PID Tuning Rubric — MARS Hexapod v2.0

Target: Teensy 4.1 controller with sparse-feedback PID per joint, estimator-based error, dt-aware PID, and shadow mode (`pid.mode=shadow`).

1. Preparation

Goal: Create a repeatable, safe environment for tuning.

1. **Firmware/config setup**
   - Ensure firmware has:
     - Estimator + filtered D enabled.
     - dt-aware PID (uses loop `dt` from `LoopTimer`).
   - In `/config.txt` (or via PID commands), set:
     - `pid.enabled=false`
     - `pid.mode=shadow`
     - `pid.kp_milli.<coxa|femur|tibia>=0`
     - `pid.ki_milli.<coxa|femur|tibia>=0`
     - `pid.kd_milli.<coxa|femur|tibia>=0`
     - `pid.kd_alpha_milli.<coxa|femur|tibia>=200`
     - `pid.shadow_report_hz=2` (or a modest value 1–5 Hz)
2. **Operating scenario**
   - Use a simple, repeatable motion:
     - **Static stance**: `STAND` at a neutral pose; or
     - **Slow tripod gait**: `MODE TEST` with conservative parameters.
   - Disable or throttle SD logging to avoid timing interference while tuning.
3. **Safety checks**
   - Confirm:
     - `safety.soft_limits=true`
     - `safety.collision=true`
     - Over-temp lockout behaves as expected.
   - Keep an emergency `DISABLE` handy during active-mode tests.

---

2. Baseline (PID Off)

Objective: Understand the controller behavior without PID corrections.

1. **Configuration**
   - `PID DISABLE`
   - `PID MODE SHADOW`
   - Keep K gains at 0 (P/I/D all zero).
2. **Observe**
   - `PID_SHADOW` lines:
     - With PID off, `diff_cd` should be ~0 (PID and base targets match).

- `STATUS [TIMING]`:
  - Ensure tick time and jitter are comfortably below the 6.024 ms budget.
- Physical behavior:
  - Evaluate how stiff/sloppy the legs feel under STAND or slow TEST gait.

This provides the reference you're trying to improve upon with PID.

---

3. Tune P (Proportional) Only

Objective: Increase stiffness and responsiveness without inducing oscillation.

1. **Enable PID with P only**
   - `PID ENABLE`
   - Set Kp:
     - Start small per joint group:
       - Coxa: `PID KP COXA 100–200` (milli)
       - Femur: `PID KP FEMUR 200–400`
       - Tibia: `PID KP TIBIA 200–400`
     - Or use `PID KP ALL <value>` if you want a uniform starting point.
   - Keep:
     - `PID KI ... 0`
     - `PID KD ... 0`
2. **Stay in shadow mode initially**
   - `PID MODE SHADOW`
   - Monitor `PID_SHADOW`:
     - `err_cd` (target - estimate) per leg/joint.
     - `diff_cd` (pid - base) per leg/joint.
3. **Interpretation rubric for P**
   - **Too low Kp**
     - `diff_cd` small.
     - Errors decay slowly; behavior close to baseline.
   - **Good Kp**
     - `diff_cd` moderate and roughly proportional to `err_cd`.
     - In shadow mode, corrections look sensible (no wild sign flips).
   - **Too high Kp**
     - `diff_cd` large; may alternate sign rapidly.
     - Indicates potential overshoot/oscillation if applied.
4. **Active-mode test**
   - Once shadow behavior looks reasonable:
     - `PID MODE ACTIVE`
   - Evaluate:
     - Leg stiffness increases appropriately.

- No high-frequency buzzing or visible oscillation.
- `STATUS [TIMING]` remains within budget.

Rule of thumb: Increase Kp until you see the first sign of oscillation in shadow mode, then reduce Kp by about 30–50%.

---

4. Add D (Derivative) for Damping

Objective: Reduce overshoot and damp oscillations introduced by P.

