MotorControl Lab Operation and Control Systems Experiment Manual

Revised 2026-10-23 | Motor Control Virtual Laboratory

Software version: v1.0.0 | Manual for v1.0.0

Author and Developer: Dr. Jack Yeh

MotorControl Lab aims to provide a motor control virtual laboratory that runs directly in a web browser. It brings motor dynamics, digital control, and power conversion behavior close to that of real systems into an environment where learners can observe, adjust, and repeatedly verify their results. Through interactive simulation and waveform analysis, learners can connect control theory with the effects of parameter settings on system response, narrowing the gap between theory and practice.

The platform supports both education and engineering. Teachers and students can use it for classroom demonstrations, independent study, and control experiments; engineers can quickly tune controllers, compare designs, and verify simulation results. Within the models and functions provided, users can perform preliminary analysis and repeated tests directly in a browser without first building a separate MATLAB/Simulink model, reducing setup and model construction time.

This manual introduces PMSM current, speed, and position control in sequence, including digital sampling, PWM, voltage limits, feedforward, and anti-windup. Transfer functions assume zero initial conditions and represent nominal, unsaturated, small-signal models; time-domain waveforms are computed using a nonlinear model with PWM switching. English labels in diagrams match the interface. In the HTML edition, select “Enlarge” to inspect the vector diagrams.

Copyright © 2026 Dr. Jack Yeh. All rights reserved. Copyright in the original MotorControl Lab software and educational content, and in the text and figures of this manual, belongs to Dr. Jack Yeh. Reproduction, modification, distribution, or commercial use without authorization is prohibited. Third-party software and resources remain the property of their respective rights holders and are used under their respective license terms.

Version baseline: numerical model dated 2026-10-22; manual revised 2026-10-23. Includes Mechanical Condition and Controller Output / Anti-Windup diagnostics.

Contents

1. Purpose and Quick Start 3

2. Modes and Signal Paths 5

3. Parameters, Units, and Defaults 7

4. PMSM Model and Block Equations 10

5. Control Loop Transfer Functions 18

6. Digital Control and PWM Timing 27

7. Reading Waveforms and Diagnostics 35

8. Bode and FFT Spectrum Analysis 39

9. Six Experiments 44

10. Data Export, FAQ, and Experiment Records 46

11. Appendix: Digital Speed and Position Models 49

MotorControl Lab | © 2026 Dr. Jack Yeh | All rights reserved.

1. Purpose and Quick Start

1.1 Interface Areas

Area	Operation and purpose
Language / Control Mode	Select English, Traditional Chinese, or Simplified Chinese, and Current, Speed, or Position mode
Control block diagram	Select a block to inspect and edit its parameters on the right; bypassed blocks do not participate in that mode
Parameter panel	Enter PI gains, motor parameters, commands, limits, PWM settings, and load conditions
Run	Run again with a snapshot of the current parameters; Stop interrupts the current calculation
Result metrics	Final speed, final iq, peak phase current, voltage saturation fraction, and computation time
Simulation Time Cursor	Inspect CNT, active/pending CCR, ADC, and sensor states at a selected time
Waveform instruments	Switch among control responses, feedback chains, controller/AW diagnostics, and PWM/dead-time waveforms
Loop Bode Analysis	Use current settings or the last simulation snapshot to inspect feedback and command-tracking frequency responses
FFT / Spectrum Analysis	Select a recorded time-domain signal, source, and time range; compute and export its spectrum; Overview is the default
CSV export	Save results, parameters, and metadata, including overview data and high-speed data from the final segment

1.2 First Run

1. Open the existing MotorControl Lab page. The first load requires an internet connection for Plotly and Pyodide/NumPy; wait until the engine is ready.

2. Select Current Control Mode, then Current Ref.

3. Set Mechanical Condition to Locked Rotor; set id=0 A, iq=5 A, Start=0.02 s, and Duration=0.25 s.

4. Select PMSM and set load torque to 0 N·m; leave other defaults unchanged.

5. Run the simulation. Under dq current control, check whether iq tracks 5 A.

6. Select Controller Output / Anti-Windup and compare q-axis P, I, PI raw, and voltages before and after vector limiting.

7. Rerun after changing parameters. The yellow rerun-required indicator means the plots still show the previous results, not results for the new settings.

Every run starts with zero current, zero speed, and zero mechanical angle; it does not continue from the previous simulation. Bode analysis using current settings can update before the time-domain plots, so compare responses from the same parameter snapshot.

2. Modes and Signal Paths

2.1 Current Mode

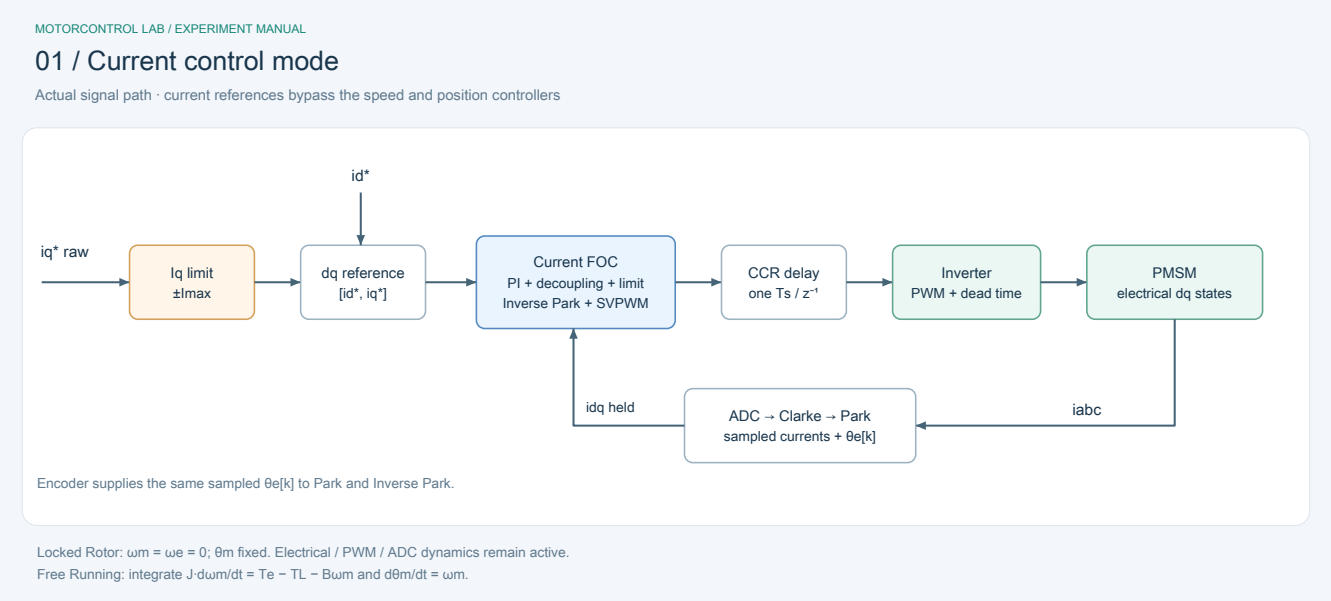

Figure 01 | Current Mode signal path. Blue denotes controllers, orange denotes limits or AW, and green arrows denote feedforward. Solid arrows show signal direction.

The speed and position controllers are bypassed. id directly commands the d axis; iq passes through the q-axis current command limit. The d-axis command applies from the start of the simulation; Start time controls only the iq step.

Mechanical Condition defines the experimental operating condition:

• Locked Rotor (default): ωm=0, ωe=0, and θm=0. Only mechanical speed and angle integration are disabled; dq currents, torque, the inverter, dead time, ADC, and FOC continue to operate. The locking device reaction torque is not plotted separately.

• Free Running: full mechanical dynamics are integrated. A constant positive iq produces positive torque. Even without a load, the rotor may continue accelerating until back-EMF and the finite DC bus prevent the current from being maintained.

Mechanical Condition applies only to Current Mode. Speed and Position always use free mechanical dynamics. Changing this condition requires a new run; it does not lock the rotor during a running simulation.

2.2 Speed Mode

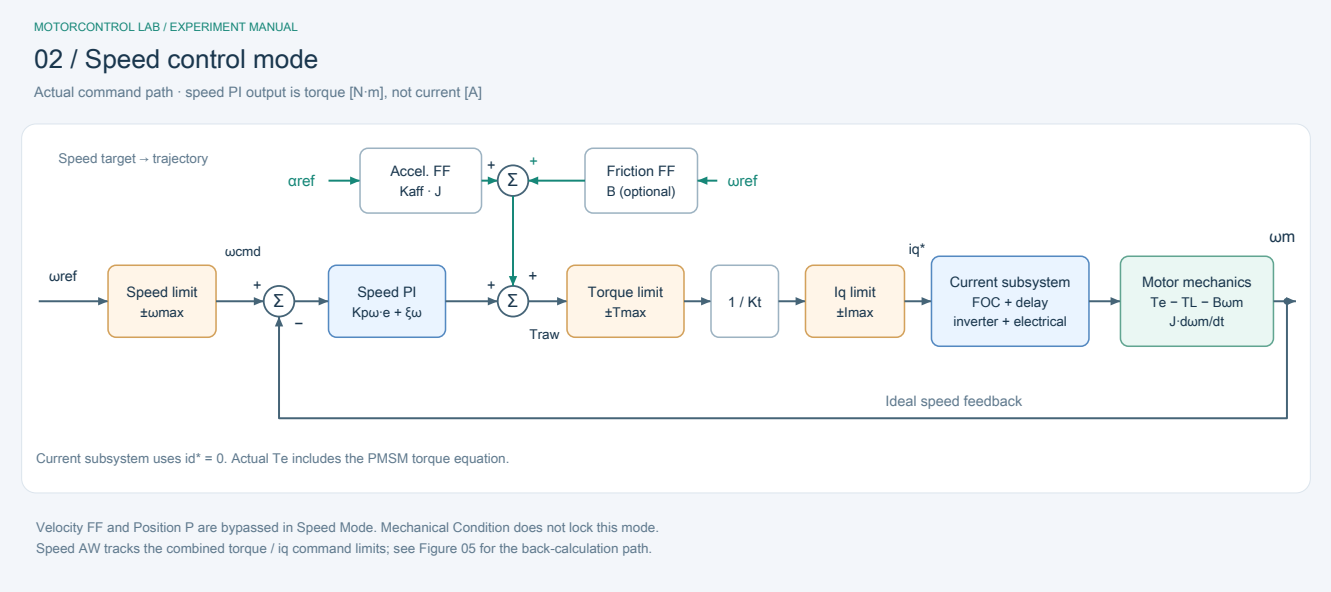

Figure 02 | Speed PI, torque/current limits, and inner current loop. Blue denotes controllers, orange denotes limits or AW, and green arrows denote feedforward. Solid arrows show signal direction.

The speed PI output is in N·m; only after conversion by 1/Kt is the command in A. This mode fixes id*=0 and bypasses position P and Velocity FF. Acceleration FF remains available.

2.3 Position Mode

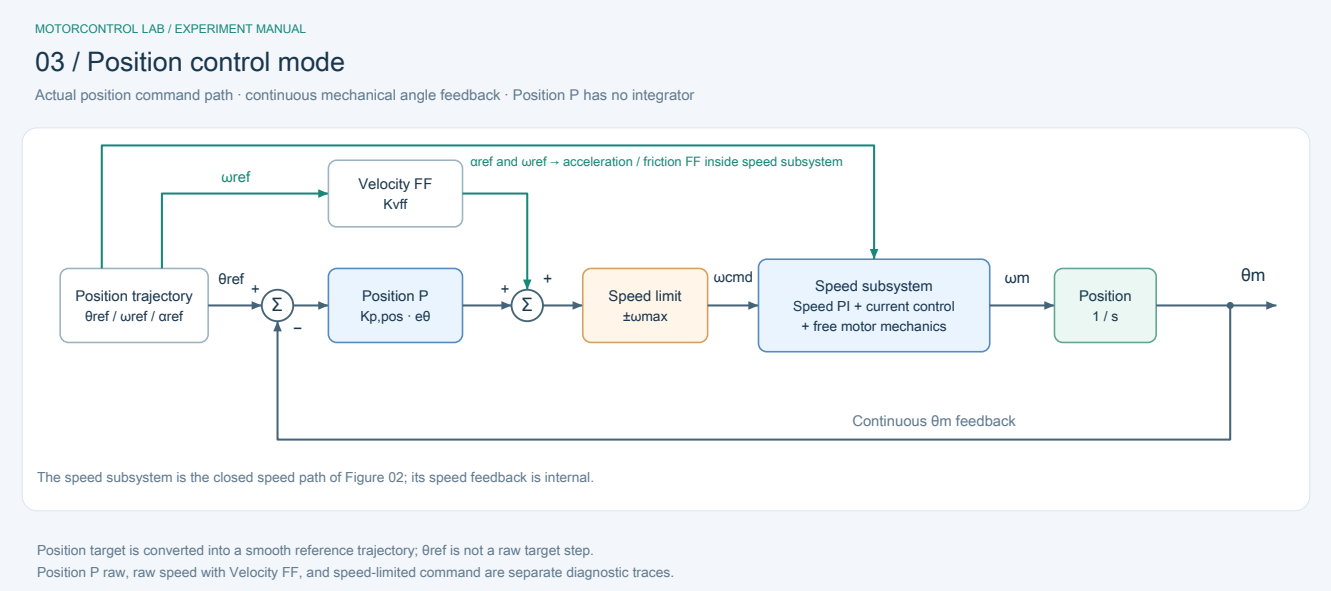

Figure 03 | Position P, trajectory feedforward, and speed subsystem. Blue denotes controllers, orange denotes limits or AW, and green arrows denote feedforward. Solid arrows show signal direction.

Position feedback uses continuous, unwrapped θm in rad, allowing multiple revolutions to be observed. The position controller is P only; there is no position integrator or position AW.

3. Parameters, Units, and Defaults

3.1 Motor and Power Stage

Parameter	Meaning	Default in this version
Rs	Equivalent per-phase resistance	1.2 Ω
Ld / Lq	d/q-axis inductance	0.0057 / 0.0125 H
ψf	Permanent-magnet flux linkage	0.03 Wb
p	Pole pairs; total poles=2p	2
J	Moment of inertia	0.00016 kg·m²
B	Viscous friction coefficient	0.0000028 N·m·s/rad
Vdc	DC bus voltage	48 V
m	Linear modulation limit factor	0.95
Td	Average dead-time model parameter	1 μs, enabled
TL / Load time	Load torque / application time	0.15 N·m / 0.15 s

Here $K_t = 1.5 p \psi_f = 0.09$ N·m/A and the voltage vector limit is approximately 26.33 V. Positive TL opposes positive rotation. It is a constant torque with a specified direction, not a friction model that reverses direction with speed.

3.2 Controllers and Commands

Parameter	Default	Description
Kpd / Kid	5.7 / 1200	V/A, V/(A·s)
Kpq / Kiq	12.5 / 1200	V/A, V/(A·s)
Current Kaw	300 s^{-1}	Voltage vector limit back-calculation
Kpω / Kiω	0.016 / 0.32	N·m/(rad/s), N·m/rad
Speed Kaw	40 s^{-1}	Torque and q-current command limit back-calculation
Kp,pos	15 s^{-1}	Position error in rad to speed command in rad/s
Torque limit / iq limit	1.8 N·m / 12 A	Two independent limits
Speed mode limit	1000 rad/s	Speed Mode command limit
Position speed limit	30 rad/s	Limit on the combined position-controller speed command
Speed target	1500 rpm	Default Speed Step
Position target	1 rad	Trajectory starting from zero
Current iq command / id*	1 A / 0 A	5 A is an experimental setting in this manual, not the factory default
Trajectory max. velocity / acceleration	10 rad/s / 100 rad/s^2	Maximum velocity applies to the position trajectory
Kvff / Kaff	1 / 1	Velocity and acceleration FF enabled by default; friction FF disabled
Start / Duration	0.02 s / 0.25 s	Duration is total simulation time, including the initial wait

Enter Ki as a continuous-time gain; do not multiply it by Ts first. The program applies the control period during integrator updates. Changing Rs, Ld, Lq, J, or other parameters does not automatically retune the PI controllers.

