AE4350
Bio-Inspired Intelligence and Learning for Aerospace Applications

Assignment Report

Author:
Reggie Johanes (5477506)

31 August 2023

Contents

Github Link

https://github.com/reggiejohanes/ae4350assignment2023

Abbreviation and Symbols

- α or "Alpha": Learning rate, learning algorithm hyperparameter.
- γ or "Gamma": Discount factor, learning algorithm hyperparameter.
- ϵ or "Epsilon": Probability of choosing explorative actions.
- S_t: State at time step t.
- A_t: Action at time step t.
- R_t: Reward at time step t.
- $Q(S_t, A_t)$: Expected rewards (Q) for action A_t taken at state S_t.
- VER: Driver code for Max Verstappen.

1. Introduction

The objective of this assignment is to apply reinforcement learning to solve a real-life problem. Reinforcement learning, like other types of machine learning, aims to solve problems for which an exact solution would otherwise be too impractical, time consuming, or resource intensive to develop. A reinforcement learning program works by repeatedly interacting with a simulated environment which represents the real problem to execute a certain task. The program chooses different actions which allows it to navigate a mathematical problem space and receives feedback, also known as "rewards", based on the actions that are chosen. It then corrects itself based on these rewards and updates its preferred actions to optimize the rewards that will be received. After a certain number of episodes, an optimal solution will be found which produces the highest possible reward.

In general, there are two types of reinforcement learning problems: discrete and continuous problems. Discrete problems have exact states and actions, for example in a game of chess there are exact coordinates for each position that can be held by a piece, and each piece can only move in an exact, specific manner. On the other hand, continuous problems have continuously varying states and/or actions. An example of a continuous problem is aircraft control, where an aircraft's state (its degrees of freedom) may change continuously along with its actions (control inputs). For this assignment, the problem that was chosen is represented as a discrete problem, and hence a relevant tabular solution algorithm was chosen.

2. Problem Description

The problem to be solved in this assignment is the calculation of the optimal strategy for a Formula One race. A Formula One race consists of numerous repeated laps around a closed circuit, throughout which each car will typically use multiple sets of tires. This is needed because Formula One tires, although high performance, cannot last through an entire race distance. As they are used, the tires degrade and will result in slower lap times as the race progresses. Hence, it is beneficial to switch to new tires at some point in the race. As pitting requires additional time, it becomes a trade-off between the time gained by using a newer set of tires and the time lost in the pit lane.

In addition, there are also other mechanics which come into play. There are three tire compounds that are available at each race, a soft compound (color-coded red), medium compound (yellow), and hard compound (white). Softer tyres generally provide more traction and hence will result in a faster lap time. However, they are less durable and hence better suited for shorter stints compared to harder compounds. Furthermore, Formula One regulations state that each car must use at least two different compounds during a race. With all these factors in mind the race strategy can therefore be defined as the determination of when to perform pit stops and which tires to use.

To solve this problem, a simplified race model is constructed which calculates each lap time throughout the race. At each lap, the lap time is calculated based on the current state and the action that is taken. The action space is comprised of four possible actions: one to stay on track and continue to the next lap, and three to pit for a new set of one of the available tire compounds. The state space is modelled using four parameters: lap number, tire age (in laps), tire compound, and number of unique compounds used.

For this assignment, the 2023 Spanish Grand Prix is used as an example. The race has 66 laps, which means there are 66 x 66 x 3 x 3 = 39,204 numerically possible states. However, as the simulation is always started with the assumption that new tires are used on the first lap, the number of lap number – tire age combinations is reduced. This brings the total number of realistic states to 19,899, with 79,596 realistic state-action combinations.

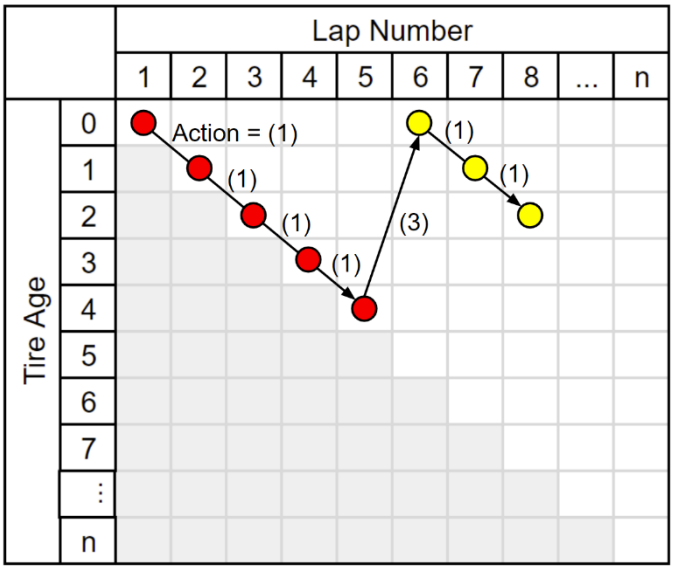

Figure 1. *Simplified illustration of the state space*

In the example shown above, the car starts with a new set of soft compound tires. It pits at the end of the fifth lap for a new set of medium compound tires, thereby resetting the tire age parameter while the lap number continues to progress.

