**F1 Simulation Model Technical Documentation**

**Parameter Sources and Calculations**

**Fixed Parameters (Regulatory/Factual)**

* **Fuel consumption rate**: 110kg total fuel ÷ 72 laps = 1.528 kg/lap (F1 regulation)
* **Points system**: Standard F1 points allocation (25-18-15-12-10-8-6-4-2-1)
* **Base pace**: 71.0s (Zandvoort representative lap time)
* **Race distance**: 72 laps (Dutch GP regulation distance)

**Zandvoort-Specific Parameters (source in parenthesis)**

* **Rain probability**: 30% (via Open Weather API)
* **SC probability**: 67% (Formula 1 official estimate)
* **VSC probability**: 67% (Formula 1 official estimate)
* **Pit time loss**: 21.52s (Formula 1 official estimate)
* **Fuel effect**: 0.035s/kg (standard F1 estimate)

**Extracted Parameters (Data-Driven via F1\_Parameter\_Extractor.py)**

* **Position penalties**: Calculated by comparing qualifying pace vs race pace (laps 8-30), accounting for compound differences
* **Tire performance**: Linear regression on stint lap vs lap time, using laps 2-25 of each stint
* **Driver error rates**: Residual analysis after fitting trend lines to stint data, defining errors as laps >2σ slower than expected
* **DRS effectiveness**: Attempts to measure lap time improvements when cars run close together, falls back to 0.25s theoretical estimate

**Fallback Parameters (Used when extraction unavailable)**

* **Position penalties**: Linear approximation (0.12 × (position-1) seconds)
* **Tire offsets**: SOFT baseline: 0.0s, MEDIUM: +0.35s, HARD: +0.65s
* **Degradation rates**: SOFT: 0.18s/lap, MEDIUM: 0.09s/lap, HARD: 0.05s/lap
* **Error rates**: 0.012 dry, 0.030 wet (base probability per lap)

**Parameter Extraction Technical Details**

**F1ParameterExtractor.extract\_position\_penalties()**

* Loads qualifying and race sessions via FastF1
* Extracts Q3/Q2/Q1 times for grid positions
* Gets race laps 8-30 (excludes start effects and tire degradation)
* Groups by driver and compound to normalize strategy differences
* Calculates expected race pace as quali time + 0.5s
* Measures actual penalty as (actual race pace - expected race pace)
* Groups by grid position and takes median penalty

**F1ParameterExtractor.extract\_tire\_performance\_soft\_baseline()**

* Filters race laps for green flag, non-pit conditions
* Creates stint lap counter (cumulative within each stint)
* Uses laps 2-25 of each stint (excludes out lap and extreme degradation)
* Fits LinearRegression(stint\_lap, lap\_time) for each compound
* Removes outliers beyond 2.5 standard deviations
* Calculates offset relative to SOFT compound base time

**F1ParameterExtractor.extract\_driver\_error\_rates()**

* Groups laps by driver and stint
* Fits linear trend line to remove tire degradation effects
* Calculates residuals (actual - predicted lap time)
* Defines errors as laps where residual > 2σ from stint trend
* Counts error rate as (error laps / total laps) per stint
* Separates dry compounds (S/M/H) from wet compounds (I/W)

**F1ParameterExtractor.extract\_drs\_effectiveness()**

* Attempts to find lap time improvements for cars running close together
* Compares current lap vs median of previous 5 laps for same compound
* Identifies improvements between 0.1-1.0s as potential DRS usage
* Removes outliers using IQR method
* Falls back to 0.25s theoretical estimate if insufficient data

**Monte Carlo Simulation Technical Details**

**simulate\_race() Function**

* **Weather generation**: Uses numpy.random with specified rain probability, creates lap-by-lap weather array
* **SC/VSC generation**: Random selection based on probability thresholds, determines start lap and duration
* **Lap time calculation**: Base pace + compound offset + (degradation rate × stint lap)
* **Fuel effects**: Decreases fuel load by consumption rate per lap, applies fuel\_effect\_per\_kg
* **Position penalties**: Applies traffic factor that decreases over race distance
* **Driver errors**: Random probability check per lap, increases with stint length
* **Pit stop execution**: Normal distribution around pit time loss, reduced during SC/VSC periods