MotorControl Lab | © 2026 Dr. Jack Yeh | All rights reserved.

3.3 Time Bases

Parameter	Default
Requested PWM frequency	10,000 Hz
Timer clock	100 MHz
ARR	5000
Current control period Ts	100 μs
Speed divider M	1
Speed/position update period $T_\omega = M \cdot T_s$	100 μs
Minimum integration steps per PWM period	80
Typical plant integration step	1.25 μs; reduced automatically when needed

Timer ticks, plant integration steps, and control periods are three different time scales. Increasing ARR does not make the plant integrate at every timer tick.

4. PMSM Model and Block Equations

4.1 Electrical, Torque, and Mechanical Models

$$
\begin{aligned}
L_d\dot{i}_d &= v_d - R_s i_d + \omega_e L_q i_q, \\
L_q\dot{i}_q &= v_q - R_s i_q - \omega_e (L_d i_d + \psi_f), \\
T_e &= \frac{3}{2} p \left[\psi_f i_q + (L_d - L_q) i_d i_q\right], \\
J\dot{\omega}_m &= T_e - T_L - B\omega_m, \\
\dot{\theta}_m &= \omega_m, \qquad \omega_e = p\omega_m.
\end{aligned}
$$

During free running, midpoint RK2 integrates four states: id, iq, ωm, and θm. With the rotor locked, ωm=0 and θm is fixed, while the electrical equations are still integrated.

Definitions of the two torque constants: Kt and Kτ (both in N·m/A)

These symbols belong to different blocks; equal units do not make them the same parameter. Kt uses the Latin letter t; Kτ uses the Greek letter tau (τ).

Symbol	Explicit definition	Role in the program and signal path
Kt: nominal permanent-magnet torque constant	Kt = (3/2)pψf	Command conversion: the limited speed PI torque command is converted into a q-axis current command by 1/Kt. It is not the inner current-loop transfer function.
Kτ: incremental torque constant at fixed d-axis operating point	Kτ = ∂Te/∂iq at fixed id=id0	Motor torque production: small-signal Δiq produces ΔTe through Kτ, including reluctance torque effects at that operating point.

$$
K_t = \frac{3}{2} p \psi_f, \qquad i^*_{q,raw} = \frac{T_{lim}}{K_t}.
$$

Deriving Kτ from the PMSM torque equation

Step 1 | Start from the full PMSM electromagnetic torque equation. Retain both permanent-magnet and reluctance torque:

$$
T_e(i_d, i_q) = \frac{3}{2} p \left[\psi_f i_q + (L_d - L_q) i_d i_q\right].
$$

Step 2 | Fix the d-axis current at id=id0. id0 is constant, so factor out iq:

$$
T_e(i_{d0}, i_q) = \frac{3}{2} p \left[\psi_f i_q + (L_d - L_q) i_{d0} i_q\right] = \frac{3}{2} p \left[\psi_f + (L_d - L_q) i_{d0}\right] i_q.
$$

It is the d-axis current id that is fixed; the q-axis current iq is then varied. This does not set id=iq and is not differentiation with respect to time. Fixing id also differs from Locked Rotor, which fixes mechanical speed.

MotorControl Lab | © 2026 Dr. Jack Yeh | All rights reserved.

Step 3 | Differentiate partially with respect to iq while holding id fixed. Everything in brackets is constant with respect to iq, so the derivative is the coefficient of iq:

$$\begin{aligned} K_\tau &\equiv \left.\frac{\partial T_e}{\partial i_q}\right|_{i_d=i_{d0}} \\ &= \frac{3}{2}p\left[\psi_f \frac{\partial i_q}{\partial i_q} + (L_d - L_q) i_{d0} \frac{\partial i_q}{\partial i_q}\right] \\ &= \frac{3}{2}p\left[\psi_f + (L_d - L_q) i_{d0}\right]. \end{aligned}$$

Kτ is therefore the slope describing how many N·m of additional electromagnetic torque result from a 1 A increase in q-axis current while d-axis current remains at id0. Its units are N·m/A. It is an incremental torque constant at the operating point, not a PI gain or current-loop transfer function.

Step 4 | Express the incremental relation used in the block diagram. Let iq0 and Te0 be the operating-point q-axis current and torque, and write iq=iq0+Δiq:

$$T_{e0} = \frac{3}{2}p\left[\psi_f + (L_d - L_q) i_{d0}\right] i_{q0},$$

$$\begin{aligned} \Delta T_e &\equiv T_e(i_{d0}, i_{q0} + \Delta i_q) - T_{e0} \\ &= \frac{3}{2}p\left[\psi_f + (L_d - L_q) i_{d0}\right]\Delta i_q \\ &= K_\tau \Delta i_q \qquad (\Delta i_d = 0). \end{aligned}$$

Thus Δiq → Kτ → ΔTe follows from the partial derivative above. For this manual's fixed-parameter PMSM equations, the incremental relation is exact when id is held strictly constant. Its use in the complete speed Bode model still assumes nominal decoupling and unsaturated small-signal operation.

Here p is the number of pole pairs, ψf is permanent-magnet flux linkage, id0 is the d-axis current at the analysis operating point, and Δ denotes a small change around that point. A fixed id is required for a single Kτ block. If the d axis is also perturbed, the general linearization includes the second term below and cannot be represented by that single path alone:

$$\Delta T_e = K_\tau \Delta i_q + \frac{3}{2}p(L_d - L_q) i_{q0} \Delta i_d.$$

Kτ=Kt when id0=0 or Ld=Lq. With nonzero d-axis current and Ld≠Lq, actual torque includes reluctance torque, so Te cannot always be treated as Kt·iq. The nonlinear time simulation retains the full PMSM torque equation; Kτ is not an extra adjustable motor parameter.

Speed and Position currently use id*=0, and nominal analysis uses id0=0, so the two constants coincide. With default p=2 and ψf=0.03 Wb, Kt=Kτ=0.09 N·m/A. This equality does not remove the current closed loop Tq(s). Section 5.4 shows the complete speed-loop torque path.

4.2 Clarke, Park, and Sensors

The Clarke transform is amplitude-invariant:

$$i_\alpha = \frac{2i_a - i_b - i_c}{3}, \qquad i_\beta = \frac{i_b - i_c}{\sqrt{3}}.$$

Park and inverse Park transforms:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}, \qquad \begin{bmatrix} v_\alpha^* \\ v_\beta^* \end{bmatrix} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e \\ \sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} v_{d,sat} \\ v_{q,sat} \end{bmatrix}.$$

The encoder is ideal: θe=(pθm) mod 2π, with no quantization, noise, or observer. Park and inverse Park use the angle latched at the same ADC event. Coordinate transforms are matrix operations, normally not a single scalar transfer function; with a fixed angle they can be treated as constant matrices.

4.3 Current PI, Decoupling, and Vector Limiting

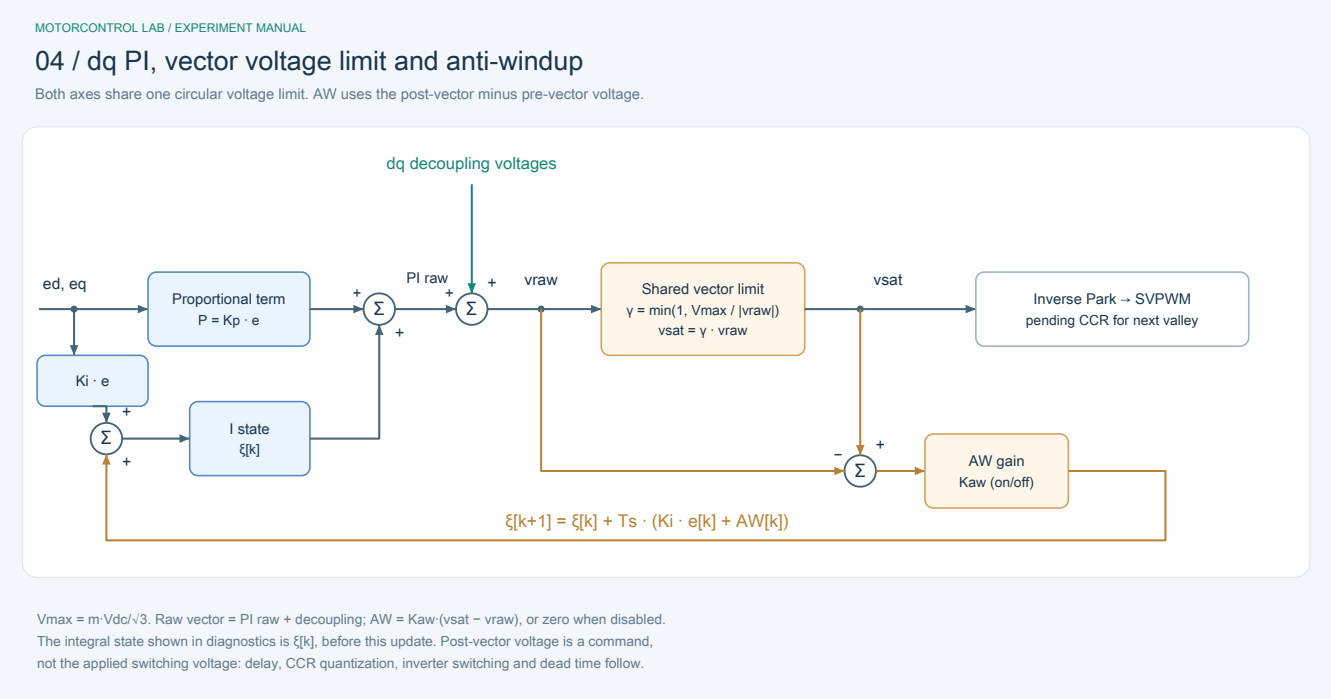

Figure 04 | dq PI, shared voltage vector limit, and AW back-calculation. Blue denotes controllers, orange denotes limits or AW, and green arrows denote feedforward. Solid arrows show signal direction.

For a∈{d,q}:

$$e_a = i_a^* - i_{a,held}, \qquad u_{a,PI} = K_{pa} e_a + x_a.$$

With decoupling enabled:

$$v_{d,ff} = -\omega_e L_q i_{q,held}, \qquad v_{q,ff} = \omega_e (L_d i_{d,held} + \psi_f).$$

With decoupling disabled, both feedforward terms are zero. Voltage before vector limiting includes the decoupling terms:

$$\boldsymbol{v}_{raw} = \begin{bmatrix} u_{d,PI} + v_{d,ff} \\ u_{q,PI} + v_{q,ff} \end{bmatrix}, \quad V_{max} = m\frac{V_{dc}}{\sqrt{3}}, \quad \gamma = \min\left(1, \frac{V_{max}}{\|\boldsymbol{v}_{raw}\|}\right), \quad \boldsymbol{v}_{sat} = \gamma \boldsymbol{v}_{raw}.$$

For a zero vector, γ=1. This is a shared circular vector limit, not independent d/q clipping at ±Vmax.

$$AW_a = \begin{cases} K_{aw}(v_{a,sat} - v_{a,raw}), & AW\ on \\ 0, & AW\ off \end{cases}$$

$$x_a[k+1] = x_a[k] + T_s\{K_{ia}e_a[k] + AW_a[k]\}.$$

The output is calculated with x[k] before updating x[k+1]. During positive saturation, AW is usually negative and opposes further integrator growth. However, the I state need not decrease or return to zero; its change depends on the combined Ki·e and AW terms.

4.4 Speed PI, Limits, and Back-Calculation

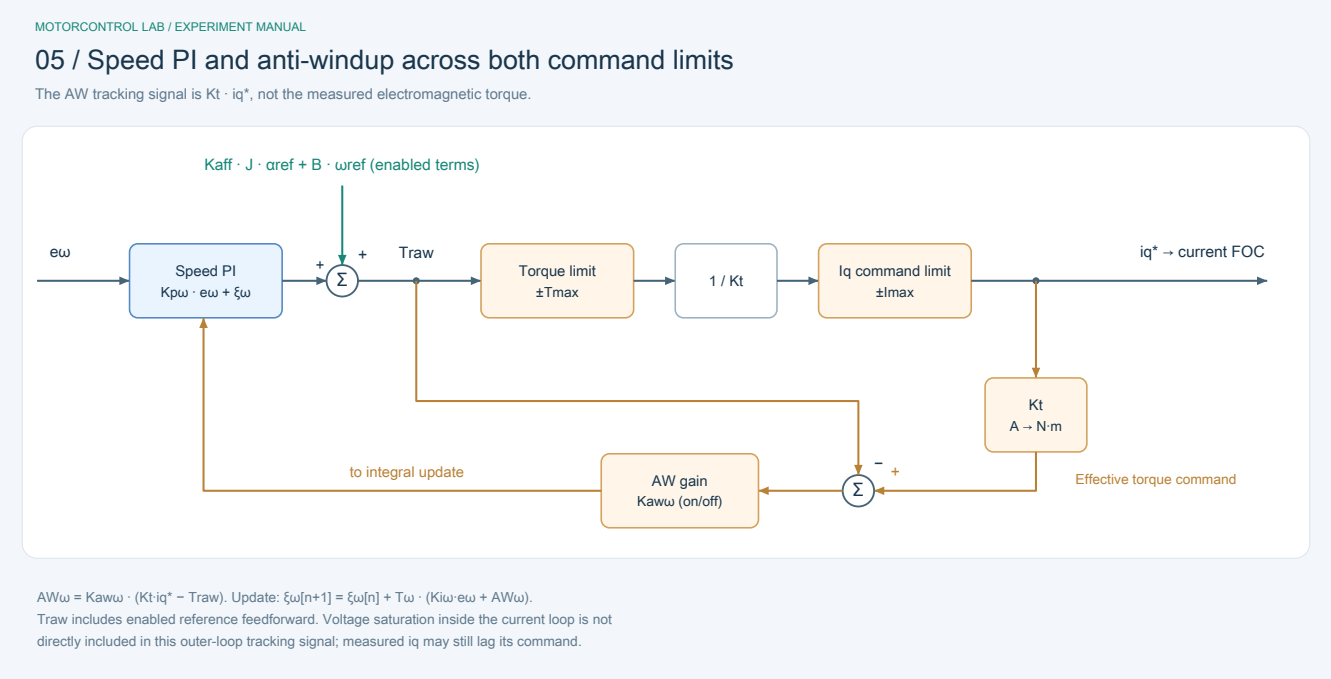

Figure 05 | Speed PI AW across torque and current command limits. Blue denotes controllers, orange denotes limits or AW, and green arrows denote feedforward. Solid arrows show signal direction.

$$e_\omega = \omega_{cmd} - \omega_m, \quad T_{fb} = K_{p\omega}e_\omega + x_\omega.$$

Let the switches σa, σB∈{0,1}:

$$T_{raw} = T_{fb} + \sigma_a K_{aff} J\alpha_{ref} + \sigma_B B\omega_{ref},$$

$$T_{lim} = \mathrm{clip}(T_{raw}, \pm T_{max}), \quad i^*_{q,raw} = \frac{T_{lim}}{K_t}, \quad i^*_q = \mathrm{clip}(i^*_{q,raw}, \pm I_{max}).$$

$$AW_\omega = \begin{cases} K_{aw\omega}(K_t i^*_q - T_{raw}), & AW\ on \\ 0, & AW\ off \end{cases}$$

$$x_\omega[n+1] = x_\omega[n] + T_\omega(K_{i\omega}e_\omega[n] + AW_\omega[n]).$$

Back-calculation includes both torque and current command limits and uses Total raw, including feedforward, as its reference. It does not directly track actual iq lag caused by insufficient voltage. Therefore, speed AW can be zero when Te* does not hit the torque/current limits, even if the inner current loop is voltage-saturated.

