SimPulse: The Formula E Simulator

Welcome to the mathematical core of SimPulse, a high-fidelity Formula E racing simulator. This presentation details the rigorous equations and models that govern every aspect of the simulated race experience, from the precise physics of vehicle dynamics to the probabilistic nuances of on-track events and strategic decisions.

For graphical plots and code visit: https://github.com/akshat3144/simpulse

Core Simulation Logic: Understanding the State

SimPulse's foundation lies in its comprehensive **state vector representation** and the **fundamental transition equation**. Every car in the simulation is defined by a 20-dimensional state vector, capturing its complete physical and operational status at any given time.

Fundamental Transition Equation

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}(t)) + \boldsymbol{\varepsilon}(t)$$

- **x(t)**: The car's state vector at time **t**.
- **f()**: Deterministic physics function, modeling known physical laws.
- **u(t)**: Control inputs (throttle, brake, steering, attack mode activation).
- **θ(t)**: Environmental parameters like weather and track conditions.
- **ε(t)**: Gaussian process noise, introducing real-world uncertainty.

State Vector Representation

$$\mathbf{x}_i = \begin{bmatrix} v_x \\ v_y \\ p_x \\ p_y \\ E_{\%} \\ T_{\text{batt}} \\ \tau \\ \mu \\ \text{AM}_{\text{active}} \\ t_{\text{AM}} \\ d_{\text{lap}} \\ d_{\text{total}} \\ L \\ P \\ \theta \\ \beta \\ \delta \\ a_{\text{lat}} \\ a \\ P_i \end{bmatrix} \in \mathbb{R}^{20}$$

- **v_x, v_y**: Longitudinal and lateral velocities.
- **p_x, p_y**: Position coordinates on the track.
- **E%**: Battery charge percentage.
- **τ**: Tire degradation level.
- **P_i**: Performance index, a composite score of car performance.

The simulator operates at a high frequency of **100 Hz**, meaning the state vector is updated every **0.01 seconds**, ensuring precision and responsiveness in the simulation. For multi-car scenarios, a **Multi-Car State Matrix** consolidates the state vectors of all participating cars, enabling vectorized computations crucial for performance.

Vehicle Dynamics: The Physics of Motion

The vehicle dynamics module rigorously calculates how forces act on the car, determining its acceleration, speed, and cornering capabilities. This involves a detailed force balance, sophisticated motor power modeling, and nuanced aerodynamic considerations.

Longitudinal Force Balance

$$F_{\text{net}} = F_{\text{motor}} - F_{\text{drag}} - F_{\text{roll}} - F_{\text{brake}} - F_{\text{gradient}}$$

- **F_net**: The net force driving the car forward or slowing it down.
- Each component represents a specific force (motor, drag, rolling resistance, braking, gradient) acting on the vehicle.

Motor Force & Aerodynamic Drag

Motor power is dynamically adjusted based on throttle input and Attack Mode status, while aerodynamic drag significantly impacts top speed.

$$P_{\text{motor}} = \begin{cases} (P_{\text{max}} + P_{\text{boost}}) \cdot \theta & \text{if AM active} \\ P_{\text{max}} \cdot \theta & \text{otherwise} \end{cases}$$

$$F_{\text{drag}} = \frac{1}{2}\rho C_d A v^2$$

- **P_max**: 350 kW, **P_boost**: 50 kW (Attack Mode).
- **ρ**: Air density, **C_d**: Drag coefficient (0.32), **A**: Frontal area (1.5 m²).

Rolling Resistance & Downforce

Rolling resistance and downforce calculations are critical for accurate grip and speed limits, especially in corners.

$$N = m \cdot g + F_{\text{down}}$$

$$F_{\text{down}} = \frac{1}{2}\rho C_l A v^2$$

- **C_r**: Rolling resistance coefficient (0.015).
- **m**: Total mass (920 kg), **g**: Gravity (9.81 m/s²).
- **C_l**: Lift coefficient (1.8).