2.1. Limitations

As each circuit has its own unique geometry which may lead to different characteristics in terms of tire performance and pit duration, this assignment will only use a model optimized specifically for the 2023 Spanish Grand Prix. This race was chosen due to the availability of usable data which is needed to create the race model. Additionally, the model used is only optimized to closely replicate the performance of Max Verstappen in his Red Bull Racing RB19 car, as each car and driver combination will also have varying lap time characteristics. Finally, it should also be noted that the problem to be solved in this assignment is vastly simplified compared to an actual Formula One race. For instance, no opponents are considered, hence the car is assumed to be running a pre-determined strategy which does not change dynamically based on the actions of any rivals. The effects of weather, accidents, safety cars, starting position, and pit time variance are also ignored.

3. Methodology

3.1. Race Modelling

In the current work a Formula One race is simulated as discrete laps, where the state (fuel load, tire age, tire compound) of the car and the action taken (pit/stay out) at the start of each lap is used to estimate the lap time produced at that particular lap. Lap times as calculated using the following equation:

$$t_{lap} = t_{base} + t_{tire} + t_{fuel} + t_{inlap} + t_{outlap} + t_{start} + t_{variance} + t_{penalty} \quad [1]$$

Where:

- t_{base} is the reference time which represents the best possible race lap time under ideal fuel and tire conditions. This time is taken from the fastest lap time set during the actual race.
- t_{tire} represents the additional lap time incurred relative to the base time due to the current tire compound being used and the age of the tires.
- t_{fuel} models time loss due to fuel weight.
- t_{inlap} and t_{outlap} represents the time needed to perform a pit stop.
- t_{start} accounts for the additional time needed during the start of the race due to the car's initial stationary condition.
- $t_{penalty}$ is used to model the two-compound rule.

Once lap times for each lap have been calculated, the total race time is simply computed as the sum of all individual lap times.

3.1.2. Fuel Load

Under current regulations, refuelling is not permitted during a Formula One race. Therefore, each car needs to carry enough fuel to last the entire race distance from the start of the race. This means that the cars will be much lighter, and therefore generally faster, towards the end of the race. As the exact amount of fuel carried by each car during a race is not published publicly, assumptions need to be made based on Formula One regulations as well as unofficial sources to account for the effects of the extra fuel weight on the lap times produced. It can be assumed that each car will have 110 kilograms of fuel at the start of the race which will decrease linearly to 0 kilograms at the end of the race. For each kilogram of fuel on board, a lap time loss of 0.03 seconds is assumed (Tracing Insights, 2022).

3.1.1. Tire Performance

To represent the effects of tire wear, it is assumed that the time loss due to tire degradation will increase linearly with respect to tire age. In addition, each compound is assumed to have a certain expected lifetime, after which they will degrade at a much faster rate (also known as the "performance drop off" effect). The y-intercept value, gradients, and drop off point for each compound was chosen to closely match the fuel-

corrected lap times produced by Max Verstappen in the actual race. The resulting curves are shown in the figure below.

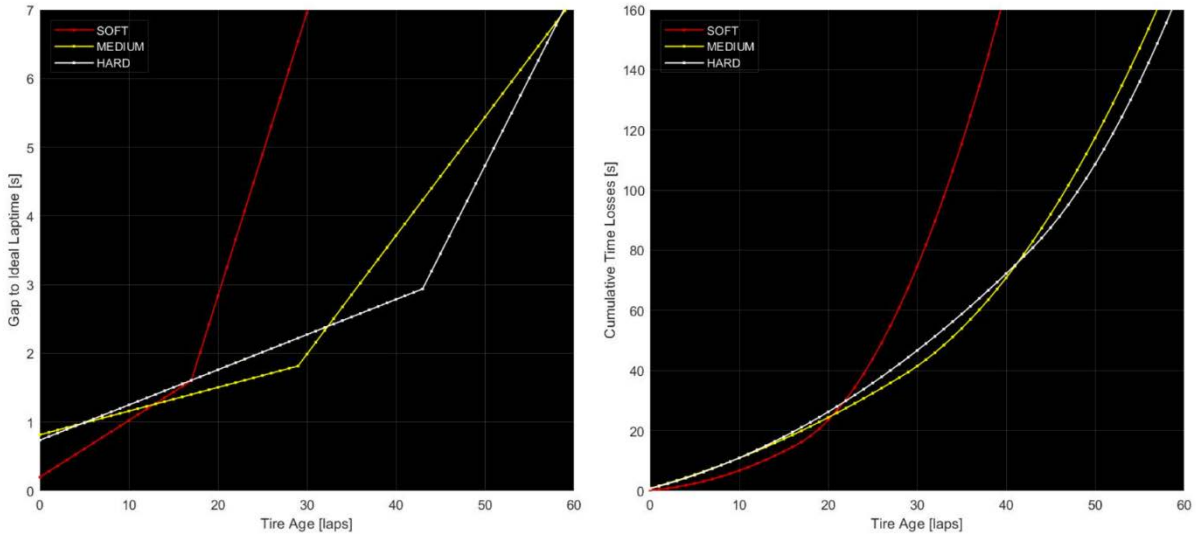

Figure 2. *Additional lap time due to tire wear vs tire age (left) and cumulative time losses due to tire wear (right) by compound.*