4.5 Position P, Trajectories, and Feedforward

$$\omega_{fb} = K_{p,pos}(\theta_{ref} - \theta_m), \quad \omega_{raw} = \omega_{fb} + \sigma_v K_{vff}\omega_{ref}, \quad \omega_{cmd} = \mathrm{clip}(\omega_{raw}, \pm\omega_{max}).$$

4.5.1 Distinguish Position, Velocity, and Acceleration Curves

In Position Mode, the Trajectory Generator produces a piecewise position trajectory, not one parabola with the same coefficients throughout. Position is quadratic during acceleration, linear during cruise, and a different quadratic during deceleration; it is constant before and after motion. Velocity is piecewise linear and acceleration is piecewise constant. This version has no S-curve or jerk limiting; acceleration can jump at segment boundaries.

Each run starts at position θ=0 and velocity=0, then stops at the specified target θf. Arbitrary initial position, initial velocity, and direct polynomial coefficient entry are not provided. Define:

$$\theta_f = \text{Target Position}, \qquad D = |\theta_f|, \qquad \sigma = \mathrm{sgn}(\theta_f), \qquad u = t - t_0.$$

t is simulation time; t0 is UI Start / Reference step time; u is local time measured from trajectory start. V is Max Velocity (position), and A is Max Acceleration. Enter positive magnitudes for both limits; the sign of θf determines motion direction.

$$t_a = \min\left(\frac{V}{A}, \sqrt{\frac{D}{A}}\right), \qquad v_p = A t_a, \qquad t_c = \max\left(0, \frac{D}{v_p} - t_a\right), \qquad T = T_{move} = 2t_a + t_c.$$

ta is acceleration time, also used for symmetric deceleration; tc is cruise time; vp is planned peak velocity and may be less than V. For D≤V²/A, tc=0 and the velocity profile is triangular. For D>V²/A, a cruise segment produces a trapezoidal velocity profile. D=0 is a stationary special case and does not use the division above. Planned completion is t0+T; simulation Duration should also allow time for actual position settling.

4.5.2 Piecewise Equations Used by the Program

For D>0, the position reference below matches the current Python trajectory function:

$$\theta_{ref}(t) = \begin{cases} 0, & u < 0, \\ \sigma\frac{A}{2}u^2, & 0 \le u < t_a, \\ \sigma\left[\frac{A}{2}t_a^2 + v_p(u - t_a)\right], & t_a \le u < t_a + t_c, \\ \theta_f - \sigma\frac{A}{2}(T - u)^2, & t_a + t_c \le u < T, \\ \theta_f, & u \ge T. \end{cases}$$

The acceleration and deceleration segments above are quadratic. When tc=0, the cruise interval has zero length and is omitted. There are still two different quadratic segments, not one quadratic curve.

Velocity and acceleration references are analytical derivatives of the same position trajectory and are used for feedforward:

$$\omega_{ref}(t) = \begin{cases} 0, & u < 0, \\ \sigma A u, & 0 \le u < t_a, \\ \sigma v_p, & t_a \le u < t_a + t_c, \\ \sigma A (T - u), & t_a + t_c \le u < T, \\ 0, & u \ge T, \end{cases}$$

$$\alpha_{ref}(t) = \begin{cases} 0, & u < 0, \\ \sigma A, & 0 \le u < t_a, \\ 0, & t_a \le u < t_a + t_c, \\ -\sigma A, & t_a + t_c \le u < T, \\ 0, & u \ge T. \end{cases}$$

Position and velocity are continuous at segment boundaries. Acceleration takes the value of the segment on the right, matching the program's interval tests. These are reference trajectories; actual θm, ωm, and dωm/dt need not track them perfectly.

4.5.3 How UI Settings Determine Quadratic Coefficients

Write each segment as a polynomial in the same local time u:

$$\theta_{ref} = c_2 u^2 + c_1 u + c_0,$$

The coefficients are not independent inputs. They follow from ta, tc, and T computed from A, V, and θf:

Segment	c2 (rad/s²)	c1 (rad/s)	c0 (rad)
Acceleration	σA/2	0	0
Cruise	0	σvp	−σA·ta²/2
Deceleration	−σA/2	σA·T	θf−σA·T²/2
Hold after completion	0	0	θf

UI Max Acceleration=A is acceleration, not the quadratic coefficient. Differentiating position twice gives 2c2, so c2=σA/2 during acceleration. For example, A=100 gives a positive-motion quadratic coefficient of 50, not 100. These coefficients use u=t−t0; do not replace u with global time t while keeping the coefficients unchanged.

UI parameter	Effect on the curve	How to set it
Target Position / θm* continuous	Sets displacement, direction, and final position θf	Target position in Position Control or Target Position in Trajectory Generator; both share one parameter
Max Acceleration	Sets A and the curvature ±A/2 of acceleration/deceleration segments	Edit under Trajectory Generator; a larger value plans faster acceleration/deceleration
Max Velocity (position)	Sets velocity limit V; affects peak velocity, cruise availability, and total time	Edit under Trajectory Generator; short moves may not reach V

UI parameter	Effect on the curve	How to set it
Start / Reference step time	Sets t0, shifting the entire trajectory in time	Edit under Target Position; coefficients expressed in u are unchanged
Duration	Sets total simulation length, not polynomial coefficients	Cover Start+Tmove and allow settling time; a shorter duration truncates observation

Position Kp, Kvff, and Kaff affect feedback/feedforward tracking, not the trajectory polynomial coefficients. Speed profile Step/Ramp applies only in Speed Mode and has no effect in Position Mode.

4.5.4 Experiment 5: Quadratic Equations and Coefficients

Experiment 5 uses θf=1 rad, V=10 rad/s, A=100 rad/s², and t0=0.02 s. Thus ta=0.1 s, tc=0, T=0.2 s, and local time u=t−0.02:

$$\theta_{ref}(t) = \begin{cases} 0, & t < 0.02, \\ 50(t-0.02)^2, & 0.02 \le t < 0.12, \\ 1 - 50(0.22-t)^2, & 0.12 \le t < 0.22, \\ 1, & t \ge 0.22. \end{cases}$$

Equivalently, expanding in u:

$$\theta_{ref} = \begin{cases} 50u^2, & 0 \le u < 0.1, \\ -50u^2 + 20u - 1, & 0.1 \le u < 0.2. \end{cases}$$

The acceleration coefficients are (c2,c1,c0)=(50,0,0); the deceleration coefficients are (−50,20,−1). Both segments use u=t−0.02 but have different coefficients. Users do not enter these three coefficients separately in the UI.

Simulation time t (s)	Local time u (s)	Position reference θref (rad)	Velocity reference ωref (rad/s)
0.02	0	0	0
0.07	0.05	0.125	5
0.12	0.10	0.5	10
0.17	0.15	0.875	5
0.22	0.20	1	0

Change Max Acceleration to adjust quadratic curvature, Target Position to adjust displacement or direction, and Max Velocity to adjust the speed limit. These settings jointly satisfy the piecewise boundary conditions of starting and ending at rest; c2, c1, and c0 cannot be independently assigned.

4.5.5 Speed Mode Step/Ramp

Speed Step changes the velocity reference immediately at the start time. The program sets αref=0 and does not generate an ideal derivative-impulse feedforward. Speed Ramp limits acceleration to A until the target speed is reached. The trajectory generator and limits are nonlinear/piecewise operations, with no single LTI transfer function covering all behavior.

4.6 SVPWM, Timer, and Inverter

$$v_a^* = v_\alpha^*, \quad v_b^* = -\frac{1}{2}v_\alpha^* + \frac{\sqrt{3}}{2}v_\beta^*, \quad v_c^* = -\frac{1}{2}v_\alpha^* - \frac{\sqrt{3}}{2}v_\beta^*,$$

$$v_0 = -\frac{\max(v_a^*, v_b^*, v_c^*) + \min(v_a^*, v_b^*, v_c^*)}{2}, \quad D_x = \text{clip}\left(\frac{1}{2} + \frac{v_x^* + v_0}{V_{dc}}, 0, 1\right).$$

$$ARR = \text{round}\left(\frac{f_{timer}}{2f_{requested}}\right), \quad f_{PWM,actual} = \frac{f_{timer}}{2ARR}, \quad CCR_x = \text{round}(D_x ARR).$$

CNT is a triangular carrier, 0→ARR→0. Sx=1 when CNT<active CCRx, and 0 otherwise.

$$v_{xO} = S_x V_{dc} + \Delta v_{xO}, \quad \Delta v_{xO} = -V_{dc} T_d f_{PWM} \,\text{sgn}(i_x), \quad v_{xn} = v_{xO} - \frac{v_{aO} + v_{bO} + v_{cO}}{3}.$$

The model combines switching waveforms with a period-average dead-time voltage disturbance. It does not individually model complementary gate blanking, diode drops, or parasitic components. In the unsaturated average small-signal model, voltage command to average voltage can have unity gain, with hold/delay treated separately. Do not multiply by Vdc again: the input is already in V.

5. Control Loop Transfer Functions

5.1 Assumptions and Notation

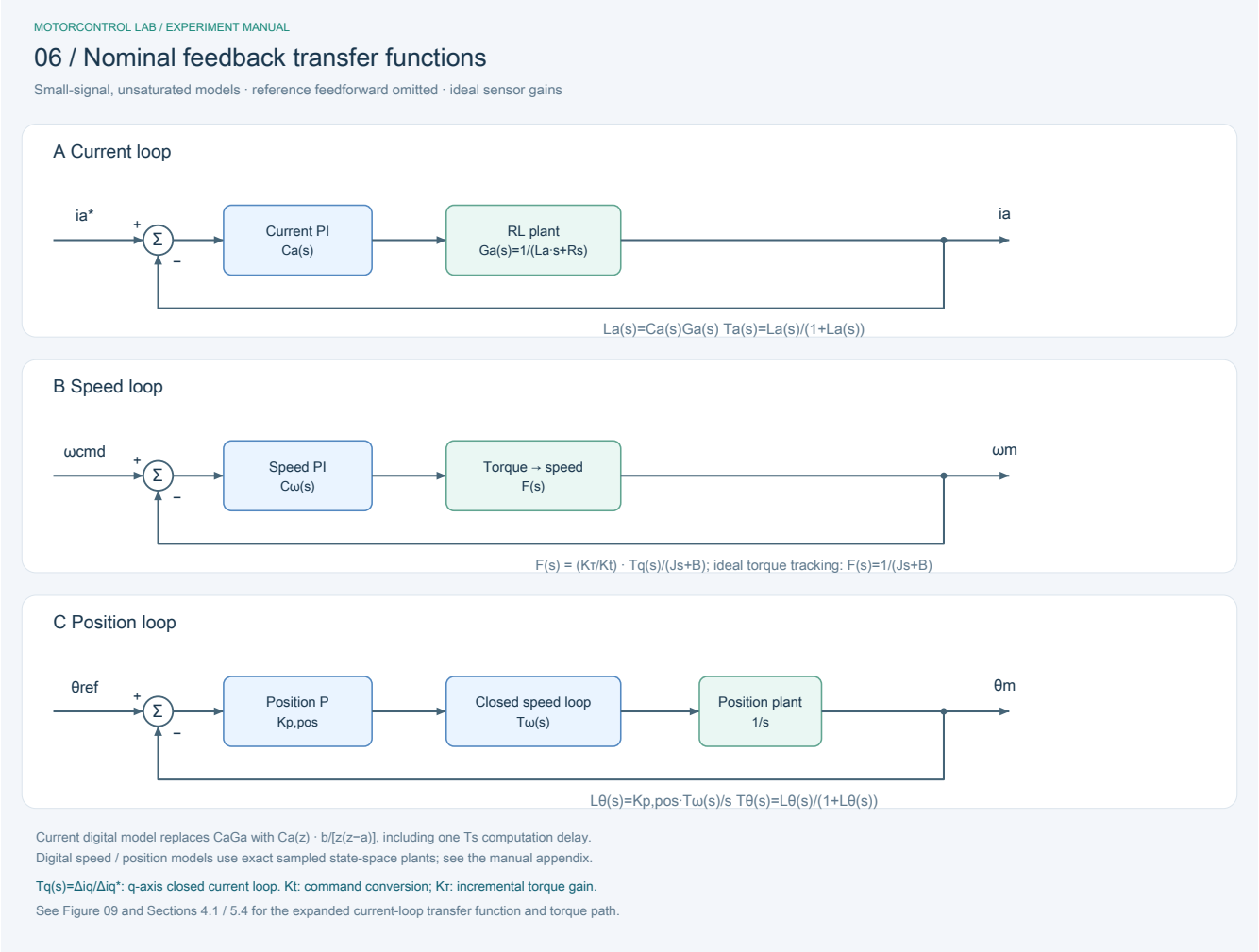

Figure 06 | Nominal negative-feedback transfer functions for current, speed, and position. Blue denotes controllers, orange denotes limits or AW, and green arrows denote feedforward. Solid arrows show signal direction.

The continuous models below assume ideal dq decoupling, fixed nominal parameters, and no limiting; average voltage replaces switching ripple. ADC/computation delay and dead-time nonlinearities are initially excluded. s is the Laplace variable, z the discrete-time variable. Nominal feedback sensor gain H=1.

5.2 Block Mathematics at a Glance

UI block	Nominal linear model / actual operation	Scope
Current PI d/q	$C_a(s)=K_{pa}+K_{ia}/s$; digital $C_a(z)=K_{pa}+K_{ia}\cdot T_s/(z-1)$	See Section 4 for AW and limiting
PMSM electrical	$G_a(s)=1/(L_a\cdot s+R_s)$	Ideal decoupling or zero speed; not the full coupled PMSM
Speed PI	$C_\omega(s)=K_{p\omega}+K_{i\omega}/s$	Input rad/s, output N·m
1/Kt	Constant gain $1/(1.5p\psi_f)$	Input N·m, output A
Torque production	Local q-axis gain K_τ	Fixed d-axis operating point; general Te is nonlinear
Mechanical speed	$G_m(s)=1/(Js+B)$	Input electromagnetic torque minus load; the free mechanical plant is not used with a locked rotor
Speed → position	$1/s$	Continuous free-rotation model; discrete analysis integrates position exactly
Position P	$C_\theta=K_{p,pos}$	No position I or AW
Velocity FF	K_{vff}	ω_{ref}→speed summing point; Position only
Acceleration FF × J	$K_{aff}\cdot J$	Constant gain for input α_{ref}; $K_{aff}\cdot J\cdot s$ only when the input is ω_{ref}
Friction FF	B	ω_{ref}→torque summing point, switch-controlled
Encoder speed/position feedback	Nominal H=1; electrical angle is $p\theta_m$ followed by wrap	Ideal sensing; wrap is not one global LTI block
Clarke / Park / Inverse Park	Matrices in Section 4.2	Time-varying angle prevents a single scalar TF description
ADC sample/hold	Discrete synchronous sampling and hold	Included in the digital loop definition, not an arbitrary extra delay
Computation / CCR update	z^{-1}	One current Ts
SVPWM / inverter	Average unsaturated voltage command channel has approximately unity gain	Actual duty calculation, quantization, PWM switching, and dead-time disturbance
Trajectory / limits / AW	Piecewise trajectories, clipping, vector scaling, and state updates	Nonlinear operations; no global LTI TF is imposed

5.3 d/q Current Loops

Decoupled plant and PI:

$$G_d(s) = \frac{1}{L_d s + R_s}, \quad G_q(s) = \frac{1}{L_q s + R_s}, \quad C_a(s) = K_{pa} + \frac{K_{ia}}{s}.$$

$$L_a(s) = C_a(s) G_a(s) = \frac{K_{pa} s + K_{ia}}{s(L_a s + R_s)},$$

$$T_a(s) = \frac{i_a}{i_a^*} = \frac{L_a}{1 + L_a} = \frac{K_{pa} s + K_{ia}}{L_a s^2 + (R_s + K_{pa}) s + K_{ia}}.$$

If Kpa=La·ωc and Kia=Rs·ωc, the ideal continuous model cancels the RL pole:

$$L_a(s) = \frac{\omega_c}{s}, \qquad T_a(s) = \frac{\omega_c}{s + \omega_c}.$$

Both axes default to ωc=1000 rad/s, giving ideal bandwidth 1000/(2π)≈159.15 Hz. The d/q current-loop frequency responses in the web Bode panel use a discrete model with delay and need not equal this ideal value. At locked rotor, ωe=0 eliminates rotational coupling terms even with decoupling disabled.

