PID Tuning Cheatsheet for a Servo with a Fast Current (Torque) Loop

Goal: Practical recipe to pick **PID** gains for a **single outer position loop** when the drive's **current loop is fast** (~20 kHz), so torque proportional to current command.

1) Assumptions & Symbols

- Inner current loop tightly closed → torque follows command:

$$\tau(t) \approx K_t \, i_q^*(t)$$

- Use an effective torque gain (K_\tau) (set (K_\tau=K_t) if your controller outputs current; otherwise fold it into PID gains).
- Mechanical plant about motor shaft (load reflected): inertia (J), viscous damping (b).
- Laplace variable (s), position (\Theta(s)), speed (\Omega(s)).
- Mechanical pole (no control):

$$\omega_m = \frac{b}{J} \quad \text{[rad/s]}$$

2) Plant and Loop Definitions

Torque → motion:

$$G_{\omega\tau}(s) = \frac{\Omega(s)}{\tau(s)} = \frac{1}{Js+b}, \qquad G_{\theta\tau}(s) = \frac{\Theta(s)}{\tau(s)} = \frac{1}{s(Js+b)}.$$

Controller (PID):

Ideal form

$$C_{\text{ideal}}(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}.$$

Practical derivative (noise-friendly)

$$C(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{1+\tau_d s}.$$

Open loop (unity position feedback):

$$L(s) = C(s)\,K_\tau\,G_{\theta\tau}(s).$$

Closed-loop complementary sensitivity (command → position):

$$T(s) = \frac{\Theta(s)}{R(s)} = \frac{L(s)}{1 + L(s)}.$$

Sensitivity (disturbance rejection; error to reference):

$$S(s) = \frac{1}{1 + L(s)}.$$

With ideal PID, these expand to

$$L_{\text{ideal}}(s) = \frac{K_\tau\big(K_d s^2 + K_p s + K_i\big)}{J s^3 + b s^2},$$

$$T_{\text{ideal}}(s) = \frac{K_\tau\big(K_d s^2 + K_p s + K_i\big)}{J s^3 + (b + K_\tau K_d) s^2 + (K_\tau K_p) s + (K_\tau K_i)}.$$

Stability margins (from Bode of (L(j\omega))):

$$\text{PM} = 180^\circ + \angle L(j\omega_{gc}), \qquad \text{GM}_{\text{dB}} = -20\log_{10}|L(j\omega_{pc})|.$$

Recommended: PM   45–60°, GM   6–10 dB.

3) One-Shot Tuning Recipe (Position Loop Only)

Shapes the loop to look like (\sim K/s) at crossover (good damping).

A. “Velocity feedback”: place controller zero near the mechanical pole

$$T_d = \frac{J}{b}, \qquad K_d = K_p\,T_d.$$

B. Pick target crossover (\omega_c). Start with (\omega_c\in[0.5„2]\omega_m).

C. Proportional gain (unity-gain at crossover)

$$\boxed{K_p = \frac{b\,\omega_c}{K_\tau}}.$$

D. Add slow integral (remove steady-state error)

$$T_i \in \left[\frac{8}{\omega_c}, \frac{16}{\omega_c}\right], \qquad \boxed{K_i = \frac{K_p}{T_i}}.$$

E. Filter derivative

$$\boxed{\tau_d \approx \frac{1}{10\,\omega_c}}.$$

Summary (compute in this order):

1. (T_d=J/b)
2. choose (\omega_c)
3. (K_p=b,\omega_c/K_\tau)
4. (K_d=K_p,T_d)
5. choose (T_i\in[8,16]/\omega_c) and set (K_i=K_p/T_i)
6. (\tau_d\approx 1/(10,\omega_c))

4) How to Verify

- **Bode of (L)** → read PM/GM.
- **Step of (T)** → rise/overshoot/settling; integrator zero steady-state error.
- **Bode of (S)** → low-frequency (|S|) small is good.
- **Control effort** (C/(1+L)) → check torque limits; add anti-windup if needed.

5) Worked Example

Given (J=0.01) $\mathrm{kg \cdot m^2}$, (b=0.001) $\mathrm{N \cdot m \cdot s/rad}$, (K_\tau=K_t=0.10) $\mathrm{N \cdot m/A}$:

- (\omega_m=b/J=0.1) rad/s
- choose (\omega_c=1.0) rad/s
- (T_d=J/b=10) s
- (K_p=b,\omega_c/K_\tau=0.010)
- (K_d=K_p T_d=0.10)
- (T_i=8/\omega_c=8) s (K_i=1.25\times10^{-3})
- (\tau_d\approx 1/(10\omega_c)=0.10) s

6) Python Snippet (Step & Bode)

```python
import control as ctl, matplotlib.pyplot as plt, math

# Example parameters
J, b, K_tau = 0.01, 0.001, 0.10
omega_c = 1.0
T_d = J/b
Kp = b*omega_c/K_tau
Kd = Kp*T_d
Ti = 8/omega_c
Ki = Kp/Ti
```

```
tau_d = 1/(10*omega_c)

s = ctl.TransferFunction.s
G = 1/(s*(J*s+b))
C = Kp + Ki/s + (Kd*s)/(1+tau_d*s)

L = C*K_tau*G
T = ctl.feedback(L,1)
S = 1/(1+L)

gm, pm, wg, wp = ctl.margin(L)
print(f"GM={gm:.2g} ({20*math.log10(gm):.1f} dB) @ {wg:.3g} rad/s, PM={pm:.1f}° @ {wp:.3g} r

t, y = ctl.step_response(T)
plt.figure(); plt.plot(t,y); plt.grid(True); plt.title("Closed-loop Step"); plt.xlabel("Time
plt.figure(); ctl.bode(L, dB=True); plt.suptitle("Open-loop Bode"); plt.show()
```

7) Practical Notes

- Too slow? Increase (\omega_c) → raise (K_p); keep (T_d=J/b), reduce (T_i); keep (\tau_d\approx 1/(10,\omega_c)).
- Ringing / low PM? Lower (\omega_c) or increase phase lead slightly (increase (T_d)); if integral is aggressive, increase (T_i).
- Noisy control? Increase (\tau_d) or lower (\omega_c).
- Friction (Coulomb/stiction): treat as disturbance; stronger integral helps but watch overshoot/windup.
- Gearing & load reflection: if gear ratio (N=\omega_m/\omega_L),

$$J = J_m + N^2 J_L, \qquad b = b_m + N^2 b_L, \qquad \theta_L = \theta_m / N.$$

- Resonances/compliance: consider notch filters or add an inner velocity loop.
- Anti-windup: clamp integrator or back-calculate when torque saturates.