It is shown that soft tires start off as the fastest option but have the shortest life by far. It can also be seen that medium compound tires start off as the slowest tire by a small margin compared to hard tires, however they degrade at a moderately slower rate before finally dropping off faster due to their shorter lifetime. Based on the cumulative time losses, it is observed that soft tires will produce the smallest time losses for stints of up to 20 laps. Medium tires are best for stints between 21 to 41 laps, while hard tires are best for stints longer than 41 laps. These characteristics are generally in line with what can be observed during actual Formula One races. To further validate the tire degradation assumptions the model was used to recreate Max Verstappen's race using the same Medium-Hard-Soft tire strategy, showing agreeable results.

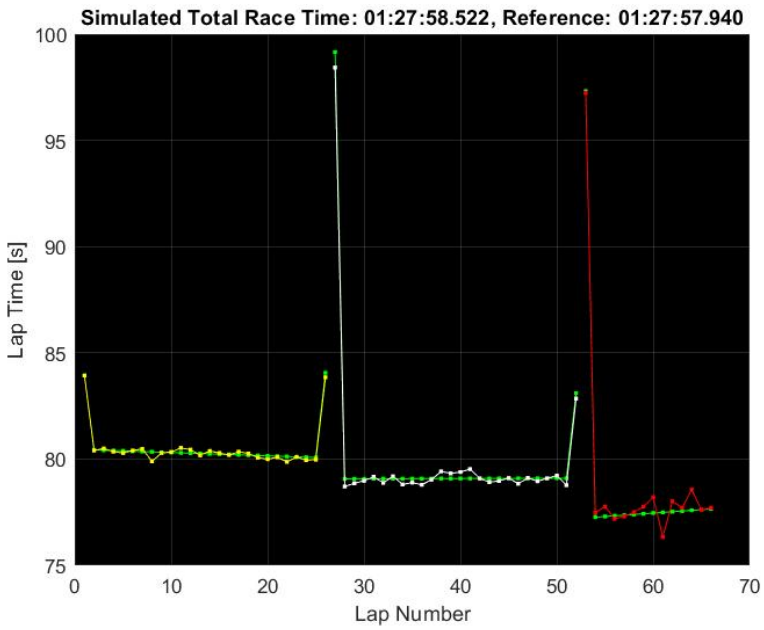

Figure 3. *Simulated lap times (green) overlaid on real lap times (yellow, white, & red indicating compound)*

3.1.3. Pit Duration

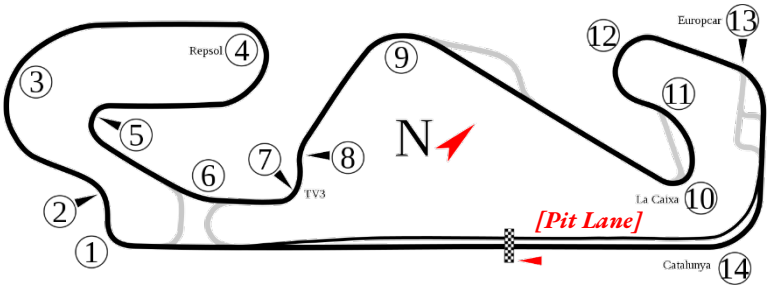

Figure 4. *Spanish Grand Prix track layout*

As the pit lane also runs through the start-finish line, this means that the first half of the pit stop duration (also known as the "in lap") can be attributed to the current lap, while the second half ("out lap") is attributed to the next lap. The in lap and out lap durations are not equal, as the car does not travel at a constant speed throughout the pit lane. For this simulation the durations of the in lap and out lap are considered constant for all pit stops and are determined by analyzing real race data. As shown in figure 5 below, the in laps and out laps can be assumed as the difference between the lap times of the pit laps and the laps directly before and after. Based on the available data, the average in lap was found to be 4.0 seconds while the average out lap is 20.1 seconds.