Setting a=q above gives Tq(s). Section 5.4 explicitly restates its input, output, and expanded expression to avoid confusing Tq(s) with a torque constant in the speed loop.

5.4 Speed Loop: Three Models

Start with the three UI Speed model options below. Speed model selects a model for Bode analysis; it differs from the Current/Speed/Position Control Mode at the top of the page.

Speed model option	Inner current-loop treatment	Digital delay included?
Ideal Inner Current Loop	Instantaneous current/torque tracking, Tq=1; no inner current-loop dynamics	No sampling/computation delay
With Current Loop Dynamics	Continuous q-axis current closed loop; generally second order before reduction, first order after default pole-zero cancellation	No sampling/computation delay
With Digital Delay	Current-loop dynamics retained, with discrete PI and an exact ZOH plant	One current control period of computation/CCR update delay, plus the included sampling/hold effects

The third option includes both current-loop dynamics and delay; it does not replace the dynamics with delay alone. It is calculated from the discrete system, not by multiplying the second option's continuous closed loop by exp(−1.5sTs). The 1.5Ts value is a continuous approximation for low-frequency average delay (Section 5.4.2); do not add it again to the digital model.

Scope of the selector: Speed model changes the speed/position Bode analysis model, not the time simulation engine. d/q current Bode always uses the delayed discrete model in Section 6; selecting Ideal does not turn these responses into unity gain. Time simulation continues to use the same actual current control, ADC/CCR timing, and PWM switching model.

The formulas for the three speed analysis models follow. Default first-order reduction is in Section 5.4.1; the exact digital model is in Section 6 and the appendix.

$$C_\omega(s) = K_{p\omega} + \frac{K_{i\omega}}{s}, \qquad G_m(s) = \frac{1}{Js + B}.$$

Ideal inner current loop: assume instantaneous torque tracking and set F(s)=Gm(s):

$$L_\omega^{ideal}(s) = \frac{K_{p\omega}s + K_{i\omega}}{s(Js + B)},$$

$$T_\omega^{ideal}(s) = \frac{K_{p\omega}s + K_{i\omega}}{Js^2 + (B + K_{p\omega})s + K_{i\omega}}.$$

Substituting default values:

$$T_\omega^{ideal}(s) = \frac{0.016s + 0.32}{0.00016s^2 + 0.0160028s + 0.32}.$$

With ideal torque tracking, the load-disturbance response is:

$$\frac{\omega_m(s)}{T_L(s)} = -\frac{G_m(s)}{1 + C_\omega(s)G_m(s)}.$$

With current-loop dynamics: explicitly includes the q-axis current closed loop Tq(s)

Tq(s) is the closed-loop transfer function from q-axis current command to actual q-axis current. It is neither a torque constant nor the speed closed loop. Subscript q denotes the q axis; both input and output are currents, so Tq(s) is dimensionless. Transfer functions below use incremental quantities around an operating point:

$$\boxed{T_q(s) \equiv \frac{\Delta i_q(s)}{\Delta i_q^*(s)}}.$$

Nominal decoupled q-axis electrical plant, current PI, and open loop:

$$G_q(s) = \frac{\Delta i_q(s)}{\Delta v_q(s)} = \frac{1}{L_q s + R_s}, \qquad C_q(s) = K_{pq} + \frac{K_{iq}}{s}, \qquad L_q^{OL}(s) = C_q(s)G_q(s).$$

Superscript OL denotes open loop, avoiding confusion with inductance Lq. Closing the current loop with unity negative feedback gives:

$$\boxed{T_q(s) = \frac{C_q(s)G_q(s)}{1 + C_q(s)G_q(s)} = \frac{C_q(s)}{L_q s + R_s + C_q(s)}}.$$

Expanding the current PI makes Kpq, Kiq, Lq, and Rs explicit:

$$\boxed{T_q(s) = \frac{K_{pq}s + K_{iq}}{L_q s^2 + (R_s + K_{pq})s + K_{iq}}}.$$

With default Lq=0.0125 H, Rs=1.2 Ω, Kpq=12.5, and Kiq=1200, the continuous q-axis closed loop is:

$$T_q(s) = \frac{12.5s + 1200}{0.0125s^2 + 13.7s + 1200}.$$

5.4.1 Generally Second Order; First Order After Default Pole-Zero Cancellation

The quadratic denominator above is the unreduced closed-loop form for general PI settings. The default parameters cancel the nominal RL pole with the PI zero, so the minimum order of this continuous input/output transfer function is one. The second-order expression alone does not describe this default simplification.

Cancellation condition and default values:

$$\frac{K_{iq}}{K_{pq}} = \frac{R_s}{L_q}, \qquad \frac{1200}{12.5} = \frac{1.2}{0.0125} = 96\ \mathrm{s}^{-1}.$$

The PI zero and electrical plant pole are both at s=−96 s⁻¹. Factoring the general expression:

$$\begin{aligned} T_q(s) &= \frac{12.5s + 1200}{0.0125s^2 + 13.7s + 1200} \\ &= \frac{12.5(s + 96)}{0.0125(s + 96)(s + 1000)} \\ &= \boxed{\frac{1000}{s + 1000} = \frac{1}{0.001s + 1}}. \end{aligned}$$

More generally, under exact cancellation:

$$C_q(s)G_q(s) = \frac{K_{pq}}{L_q s} = \frac{\omega_c}{s}, \qquad T_q(s) = \frac{\omega_c}{s + \omega_c}, \qquad \omega_c = \frac{K_{pq}}{L_q}.$$

Thus the defaults give ωc=1000 rad/s, time constant 1/ωc=1 ms, and continuous no-delay −3 dB bandwidth approximately 1000/(2π)=159.15 Hz. This is transfer-function cancellation under zero initial conditions, not removal of simulator integrators or motor states. If PI, Rs, or Lq changes without maintaining cancellation, return to the general second-order expression. Parameter errors, incomplete decoupling, digital sampling, and delay also limit the ideal result.

This first-order Tq(s) remains the no-delay continuous model used by With current-loop dynamics. The next subsection explains why adding delay inside the feedback loop does not leave the complete closed loop equal to this no-delay first-order expression.

Separate the four blocks 1/Kt, Tq(s), Kτ, and Gm(s) before combining them into F(s), the plant seen by the speed PI. First, 1/Kt converts torque command to current command; Tq(s) produces actual iq; Kτ converts current to electromagnetic torque; Gm(s) produces speed. This nominal feedback path excludes limiting, feedforward, and the load-disturbance input.

$$\Delta T_e^* \xrightarrow{1/K_t} \Delta i_q^* \xrightarrow{T_q(s)} \Delta i_q \xrightarrow{K_\tau} \Delta T_e \xrightarrow{G_m(s)} \Delta \omega_m.$$

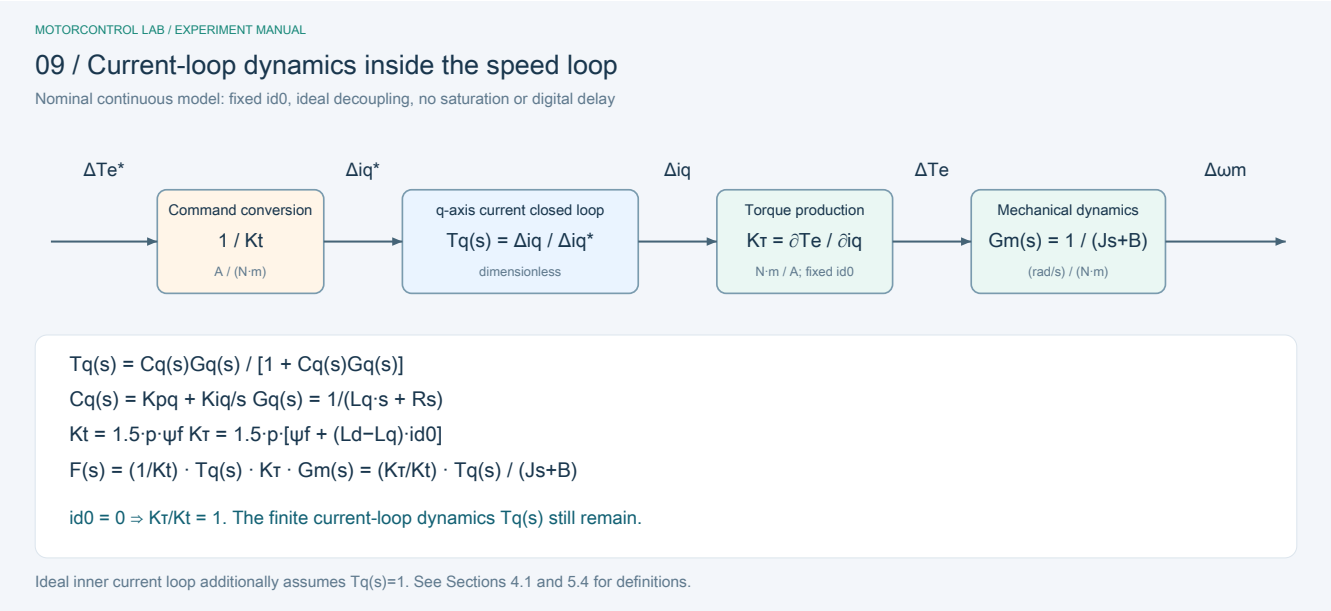

Figure 09 | Tq(s) is command-to-actual-current closed-loop response; 1/Kt converts the command, while Kτ is the motor incremental torque gain at a fixed d-axis operating point.

$$F(s) \equiv \frac{\Delta\omega_m(s)}{\Delta T_e^*(s)} = \frac{1}{K_t}\, T_q(s)\, K_\tau\, G_m(s) = \frac{K_\tau}{K_t} T_q(s) \frac{1}{Js + B}.$$

With outer unity negative speed feedback:

$$L_\omega(s) = C_\omega(s) F(s), \qquad T_\omega(s) = \frac{\Delta\omega_m(s)}{\Delta\omega_m^*(s)} = \frac{C_\omega(s) F(s)}{1 + C_\omega(s) F(s)}.$$

Section 4.1 defines Kt, Kτ, and their linearization conditions. The nominal Speed/Position operating point is id0=0, hence Kτ/Kt=1, but:

$$F(s) = T_q(s) G_m(s) \neq G_m(s) \quad (\text{in general}).$$

Only the ratio Kτ/Kt cancels to one; Tq(s) does not disappear. Tq(s)=1 only for Ideal inner current loop, which assumes instantaneous current/torque tracking. With current-loop dynamics uses the continuous current closed loop above, ignoring sampling/computation delay. It assumes nominal ideal dq decoupling, no saturation, and a fixed d-axis operating point; it excludes dead time, CCR quantization, limiting, and AW nonlinearities. Time simulation retains the complete switching/control model.

With digital delay: includes discrete current PI, one current control period of computation delay, an exact held-voltage plant, and multirate outer-loop hold. The continuous second-order expression above cannot be used directly. See Section 6 and the appendix for the full definition.

5.4.2 Continuous Delay Approximation: Why 1.5Ts, Not Another Full Period

This is an educational continuous average/low-frequency approximation, not a new UI model or a change to the exact With digital delay calculation. Here Ts is the current-loop period. Sampling and updating occur once per PWM period, so Ts=TPWM. Td,eq denotes total equivalent control delay; it differs from power-stage Dead Time.

Program timing is: at valley k, apply the previous pending CCR, then latch ADC[k]. New duty/CCR[k+1] computed from that sample becomes active at valley k+1. Thus ADC[k] to new CCR activation takes one Ts, covering computation, transfer, and waiting for the next update point. It is not a measurement that the CPU takes a full Ts to compute. Browser Simulation Cost is not this control delay either.

Contribution	Meaning in this version	Continuous average/low-frequency approximation
Sampling to new CCR activation	ADC[k] → pending CCR[k+1] → application at next valley	Ts
Equivalent PWM/command hold delay	New duty produces an average effect over one PWM period; approximated by ZOH/single-rate PWM	Ts/2
Total	Combined approximation of two distinct effects	Td,eq≈1.5Ts

This does not mean “compute for Ts, then wait another full PWM period.” Waiting for the next valley is already included in the first row's Ts. The second row represents an equivalent half-period hold/modulation effect, not another one-sample register. A real second-sample delay would need separate identification and would invalidate this version's 1.5Ts approximation.

A continuous reconstruction model from average command to average output combines a one-sample delay with a normalized ZOH:

$$D_{avg}(s) = e^{-sT_s}\frac{1-e^{-sT_s}}{sT_s}.$$

ZOH frequency response separates into a magnitude factor and a half-period phase factor:

$$\frac{1-e^{-j\omega T_s}}{j\omega T_s} = e^{-j\omega T_s/2}\frac{\sin(\omega T_s/2)}{\omega T_s/2}.$$

At sufficiently small ωTs, sinc magnitude attenuation is approximately negligible, giving the common approximation:

$$\boxed{D_{avg}(s) \approx e^{-sT_{d,eq}}, \qquad T_{d,eq} \approx T_s + \frac{T_s}{2} = 1.5T_s}.$$

Not every synchronous sampling system necessarily has 1.5Ts delay: sampling phase, update point, whether computation meets the update deadline, and single/double-rate operation all matter. This version samples at a valley and applies the result at the next valley, supporting the one-period-plus-half-period average approximation. No extra sensor, analog-filter, or ADC conversion delay is added.

Default timing in this version: fPWM=10 kHz, ARR=5000, timer=100 MHz, hence:

$$T_s = \frac{2ARR}{f_{timer}} = 100\ \mu\text{s}, \qquad T_{d,eq} \approx 150\ \mu\text{s} = 0.15\ \text{ms}.$$

MotorControl Lab | © 2026 Dr. Jack Yeh | All rights reserved.

This is 1.5 times Ts, not 1.5 ms. Only Ts=1 ms makes 1.5Ts equal to 1.5 ms. The default first-order current closed-loop time constant of 1 ms also differs from this equivalent delay of 0.15 ms. Recompute using actual Ts when fpwm/ARR changes. With speedDiv≠1, the multirate outer loop cannot simply be assigned a delay of 1.5 speed periods.

