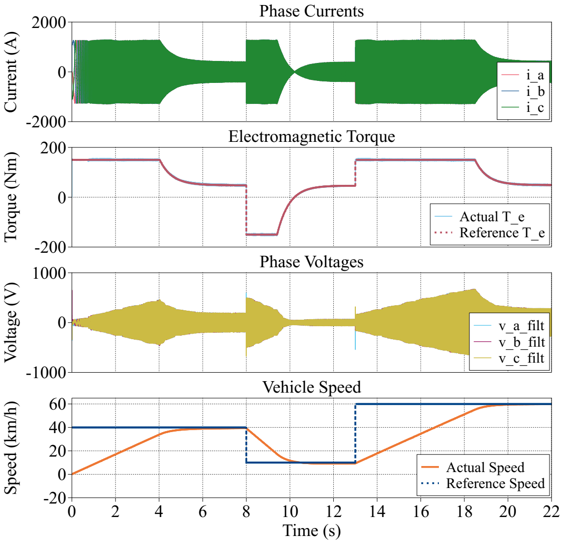
Speed Control using FOC Block Diagram



Offline Simulation Results of Full System

- Results from offline PLECS simulation, $\Delta t = 5\ \mu s$
- Torque and current magnitudes increase proportionally ($|\tau| \propto |\vec{i}_{abc}|$)
- Speed and voltage magnitudes increase proportionally ($|\omega| \propto |\vec{v}_{abc}|$)
- Voltages also depend on currents

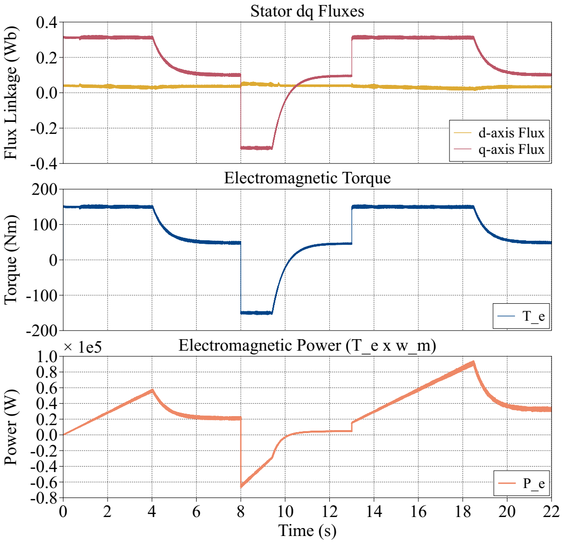
Time (s)	Ref. Speed (km/h)
0	40
8	10
13	60



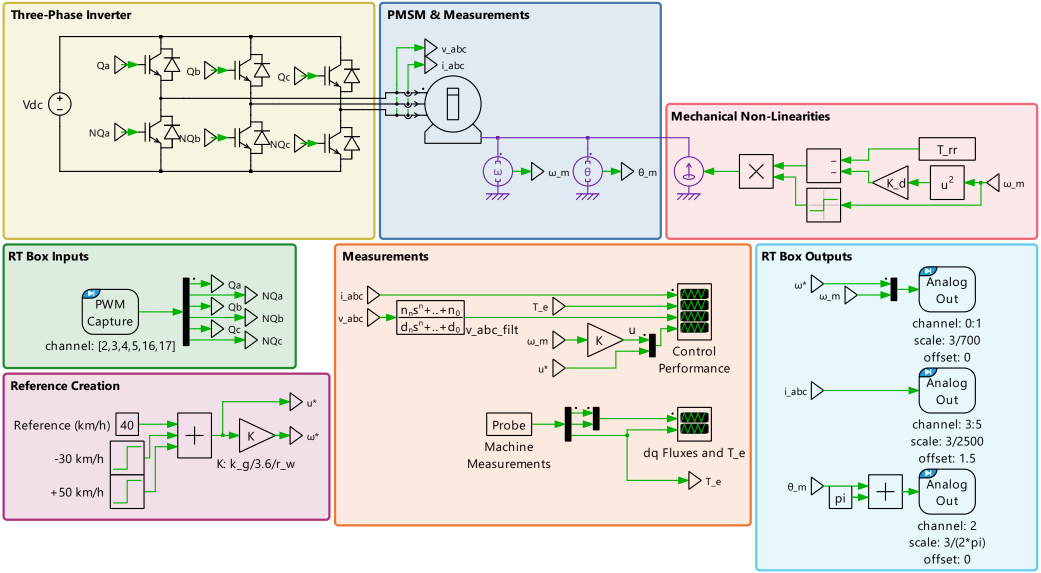
Other Interesting Plots. . .

