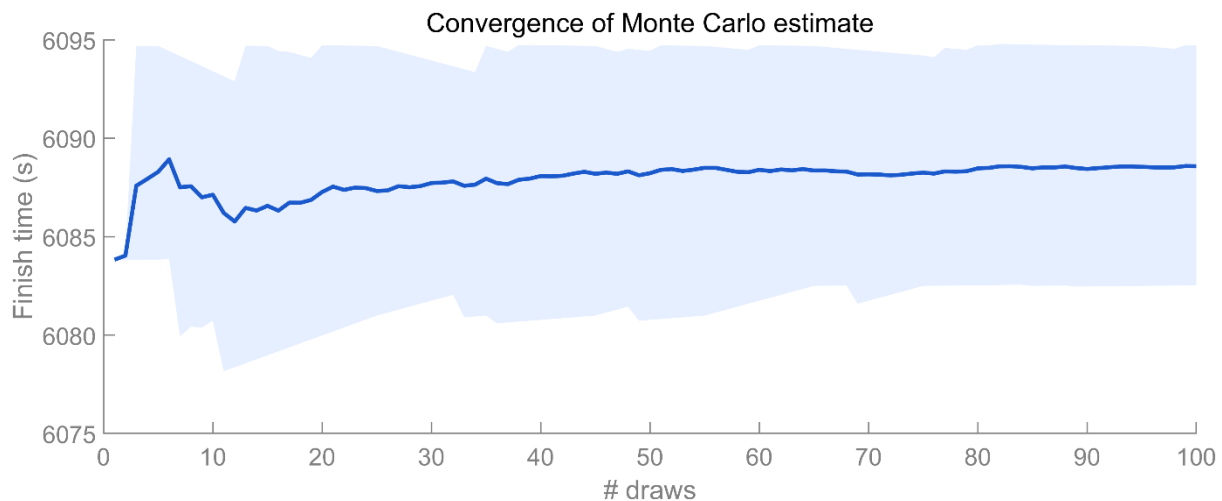


University of Wales Trinity Saint David

Wales institute of Science and Arts



The Application of Monte Carlo Simulation Methods for Race Strategy Optimisation

MSc Motorsport engineering

Joseph Britton (2104278)

ACFA7003 Masters project

Supervisors: Andrew Noble & Ryan James

September 2025

Declaration of authenticity

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: 

Date: 07/09/2025

Abstract

This research constructs a probabilistic race-strategy optimisation tool with Monte Carlo simulation for the treatment of the stochastic complexity of motorsport strategy. A deterministic lap-by-lap MATLAB model was gradually enhanced with calibrated tyre wear curves, nonlinear fuel consumption modelling, pit stop distributions, and probabilistic traffic penalties. Progressive cropping was introduced to enable large-scale simulation efficiency. Validation against Formula 1 race data (2024–2025 seasons) proved that the degradation–fuel model recovered realistic lap time behaviours and pit timing, and strategy choice very closely replicated real-world behaviour. Convergence testing assured that no individual "best" solution was established, but rather a distribution of approximately optimal strategies consistent with multi-modality of race strategy under uncertainty. The cropping algorithm resulted in wall time saving of ~24–27% with high fidelity (Overlap @ $K \approx 0.95$ –1.00, Spearman's $\rho \approx 0.93$ –0.99, MAE ≈ 9 –14 s). Traffic modelling produced realistic penalty distributions (mean ≈ 1.5 s per interaction) but requires tuning to avoid early-lap artefacts. Overall, the tool demonstrates that Monte Carlo methods can generate fast, robust, and data-driven strategy proposals. The model is a basis for endurance racing use, where extended features including multi-driver rosters, refuelling, tyre assignments, and day-of-week performance impacts can be added.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Andrew Noble and Ryan James, for their guidance, support, and constructive feedback throughout the course of this project. Their expertise and encouragement have been invaluable in shaping the direction of this research. I am also grateful to the University of Wales Trinity Saint David for providing the resources and environment that enabled this work to be undertaken. Finally, I wish to thank my family and friends for their patience, encouragement, and unwavering support during my studies.

Table of Contents

Nomenclature	v
Table of figures	ix
Table of tables	xi
Table of equations	xi
1. Introduction	1
1.1. Project relevance	1
2. Aims and objectives	3
2.1. Aims of the project	3
2.2. Objectives of the project	3
2.3. Project timeline	3
3. Literature review	5
3.1. Monte Carlo simulation	5
3.1.1. Classic Monte Carlo	6
3.1.2. Monte Carlo integration	7
3.1.3. Markov chain Monte Carlo	8
3.1.4. Motorsport applications of Monte Carlo simulation	9
3.1.5. Strengths and weaknesses of Monte Carlo simulation	10
3.2. KPI's for simulation	11
3.2.1. Tyre degradation	11
3.2.2. Fuel consumption and mass effect	12
3.2.3. Pit stop loss	12
3.2.4. Traffic and multiclass interference	13
3.3. Race strategy techniques	14
3.3.1. Sprint racing	14
3.3.2. Endurance Racing	16
4. Methodology	19
4.1. Initial model building	19
4.1.1. Race selection	19

4.1.2.	Data extraction and processing	19
4.1.3.	Baseline degradation model.....	20
4.1.4.	Simulation of pit strategies	21
4.1.5.	Improved tyre degradation model.....	23
4.1.6.	Fuel burn modelling	24
4.1.7.	Combined simulation (tyre degradation and fuel modelling)	27
4.1.8.	Validation of baseline model	27
4.2.	Monte Carlo simulation implementation	28
4.2.1.	Randomised pit stops	28
4.2.2.	Traffic interaction modelling	32
4.2.3.	Cropping algorithm.....	34
4.2.4.	Driver thermal management modelling.....	41
4.3.	Simulation evaluation and validation	43
5.	Results and analysis	49
5.1.	Convergence testing.....	49
5.2.	Simulation parameter validations	51
5.3.	Traffic modelling integration	53
5.4.	Endurance race strategy implementation and adaptations.....	58
5.4.1.	Multi-driver roster	58
5.4.2.	Refuelling and stops	58
5.4.3.	Tyre allocation across event	58
5.4.4.	Multiclass traffic as a persistent stochastic process	59
5.4.5.	Night to day performance differences.....	59
5.4.6.	Objectives and evaluation of endurance racing.....	59
5.5.	Simulation shortcomings.....	59
6.	Conclusion	62
	References	64
	Appendices	69

Nomenclature

<u>Term</u>	<u>Definition</u>
l	Current lap number within a stint or race. Dimensionless.
$No. laps$	Total laps in the simulated event
k	Cutoff for “top-K” lists when measuring agreement (Overlap @ K). Dimensionless
N	— May denote initial grid size or number of Monte-Carlo iterations, depending on context. Dimensionless.
$Seed/rng$	— Random-number generator seed controlling stochastic reproducibility. Dimensionless
t_{base}	Base performance with no degradation and reference mass. Units: seconds
Δt_{deg}	Added lap-time from tyre wear at lap ℓ . Units: seconds
Δt_{fuel}	Lap-time <i>benefit</i> from mass reduction at lap ℓ . Units: seconds
a, b	Shape/scale parameters of the tyre-degradation function (Eq. 2), tuned per compound. Dimensionless.
k_{fuel}	Coefficient converting mass change to lap-time effect in the fuel model (Eq. 4). Units: seconds
m_0	Starting fuel mass at race/stint start. Units: kg
$m(l)$	Fuel mass at lap ℓ . Units: kg

$\Delta T_{fuel}(l)$	Cumulative lap-time loss relative to start from the fuel model (logarithmic approach). Units: seconds
<i>Constant burn rate</i>	“Fuel mass / No. laps” (Eq. 3) for per-lap mass decrement. Units: kg/lap
<i>Fuel time – gain per kg</i>	Empirical effect used in early model: 0.03 s/kg benefit. Units: s/kg
<i>Min pit lap, max pit lap</i>	Bounds of the admissible pit window for randomised pit-lap selection (uniform distribution via randi). Dimensionless (laps
<i>Pit time loss</i>	— Deterministic component for pit-lane transit (speed-limited time). Units: seconds
$\hat{s}(l)$	Sector-wise position proxy to detect proximity events; Equation 7. Units: m
$t_{cum}(l)$	Elapsed time proxy used with $\hat{s}(\ell)$ for interaction checks; Equation 8. Units: seconds
d_{int}	Distance/condition under which a faster car is in “traffic” with a slower one. Units: m
R	Number of sequential culls (e.g., 6 in baseline). Dimensionless.
s_{end}	Count of final survivors after cropping. Dimensionless
t_{wall}	Execution time; used for speedup and % reduction metrics. Units: seconds
<i>Overlap @ k</i>	Fractional overlap of top-K lists between runs/seeds. Dimensionless in [0,1]
<i>Spearman's ρ</i>	Rank-order correlation between lists (survivors). Dimensionless in [−1,1]

MAE	Mean $ \Delta \text{ race-time} $ among comparable survivors. Units: seconds
$Dominant \text{ Share}$	Maximum probability mass of a single recommended strategy among survivors. Dimensionless
$Entropy, Gini - simpson$	Diversity of recommended strategies (Entropy $H = -\sum p_i \log p_i$; Gini-Simpson $1 - \sum p_i^2$). Dimensionless
$P \text{ (push factor)}$	Driver aggression/thermal push level, 0–1 per run. Dimensionless
g_{max}	Maximum attainable <i>instant</i> lap-time gain from push. Units: seconds
s_{comp}	Compound-scale factor to differentiate push response across soft/medium/hard. Dimensionless
$V \text{ (concavity of gains)}$	— Exponent/shape for diminishing returns in push-gain curve (Eq. 10). Dimensionless
w_{rate}	Compound-specific baseline wear rate. Dimensionless
α, β	— Exponent/scale controlling how push accelerates wear (Eq. 11). Dimensionless
ε	Random factor around 1 to avoid extremes in wear amplification. Dimensionless
p^*	Optimal push (minimiser of average stint lap time). Solved by differentiating Eq. 9 ($\rightarrow$ Eq. 12–15). Dimensionless
$Compound \text{ multipliers}$	Per-compound grip multiplier and compound factor used in switch-case selection for tyre behaviour. Dimensionless
$Compound \text{ base}$	(initial offset between compounds)

<i>Warm – up decay</i>	term (exponential to 0)
<i>Plateau drift</i>	slope (near-linear mid-stint degradation)
<i>Cliff (logistic)</i>	centred logistic scaled to govern late-stint drop. Dimensionless parameters
<i>Standing start & warm – up penalty</i>	+3% time on first 3 laps to emulate launch/traffic warm-up in validation. Dimensionless (%).
<i>Pit window logic</i>	“No pitting in first X laps”; “No pitting in last X laps”; “Max X stops”; “Minimum two compounds.” Dimensionless
<i>Degradation – triggered pit</i>	Pit when average lap-time loss over last 3 laps exceeds 3% of base lap time (percentage model). Dimensionless
<i>Offset – free RMSE</i>	RMSE after removing constant offset to emphasise <i>shape</i> matching of time-series. Units: seconds
<i>Mean slope bias</i>	Difference in fitted lap-time slope between sim and real data. Units: s/lap.
<i>Pit – timing alignment</i>	Absolute lap-difference between simulated and actual pit events. Units: laps
<i>Jaccard index (strategy set)</i>	$ A \cap B / A \cup B $ agreement across seeds/iterations; dimensionless

Table of figures

Figure 1 - Example Monte Carlo working schematic	5
Figure 2 - Estimation of pi using Monte Carlo integration from the area of a circle (Pipis, 2019)	7
Figure 3 - Pirelli tyre range and operating limits for 2025 pre-season testing at Sakhir (Motor Sport, 2025)	15
Figure 4 - Extraction of race and free practime 1 lap times and calculation of baseline lap time along with estimation of pitloss	19
Figure 5 – Lando Norris race lap times from the 2024 Singapore GP (FIA, 2024)	20
Figure 6 - MATLAB function for calculating lap times and race times for a no stop strategy	22
Figure 7 - MATLAB function for calculating lap times and race times for a one-stop strategy	22
Figure 8 - MATLAB function for calculating lap times and race times for a two-stop strategy	22
Figure 9 - Comparing baseline simulated lap times against Oscar Piastri's race lap time from Suzuka 2025	23
Figure 10 - Comparing baseline simulated cumulative times against Oscar Piastri's race lap time from Suzuka 2025	23
Figure 11 – (A) Improvements to tyre degradation through equation 2 (B) Adjusting degradation coefficients to $a = 2.35$ and $b = 0.05$ for better degradation curve	24
Figure 12 – (A) Lap time Improvements form adjusting tyre degradation through equation 2 (B) Lap times from adjusting degradation coefficients to $a = 2.35$ and $b = 0.05$ for improved degradation curve	24
Figure 13 - Comparing simulation with fuel effects and without against Oscar Piastri's race at Suzuka 2025	25
Figure 14 – Fuel effect using logarithmic model for fuel burn	26
Figure 15 - Simulation comparison with advanced fuel modelling to Piastri at Suzuka 2025	27
Figure 16 - Comparison of simulated strategy against all race participants for Formula 1 Barcelona 2025	28
Figure 17 - Randomised pit plan coding methodology	29
Figure 18 - Beta distribution of pit stop times for randomisation of pit stops	29
Figure 19 - Tyre health proxy code for emergency pit stop scenario	29
Figure 20 - MATLAB function to switch tyre compound multipliers between compounds	30
Figure 21 - Pit stop time distribution visualised through 10000 simulation iterations	31

Figure 22 - Distribution of pit stop time durations from 2024 Formula 1 season (F1Pace, 2024)	31
Figure 23 - Updated distribution for stopped time in pits using historic Formula 1 data	32
Figure 24 - Validation of vehicle interaction showing gap between two cars and interaction threshold	33
Figure 25 – (A) Beta distribution of traffic time penalties for faster vehicle (B) Impact of probabilistic traffic penalties on lap time progression	34
Figure 26 - Performance and fidelity evaluation of cropping algorithm method versus identical simulation without cropping algorithm (A) Wall time for Crop and no Crop simulations (B) Ratio of no Crop to Crop times (C) Overlap@K (left axis) and Spearman rank (D) MAE of total race time for remaining strategies (E) Percentage wall time reduction from cropping algorithm (F) speedup and Overlap@K versus starting grid size, showing the accuracy/efficiency trade-off	36
Figure 27 - Thermal management trade-off between push for faster lap times vs increased tyre degradation	41
Figure 28 - Effects of push on pace and tyre wear over a 20-lap constant-push stint (Soft / Medium / Hard) using initial specified parameters	42
Figure 29 - Calibrating push through g_{max} over 20 lap constant push for all compounds	43
Figure 30 - Simulation output example for fastest survivor through the cropping algorithm (50000 iterations)	43
Figure 31 – MATLAB function for calculating tyre degradation per lap from equation 16	45
Figure 32 - Tyre degradation per lap using equation 16 for soft, medium and hard compounds	45
Figure 33 - Degradation coefficients illustrated from equation 16 for soft, medium and hard compounds	46
Figure 34 - Example output of unrestrained simulation to restrained simulation with logical pit entry conditional statements added	47
Figure 35 - Tyre degradation time penalty per lap for soft, medium and hard compounds (Catapult, 2024)	48
Figure 36 - Convergence of strategy recommendations versus iteration count for Catalunya, Silverstone, Sakhir and Circuit of the Americas	50
Figure 37 - Fastest simulated car vs. Lando Norris, Lewis Hamilton and Yuki Tsunoda at Marina Bay	52
Figure 38 - Fastest simulated car vs. Lando Norris, Lewis Hamilton and Yuki Tsunoda at Marina Bay after implementation of standing start and warm up phases	53
Figure 39 – Pre and post traffic penalty analysis	54

Figure 40 - Lap time analysis for traffic interaction modelling, comparing baseline run to added traffic.....	55
Figure 41 - Single representation of lap time with traffic penalties against baseline run	56
Figure 42 - Highlighting extreme penalties in early laps from traffic model	57
Figure 43 - Fastest simulated car vs. Oscar Piastri, Nico Hulkenberg and Pierre Gasly at Catalunya.....	60

Table of tables

Table 1 - Project timeline	4
Table 2 - Tyre strategies for all competitors for 2024 Singapore GP (Pitpass.com, 2024) ...	21
Table 3 – Base seed sweep (40–50): Stability of runtime and top-K agreement.....	37
Table 4 - Memory usage snapshots for cropping and no cropping simulation methods, showing peak, average, and largest variable sizes across different execution phases	38
Table 5 - Comparison of memory usage between struct array and cell array implementations across different iteration counts, showing peak, average, and largest variable sizes before, during, and after execution.....	40
Table 6 - Example results of remaining 20 fastest vehicles from cropping simulation showing poor convergence on strategy suggestion (50000 iterations)	44
Table 7 - Tyre degradation coefficients for use in equation 16 for soft, medium and hard compounds	46
Table 8 - Convergence of strategy recommendations versus iteration count (N) performed using Singapore GP track physics.....	49
Table 9 -Strategy summary of the top-20 survivors after progressive cropping for 2,000,000 iterations	51
Table 10 - Tyre strategy Decisions from 2025 Spanish GP for all competitors	61

Table of equations

Equation 1.....	7
Equation 2.....	24
Equation 3.....	25
Equation 4.....	25
Equation 5.....	27
Equation 6.....	29
Equation 7.....	32

Equation 8.....	32
Equation 9.....	41
Equation 10.....	41
Equation 11.....	41
Equation 12.....	42
Equation 13.....	42
Equation 14.....	42
Equation 15.....	42
Equation 16.....	44
Equation 17.....	52
Equation 18.....	52
Equation 19.....	52
Equation 20.....	52
Equation 21.....	52

1. Introduction

Strategic decision-making is a key ingredient to success in motorsport, where the outcome of races is as much determined by car performance and driver talent as by the ability of teams to make optimal tactical choices. In sprint car competitions such as Formula One, strategy is about pit stops timing, tyre compound selection, and exploiting short-term gains in performance. These decisions are taken under highly constrained circumstances and over relatively short time frames, for which deterministic planning methods are normally sufficient. Endurance racing introduces an entirely different set of issues. Events such as the FIA World Endurance Championship (WEC) or the 24 Hours of Le Mans include handling multiple drivers, restricted tyre allocations, and extensive fuel strategies, all while coping with the intricacies of multiclass traffic. The considerably longer race length amplifies the influence of uncertainty, with aspects such as tyre degradation, fuel consumption, traffic jams, and random event race incidents accumulating to create extremely dynamic conditions. These are less adequately handled by traditional deterministic model methods, demanding more advanced techniques able to accommodate variability and uncertainty.

Monte Carlo simulation provides a solid approach for solving this issue. By continuously sampling from probability distributions of significant race parameters, the method produces a distribution of results rather than one projection. This enables risk analysis, the comparison of resilience for several race conditions, and the establishment of methods that are still optimal under a huge number of scenarios. Within the context of endurance racing, Monte Carlo simulations are particularly well-suited to quantifying the combined effect of stochastic pit stop losses, probabilistic traffic impacts, nonlinear fuel burn dynamics, and heterogeneous tyre performance.

The research develops a simulation-based strategy optimisation model constructed in MATLAB operating lap-by-lap and considering the effect of tyre wear, fuel consumption, pit stop strategy, and traffic disruption. A deterministic model is enhanced with the addition of randomness through Monte Carlo simulation, and probabilistic analysis of strategy performance is possible. Validation is in the form of comparison with historical racing data to match observed patterns of performance.

1.1. Project relevance

Race strategy in motorsport is inherently complex, subject to stochastic effects of tyre wear, fuel weight, pit stop variability, traffic interaction, and race interruption. While sprint formats such as Formula 1 have received most interest in simulation-based strategy tools, these formats are well-suited to deterministic optimisation and do not lend themselves to modelling competition's random nature. This project fills that gap by developing a framework that

integrates deterministic lap-time modelling and Monte Carlo simulation to quantify uncertainty and compute robust strategies across a range of scenarios.

The benefit of this work is that it generates not just single "optimal" solutions, but distributions of near-optimal outcomes. This enables decision-makers to evaluate risk, resilience, and margins across different race conditions, supporting strategies that are robust in the face of variability. This echoes the growing industry emphasis on data-driven decision-making and probabilistic modelling, wherein performance must be quantified not only by outright speed but also by robustness against low-probability, high-consequence events.

Beyond motorsport, the methodology demonstrates the wider promise of stochastic optimisation for applied science and engineering. Search in large, uncertain spaces has direct equivalences in autonomous vehicle planning, adaptive control, and predictive maintenance. By developing and proving the methodology in the context-rich world of motorsport where performance matters most, the research demonstrates decision support principles of wide applicability across many engineering problems.

2. Aims and objectives

2.1. Aims of the project

The aim of the project is to develop and validate a probabilistic race strategy optimisation tool. The tool will use Monte Carlo simulation to integrate the effects of tyre degradation, fuel burn, pit stop variability, and traffic interactions to generate probabilistic race outcomes and identify robust strategy solutions under uncertainty.

2.2. Objectives of the project

- Identify and quantify key performance indicators (KPI's): Identify the most significant parameters regulating endurance race outcomes, including tyre degradation, mass and fuel effect, pit stop time penalty, and traffic effect.
- Develop a deterministic lap-by-lap simulation model: Develop a MATLAB simulation program to simulate the lap time evolution over a race with degradation curves, fuel load profiles, and pit stop strategy.
- Extend the model with stochastic methods: Apply Monte Carlo simulation to introduce randomness into pit stop durations, tyre wear, traffic interactions, and rare race accidents to enable probabilistic distributions of outcomes to be simulated.
- Validate against historic data: Compare the model outputs with endurance racing data sets in terms of accuracy, realistic replication of lap time evolution, race times accumulated, and strategic decision points.
- Optimise and assess strategy robustness: Utilise the probabilistic model to contrast various tyre and pit stop strategies on performance expectations and for strategy robustness under various race conditions.

2.3. Project timeline

See table 1 for a timeline created for this work to allow optimal progression with clear goals throughout the project. Failure to associate work progression with table 1 may result in poorly validated systems that do not meet the specified performance criteria. Also, if work is not clearly documented throughout the progression of the project, there is a risk of lost methods which will disrupt workflow causing delays down the line.

Table 1 - Project timeline

Task	Start date	End date
Data collection and track selection	01/06/2025	07/06/2025
Baseline simulation creation	08/06/2025	11/06/2025
Develop tyre degradation model	12/06/2025	21/06/2025
Develop Monte Carlo framework	22/06/2025	27/07/2025
Develop push/maintain tyre model	28/07/2025	15/08/2025
Solver validation	16/08/2025	
Write up report	01/08/2025	05/09/2025

3. Literature review

3.1. Monte Carlo simulation

The use of MC simulation extends beyond the motorsport industry, it has applications across numerous disciplines including finance, project management, physics and typical engineering (Kenton, 2024). These simulations use varying inputs and provide probabilistic insights into potential outcomes, showing their value for fields where there are uncertainties, see figure 1 for a Monte Carlo schematic (Polat, et al., 2021).

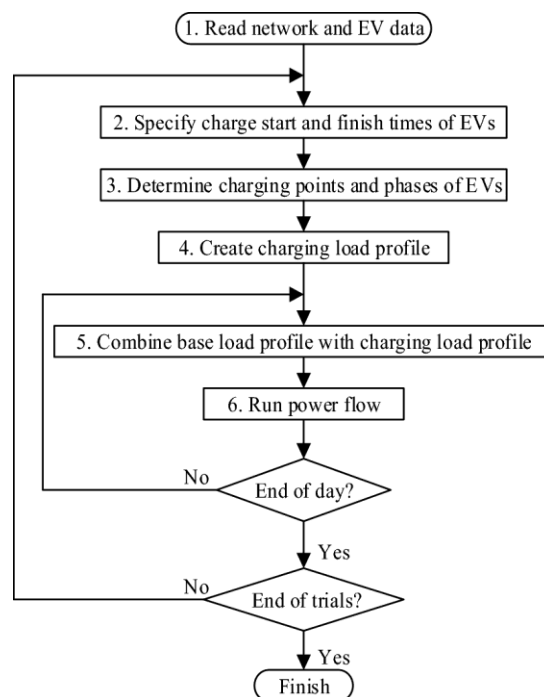


Figure 1 - Example Monte Carlo working schematic

Within motorsport applications, it's used to simulate millions of potential race outcomes allowing analysis of them to optimise strategy and approach accordingly. The simulation works based off initial identification of input variables and definition of the problem. Each of the input variables will then be assigned probability distributions based on historical data, experimental data or estimates. Distributions used in the simulation vary from normal distribution, uniform distribution and exponential distribution. A simulation will then generate random sample values and simulate the outcomes. Each of these simulation iterations would generate one possible race outcome, which will then be aggregated to analyse the probabilities of each outcome. From this the engineer predicts the most likely outcome based on the number of times each is seen.

There are three classes of Monte Carlo method that are prevalent in the literature: Classical Monte Carlo simulation, Monte Carlo integration, and Markov Chain Monte Carlo (MCMC). They differ in their principles and development. Classical Monte Carlo tends to use

independent random samples for iteratively simulating a process or a system (Taboga, 2025). Monte Carlo integration is one specific use of Monte Carlo sampling to calculate definite integrals or expected values computationally (Kong, et al., 2003). Markov Chain Monte Carlo generates an interdependent sequence of samples (a Markov chain) with the goal of converging toward a target distribution that is difficult to sample directly (Taboga, 2025). These techniques have found use in an enormous number of applications, from motorsport strategy optimisation to risk analysis for engineering, finance, and physics simulations, wherever predictive modelling under uncertainty is required.

3.1.1. Classic Monte Carlo

Classical Monte Carlo simulation is the use of uncorrelated random sampling for simulating and analysing complex systems involving stochastic elements. A classic definition is that Monte Carlo simulation "uses random sampling to study properties of systems with components that behave in a random fashion". You build a computational model of the system, then run it repeatedly with randomly sampled inputs, and observe the results. If one aggregates the outcomes of many simulations runs, one can generate an estimate of the probability distribution of outcomes and quantities of interest.