Where does delay belong? Inside the current feedback loop, in the actuation channel. Under the continuous approximation above:

$$L_{q,d}(s) \approx C_q(s)G_q(s)e^{-sT_{d,eq}}, \qquad T_{q,d}(s) \approx \frac{C_q(s)G_q(s)e^{-sT_{d,eq}}}{1 + C_q(s)G_q(s)e^{-sT_{d,eq}}}.$$

After default PI/RL pole-zero cancellation, CqGq=ωc/s, so:

$$\boxed{T_{q,d}(s) \approx \frac{\omega_c e^{-sT_{d,eq}}}{s + \omega_c e^{-sT_{d,eq}}}}.$$

Delay appears in the denominator too; the result is no longer a simple no-delay first-order rational function. It differs from adding an output delay after closing a no-delay loop, as shown below: the feedback characteristic equations are different.

$$\frac{\omega_c e^{-sT_{d,eq}}}{s + \omega_c e^{-sT_{d,eq}}} \neq \frac{\omega_c}{s + \omega_c} e^{-sT_{d,eq}}.$$

For example, at ω=1000 rad/s and Td,eq=150 μs, this delay factor adds approximately −0.15 rad=−8.59° phase and reduces phase margin. A no-delay first-order current closed loop cannot show this feedback-stability effect.

Mapping to the UI: With current-loop dynamics retains continuous, no-delay Tq(s). With digital delay uses discrete PI, one-sample delay, and the exact ZOH plant in Section 6, with state-space multirate treatment for speed/position. These effects are already included: do not multiply by exp(−1.5sTs) again or add another half-period, which would double-count delay. The 1.5Ts approximation explains low-frequency average phase behavior; it does not replace the exact digital model.

Reference: imperix: sampling, control update, and PWM modulation delays. Timing in this version was checked against the CCR queue in simulation.py and digital Bode implementation in ui-2.js.

5.5 Position Loop

First close the speed feedback loop, with closed-loop response Tω(s). Outer position P is Cθ=Kp,pos, and speed-to-position is 1/s:

$$L_\theta(s) = K_{p,pos}\frac{T_\omega(s)}{s}, \qquad T_\theta(s) = \frac{\theta_m}{\theta_{ref}} = \frac{K_{p,pos}T_\omega(s)}{s + K_{p,pos}T_\omega(s)}.$$

If the speed closed loop is approximately one, Tθ≈Kp,pos/(s+Kp,pos), giving default ideal position bandwidth about 15/(2π)=2.39 Hz. This assumes sufficiently fast inner loops and no speed limiting.

With ideal current/torque tracking but retained Speed PI dynamics:

$$T_\theta(s) = \frac{K_{p,pos}(K_{p\omega}s + K_{i\omega})}{Js^3 + (B + K_{p\omega})s^2 + (K_{i\omega} + K_{p,pos}K_{p\omega})s + K_{p,pos}K_{i\omega}}.$$

With default values:

$$T_\theta(s) = \frac{0.24s + 4.8}{0.00016s^3 + 0.0160028s^2 + 0.56s + 4.8}.$$

5.6 Feedforward and Command Tracking

Define F(s) as the nominal plant from combined torque command to speed, and let:

$$Q(s) = \sigma_a K_{aff} J s + \sigma_B B, \qquad K_v = \sigma_v K_{vff}.$$

For smooth trajectories, αref=sωref and ωref=sθref; without limiting:

$$\frac{\omega_m}{\omega_{ref}} = \frac{F(s)[C_\omega(s) + Q(s)]}{1 + F(s)C_\omega(s)}.$$

For Position Mode:

$$\frac{\theta_m}{\theta_{ref}} = \frac{F(s)\{C_\omega(s)[K_{p,pos} + K_v s] + Q(s)s\}}{s + F(s)C_\omega(s)[s + K_{p,pos}]}.$$

Feedforward changes the numerator from command to output, leaving the feedback characteristic equation unchanged. Thus, in this unsaturated linear model, Kvff/Kaff do not change feedback PM/GM but may change Command-Tracking Bandwidth.

UI Command Tracking analyzes the path from the trajectory output reference to actual output, not the full nonlinear path from the target number through the trajectory generator. In time simulation, Speed Step has αref=0, so it cannot be treated as the same step response as smooth-reference Bode analysis with Js feedforward.

6. Digital Control and PWM Timing

6.1 Discrete PI Matching the Engine

Without saturation, output and integrator updates occur in this order:

$$u[k] = K_p e[k] + x[k], \qquad x[k+1] = x[k] + K_i T_s e[k].$$

Therefore:

$$C(z) = K_p + \frac{K_i T_s}{z-1}.$$

Do not arbitrarily replace this with Kp+KiTs·z/(z−1); that implies different output/integrator timing. For speed PI, replace Ts with Tω.

6.2 Full Derivation: Continuous RL, Input ZOH, and the Digital Current Loop

6.2.1 The Physical Plant Remains Continuous; ZOH Precedes It

PI computation and sampled signals are digital; the physical motor does not become a digital component. For one nominally decoupled current axis $\ell \in \{d,q\}$, the continuous electrical equation is:

$$L_\ell \frac{di_\ell(t)}{dt} + R_s i_\ell(t) = v_\ell(t), \qquad G_\ell(s) = \frac{I_\ell(s)}{V_\ell(s)} = \frac{1}{L_\ell s + R_s}.$$

L_ℓ is inductance in H; $G_\ell(s)$ is the continuous RL voltage-to-current plant transfer function. This nominal equation assumes valid dq decoupling and neglects saturation, dead time, switching harmonics, and uncancelled coupling/back-EMF disturbances.

The input sequence $v_\ell[k]$ passes through the input ZOH, becoming a piecewise constant continuous-time voltage $v_\ell(t)$ applied to the continuous RL plant. The ADC samples its output at kTs to produce $i_\ell[k]$. The path is:

Discrete voltage sequence → input ZOH → continuous RL → output sampling → discrete current sequence.

$G_{\ell,\text{hold}}(z)$ below is the discrete equivalent of the entire ZOH + continuous RL + output sampling combination. It neither makes the physical motor discrete nor moves the ZOH behind the RL plant. Time simulation still numerically integrates continuous states with PWM switching; this Bode model instead uses held average voltage for nominal linear analysis. Actual switching voltage is not perfectly flat throughout each period.

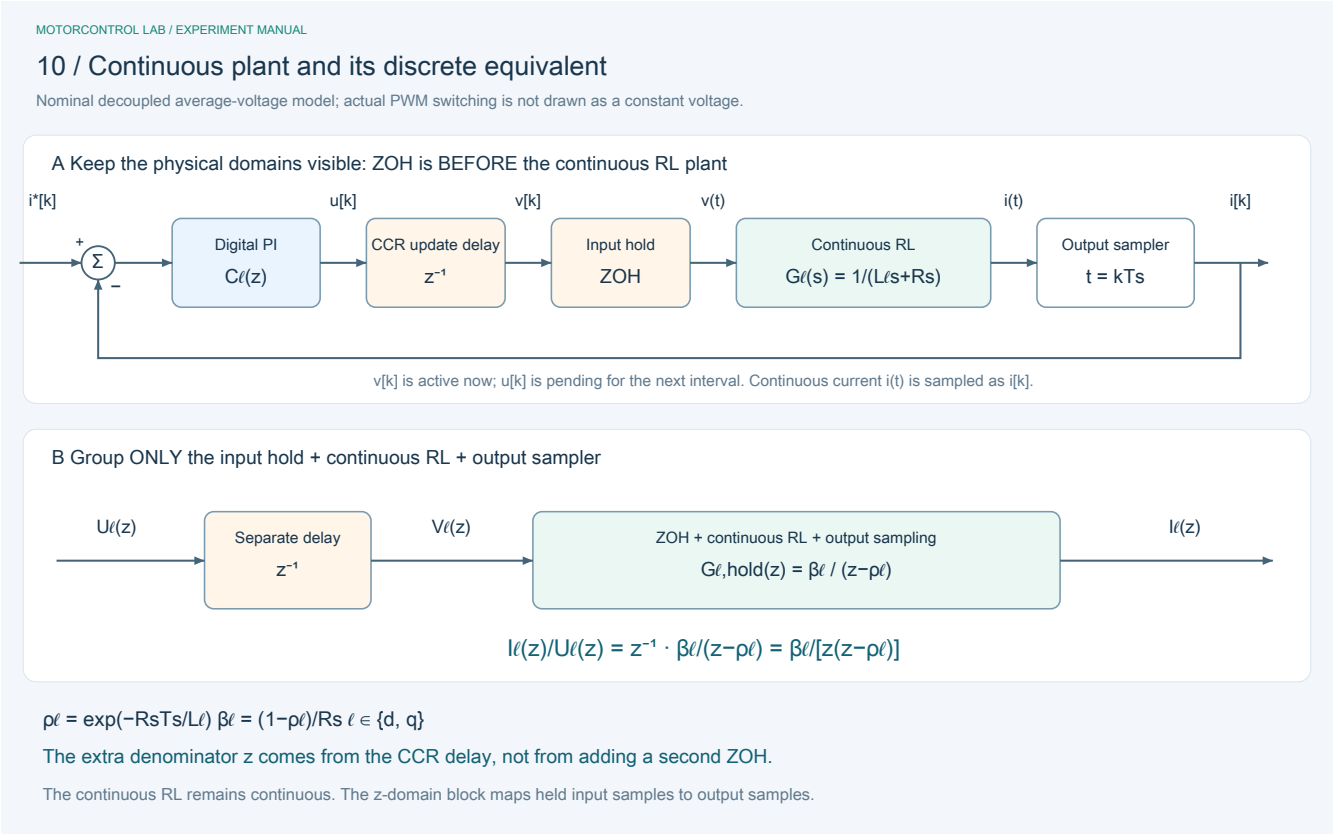

Figure 10 | Top: actual locations of ZOH, continuous plant, and sampler. Bottom: their combined discrete equivalent, with the extra one-sample CCR update delay kept outside the equivalent block.

6.2.2 Deriving a and b with an Integrating Factor: Differential to Difference Equation

A differential equation describes continuous-time rates of change; a difference equation relates values at adjacent sampling instants. We do not replace di/dt directly with a finite difference. Instead, solve the continuous differential equation exactly within each ZOH interval, then sample the solution.

To make lowercase a and b explicit, temporarily omit the d/q axis subscripts and let L denote that axis's inductance; we map back to ρ_ℓ and β_ℓ afterward. Assume Rs>0, L>0, and Ts>0.

Step 1 | Write the continuous equation and known initial value for this period. During period k, the ZOH keeps voltage at v[k]. Define local time τ=t−kTs and local current $\tilde{\imath}(\tau)=i(kT_s+\tau)$, so:

$$L\frac{d\tilde{i}(\tau)}{d\tau} + R_s\tilde{i}(\tau) = v[k], \qquad \tilde{i}(0) = i[k], \qquad \tilde{i}(T_s) = i[k+1].$$

Let r=Rs/L, in s⁻¹. Divide both sides by L:

$$\frac{d\tilde{i}}{d\tau} + r\tilde{i} = \frac{v[k]}{L}, \qquad r = \frac{R_s}{L}.$$

Over this interval, v[k], Rs, and L are constant; continuous current $\tilde{\imath}(\tau)$ varies.

What is the integrating-factor method, and why multiply by this function? It solves first-order linear differential equations by using the product rule to combine two separate terms into the derivative of a product, which can then be integrated directly. First suppose we multiply by an unknown function μ(τ):

$$\mu(\tau)\frac{d\tilde{i}}{d\tau} + r\mu(\tau)\tilde{i} = \frac{v[k]}{L}\mu(\tau).$$

The product rule gives:

$$\frac{d}{d\tau}\left[\mu(\tau)\tilde{i}(\tau)\right] = \mu(\tau)\frac{d\tilde{i}}{d\tau} + \frac{d\mu}{d\tau}\tilde{i}(\tau).$$

Comparing the equations, choose dμ/dτ=rμ to make the left sides identical. The simplest choice is μ(τ)=exp(rτ), whose derivative is r exp(rτ). Any nonzero constant multiplier also works but ultimately cancels, so take it as one. The factor follows from the product rule; it is not an arbitrary guess of an exponential.

Step 2 | Multiply by the integrating factor exp(rτ). By the product rule, the two left-hand terms combine into a single total derivative:

$$e^{r\tau}\frac{d\tilde{i}}{d\tau} + re^{r\tau}\tilde{i} = \frac{v[k]}{L}e^{r\tau},$$

$$\frac{d}{d\tau}\left[e^{r\tau}\tilde{i}(\tau)\right] = \frac{v[k]}{L}e^{r\tau}.$$

This explains the exponential: the RL natural response has first-order decay rate r=Rs/L, and exp(rτ) converts the equation into an easily integrated form.

Step 3 | Integrate from this period's start, 0, to the next sample, Ts. The left side becomes endpoint values; on the right, v[k] is constant:

$$\left[e^{r\tau}\tilde{i}(\tau)\right]_0^{T_s} = \frac{v[k]}{L}\int_0^{T_s} e^{r\tau}\,d\tau.$$

How is the left-hand definite integral evaluated? Let F(τ)=exp(rτ)ĩ(τ). By the fundamental theorem of calculus, the integral of a derivative from 0 to Ts equals the original function at the upper limit minus its value at the lower limit. Expanding each substitution:

$$\begin{aligned}\int_0^{T_s}\frac{d}{d\tau}\left[e^{r\tau}\tilde{i}(\tau)\right]d\tau &= \left[e^{r\tau}\tilde{i}(\tau)\right]_0^{T_s}\\ &= e^{rT_s}\tilde{i}(T_s) - e^{r\cdot 0}\tilde{i}(0)\\ &= e^{rT_s}i[k+1] - i[k].\end{aligned}$$

The bracket's lower 0 and upper Ts mean “substitute Ts, then subtract the value at 0,” not multiply by Ts. Since exp(0)=1, the second term is i[k]. Local τ=0 is the start of period k; it does not mean zero current. i[k] is zero only if separately specified by the initial condition.

Watch the notation: the left side is exp(rTs)·i[k+1]−i[k], not i[k+1]/(rTs)−i[k]. The factor 1/r appears in the antiderivative of the exponential on the right. The two sides are evaluated differently.

How is the right-hand integral evaluated? v[k]/L is constant over this period, so it can be taken outside the integral. Since d[exp(rτ)/r]/dτ=exp(rτ):

$$\int_0^{T_s} e^{r\tau}\,d\tau = \left[\frac{e^{r\tau}}{r}\right]_0^{T_s} = \frac{e^{rT_s}-1}{r}.$$

Using Lr=Rs and endpoint values ĩ(0)=i[k], ĩ(Ts)=i[k+1], we obtain:

$$e^{rT_s} i[k+1] - i[k] = \frac{v[k]}{Lr}(e^{rT_s} - 1) = \frac{v[k]}{R_s}(e^{rT_s} - 1).$$

Step 4 | Multiply both sides by exp(−rTs) and rearrange as “next sample = decayed old current + this period's input contribution.”

$$\begin{aligned} i[k+1] &= e^{-rT_s} i[k] + \frac{1 - e^{-rT_s}}{R_s} v[k] \\ &= e^{-R_s T_s / L} i[k] + \frac{1 - e^{-R_s T_s / L}}{R_s} v[k]. \end{aligned}$$

Name the coefficient of i[k] as a and the coefficient of v[k] as b to obtain the required difference equation:

$$\boxed{i[k+1] = a\, i[k] + b\, v[k]}, \qquad \boxed{a = e^{-R_s T_s / L}, \quad b = \frac{1 - e^{-R_s T_s / L}}{R_s} = \frac{1 - a}{R_s}}.$$

a is not assigned arbitrarily: it is the fraction of the old current remaining after one Ts. b is the coefficient describing how this period's held voltage affects the next current sample. a is dimensionless and b has units A/V, so both terms are currents.