1. **Initial D settings**
   - With Kp fixed from the previous step:
     - Coxa: `PID KD COXA 50–100`
     - Femur: `PID KD FEMUR 100–200`
     - Tibia: `PID KD TIBIA 100–200`
   - Derivative smoothing:
     - `PID KDALPHA ALL 200` (0..1000; higher = faster tracking, lower = more smoothing).
2. **Tune in shadow mode**
   - `PID MODE SHADOW`
   - For small steps or gait transitions, observe:
     - `diff_cd` behavior around direction changes.
     - D should:
       * Reduce overshoot in the hypothetical PID target.
       * Make `diff_cd` less "spiky‘.
3. **Adjustments**
   - If `diff_cd` looks noisy:
     - Lower (e.g., 150 or 100) for more smoothing; or reduce `KD`.
   - If response feels sluggish in shadow:
     - Slightly raise `KD` or increase .
4. **Active-mode verification**
   - `PID MODE ACTIVE`
   - Check:
     - Movements are smooth with less overshoot.
     - No "ringing" after leg motions.
     - Timing remains within limits.

Rubric: - **Too little D:** overshoot and ringing after movements. - **Good D:** crisp motion, minimal overshoot, quick settling. - **Too much D:** sluggish response; legs resist motion too strongly and correct slowly.

---

5. Add I (Integral) for Steady-State Accuracy

Objective: Remove persistent steady-state error (e.g., gravity sag).

1. **Start very conservatively**
   - With Kp and Kd tuned:
     - Set KI small:
       * Coxa: `PID KI COXA 0–10`
       * Femur: `PID KI FEMUR 10–30`
       * Tibia: `PID KI TIBIA 10–30`
2. **Static test (STAND)**
   - Let the robot stand in a neutral pose for several seconds.
   - Observe:
     - Does the leg settle closer to the commanded angles?
     - Does `diff_cd` drift slowly in a direction that reduces `err_cd`?
3. **Watch for integral windup behavior**
   - Signs of too much I:
     - Slow, large-amplitude oscillations in joint angles.
     - `diff_cd` continues to grow even when `err_cd` is small or reversed.
   - Mitigation:
     - Reduce KI.
     - Confirm integral clamps are effective (no runaway).

Rubric: - **No I (KI=0):** steady-state error remains (leg sags a bit). - **Good I:** error diminishes near zero without introducing slow oscillations. - **Too much I:** system "hunts" around the target with slow big swings.

---

6. Scenario Validation

Run through multiple scenarios with final candidate gains:

1. **Static stance**
   - `STAND`:
     - Legs hold position with minimal drift.
     - No visible jitter or micro-oscillations.
2. **Slow tripod gait**
   - `MODE TEST`:
     - Smooth transitions between stance and swing.
     - No jerk at lift/landing.
     - `STATUS [TIMING]` shows no persistent overruns.
3. **Disturbance test**
   - While in STAND, gently push the body or a tibia:
     - Robot should resist and return smoothly to target.
     - Response should be firm but not violently stiff.
4. **Temperature / logging check**
   - Ensure PID behavior remains stable under:
     - Logging enabled at realistic rates.
     - Extended runtime (thermal conditions).

---

7. Finalization and Documentation

1. **Persist tuned gains**
   - Use PID commands that update `/config.txt`:
     - `PID KP ...`
     - `PID KI ...`
     - `PID KD ...`
     - `PID KDALPHA ...`
   - Confirm `/config.txt` reflects your tuned values on reboot.
2. **Record a snapshot**
   - Save:
     - Tuned Kp/Ki/Kd/kd_alpha per joint.
     - Representative `STATUS [TIMING]` output.
     - Sample `PID_SHADOW` lines for:
       * STAND
       * TEST gait
   - Optionally store notes in `docs/PROJECT_SPEC.md` or a dedicated tuning document.
3. **Operational defaults**
   - Decide:
     - Normal operating mode (`pid.mode=active` or `shadow` for further validation).
     - Default `pid.enabled` at boot (probably `false` until PID is fully vetted on hardware).

Quick Reference (Starting Gains)

These are only **starting points**; tune using the rubric above:

- **Coxa**
  - Kp 100–200
  - Ki 0–10
  - Kd 50–100
- **Femur**
  - Kp 200–400
  - Ki 10–30
  - Kd 100–200
- **Tibia**
  - Kp 200–400
  - Ki 10–30
  - Kd 100–200
- **Derivative smoothing**
  - `kd_alpha_milli` 200 as a starting point.