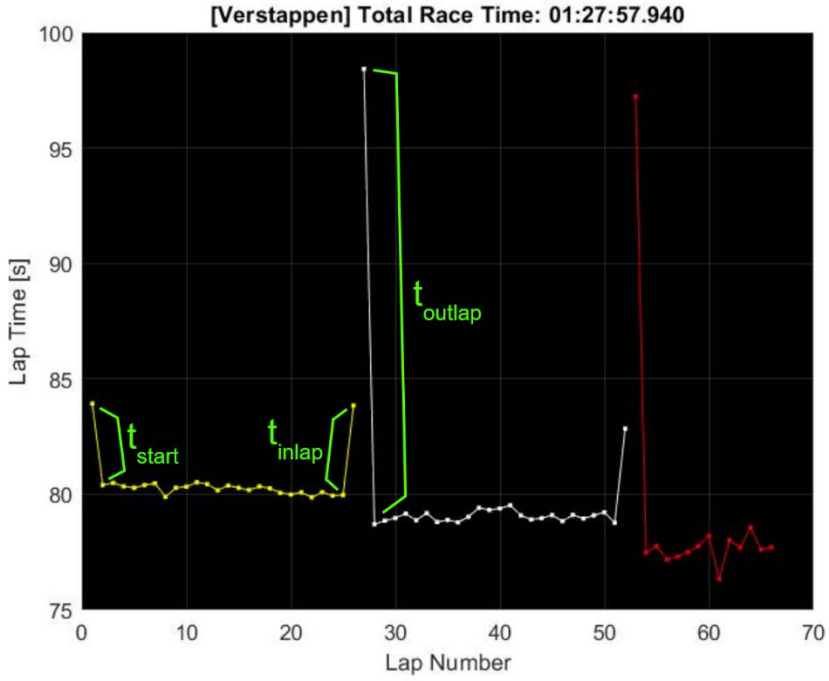

Figure 5. *Time losses due to start, in lap, and out lap indicated on Max Verstappen's race.*

3.1.4. Time Loss at Start

Typically, a Formula One car will start a new lap by crossing the start-finish line at high speed as it is in the middle of a straight section of the track (as shown in figure 4 above). One exception is during the first lap, as the car starts from a stationary position. Because of this, extra time is lost at the start of the first lap as

the car accelerates from rest before reaching the first turn. This time loss is calculated as the difference between the first and second laps, as shown in Figure 5 above. It was found that this duration is typically around 3.5 seconds.

3.1.5. Lap Time Variance

Under real conditions, there will always be some random variation in each lap time which can be attributed to minor driving adjustments from the driver, weather, track conditions, and other effects. To account for this, t_{lapvar} is added to each lap time which represents additional time loss/gain due to these factors. The value of t_{lapvar} is a randomly generated with a mean of 0 and standard deviation of 0.2 seconds. This standard deviation value was chosen to approximately match Max Verstappen's level of consistency during the 2023 Spanish Grand Prix, as shown by Figure 6 below.

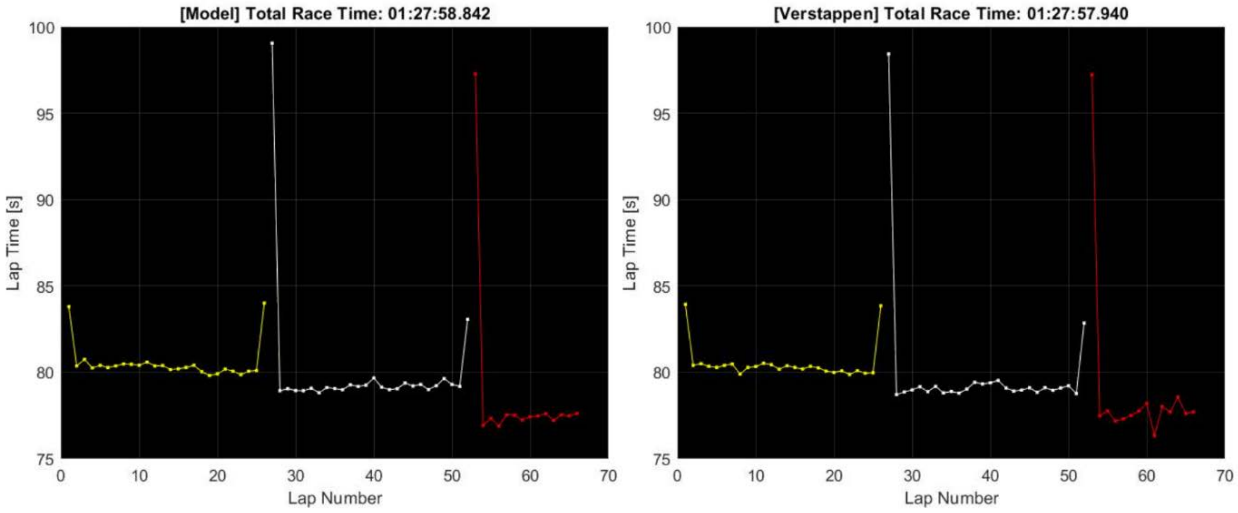

Figure 6. *Comparison between model with variance added (left) and real data (right)*

During the learning process, the lap variance is recalculated at the start of each episode to properly reflect uncertainty in the rewards that are received.