The result can also be interpreted using the electrical time constant τe=L/Rs:

$$\tau_e = \frac{L}{R_s}, \qquad a = e^{-T_s / \tau_e}, \qquad i[k+1] = a\, i[k] + (1 - a)\frac{v[k]}{R_s}.$$

Here v[k]/Rs is the steady current approached if this voltage were maintained indefinitely. After one period, the current retains a times its old value and advances a fraction 1−a toward the new steady value. For positive parameters, 0<a<1; current does not jump instantly to v[k]/Rs.

Step 5 | Map back to this chapter's notation and UI defaults. The a and b in this subsection are the axis-ℓ quantities:

$$a = \rho_\ell = e^{-R_s T_s / L_\ell}, \qquad b = \beta_\ell = \frac{1 - \rho_\ell}{R_s}, \qquad i_\ell[k+1] = \rho_\ell i_\ell[k] + \beta_\ell v_\ell[k].$$

For q-axis Rs=1.2 Ω, Lq=0.0125 H, and Ts=100 μs, the electrical RL time constant Lq/Rs≈10.4167 ms, a≈0.9904459329, and b≈0.0079617226 A/V. This is the continuous RL plant's electrical time constant, not the 1 ms closed-loop time constant after PI pole-zero cancellation, nor the equivalent delay of 0.15 ms.

Conditions during this period	Next sample	Coefficient meaning
v[k]=0, i[k]=1 A	i[k+1]=a·1 A≈0.9904459329 A	a is the fraction remaining after natural current decay with no applied voltage
v[k]=1 V, i[k]=0	i[k+1]=b·1 V≈0.0079617226 A	b describes the current generated from zero by 1 V held for one period

No extra one-sample CCR update delay has yet been added. v[k] is the held voltage already active in this period. To use the newly computed PI command u[k], substitute v[k]=u[k−1] as in Section 6.2.4. Keep these two commands distinct when deriving a and b.

6.2.3 z-Transforming the Difference Equation: β/(z−ρ)

Set initial current to zero and let Iℓ(z), Vℓ(z) be the z-transforms of iℓ[k], vℓ[k]. Since iℓ[k+1] transforms to zIℓ(z), the equation becomes:

$$zI_\ell(z) = \rho_\ell I_\ell(z) + \beta_\ell V_\ell(z).$$

Rearrange and form the output/input ratio:

$$(z-\rho_\ell)I_\ell(z) = \beta_\ell V_\ell(z), \qquad \boxed{G_{\ell,hold}(z) \equiv \frac{I_\ell(z)}{V_\ell(z)} = \frac{\beta_\ell}{z-\rho_\ell}}.$$

No extra computation/CCR update delay has yet been included. The input is vℓ[k], already active during this period. Evolution from iℓ[k] to the next sample iℓ[k+1] is the continuous RL plant's response during one hold interval.

The same expression can also be written as:

$$G_{\ell,hold}(z) = \frac{\beta_\ell z^{-1}}{1-\rho_\ell z^{-1}}.$$

These are algebraically identical transfer functions. The z^{-1} in this form is not another CPU delay; it arises from the state equation's time indices and plant evolution. An additional scheduling delay depends on whether the newly computed command becomes active in the same period; this is added next.

6.2.4 Why Another z in the Denominator? New Commands Act One Period Later

Distinguish two voltage sequences:

Symbol	Meaning	Timing in this version
uℓ[k]	New voltage command calculated by PI after sample k	Pending, intended for the next period
vℓ[k]	Active voltage command entering the input ZOH during period k	Active, obtained from the previous uℓ[k−1]

The rule “apply at the next valley” is therefore:

$$v_\ell[k] = u_\ell[k-1], \qquad V_\ell(z) = z^{-1}U_\ell(z).$$

This is the additional one-sample CCR update delay. Gℓ,hold(z) originally relates Vℓ to Iℓ. To relate newly computed Uℓ to measured Iℓ, multiply by Vℓ/Uℓ=1/z:

$$\boxed{\frac{I_\ell(z)}{U_\ell(z)} = \frac{1}{z}\,\frac{\beta_\ell}{z-\rho_\ell} = \frac{\beta_\ell}{z(z-\rho_\ell)}}.$$

Thus the extra z in the original denominator is not arbitrary, an extra inductor state, or a second ZOH. It arises because the newly computed command waits one Ts before entering the existing hold. Substituting $v_\ell[k]=u_\ell[k-1]$ into the state equation gives the same result:

$$i_\ell[k+1] = \rho_\ell i_\ell[k] + \beta_\ell u_\ell[k-1].$$

Compare the first few samples. Start with zero current and zero old command, and apply a constant new voltage test command U0=1 V from k=0. This tests only the voltage-to-current path; PI feedback is not yet closed:

Sampling instant	New command active immediately; no extra one-sample delay	New command active next period; this version's timing
k=0, t=0	i[0]=0	i[0]=0
k=1, t=Ts	$i[1]=\beta_\ell \cdot U0$	i[1]=0; the first period still uses old voltage 0
k=2, t=2Ts	$i[2]=\beta_\ell(1+\rho_\ell)U0$	$i[2]=\beta_\ell \cdot U0$

The first nonzero current sample occurring one index later is the effect of the additional 1/z between the two transfer functions. Distinguish it from the time the RL plant inherently takes to change current.

6.2.5 Reconnect Discrete PI to Obtain Open and Closed Loops

Section 6.1 defines the controller that computes output before updating the integrator:

$$C_\ell(z) = K_{p\ell} + \frac{K_{i\ell} T_s}{z-1}.$$

Let $E_\ell(z)=I_\ell^*(z)-I_\ell(z)$ and PI output $U_\ell(z)=C_\ell(z)E_\ell(z)$. Multiply along the negative-feedback path PI → one-sample delay → ZOH/RL/output sampling:

$$\boxed{\mathcal{L}_\ell(z) = C_\ell(z)\,\frac{1}{z}\,G_{\ell,hold}(z) = \left(K_{p\ell} + \frac{K_{i\ell} T_s}{z-1}\right)\frac{\beta_\ell}{z(z-\rho_\ell)}}.$$

Calligraphic $\mathscr{L}(z)$ denotes open loop to distinguish it from inductance L_ℓ. Closing unity negative feedback gives:

$$T_\ell(z) \equiv \frac{I_\ell(z)}{I_\ell^*(z)} = \frac{\mathcal{L}_\ell(z)}{1+\mathcal{L}_\ell(z)}.$$

Evaluate Bode frequency response at $z=\exp(j\omega T_s)$, $\omega=2\pi f$. This equivalent model describes current at sampling instants; it does not claim continuous current is piecewise constant throughout each sampling interval.

ZOH and the extra one-sample delay are each counted once. Do not add another half PWM period or $\exp(-1.5sT_s)$ to $\mathscr{L}(z)$. Section 5.4.2's 1.5Ts is an alternative continuous low-frequency average approximation and must not be stacked onto this exact sampled model.

6.2.6 Previous Notation and Default q-Axis Values

In the earlier notation, “axis a” meant a∈{d,q}, not phase a of abc. Using La for both inductance and open loop was also ambiguous. This section separates the symbols:

Previous symbol	Notation here	Meaning
La, without (z)	L_ℓ	d- or q-axis inductance in H
aa	ρ_ℓ	e^(−RsTs/L_ℓ), state decay coefficient
ba	β_ℓ	(1−ρ_ℓ)/Rs, discrete held-voltage input coefficient
La(z)	$\mathscr{L}$(z)	Open-loop transfer function of the complete current feedback loop, not inductance
Ga,ZOH(z)	G_ℓ,hold(z)	Input ZOH + continuous RL + output sampling equivalent, without the extra update delay

With default q-axis Rs=1.2 Ω, Lq=0.0125 H, and Ts=100 μs:

$$\rho_q = e^{-0.0096} \approx 0.9904459329, \qquad \beta_q = \frac{1-\rho_q}{1.2} \approx 0.0079617226\ \mathrm{A/V}.$$

Thus active voltage Vq to sampled current Iq is βq/(z−ρq); newly computed voltage Uq to Iq is βq/[z(z−ρq)]. For the d axis, use Ld, Kpd, and Kid; the derivation is identical.

6.2.7 What Are the Web Current-Loop Bode Plots?

The earlier shorthand “page current Bode” did not denote a special physical quantity and was unclear. It means the magnitude and phase curves for the d- and q-axis current control loops in MotorControl Lab's Loop Bode Analysis panel. Bode describes frequency-response plots, not a kind of “page current.”

D-axis current loop and Q-axis current loop each use $\mathscr{L}$(z) for open-loop response and T$_\ell$(z)=$\mathscr{L}$/(1+$\mathscr{L}$) for closed-loop response. These curves are analytical nominal-model results, not FFTs of current time waveforms or the three-phase system's phase-a current.

They always use this section's delayed discrete model. Selecting Ideal Inner Current Loop under Speed model does not set their Tq to one; it changes only the inner-loop model used for speed/position Bode. Nominal analysis still omits switching ripple, dead time, limits, and AW nonlinearities.

For the underlying concepts, see MathWorks: continuous/discrete equivalence with ZOH and Unit Delay: one sample corresponds to z^{-1}.

6.3 Event Order at Valley k

1. Apply pending CCR from the previous valley as active CCR[k].

2. Simultaneously latch phase currents and encoder angle to produce ADC[k].

3. Update position/speed controllers when scheduled.

4. Execute Clarke, Park, current PI, decoupling, vector limiting, inverse Park, and SVPWM.

5. Compute pending CCR[k+1], activated only at the next valley.

6. During this period, the plant is driven by switching voltage from active CCR[k].

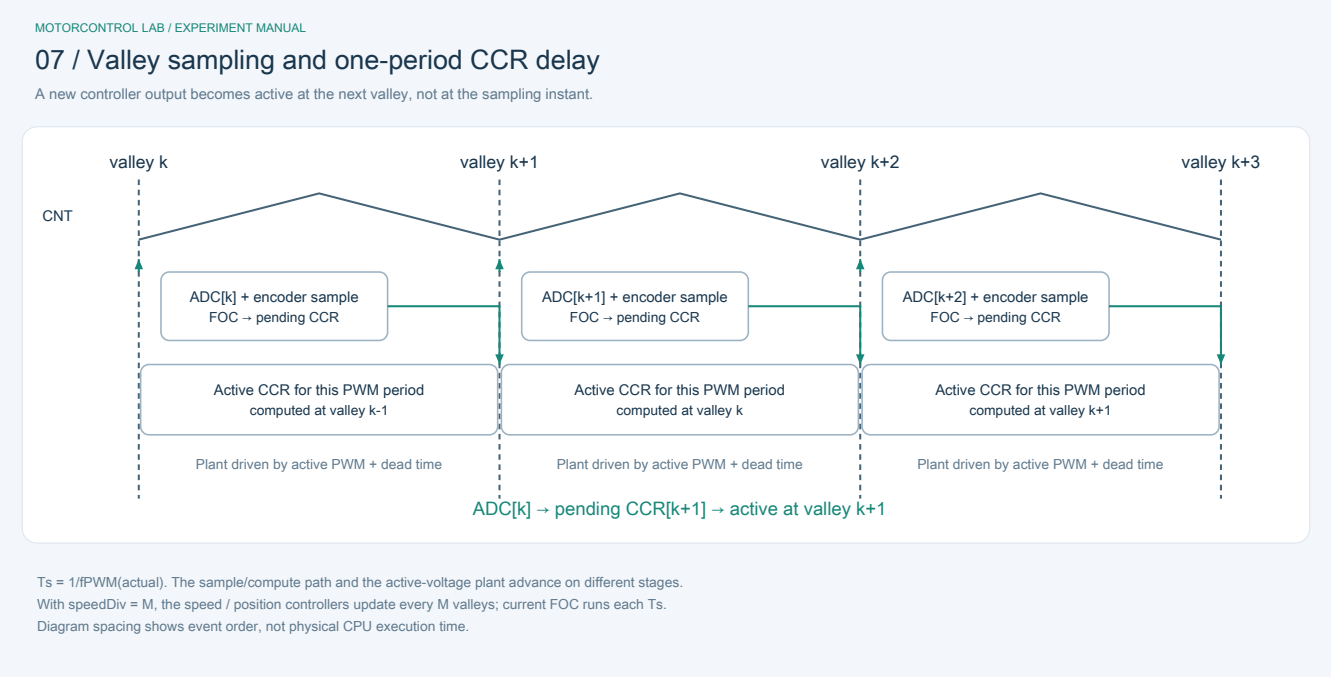

Figure 07 | Valley sampling, pending CCR, and next-period activation. Blue denotes controllers, orange denotes limits or AW, and green arrows denote feedforward. Solid arrows show signal direction.

Default M=1 updates both speed and current PI at 10 kHz. Increasing speedDiv only lowers the outer-loop update rate. The digital speed model adds no separate outer-loop computation delay: its torque output enters current FOC at that outer update, while current FOC retains its established one-period delay.

7. Reading Waveforms and Diagnostics

7.1 Control Response / Overview

Waveform	Signals and units	Interpretation
Speed response	Speed ref, Speed actual; rpm	ref is limited ωcmd converted to rpm, not necessarily the raw target; hidden in Current Mode
dq current control	Id, Iq, Id, Iq; A	Actual dq states and effective commands; iq* already includes q-current limiting
Three-phase currents	Ia, Ib, Ic; A	Actual instantaneous phase currents; locked-rotor currents need not form three time-rotating sinusoids
Electromagnetic torque / load	Te raw, Te limited, Te actual, TL; N·m	Command, torque limiting, and actual torque; the bypassed speed-loop command in Current Mode is not an actual torque target
Position servo	θm*, θm; rad	θm* is trajectory θref, not an instantaneous jump to final target; Position only
Trajectory speed / command	ωref, ωm, ωcmd; rad/s	Distinguish planned speed, actual speed, and the command including position correction and speed limiting
Trajectory acceleration	αref; rad/s²	Planned acceleration used by FF, not actual dωm/dt
Torque command contributions	Te feedback, Jα FF, Bω FF, Total raw, Limited; N·m	FF curves already include their switches and gains
Position following error	θref−θm; rad	Error relative to the moving trajectory, not final target−θm

Even with constant iq in Current Mode, Te can depend on id, Ld−Lq, and ripple. Unchanged speed does not imply zero electromagnetic torque with the rotor locked.

MotorControl Lab | © 2026 Dr. Jack Yeh | All rights reserved.

7.2 Sensored feedback chain

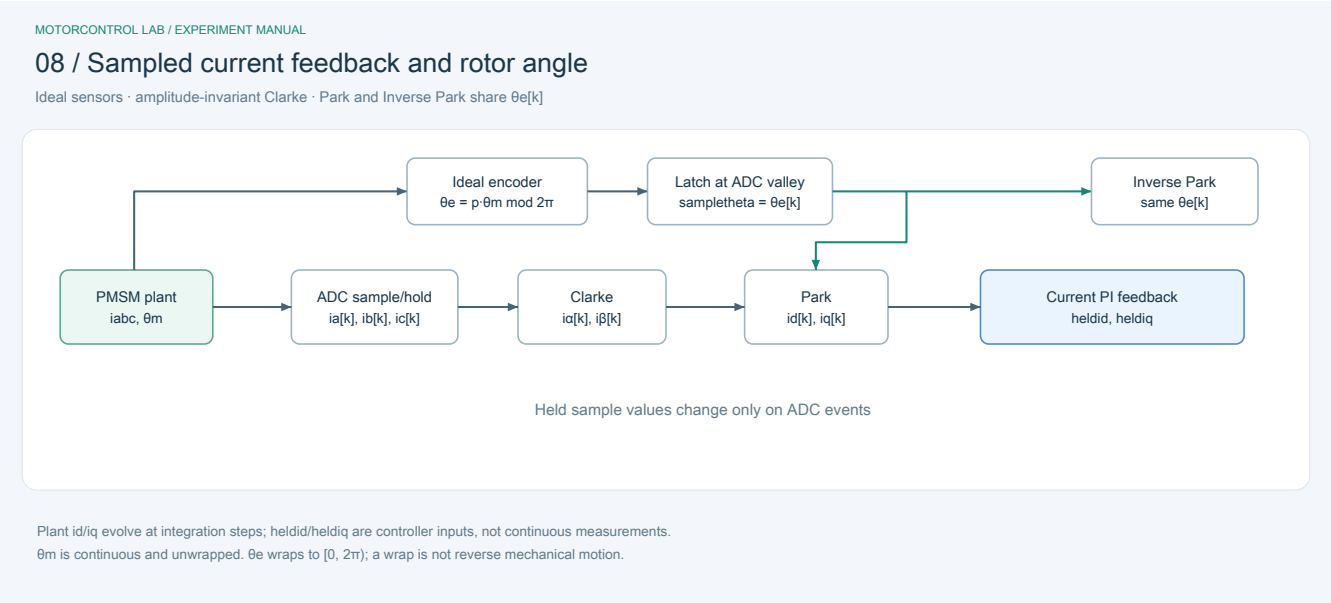

Figure 08 | ADC, Clarke, Park, and synchronously latched electrical-angle feedback chain. Blue denotes controllers, orange denotes limits or AW, and green arrows denote feedforward. Solid arrows show signal direction.

