-->

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

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

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

The total force is a summation of base physics, seat-of-pants effects, and dynamic vibrations, scaled by the **Traction Loss Multiplier**:

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

Note: $M_{\text{spin-drop}}$ reduces total force implementation during wheel spin (see Section E.3).

---

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. (Updated from legacy 4000 unitless reference).
- **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 normalized tire load (`Load / 4000N`) to simulate connection with the road.

1. **Texture Load Factor (Road/Slide)**:

   - **Input**: `AvgLoad = (FL.Load + FR.Load) / 2.0`.
   - **Robustness Check**: Uses a hysteresis counter; if `AvgLoad < 1.0` while `|Velocity| > 1.0 m/s`, it defaults to **4000N** (1.0 Load Factor) to prevent signal loss during telemetry glitches.
   - $F_{\text{load-texture}} = \text{Clamp}(\text{AvgLoad}/4000.0, 0.0, m_{\text{texture-load-cap}})$
   - **Max Cap**: 2.0.

2. **Brake Load Factor (Lockup)**:

  - $F_{\text{load-brake}} = \text{Clamp}(\text{AvgLoad}/4000.0, 0.0, m_{\text{brake-load-cap}})$
  - **Max Cap**: 3.0 (Allows stronger vibration under high-downforce braking).

B. Base Force Components

1. Base Force Calculation (F_{base})
Modulates the raw steering torque (`mSteeringShaftTorque`) based on front tire grip.

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

- **Operation Modes ($m_{\text{base-force-mode}}$):**
  - **Mode 0 (Native)**: $\text{BaseInput} = T_{\text{shaft}}$. (Default precision mode).
  - **Mode 1 (Synthetic)**: $\text{BaseInput} = \text{Sign}(T_{\text{shaft}}) \times m_{\text{max-torque-ref}}$.
    - Used for debugging direction only.
    - **Deadzone**: Applied if $|T_{\text{shaft}}| < 0.5\text{Nm}$ to prevent center oscillation.
  - **Mode 2 (Muted)**: $\text{BaseInput} = 0.0$.
- **Steering Shaft Smoothing**: Time-Corrected LPF ($\tau = m_{\text{shaft-smooth}}$) applied to raw torque.

2. Grip Estimation & Fallbacks
If telemetry grip (`mGripFract`) is missing or invalid (< 0.0001), the engine approximates it:

- **Low Speed Trap**: If `CarSpeed < 5.0 m/s` , `Grip` is forced to **1.0** (Prevents singularities at parking speeds).
- **Combined Friction Circle**:
  - **Metric Formulation**:
    - $\text{Metric}_{\text{lat}} = |\alpha|/\text{OptAlpha}$ (Lateral Slip Angle). **Default**: 0.10 rad.
    - $\text{Metric}_{\text{long}} = |\kappa|/\text{OptRatio}$ (Longitudinal Slip Ratio). **Default**: 0.12 (12%).
  - $\text{Combined} = \sqrt{\text{Metric}_{\text{lat}}^2 + \text{Metric}_{\text{long}}^2}$
  - $\text{ApproxGrip} = (1.0 \text{ if Combined} < 1.0 \text{ else } 1.0/(1.0 + (\text{Combined} - 1.0) \times 2.0))$
- **Safety Clamp**: Approx Grip is usually clamped to min **0.2** to prevent total loss of force.

3. Kinematic Load Reconstruction
If `mSuspForce` is missing (encrypted content), tire load is estimated from chassis physics:

- $$F_z = F_{\text{static}} + F_{\text{aero}} + F_{\text{long-transfer}} + F_{\text{lat-transfer}}$$
- **Static**: Mass (1100kg default) distributed by Weight Bias (55% Rear).
- **Aero**: $2.0 \times \text{Velocity}^2$.
- **Transfer**:
  - Longitudinal: $(\text{Accel}_Z/9.81) \times 2000.0$.

- Lateral: $(\text{Accel}_X/9.81) \times 2000.0 \times 0.6$ (Roll Stiffness).