3.1.6. Two-Compound Rule

Formula One regulations state that each car must use at least two different compounds at each race. This rule was introduced for the sake of entertainment, effectively adding extra complexity to pit strategies in the hopes of creating a more eventful race. In reality, any car which does not satisfy this rule by the end of race is automatically disqualified. To simulate this mechanism, a penalty of 100 seconds is added to the final lap if the condition is not met. This value was chosen to introduce a sufficient incentive for the learning algorithm to use a strategy which complies with the two-compound rule.

3.2. Learning Algorithm

For this assignment, the Sarsa algorithm was chosen. The Sarsa algorithm is an on-policy learning algorithm which updates the policy based on actions taken as the agent interacts with the environment. The algorithm is represented by the notation below (Sutton & Barto, 2018). In the original reference, the reward used is written as R_{t+1} while here it is slightly modified to R_t. This is because for this problem the reward is formally assigned to the current time step, as the lap time corresponds to the current lap based on the state at the start of the lap.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_t + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \quad [2]$$

As the problem aims to minimize the total race time, the rewards at each time step are the lap times calculated at each lap multiplied by -1. The discount factor γ is set to 1 by default as the main goal is to minimize the total race time, while individual lap times are less important. The learning rate α is also set to 1 by default as it is desirable for the agent to consider the most recent update.

3.2.1. Terminal State

At the terminal state (i.e., the last racing lap), there is obviously no next state or action. In this case the value of $Q(S_{t+1}, A_{t+1})$ is defined as zero (Sutton & Barto, 2018). Hence at the last lap the algorithm becomes:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_t - Q(S_t, A_t)] \quad [3]$$

3.2.2. Initial Conditions

The Sarsa algorithm requires the initial conditions to be defined before the first-time step update. However, the determination of which tire compound to start with is also part of the problem to be solved. To overcome this, a "pseudo-start" is introduced which adds an additional time step before the first real lap. On this first time step, the agent may take any of the four actions (and hence choose which tire compound will be used on the second time step) but the reward generated at this time step is always zero. Therefore, the total race time will still be the sum of the lap times obtained from 66 laps. Additionally, one the second step (i.e., the first real lap) the tire age and number of unique compounds used are reset to the initial values, while the lap number is allowed to continue accumulating to $66 + 1$ as the absolute lap number does not affect lap time calculation and can be rectified after the learning process. An initial tire compound state must still be given for the first time step, but it no longer reflects the tire compound used at the start of the first real lap as it may be changed by the agent during the pseudo-start. This also means the number states is slightly increased as an additional "lap" is added.

4. Results and Discussion

To first validate the simulation and explore the effects of selecting different hyperparameter values, the simulation was run several times with modified values. An overview of the results is shown in Table 1 below, with reference results shown in the first three rows based on real data and a recreation of Verstappen's strategy using the race model. Runs were initially done without lap time variance in order to easily isolate the effects of each parameter. From runs 1-2 it can be confirmed that the program reaches an identical solution when no parameters are changed and variance is turned off. Runs 3-4 were then executed to validate that the pseudo-start mechanism is working as intended by modifying the initial starting compound of the first time step. As shown, the program results in the same Soft-Medium-Soft tire strategy throughout runs 1-4 regardless of the selection of initial starting compound by the user, as the agent will select the best starting compound for the first real lap. For all remaining runs, the initial compound was set to Soft.

Run No.	Alpha	Gamma	Epsilon	Initial Compound	Variance	Episodes to Convergence (Approx.)	Results		
							Total Race Time	Difference to Baseline	Tire Strategy
Baseline reference (VER)							1:27:57.940	(Baseline)	Med-Hard-Soft
Model with VER strategy, variance off							1:27:58.522	+0.582	Med-Hard-Soft
Model with VER strategy, variance on							1:27:58.842	+0.902	Med-Hard-Soft
1	1	1	0	Soft	Off	74000	1:27:51.866	-6.074	Soft-Med-Soft
2	1	1	0	Soft	Off	74000	1:27:51.866	-6.074	Soft-Med-Soft
3	1	1	0	Med	Off	75000	1:27:51.866	-6.074	Soft-Med-Soft
4	1	1	0	Hard	Off	73500	1:27:51.866	-6.074	Soft-Med-Soft
5	1	0.9	0	Soft	Off	28000	1:27:59.342	+1.402	Soft-Soft-Med
6	0.9	1	0	Soft	Off	103000	1:27:51.866	-6.074	Soft-Med-Soft
7	0.7	1	0	Soft	Off	155000	1:27:51.866	-6.074	Soft-Med-Soft
8	1	1	0.01	Soft	Off	Not Converged			
9	1	0.9	0	Soft	On	35000	1:27:57.547	-0.393	Soft-Soft-Med
10	0.9	1	0	Soft	On	108000	1:27:52.211	-5.729	Med-Soft-Soft
11	1	1	0.01	Soft	On	Not Converged			
12	1	1	0	Soft	On	75000	1:27:50.980	-6.960	Soft-Med-Soft
13	1	1	0	Soft	On	75000	1:27:53.551	-4.389	Med-Med-Soft
14	1	1	0	Soft	On	75000	1:27:50.676	-7.264	Soft-Soft-Med
15	1	1	0	Soft	On	75000	1:27:48.454	-9.486	Soft-Med-Soft
16	1	1	0	Soft	On	75000	1:27:51.820	-6.120	Soft-Med-Soft