- $\psi_d = L_d i_d + \lambda$ and $\psi_q = L_q i_q$
- i_q is directly proportional to torque
- i_d is controlled to zero, permanent magnet flux $\lambda = 0.04\,\text{Wb}$
- Electromagnetic power ($P_e = \tau_e \omega_m$) peaks at about 95 kW

Time (s)	**Ref. Speed (km/h)**
0	40
8	10
13	60

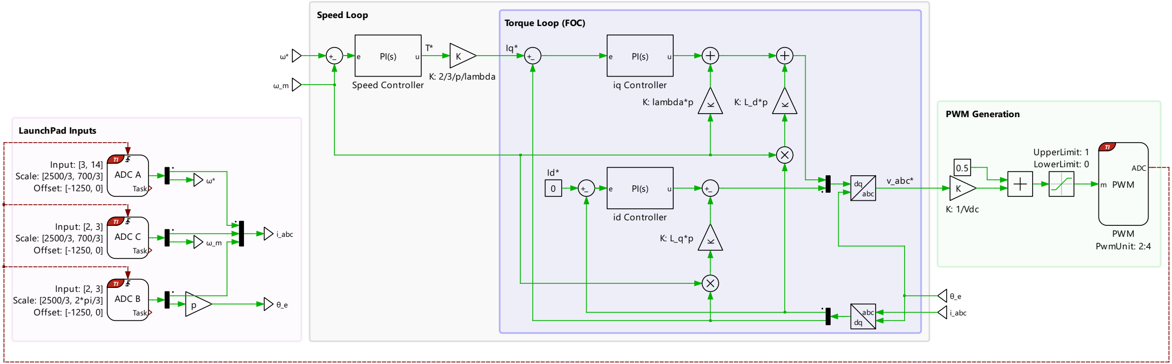


Plant PLECS Schematic for Real-Time Simulation



Controller PLECS Schematic for Real-Time Simulation

- Model flashed onto TI LaunchPad using TI C2000 Support Package
- Six analog inputs feed LaunchPad, six digital outputs go to RTBox

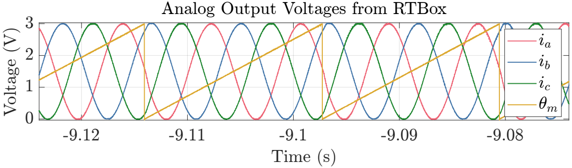


Hardware-in-the-Loop Implementation

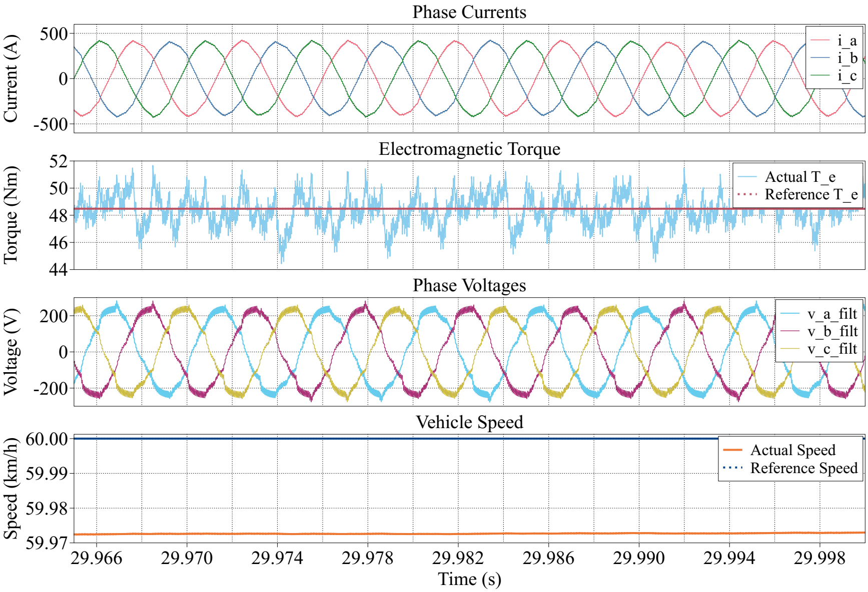
- Plant running on RTBox, $\Delta t = 5$ µs
- Controller running on MCU, $\Delta t = 10$ µs
- Oscilloscopes to view and capture waveforms
- External mode in PLECs to view signals live