Braking and Gradient Forces

Braking includes regenerative components, while gradient force accounts for track elevation changes.

$$F_{\text{brake}} = \beta \cdot m \cdot a_{\text{max,brake}}$$

$$F_{\text{gradient}} = m \cdot g \cdot \sin(\alpha)$$

- **a_max,brake**: Max deceleration (5.5 m/s²).
- **α**: Road gradient angle.

The simulator also models **Ackermann Steering** and calculates **Cornering Speed Limits** based on tire grip, track conditions, and banking angles, ensuring that lateral dynamics are realistically represented. Maximum speed is capped at 322 km/h (89.44 m/s) for Gen3 cars.

Energy System: Strategic Power Management

Formula E's unique energy management challenge is meticulously modeled in SimPulse. The simulator tracks battery state, energy consumption, and recovery, reflecting the critical balance between speed and efficiency.

Battery State Update

$$E(t+1) = E(t) - E_{\text{consumed}}(t) + E_{\text{regen}}(t)$$

The battery's energy level is a net calculation of energy consumed by the motor and energy recovered through regenerative braking. Battery capacity is set to **51 kWh (183.6 MJ)** for a Gen3 car.

Energy Consumption

$$E_{\text{consumed}} = \frac{P_{\text{motor}}}{\eta_{\text{motor}}} \cdot \Delta t$$

Consumption is directly proportional to motor power demand and adjusted for motor efficiency. Attack Mode activation introduces a **30% multiplier** on energy consumption, necessitating careful strategic use.

Energy Recovery

$$E_{\text{regen}} = P_{\text{regen}} \cdot \eta_{\text{regen}} \cdot \Delta t$$

Regenerative braking recovers kinetic energy, with a maximum regen power of **600 kW** and an efficiency of **40%**. This mechanism is crucial for extending battery life throughout the race.

Energy Management Modifier

$$k_E = \begin{cases} 0.92 & E_{\%} < 15\% \\ 0.95 & 15\% \leq E_{\%} < 30\% \\ 1.00 & E_{\%} \geq 30\% \end{cases}$$

A dynamic modifier reduces throttle input at lower battery percentages, simulating an autonomous energy conservation strategy to ensure the car finishes the race.

Effective energy management is not just about maximizing speed but optimizing the trade-off between power delivery and battery longevity. SimPulse's detailed model allows for deep analysis of different energy strategies.

Tire Model: Grip, Degradation, and Temperature

Tires are the sole point of contact with the track, making their behavior critical to performance. SimPulse incorporates a multi-factor tire model that accounts for degradation, grip changes, and thermal dynamics.

Tire Degradation Rate

$$\frac{d\tau}{dt} = k_{\text{base}} + k_{\text{temp}}|T_{\text{tire}} - T_{\text{opt}}| + k_{\text{speed}}v^2 + k_{\text{lat}}|a_{\text{lat}}|^2 + k_{\text{lock}}\mathbb{I}_{\text{lock}}$$

Tire wear (τ) increases based on a base rate, temperature deviation from optimal (90°C), speed, lateral acceleration, and severe events like wheel lock-ups. This intricate model captures the nuances of tire management in real racing scenarios.

Grip Coefficient Evolution

$$\mu(\tau) = \mu_{\text{max}} - (\mu_{\text{max}} - \mu_{\text{min}}) \cdot \tau$$

As tires degrade, their grip coefficient (μ) decreases linearly from a maximum of **1.2** (new tires) to a minimum of **0.9** (worn tires). This directly impacts cornering speed and braking performance.

Weather Effect on Grip

$$\mu_{\text{wet}} = \mu \cdot (1 - 0.25 \cdot I_{\text{rain}})$$

Rain significantly reduces grip. The simulator models up to a **25% grip loss** in heavy rain, impacting driver strategy and car behavior on a wet track.

Tire Temperature Evolution

$$\frac{dT_{\text{tire}}}{dt} = Q_{\text{friction}} - Q_{\text{cooling}}$$

Tire temperature is influenced by heat generated from friction (proportional to acceleration and lateral forces) and dissipated through cooling to the ambient air. Maintaining optimal tire temperature is crucial for peak performance.

These models collectively define a realistic tire behavior, forcing drivers to consider tire wear and temperature in their race strategy.