Table 1. *Overview of parameters used and results from multiple runs.*

Runs 5-8 were done to investigate the effects of changing hyperparameter values. On run 5, the discount factor γ was reduced to 0.9. As a result, the learning process converged much faster at around 28,000 episodes compared to ~75,000 episodes for undiscounted runs. However, the results that were obtained were noticeably worse than undiscounted runs (+1.4 seconds from the baseline compared to -6.1 seconds). In addition, it was also observed that the algorithm did not settle on the best strategy that was discovered, i.e. there were multiple instances where the program found a better strategy than the final strategy. This result is consistent with the understanding that undiscounted formulations are more appropriate for episodic tasks, while discounted formulations are more appropriate for continuing tasks (Sutton & Barto, 2018). On runs 6-7, the learning rate α was reduced resulting in postponed learning process convergence. Using $\alpha = 0.7$, the

program took more than twice as long to converge (155,000 episodes compared to ~75,000 episodes) while obtaining the same optimal result. Hence, it was concluded that $\alpha = 1$ was the best value to use. Lastly, run 8 was done to explore the usage of an ϵ −greedy method. It was observed that increasing the value of ϵ to 0.01 resulted in failure to converge even after 500,000 episodes, therefore a greedy action selection method was used for subsequent runs. The runs with modified values of α, γ, and ϵ were also repeated with lap time variance on runs 9-11 with similar effects seen towards the convergence characteristics and results.

Finally, five runs were executed with variance on using $\alpha = 1, \gamma = 1$, and $\epsilon = 0$ as shown on runs 12-16 in Table 1 above. As expected, due to the lap time variance the program may settle on different strategies during different runs. However, total race times generated are still generally similar and within the range of -4.4 to -9.5 seconds compared to baseline, which is reasonably consistent with the optimal strategy found without variance of -6.1 seconds compared to baseline. In addition, the optimal strategy found without variance (soft-medium-soft) is still reached on the majority of the runs. Figure 7 below shows the results from runs 1 and 16 to compare results with and without lap time variance.

