

-->

FFB Mathematical Formulas (v0.6.2)

⚠ API Source of Truth

All telemetry data units and field names are defined in `src/lmu_sm_interface/InternalsPlugin.hpp`.

Critical: `mSteeringShaftTorque` is in Newton-meters (Nm).

The final output sent to the DirectInput driver is a normalized value between -1.0 and 1.0.

1. The Master Formula

$$F_{\text{final}} = \text{Clamp}(\text{Normalize}(F_{\text{total}}) \times K_{\text{gain}}, -1.0, 1.0)$$

where normalization divides by `m_max_torque_ref` (with a floor of 1.0 Nm).

$$F_{\text{total}} = (F_{\text{base}} + F_{\text{sop}} + F_{\text{vib-lock}} + F_{\text{vib-spin}} + F_{\text{vib-slide}} + F_{\text{vib-road}} + F_{\text{vib-bottom}} + F_{\text{gyro}} + F_{\text{abs}})$$

2. Signal Scalers (Decoupling)

To ensure consistent feel across different wheels (e.g. G29 vs Simucube), effect intensities are automatically scaled based on the user's `Max Torque Ref`.

- Reference Torque: 20.0 Nm.
- Decoupling Scale: `K_decouple = m_max_torque_ref / 20.0`.
- Note: This ensures that 10% road texture feels the same physical intensity regardless of wheel strength.

3. Component Breakdown

A. Load Factors (Safe Caps)

Texture and vibration effects are scaled by tire load to simulate connection with the road. To prevent dangerous force spikes during glitches or aero anomalies, "Load Factors" are clamped.

1. Texture Load Factor (Road/Slide):

- $F_{\text{load-texture}} = \text{Clamp}(\text{Load}/4000.0, 0.0, m_{\text{texture-load-cap}})$
- Max Cap: 2.0.

2. Brake Load Factor (Lockup):

- $F_{\text{load-brake}} = \text{Clamp}(\text{Load}/4000.0, 0.0, m_{\text{brake-load-cap}})$

- Max Cap: 3.0. (Allows stronger vibration under heavy braking).

B. Base Force (Understeer / Grip Modulation)

Modulates the raw steering torque (`mSteeringShaftTorque`) based on front tire grip.

$$F_{\text{base}} = (T_{\text{shaft}} \times K_{\text{shaft-smooth}}) \times K_{\text{shaft-gain}} \times (1.0 - (\text{GripLoss} \times K_{\text{understeer}}))$$

- **Steering Shaft Smoothing:** Time-Corrected LPF ($\tau = m_{\text{shaft-smooth}}$) applied to raw torque.
- **Grip Approximation:** If telemetry grip is missing, it is estimated from **Slip Angle** (α) and **Slip Ratio** (κ) using a **Combined Friction Circle**:
 - Metric = $\sqrt{(\alpha/\text{OptAlpha})^2 + (\kappa/\text{OptRatio})^2}$
 - Used only when valid suspension force is present (to avoid airborne glitches).

C. Seat of Pants (SoP) & Oversteer

Simulates chassis rotation and weight transfer.

1. Lateral G Force:

$$F_{\text{sop-base}} = G_{\text{smooth}} \times K_{\text{sop}} \times K_{\text{sop-scale}} \times K_{\text{decouple}}$$

- G_{smooth} : Time-Corrected LPF of `mLocalAccel.x`.
- **Inversion:** Removed in v0.4.30 (Signal is physically correct).

2. Yaw Acceleration (The Kick):

$$F_{\text{yaw}} = -\text{YawAccel}_{\text{smooth}} \times K_{\text{yaw}} \times 5.0\text{Nm} \times K_{\text{decouple}}$$

- **Filters:** Low Speed Cutoff (< 5m/s), Noise Gate (< 0.2 rad/s²), and LPF ($\tau = 10\text{ms}$).
- **Physics:** Provides a predictive "kick" counter to rotation.