Thermal Dynamics: Battery and System Health

Beyond tire temperatures, the simulator models the critical thermal management of the battery and other components, which directly influences performance and reliability.

Battery Temperature

$$\frac{dT_{\text{batt}}}{dt} = Q_{\text{gen}} - Q_{\text{cool,active}} - Q_{\text{cool,passive}}$$

Battery temperature (**T_batt**) changes based on internally generated heat (**Q_gen**) from energy losses, active cooling, and passive cooling to the environment. The optimal temperature is **40°C**, with a safe operating range of **20-60°C**.

Heat Generation

$$Q_{\text{gen}} = \frac{P_{\text{loss}}}{m_{\text{batt}} \cdot c_p}$$

Waste heat power (**P_loss**) arises from inefficiencies in energy consumption and regeneration. This heat is distributed across the battery mass (**m_batt**) and specific heat capacity (**c_p**), causing temperature to rise.

Active & Passive Cooling

$$Q_{\text{cool,active}} = k_{\text{cool}}(T_{\text{batt}} - T_{\text{opt}})$$

$$Q_{\text{cool,passive}} = k_{\text{passive}}(T_{\text{batt}} - T_{\text{ambient}}) \cdot \Delta t$$

Both active and passive cooling mechanisms work to dissipate heat. Active cooling engages when the battery exceeds its optimal temperature, while passive cooling continuously accounts for the temperature differential with the ambient environment.

Maintaining optimal battery temperature is paramount for consistent power delivery and preventing degradation. Exceeding safe ranges can lead to power derating, impacting overall performance.

Driver Control: Algorithmic Precision

The simulator features an advanced driver control system that autonomously navigates the track, managing speed, braking, steering, and Attack Mode activation. This system emulates human decision-making based on target speeds and real-time conditions.

Target Speed Calculation

$$v_{\text{target}} = v_{\text{max,segment}} \cdot \alpha_{\text{skill}} \cdot \alpha_{\text{aggr}}$$

The desired speed is a function of the maximum safe speed for a given track segment, modulated by the driver's skill and aggression factors.

Attack Mode Decision

$$P(\text{activate AM}) = \begin{cases} 0.6 + 0.3\alpha_{\text{aggr}} & \text{if conditions met} \\ 0 & \text{otherwise} \end{cases}$$

Attack Mode activation is a probabilistic event, influenced by conditions such as remaining activations, proximity to other cars, current race position, and the driver's aggression.

Throttle Control

Throttle input (**θ**) is proportional to the speed error (**e_v** = target speed - current speed), with a deadband to prevent oscillations. Aggression factors influence the throttle gain.

$$e_v = v_{\text{target}} - v_{\text{current}}$$

Brake Control

Brake input (**β**) is activated when the current speed exceeds the target speed. Different gains are applied for corners (more aggressive) versus straights (gentler braking).

Steering Control

$$\delta = \arctan\left(\frac{L_{\text{wb}}}{r}\right) \cdot s$$

Steering angle (**δ**) is calculated using Ackermann geometry, based on the wheelbase (**L_wb**) and corner radius (**r**), with an assigned turn direction (**s**).

This sophisticated control system ensures that each simulated driver reacts dynamically to track conditions, opponent behavior, and resource management, providing a rich and realistic racing experience.

Stochastic Dynamics: The Unpredictable Edge

Real-world racing is never perfectly deterministic. SimPulse embraces this reality by incorporating sophisticated stochastic models that introduce controlled randomness, reflecting the inherent uncertainties in driver inputs, physical processes, and component wear.

Control Input Noise

$$\mathbf{u}_{\text{actual}} = \mathbf{u}_{\text{ideal}} + \boldsymbol{\varepsilon}_{\text{control}}$$

Driver inputs are not perfectly precise. Gaussian noise is added to throttle, brake, and steering inputs, with the magnitude of this noise inversely proportional to the driver's consistency factor. This simulates slight variations in driver execution.