Classical Monte Carlo relies on the law of large numbers and random sampling theory. To estimate an unknown, you can draw N independent samples and take the empirical average. For large N , the sample mean will be close to the true expectation with probability 1 (law of large numbers). A common use is to estimate the probability $P(A)$ of some event A by simulating the system many times and measuring the proportion of trials on which A occurs (Song, et al., 2025). Similarly, to estimate an expected performance metric, one computes the average outcome across simulations. Each simulation run is independent, which gives unbiased estimators; and you can measure the uncertainty in estimations (e.g. via confidence intervals of narrowing width as N becomes large, by the central limit theorem) (Kwak & Kim, 2017) (Fisher, 2011). N describes the total number of trials you run in a Monte Carlo simulation, the more runs you do, the higher accuracy achieved however this leads to a longer running time for the simulation. $P(A)$ is the chance of a specific event occurring, such as a crash or safety car deployment in this context, if A = rain during race and $P(A)$ may be 0.3, meaning a 30% chance of this occurrence, where A is a specific event or outcome.

Classical Monte Carlo simulation is a straightforward but powerful technique: build a stochastic model, draw random inputs, and iterate to observe the variety of results. It is both simplicity and universality that render it powerful; it can be applied to virtually any system that is computable. Nonetheless, it may be computationally expensive if high precision is needed, as error reduction by a factor of 2 is typically achieved using approximately 4 times as many samples. The more advanced variants find their root in Classical Monte Carlo.

3.1.2. Monte Carlo integration

Monte Carlo integration is a specific application of the Monte Carlo method to numerical integration problems. It involves using random sampling to approximate the value of definite integrals, especially in high-dimensional spaces where traditional deterministic quadrature is infeasible. In Monte Carlo integration, one interprets the integral of a function as an expected value and estimates it by randomly sampling points from the domain. To estimate a complicate integral using Monte Carlo integration you randomly sample N points for the input source Ω , then for each sample evaluate the function $f(x)$, then taking the average of these values and scaling by the size of the space, giving the estimate in equation 1 (Press, et al., 2007).

$$\hat{I} \approx Vol(\Omega) \cdot \frac{1}{N} \sum_{i=1}^N f(x_i) \quad \text{Equation 1}$$

In many practical problems (e.g. multidimensional integrals in physics or finance), the number of evaluation points required by grid-based methods grows exponentially with dimension, which is the curse of dimensionality. Monte Carlo methods avoid this explosion by sampling randomly; their convergence rate does not depend on dimension, making them scalable. An illustrative example is using Monte Carlo to estimate π by randomly throwing points in a square and checking what fraction land inside the inscribed circle. Each random throw provides a probabilistic estimate of the area, and the fraction inside the circle approximates $\pi/4$, converging as more points are sampled, see figure 2 (Pipis, 2019).

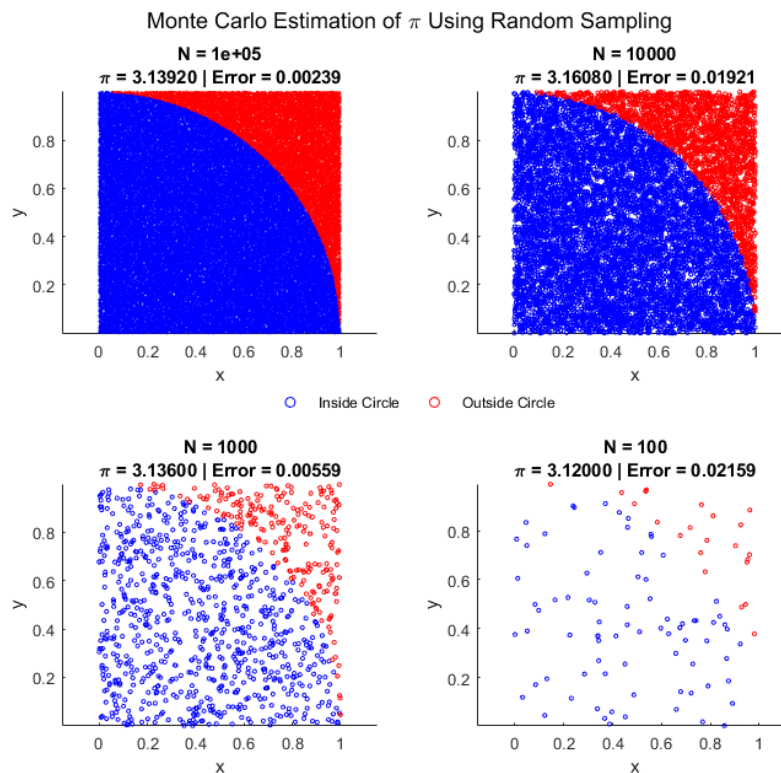


Figure 2 - Estimation of π using Monte Carlo integration from the area of a circle (Pipis, 2019)

Currently, Monte Carlo integration versions underlie advanced simulations – e.g., rendering and computer graphics, light transport integrals are calculated through Monte Carlo sampling, and quantitative finance, complicated derivative prices can be expressed in terms of high-dimensional integrals calculated through Monte Carlo. Current research still strives to improve the efficiency of Monte Carlo integration, for example, demonstrated acceleration of Monte Carlo integration for particle physics using GPU computing (Carrazza & Cruz-Martinez, 2020). The accuracy of Monte Carlo integration can be enhanced with techniques such as importance sampling and stratified sampling or Latin hypercube sampling. These reduce variance without introducing bias in the estimation (Glasserman, 2004). Nonetheless, one of the benefits of Monte Carlo integration is that its error grows as $\sqrt{N}$ regardless of the dimensionality of the integrand, so it is a method of preference for integrals over dozens or hundreds of dimensions where deterministic techniques are not applicable. $\sqrt{N}$ describes how the error (or uncertainty) in a Monte Carlo estimate decreases when the number of simulations increases.

3.1.3. Markov chain Monte Carlo

MCMC algorithms are a collection of algorithms that sample from high-dimensional probability distributions by implementing an astutely constructed stochastic process. Whereas classic Monte Carlo samples in an independent manner from a given distribution, MCMC is used for situations where sampling directly is impossible but the probability density can be estimated for different states. The idea is to construct a Markov chain whose target distribution is stationary, meaning that after some number of steps, the state of the chain is effectively a sample from the target distribution. In contrast to plain Monte Carlo, with independent draws, MCMC produces a correlated series of samples, each new sample is determined from the last one, so successive samples are correlated (Taboga, 2025). With lots of iterations, however, the chain "forgets" the starting point and the distribution of the states will converge to the target distribution under the right conditions. The sample dependence is the expense of drawing from very complicated, high-dimensional distributions which otherwise are intractable. Today, MCMC is a cornerstone of Bayesian statistics and is used in a vast range of fields to perform inference from posterior distributions that have no closed-form expression (Hogg & Foreman-Mackey, 2017).

The Metropolis–Hastings algorithm is a technique for obtaining random samples from complex probability distributions, especially when direct sampling is challenging. It achieves this by creating a sequence of dependent samples that eventually follow the desired distribution (Luengo, et al., 2020). The algorithm begins in state x_0 , at each step, it suggests a new state y , as a function of the current state, based on a rule known as the proposal distribution. The algorithm accepts or rejects the suggested state, probabilistically, in relation to the current

state. The decision is taken based on an "acceptance probability," giving more probable states higher chances of being picked more often but without completely excluding the less probable ones. Once a new state is accepted, it is employed as the next state in the chain; otherwise, the algorithm stays at its current state. This is continued many times to develop a chain of samples. After an initial period is thrown away, the rest of the samples can be used to approximate properties of the target distribution. This is an effective method as it allows one to sample from complex, high-dimensional distributions through a series of steps that search the space incrementally, even when direct sampling is not feasible.

MCMC is most prevalent in Bayesian statistics and any discipline requiring integration over complex distributions. For instance, in Bayesian machine learning, one might use MCMC to sample from a posterior distribution of model parameters given. In physics, MCMC methods remain vital in simulating systems in equilibrium. The strength of MCMC is its generality: it can sample from virtually any distribution if we can compute a criterion to accept/reject moves. Its weakness is the reliance on long chains and correlation between samples. It can be computationally slow to get a large effective sample size of independent draws, especially if the chain mixes poorly. Diagnostics for convergence and mixing must be used to ensure the chain has adequately explored the target distribution (Vehtari, et al., 2020).

MCMC is most common in Bayesian statistics and any field that needs integration with complicated distributions. An example from Bayesian machine learning is that one could use MCMC to take samples from a posterior distribution of model parameters given. In physics, MCMC techniques continue to be important in simulating systems at equilibrium. MCMC's strength lies in its generality: it can sample from essentially any distribution if we can calculate an acceptance/rejection criterion for moves. It is weak in its dependence on long chains and sample correlation. It may be extremely slow to achieve an effective sample size of many independent draws, and this will be particularly so if the chain mixes slowly. Convergence and mixing diagnostics must be employed to determine whether the chain has explored the target distribution sufficiently.

3.1.4. Motorsport applications of Monte Carlo simulation

In motorsports, specifically high-level racing like Formula 1 or championship endurance racing, Monte Carlo simulation is one of the most important methodologies used in the formulation of race strategy. A Monte Carlo solution allows the team to consider the inherently random elements of a race – e.g. variation of lap time, safety car deployment, variation of weather and accidents by simulating the race tens of thousands of times with randomised variation in these elements. The result is not one projected outcome but an array of possible results, from which strategists can gauge risks and likelihoods of success for any given strategy (Heilmeier, et al., 2020).

The motorsport industry has discussed openly the use of Monte Carlo simulations. To put this in context, McLaren's racing director describes how after generating high-fidelity models of tyre performance, pit losses, overtaking, fuel consumption and other factors, teams "use a statistical analysis method called 'Monte Carlo' to simulate a race" and obtain the probability distribution of all outcomes (Hatton, 2024). Strategists can try out the robustness of strategies by varying inputs and re-simulating. In essence, classical Monte Carlo simulation provides the basis for strategy modeling: it approximates how often a strategy would lead to victory or a given outcome in many random realisations of a race. There is recent literature confirming this approach. A 2020 article depicted how Monte Carlo simulation enables race strategy resilience to be checked, giving engineers probabilistic ground rather than a single deterministic forecast (Heilmeyer, et al., 2020).

Quasi-Monte Carlo methods have come to the racetrack too, teams sometimes employ low-discrepancy sequence sampling to ensure that they sample the space of circumstances more uniformly than ordinary random samples. Hatton refers to using a "quasi-Monte Carlo" method to systematically sweep through different opponents' strategies and tyre deterioration scenarios to enhance the learning derived from simulations (Hatton, 2024). MCMC is not explicitly mentioned as a technique used in racing simulation because the required random events are often simply able to be sampled independently without requiring the use of an MCMC approach.

3.1.5. Strengths and weaknesses of Monte Carlo simulation.

Monte Carlo simulation is central to predictive modelling for endurance motorsport strategy, a sound foundation for analysis and management of the principal uncertainties that influence racing results. In races like 6-, 12-, or 24-hour races, teams are facing key decisions, in a work environment that is dynamic and uncertain. Some of these decisions include when to make a pit, which tyre compounds to deploy, how much to stretch the stints, and how hard to push fuel and pace. Key variables such as tyre wear rates, fuel use, congestion from slow vehicles in multi-class classes, weather uncertainty, and driver fatigue all introduce substantial uncertainty into planning. Monte Carlo methods avoid this complexity by simulating the entire race process hundreds of times with randomly chosen different values for these variables to create a probability distribution of results. This enables engineers and planners not only to measure the predicted result of a particular strategy, but also the variability and risk profile. Whereas other approaches provide one deterministic result, Monte Carlo simulation gives the set of potential outcomes. This probabilistic data can support more informed decision-making, particularly when comparing strategies which may achieve an equivalent performance on average but deal with extreme or unusual circumstances in very different. Furthermore, Monte Carlo methods are best suited to sensitivity analysis and "what-if" scenario simulation.

Strategists can alter single parameters, such as tyre wear rate assumptions or weather conditions, and re-simulate quickly to observe how these changes affect the optimum course of action. The result is a highly flexible tool that not only identifies the optimal strategy under conditions as forecasted but also highlights strategies more robust to deviations from the forecasted.

Yet, Monte Carlo simulation depends greatly on the quality of the input distributions and the models used. Poorly estimated inputs, such as over-simplified degradation curves or unrealistic incident frequency assumptions, can yield misleading outputs regardless of the number of simulations run. This underscores the necessity of integrating good-quality historical data, domain-expert knowledge, and perhaps other modelling methods, such as Bayesian inference, to calibrate the inputs more precisely. Computational demands of Monte Carlo simulations can also be significant, especially for full-race simulations with a wide range of stochastic variables. But quasi-random sampling, variance reduction, and parallel processing are all techniques that can be used to reduce performance bottlenecks and enhance simulation efficiency.

In practice, Monte Carlo simulation is an everyday utility in racing strategy both for its ability to deal with multi-factorial uncertainty but also because it is open and responsive. It allows engineers to talk in terms of the trade-offs and confidence levels associated with different strategies and to incrementally refine their models as more data are obtained. Coupled with optimisation techniques or artificial intelligence methodologies, however, Monte Carlo frameworks can be made fully adaptive decision-support tools responding to actual race conditions.

3.2. KPI's for simulation

4 KPI's are found that effect race outcomes for sprint racing and endurance style events, these being tyre degradation, fuel consumption and mass effect, pit stop losses and traffic and multiclass interference.

3.2.1. Tyre degradation

A major influence on race outcomes and strategy is tyre degradation which must be carefully considered to perform successfully. Tyres effect grip when cornering, along with braking performance and power out of a corner when grip may be on the limit (Heilmeier, et al., 2020). Fresh tyres must be treated carefully as pushing grip levels immediately will enhance the degradation seen and the driver may not be able to use the full potential of the tyre for the desired amount of time as was intended. This leaves difficult decisions for race engineers to decide over which is where a Monte Carlo simulation can prove extremely important.

Along with this, certain racing series will impose regulations around the number of tyres used, including how many of a certain compound is allocated throughout a race weekend, which leaves another area of investigation. For example, in the FIA world endurance championship (WEC) 2025 prototype hypercar teams face tyre allocation limits that include 12 dry tyres during free practise sessions, 4 tyres specifically for hyperpole qualifying and 18 slick tyres during combined qualifying and race for a 6-hour race period. These allocations then increase for longer race periods to 26 for 8-hour races, 32 for 10-hour races and 56 dry tyres for a 24-hour event such as Le-mans (FIA , 2025).

These regulations force teams to keep a close eye on tyre life over the whole weekend. Should compound allocations or limits be consumed too rapidly, strategists will be forced to push hard on sub-optimum tyres or unloved double-stinting options, tyres already run being recycled, losing their grip and consistency (Mobile Pitstop, 2022).

3.2.2. Fuel consumption and mass effect

As fuel consumption and the reduction in mass because of it directly effects lap times, there is a necessity for a direct correlation between the two (Bekker & Lotz, 2009) (Boettinger & Klottz, 2023). Typical strategies of projection are based on a model of fuel consumption per lap and project how many laps can be run per tank, ultimately deciding optimal refuelling schedule (Kampen, et al., 2022). This shift is subtle over a few laps but becomes more significant in longer stints. Modelling race strategies using discrete-event simulation shows measurable gains in race time through accurate fuel/mass-lap time correlations. Boettinger and Klottz noted lap time improvements of up to 0.3 seconds per lap over a full fuel stint.

From a simulation standpoint, this relationship is typically implemented as a penalty coefficient per unit fuel mass, derived from historical data or track-specific regression models. Simulations iterate the vehicle's fuel load on a lap-by-lap basis, recalculating lap time based on the mass effect. As an example, a linear penalty model might assume that every 10 kg of extra fuel adds 0.05 s penalty per lap—a 100 kg fuel load would then offer a 0.5 s penalty per lap compared to an empty tank. However, depending on circuit layout and car configuration, this is usually a non-linear relationship, particularly for high-altitude or drag-sensitive circuits.

3.2.3. Pit stop loss

Pit stops are critical to race strategy, introducing a discrete time penalty to simulations when a vehicle is stopped and for the time lost due to pitting which differs between tracks, where different pit limits are enforced along with different pit lengths. Accurately quantifying pit stop loss is therefore essential in race simulation, as it allows comparison between multiple strategy options (e.g., no-stop, one-stop, two-stop) and helps define the break-even points where the performance gains from fresh tyres or reduced mass outweigh the time spent in the pits (Garcia, 2024) (Statathlon, 2016).

Moreover, tyre allocation rules and refuelling time limits imposed by regulations affect how much time is lost during each stop. For example, in WEC, refuelling and tyre changes must occur sequentially, extending total pit duration compared to simultaneous operations (WEC, 2018). Accounting for these constraints ensures the simulation reflects the strategic impact of the rulebook.

There is also the difficulty of needing to perform two pit stops near each other from a team standpoint where the team decides to 'double stack' a pit stop (A team with multiple race cars being cared for by a single team of mechanics meaning pit stops for both cars cannot happen in parallel). Meaning both teams' drivers will enter the pits after each other where the team will need to time entries as to not delay the trailing vehicle in their pit. There is also an increased chance of mistakes to be made from the mechanics performing the pitstop due to the increased complexity caused by the double stack. This level of complexity is likely not to be considered for this investigation as the focus will remain around the optimisation of a single vehicle strategy rather than team strategy, yet it is still important to understand this concept.

3.2.4. Traffic and multiclass interference

Current research suggests time lost behind slower moving vehicles can cost losses in lap times of seconds, meaning pitting strategy must be implemented to avoid heavy traffic can be suggested. Paparusso's research in this area inputs traffic delays probabilistically, yet other models use multi-agent modelling or game theory to simulate interactions (Paparusso, et al., 2022). This area is still underdeveloped within the motorsport industry and would require new work to contribute towards a suitable methodology.

In multi-class racing or events with pace staggered throughout the field, traffic interaction is a non-negligible source of lap time variance. Faster cars will lose a few tenths of a second per lap, or more, as a function of where on the track they catch slower traffic. Time loss is typically worst in medium- to high-speed corners where overtaking opportunities are not plentiful. This can be added to the Monte Carlo simulation model by introducing traffic interference as a probabilistic time penalty, added to laps where traffic encounters are likely based on historical or synthetic timing data.

Recent research investigates multi-agent reinforcement learning (MARL) and cooperative game theory as a method for traffic modelling. These models simulate each driver as an agent that optimises their behaviour based on others' actions, including overtaking logic, defensive driving, and cooperative yielding. Yet the computational complexity of these techniques and training requirements makes them rather difficult to implement in a successful way that aligns with the aims for the project, where fast Monte Carlo methods provide race strategy decisions swiftly (Zhou, et al., 2022) (Liu, et al., 2023).

Pit stop timing as a strategic method of traffic avoidance is another consideration. In some race series, teams deliberately time pit stops to emerge onto the track in clear air, with less exposure to slower traffic and to maximize performance on fresh tyres. The placement of pit timing, at the start or end of slow corners or chicanes, can significantly reduce the likelihood of rejoining behind slower rivals in a traffic-heavy sector (Aguad & Thraves, 2024). This tactic bears resemblance to classic undercut and overcut techniques in races, which rely on clear-track opportunity to yield faster laps once pitted.

3.3. Race strategy techniques

Modern motorsport demands not only driver skill and vehicle performance, but also sharp strategic decision making by race engineers and strategists. In real-world racing, teams continuously adjust tactics such as pit stop timing, tyre choices, and on-track pacing to gain a competitive edge.

3.3.1. Sprint racing

One of the most discussed strategy tactics within shorter form motorsport are the undercut and overcut manoeuvres during pit stops. The undercut describes pitting a vehicle for fresh tyres prior to a rival vehicle that is ahead, aiming to use the fresh tyres faster lap time to pass the further ahead vehicle when they eventually stop for a tyre change. An example of this is during the Singapore Grand Prix where Sebastian Vettel performed an undercut running in 3rd place at the time where he eventually emerged ahead of the race leader after their respective stops (Hall, 2023). In contrast, an overcut describes a situation in which a vehicle will stay out longer than intended when a competitor has pitted, with the aim of gaining time when a rival struggles to get heat into fresh tyres or when a competitor runs into slower moving traffic. A notable case of just this calls back to the 2021 Monaco Grand Prix where Lewis Hamilton tried to perform an undercut manoeuvre on Pierre Gasly yet struggled to get heat into his tyres; Gasly, running longer on worn tyres, maintained faster laps and stayed ahead after Hamilton's stop. These examples show that undercuts are most effective when fresh tyre advantage outweighs track position, and overcuts can be effective if the undercutting car cannot immediately capitalise on its fresh tyres or if clear track is the most important consideration. Closely related is pit window optimisation, selecting the best spread of laps upon which to enter a pit stop to lose the least time and gain the most in performance. Predictive modeling and real-time information are utilised by strategists to decide when a vehicle can pit and return to racing without lingering behind slower cars (back markers). If a car stops too soon or too late in relation to rivals, it will emerge into a pack of cars and sacrifice any time it is made. Because of this, crews strategize every "clean air" window for stopping carefully and keeping room for making last-minute changes. In practice, a first-race plan will include a target pit lap,

as the race goes on, however, teams modify this strategy according to actual tyre degradation and race position. Engineers constantly check whether to extend stints, shortening them, or switch to a different approach if situations change. The goal is to execute pit stops now that realises the maximum net gain. An off-timely stop can be worth several seconds, enough to gain or lose a position, and meticulous planning and split-second coordination are necessary. As Red Bull Racing sporting director stated, decisions on when to pit need to be made sometimes ahead of time so that when the moment is there "everybody [knows] what they [are] doing and the team can act in seconds" (Red Bull Racing, 2022).

Another key element of sprint race strategy is tyre compound selection and management. In Formula 1, for instance, Pirelli provides a number of rubber compounds each weekend, and teams must decide which one to start on and when to switch, compromising performance for longevity. Softer tyres are faster over a single lap but wear more, while harder tyres last longer but at the expense of pace, see figure 3.

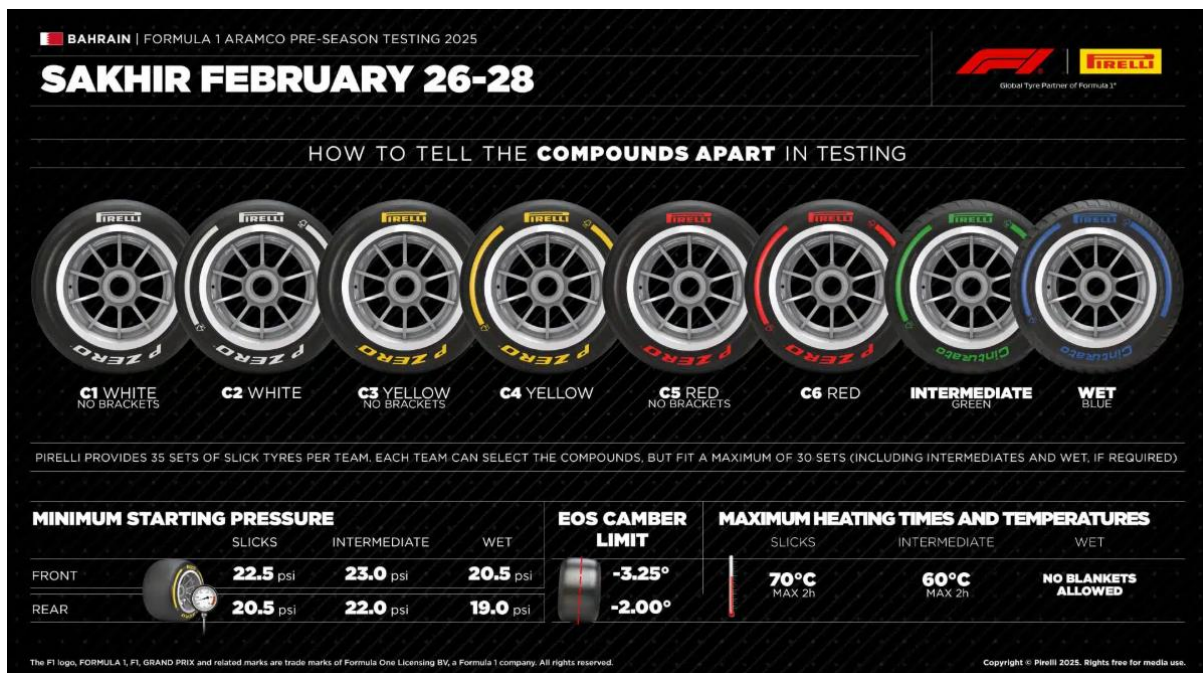


Figure 3 - Pirelli tyre range and operating limits for 2025 pre-season testing at Sakhir (Motor Sport, 2025)

Teams simulate practice data to estimate stint lengths and lap time trade-offs for each compound and devise strategies like starting on a softer compound to get early track position or, on the other hand, starting on a harder tyre to go longer and have flexibility later. These choices are also affected by race regulations, e.g., in F1 a dry race must utilise at least two different dry compounds, effectively mandating at least one pit stop for tyre change. The regulation is to avoid a "no-stop" race and to force strategists to include tyre changes in their strategy. Typically, the fastest race strategy in recent years has been a one-stop, with variations depending on track type and degradation levels. Heavy-wear or hot circuits can put

teams into a two-stop strategy to avoid experiencing too much tyre drop-off, whereas one-stop is sufficient at lower-wear events if managed properly. Occasionally F1's governing body experiment with rules to bring extra spice to tyre strategy, like a 2025 Monaco rule that compelled drivers to run three sets of tyres, essentially requiring a minimum of two pit stops (Reuters, 2025). All these innovations are aimed at encouraging increased variability and passing chances with different tyre strategies. Even without rules, top teams place huge value on choosing the optimum tyre for each stage of a race. By choosing the right compound for developing track conditions and changing over at exactly the right time, strategists attempt to keep their cars in the ideal performance range of the tyre for as long as possible.