**Tire Performance Model**

* **Base calculation**: get\_tire\_performance(compound, lap\_in\_stint, base\_pace, weather)
* **Dry conditions**: base\_pace + compound\_offset + (degradation\_rate × lap\_in\_stint)
* **Wet conditions**: base\_pace + 9s + intermediate\_offset + (wet\_degradation × lap\_in\_stint)
* **Non-linear degradation**: Additional penalty for laps >25 in stint
* **Temperature effects**: Additional penalties for SOFT overheat, HARD cold tire phases

**Position Change Mechanics**

* **Undercut probability**: 25% first stop, 12% subsequent stops
* **Position changes**: Discrete random selection from [-2, -1, 0, +1, +2] with defined probabilities
* **Strategic aggressiveness**: Bonus position changes for strategies using 2+ aggressive compounds
* **Bounds checking**: Positions clamped between 1 and 20

**Bayesian Tire Modeling**

**build\_tire\_model() Function**

* Uses numpyro with NUTS sampler for MCMC
* **Prior distributions**:
  + alpha ~ Normal(base\_pace, 4) for intercept
  + beta ~ Normal(expected\_degradation, 0.02) for slope
  + sigma ~ HalfNormal(0.5) for noise
* **500 warmup samples + 1000 posterior samples**
* **Model**: LapTime ~ Normal(alpha + beta × StintLap, sigma)
* Only built if compound has >5 data points from 2024 historical data

**Validation Model Technical Details**

**generate\_simulation\_results()**

* Imports run\_monte\_carlo\_with\_grid from the main simulation
* Runs 1000 simulations per grid position
* Converts output format to validation structure (final\_positions, points, times arrays)
* Falls back to approximation method if simulation import fails

**validate\_position\_predictions()**

* Filters strategies based on actual weather conditions (wet vs dry)
* Calculates Mean Absolute Error between predicted and actual final positions
* Computes accuracy within ±2 positions as percentage
* Selects best strategy based on lowest MAE
* Records position error as (predicted - actual) for bias analysis

**validate\_race\_conditions()**

* Compares simulation rain probability (30%) against actual weather occurrence
* Detects safety cars by identifying laps >25% slower than rolling median baseline
* Compares simulation SC probability (67%) against detected incidents
* Reports parameter quality based on extraction sample sizes

**Weather/SC Detection Methods**

* **Rain detection**: Counts INTERMEDIATE/WET tire usage vs dry compound usage
* **SC detection**: Rolling 5-lap median baseline, identifies anomalous slow laps
* **Strategy classification**: Counts pit stops and identifies compound sequences

**Model Architecture Flow**

**1. Parameter Extraction Phase**

Historical Data (FastF1) → Data Cleaning → Statistical Analysis → Parameter File

**2. Simulation Phase**

Parameters + Strategy → Monte Carlo Loop → Weather/SC Generation → Lap Simulation → Position Tracking → Results

**3. Validation Phase**

Actual Race Data → Simulation Results → Comparison Analysis → Accuracy Metrics → Validation Report

**Key Model Limitations**

**Parameter Extraction**

* Limited to data available in FastF1 (may miss some data sources)
* Position penalty calculation assumes quali-race delta is purely traffic-related
* DRS effectiveness measurement is proximity-based, not direct telemetry
* Small sample sizes for some parameters reduce reliability

**Simulation Model**

* Binary weather states (dry/wet) vs gradual transitions
* Simplified position change mechanics vs complex race dynamics
* Fixed car performance gaps vs variable competitive windows
* No driver-specific performance modeling

**Validation Model**

* Only validates against single race outcome vs statistical expectations
* Cannot validate weather/SC probabilities from single event
* Strategy filtering assumes perfect hindsight about race conditions
* Position prediction accuracy depends heavily on parameter quality

**Data Quality Indicators**

**High Quality (>100 samples)**

* Position penalties for grid positions 1-10
* Tire performance for major compounds (S/M/H)
* Driver error rates in dry conditions

**Moderate Quality (20-100 samples)**

* Position penalties for grid positions 11-20
* Driver error rates in wet conditions
* DRS effectiveness measurements

**Low Quality (<20 samples)**

* Track-specific parameters for rarely visited circuits
* Weather-specific degradation rates
* Qualifying vs race pace correlations in mixed conditions

**Tire Strategies**

Based on historical averages from 2021 – 2024 Dutch Grands Prix (via FastF1 API)