C. Seat of Pants (SoP) & Oversteer

1. **Lateral G Force** ($F_{\text{sop-base}}$):
    - **Input**: `mLocalAccel.x` (Clamped to **+/- 5.0 G**).
    - **Smoothing**: Time-Corrected LPF ($\tau \approx 0.0225 - 0.1$s mapped from scalar).
    - **Formula**: $G_{\text{smooth}} \times K_{\text{sop}} \times K_{\text{sop-scale}} \times K_{\text{decouple}}$.
2. **Lateral G Boost** (F_{boost}):
    - Amplifies the SoP force when the car is oversteering (Front Grip > Rear Grip).
    - **Condition**: `if (FrontGrip > RearGrip)`
    - **Formula**: `SoP_Total *= (1.0 + ((FrontGrip - RearGrip) * K_oversteer_boost * 2.0))`
3. **Yaw Acceleration ("The Kick")**:
    - **Input**: `mLocalRotAccel.y` (rad/s²). **Note**: Inverted (-1.0) to comply with SDK requirement to negate rotation data.
    - **Conditioning**:
        - **Low Speed Cutoff**: 0.0 if Speed < 5.0 m/s.
        - **Noise Gate**: 0.0 if $|Accel| < 0.2$ rad/s².
    - **Logic**: Provides a "predictive kick" when car rotation starts, before lateral G builds up.
    - **Formula**: $-\text{YawAccel}_{\text{smooth}} \times K_{\text{yaw}} \times 5.0\text{Nm} \times K_{\text{decouple}}$.
    - **Note**: Negative sign provides counter-steering torque.
4. **Rear Aligning Torque** (T_{rear}):
    - **Workaround**: Uses `RearSlipAngle * RearLoad * Stiffness(15.0)` to estimate lateral force.
    - **Derivation**: `RearLoad = SuspForce + 300.0` (where 300N is estimated Unsprung Mass).
    - **Formula**: $-F_{\text{lat-rear}} \times 0.001 \times K_{\text{rear}} \times K_{\text{decouple}}$.
    - **Clamp**: Lateral Force clamped to **+/- 6000N**.

D. Braking & Lockup (Advanced)

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

- **Safety Trap**: If `CarSpeed < 2.0 m/s`, Slip Ratio is forced to 0.0 to prevent false lockup detection at standstill.
- **Predictive Logic (v0.6.0)**: Triggers early if `WheelDecel > CarDecel * 2.0` (Wheel stopping faster than car).
- **Bump Rejection**: Logic disabled if `SuspVelocity > m_lockup_bump_reject` (e.g. 1.0 m/s).
- **Severity**: $\text{Severity} = \text{pow}(\text{NormSlip}, m_{\text{lockup-gamma}})$ (Quadratic).
- **Logic**:

    - **Axle Diff**: Rear lockups use **0.3x Frequency** and **1.5x Amplitude**.
    - **Pressure Scaling**: Scales with Brake Pressure (Bar). Fallback to 0.5 if engine braking (Pressure < 0.1 bar).
- **Oscillator**: `sin(Phase)` (Wrapped via `fmod` to prevent phase explosion on stutter).

2. ABS Pulse (F_{abs})

- **Trigger**: Brake > 50% AND Pressure Modulation Rate > 2.0 bar/s.
- **Formula**: `sin(20Hz) * K_abs * 2.0Nm`.

E. Dynamic Textures & Vibrations

1. Slide Texture (Scrubbing)

- **Scope**: `Max(FrontSlipVel, RearSlipVel)` (Worst axle dominates).
- **Frequency**: $10\text{Hz} + (\text{SlipVel} \times 5.0)$. Cap 250Hz. (Updated from old 40Hz base).
- **Amplitude**: $\text{Sawtooth}(\phi) \times K_{\text{slide}} \times 1.5\text{Nm} \times F_{\text{load-texture}} \times (1.0 - \text{Grip}) \times K_{\text{decouple}}$.
- **Note**: Work-based scaling `(1.0 - Grip)` ensures vibration only occurs during actual scrubbing.

2. Road Texture (Bumps)

- **Main Input**: Delta of `mVerticalTireDeflection` (effectively a **High-Pass Filter** on suspension movement).
- **Safety Clamp**: Delta is clamped to **+/- 0.01m** per frame to prevent physics explosions on teleport or restart.
- **Formula**: `(DeltaL + DeltaR) * 50.0 * K_road * F_load_texture * Scale`.
- **Scrub Drag (Fade-In)**:
    - Adds constant resistance when sliding laterally.
    - **Coordinate Note**: LMU uses **+X = Left**. Drag must oppose velocity, so we invert direction.
    - **Fade-In**: Linear scale 0% to 100% between **0.0 m/s** and **0.5 m/s** lateral velocity.
    - **Formula**: `(SideVel > 0 ? -1 : 1) * K_drag * 5.0Nm * Fade * Scale`.