3.3.2. Endurance Racing

One of the most unique features of endurance strategy is having multiple drivers per vehicle and organising driver stints. For instance, the FIA WEC generally accommodates up to 2–3 drivers for a 6 or 12-hour race and strictly mandates three drivers for a 24-hour race. The teams plan out in advance who is driving in the car for each segment, controlling driver rotations is essential to ensure that each driver remains well-rested and refreshed. Rules limit how long a specific driver can drive consecutively, for example, Le Mans regulations historically have a maximum of 4 consecutive hours driving by any driver within a 6-hour window, and approximately 14 total hours (FIA, 2025) (Piccinotti, 2020). Teams usually coordinate driver swaps with fuel stops, switching drivers often every double stint or at least every few fuel loads to keep additional pit time low (Circuit of the Americas, 2024). One of the strategies is to have drivers rotated so that they get the most out of their performance: no driver must stay tired in the car, and every driver gets sufficient rest before their next stint (Campbell Racing, 2024). Teams will have a pre-planned stint plan, but this can be adjusted if conditions call for it. If the specific driver performs better in night conditions especially, the team might allocate more nighttime to them, while a less experienced teammate might do more daylight running. Night-to-day rotation planning enters play at events like the 24 Hours of Le Mans or Daytona. Nighttime racing adds other issues such as less visibility, lower track temperatures and fatigue. Driver consistency is also something that engineers account for: a slower but faultless driver may be the best bet in the middle of the night to conserve the car, holding off the fastest driver until a late-night charge. Fatigue management is critical; a fatigued driver will make more errors costing time or ending races in crashes. As one guide to endurance racing points out, drivers will be rotated so each is "rested and ready to run at their best during their stints," since a tired driver is likely to produce slow laps or mistakes (Campbell Racing, 2024). This human aspect of strategy, managing drivers' work-rest cycles and taking

advantage of their personal strengths, is as important as any mechanical aspect in endurance racing.

Another pillar of endurance strategy is fuel and pit stop management, which involves the choice of when to fuel up, how aggressively to drive each stint, and how to keep time in the pits to a minimum. In sprints where a single pit stop can win or lose the race, for endurance races teams need to carry out numerous pit stops perfectly throughout the duration of the event. All pit stops are refuelling and may include tyre changes and driver change. Total time spent in pit lane can easily run into many minutes in a long race, so the procedure must be economical: getting the pits done as fast as possible without immobilising the car (Campbell Racing, 2024). Teams plan out how many pit stops based on fuel consumption rates, tank size, and race duration. One of the key calculations is the number of laps the car can get to run on a single tank of fuel and whether placing one extra lap to that trip can decrease the number of stops. During a 24-hour race, saving one extra lap of fuel per run can be worth a great deal, for example, one lap longer between stops can save one full stop by the end of the race, easily 60–90 seconds (24H Le Mans, 2013). It has been shown historically by the 24 Hours of Le Mans: "the mere margin of a lap between refuelling stops can save 2 stops in the 24 hours, i.e., around 2 minutes" saved at pit stops. Therefore, teams constantly examine the trade-off between speed and fuel efficiency. If a car can lift-and-coast to save fuel without losing too much time on track, it could possibly do more laps per stint and thus require fewer refuelling stops. A rule for an endurance strategist is often "go fast enough to be in the lead, but slow enough to take the finish." As an example, on long green-flag runs, engineers will tell drivers when to push hard and when to take it easy to achieve fuel goals (Campbell Racing, 2024). New telemetries allow real-time fuel burn monitoring; drivers adjust that data by changing engine modes or driving style.

One of the odd challenges of endurance racing is working through multi-class traffic. In WEC or IMSA races, faster prototype classes mix with slower GT cars, so even the leader car is constantly passing backmarkers. This reality has a big effect on strategy and race craft: time lost in traffic will accumulate to erode a lead or prevent a pursuer from closing a gap. Drivers are trained by teams to manage traffic well, when and where to pass slower cars with the minimum amount of lost time. For instance, a Hypercar driver at Le Mans will ease off before a series of corners so that they don't surprise a GT car and then use better acceleration to pass the next straight. This takes discipline but can reclaim seconds otherwise spent getting stuck in tight spots in the back of a slower-moving car. Strategists even plan to stop for tires to clear the car from a bad traffic position. Traffic impact is also track-dependent: on the 13.6 km Le Mans circuit, traffic is diluted over the long lap in part and there are a lot of long straights upon which to overtake. A Ferrari Hypercar driver remarked that at Le Mans, the circuit's vast

size and multiple long straights mean slower traffic is rarely a major hindrance—overtaking opportunities abound. ‘It’s not like Imola,’ he noted, where you can remain stuck behind slower cars for ‘4–5 turns’ with little chance to pass (Ferrari, 2023).

Hence, traffic management is a strategy component: crews attempt to minimise wasted time in traffic via driver awareness and sometimes through the timing of the stints. On evening runs, traffic management is even more challenging where visibility is worst, drivers are tired, and it is harder to estimate closure rates. As one Le Mans winner explained, at night "your concentration starts to slip a bit, and the traffic starts getting more difficult to manage as well". This is combated by reminding the drivers on the radio of approaching groups of cars and by placing their best, most alert drivers in heavy traffic periods. Finally, getting through traffic unscathed and without losing much time is a win-or-lose factor in endurance races, and it depends on planning and the ingenuity of the drivers' on-track expertise.

4. Methodology

This research employs a simulation-based approach to model and optimise endurance race strategy, with MATLAB used as the primary environment for numerical modelling, data processing, and iterative analysis. The model is constructed as a lap-by-lap simulation, incorporating key performance variables such as tyre degradation, fuel consumption, and pit stop losses, which collectively influence race time. For handling uncertainties typical in racing, such as variable lap times, strategic variability, and fuel effect, Monte Carlo simulation framework will be introduced to the deterministic model. It can provide probabilistic race outcomes by repeatedly sampling distribution variables until good strategy analysis and optimisation can be enabled. Similar approaches have been used successfully in motorsport and operations research contexts, such as discrete-event simulations for F1 pit strategy (Bekker & Lotz, 2009), competitor-aware modelling for hybrid race vehicles (Paparusso, et al., 2022), and real-time predictive models for electric endurance cars (Kampen, et al., 2022).

4.1. Initial model building

4.1.1. Race selection

To initiate model development, a test case must be studied and compared to. In this application, the 2025 formula 1 round at Suzuka is selected, specifically looking at Oscar Piastri's race, qualifying and testing data to draw comparisons from (FIA, 2025). This allows a controlled environment to begin model development, benchmarking and validation.

4.1.2. Data extraction and processing

Data is gathered through the FIA documentation posted to their website for public access (FIA, 2025). Lap times are extracted, and baseline lap time is calculated through the average time of laps from free practise 1 (FP1), excluding invalid times such as in/out laps or laps where significant traffic or mistakes occur. A pit stop loss is estimated through the difference between the in lap of Oscar Piastri to the mean of all other laps done throughout the race. These calculations and extractions are shown through figure 4.

```

1  clc,clear,close all
2  laps = 53;
3  op81lapsfp1=readmatrix...
4      ("op81_suzuka2025.xlsx","Sheet","fp1","Range","D4:D27");
5  op81lapsrace=readmatrix...
6      ("op81_suzuka2025.xlsx","Sheet","race","Range","D4:D55");
7  validlaps=op81lapsfp1(op81lapsfp1<100);
8  av_lap=mean(validlaps);
9  pitloss=op81lapsrace(20)-mean(op81lapsrace([1:19,21:end]));

```

Figure 4 - Extraction of race and free practise 1 lap times and calculation of baseline lap time along with estimation of pitloss

4.1.3. Baseline degradation model

To begin a baseline degradation model is made through analysis of Lando Norris's race lap times from the 2024 Singapore GP. With analysis of figure 5 (FIA, 2024), between laps 10 and 28, a visible trend of increasing lap times is seen, this initial drop in lap times between laps 1 and 10 is from the tyre warm-up phase, reinforced through the fact of Norris beginning this race on a set of used soft tyres, see table 2 (Pitpass.com, 2024). By isolating the lap times between laps 10 and 28, a rough per lap degradation percentage can be gathered, in this case a value of 0.2744% per lap is calculated.

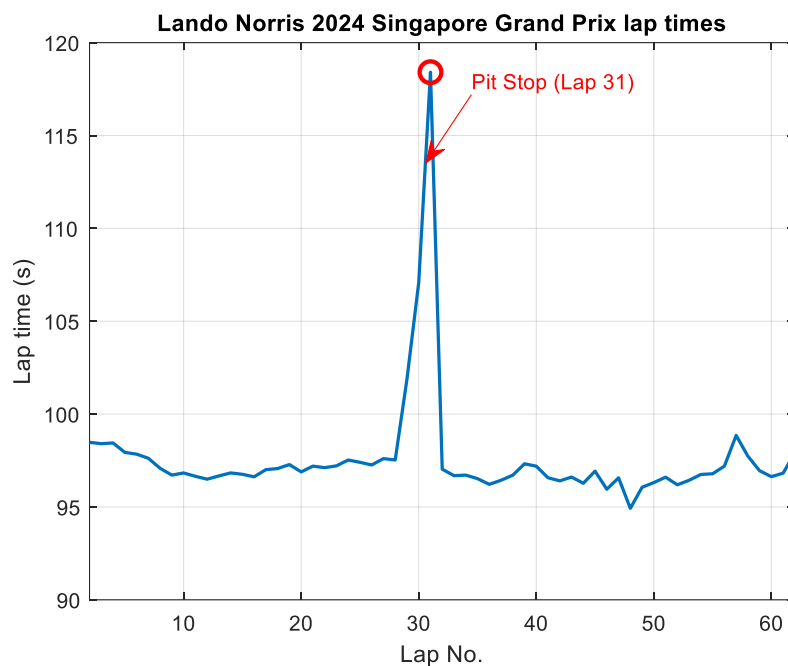


Figure 5 – Lando Norris race lap times from the 2024 Singapore GP (FIA, 2024)

Table 2 - Tyre strategies for all competitors for 2024 Singapore GP (Pitpass.com, 2024)

Driver	Team	Start	Stint 2	Stint 3	Stint 4
Norris	Mclaren	New medium	New hard	NA	NA
Verstappen	Red bull	New medium	New hard	NA	NA
Piastrri	Mclaren	New medium	New hard	NA	NA
Russell	Mercedes	New medium	New hard	NA	NA
Leclerc	Ferrari	New medium	New hard	NA	NA
Hamilton	Mercedes	Used soft	New hard	NA	NA
Sainz	Ferrari	New medium	New hard	NA	NA
Alonso	Aston Martin	Used medium	Used hard	NA	NA
Hulkenberg	Haas	New medium	New hard	NA	NA
Perez	Red bull	New medium	New hard	NA	NA
Colapinto	Williams	New medium	New hard	NA	NA
Tsunoda	RB	New medium	Used soft	NA	NA
Ocon	Alpine	New medium	New hard	NA	NA
Stroll	Aston Martin	Used hard	Used medium	NA	NA
Zhou	Stake	New hard	New medium	NA	NA
Bottas	Stake	New hard	New medium	NA	NA
Gasly	Alpine	New medium	New soft	NA	NA
Ricciardo	RB	New hard	New medium	New soft	Used soft
Magnussen	Haas	New medium	New medium	New soft	NA
Albon	Williams	New medium	New hard	NA	NA

This simple approach establishes a baseline but is clearly unrealistic over long stints, motivating the need for a more flexible degradation curve.

4.1.4. Simulation of pit strategies

Analysis of the accuracy of this degradation model is done through three separate MATLAB functions, one simulating no stops, the second for a one-stop strategy and third for a two-stop strategy (Bekker & Lotz, 2009) (Garcia, 2024). The inputs for these functions being number of laps, baseline lap time, tyre degradation, time lost due to pitting and stop laps. These functions can be seen in figure 6, figure 7 and figure 8.

```

1 function [racetime0,cum0] = nostop(laps,av_lap,tdeg)
2 %% Computes race time without stopping
3
4 % laps = Number of laps
5 % av_lap = Baseline lap time (s)
6 % tdeg = Tyre degradation percentage per lap
7 racetime0=zeros(1,laps);
8 racetime0(1)=av_lap;
9 for i=1:1:laps-1
10     racetime0(i+1)=racetime0(i)*tdeg;
11 end
12 cum0=cumsum(racetime0);
13 end
  
```

Figure 6 - MATLAB function for calculating lap times and race times for a no stop strategy

```

1 function [racetime1,cum1] = onestop...
2     (laps,av_lap,tdeg,stoplap,pitloss)
3 %% Computes race time with 1 stop
4
5 % laps = Number of laps
6 % av_lap = Baseline lap time (s)
7 % tdeg = Tyre degradation percentage per lap
8 % stoplap = Lap for pitstop to be performed on
9 % Pitloss = Stop lost per lap due to pit stop
10 racetime1=zeros(1,laps);
11 racetime1(1)=av_lap;
12 for i=1:1:stoplap-1
13     racetime1(i+1)=racetime1(i)*tdeg;
14 end
15 racetime1(stoplap)=racetime1(19)+pitloss;
16 racetime1(stoplap+1)=av_lap;
17 for i=stoplap+1:1:laps-1
18     racetime1(i+1)=racetime1(i)*tdeg;
19 end
20 cum1=cumsum(racetime1);
21 end
  
```

Figure 7 - MATLAB function for calculating lap times and race times for a one-stop strategy

```

1 function [racetime2,cum2] = twostop...
2     (laps,av_lap,tdeg,stoplap1,stoplap2,pitloss)
3 %% Computes race time with 2 stops
4
5 % laps = Number of laps
6 % av_lap = Baseline lap time (s)
7 % tdeg = Tyre degradation percentage per lap
8 % stoplap1 = Lap for first pitstop to be performed on
9 % stoplap2 = Lap for second pitstop to be performed on
10 % Pitloss = Stop lost per lap due to pit stop
11 racetime2=zeros(1,laps);
12 racetime2(1)=av_lap;
13 for i=1:1:stoplap1-1
14     racetime2(i+1)=racetime2(i)*tdeg;
15 end
16 racetime2(stoplap1)=racetime2(13)+pitloss;
17 racetime2(stoplap1+1)=av_lap;
18 for i=stoplap1+1:1:stoplap2-1
19     racetime2(i+1)=racetime2(i)*tdeg;
20 end
21 racetime2(stoplap2)=racetime2(32)+pitloss;
22 racetime2(stoplap2+1)=av_lap;
23 for i=stoplap2+1:1:laps-1
24     racetime2(i+1)=racetime2(i)*tdeg;
25 end
26 cum2=cumsum(racetime2);
27 end
  
```

Figure 8 - MATLAB function for calculating lap times and race times for a two-stop strategy

This exponential approach models the cumulative effect of tyre degradation between pit stops. Pit stop logic resets the tyre age on the designated lap/laps, thus restarting the degradation cycle from a fresh baseline. These three approaches highlight the importance of tuning the degradation model to real-world tyre characteristics. The 0.2744% degradation rate, while useful for initial testing, proves to oversimplify real tyre behaviour and leading to unrealistically exponential lap time increases, see figure 9 and 10.

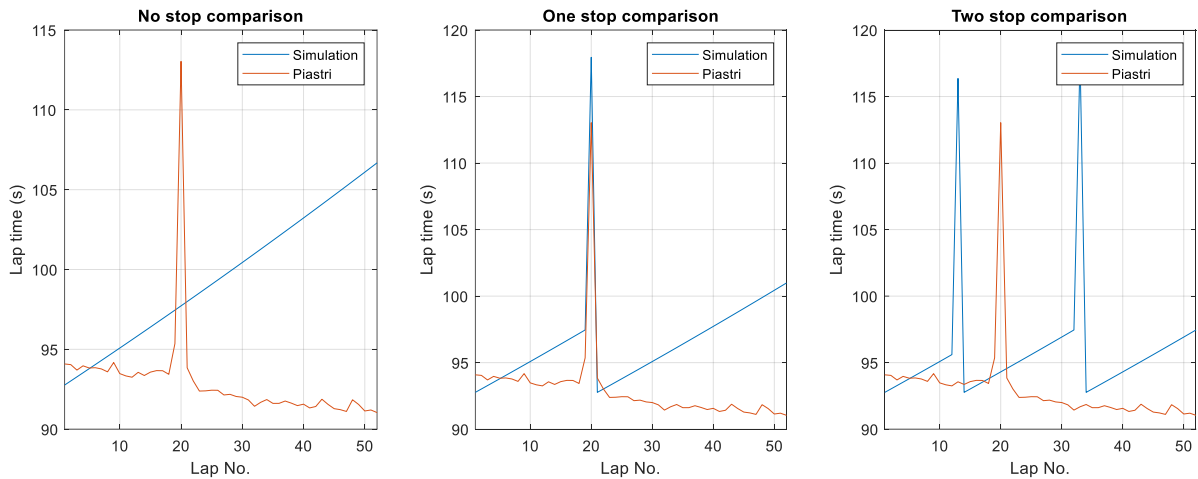


Figure 9 - Comparing baseline simulated lap times against Oscar Piastri's race lap time from Suzuka 2025

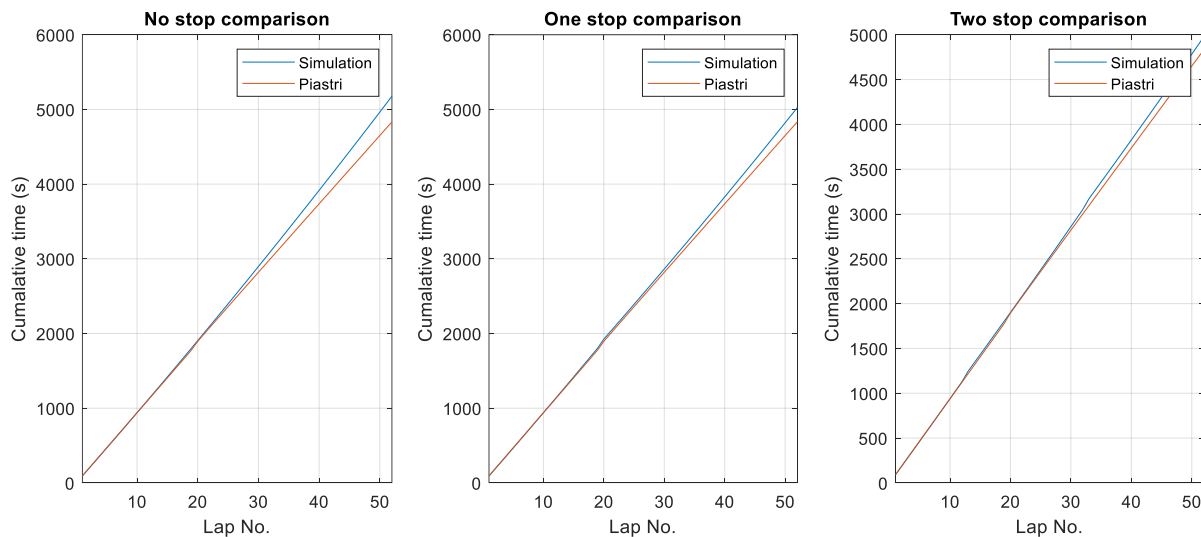


Figure 10 - Comparing baseline simulated cumulative times against Oscar Piastri's race lap time from Suzuka 2025

4.1.5. Improved tyre degradation model

This new function allows tuning of the mathematical curve through adjustments to the two components a and b , seen in equation 2, this is done to model tyre degradation in a more accurate cycle through tyre life (West & Limebeer, 2020) (Piccinotti, 2020). The resultant curve is shown through figure 11 A, this uses the coefficients $a = 1.65$ and $b = 0.3$. Further

improvements to this curve can be made through alterations made to these values, for example a more realistic curve is shown in figure 11 B, where the coefficients apply harsher degradation throughout prolonged use rather than capping degradation at 1.65 seconds, using the coefficients $a = 2.35$ and $b = 0.05$. The improved realism of degradation is displayed through figure 12 A and B.

$$deg(i) = a \times (1 - e^{-b \times i}) \quad \text{Equation 2}$$

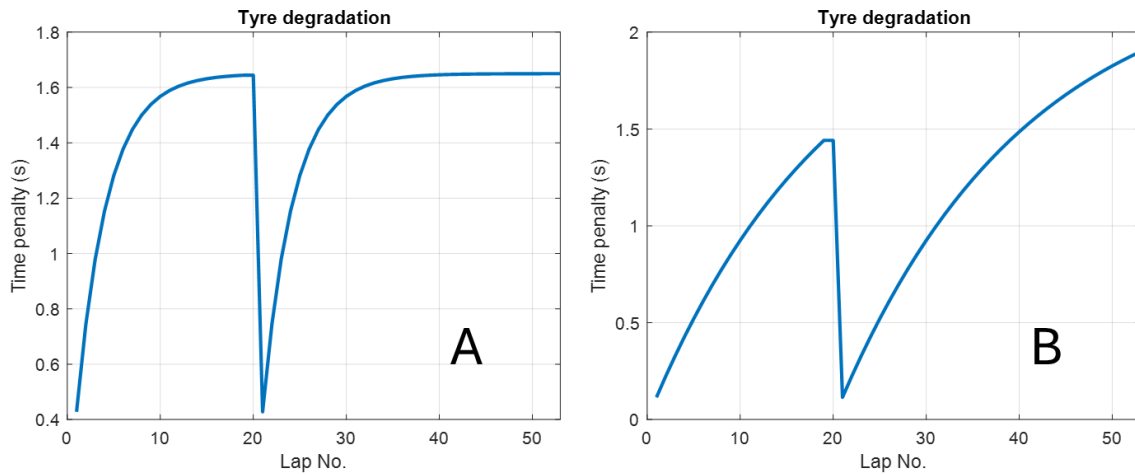


Figure 11 – (A) Improvements to tyre degradation through equation 2 (B) Adjusting degradation coefficients to $a = 2.35$ and $b = 0.05$ for better degradation curve

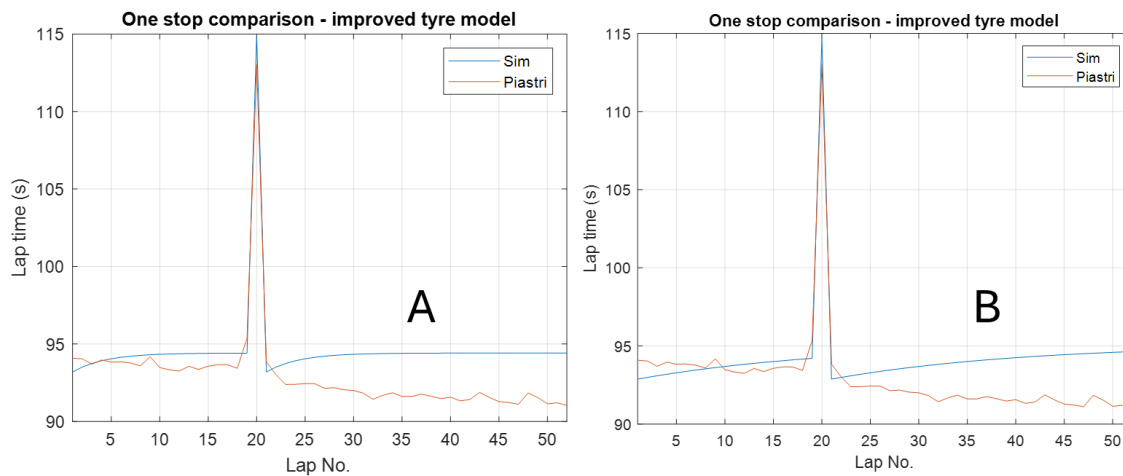


Figure 12 – (A) Lap time Improvements form adjusting tyre degradation through equation 2 (B) Lap times from adjusting degradation coefficients to $a = 2.35$ and $b = 0.05$ for improved degradation curve

4.1.6. Fuel burn modelling

Under contemporary Formula 1, refuelling during the track is prohibited, so the car begins with the complete race fuel load and becomes progressively lighter throughout the lap as fuel is consumed. This linear weight loss creates a measurable lap time benefit over the duration of a stint. While the effect is relatively minor on a single lap, it accumulates significantly through

a race distance. To simulate initially, this impact is represented by assuming a constant burn rate over the course of a lap, as shown in equation 3.

$$\text{burn per lap} = \frac{\text{Fuel mass}}{\text{No.laps}} \quad \text{Equation 3}$$

Where ‘Fuel mass’ represents the starting fuel load and No. laps being the race length, effectively assuming that over the race period every bit of fuel is burnt, leaving none left at the end. This is then applied to the simulation through a subtraction to the lap time that is justified through an effect per kg of fuel burnt. In this case through research on literature a suggested value of 0.03s benefit per kg of fuel burnt is used (Boettinger & Klottz, 2023). The effect of this is shown through figure 13, it is clear that little improvement is gained from this fuel model, leaving further investigation and improvements necessary.

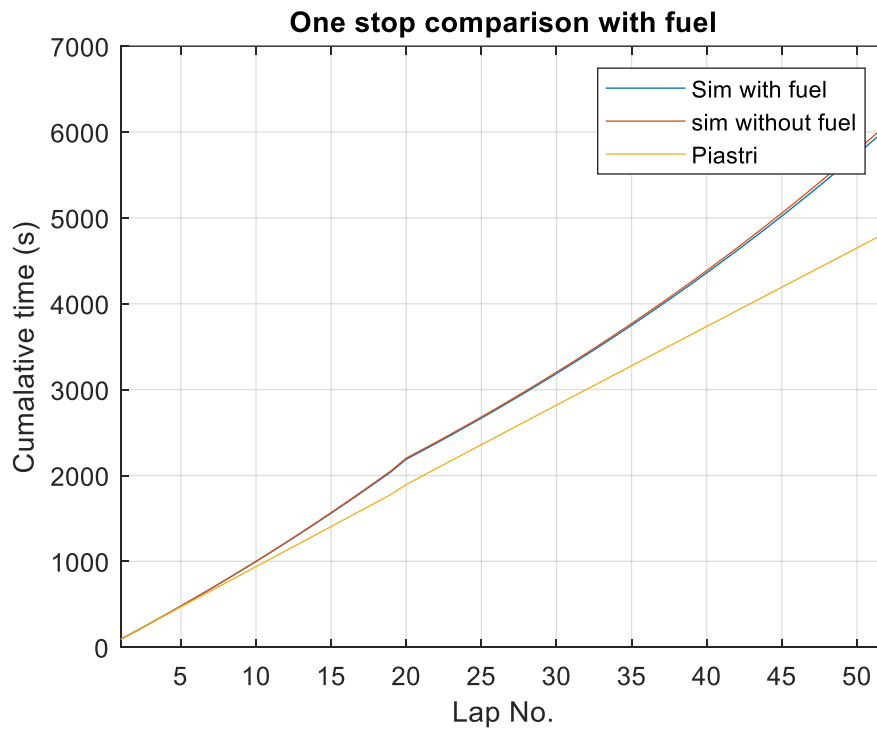


Figure 13 - Comparing simulation with fuel effects and without against Oscar Piastrri's race at Suzuka 2025

After analysis of figure 13, it is decided that a more advanced approach to fuel burn modelling is required using a logarithmic model (Boettinger & Klottz, 2023). A linear model assumes each kilogram always gives the same lap-time gain, but in practice the sensitivity per kg changes across the stint (a lighter car gets a bigger per-kg benefit late on). See equation 4 for this logarithmic fuel burn calculation.

$$\Delta t_{fuel}(i) = k \ln \left(\frac{M_o}{M_i} \right) \quad \text{Equation 4}$$

Where:

- $\Delta t_{fuel}(i)$ is the cumulative lap time loss by lap i relative to the start of the race
- M_0 Is the initial fuel mass
- M_i Is the remaining fuel mass at lap i
- k is an effect factor

This approach has a roughly linear trend over small mass changes early in the stint, but ever-increasing sensitivity with the tank near empty, as observed in on-track trends. The revised model therefore provides a more accurate representation of the fuel/mass–lap-time relationship, improving simulation fidelity to sprint and endurance strategy optimisation. See figure 14 for this curve.