Waveform	Description
ADC sampled phase currents	sampleia/b/c are latched at the ADC event and held until the next sample
Clarke / stationary current frame	ialpha, ibeta are held values transformed from sampled phase currents
Park feedback vs plant dq state	heldid/heldiq are feedback seen by the controller; id/iq are continuous plant states
Encoder / mechanical and electrical angle	θm is continuous; θe wraps to [0,2π); sampletheta is sampled electrical angle
Inverse Park voltage commands	Vα, Vβ are post-vector-limit commands sent to SVPWM, not actual phase voltages
ADC hold detail	Sampled/plant comparison over the final 8 PWM periods, useful for observing sample-and-hold

Sawtooth resets in θe are angle wrapping, not rotor reversal. Use unwrapped θm to judge multiple-revolution position.

7.3 Controller Output / Anti-Windup

Diagnostic values from each control event are held until the next update; overview data may still be downsampled. Select legend entries to hide unwanted curves.

d/q-axis PI / vector saturation:

Legend	d/q CSV fields	Meaning	Unit
P term	dp / qp	Kp·e	V

Legend	d/q CSV fields	Meaning	Unit
I state (used)	di / qi	x[k] used to form this output	V
PI raw	dpi / qpi	P+I, without decoupling	V
Decoupling voltage	dff / qff	Electrical-speed cross-coupling and back-EMF compensation	V
Pre-vector voltage	draw / qraw	PI raw + decoupling	V
Post-vector voltage	dsat / qsat	Command after shared vector limiting	V
AW correction	daw / qaw	Kaw·(post−pre) when enabled, otherwise 0	V/s

Examine dsat and qsat together as a vector magnitude. One axis below Vmax does not imply no saturation. AW correction is a correction to the integrator rate, not a voltage or the entire integrator rate; the full rate is Ki·e+AW.

Post-vector voltage has not yet passed through the one-period CCR delay, quantization, switching, or dead time; it is not the applied instantaneous plant dq voltage. CSV vd/vq are transformed from actual switching phase voltages and will not equal dsat/qsat sample by sample.

Speed PI / torque limits：

Legend	CSV field	Meaning
P term	speedP	Kpω·eω, N·m
I state (used)	speedI	Integrator state used for this PI output, N·m
PI raw	tefb	P+I, N·m
Total raw	teraw	PI + acceleration FF + friction FF, N·m
Torque-limited	teref	After torque command limiting, N·m
After current limit	speedEffective	Kt·iq*, N·m; equivalent torque of the effective command, not actual Te
AW correction	speedAW	Kawω·(speedEffective−teraw), N·m/s

CSV speedIntegral is updated xω[n+1], whereas speedI is xω[n] used for the current output. Use speedI to check P+I=PI raw.

Position P / speed limit：

• Position P raw / omegafb: Kp,pos·(θref−θm), rad/s.

• With velocity FF / positionRawSpeed: position P plus velocity feedforward, ωfb+Kvff·ωref, rad/s. Acceleration feedforward is not included here; it enters the torque branch.

• Speed-limited / speedcmd: limited ωcmd, rad/s.

The position loop contains only this P controller and limit; there is no position I or position AW in the plots.

7.4 Switching / Dead time

Waveform	Purpose
Center-aligned CNT / CCR	CNT and active CCR A/B/C in counts; inspect comparison crossings
PWM switching / ADC / apply event	Sa/Sb/Sc logic, ADC event, and CCR apply event; apply means loading pending CCR and can be 1 even when its numerical value is unchanged
Phase-to-neutral voltage	Van/Vbn/Vcn in V, with floating neutral removed and enabled dead-time disturbance included
Dead-time voltage error	ΔVan/ΔVbn/ΔVcn are errors after neutral reconstruction; ΔVaO is phase-A pole error
Switching current detail	Final-segment phase-current ripple, showing switching and dead-time effects
Sample & hold / current state	Final-segment differences between id/iq and heldid/heldiq

High-speed plots retain only the last 8 PWM periods of the simulation. Their horizontal axis is μs from that window's start; overview time is ms from simulation start. Moving the cursor to an earlier time does not automatically move the high-speed window there.

7.5 Metrics and Data Resolution

• Final rpm/iq: final integrated states, not final-segment averages or guaranteed steady-state values.

• Peak current: maximum absolute phase current at simulation integration nodes.

• Saturation %: fraction of current-control updates with voltage vector limiting, not the torque-limit fraction.

• Cost: computation time, not simulated Duration.

• Overview retains roughly up to 6000 points and can miss switching edges or brief events. High-speed plots retain all final-segment plant nodes, but switching-edge resolution remains limited by the plant step.

MotorControl Lab | © 2026 Dr. Jack Yeh | All rights reserved.

8. Bode and FFT Spectrum Analysis

1. Select current settings or the last simulation snapshot. Prefer the snapshot when explaining existing waveforms.

2. Current Mode shows d/q current loops only; Speed Mode adds the speed loop; Position Mode also adds the position loop.

3. Choose Ideal, With current-loop dynamics, or With digital delay under Speed model (see the comparison and scope in Section 5.4). This changes analysis only, not the time simulation engine; current Bode always uses the digital model in Section 6.

4. Update Bode and inspect crossover frequencies, PM, GM, and the two bandwidths.

Curve / field	Definition
Open loop L	Open-loop transfer function around the selected negative-feedback loop
Feedback Closed-Loop	L/(1+L), excluding reference feedforward
Command Tracking	Reference→output including enabled reference FF; magnitude only
Gain crossover / PM	Open-loop 0 dB crossing and 180°+open-loop phase
Phase crossover / GM	Open-loop −180° crossing, including later phase branches, and required gain margin
Nyquist	Half the loop sampling rate; not a recommended control bandwidth
Feedback BW	First downward crossing 3.0103 dB below the feedback closed loop's own flat low-frequency gain
Command-Tracking BW	Bandwidth of command tracking defined in the same way

N/A may mean no reliable crossing, nonflat low-frequency response, an invalid scan, or no gain crossover with positive PM. It does not automatically mean instability or infinite bandwidth. Margins are estimated by frequency scanning, not a complete proof of stability.

Bode analysis excludes current/voltage limiting, AW, dead time, switching ripple, and full nonlinear dq coupling. The ideal decoupled reference remains displayed even when decoupling is disabled. With actual saturation, small-signal bandwidth should not predict large-step settling time.

8.1 FFT: From Time-Domain Waveforms to Spectra

FFT / Spectrum Analysis sits between Waveform instruments and Loop Bode Analysis. It analyzes recorded time-domain signals from the last completed simulation, not the nominal Bode model or an FFT of a Bode curve. Current, voltage, torque, speed, position, trajectory, controller/AW diagnostics, and PWM/timer channels are selectable, along with derived mechanical speed omega_m in rad/s converted from rpm.

MotorControl Lab | © 2026 Dr. Jack Yeh | All rights reserved.

8.1.1 Procedure and Current Defaults

1. Run a simulation, then go to FFT / Spectrum Analysis.

2. Data Source defaults to Overview (decimated). Choose a signal such as Ia [ia], Vd actual [vd], or I state [qi]. Brackets identify the actual data field; its unit is shown alongside.

3. Set FFT start / FFT end in ms. Changing the data source or receiving a new simulation result resets the range to that source's full available record.

4. Select a window: Hann by default, with Blackman and Rectangular also available. Remove mean (DC) is checked by default; the selected samples' mean is subtracted before windowing.

5. Select Analyze FFT. Inspect the spectrum, Fs, Nyquist, Δf, actual sample range, and sample count. Use the chart zoom tools to inspect a frequency band.

6. Spectrum scale switches between peak amplitude, Amplitude (dB re 1 unit), and PSD, reusing the same calculation. Changing the signal, source, time range, window, or mean-removal setting requires a new calculation.

7. Select Export FFT CSV to save every frequency bin and analysis metadata. Editing motor/controller parameters without rerunning does not create a new measurement: FFT continues to use the previous simulation snapshot.

Setting	Current default	Meaning
Data Source	Overview (decimated)	Starts with a longer record; decimation aliasing limitations still apply
Signal	Ia [ia] · A	Phase-A current; select other channels from the dropdown
Start / End	Full available source range	Entered in ms; separate from simulation command Start / Duration
Window	Hann	Reduces truncation-edge leakage but broadens the main lobe
Remove mean (DC)	Enabled	Subtracts the selected segment's mean; does not remove all low-frequency variation
Spectrum scale	Peak amplitude	Single-sided peak amplitude in the original signal unit, not RMS

8.1.2 Sampling Limits of Overview and High-Speed Data

Source	Actual record	Uses and limitations
Overview (decimated), default	Decimated data spanning the full simulation: approximately up to 6000 regularly spaced points, possibly plus a final endpoint	The longer record helps inspect low-frequency current, voltage, control-response components, and startup transients. No anti-alias filter is applied. Switching components can fold into low frequencies; this is not a precise low-pass-filtered acquisition.
High-speed (last 8 PWM periods)	All plant integration nodes from the final 8 PWM periods	Useful for switching-related components. A short record cannot resolve a low electrical fundamental. If the whole simulation is shorter than 8 periods, only the existing data are available.

FFT Fs is the actual sampling rate of the selected data source, not automatically the PWM frequency or timer clock. With sample interval Δt and N selected samples:

$$F_s = \frac{1}{\Delta t}, \qquad T_{record} = N\Delta t, \qquad \Delta f = \frac{F_s}{N} = \frac{1}{T_{record}}, \qquad f_{Nyquist} = \frac{F_s}{2}.$$

Δf is frequency-bin spacing. Window main-lobe width also affects whether nearby tones can be distinguished. The first and last sample are (N−1)Δt apart, whereas the frequency grid uses record duration NΔt; the UI reports actual endpoint times and NΔt separately.

This version neither zero-pads nor truncates to a power of two; 8–262144 samples are required. Overview may include one extra endpoint off its regular decimation grid. FFT omits this sample if it forms a shorter final interval and reports the omitted-endpoint count. Other nonuniform sampling is rejected, not silently interpolated; slow and fast data are not concatenated.

Overview increases available record duration; it does not increase the original sampling rate or eliminate aliasing. For a steady-state fundamental, first confirm approximate steady state and select several electrical periods. An FFT combining startup, load application, and changing speed describes that nonstationary record, not one final operating-point frequency. A shorter selection reduces frequency resolution; recalculating alone cannot change this tradeoff.

8.1.3 Windows, Amplitude, dB, and PSD

Let x[n] be the selected signal and w[n] the window. With mean removal checked, replace x[n] by x[n] minus its mean before applying the expression below; otherwise use the original signal:

$$X_w[k] = \sum_{n=0}^{N-1} x[n]w[n]e^{-j2\pi kn/N}, \qquad f_k = \frac{kF_s}{N}, \quad 0 \le k \le \lfloor N/2 \rfloor.$$

Window	Periodic window used here	Interpretation
Hann, default	0.5−0.5 cos(2πn/N)	A starting point for general spectra; reduces edge leakage with a wider main lobe than a rectangular window

Window	Periodic window used here	Interpretation
Blackman	0.42−0.5 cos(2πn/N)+0.08 cos(4πn/N)	Lower sidelobes at the cost of a wider main lobe
Rectangular	w[n]=1	Unweighted truncation; coherent sampling supports accurate amplitude readings, but noninteger-period records leak more

Single-sided peak amplitude includes coherent-gain correction. Define c_k, then amplitude and PSD as follows:

$$c_k = \begin{cases} 1, & k = 0 \text{ or } (N \text{ even and } k = N/2), \\ 2, & \text{otherwise}, \end{cases} \qquad A[k] = \frac{c_k |X_w[k]|}{\sum_n w[n]}, \qquad CG = \frac{1}{N}\sum_n w[n].$$

$$PSD[k] = \frac{c_k |X_w[k]|^2}{F_s \sum_n w[n]^2}.$$

Display	Units and definition	Notes
Peak amplitude	Original signal unit, such as A, V, N·m, or rpm	Calibrated for coherent sinusoid peak amplitude, not RMS. DC and the even-N Nyquist bin are not doubled. Off-bin tones still leak and their peak can be underestimated.
Amplitude (dB re 1 unit)	Amplitude ratio relative to one original signal unit	Current references 1 A and voltage references 1 V; this is not dBFS. The display and CSV dB column have a −240 dB floor; linear amplitude and PSD retain their original values.
PSD	Signal-unit squared per Hz, such as A²/Hz or V²/Hz	Power spectral density; bin height is not directly the peak amplitude of a sinusoid.

$$A_{dB}[k] = 20\log_{10}\left(\frac{A[k]}{1 \text{ signal unit}}\right).$$

Mean, RMS, and AC RMS are computed from the selected unwindowed original segment. AC RMS subtracts its mean first. They are not the height of the largest FFT bin. Strongest non-DC bin finds only the largest bin above zero frequency; it is not a fundamental-frequency estimator, particularly for short, nonstationary, or aliased records.

8.1.4 Default Speed Mode IA Example: Why 1.25 kHz and 0.2897 A Appear

Retain default Speed Mode and choose Ia [ia], High-speed, Hann, mean removal, and Peak amplitude. Recomputing this version's default model gives a strongest non-DC bin of approximately 1.25 kHz, 0.289727 A. This is current amplitude in A, not a maximum voltage.

Quantity	Default case in this version
PWM frequency	10 kHz

Quantity	Default case in this version
High-speed sample interval / rate	1.25 µs / 800 kHz
High-speed sample count	640, the final 8 PWM periods
Effective record duration / bin spacing	0.8 ms / 1.25 kHz
Final actual speed / pole pairs	Approximately 1008.215 rpm / 2; 1500 rpm is the command target, not actual speed here
Estimated final electrical frequency	Approximately 33.607 Hz, calculated from actual speed

$$T_{record} = \frac{8}{10000} = 0.0008\ \text{s}, \qquad \Delta f = \frac{1}{0.0008} = 1250\ \text{Hz}, \qquad f_e = \frac{p\, n_{rpm}}{60} \approx 33.607\ \text{Hz}.$$

This window covers only about 0.027 electrical periods, less than a single cycle. FFT bins start at 0, 1250, and 2500 Hz and cannot resolve a fundamental near 33.6 Hz. After mean removal and windowing, slow variation within the short segment can be largest at the first non-DC bin. 0.2897 A is the spectrum amplitude of that short segment, not the full-cycle IA current peak or proof of a dominant physical oscillation at 1.25 kHz.