3. Traction Loss (Wheel Spin)

- **Trigger**: Throttle > 5% and SlipRatio > 0.2 (20%).
- **Torque Drop ($M_{\text{spin-drop}}$)**: The *Total Output Force* is reduced to simulate "floating" front tires.
    - `M_spin-drop = (1.0 - (Severity * K_spin * 0.6))`
- **Vibration**:
    - **Frequency**: $10\text{Hz} + (\text{SlipSpeed} \times 2.5)$. Cap 80Hz.
    - **Formula**: $\sin(\phi) \times \text{Severity} \times K_{\text{spin}} \times 2.5\text{Nm} \times K_{\text{decouple}}$.

4. Suspension Bottoming

- **Triggers**:
    - Method A: `RideHeight < 2mm`.

    - Method B: `SuspForceRate > 100,000 N/s`.
    - Legacy: `TireLoad > 8000.0 N`.
- **Formula**: `sin(50Hz) * K_bottom * 1.0Nm`. (Fixed sinusoidal crunch).
- **Legacy Intensity**: $\sqrt{\text{ExcessLoad}} \times 0.0025$. (Scalar restored to legacy value for accuracy).

F. Post-Processing & Filters

1. Signal Filtering

- **Notch Filters**:
    - **Dynamic**: $Freq = \text{Speed}/\text{Circumference}$. Uses Biquad.
    - **Static**: Fixed frequency (e.g. 50Hz) Biquad.
- **Frequency Estimator**: Tracks zero-crossings of `mSteeringShaftTorque` (AC coupled).

2. Gyroscopic Damping (F_{gyro})

- **Input Derivation**:
    - $\text{SteerAngle} = \text{UnfilteredInput} \times (\text{RangeInRadians}/2.0)$
    - $\text{SteerVel} = (\text{Angle}_{\text{current}} - \text{Angle}_{\text{prev}})/dt$
- **Formula**: $-\text{SteerVel}_{\text{smooth}} \times K_{\text{gyro}} \times (\text{Speed}/10.0) \times 1.0\text{Nm} \times K_{\text{decouple}}$.
- **Smoothing**: Time-Corrected LPF.

3. Time-Corrected LPF (Algorithm)
Standard exponential smoothing filter used for Slip Angle, Gyro, SoP, and Shaft Torque.

- **Formula**: $State+ = \alpha \times (Input - State)$
- **Alpha Calculation**: $\alpha = dt/(\tau + dt)$
    - dt: Delta Time (e.g., 0.0025s)
    - τ (Tau): Time Constant (User Configurable, or derived from smoothness). **Target**: ~0.0225s (from 400Hz).

4. Min Force (Friction Cancellation)
Applied at the very end of the pipeline to `F_norm` (before clipping).

- **Logic**: If $|F| > 0.0001$ AND $|F| < K_{\text{min-force}}$:
    - $F_{\text{final}} = \text{Sign}(F) \times K_{\text{min-force}}$.
- **Purpose**: Ensures small forces are always strong enough to overcome the physical friction/deadzone of gear/belt wheels.

---

7. Telemetry Variable Mapping

Math Symbol	API Variable	Description
T_{shaft}	`mSteeringShaftTorque`	Raw steering torque (Nm)
Load	`mTireLoad`	Vertical load on tire (N)
GripFract	`mGripFract`	Tire grip scaler (0.0-1.0)
Accel_X	`mLocalAccel.x`	Lateral acceleration (m/s²)
Accel_Z	`mLocalAccel.z`	Longitudinal acceleration (m/s²)
YawAccel	`mLocalRotAccel.y`	Rotational acceleration (rad/s²)
Vel_Z	`mLocalVel.z`	Car speed (m/s)
$\text{SlipVel}_{\text{lat}}$	`mLateralPatchVel`	Scrubbing velocity (m/s)
SuspForce	`mSuspForce`	Suspension force (N)
$\text{Pedal}_{\text{brake}}$	`mUnfilteredBrake`	Raw brake input (0.0-1.0)

---

8. Legend: Physics Constants (Implementation Detail)

Constant Name	Value	Description
`BASE_NM_LOCKUP`	4.0 Nm	Reference intensity for lockup vibration
`BASE_NM_SPIN`	2.5 Nm	Reference intensity for wheel spin
`BASE_NM_ROAD`	2.5 Nm	Reference intensity for road bumps
`REAR_STIFFNESS`	15.0	N/(rad·N) - Estimated rear tire cornering stiffness
`WEIGHT_TRANSFER_SCALE`	2000.0	N/G - Kinematic load transfer scaler
`UNSPRUNG_MASS`	300.0 N	Per-corner static unsprung weight estimate
`BOTTOMING_LOAD`	8000.0 N	Load required to trigger legacy bottoming
`BOTTOMING_RATE`	100kN/s	Suspension force rate for impact bottoming
`MIN_SLIP_VEL`	0.5 m/s	Low speed threshold for slip angle calculation