Process Noise

$$\mathbf{x}(t + \Delta t) = \mathbf{x}^*(t + \Delta t) + \sqrt{\Delta t} \cdot \boldsymbol{\varepsilon}_{\text{process}}$$

Even with ideal controls, the physical world introduces unpredictability. Process noise, scaled by the square root of the timestep (Brownian motion scaling), affects various state components like velocity, position, and temperatures, simulating micro-fluctuations on the track.

Tire Degradation Noise

Tire wear isn't perfectly uniform. A noise component is added to the base tire degradation, with its variance increasing as tires get hotter. This reflects the reality that hotter tires can lead to more unpredictable and accelerated wear patterns.

Energy Consumption Noise

Energy consumption isn't always exact. Noise is applied to the base energy consumption, with its variance dependent on the battery temperature. Deviations from the optimal battery temperature increase the unpredictability of energy usage.

By incorporating these stochastic elements, SimPulse generates more dynamic, unpredictable, and ultimately realistic race simulations, mirroring the challenges and fortunes of real Formula E events.

Event Probabilities: Racing's Critical Moments

Beyond continuous dynamics, racing is defined by discrete events. SimPulse features a robust probabilistic event system that models crashes, overtakes, safety car deployments, and mechanical failures, all emerging organically from underlying mathematical distributions and risk factors.

Crash Probability

$$P(\text{crash}|\mathbf{x}, t) = p_{\text{base}} \cdot (1 + \kappa \cdot R(\mathbf{x}))$$

Crash risk escalates based on a sigmoid function, where a base probability is amplified by a risk factor **R(x)**. This factor is a weighted sum of speed, tire degradation, driver aggression, car proximity, and energy stress, ensuring crashes are contextually relevant.

Overtake Probability

$$P(\text{overtake}|\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1 + e^{-z}}$$

Overtaking success is modeled by a logistic function, with the exponent **z** incorporating speed differential, energy advantage, Attack Mode status, tire condition, and track segment factors. This model creates realistic overtaking scenarios based on car performance and strategic decisions.

Safety Car

$$P(\text{SC in lap } \ell) = 1 - e^{-\lambda(\ell)}$$

Safety Car deployments follow a Poisson process, meaning they occur randomly but with a rate **λ(l)** that increases based on recent crashes. Guards prevent immediate deployments after previous Safety Cars or on the first lap.

Mechanical Failure

$$F(t) = 1 - e^{-(t/\lambda)^k}$$

Mechanical failures are governed by a Weibull distribution, which accurately models an increasing hazard rate over time, meaning components are more likely to fail as they age. Stress accelerates this aging process.

These probabilistic models ensure that each simulated race tells a unique and compelling story, reflecting the high stakes and inherent uncertainty of Formula E racing.

Performance Metrics & GPU Vectorization

SimPulse provides a comprehensive suite of performance metrics to evaluate race outcomes and car behavior. To ensure high-fidelity simulations run efficiently, especially with multiple cars, the backend leverages GPU vectorization for parallel processing.

Performance Index (P_i)

$$P_i(t) = \sum_{j=1}^{5} w_j \cdot c_j(t)$$

A composite score combining normalized velocity, acceleration, energy, tire condition, and strategy effectiveness. This index provides a single metric for overall car performance, with weights assigned to each component reflecting its importance.

Gap Calculations

$$\Delta t^i_{\text{leader}} = \frac{d^{\text{leader}}_{\text{total}} - d^i_{\text{total}}}{v^i}$$

Critical for race strategy, these calculations provide time gaps to the race leader and the car immediately ahead. These approximations assume a constant average speed for simplicity in real-time analysis.

GPU Vectorization

$$\mathbf{X}(t + \Delta t) = \mathbf{F}(\mathbf{X}(t)) + \mathbf{G}\mathbf{U}(t) + \mathcal{E}(t)$$

For multi-car simulations, the entire state matrix (X) and control inputs (U) are processed in parallel on the GPU. This vectorized approach drastically reduces computation time. Empirical scaling shows significant speedups: **S(100 cars) ≈ 100×**, **S(500 cars) ≈ 160×**.

These tools allow engineers and developers to gain deep insights into simulation behavior and optimize computational performance, ensuring SimPulse remains a cutting-edge Formula E simulator.