RTBox Analog Output Signal Translation

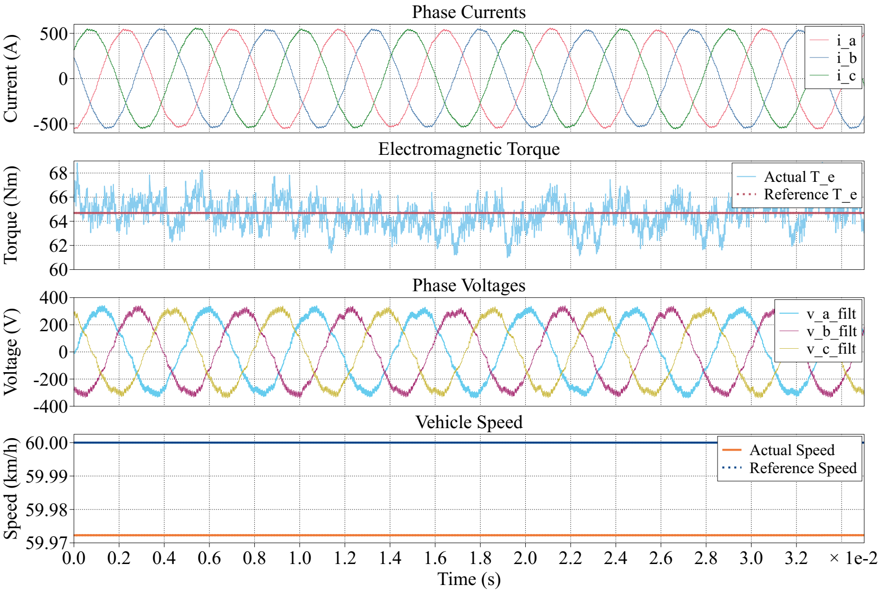
Signal	Scaling Factor	Offset
Rotor Angle (θ_m)	3/2π	1.5
Rotor Angular Speed (ω_m^*, ω_m)	3/700	0
Phase Current (i_a, i_b, i_c)	3/2500	1.5