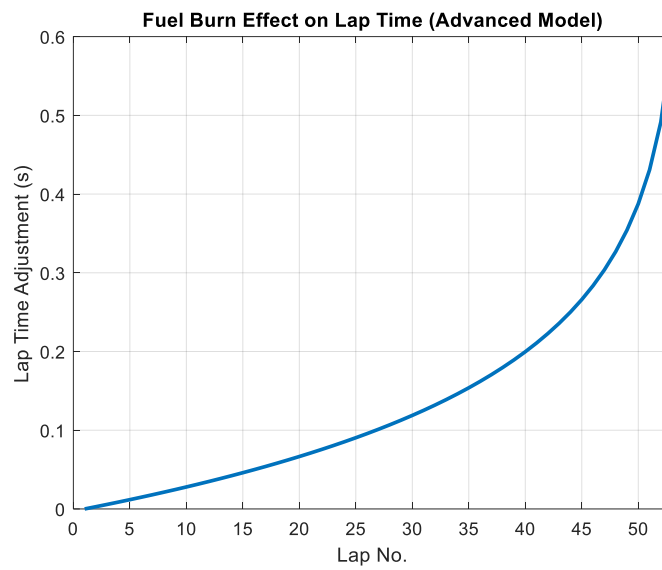


Figure 14 – Fuel effect using logarithmic model for fuel burn

With the improvements made to the fuel modelling, a better representation of fuel burn is shown through figure 15, capturing the gradual lap time reduction across the stint and aligning more closely with the trend observed in Piastri's real race data.

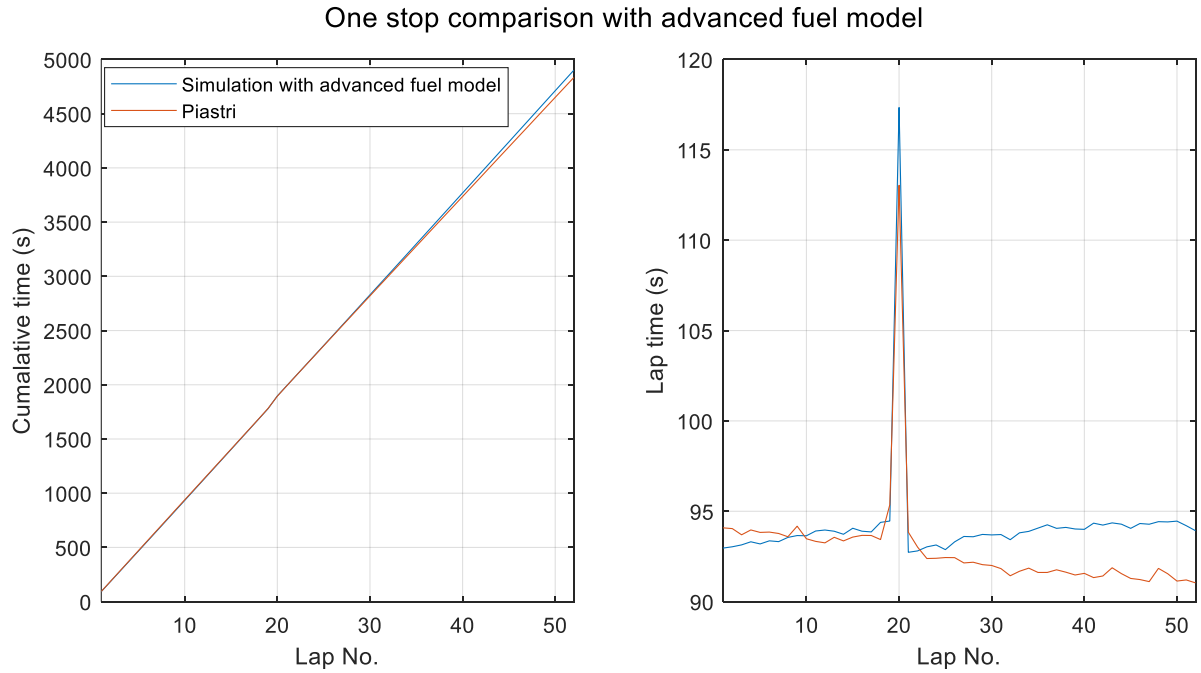


Figure 15 - Simulation comparison with advanced fuel modelling to Piastri at Suzuka 2025

4.1.7. Combined simulation (tyre degradation and fuel modelling)

Once both independent models are established, a combination of the two is made to solve the cumulative effect on lap time evolution. Providing a more realistic representation of pace development through a race, where tyre wear increases but mass decreases gradually (Piccinotti, 2020). The combined equation is shown in equation 5 where:

- t_{base} is the baseline lap time
- $\Delta t_{tyre}(i, a, b)$ is the degradation penalty per lap
- $\Delta t_{fuel}(i, k)$ is the fuel mass effect per lap
- a and b are the degradation curve coefficients
- k is the fuel sensitivity coefficient

$$t_i = t_{base} + \Delta t_{tyre}(i, a, b) - \Delta t_{fuel}(i, k) \quad \text{Equation 5}$$

These refinements mean the model now produces stint evolution curves that better reflect observed race behaviour, allowing subsequent Monte Carlo tests to yield realistic outcomes.

4.1.8. Validation of baseline model

Through analysis of figure 16, there is a good trend between actual data and simulated strategy. The main area of difference between the two is because of the safety car period towards the end of the race, as identified by the high lap times by all competitors except the simulated strategy, as this does not include the penalties of using a safety car. The close relationship between simulated and real lap time development under green-flag racing

conditions confirms that the integrated tyre degradation and fuel burn modelling produces a realistic race dynamic simulation. While the model does not yet incorporate the effect of race interruption or random variables, it establishes a solid foundation for development stages where the simulation will incorporate these variables.

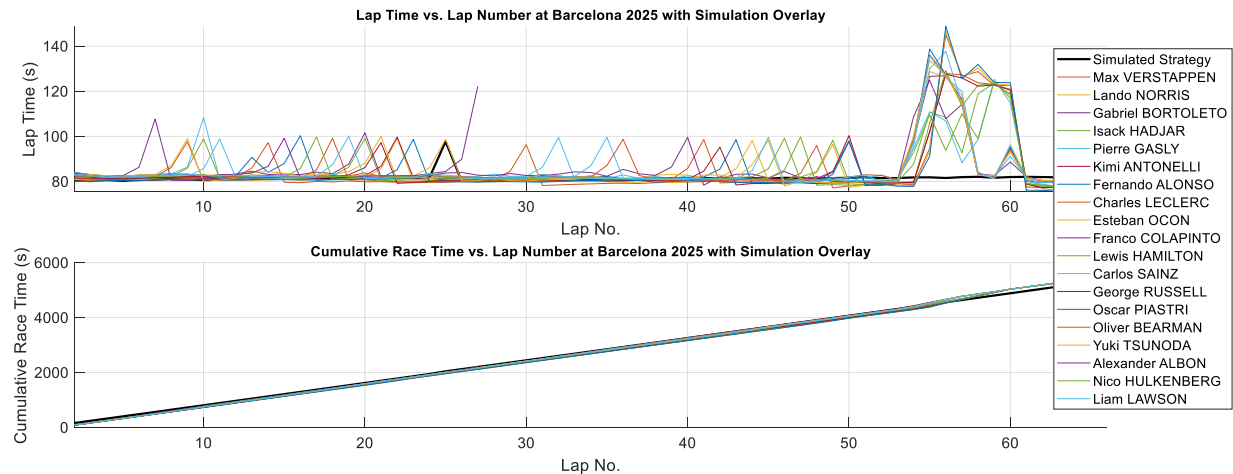


Figure 16 - Comparison of simulated strategy against all race participants for Formula 1 Barcelona 2025

4.2. Monte Carlo simulation implementation

The decision is made to test each randomised element in isolation initially to aid in validation, this allows finer tuning of the applicable parameters for every element. The underlying philosophy for this step is the law of large numbers: the more simulation runs one does, the more the distribution of the sampled results will approximate the actual underlying probability distribution (Heilmeyer, et al., 2020) (Kenton, 2024). What this means in practice is that by simulating thousands of races, probabilities for the success of a given strategy can be calculated as opposed to being approximated from a single deterministic projection.

4.2.1. Randomised pit stops

In the Monte Carlo system, pit entry is a random variable drawn from a given distribution rather than being a deterministic choice. This is fundamental to Monte Carlo simulation: approximating the probability distribution of race outcome by many repeated samplings of the uncertain parameters (Song, et al., 2025). Pit entry randomisation is underpinned by the law of large numbers and the central limit theorem, the basis for Monte Carlo methods. By running an incredibly large number of simulations with pit laps sampled from bounded or uniform distributions, the total of results converges to a steady estimate of outcome probabilities (Kong, et al., 2003) (Taboga, 2025).

Pit laps are decided within the simulation through a uniform probability distribution with lower and upper limits to reduce the number of unrealistic simulations and improve the feasibility of suggested strategies, this removes the possibility for the simulation to decide to pit on the final

lap, which would not be optimal, thus improving the efficiency of the simulation (Song, et al., 2025). The distribution is created using the MATLAB function `randi` and bound within the limits of `min_pit_lap` and `max_pit_lap`, see equation 6 and figure 17.

$$L_{pit} \sim U(L_{min}, L_{max}) \quad \text{Equation 6}$$

```

% Initial random pit plan
min_pit_lap = 8;
min_laps_remaining = 8;
max_pit_lap = laps - min_laps_remaining;
scheduled_pit_lap = randi([min_pit_lap, max_pit_lap]);

```

Figure 17 - Randomised pit plan coding methodology

The duration of these stops is dictated by a fixed length of time to account for loss in time due to pitting (reduced speed due to pit limiter restrictions), and then a sampled value from the distribution shown in figure 18 which decides the duration of time spent stopped to simulate the imperfections and small mistakes made within pit stops.

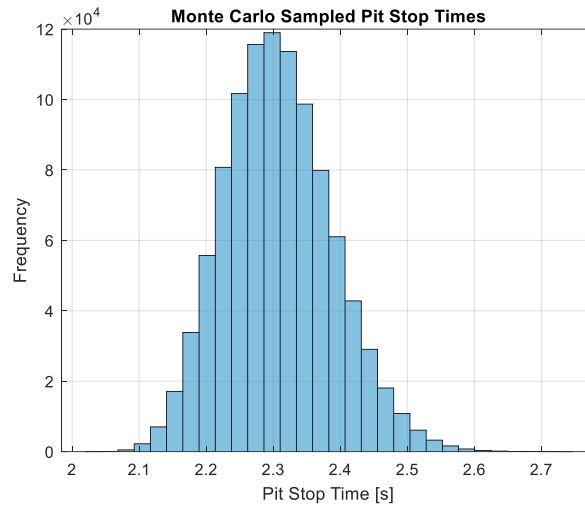


Figure 18 - Beta distribution of pit stop times for randomisation of pit stops

This initial simulation also incorporates a tyre model that will force a stop once the tyre health proxy dictates is necessary. This health proxy is calculated using the previous tyre degradation from section 4.1.5, and degradation is stored and deducted from the “tyre life”, see figure 19. A condition is made stating when tyre life drops below a designated threshold a forced pit entry will ensue. From a Monte Carlo perspective, this is an example of a rare-event probability, where low-probability outcomes are allowed to manifest in the sample space.

```

% Degradation
stint_lap = stint_lap + 1;
deg = compound_factor * a * (1 - exp(-b * stint_lap));
tyre_life = tyre_life - deg;
tyre_life_history(i) = tyre_life;

```

Figure 19 - Tyre health proxy code for emergency pit stop scenario

This method increases fidelity because in a large set of simulations, most will follow a typical scheduled pit strategy, and a few will deviate into unscheduled premature stops due to stochastic degradation beyond the failure boundary. This is consistent with Monte Carlo reliability analysis techniques, in which the model is perturbed hundreds of times to see if there are any rare limit-state failures (Song, et al., 2025). Throughout many simulations, the frequency with which such emergency stops are made tends towards an estimation of how likely they are, enabling quantitative assessment of their impact on race time distributions. By compounding uniform and stochastic distributions, the model represents not merely a mean-field method, but the dispersion of methods and outcomes, which is the primary advantage of Monte Carlo methods over determinist optimisation (Giannelos, et al., 2025) (Zhang, 2020).

4.2.1.1. *Randomised tyre compound*

Despite the previous comment on individual randomisation testing, tyre compound randomisation is tested alongside pit stop decisions, as tyre compound change is part of the pit stop process. Each compound is selected at both start of race and through the race using the randi function described earlier, hence there is a uniform distribution of tyre selection. The compounds differ from each other with a grip multiplier and compound factor which each have their effects on lap times and degradation of tyre health. The values for these multipliers are shown in figure 20.

```

function [compound_factor, grip_multiplier] = getCompoundFactor(compound)
    switch lower(compound)
        case 'soft'
            compound_factor = 1.3;
            grip_multiplier = 0.98;
        case 'medium'
            compound_factor = 1.0;
            grip_multiplier = 1.00;
        case 'hard'
            compound_factor = 0.8;
            grip_multiplier = 1.02;
    end
end
  
```

Figure 20 - MATLAB function to switch tyre compound multipliers between compounds

4.2.1.2. *Pit time loss distributions*

Seen in figure 21 is the poor representation of pit stop time from the distribution shown in figure 18, this graphical representation illustrates the distribution of results over 10000 simulation iterations with a fixed pit stop lap. In practise, you could expect through 10000 pit stop attempts there would be instances where the fastest performed stop would be faster than 1.90 seconds and would expect a slower time than 3.48 seconds. This assumption is reinforced through the data displayed by F1Pace in figure 22, which illustrates that the previously used distribution missed the upper portion of the slow pit stop times from the 2024

F1 season, meaning improvements to the distribution must be made to improve simulation accuracy.

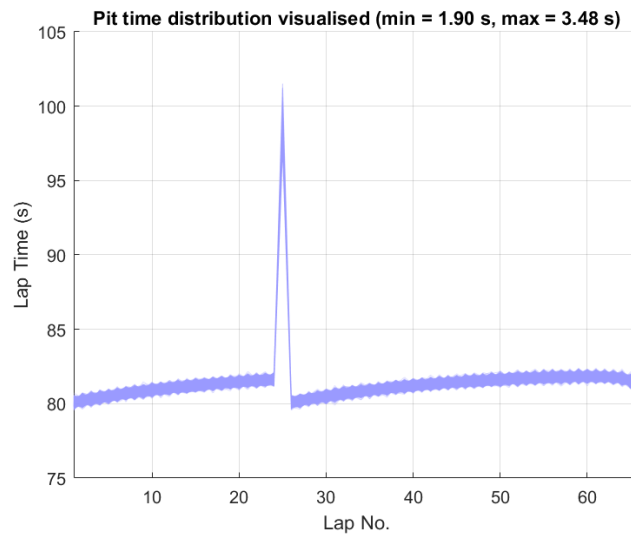


Figure 21 - Pit stop time distribution visualised through 10000 simulation iterations

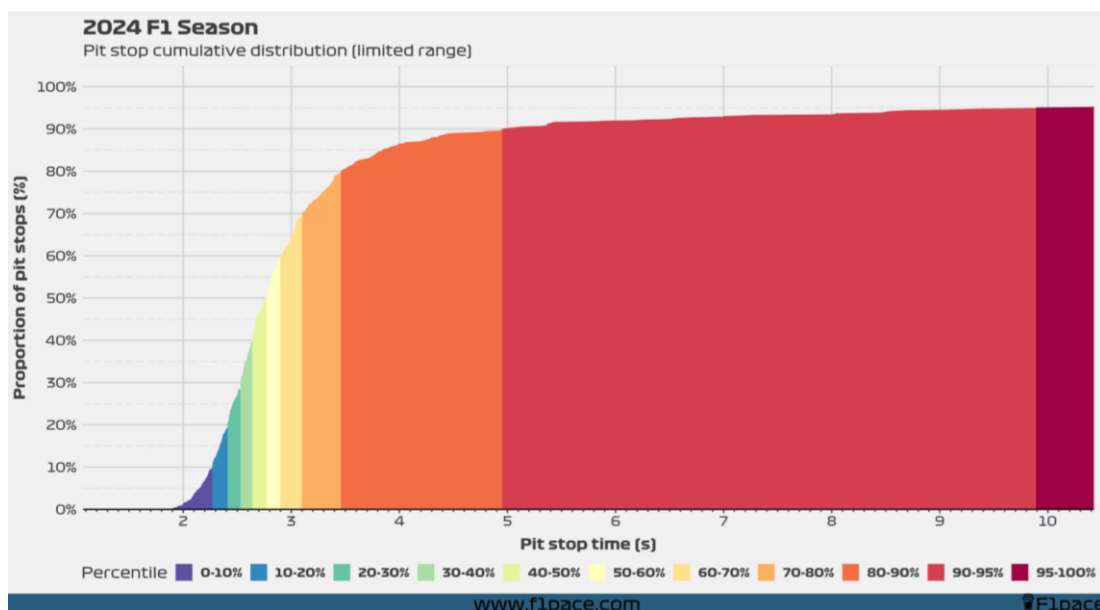


Figure 22 - Distribution of pit stop time durations from 2024 Formula 1 season (F1Pace, 2024)

Similarly to the previous distribution, this assumes a fixed period for pit time loss and a beta distribution for time loss during vehicle service. This is due to the need to make multiple distributions for multiple tracks using a distribution for the entire time loss, with this method a fixed value can be changed and still contain the distributed effect from stopped time whilst only changing a single value. The new distribution is created using 4 variables, mean, standard deviation, skew and kurtosis. With fine tuning of these 4 variables a distribution that better represents current pit stop timings is made, see figure 23, based on the data shown in figure 22 (F1Pace, 2024).

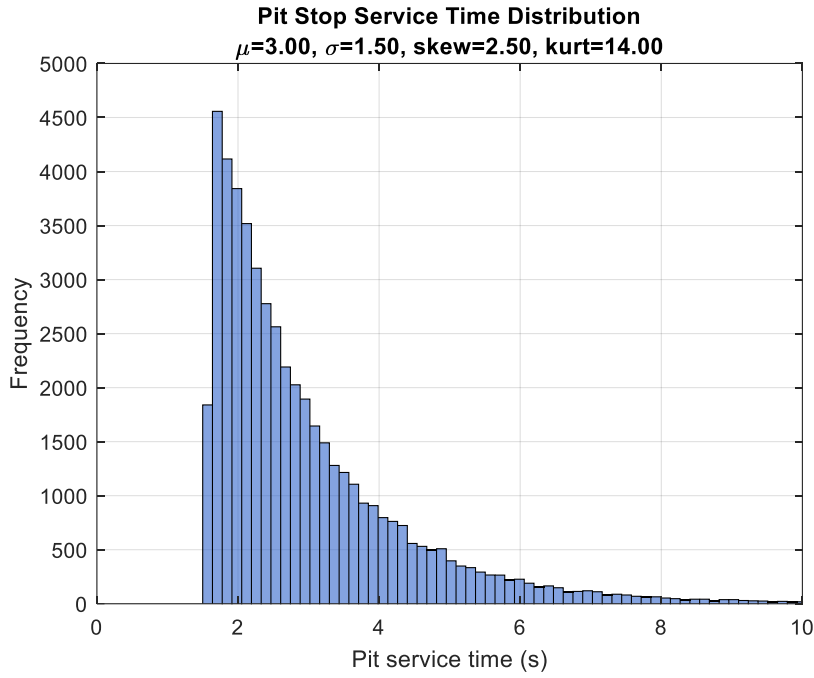


Figure 23 - Updated distribution for stopped time in pits using historic Formula 1 data

4.2.2. Traffic interaction modelling

To model traffic interactions, track segments need to be shortened so that it is possible to have known vehicle positions at smaller intervals, compared to once per lap. Initially the sectors inputs are based on the Barcelona GP for ease of verification, these inputs being sectors distances and an average velocity for the whole track.

The traffic interaction is evaluated using estimated positions using both distance and cumulative time, see equation 7 for the distance proxy and equation 8 for the cumulative time calculation. If a vehicle is within the interaction space, a lap time penalty is applied (for validation in initial case a set time is applied) on the faster vehicle, emulating a scenario in which a leading car reaches a backmarker and performs overtaking manoeuvre.

$$d_l^{(i)} = (l - 1) \times L_{lap} \quad \text{Equation 7}$$

$$T_{l-1}^{(i)} = \sum_{k=1}^{l-1} t_k^{(i)} \quad \text{Equation 8}$$

Figure 24 displays the vehicle interactions, during said interactions a fixed time penalty is applied initially for test purposes, this penalty would not be fixed under real racing conditions due to track layout, backmarker pace and numerous other variables. These variables can be accounted for through a distribution that encapsulates many possibilities for the penalty of overtaking and battling. Also, within the Monte Carlo framework, this will be a simple addition, which just requires replacement of the singular value, with the distribution.

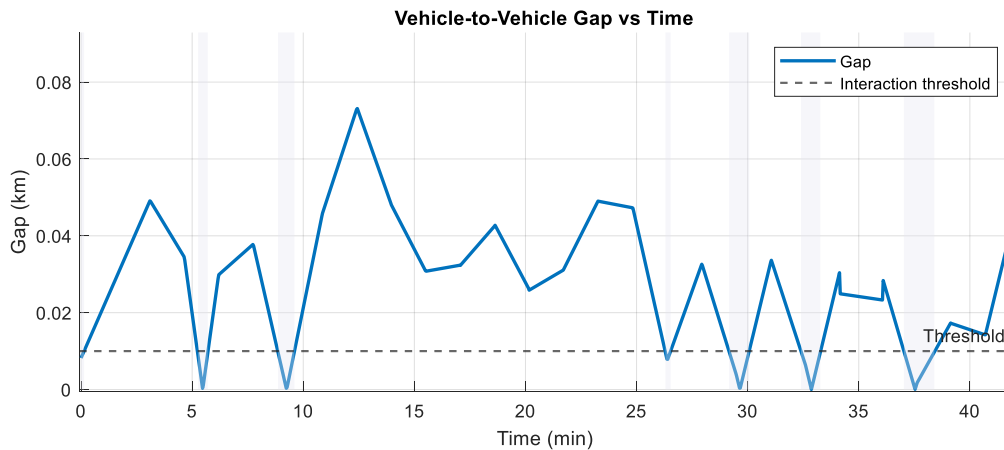


Figure 24 - Validation of vehicle interaction showing gap between two cars and interaction threshold

4.2.2.1. Traffic time penalty distribution

The initial interaction model employed a hard-coded time penalty every time a faster vehicle overtakes a slower one within the interaction distance range defined. While as good as it was for a proof of concept, it oversimplifies the randomness of endurance racing traffic as well. In practice, time spent during an overtaking manoeuvre is very variable and dependent on where along the track the meeting occurs, the relative disparity in speed between classes of car, and prevailing race conditions. Meetings on straights usually have an unnoticeable loss, whereas meetings of strings of medium- to high-speed corners can have substantial delays (Paparusso, et al., 2022) (Heilmeier, et al., 2020).

To better represent this variability, a probabilistic penalty distribution is introduced. This distribution is shown through figure 25 A. The majority of penalties fall within the range of 0.5–2.0 s, with a median of approximately 1.32 seconds and a mean of 1.48 seconds.

This skew distribution, where there were mostly small losses but with a long upper tail of larger penalties, is modeled with a beta distribution with limits of 0 s and 6 s. Parameters are set to simulate the observed mean, median, and upper-percentile behavior of the data. Drawing from this distribution at each interaction event within the Monte Carlo environment allows the simulation to capture the high-volume, low-impact interactions typical of racing and the low-volume, high-consequence perturbations that may more than compensate for their infrequency with their effect. Figure 25 B demonstrates how sampling from the distribution generates a spread of lap time outcomes at fixed interaction points. Most interactions result in small, near-negligible delays, while occasional larger penalties produce noticeable deviations, reflecting the sporadic but influential nature of endurance traffic events. The distribution can be altered to better suit the style of racing whether the interaction is with slower moving vehicles (multiclass racing), or whether it is a backmarker in a vehicle of similar performance level but at the back of the pack. This addition enables the model to represent the trade-off

between short-term lap gains and long-term tyre health, a central concern in both sprint and endurance racing strategy.