3. Rear Aligning Torque:

- Approximated as `SlipAngle_Rear * RearLoad * 15.0`.
- $T_{\text{rear}} = -F_{\text{lat-rear}} \times 0.001 \times K_{\text{rear}} \times K_{\text{decouple}}$.
- **Sign:** Negative to provide stabilizing (counter-steering) torque.

4. Lateral G Boost (Slide):

- Amplifies $F_{\text{sop-base}}$ when `FrontGrip > RearGrip` (Oversteer).

D. Braking & Lockup (Advanced)

1. Progressive Lockup ($F_{\text{vib-lock}}$)

- Triggers: Brake > 2% AND Load > 50N .
- Predictive Logic (v0.6.0):
 - Monitors wheel angular deceleration vs. car deceleration.
 - If WheelDecel > CarDecel * 2.0 (rapid slowing), the trigger threshold drops from Full% to Start% .
 - **Bump Rejection:** disabled if SuspVelocity > m_lockup_bump_reject .
- Severity:
 - NormSlip = (Slip - Start%) / (Full% - Start%)
 - Severity = pow(NormSlip, m_lockup_gamma) (Quadratic Response).
- Axle Differentiation:
 - Front: Pitch ≈ 10 – 60Hz.
 - Rear: Pitch × 0.3 ("Heavy Judder") + 1.5x Amplitude Boost.
- Pressure Scaling:
 - Intensity scales with mBrakePressure (Bar).
 - Fallback: If Pressure < 0.1 but Slip > 50% (Engine Lock), intensity = 50%.
- Force: Severity * PressureFactor * K_lockup * 4.0Nm * Scale * sin(Phase) .

2. ABS Pulse (F_{abs})

- Trigger: Brake > 50% AND PressureModulation > 2.0 bar/s .
- Output: 20Hz Sine Wave pulse.
- Force: sin(20Hz) * K_abs * 2.0Nm .

E. Dynamic Textures

1. Slide Texture (Scrubbing)

- Scope: Uses Max(FrontSlipVel, RearSlipVel) (Detects both Understeer and Drifts).
- Frequency: 10Hz + (SlipVel × 5.0). Cap 250Hz.
- Work-Based Scrubbing: Amplitude scales with (1.0 - Grip) * Load .
- Force: Sawtooth(Phase) * K_slide * 1.5Nm * Scale .

2. Road Texture

- **Input:** Delta of `mVerticalTireDeflection`.
- **Scrub Drag:** Constant resistance force opposing lateral slide (`DragDir * K_drag`).
- **Force:** `(DeltaL + DeltaR) * 50.0 * K_road * F_{\text{load-texture}} * Scale`.

3. Suspension Bottoming

- **Method A:** `RideHeight < 2mm`.
- **Method B:** `SuspensionForceRate > 100kN/s` (Impact).
- **Force:** `sin(50Hz) * K_bottom * 1.0Nm`.

F. Gyroscopic Damping (F_{gyro})

- Simulates steering weight at speed.
- Force opposes `SteeringVelocity` (Smoothed).
- $F_{\text{gyro}} = -\text{Vel}_{\text{steer}} \times K_{\text{gyro}} \times (\text{Speed}/10.0) \times 1.0\text{Nm}$.

4. Signal Filtering & Conditioning

1. Frequency Estimator:

- Analyzes `mSteeringShaftTorque` zero-crossings to estimate vibration frequency.

2. Dynamic Notch Filter (Flatspot Suppression):

- Calculates `WheelFreq = Speed / Circumference`.
- Applies Biquad Notch Filter at `WheelFreq`.
- Blends raw/filtered signal based on `m_flatspot_strength`.

3. Static Notch Filter:

- Fixed frequency notch (e.g. 50Hz) to remove specific hardware resonances.

5. Legend: Physics Reference Values (at Gain 1.0)

- **Lockup:** 4.0 Nm
- **Spin:** 2.5 Nm
- **Slide:** 1.5 Nm

- **Road:** 2.5 Nm
- **Scrub Drag:** 5.0 Nm
- **Yaw Kick:** 5.0 Nm
- **Rear Align:** 3.0 Nm