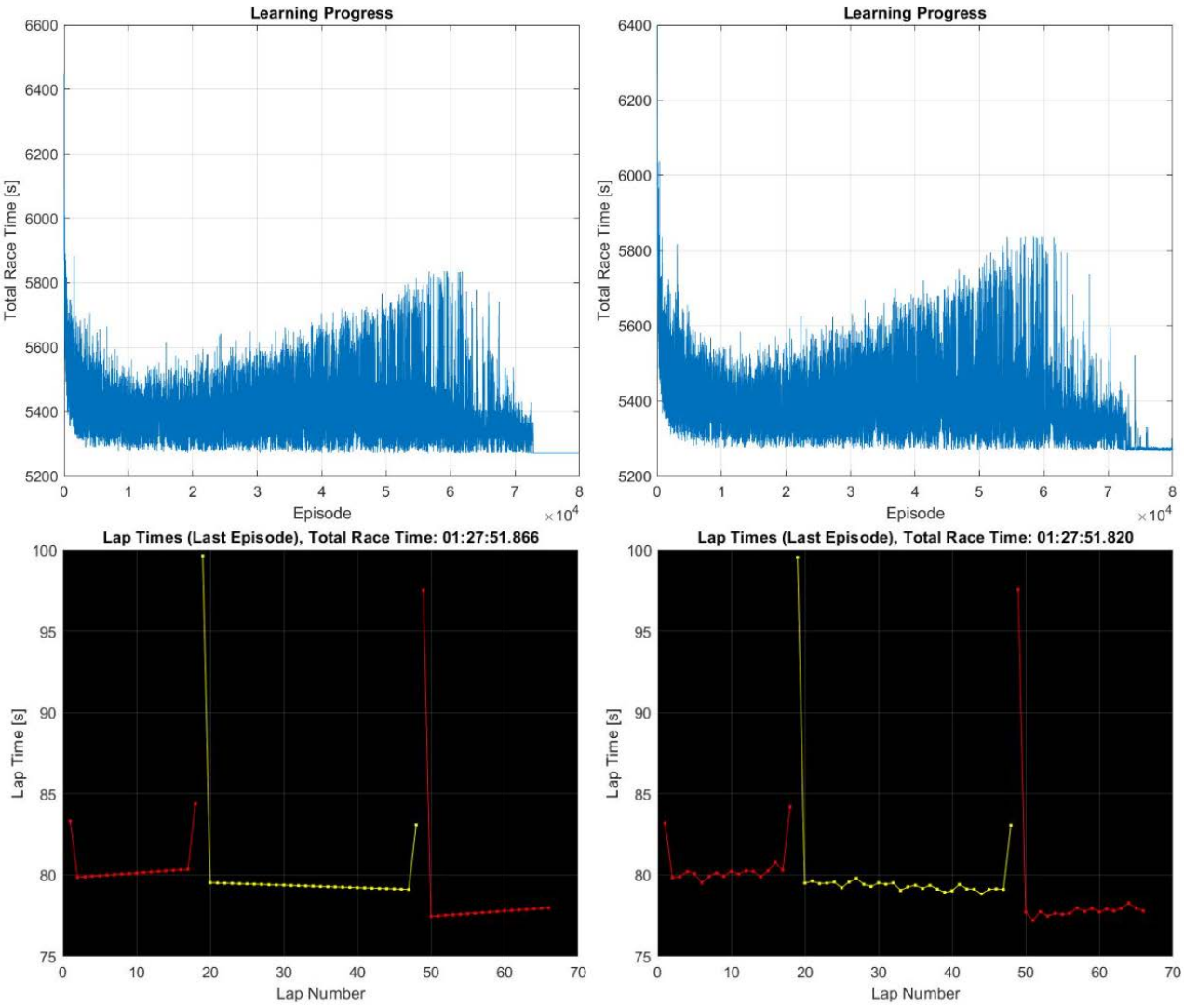

Figure 7. *Learning progress and final strategy (lap times) obtained without lap time variance (run 1, left) and with lap time variance (run 16, right)*

The results generally favour a soft-medium-soft strategy, with the first pit stop at around lap 16-18, and the second pit stop around lap 47-48. This is a realistic strategy which was in fact used by Lewis Hamilton and George Russel during the 2023 Spanish Grand Prix, although they performed their first and second pit stops at around lap 24-25 and 45-50 respectively. Based on this comparison, it was concluded that the program was successful in determining a realistic race strategy based on the race model that was available. The algorithm was able to correctly pick the best tire compounds and time the pit stops in line with the useful lifetime of each compound. Additionally, the model also respected the two-compound rule to ensure that the strategy complies with Formula One regulations.

4.1. Conclusions and Future Work

In this assignment, the Sarsa algorithm was used to determine an optimal race strategy for the Spanish Grand Prix, where the race model was constructed based on data from Max Verstappen's performance. To identify the best hyperparameter values, multiple trial runs were conducted. The program was then successfully able to produce an optimal strategy which was predicted to be roughly 6 seconds faster than Max Verstappen's actual race strategy. In reality, the race winning strategy may have been decided with several other factors which were not considered in this model. Examples of these factors include preferences to extend stints as long as possible to benefit from the possibility of a safety car occurrence (during which pit stops can be performed much more quickly), strategic reactions to the actions of other drivers on the track, and tire availability.

Although the learning algorithm was successful in determining the optimal strategy based on the current model, there is plenty of room for future extensions to this program. As mentioned in Chapter 2.1, there are many factors that are not included in the current model such as safety cars, weather, starting position, and etcetera. Further work can be done to include the effects of these factors to the calculation of lap times. This may also mean that the state and/or action space could be expanded to include more parameters such as presented by (Heilmeier, Graf, Betz, & Lienkamp, 2020) and (Piccinotti, 2020). In addition, the factors that were included could be modelled in more detail, with statistical analysis from larger sets of data to determine more complex regression models for metrics such as tire degradation and pit durations. In order to develop this program into a practical tool for race planning, a more extensive data analysis routine could also be developed to be able to adapt the model for different tracks and use data from practice & qualifying sessions, as the data from the main race session as used in the current model would obviously not be available during pre-race planning.