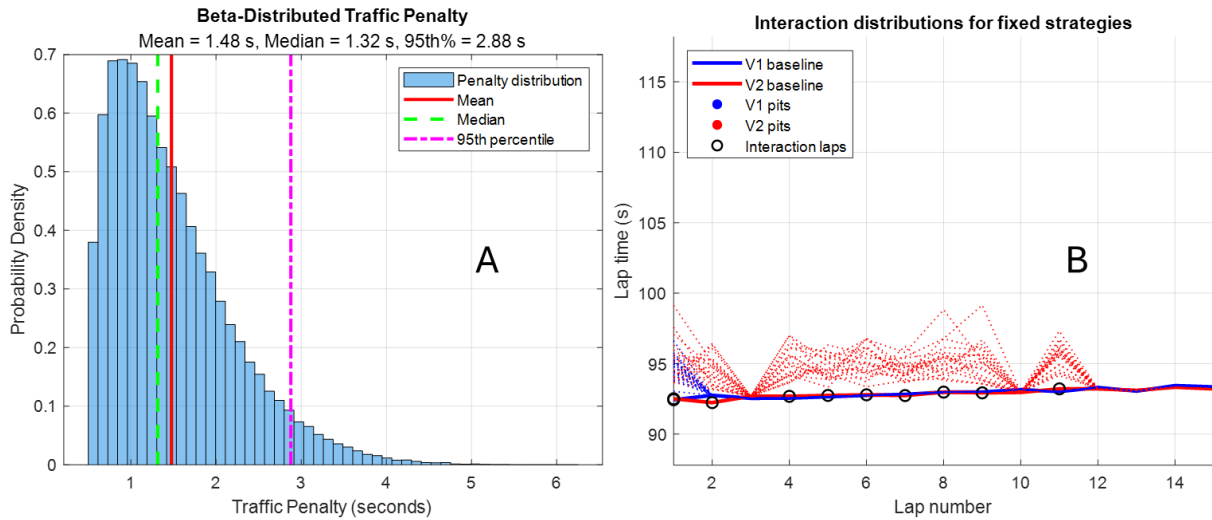


Figure 25 – (A) Beta distribution of traffic time penalties for faster vehicle (B) Impact of probabilistic traffic penalties on lap time progression

4.2.2.2. Multi-agent reinforcement learning and game-theoretic methods

Recent studies have explored competitor aware race simulations using MARL and game theory as discussed in 3.2.4, such methods have been shown to capture more realistic emergent behaviour in traffic interactions, particularly under overtaking scenarios and when drivers compete for track position (Zhou, et al., 2022) (Liu, et al., 2023).

While the promise of MARL and game-theoretic approaches is promising, their use in this project is precluded on three grounds. To begin with, these approaches require extremely large datasets, besides extensive training time to converge to stable policies. Given the available computational resources for this research, such computationally demanding training would not be feasible within the project scope. Second, MARL solutions introduce significant complexity in hyperparameter tuning, reward engineering, and convergence to realistic policies rather than unstable behaviours. This threatens to divert the project focus from optimisation of race strategy to AI methodology research. Third, the objective of this work is not the replication of competitor intelligence in its entirety but the provision of a strategy support tool that can simulate thousands of scenarios within realistic timescales. To this end, a stochastic penalty framework will suffice, as it allows the traffic delay distribution to be incorporated within the Monte Carlo runs without requiring agent-level modelling.

4.2.3. Cropping algorithm

To efficiently test many simulated race strategies, a progressive cropping algorithm is introduced. The purpose of this method is to thin the initially generated grid of competitors down to the subset of survivors by sequentially eliminating poorly performing strategies,

retaining those which remain competitive for the full race distance. This technique guarantees that computational resources are focused on promising candidates without impairing the statistical diversity required for reliable optimisation (Jamieson & Talkwalker, 2016) (Li, et al., 2018).

The algorithm starts by creating a complete population of race simulations, each car given independent stochastic variations in lap time, pit stop performance, tyre wear, and fuel considerations. For the baseline implementation described here, an initial grid of 50,000 virtual opponents is created.

To progressively reduce the field, the race is divided into a series of cropping rounds of size. After each block, the worst-performing actors are eliminated according to a cut plan spreading eliminations over the race length. This prevents excessive early eliminations while eliminating less competitive strategies in an orderly fashion. For example, after the first cut, the grid was reduced to 50,000 from 45,456 competitors, and each subsequent round diminished the population until there were only 20 survivors remaining at the end of the race.

Survivors at each level thus form the best-performing subset under the simulated stochastic conditions. At the end of the cropping process, the final set of strategies can then be analysed in detail.

4.2.3.1. Cropping algorithm performance testing

To validate the effectiveness of the cropping algorithm and ascertain whether it is a helpful and necessary addition, a test is performed which directly compares identical runs using the rng function through MATLAB, one with the cropping algorithm and one without. This will allow assessment of whether the cropping algorithm cuts any runs which may have otherwise been viable and part of the strategy and assess the performance of the algorithm regarding cutting the time it takes to solve these vast simulations, see Appendix A for list on analysis metrics.

Through analysis of figure 26 A, B & E, the performance of the algorithm can be evaluated. It is shown that runtime is seen to scale sharply with N (Initial grid size) for both approaches, but the cropped run always completes faster. Over the range of N tested, speedup is ~ 1.3 – $1.38\times$, or 24–27% wall-time reduction, with biggest savings at greater N. This makes sense: the larger the initial population, the more low-quality candidates are eliminated up front, so there are fewer laps simulated in total (Li, et al., 2018).

Figure 26 C & D show agreement between methods is high. Overlap at K is typically within the 0.95–1.00 range (only 0–1 vehicles vary in the top 20), and Spearman's Rho is within the 0.93–0.99 range, indicating high rank order preservation amongst typical survivors. The mean absolute variation of overall race time between comparable cars is ≈ 9 – 14 s, with minimal large excursions, as expected from the infrequent near-cutline cases where initial pruning nominally disturbs end-of-race totals. (Zar, 2005).

For all sizes of grids tested, the cropping algorithm offers significant computational savings (~25%) with excellent selection fidelity and very minimal timing error among regular survivors ($MAE \approx 9\text{--}14$ s). These results are consistent with theoretical and empirical results for successive-halving/early stopping methods, which show that budget-aware culling can retain the best candidates without computing on the less optimal ones. Thus proving worthwhile for large-N Monte Carlo strategy in this case, in agreement with (Jamieson & Talkwalker, 2016) and (Zar, 2005).

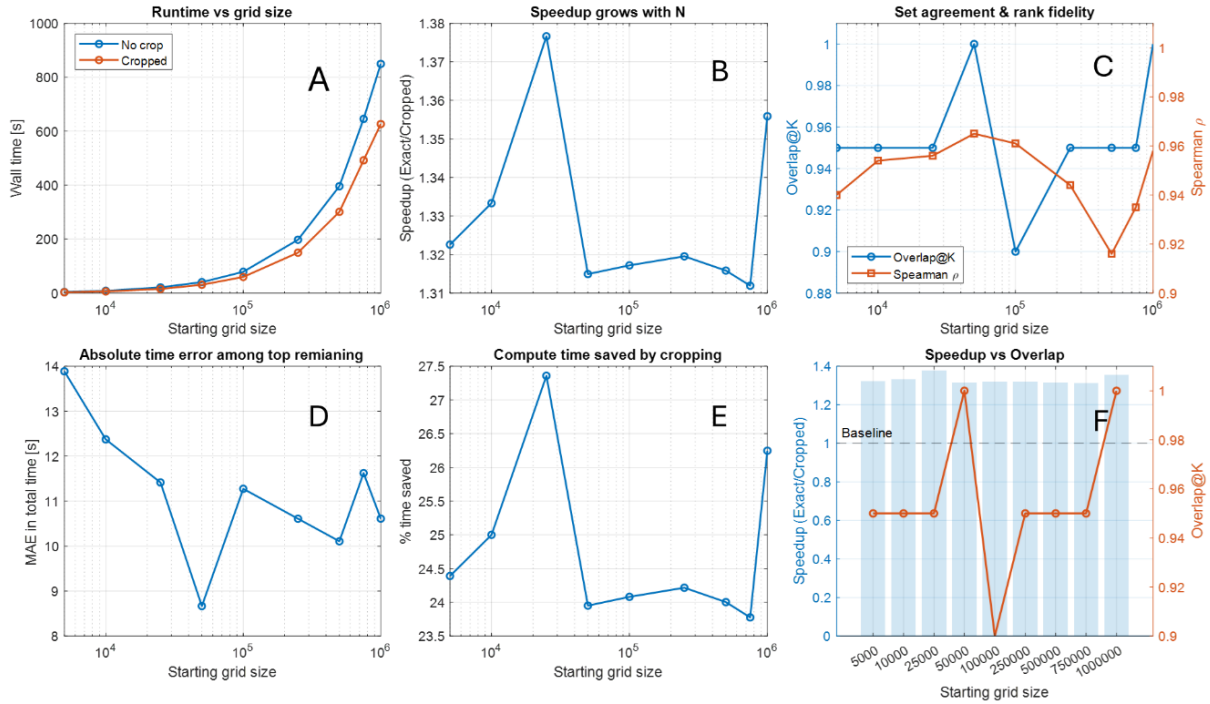


Figure 26 - Performance and fidelity evaluation of cropping algorithm method versus identical simulation without cropping algorithm (A) Wall time for Crop and no Crop simulations (B) Ratio of no Crop to Crop times (C) Overlap@K (left axis) and Spearman rank (D) MAE of total race time for remaining strategies (E) Percentage wall time reduction from cropping algorithm (F) speedup and Overlap@K versus starting grid size, showing the accuracy/efficiency trade-off

4.2.3.2. Seed robustness testing for cropping algorithm comparison

It is also valuable to test alternate base seed values to verify that the efficiency and fidelity, testing at base seeds 40–50 with everything modelling options fixed constant. Reporting across multiple random seeds is recommended in stochastic experiments to avoid misleading conclusions driven by a single seed (Henderson, et al., 2019), see table 3 for the results of this study. Performance was very stable across seeds: No crop 39.42 ± 0.75 s vs Cropping 30.14 ± 0.54 s, with an obtained speedup of 1.306 ± 0.009 and time saved of $23.54 \pm 0.54\%$. Fidelity was also strong, Overlap @ K was 0.955 ± 0.028 : 9/11 seeds ≥ 0.95 , two dropped to 0.90 and 0.85, and two reached 1.00. Spearman's ρ was 0.906 ± 0.043 , 3/11 seeds < 0.90 , the rest 0.91–0.97. MAE of overall race time among surviving denominators was 11.55 ± 1.30

seconds. The intermittent overlap dip conforms to near-cutline volatility, border candidates which look weak in an initial block but recover after a follow-up stop/compound. Two real-world mitigations are the tail of the early-stopping body: a buffered final round and a concave cut-schedule; both at the expense of false negatives only with a slight runtime penalty, as shown for successive halving/Hyperband methods which budget adapt and cull underperformers as stated by (Jamieson & Talkwalker, 2016) and (Li, et al., 2018).

Table 3 – Base seed sweep (40–50): Stability of runtime and top-K agreement

Base seed	Grid size N	Runtime (No cropping) (s)	Runtime (With cropping) (s)	Speedup (x)	Overlap @ K	spearman ρ	MAE of total time (s)
40	50000	41.2	31.5	1.31	0.95	0.919	10.462
41	50000	39.5	30.4	1.30	0.95	0.749	12.421
42	50000	39.1	29.8	1.31	1	0.965	8.667
43	50000	39.7	30.7	1.29	0.85	0.885	14.216
44	50000	41.1	31.3	1.31	0.95	0.94	13.019
45	50000	38.6	29.4	1.31	0.95	0.828	9.803
46	50000	39.4	30	1.31	0.95	0.926	10.444
47	50000	40.4	30.4	1.33	0.95	0.949	14.351
48	50000	37.9	29.6	1.28	1	0.943	9.325
49	50000	38.4	29	1.32	0.95	0.911	12.456
50	50000	38.3	29.4	1.30	0.9	0.946	11.868

4.2.3.3. *Memory storage improvements through cropping algorithm*

Another investigation into the cropping algorithm is performed, this time looking at memory storage of this method. In theory, the cropping algorithm should reduce the amount of stored memory in comparison to the simulation without the cropping method (Mathworks, 2025). However, with analysis of table 4, this is not the case, see Appendix A for list on analysis metrics.

Table 4 - Memory usage snapshots for cropping and no cropping simulation methods, showing peak, average, and largest variable sizes across different execution phases

Iterations	Phase	Workspace peak	Workspace average	Largest variable peak
		MB		
1000000	Cropping before	10550	10550	10500
	Cropping during loop	10400	10400	10350
	Cropping final	10550	10500	10500
	Without cropping before	31.51	31.51	7.63
	Without cropping during loop	31.48	31.48	7.63
	Without cropping final	31.52	31.51	7.63
750000	Cropping before	7910	7910	7880
	Cropping during loop	7800	7800	7770
	Cropping final	7910	7870	7880
	Without cropping before	23.63	23.63	5.72
	Without cropping during loop	23.61	23.61	5.72
	Without cropping final	23.65	23.64	5.72
500000	Cropping before	5270	5270	5250
	Cropping during loop	5200	5200	5180
	Cropping final	5270	5250	5250
	Without cropping before	15.76	15.76	3.81
	Without cropping during loop	15.74	15.74	3.81
	Without cropping final	15.77	15.77	3.81
250000	Cropping before	2640	2640	2630
	Cropping during loop	2600	2600	2590
	Cropping final	2640	2620	2630
	Without cropping before	7.88	7.88	1.91
	Without cropping during loop	7.88	7.88	1.91
	Without cropping final	7.9	7.89	1.91
100000	Cropping before	1050	1050	1050
	Cropping during loop	1040	1040	1040
	Cropping final	1050	1050	1050
	Without cropping before	3.16	3.16	0.78125
	Without cropping during loop	3.15	3.15	0.78125
	Without cropping final	3.17	3.17	0.78125
50000	Cropping before	540.07	540.07	537.7
	Cropping during loop	532.41	532.41	530.11
	Cropping final	540.1	537.35	537.7
	Without cropping before	1.58	1.58	0.39062
	Without cropping during loop	1.58	1.58	0.39062
	Without cropping final	1.6	1.6	0.39062
25000	Cropping before	270.05	270.05	268.86
	Cropping during loop	266.21	266.21	265.06
	Cropping final	270.07	268.68	268.86
	Without cropping before	0.81466	0.81466	0.19531
	Without cropping during loop	0.81369	0.81369	0.19531
	Without cropping final	0.82865	0.82833	0.19531
10000	Cropping before	108.01	108.01	107.52
	Cropping during loop	106.49	106.49	106.03
	Cropping final	108.01	107.46	107.52
	Without cropping before	0.33078	0.33078	0.07812
	Without cropping during loop	0.33029	0.33029	0.07812
	Without cropping final	0.34477	0.34469	0.07812
5000	Cropping before	53.99	53.99	53.74
	Cropping during loop	53.25	53.25	53.02
	Cropping final	53.99	53.72	53.74
	Without cropping before	0.16948	0.16948	0.03906
	Without cropping during loop	0.16916	0.16916	0.03906
	Without cropping final	0.18348	0.18348	0.03906

The cropping method, while effective at reducing computation time, is memory intensive in its current state. This is because all states of drivers are pre allocated during run initialisation and stored in memory until the end, even though the corresponding car may then be pruned from the simulation. As a result, memory usage remains dominated by the huge states array holding entire race histories and lap-level information for all cars, including even the cars eliminated in early rounds. The initial method, on the other hand, removes temporary variables following simulation for each car, and only compact summary arrays remain in memory. To render cropping less memory-hungry, the state container needs to be restructured: retaining states in a cell array rather than a struct array allows culled entries to be cleared directly (Mathworks, 2025). By contrast, cell arrays allow heterogeneous and variably sized elements, meaning deleted or cropped entries can be freed, which reduces storage overhead in theory. However, cell arrays also carry additional pointer overhead for each element, and MATLAB's memory manager does not always immediately reclaim fragmented space, especially when working with large pre-allocated arrays. While the utilisation of cell-based storage in the cropping method provides some memory improvement over storing large struct arrays for every car, testing show that it is still less efficient than the non-cropping simulation, see table 5. This is because cropping requires many driver states to coexist in memory at a time until an elimination step, whereas the original method processes car by car and deletes temporary variables on the go. Thus, while the pruned entries of the cell array can be released for reclaiming some memory space, the peak and mean memory spaces are still significantly larger during execution. As a result, the cropping algorithm still shows higher peak and mean usage compared to the non-cropped simulation, where temporary per-car variables are cleared as they are processed. This trade-off is typical of MATLAB's memory model: struct arrays maximise access speed and compactness when the data shape is fixed, whereas cell arrays provide flexibility at the cost of greater memory fragmentation and bookkeeping (Mathworks, 2025). The main advantage therefore arises from reducing memory after vehicles have been eliminated and the major simulation time reductions discussed previously.

Table 5 - Comparison of memory usage between struct array and cell array implementations across different iteration counts, showing peak, average, and largest variable sizes before, during, and after execution

Iterations	Phase	Workspace peak	Workspace average	Largest variable peak
		MB		
1000000	Struct array before	10550	10550	10500
	Struct array during loop	10400	10400	10350
	Struct array final	10550	10500	10500
	Cell array before	145.43	145.43	99.44
	Cell array during loop	12400	12400	12360
	Cell array final	11290	57500	11240
750000	Struct array before	7910	7910	7880
	Struct array during loop	7800	7800	7770
	Struct array final	7910	7870	7880
	Cell array before	109.14	109.14	74.64
	Cell array during loop	9300	9300	9270
	Cell array final	8470	4310	8430
500000	Struct array before	5270	5270	5250
	Struct array during loop	5200	5200	5180
	Struct array final	5270	5250	5250
	Cell array before	72.85	72.85	49.85
	Cell array during loop	6200	6200	6180
	Cell array final	5650	2870	5620
250000	Struct array before	2640	2640	2630
	Struct array during loop	2600	2600	2590
	Struct array final	2640	2620	2630
	Cell array before	36.56	36.56	25.05
	Cell array during loop	3100	3100	3090
	Cell array final	2820	1440	2810
100000	Struct array before	1050	1050	1050
	Struct array during loop	1040	1040	1040
	Struct array final	1050	1050	1050
	Cell array before	14.78	14.78	10.17
	Cell array during loop	1240	1240	1240
	Cell array final	1130	588.88	1120
50000	Struct array before	540.07	540.07	537.7
	Struct array during loop	532.41	532.41	530.11
	Struct array final	540.1	537.35	537.7
	Cell array before	7.52	7.52	5.21
	Cell array during loop	634.92	634.92	632.72
	Cell array final	578.32	294.51	575.67
25000	Struct array before	270.05	270.05	268.86
	Struct array during loop	266.21	266.21	265.06
	Struct array final	270.07	268.68	268.86
	Cell array before	3.9	3.9	2.74
	Cell array during loop	317.47	317.47	316.36
	Cell array final	289.18	147.35	287.85
10000	Struct array before	108.01	108.01	107.52
	Struct array during loop	106.49	106.49	106.03
	Struct array final	108.01	107.46	107.52
	Cell array before	1.72	1.72	1.25
	Cell array during loop	127	127	126.55
	Cell array final	115.69	59.03	115.15
5000	Struct array before	53.99	53.99	53.74
	Struct array during loop	53.25	53.25	53.02
	Struct array final	53.99	53.72	53.74
	Cell array before	1.02	1.02	0.76946
	Cell array during loop	63.5	63.5	63.28
	Cell array final	57.86	29.59	57.59

4.2.4. Driver thermal management modelling

The modelling of thermal management regarding tyre performance vs degradation is done through a simple calculation of cumulative wear state on a compound reference multiplied by a push input (West & Limebeer, 2020). This push input is randomised per run and constant per N , where increased push factors reduce lap times through equation 9, where the push factor effect on lap time is calculated through equation 10 using g_{max} as the maximum gain, s_c as a compound scale to differentiate between soft, medium and hard and β_g representing the concavity of gains. The push factor then amplifies the tyre degradation shown through equation 11. This wear uses k_c as the compound specific wear rate and γ and β_w to control the strength and curvature of the thermal amplification, where η_t is a bounded jitter around 1 to avoid extremes. Simply, increasing p_t lowers the current lap time through equation 10 and simultaneously increases wear through equation 11.

$$L_t = B_t(1 + w_t) - G_t^{push} + \varepsilon_t \quad \text{Equation 9}$$

$$G_t^{push} = g_{max} s_c p_t^{\beta_g} \quad \text{Equation 10}$$

$$\Delta w_t = k_c (1 + \gamma p_t^{\beta_w}) \eta_t, w_t \leftarrow w_{t-1} + \Delta w_t \quad \text{Equation 11}$$

Initial parameters set g_{max} at 2 seconds, $\beta_g = 0.8$, $\gamma = 1.1$ and $\beta_w = 1.1$. Push factor is swept through 0 to 1 and the results returned are shown through figure 27, to ensure comparability the seed is fixed between all simulations here. Indicated here is a higher push reduces lap time immediately but accelerates degradation, the chosen exponents mean small to moderate push values are quite rewarding with wear cost is near proportional to push ($\beta_w = 1.1$).

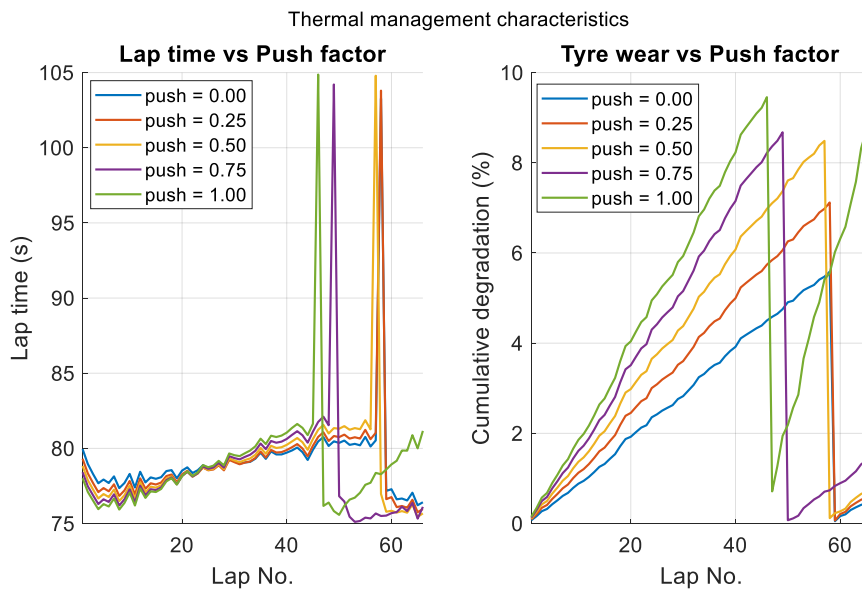


Figure 27 - Thermal management trade-off between push for faster lap times vs increased tyre degradation

Shown in figure 28 are the effects on push, yielding an optimum push factor for each compound. To tune the push model parameters for each compound, an optimal push factor (the minimiser of average lap time over a stint) can be predefined and solved for through differentiation of equation 9, which provides leaves equation 14. When solving for g_{max} to get the maximum time benefit for each compound, you are left with equation 15.

$$\bar{L}_c(p, N) = B_c \left(1 + \frac{N+1}{2} k_c (1 + \gamma p^{\beta_w}) \right) - g_{max,c} p^{\beta_g} \quad \text{Equation 12}$$

$$\frac{d\bar{L}_c}{dp} = 0 \Rightarrow p^{\beta_w - \beta_g} = \frac{g_{max,c} \beta_g}{B_c k_c \gamma \beta_w \frac{N+1}{2}}, \quad \text{Equation 13}$$

$$p = \left[\frac{g_{max,c} \beta_g}{B_c k_c \gamma \beta_w \frac{N+1}{2}} \right]^{\frac{1}{\beta_w - \beta_g}} \quad \text{Equation 14}$$

$$g_{max,c} = \frac{B_c k_c \gamma \beta_w \frac{N+1}{2} p^{\beta_w - \beta_g}}{s_c \beta_g} \quad \text{Equation 15}$$

The optimal push factor is decided as 0.5 for all compounds, yielding an exact ratio between tyre maintenance and performance pushing. Once solved the results are as shown through figure 29 yielding $g_{max,s} = 2.132$, $g_{max,m} = 1.001$ and $g_{max,h} = 0.721$.

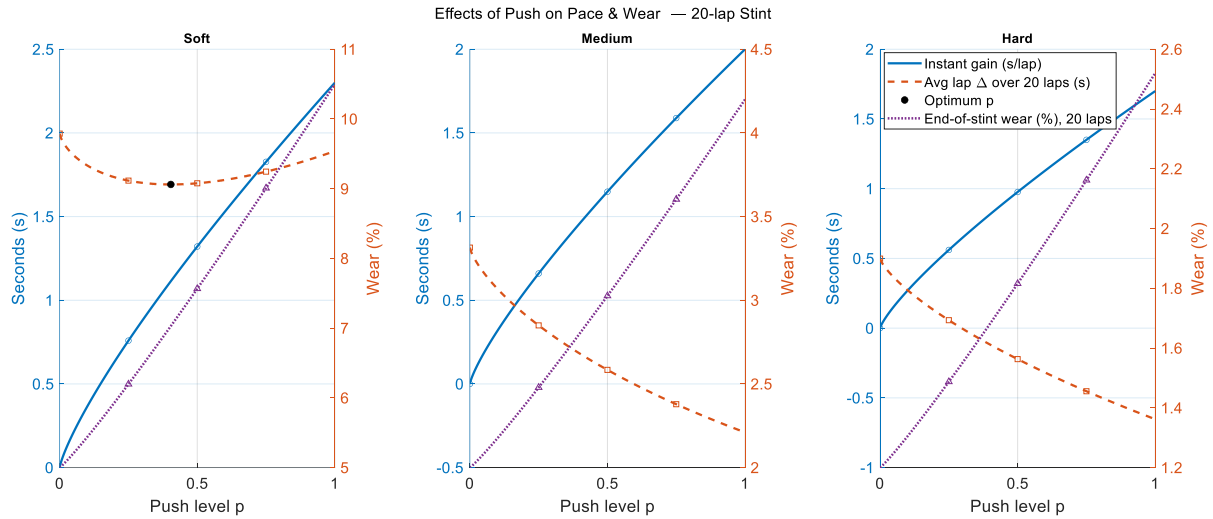


Figure 28 - Effects of push on pace and tyre wear over a 20-lap constant-push stint (Soft / Medium / Hard) using initial specified parameters

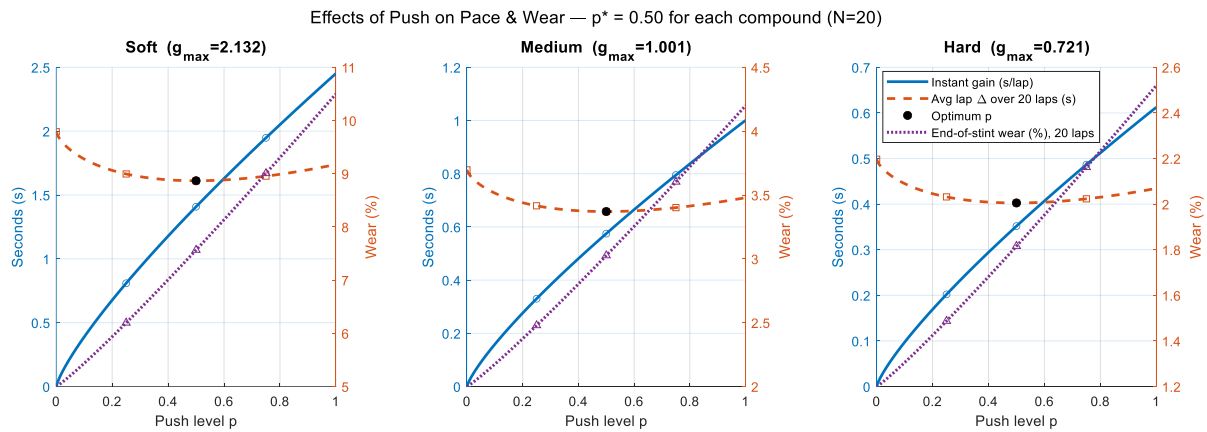


Figure 29 - Calibrating push through g_{max} over 20 lap constant push for all compounds

4.3. Simulation evaluation and validation

Upon validation of this solver, issues arose with the accuracy of the simulation. The main issue being the lack of convergence on a single strategy as illustrated by table 6, this is due to the poor performance of the tyre degradation model as illustrated through figure 30. Despite this poor representation of tyre degradation, this would not cause a lack of convergence as all simulation iterations will have the same boundary conditions regarding the degradation. Meaning that another factor is at play, which is decided as being a lack of conditional statements within the logic section of the pit decision window. Even when increasing the simulation size, a suitable level of convergence is not found, despite the large increases in solve times, when increasing the iteration count from 50000 simulations to 500000 simulations, the solve time increased by 391.30% from 3.46 seconds to 13.54 seconds with no real improvements to convergence.