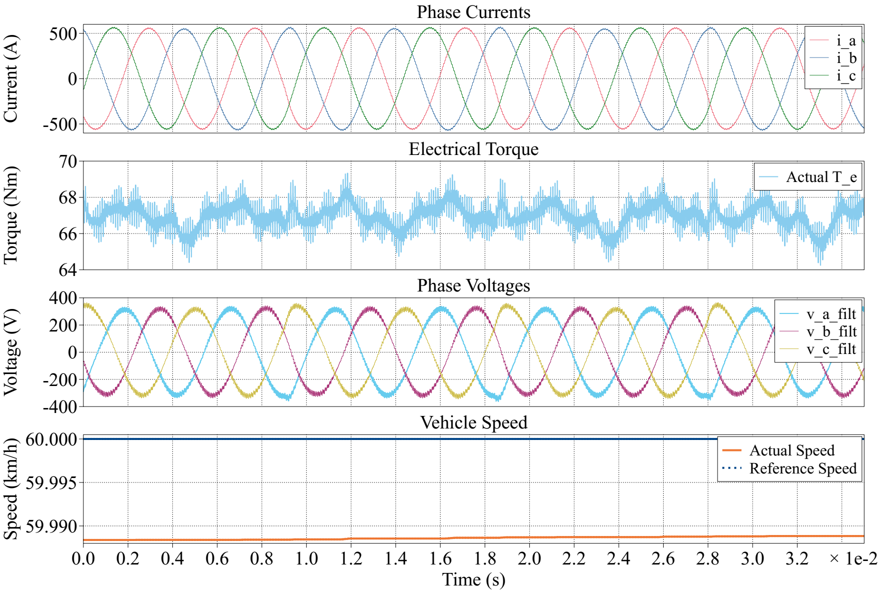
Offline Simulation Waveforms



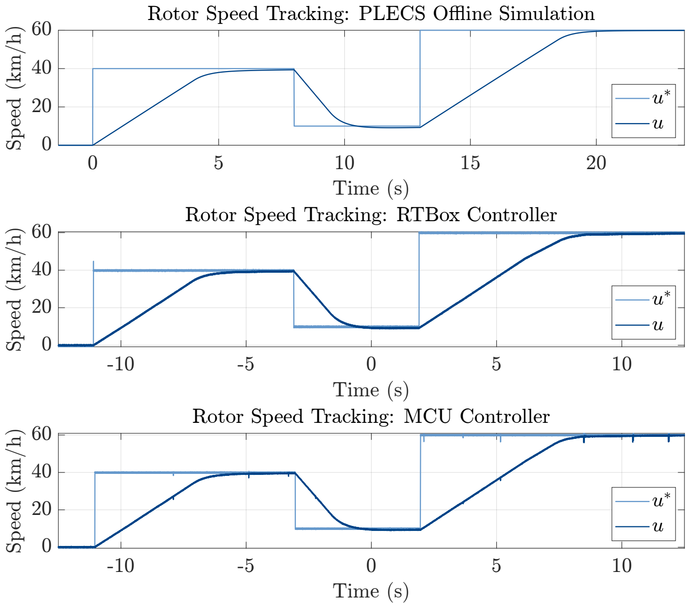
RTBox Controller Simulation Waveforms



MCU Controller Simulation Waveforms



Scope Data: Transient Response Comparison

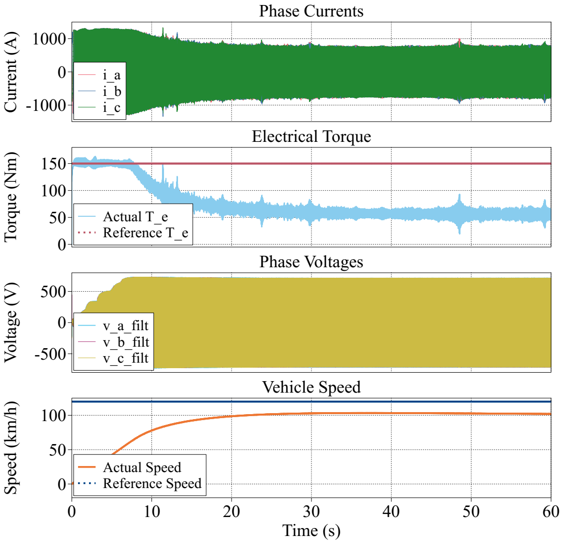


- Transient performance consistent across all simulations
- Linear region from torque saturation
- Faster speed controller bottlenecked by torque

Time (s)	Ref. Speed (km/h)
0	40
8	10
13	60

Evaluation of Performance

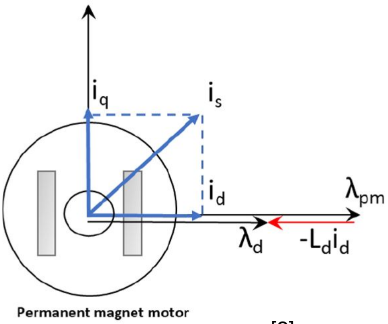
- Speed rise time depends on reference
  - Due to 150 N m saturation torque
- Torque rise time of 0.805 ms
  - Confirms inner loop is much faster than outer loop
- Maximum speed limited by
  - DC-link voltage V_{dc}
  - Switching frequency f_{sw}
    - Requires smaller time step
- Despite tracking 100 km/h, we get non-sinusoidal waveforms and large torque ripple



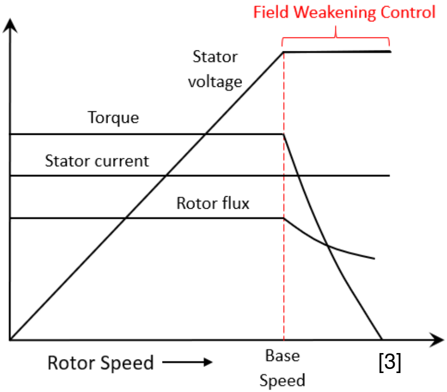
Limitations and Future Work

- High speeds cause problems with the controller when using the MCU
- Different steady state torques in the offline and real-time simulations

- Implement field-weakening to achieve higher speeds (control for negative i_d)
- Test controller on a real inverter and PMSM setup



[2]

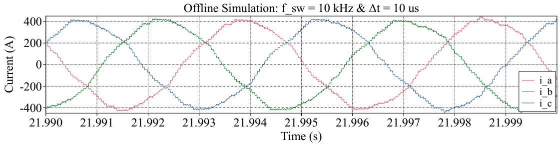


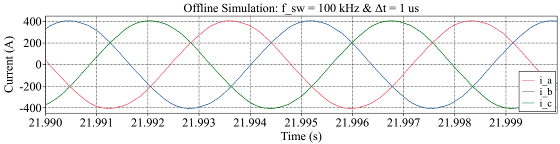
[3]