Appendix

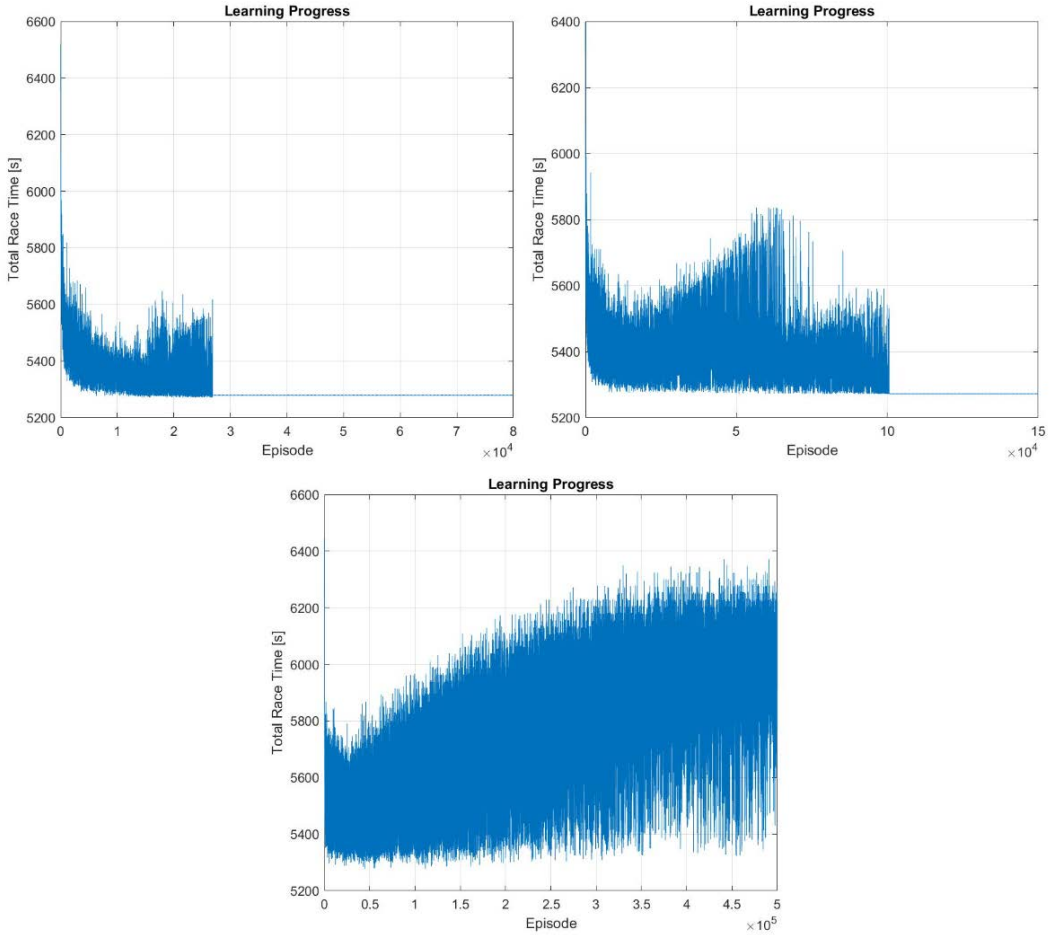

Figure 8. *Learning progress from run 5 (top left), run 6 (top right), and run 8 (bottom).*

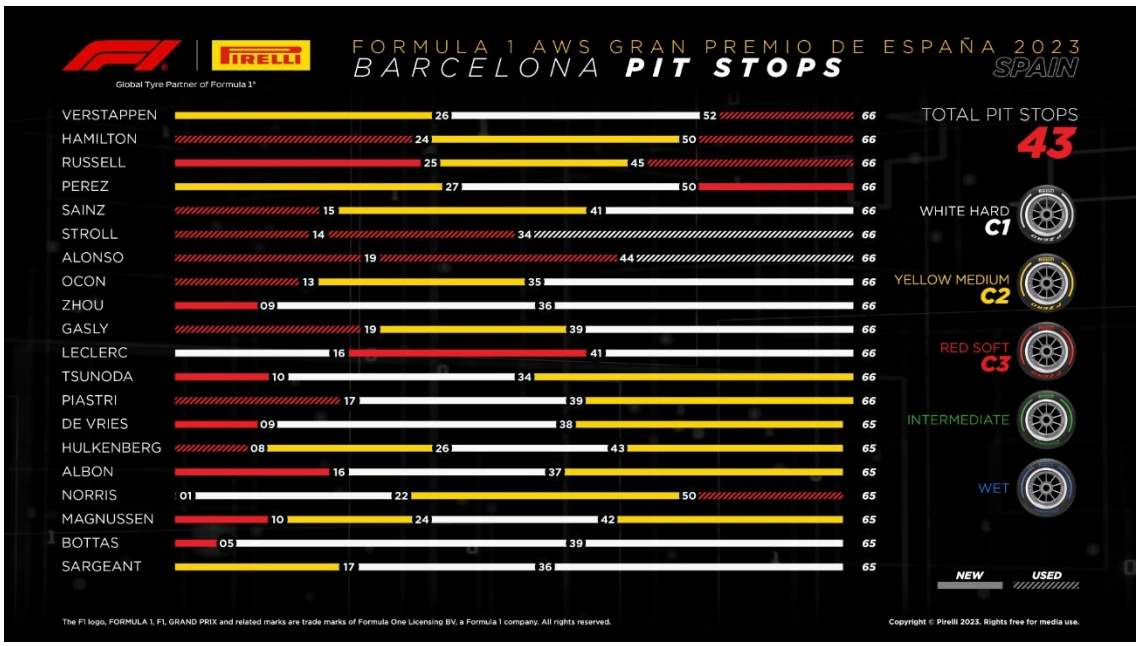

Figure 9. *Real tire strategies used during the 2023 Spanish Grand Prix. Taken from* https://press.pirelli.com/2023-spanish-grand-prix---sunday/

References

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*. doi:10.3390/app10124229

Piccinotti, D. (2020). *Open Loop Planning for Formula 1 Race Strategy Identification.* Politecnico di Milano.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction, second edition.* MIT Press.

Tracing Insights. (2022, August 4). Retrieved from https://tracinginsights.substack.com/p/ferrari-disaster-class-is-hard-compound

Tracing Insights. (2023). *F1 Analysis*. Retrieved from https://tracinginsights-f1-analysis.hf.space/Download_Raw_Data