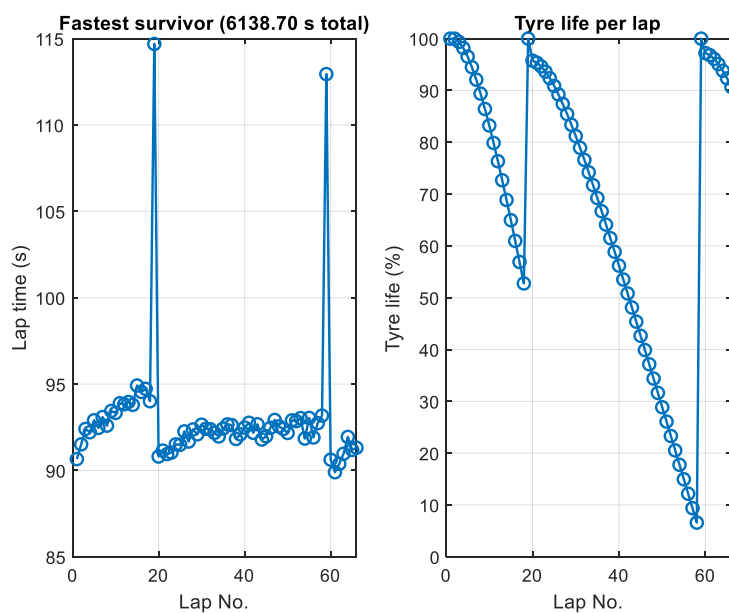


Figure 30 - Simulation output example for fastest survivor through the cropping algorithm (50000 iterations)

Table 6 - Example results of remaining 20 fastest vehicles from cropping simulation showing poor convergence on strategy suggestion (50000 iterations)

Total time (s)	No. of stops	Best lap time (s)	Average lap time (s)	Stop lap 1	Stop lap 2	Start compound	Compound at stops
6139.00	1	90.24	93.02	29.00		soft	hard
6140.80	2	90.02	93.04	19.00	59.00	soft	hard,hard
6142.46	1	90.68	93.07	29.00		soft	hard
6143.37	1	89.51	93.08	27.00		soft	hard
6143.80	1	90.18	93.09	27.00		soft	hard
6143.80	1	90.67	93.09	29.00		soft	hard
6143.94	1	90.09	93.09	29.00		soft	hard
6144.07	1	90.52	93.09	27.00		soft	hard
6144.16	2	90.50	93.09	20.00	60.00	soft	hard,hard
6144.24	1	89.72	93.09	29.00		soft	hard
6144.43	1	89.67	93.10	29.00		soft	hard
6144.45	1	90.38	93.10	27.00		soft	hard
6144.53	1	90.20	93.10	29.00		soft	hard
6144.72	1	89.76	93.10	27.00		soft	hard
6144.73	1	90.05	93.10	29.00		soft	hard
6144.80	2	90.12	93.10	19.00	59.00	soft	hard,hard
6144.83	2	90.23	93.10	19.00	59.00	soft	hard,medium
6144.88	2	89.50	93.10	20.00	60.00	soft	hard,hard
6144.92	1	90.39	93.10	29.00		soft	hard
6144.95	2	89.62	93.11	19.00	59.00	soft	hard,hard

The previous model used the tyre degradation model explained through equation 2, this degradation factor then being multiplied by the compound factors shown in figure 20. As this model gives a basic curve for a singular compound and then multiplied, it cannot simulate how different compounds act throughout their lives and warm up periods. Due to this, a new function is created that introduces several parameters to introduce differing curves for different compounds with a similar switch case function as previously shown in Figure 20 to swap between compounds, see equation 16 for this new degradation model.

$$\Delta(i) = \underbrace{\beta_0}_{\text{compound base}} + \underbrace{We^{-\lambda(i-1)}}_{\text{warm-up}} + \underbrace{\gamma i}_{\text{plateau drift}} + \underbrace{A[\sigma(\kappa(i - L_c)) - \sigma(\kappa(1 - L_c))]}_{\text{late cliff}} \quad \text{Equation 16}$$

The compound base describes an initial offset to differentiate between compounds initially. The warmup phase smoothly decays to 0, picking up tyres approaching a steady operating condition. Exponentials converge is usual for tyre thermal/warm-up dynamics; tread temperatures rise/fall and decay exponentially on straights, which aids this choice (West & Limebeer, 2020). The plateau of drift gives the near-linear, progressive performance degradation once the tyre is within its operating window; linear (or low-order polynomial) degradation of lap-time against tyre age is also typical in race-strategy simulators, where the stint is modelled as warm-up to short peak to near-constant drop-off (Piccinotti, 2020). The cliff in terminal stints uses a centred logistic rise scaled by A . Subtracting $\sigma(\kappa(1 - L_c))$ zero-anchors the curve at lap 1 so the cliff contributes ~ 0 early in the stint and then ramps sharply near L_c , meaning a shift in the logistic so that it starts at 0 at the beginning of the stint, instead

of already having some penalty. Logistic S-Curves are the standard way to model rapid transitions around a threshold (Wachenheim, et al., 2003). The function used for this within the simulation is shown in figure 31.

```

1 function delta = tyre_delta_curve(laps, P)
2 % delta(i) = compound_base + warmup(i) + plateau(i) + cliff(i)
3 % Positive delta = added time (slower). Negative = time gain (faster).
4
5 i = 1:laps;
6
7 % Base compound offset (constant over stint)
8 base = P.base_offset;
9
10 % Warm-up term: can be positive (slow start) or negative (quick start)
11 % Decays smoothly to 0.
12 warmup = P.warmup_target * exp(-P.warmup_rate * (i-1));
13
14 % Plateau: slow, near-linear drift
15 plateau = P.plateau_rate * i;
16
17 % Late fall-off: logistic rise centred at cliff_lap
18 L0 = 1/(1 + exp(-P.cliff_steep*(1 - P.cliff_lap))); % value at lap 1
19 cliff_raw = 1./(1 + exp(-P.cliff_steep*(i - P.cliff_lap))) - L0;
20 cliff = P.cliff_amp * cliff_raw;
21
22 delta = base + warmup + plateau + cliff; % total lap-time delta vs av_lap
23 end
  
```

Figure 31 – MATLAB function for calculating tyre degradation per lap from equation 16

By successive testing and validation, the Soft, Medium, and Hard compound degradation-curve parameters are tuned. The Soft tyre's base offset, and steep exponential warmup reflect its rapid peak grip; the Medium lies between Soft and Hard in terms of early pace and decay rate; while the Hard compound's higher base offset, slow warmup, low plateau drift, and absence of sharp cliff reflect its steady, long-run performance over a stint. Each of the values is chosen to represent the expected compound behaviour, validation against actual F1 tyre-degradation curves within simulated, controlled conditions assures accuracy to actual tyre dynamics (Catapult, 2024). See figure 32 and figure 33 for degradation curves for all compounds and table 7 for the coefficients used with equation 16.

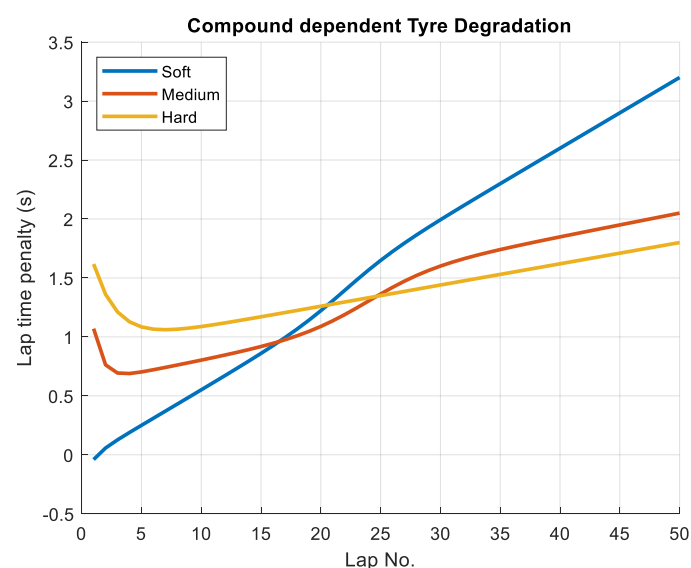


Figure 32 - Tyre degradation per lap using equation 16 for soft, medium and hard compounds

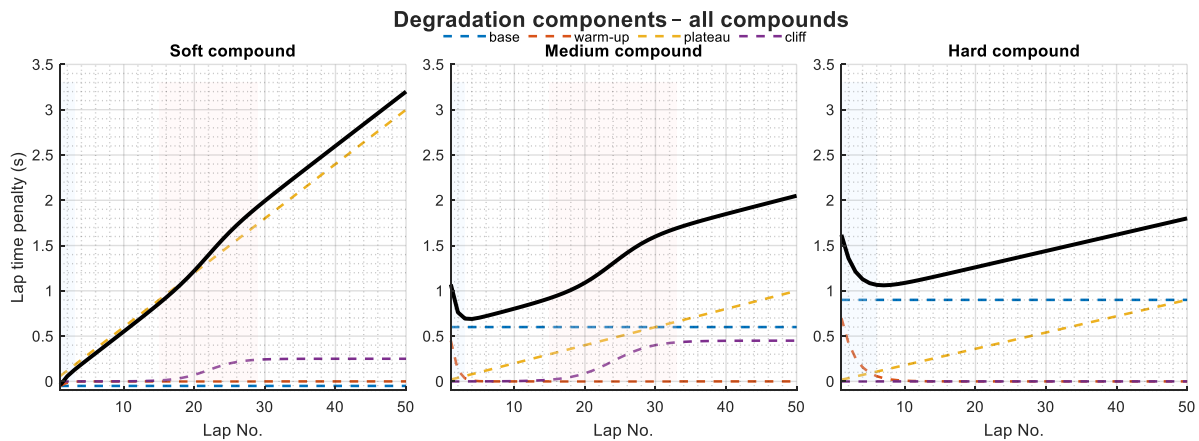


Figure 33 - Degradation coefficients illustrated from equation 16 for soft, medium and hard compounds

Table 7 - Tyre degradation coefficients for use in equation 16 for soft, medium and hard compounds

Tyre Compound	Soft	Medium	Hard
Base offset	-0.050	0.600	0.900
Warmup target	-0.050	0.450	0.700
Warmup rate	1.400	1.300	0.500
Plateau rate	0.060	0.020	0.018
Cliff amplitude	0.250	0.450	0.000
Cliff steep	0.450	0.350	0.250
Cliff lap	22.000	24.000	60.000

To aid into the issue discussed earlier with simulation bounds, restrictions covering the pit entry conditions are introduced. These will aid in simulation efficiency as many of the iterations that are wasted due to poor pit strategy such as trying to pit over X amount of pit within a fraction of remaining laps where no matter ‘tyre health’ pitting would be less effective than staying out. These iterations are filtered out allowing the simulation to focus on effective strategies that are logical to the aim of the simulation. These being:

- No pitting within the first X laps
- No pitting within the remaining X laps
- Maximum of X pit stops
- Minimum of two compounds used

With these additions, simulation accuracy will be improved yet quantifiable metrics for this are not present, despite this, figure 34 shows an example run between an unrestricted simulation and a restricted one. This clearly shows the poor logic that is cut out with these additions along with closer accuracy to real world regulations with the addition of the minimum compound selection logic.

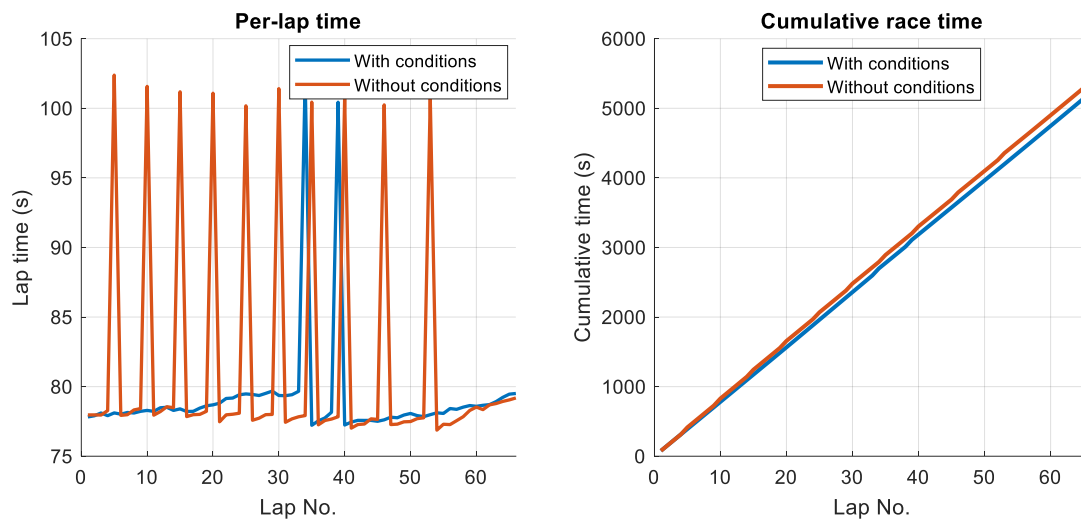


Figure 34 - Example output of unrestrained simulation to restrained simulation with logical pit entry conditional statements added

Due to the lack of historic data to properly validate tyre degradation, alternate methods of calculating degradation are tested. The current method aims to recreate the suggested degradation curves presented by Catapult (see figure 35); two problems are present with this. Firstly, the degradation is shown as a time penalty against lap number, meaning that the curves are track dependant due to different course lengths and effects on tyres. Secondly, this data doesn't come with any validated data, meaning the accuracy of the data cannot be completely trusted. For this reason, alternate methods of tyre degradation are investigated. The method chosen requires a large change to the logic used within the code, along with the removal of the tyre health proxy created earlier. Instead, the pit decision will be decided by a loss over a set base lap time. For example, when an average 3% of time is lost over the last 3 laps, the decision to pit is made. This eliminates the chance of pitting where a singular bad lap is performed and only allows when continued poor performance is seen, like methods employed by Heilmeyer (Heilmeyer, 2022). The degradation use for this goes back to a simpler percentage-based model due to the ease of validation, with one percentage multiplier for performance boosting and one that will dictate the degradation percentage per lap, a method that is supported by the work of Boettinger and Klotz (Boettinger & Klotz, 2023). Alongside this, a push factor is introduced that allows the simulation to investigate the effect of pushing a tyre harder for faster lap times with at a detriment to tyre degradation. The lack of convergence does not reflect model failure but the reality of multiple equally viable strategies, reinforcing the need for probabilistic decision support.

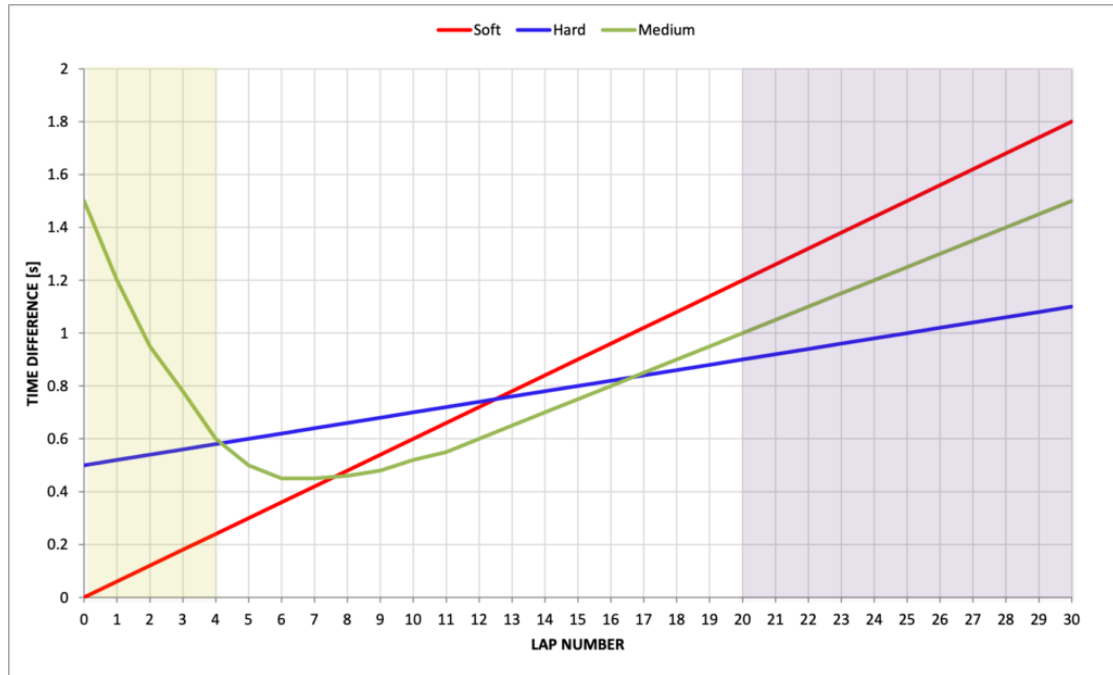


Figure 35 - Tyre degradation time penalty per lap for soft, medium and hard compounds (Catapult, 2024)

5. Results and analysis

5.1. Convergence testing

To test the accuracy of the simulation, it is important to first perform a convergence test for the number of simulation iterations. This avoids ‘over simulating’, where suitable strategy proposal is found in fewer simulations (Kong, et al., 2003) (Press, et al., 2007) (Taboga, 2025) (Song, et al., 2025). A simple test can be performed investigating convergence parameters around the suggested strategies from the remaining 20 competitors from simulation through a sweep of iterations. Initially, the test is performed using the Marina bay track physics (sector lengths and average race pace) for validation against the 2024 grand prix which saw no safety car introduction meaning a full race distance can be analysed and compared to the suggested strategy, returning the results in table 8 where higher dominant share and Jaccard indicate stronger convergence toward a single strategy, see Appendix A for list on analysis metrics.

Table 8 - Convergence of strategy recommendations versus iteration count (N) performed using Singapore GP track physics

N	Runs at N	Dominant strategy share		Shannon entropy		Gini-Simpson diversity index		Number of unique mean	Mean Jaccard
		Mean	Std	Mean	Std	Mean	Std		
10000	3	0.533	0.104	1.165	0.195	0.620	0.088	4.333	0.667
20000	3	0.367	0.076	1.385	0.073	0.717	0.028	5.000	0.767
50000	3	0.533	0.144	1.097	0.253	0.600	0.121	4.000	0.717
100000	3	0.467	0.058	1.073	0.083	0.622	0.035	3.667	0.833
200000	3	0.483	0.076	1.048	0.049	0.608	0.035	3.667	0.833
300000	3	0.500	0.087	1.053	0.099	0.598	0.049	4.000	1.000
400000	3	0.433	0.058	1.160	0.131	0.642	0.049	4.333	0.867
500000	3	0.417	0.029	1.251	0.092	0.677	0.024	4.333	0.867

Scrutinising table 8 shows across iteration counts, the metrics indicate no collapse onto a single strategy despite the large iterations counts. The dominant share stays ~0.42–0.53 with small standard deviations, entropy is ~1.05–1.25 and Gini-Simpson ~0.60–0.68, and the mean number of unique strategies remains ~3.7–5, consistent with several near-optimal sequences coexisting. Yet agreement between seeds is high (Mean Jaccard closer to 1), meaning the same set of strategies are found between seeds. This indicates that there is no under-sampling, and the search has discovered equal performing strategies.

To investigate this further, the convergence testing is performed for other track layouts (Catalunya, Silverstone, Sakhir and Circuit of the Americas) to analyse whether this trend is specific to this track distance and pace.

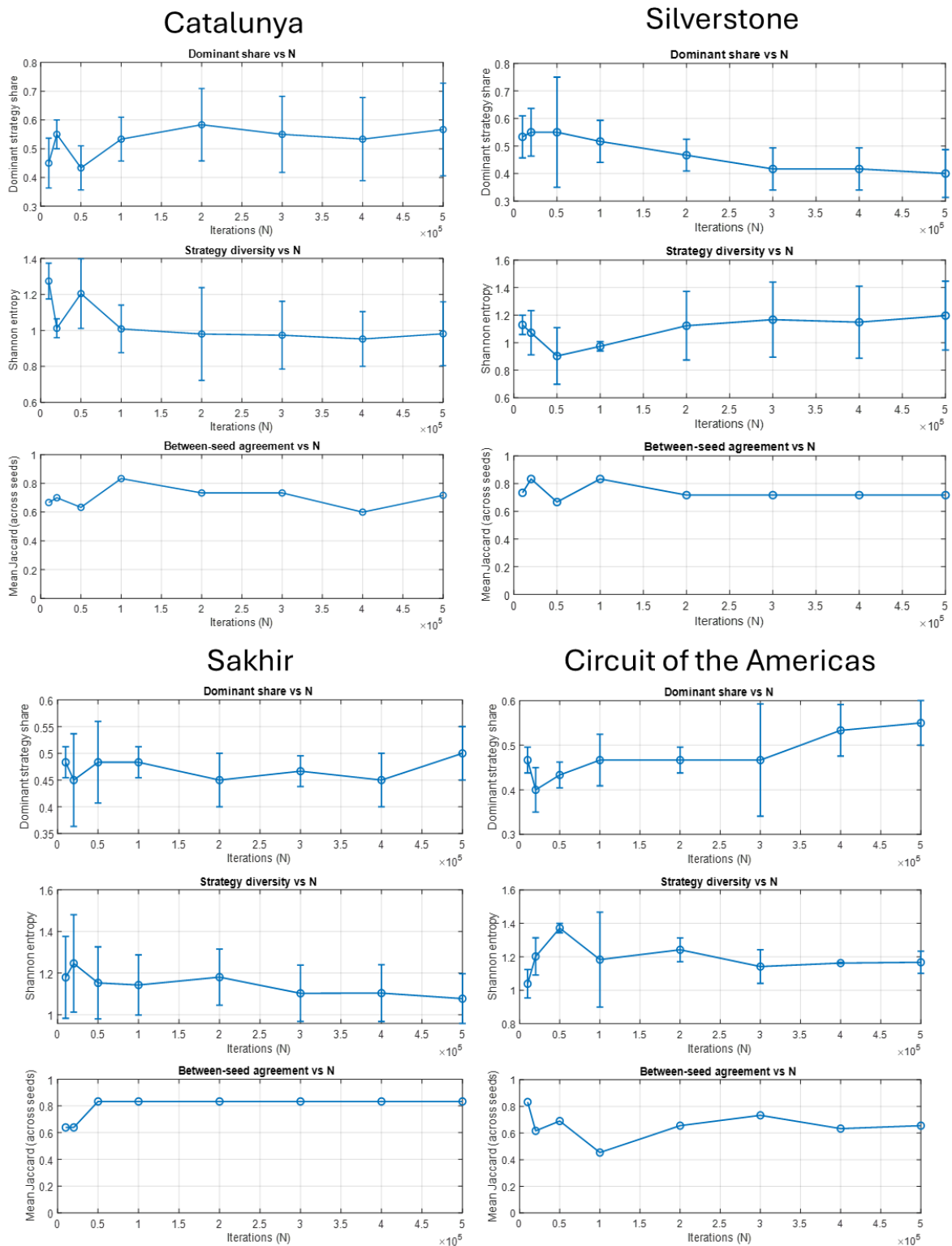


Figure 36 - Convergence of strategy recommendations versus iteration count for Catalunya, Silverstone, Sakhir and Circuit of the Americas

The additional results from further convergence studying are shown in figure 36, this shows very similar trends compared to table 8 with regards to between seed agreement and dominant strategy share, as Henderson suggests (Henderson, et al., 2019). Thus, leading to one of two

conclusions for the simulation, either iteration count is not high enough at 500,000 or there is a lack of validation for the simulation where it is not performing as it should, given the large amount of input parameters. To test this theory, a single simulation is run with 2,000,000 simulation iterations. This will therefore prove whether the simulation is failing to converge on a single result, or whether there are multiple viable strategies to yield optimal race times.