Discussion: Time Step and Switching Frequency

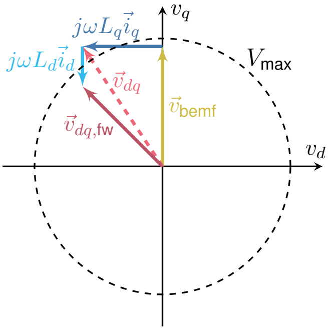
University of Manitoba

- Speed bottleneck: At higher motor speeds, electrical frequency (of voltages and currents) increases and becomes comparable to switching frequency since $\omega_e = p\omega_m$
- "Why not increase switching frequency to get higher motor speeds?"
  - Currently at 10 kHz, time constant is $1/f_{\text{sw}} =$ 100 µs
  - Simulation should be at least 10 times faster than switching frequency, i.e., $\Delta t \leq$ 10 µs
  - Already operating RTBox at 5 µs and MCU at 10 µs!
- "Why not decrease the time step to get higher switching frequency?"
  - RTBox only supports down to $\Delta t =$ 2.5 µs for the plant model with external monitoring
  - TI LaunchPad MCU only supports down to $\Delta t =$ 8 µs for the controller model
    - TI LaunchPad *can* go faster, but PLECS code generation adds some overhead
- In the future, could manually code MCU and use faster RTBox model to get higher f_{sw}

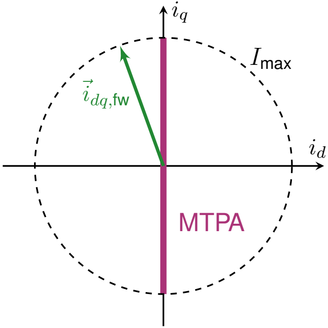




- Above the base speed, $\vec{v}_{dq}$ exceeds $V_{\text{max}} = V_{\text{dc}}/2$
  - End up outside $v_q^2 + v_d^2 = V_{\text{max}}^2$ circle
- Can apply a negative i_d such that $\vec{v}_{dq,\text{fw}} < V_{\text{max}}$

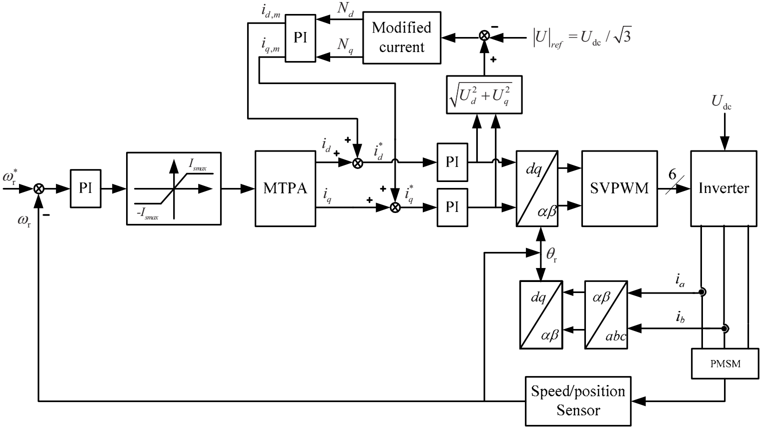


- $\vec{i}_{dq,\text{fw}}$ is no longer on the MTPA curve
  - We lose torque per amp efficiency
- Due to the lower effective $\vec{v}_{\text{bemf}}$, motor can reach higher speeds
  - Rotor's field has been weakened!



Discussion: Field-Weakening Control Algorithm

1. Compute $\vec{v}_{dq}$ from $\vec{i}_{dq}$
2. Measure $|\vec{v}_{dq}|$
   a. If $|\vec{v}_{dq}| > V_{\text{max}}$, go to step 3
   b. Else, apply $\vec{v}_{dq}$ to inverter
3. Generate error signal $v_e = |\vec{v}_{dq}| - V_{\text{max}}$
4. Compute $\vec{i}_{dq,\text{fw}}$ from v_e using a known function or PI controller
5. Update and apply $\vec{v}_{dq,\text{fw}}$ to inverter



[4]

References

[1] D. Wilson, "Intro to Field Oriented Control," Texas Instruments, 2014. [Online]. Available: https://eggelectricunicycle.bitbucket.io/EmbeddedFiles/26-02%20Intro%20to%20FOC.pdf

[2] "Field-Weakening Control," Mathworks. [Online]. Available: https://www.mathworks.com/discovery/field-weakening-control.html

[3] "Field-Weakening Control (with MTPA) of PMSM," Mathworks. [Online]. Available: https://www.mathworks.com/help/mcb/gs/field-weakening-control-mtpa-pmsm.html

[4] K. Sun and Q. Shu, "Field Weakening Operation Control Strategies of Interior Permanent Magnet Synchronous Motor for Electric Vehicles," in *Proceedings of the 30th Chinese Control Conference*, 2011, pp. 3640–3643.