Selecting the default Overview full range in this case gives approximately 5883 valid uniform samples, Fs≈23.529 kHz, and Δf≈4 Hz, making low-frequency distributions more visible. Decimation aliasing and startup/load transients still matter. Quantitative steady-state fundamental/harmonic analysis requires a record spanning several periods with adequate sampling and anti-aliasing conditions. The current high-speed source stores only the final 8 PWM periods; zero padding or a different window cannot recover the missing longer record.

8.1.5 FFT CSV and Reading Checklist

Export filenames follow motor-lab-fft-signalfield-datasource.csv. Comment metadata includes the field, unit, slow/fast source, N, Fs, Δf, actual time range, window, mean-removal setting, statistics, and simulation parameter snapshot. Columns are frequency_Hz, peak_amplitude, amplitude_dB_re_1_unit, and PSD_unit_squared_per_Hz. All three scalings are included even if only one is displayed.

Read spectra in this order: signal/unit → source → actual time range → Fs/Nyquist → Δf/window → mean removal → steady state and aliasing risk → physical meaning of peaks. FFT does not alter the controller, time simulation, or Bode model.

References: NumPy FFT definitions and normalization, SciPy periodogram single-sided and PSD scaling.

MotorControl Lab | © 2026 Dr. Jack Yeh | All rights reserved.

9. Six Experiments

Record complete parameters, the post-run snapshot, and CSV for each experiment. Change only one condition at a time when comparing. The settings below override the current defaults.

Experiment 1: Locked-Rotor dq Current Step

Purpose: Remove mechanical acceleration and back-EMF to establish a current-loop baseline.

Settings: Current, Locked Rotor, id=0, iq=5 A, TL=0, Start=0.02 s, Duration=0.25 s; other settings at default.

Procedure: Run and inspect dq current, q-axis PI/vector diagnostics, and encoder angle. Record final-segment mean iq and ripple, not Final iq alone.

Expected: iq rises to approximately 5 A and id stays near zero; ωm and θm remain zero throughout. Steady q voltage is approximately Rs·iq=6 V, with actual command also compensating dead time, quantization, and ripple. A large initial error may briefly cause voltage saturation without implying failed steady-state tracking.

Questions: Why is Te nonzero at zero speed? Why can iq be DC while phase currents are not three time-domain sinusoids?

Experiment 2: Free Running, Back-EMF, and Loss of Current Tracking

Purpose: Distinguish PI steady-state error from insufficient voltage capability.

Settings: As Experiment 1, changing only to Free Running.

Expected observations: iq initially reaches 5 A and torque accelerates the rotor. As ωeψf rises, q decoupling voltage increases; Pre-vector and Post-vector diverge and iq may decrease. Loss of current tracking does not necessarily indicate faulty PI design.

For id≈0 and nearly steady iq, estimate:

$$v_d \approx -\omega_e L_q i_q, \qquad v_q \approx R_s i_q + \omega_e \psi_f.$$

Assess voltage demand using sqrt(vd²+vq²)≤Vmax, not vq≤Vmax alone. Once iq has fallen and electrical states are still changing, these expressions indicate trends only.

Previous model regression reference: At 40 steps/PWM and 0.25 s with these settings, locked-rotor Final iq was approximately 5.002 A; free running gave approximately 2973 rpm and Final iq≈1.177 A. These are earlier model test values, not exact acceptance standards promised by this manual. Current defaults use 80 steps/PWM, so readings can differ.

Experiment 3: Current Vector Saturation and AW On/Off

Purpose: Directly observe integrator accumulation and back-calculation correction.

Settings: Current, Locked Rotor, Vdc=12 V, iq=30 A, Iq limit=40 A, id=0, TL=0, Start=0.02 s, Duration=0.10 s.

Procedure: Run and export with AW on, then repeat with AW off. Compare q I state, PI raw, Pre/Post-vector, and AW correction.

Expected: Neither run reaches 30 A under sustained limiting. With AW on, the integrator is usually restrained. With AW off, correction stays zero and integration continues accumulating. A larger I state does not mean better control.

Numerical checks: qpi=qp+qi; qraw=qpi+qff; with AW on, qaw=Kaw·(qsat−qraw). This page has no arbitrary piecewise current-command editor. A sustained high-command experiment therefore cannot establish recovery time after saturation is removed.

Experiment 4: Speed PI and Two Command Limits

Purpose: Understand the torque units of Speed PI and what its AW tracks.

Settings: Speed, 1500 rpm, Step, Start=0.02 s, Duration=0.5 s, TL=0; Torque limit=0.10 N·m, Iq limit=12 A; otherwise defaults.

Procedure: Compare speedI, teraw, teref, speedEffective, and speedAW with AW on/off. Then keep AW on, restore Torque limit=1.8 N·m, set Iq limit=1 A, and rerun.

Expected: The first case primarily reaches torque limiting. In the second, effective command torque is limited to about Kt·1 A=0.09 N·m. teref may remain below its limit while speedEffective is already constrained. Check that speedAW back-calculates the effective torque difference.

Extension: Restore command limits and add a 0.15 N·m load at 0.15 s. Observe speed drop, increased current, and PI integral compensation. If voltage saturates, inspect inner-loop diagnostics too; speed PI curves alone cannot explain the response.

Experiment 5: Position Trajectory and Velocity/Acceleration Feedforward

Purpose: Distinguish trajectory planning, position feedback, and feedforward.

Settings: Select Position Control as Control Mode. Target Position=1 rad, Max Velocity (position)=10 rad/s, Max Acceleration=100 rad/s², Start=0.02 s, Duration=0.5 s, TL=0. With FF enabled, Kvff=1 and Kaff=1; keep Friction FF disabled.

Sections 4.5.1–4.5.4 give the piecewise position equations, coefficients, and UI parameter mapping.

Step or Ramp? Speed profile does not apply in this experiment; leave its default Step selected. This option determines Step or Ramp commands only in Speed Control; Position Mode does not read it. Even if shown in the current UI, choosing Ramp does not change this position trajectory. Leaving Step selected does not make position reference jump directly to 1 rad. The target first passes through the trajectory generator, which automatically produces a triangular or trapezoidal velocity profile from displacement, maximum velocity, and acceleration.

Trajectory timing: ta is acceleration time, also used for symmetric deceleration; tc is cruise time; Tmove=2ta+tc is reference motion duration, not total simulation Duration. These settings give ta=0.1 s, tc=0, and Tmove=0.2 s, a triangular velocity profile: start at 0.02 s, peak velocity at 0.12 s, and reference completion at 0.22 s. Use waveforms to determine whether the actual motor has reached and settled at the target.

Where is this in the UI? Select Position Control, then Trajectory Generator to set maximum velocity and acceleration. Below the parameter panel, trajectory type and duration=0.200000 s (Tmove) are shown; separate numerical ta/tc values are not currently displayed. Under Control response, inspect position reference θref, velocity reference ωref, and acceleration reference αref.

Procedure: First run with Velocity FF and Acceleration FF both off; then enable Velocity FF only; finally enable both. Keep other parameters identical. Compare position following error, ωref/ωcmd/ωm, and torque contributions.

Expected: Suitable FF can reduce tracking error during motion. Its effectiveness depends on the model, limits, and discrete timing; arbitrary gains do not guarantee improvement. The difference between Position P raw and

With velocity FF should equal enabled Kvff·ωref.

Speed-limit extension: Reduce Position speed limit to 3 rad/s. Observe increased trajectory error after the combined speed is limited. Planned ωref is not automatically replanned by this downstream limit.

Experiment 6: Sampling Delay, Dead Time, and Bode Analysis

Purpose: Connect the digital model to time-domain waveforms.

Settings: Return to Experiment 1, retain 10 kHz, and inspect final high-speed plots and the timing cursor.

Procedure:

1. Compare the ADC valley, pending CCR, and active CCR at the next valley to confirm the one-period delay.

2. Run with dead time on and off; compare ΔV, q I state, and phase-current ripple. At locked-rotor θe=0, do not necessarily expect the sinusoidal zero-crossing distortion typical of rotation.

3. Record current Bode bandwidth and PM. Lower PWM frequency, rerun, and update Bode to inspect Ts and delay effects.

4. In Speed Mode, compare PM/bandwidth across the three Bode models; the time simulation engine continues using the same actual model.

Questions: Why do identical PI settings yield different margins in the ideal continuous model and the model with z^{-1} delay? Why can increasing ARR not replace reducing the plant integration step?

10. Data Export, FAQ, and Experiment Records

10.1 Using CSV

The exported file is motor-lab-results.csv. Initial # lines contain JSON parameters and metadata, followed by column headers. The first column, dataset, distinguishes slow/fast. Skip comment lines and group by dataset when importing into analysis tools. Do not concatenate both groups as a monotonic time sequence: the final high-speed window overlaps the slow data.

t is in seconds; ms/μs on plots are display conversions. rpm/ref are in rpm; speedcmd, velocityref, and omegafb in rad/s; theta_m/positionref in rad. sat is the held voltage-vector-limit flag. Metadata saturation fraction uses the complete control-update count; averaging downsampled slow sat values need not reproduce it.

da/db/dc and ccra/b/c are active duty/CCR, not freshly computed pending values. Complete control-period pending/active data remain in the page result's timing structure; CSV does not separately export the timing array.

10.2 Frequently Asked Questions

Observation	Check first
iq reaches its command, then falls	Free Running? Are speed, qff, and voltage vector demand increasing?
iq fails to track with rotor locked	Vdc, Rs·iq demand, modulation limit, PI, q-command limit, and numerical resolution
Speed AW is zero but current does not track	Outer command limits may be inactive while the inner loop is voltage-saturated; speed AW does not track actual iq directly
I term and PI raw do not add up	Use pre-update di/qi/speedI; speedIntegral is the updated state
Position is not settled by 0.25 s	When does the trajectory finish? Is Duration sufficient? Are speed/torque/voltage limits active?
Waveforms unchanged after editing	Was the simulation rerun? Is Bode using live settings or the result snapshot?
θe suddenly resets to zero	Normal wrap; inspect continuous θm for actual position
Good Bode but poor large-step response	Small-signal analysis omits saturation, AW, switching, and nonlinearities; inspect actual limits
Numerical divergence warning	Check gains, step size, and extreme parameters; numerical failure alone is not a physical conclusion
Engine/plots fail to load	Check network, CDN, Web Worker, and browser errors; the first load is not fully offline

10.3 Experiment Report Template

Item	Entry
Experiment name / date / page version	
Control Mode / Mechanical Condition	
Target, Start, Duration, trajectory settings	
Rs, Ld, Lq, ψf, p, J, B	
Vdc, m, dead time, actual PWM, ARR, Ts, Tω, plant step	
Each loop's Kp, Ki, Kaw, AW/decoupling/FF switches	
Current/torque/speed limits and load	
Observed waveforms, CSV filename, analysis snapshot	
Final-segment mean error, ripple, saturation fraction	
Feedback BW / Command BW / PM / GM and model selection	
Observed causal relationships, model assumptions, and limitations	

The software does not model magnetic saturation, temperature rise, complete losses, hardware protection, or nonideal position sensing. Parameter experiments support understanding of control and numerical models; current command limiting is not a complete hardware protection model.

MotorControl Lab | © 2026 Dr. Jack Yeh | All rights reserved.

11. Appendix: Digital Speed and Position Models

This section supports readers comparing the page's bodeModel. It precisely describes nominal digital analysis, not a separate switching model.

11.1 Combined Discrete Current and Mechanical Model

Under ideal decoupling and fixed id0, let continuous plant state y=[iq,ωm]☐:

$$\dot{y} = A_c y + B_c v_q, \quad A_c = \begin{bmatrix} -R_s/L_q & 0 \\ K_\tau/J & -B/J \end{bmatrix}, \quad B_c = \begin{bmatrix} 1/L_q \\ 0 \end{bmatrix}.$$

Exact ZOH discretization:

$$\Phi = e^{A_c T_s}, \qquad \Gamma = \int_0^{T_s} e^{A_c \tau} B_c \, d\tau.$$

Including the PI integrator and active voltage, state x=[iq,ωm,ξq,vactive]☐; input r is a torque command in N·m:

$$x[k+1] = Ax[k] + B_r r[k],$$

$$A = \begin{bmatrix} \Phi_{11} & \Phi_{12} & 0 & \Gamma_1 \\ \Phi_{21} & \Phi_{22} & 0 & \Gamma_2 \\ -K_{iq}T_s & 0 & 1 & 0 \\ -K_{pq} & 0 & 1 & 0 \end{bmatrix}, \quad B_r = \begin{bmatrix} 0 \\ 0 \\ K_{iq}T_s/K_t \\ K_{pq}/K_t \end{bmatrix}.$$

The last row means new voltage becomes active in the next period, already including current computation delay. If the outer loop updates every M current updates, the torque command is held across those M updates:

$$A_M = A^M, \qquad B_M = \sum_{k=0}^{M-1} A^k B_r, \qquad F_\omega(z_\omega) = c_\omega (z_\omega I - A_M)^{-1} B_M,$$

Here cω=[0,1,0,0] and zω=exp(j2πfTω).

$$C_\omega(z_\omega) = K_{p\omega} + \frac{K_{i\omega} T_\omega}{z_\omega - 1}, \quad L_\omega(z_\omega) = C_\omega(z_\omega) F_\omega(z_\omega).$$

11.2 Position State and Closed Speed Loop

Add θm to form continuous plant state [iq,ωm,θm], with θm=ωm. The same matrix exponential gives exact position integration over one Ts. Combine this with the current PI integrator and active voltage into five-state xp=[iq,ωm,ξq,vactive,θm]☐, then lift to the outer-loop period:

$$x_p[n+1] = F_p x_p[n] + G_p T^*[n].$$

Let cω=[0,1,0,0,0] and cθ=[0,0,0,0,1]. After adding Speed PI integrator ξω, the closed speed loop has the six-state matrix:

$$A_{cl} = \begin{bmatrix} F_p - G_p K_{p\omega} c_\omega & G_p \\ -K_{i\omega} T_\omega c_\omega & 1 \end{bmatrix}, \quad B_{cl} = \begin{bmatrix} G_p K_{p\omega} \\ K_{i\omega} T_\omega \end{bmatrix}.$$

Speed command to position:

$$H_\theta(z_\omega) = [c_\theta \; 0](z_\omega I - A_{cl})^{-1} B_{cl}, \quad L_\theta(z_\omega) = K_{p,pos} H_\theta(z_\omega).$$

$$T_\theta(z_\omega) = \frac{L_\theta(z_\omega)}{1 + L_\theta(z_\omega)}.$$

The page therefore integrates the held-voltage plant's position state exactly rather than arbitrarily appending an approximate 1/s to the digital speed closed loop. Digital Command Tracking adds FF through smooth-trajectory analytical derivatives jω and (jω)², not backward-difference approximations. Limiting and the nonlinear trajectory generator remain excluded.

11.3 Version and Verification Scope

Symbol clarification revision dated 2026-10-23: Section 4.1 completes Kt/Kτ definitions, units, and fixed-d-axis conditions; Section 5.4 expands Tq(s) explicitly and adds Figure 09. Simulation code and numerical equations were not changed.

Diagram revision: ten SVG figures share one visual style, including Figure 09's torque path and Figure 10's ZOH/RL equivalent model. Current/Speed/Position diagrams present the actual computational paths hierarchically, whereas Figure 06 deliberately shows a nominal unsaturated model without feedforward. Read them with the relevant sections because they serve different purposes.

This manual is based on local source code, control equations, Bode implementation, waveform fields, and existing model regression results as of the dates above. The simulator was not changed to write this manual. Document verification does not constitute new real-browser or hardware validation. Update the manual alongside future changes to control paths, defaults, or data fields.