Table 9 -Strategy summary of the top-20 survivors after progressive cropping for 2,000,000 iterations

Total time	No. of stops	Stop laps	Start compound	Compound at stops
6007.9	2	[31 37]	hard	[soft hard]
6008.4	2	[31 38]	hard	[soft hard]
6009.4	1	[31]	medium	[hard]
6010.9	2	[31 38]	hard	[soft hard]
6011.1	1	[27]	medium	[hard]
6011.1	2	[31 37]	hard	[soft hard]
6011.3	1	[27]	medium	[hard]
6012.1	2	[31 40]	hard	[soft medium]
6012.3	1	[29]	medium	[hard]
601.2	1	[29]	medium	[hard]
6013.4	1	[27]	medium	[hard]
6013.6	2	[31 39]	hard	[soft hard]
6013.7	2	[31 38]	hard	[soft hard]
6013.8	1	[25]	medium	[hard]
6014.5	2	[31 37]	hard	[soft hard]
6014.7	1	[27]	medium	[hard]
6015	2	[31 39]	hard	[soft hard]
6015.1	2	[31 38]	hard	[soft hard]
6015.1	2	[31 39]	hard	[soft hard]
6015.7	1	[30]	medium	[hard]

With the results of table 9 leaving no improvement on strategy convergence, a suitable value for iteration count for subsequent parameter validation must be selected (Press, et al., 2007) (Kong, et al., 2003). To provide a representative share of results along with stability, a value of 200,000 is selected for following analysis. 200,000 provides a stable enough sampling resolution for parameter testing without excessive runtime.

5.2. Simulation parameter validations

To gain a suitable level of realism within results, parameter validation through comparisons to real race times is performed. As prior, initial analysis is done through comparisons with the 2024 Singapore GP due to the lack of safety car introduction (FIA, 2024). This initial simulation can be seen through figure 37. Upon initial analysis, it can be noted that lap time fluctuations are larger in areas for real drivers in comparison to the simulation, along with an oversight within model creation. This being the beginning of the race where drivers start from a standstill,

meaning initial lap times are substantially higher from the real data in this section. From a strategy suggestion point this should not affect outcomes for the simulation, yet it is important to note. This would not influence degradation at the beginning of the race due to the calculation of tyre degradation which the changes made discussed in 4.3. The validity of the suggested tyre strategy is reinforced through the decisions made by the eventual winner of the 2024 Singapore GP winner, Lando Norris, who ran the same strategy as suggested by the simulation (Medium to Hard), despite alternate pit laps (Pitpass.com, 2024).

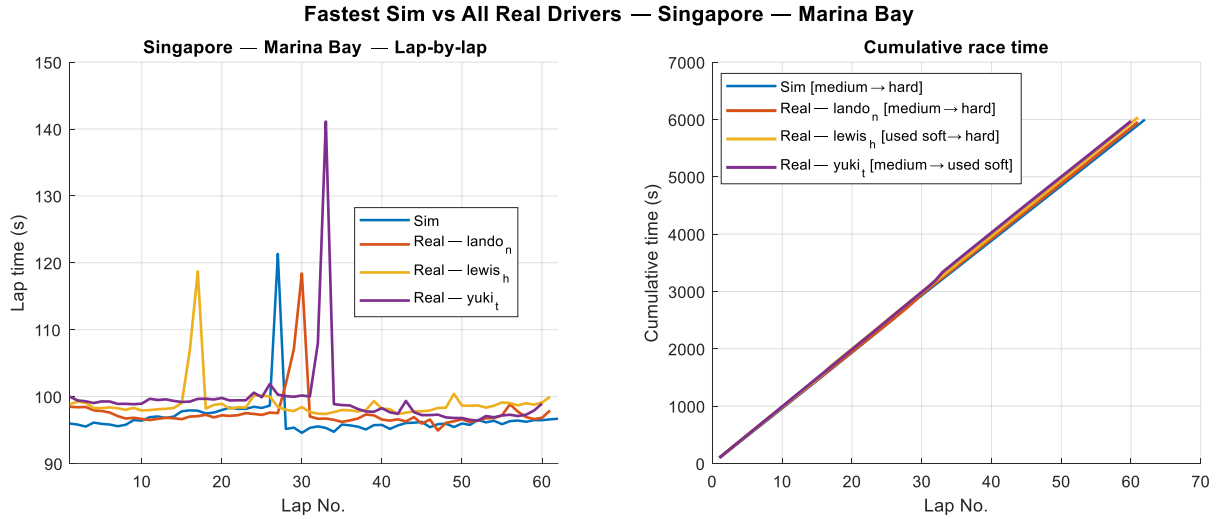


Figure 37 - Fastest simulated car vs. Lando Norris, Lewis Hamilton and Yuki Tsunoda at Marina Bay

To further analyse the simulation and validate, the behavioural trends of the simulation are calculated as opposed to exact replicate of race results: metrics emphasize shape (offset-free RMSE), degradation trend, pit-timing alignment and uncertainty coverage (Press, et al., 2007), see Appendix A for list on analysis metrics. See equations 17 through 21 for the relevant calculations required for the following analysis.

$$RMSE_{shape} = \sqrt{\frac{1}{N} \sum_{t=1}^N \left[(\hat{T}_t - \bar{\hat{T}}) - (T_t - \bar{T}) \right]^2} \quad \text{Equation 17}$$

$$\beta_s^{sim} = \frac{\sum_{j \in s} (j - \bar{j}_s) (\hat{T}_j - \bar{\hat{T}}_s)}{\sum_{j \in s} (j - \bar{j}_s)^2}, \beta_s^{real} = \frac{\sum_{j \in s} (j - \bar{j}_s) (T_j - \bar{T}_s)}{\sum_{j \in s} (j - \bar{j}_s)^2} \quad \text{Equation 18}$$

$$Bias_{trend} = \frac{1}{S} \sum_{s=1}^S (\beta_s^{sim} - \beta_s^{real}) \quad \text{Equation 19}$$

$$\Delta_{pit}^{mean} = \frac{1}{k} \sum_{k=1}^k \hat{l}_k - l_k \quad \text{Equation 20}$$

$$Coverage_{(1-\alpha)}(\%) = \frac{100}{N} \sum_{t=1}^N 1 \left\{ T_t \in \left[q_t^{\alpha/2}, q_t^{1-\alpha/2} \right] \right\} \quad \text{Equation 21}$$

Once simulated, the following values are returned. The mean slope bias ($Bias_{trend}$) returns a value of 0.0515 seconds per lap, indicating the degradation trend per stint is very close to the

true data. This says that both thermal push modelling along with base degradation are working at a high accuracy in comparison to expected trends, meaning not further work is required to enhance this area.

The $RMSE_{shape}$ ignores absolute offsets yet still yields a value of 4.236 seconds for the race and 4.209 seconds per stint. Meaning the lap-to-lap structure is not matching real traces. There are several factors which may be contributing to this, beginning with the issue surrounding standing starts, as discussed previously. To validate whether this is to main issue for the $RMSE_{shape}$ it must be fixed to analyse whether this is causing the issue or whether there are other factors affecting. Pit timing alignment came out with a mismatch of one lap, which is very good, as pit times are not predetermined, but rather defined as a 'pit window', this shows very good agreement.

To reduce $RMSE_{shape}$, a condition is applied to add 3% of lap times to the first 3 laps to simulate a standing start along with a high traffic phase of the race. Once implemented $RMSE_{shape}$ drops to 4.220 seconds for the race and 4.200 seconds per stint, these are improvements yet extremely marginal. Other ways to improve $RMSE_{shape}$ revolve around fine tuning of driver performance jitter, since this jitter does not affect the outcome of strategy suggestion, these values will remain untouched from their current values. As a benefit, from a visual point, the representation of lap times has a closer alignment to real data as seen comparing figure 37 to figure 38.

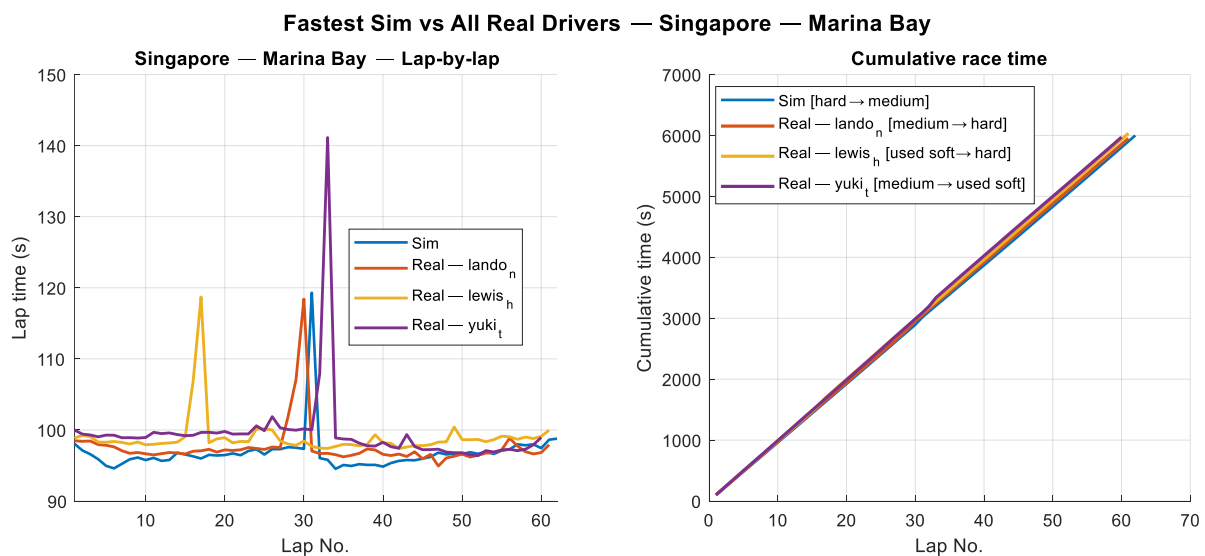


Figure 38 - Fastest simulated car vs. Lando Norris, Lewis Hamilton and Yuki Tsunoda at Marina Bay after implementation of standing start and warm up phases

5.3. Traffic modelling integration

Implementation of a traffic model within the current Monte Carlo strategy tool is done through simulating a grid of 19 fixed vehicles to be saved for use at a later stage. An iterative process

is then begun where new strategies will be simulated and have the appropriate traffic penalties applied where interactions are found. The traffic penalties are applied and analysed from a distribution for each strategy. This process is simulated using the existing Monte Carlo framework and cropping algorithm to find the optimal strategy when competing against a grid of vehicles. Initial results, shown in figure 39, highlight large penalties throughout the race, with the resulting strategy finishing last. From the distribution used in figure 25 A, a mean traffic penalty is seen to be 1.48 seconds, seen in figure 39 a mean traffic loss throughout a race distance is 69.66 seconds meaning roughly 47 interactions are seen. This may be a result of the large grouping at the beginning of the race.

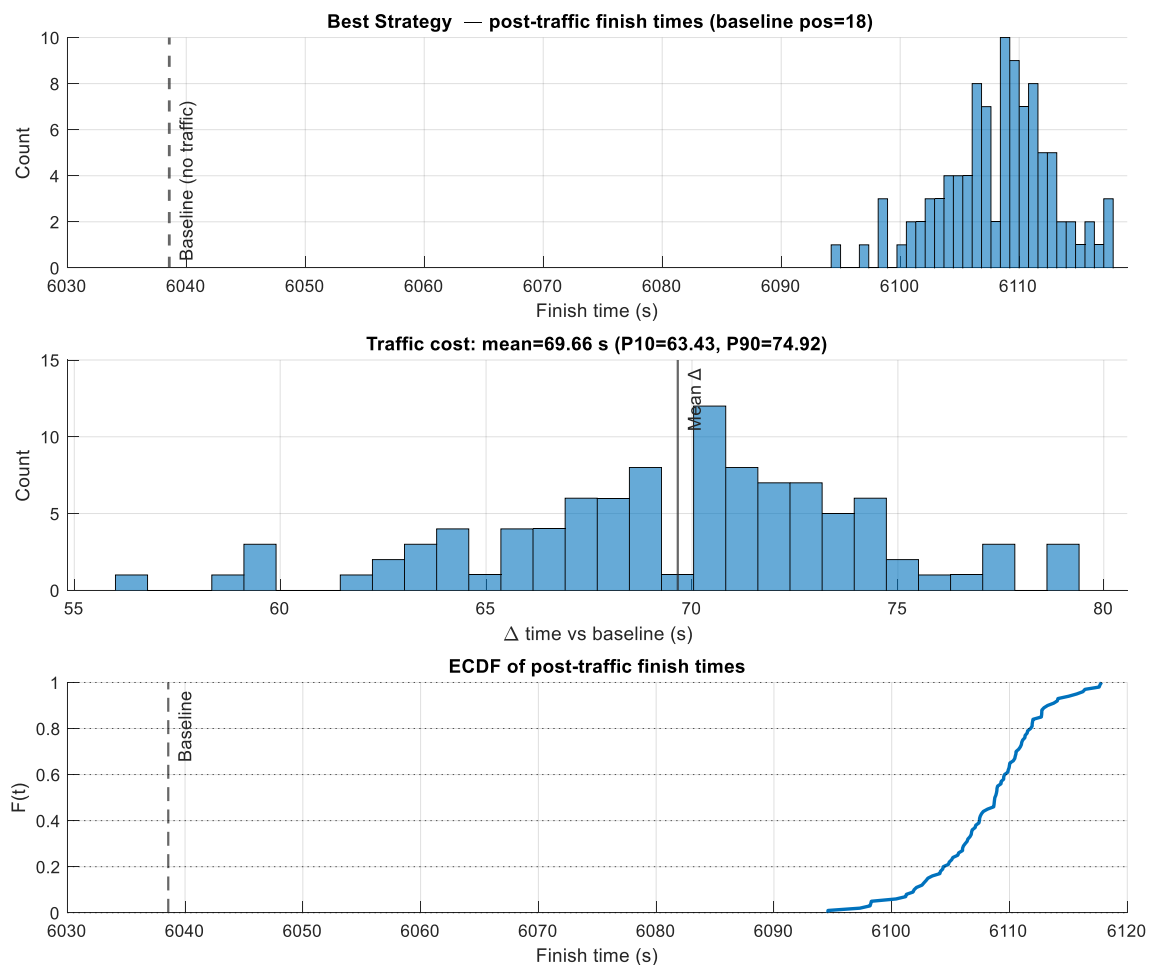


Figure 39 – Pre and post traffic penalty analysis

Throughout the lap-time decomposition, figure 40, the baseline trace is approximately at 95–101 s with a pronounced spike at ~lap 31 because of pit loss/warm-up, and the traffic-affected mean is close to it but surrounded by a tight P10–P90 band, and a normal draw contours over that mean precisely. Cumulatively, the traffic curve strays from baseline near-linearly, representative of many small losses rather than a handful of catastrophes, and the thin uncertainty envelope ensures that there is low dispersion of results. The single realisation

shares the same results, figure 41: the prevalence of numerous ~0.2–1.0 s stints of uplift by non-traffic reason with lap punishment maintaining, again, penalty impulses mostly <1.5 seconds with occasional peaks of 3 to 4 seconds, once again clumped in those same windows. The tactic is naturally expensive in traffic in those very phases. Improvements to the modelling can come from increasing iteration number to verify whether Monte Carlo method allows evading through massive simulations or whether large amounts of interactions are caused by poor conditional blocking.

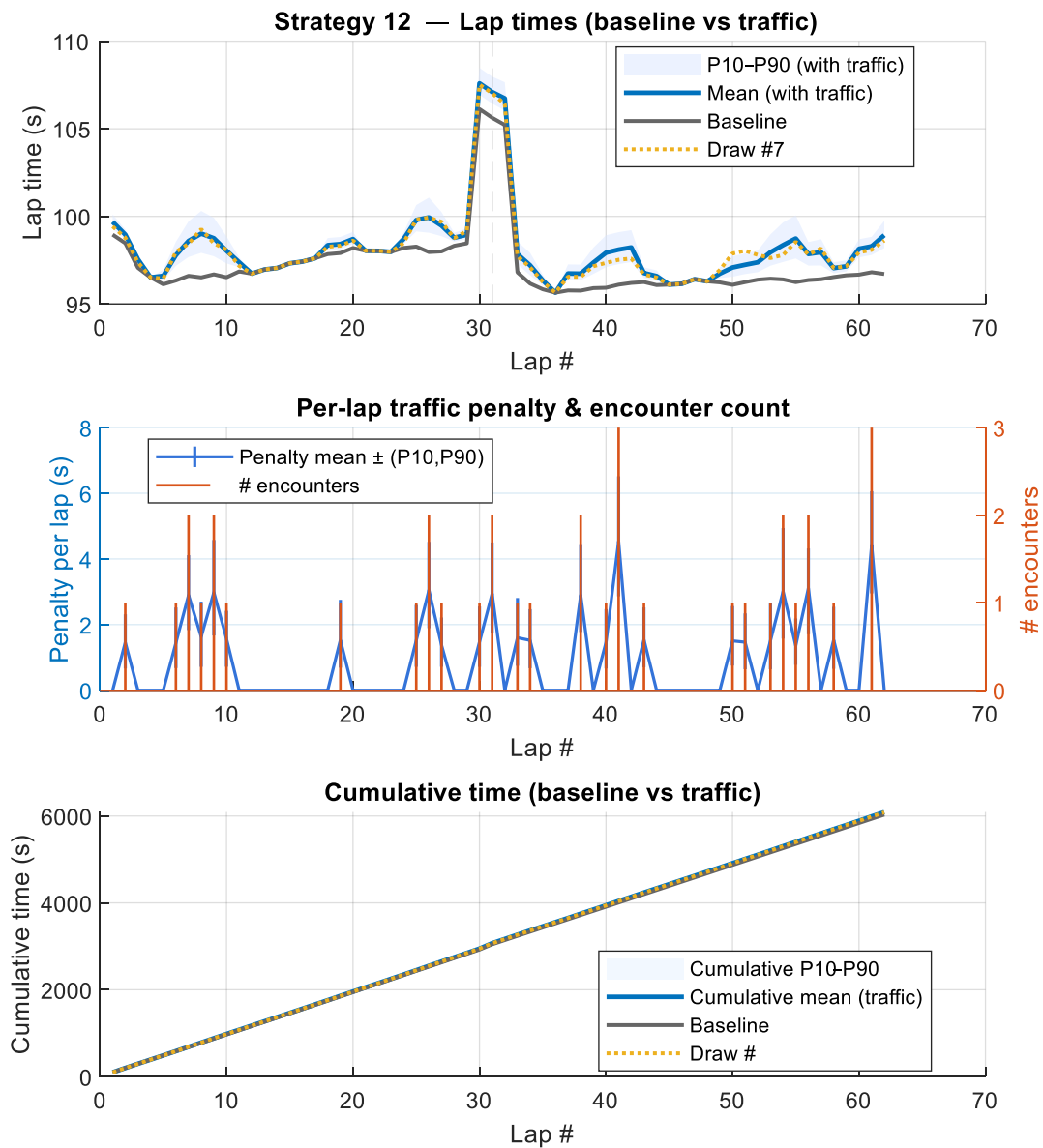


Figure 40 - Lap time analysis for traffic interaction modelling, comparing baseline run to added traffic

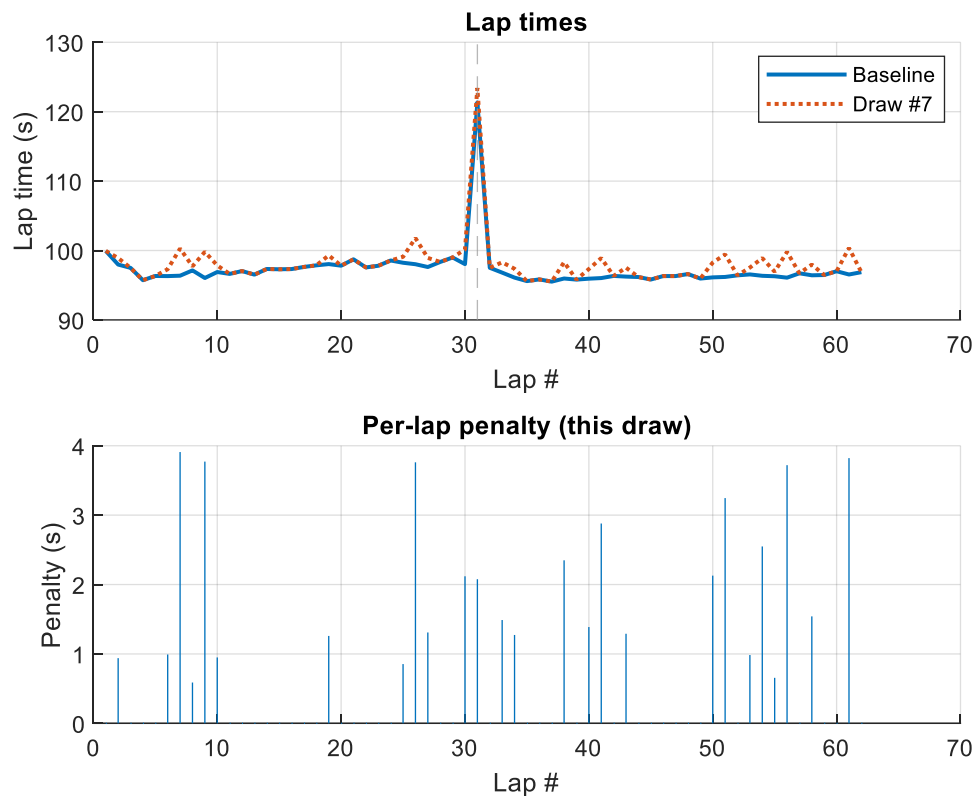


Figure 41 - Single representation of lap time with traffic penalties against baseline run

With an increase in the iteration count as the search moves into an increasingly large strategy space, results do get better in general; yet figure 42 indicates a structural defect with the current traffic layer. The penalty justification over-saturates at race-start: as cars are bunched with minimal gap, the proximity rule sends a high volume of "encounters" on nearby samples and between several opponents, and as a result, many penalties are sent on the same first few laps. This makes for an excessive traffic cost at the front that determines the outcome, and which is not characteristic of single overtakes. The effect is compounded by first, the multiplicity of occasion when several detections against several ghosts are being separately counted over the same lap. Second, asymmetry when the target is always treated as the penalised party while the ghosts remain unaffected, and finally, discretisation sensitivity (a fixed time grid at standing start over-sample the dense pack). Consequence is that early-lap penalties can reach high magnitudes, mid-race traffic effects are under-weighted, and strategy rank is over-dominated by start-phase artefacts in lieu of important mid-stint interactions. In short, the current demonstration mingles opening-lap pack proximity with actual traffic management during the latter race, distorting the estimated traffic penalties upwards and towards the original stint.

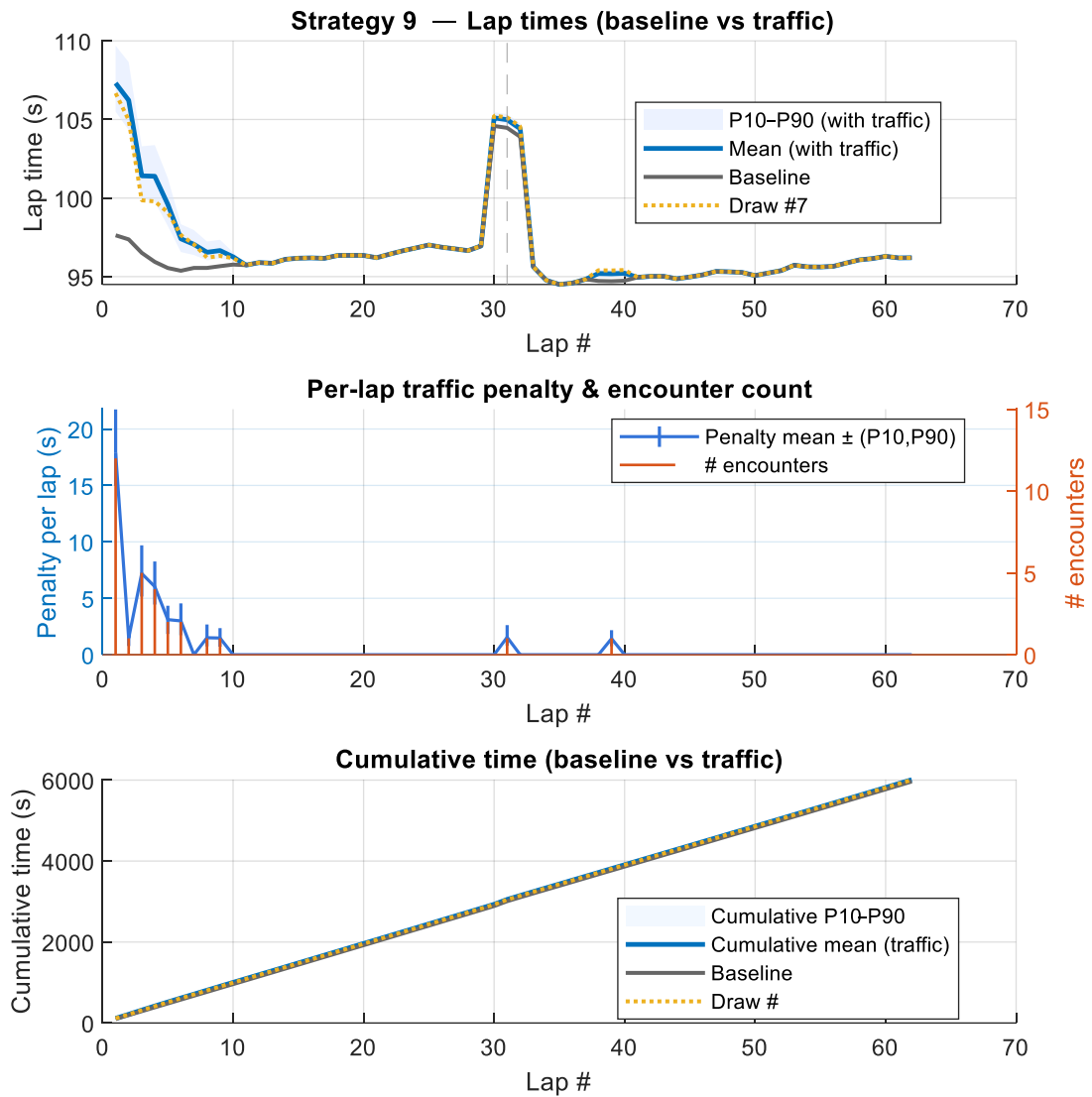


Figure 42 - Highlighting extreme penalties in early laps from traffic model

The current proximity rule tallies up more collisions when the field is close together at the start, piling up penalties on early laps. An effective solution would model traffic as hysteresis overtake episodes: an episode begins when the target vehicle closes from a larger gap d_{hi} to a smaller gap d_{lo} with positive relative velocity, lasts for a short passing window, and ends as soon as separation exceeds d_{hi} ; a penalty is imposed per episode. Concurrent observations in the same window would be additive under a per-lap cap and a quick per-rival cool-down to avoid multiplication in trains. A grid-start penalty would require more compelling closing-rate evidence early laps to avoid punishing pack randomness. Parameters would be calibrated against the reference overtake counts and phase-of-race traffic cost distribution to avoid early-lap inflation without compromising fidelity for mid-stint interactions.

5.4. Endurance race strategy implementation and adaptations

Endurance racing differs from sprint formats in three main ways: First driver rotations and stint time limits, secondly, tyres and fuel are preserved at a much larger rate meaning to modelling of them is very different in comparison to sprint racing and finally multiclass traffic and day to night transitions apply further uncertainties around driver performance. To adapt the simulation to cater to these needs, structural changes are required in the form of more constraints, more state carried across stints, and stochastic processes that evolve over hours rather than laps.

5.4.1. Multi-driver roster

The addition of a driver layer including alternate base times for drivers, fatigue and warm up modifiers along with maximum continuous drive time and maximum total drive time per driver to match series specific regulations (Campbell Racing, 2024) (Circuit of the Americas, 2024). This would mean strategy becomes a joint optimisation of pit management and driver use optimisation.

5.4.2. Refuelling and stops

The addition of a refuelling model would be required due to the greater distances of racing. This model would be dependant on fuel load, a per lap burn curve along with pit refuelling times where series regulations may dictate parallel refuelling and tyre changes or sequential (Bekker & Lotz, 2009) (Garcia, 2024). This would also introduce another pit decision trigger alongside tyre degradation, which may dictate whether tyre changes and refuelling are required at the same point.

5.4.2.1. Fuel preservation

An additional layer of complication comes from the idea of fuel preservation, where the model would be allowed to increase lap times, simulating fuel preservation where that may benefit the vehicle where they may save a pitstop. On the contrary, if a driver is coming to the end of a stint, and low on fuel, as a pitstop for a driver change is already a necessity, decrease lap times whilst fuel load is low for a compounded effect (24H Le Mans, 2013).

5.4.3. Tyre allocation across event

There are alternate tyre allocations for a full weekend of endurance racing based on race length (e.g., 6h/8h/10h/24h tyre caps) (FIA, 2025). Modelling tyre use based on the weekend allocation along with reuse of testing/qualifying tyres throughout the race adds another layer of realism to the simulation, decisions now trade lap pace against conserving sets for the night and the finish.

5.4.4. Multiclass traffic as a persistent stochastic process

Simulating a lap position model would benefit the accuracy of the simulation allowing for more accurate traffic prediction based on race time and progression from different classes, introducing track-segment-dependent losses and heavier tails for bad clusters (Heilmeier, et al., 2020) (Paparusso, et al., 2022). Additionally, this would allow for the decisions for timing stops to avoid traffic and join the circuit in free air. Despite the obvious benefit of introducing a lap position model, the increase in computational requirement from this would be very large as reported by (Scardino, 2025).

5.4.5. Night to day performance differences

All pace, push and degradation could have the ability to be modelled as a function of the time of day, simulating changing track temperatures, visibility and fatigue levels. This opens the possibility for driver suggestions based on performance under different scenarios where a pro driver would perform better during the harder hours of racing, along with scheduling tyres that tolerate cooler track temps.

5.4.6. Objectives and evaluation of endurance racing

Another simple modification required to implement endurance racing strategy is the calculation of maximum distance rather than 'best time' wherever they may differ dependant on event. Outputs of the simulation could cover risk of strategy dependant on unknown external factors and compliance with drive-time rules, showing a safety buffer that may be altered under race conditions where true driver performance can be evaluated. These adaptations show that while the current model has been validated on sprint-style races, its framework is directly extensible to endurance racing where multi-driver, refuelling, and allocation rules dominate strategy.

5.5. Simulation shortcomings

Despite previous arguments relating to the accuracy of the simulation, the decision on when to pit is not one set in stone prior to a race, rather one made during the race dependant on numerous factors. These factors, including track evolution, traffic/clean air windows and rival behaviour, would determine whether an undercut/overcut strategy would best benefit a driver in the race, as discussed in 3.3.1 (Catapult, 2024) (Hall, 2023). Yet with the introduction of a safety car, these decisions could be changed at moment's notice where a new compound would significantly benefit a struggling driver on highly degraded tyres. Evidence of this is present from figure 43 which shows where the simulation cannot give accurate outcomes with the implementation of a safety car. This simulation is compared with the 2025 Spanish GP where there was a period towards the end of the race under safety car restrictions which

resulted in all drivers opting to pit for a different set of tyres, see table 10. The strategy decisions made in the 2025 Spanish GP cannot be predicted due to the introduction of the safety car towards the end of the race, despite this, prior to the safety car there are still large disagreements in strategy between simulation real strategies. Where the simulation decides that a hard compound would be best suited to start on, moving over to soft and then back to hard. The only time a hard compound is used within the race is the final pit stop forced for Verstappen towards the end of the race where he had to run on a hard compound due to it being the only remaining allocated tyre set for the weekend of testing, qualifying and racing, see table 10. The one correct decision from the simulation is the choice of multiple stops which is indicated by the short time loss due to pitting. Despite this the decision for soft tyres is only made once in comparison to the strategy chosen for Pierre Gasly which included three sets of used soft tyres. With some research on the specific tyres used for the 2025 Spanish GP, it is found that Pirelli provided the hardest trio of tyres for this race (C1/C2/C3) due to the tracks high energy on long corners which pointed towards a two-stop strategy using the soft-medium-soft compounds (Pirelli, 2025), see figure 3 for tyre compound graphic. Supporting this McLaren stated that teams found the hard compound ‘lacked grip and slid more’, degrading nearly as quickly as soft and medium compounds with significantly worse lap times, alongside the undercut dynamics found at the Spanish GP where the pit-loss and tyre warm-up profiles make the undercut a ‘powerful but short-lived’ strategy (McLaren, 2025).

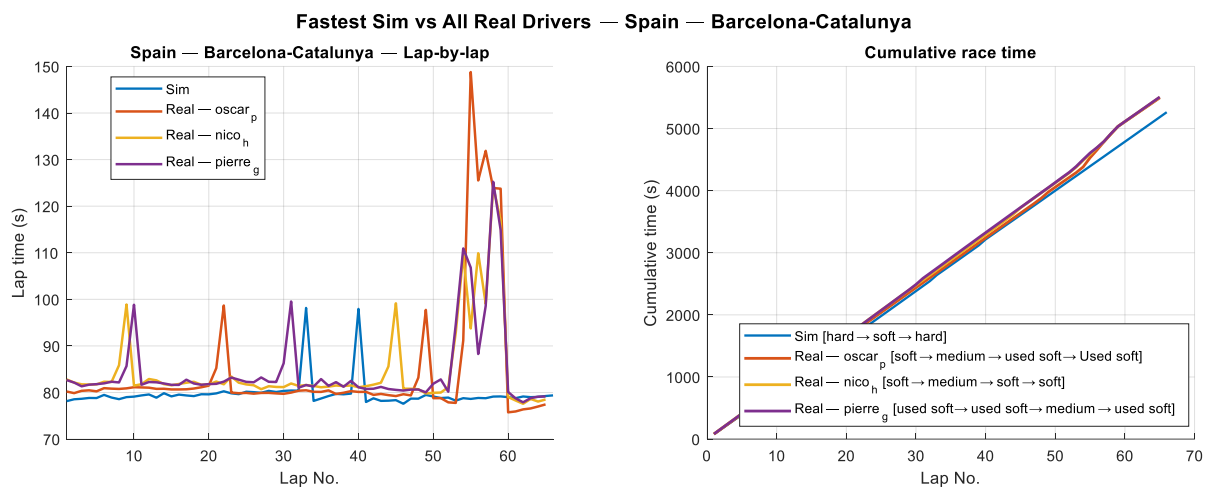


Figure 43 - Fastest simulated car vs. Oscar Piastri, Nico Hulkenberg and Pierre Gasly at Catalunya

Table 10 - Tyre strategy Decisions from 2025 Spanish GP for all competitors

Driver	Team	Start	Stint 2	Stint 3	Stint 4	Stint 5
Piastri	Mclaren	Soft	Medium	Used Soft	Used Soft	NA
Norris	Mclaren	Soft	Medium	Used Soft	Used Soft	NA
Leclerc	Ferrari	Used Soft	Medium	Medium	Used Soft	NA
Russell	Mercedes	Used Soft	Medium	Soft	Used Soft	NA
Hulkenberg	Stake	Soft	Medium	Soft	Soft	NA
Hamilton	Ferrari	Used Soft	Medium	Used Soft	Used Soft	NA
Hadjar	Racing bulls	Used Soft	Medium	Used Soft	Used Soft	NA
Gasly	Alpine	Used Soft	Used Soft	Medium	Used Soft	NA
Alonso	Aston martin	Used Soft	Used Medium	Used soft	Used Soft	NA
Verstappen	Red bull	Used Soft	Soft	Medium	Used Soft	Hard
Lawson	Racing bulls	Soft	Medium	Used soft	NA	NA
Bortoleto	Stake	Used Soft	Medium	Soft	NA	NA
Tsunoda	Red bull	Medium	Soft	Soft	Soft	Used Soft
Sainz	Williams	Soft	Soft	Medium	Soft	NA
Colapinto	Alpine	Soft	Medium	Soft	Used Soft	NA
Ocon	Haas	Soft	Medium	Medium	NA	NA
Bearman	Haas	Soft	Medium	Medium	Used Soft	NA
Antonelli	Mercedes	Used Soft	Medium	Used Soft	NA	NA
Albon	Williams	Used Soft	Soft	NA	NA	NA

One of the main take aways from this segment is the lack of validation for tyre modelling and their performance around different circuits, this issue is very hard to solve with the lack of interpretive data available for tyre degradation that removes many of the external factors at play with their influence on lap times. This highlights the main area for improvement within the model where it can simply predict lap times for a baseline performance at a steady 'push rate' where degradation is steady. A more robust model which involves tyre temperature calculations would allow improved tyre degradation modelling and allowing for a more robust push model to be integrated, yet this data simply is not available. Once more the computational limitations would play an increased factor with the required calculations to make this theoretical thermal model possible. This shortcoming does not undermine the probabilistic approach itself but shows where real world tyre data is the limiting factor

6. Conclusion

This research set out to design and develop a probabilistic race-strategy optimisation system that included tyre degradation, fuel mass effects, pit-stop variability, and traffic interference within a Monte Carlo framework. A deterministic lap-by-lap MATLAB model is built and progressively expanded: a tuneable degradation curve replaced a too simplistic baseline model (West & Limebeer, 2020) (Piccinotti, 2020); a logarithmic relationship between fuel mass yielded enhanced late-stint sensitivity compared to a linear surrogate (Bekker & Lotz, 2009); and the two effects combined to simulate realistic pace development (Heilmeyer, et al., 2020). Correlation with race data showed thus, having removed green-flag conditions and post-stand-start/warm-up stages, the simulator replicated degradation patterns and pit-timing behaviour with reasonable accuracy, though with less-than-perfect absolute lap-to-lap formation, particularly in safety car and event-related tyre behaviour influenced situations.

The chief methodological improvement is an MC engine combined with a progressive cropping algorithm that enables very large strategy searches at affordable cost. Across seeds and grid sizes, cropping yielded ~24–27% wall-time savings at good selection fidelity (Overlap @ K typically ~0.95–1.00; Spearman's ρ ~0.93–0.99) and modest timing error for survivors (MAE ~9–14 s) (Jamieson & Talkwalker, 2016) (Li, et al., 2018) (Zar, 2005). Seed-sweep experiments confirm stability, with the speed improvements not coming at the expense of compromising genuinely competitive strategies (Kong, et al., 2003) (Press, et al., 2007). While the current implementation increases memory pressure state is stored between the elimination rounds, the performance–accuracy trade-off is acceptable for large-N exploration, and the storage implementation can be re-engineered in future work (Mathworks, 2025).

Traffic interaction is included by simulating a frozen field of competitors and applying probabilistic, per-event penalties where encounters occur (Paparusso, et al., 2022) (Heilmeyer, et al., 2020). However, the known weakness of the model prevents acceptable calibration; one possible remedy using redefinition of traffic as hysteretic overtake events with event accumulation, per-lap quotas, and a grid-start guardian, is outlined for future consideration (Zhou, et al., 2022) (Li, et al., 2018).

A second important empirical finding is the failure to converge to a single "best" strategy even at very high levels of iteration. Convergence studies on a variety of circuits indicate the presence of numerous near-optimal plans, with high between-seed agreement on the set but not on a single winner (Henderson, et al., 2019) (Song, et al., 2025). This is consistent with the intrinsic multi-modality of race strategy under uncertainty: small timing adjustments in stops, compound order, or traffic phasing can yield statistically indistinguishable totals. The tool therefore supports decision-making by measuring distributions, margins, and robustness,

not by prescribing a single deterministic solution, in a manner consistent with the way elite teams conceptualise risk and resilience (Hatton, 2024).

Overall, the work provides a believable, extensible probabilistic strategy analysis framework. It combines a physically motivated lap-time kernel, stochastic components calibrated where data permit, and a compute-efficient search algorithm scaling to millions of candidates (Polat, et al., 2021) (Carrazza & Cruz-Martinez, 2020). Constraints like safety-car logic, tyre-temperature/thermal models, start-phase traffic artefacts, and memory overheads continue to hold. Its application to endurance racing is conceptually straightforward: append multi-driver rostering and duty-limits, refuelling and parallel/serial pit-lane constraints, tyre-allocation management over the course of an event, diurnal performance envelopes, and continuous multi-class traffic models (Paparusso, et al., 2022) (Heilmeier, et al., 2020). Together, the results demonstrate that MC-based optimisation can generate strategy recommendations that are both fast to compute and robust to the stochastic realities of motorsport, comprising a practical decision-support foundation for sprint and endurance situations alike.

References

- 24H Le Mans, 2013. *Strategy and tyres*. [Online]
Available at: <https://www.24h-lemans.com/en/news/strategy-and-tyres-9999>
[Accessed 07 August 2025].
- Aguad, F. & Thraves, C., 2024. Optimizing pit stop strategies in Formula 1 with dynamic programming and game theory. *European Journal of Operational Research*, 319(3), pp. 908-919.
- Bekker, J. & Lotz, W., 2009. Planning Formula One race strategies using discrete-event simulation. *Journal of the Operational Research Society*, 60(1), pp. 952-961.
- Boettinger, M. & Klottz, D., 2023. *Mastering Nordschleife - A comprehensive race simulation for AI strategy decision-making in motorsport*, Stuttgart: Elsevier.
- Campbell Racing, 2024. *The Role of Strategy in Endurance Racing: More Than Just Speed*. [Online]
Available at: <https://campbellracingteam.com/the-role-of-strategy-in-endurance-racing-more-than-just-speed/>
[Accessed 07 August 2025].
- Carrazza, S. & Cruz-Martinez, J., 2020. *Vegasflow: accelerating Monte Carlo simulation across multiple hardware platforms*, Milan: s.n.
- Catapult, 2024. *Optimizing Race Strategy in Motorsports: The Power of the Undercut and Overcut*. [Online]
Available at: <https://www.catapult.com/blog/motorsports-race-strategy-undercut-overcut>
[Accessed 26 August 2025].
- Circuit of the Americas, 2024. *3 Unique Things About Endurance Racing*. [Online]
Available at: <https://circuitoftheamericas.com/blog/2024/2/23/3-unique-things-about-endurance-racing/>
[Accessed 07 August 2025].
- F1Pace, 2024. *2024 F1 Season: The pit crew*. [Online]
Available at: <https://f1pace.com/p/2024-f1-season-the-pit-crew/>
[Accessed 17 August 2025].
- Ferrari, 2023. *Endurance, from the inside*. [Online]
Available at: <https://www.ferrari.com/en-EN/hypercar/interviews-endurance>
[Accessed 06 September 2025].
- FIA , 2025. *WEC 2025 D28 All Sporting clarification LEMANS*, s.l.: s.n.
- FIA, 2024. *2024 Singapore Grand Prix*. [Online]
Available at: <https://www.fia.com/events/fia-formula-one-world-championship/season->

[2024/singapore-grand-prix/eventtiming-information](#)

[Accessed 29 August 2025].

FIA, 2025. *Japanese Grand Prix*. [Online]

Available at: <https://www.fia.com/documents/championships/fia-formula-one-world-championship-14/season/season-2025-2071/event/Japanese%20Grand%20Prix>

[Accessed 24 June 2025].

Fisher, H., 2011. *A History of the Central Limit Theorem: From Classical to Modern Probability Theory*. 1st ed. New York: Springer.

Garcia, M. P. Q., 2024. *OPTIMIZATION OF PIT STOP STRATEGIES IN FORMULA 1 RACING: A DATA-DRIVEN APPROACH*, Santiago: UNIVERSIDAD DE CHILE.

Giannelos, S., Pudjianto, D. & Strbac, G., 2025. Smart home economic operation under uncertainty: comparing monte carlo and stochastic optimization using gaussian and KDE-based data. *Operations Research Perspectives*, Volume 15.

Glasserman, P., 2004. Variance Reduction techniques. In: *Monte Carlo Methods in Financial Engineering*. New York: Springer, pp. 185-186.

Hall, S., 2023. *F1 strategy explained: What's an undercut, overcut, a DRS train and more*.

[Online]

Available at: <https://www.motorsport.com/f1/news/f1-strategy-explained-whats-an-undercut-overcut-a-drs-train-and-more/10500434/>

[Accessed 07 August 2025].

Hatton, G., 2024. *How Formula 1 teams determine the fastest race strategy*. [Online]

Available at: <https://www.raceteq.com/articles/2024/07/how-formula-1-teams-determine-the-fastest-race-strategy>

[Accessed 06 August 2025].

Heilmeier, A., Graf, M., Betz, J. & Lienkamp, M., 2020. Application of Monte Carlo Methods to Consider Probabilistic Effects in a Race Simulation for Circuit Motorsport. *applied sciences*, 10(2).

Heilmeier, A. M., 2022. *Simulation of Circuit Races for the Objective Evaluation of Race Strategy Decisions*, Munich: TUM School of Engineering and Design.

Henderson, P. et al., 2019. *Deep Reinforcement Learning that Matters*. s.l., Cornell.

Hogg, D. W. & Foreman-Mackey, D., 2017. Data analysis recipes: using Markov Chain Monte Carlo. *The Astrophysical Journal Supplement Series*, 236(1).

Jamieson, K. & Talwalkar, A., 2016. *Non-stochastic Best Arm Identification and Hyperparameter optimization*. Berkeley, s.n.

Kampen, J., Hermann, T. & Salazar, M., 2022. *Maximum-distance race strategies for a fully electric endurance race car*, s.l.: s.n.

Kenton, W., 2024. *Monte Carlo Simulation: What it is, How it works, History, 4 key steps.*

[Online]

Available at: <https://www.investopedia.com/terms/m/montecarlosimulation.asp>

[Accessed 24 July 2025].

Kong, A. et al., 2003. A Theory of Statistical Models for Monte Carlo Integration. *Journal of the Royal Statistical Society*, 65(3), pp. 585-604.

Kwak, S. G. & Kim, J. H., 2017. Central limit theorem: the cornerstone of modern statistics. *Korean Journal of Anesthesiology*, 70(2), pp. 144-156.

Li, L. et al., 2018. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research*, Volume 18, pp. 1-52.

Liu, M. et al., 2023. Potential Game-Based Decision-Making for Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, 24(8), pp. 8014 - 8027.

Luengo, D. et al., 2020. A survey of Monte Carlo methods for parameter estimation. *EURASIP Journal on Advances in Signal Processing*, Volume 25.

Mathworks, 2025. *Strategies for Efficient Use of Memory.* [Online]

Available at: https://www.mathworks.com/help/matlab/matlab_prog/strategies-for-efficient-use-of-memory.html

[Accessed 30 August 2025].

Mclaren, 2025. *Why the reactive strategy wouldn't have been the right strategy in Spain – presented by FxPro.* [Online]

Available at: <https://www.mclaren.com/racing/formula-1/2025/spanish-grand-prix/strategy-debrief/>

[Accessed 01 September 2025].

Mobile Pitstop, 2022. *Tire Management: Rolex 24 Hours At Daytona.* [Online]

Available at: <https://www.mobilepitstopvan.com/2022/01/17/tire-management-rolex-24-hours-at-daytona>

[Accessed 06 August 2025].

Motor Sport, 2025. *2025 F1 testing — new tyre compounds but only 2 teams choose wets for Bahrain.* [Online]

Available at: <https://www.motorsportmagazine.com/articles/single-seaters/f1/2021-f1-testing-tyres-why-pirelli-has-brought-a-prototype-to-bahrain/>

[Accessed 01 September 2025].

Paparusso, L., Riani, M. & Ruggeri, F., 2022. *Competitors-aware stochastic lap strategy optimisation for race hybrid vehicles*, s.l.: s.n.

Piccinotti, D., 2020. *Open loop planning for Formula 1 race strategy identification*, Milan: POLITECNICO DI MILANO.

- Pipis, G., 2019. *Estimate Pi with Monte Carlo*. [Online]
Available at: <https://predictivehacks.com/estimate-pi-with-monte-carlo/>
[Accessed 06 August 2025].
- Pirelli, 2025. *The hardest tyres are back for Spain*. [Online]
Available at: <https://press.pirelli.com/the-hardest-tyres-are-back-for-spain>
[Accessed 01 September 2025].
- Pitpass.com, 2024. *Singapore Grand Prix: Driver Tyre Strategies*. [Online]
Available at: <https://www.pitpass.com/78788/Singapore-Grand-Prix-Driver-Tyre-Strategies>
[Accessed 29 August 2025].
- Polat, O., Eyuboglu, O. H. & Gul, O., 2021. Monte Carlo simulation of electric vehicle loads respect to return home from work and impacts to the low voltage side of distribution network. *Electrical Engineering*, 103(1).
- Press, W., Teukolsky, S., Vetterling, W. & Flannery, B., 2007. *Numerical Recipes: The art of Scientific Computing*. 3rd ed. New York: Cambridge University Press.
- Red Bull Racing, 2022. *Bulls' Guide To: Pit Stops*. [Online]
Available at: <https://www.redbullracing.com/int-en/bulls-guide-to-pit-stops>
[Accessed 07 August 2025].
- Reuters, 2025. *F1's Monaco Grand Prix to have mandatory two stops*. [Online]
Available at: <https://www.reuters.com/sports/formula1/f1s-monaco-grand-prix-have-mandatory-two-stops-2025-02-26/>
[Accessed 06 September 2025].
- Scardino, A. L., 2025. *Probabilistic Simulator for Race Event Forecasting and Based on Telemetry and Historical Data*, s.l.: POLITECNICO DI TORINO.
- Song, X., Zou, L. & Tang, M., 2025. An Improved Monte Carlo Reliability Analysis Method Based on BP Neural Network. *applied sciences*, 15(8).
- Statathlon, 2016. *A Thorough Analysis of the Pit Stop Strategy in Formula 1*. [Online]
Available at: <https://statathlon.com/analysis-of-the-pit-stop-strategy-in-f1>
[Accessed 06 August 2025].
- Taboga, M., 2025. *Markov Chain Monte Carlo (MCMC) methods*. [Online]
Available at: <https://www.statlect.com/fundamentals-of-statistics/Markov-Chain-Monte-Carlo>
[Accessed 06 August 2025].
- Vehtari, A. et al., 2020. Rank-Normalization, Folding, and Localization: An Improved R for assessing convergence of MCMC. *Bayesian Analysis*, pp. 1-28.
- Wachenheim, D., Patterson, J. & Ladisch, M., 2003. Analysis of the logistic function model: derivation and applications specific to batch cultured microorganisms. *Bioresource Technology*, 86(2), pp. 157-164.

WEC, 2018. *What happens in WEC pitstops?*. [Online]

Available at: <https://www.fiawec.com/en/news/what-happens-in-wec-pitstops/5911>

[Accessed 06 September 2025].

West, W. J. & Limebeer, D. J. N., 2020. *Optimal Tyre Management of a Formula One Car*.

Berlin, Preprints of the 21st IFAC World Congress (Virtual).

Zar, J. H., 2005. Spearman Rank Correlation. In: *Encyclopedia of Biostatistics*. s.l.:s.n.

Zhang, J., 2020. *Modern Monte Carlo Methods for Efficient Uncertainty Quantification and Propagation: A Survey*, s.l.: s.n.

Zhou, W. et al., 2022. Multi-agent Reinforcement Learning for Cooperative Lane Changing of Connected and Autonomous Vehicles in Mixed Traffic. *Autonomous Intelligent Systems* , 2(1).

Appendices

Appendix A

Full Name	Abbreviation	Comments
Root Mean Square Error	RMSE	Average squared difference between predicted and observed values. Used to validate lap-time trace shape.
Mean Absolute Error	MAE	Average absolute difference between predicted and observed values. Interpretable in seconds.
Slope Bias		Difference in rate of change (trend) between simulation and real lap-time degradation.
Pit-Timing Alignment		Mismatch in pit-stop timing between simulated and real races, reported in laps.
Jaccard Index		Overlap of strategy sets across seeds/runs, defined as $ A \cap B / A \cup B $.
Overlap at K	Overlap@K	Fraction of strategies appearing in both top-K lists across seeds or methods.
Spearman's Rank Correlation	Spearman's ρ	Rank correlation measure between two ordered lists of strategies.
Shannon Entropy	Entropy (H)	Measures unpredictability/diversity in strategy distribution; higher = more diverse.
Gini-Simpson Index		Diversity index defined as $1 - \sum p_i^2$; closer to 1 = more diversity.
Dominant Share		Probability mass of the single most likely strategy. Indicates strategy convergence.
Computational Wall Time		Elapsed real time required to complete a simulation run.
Speedup		Ratio of no-optimisation runtime to optimised runtime. >1 = faster.
Memory Usage (Peak)		Maximum memory consumed during execution of the simulation.
Memory Usage (Average)		Mean memory consumption during simulation execution.
Memory Usage (Largest Variable)		Size of the biggest in-memory object created during simulation.
Coverage		Extent to which simulated distributions include observed race outcomes.