

A Feasibility Study on Driver Model Based Lap Time Simulation Using Genetic Algorithms

Alvaro Baleato Varela
Technische Universität München, BMW AG

Franz Irlinger
Technische Universität München

ABSTRACT

Lap time simulation has always been a topic of interest in the automotive industry as it summarizes the whole dynamic performance of an automobile in a single value. During the development of road and race cars, to avoid expensive testing and to prove different design solutions, it is useful to simulate the maximum performance of the vehicles. The cars are driven to their limits to exploit their capabilities, where their dynamic behaviour can be highly non-linear. The vehicle models need to replicate these characteristics as precisely as possible. Due to this, the problem of achieving the minimum lap time with a certain car around a race track is a challenging problem to solve. A method to evaluate the minimum lap time is presented, approaching the optimal solution by coupling a driver model, a simulation environment and genetic algorithms to perform the optimization. The algorithm also offers the possibility to add vehicle parameters to be optimized regarding the lap time. The process can be adapted to different vehicle, track and driver models. Not being limited to a single simulation environment, the high flexibility of this method allows it to work with diverse software. The optimization algorithm uses a newly developed penalty function to improve the convergence and accuracy. Real tests on a race track have been performed to assess the correlations between the measurements and the suggested method, revealing its high accuracy.

CITATION: Baleato Varela, A. and Irlinger, F., "A Feasibility Study on Driver Model Based Lap Time Simulation Using Genetic Algorithms," *SAE Int. J. Passeng. Cars - Mech. Syst.* 10(2):2017, doi:10.4271/2017-01-9679.

INTRODUCTION

The minimum lap time around a race track is the main target in the motorsport industry. In order to avoid losing their edge against competitors, a high level of secrecy is adopted pertaining to the detailed information about the LTS (Lap Time Simulation) packages used by teams racing on high-end motorsport series, like the Formula 1 Championship. Most of these teams use codes developed in-house [1]. Commercial LTS packages are also available [2], but this field of race car technology is largely under-reported, as the developer companies intend to attain unique and profitable software. These software codes are commonly developed as a "black-box", where the algorithms are hidden behind a closed interface, only providing access to the inputs and outputs, or directly locked to guarantee a certain level of privacy and competitiveness.

In the present research, lap time simulators will be divided into three different categories: Steady-State (SS) based simulations, Optimal Control (OC) simulations and Driver Model based simulations. All of these simulations can be further sorted into two different categories; either depending on the method in which the performance of the car is evaluated, *i.e.*, the methods that obtain the limits of the vehicle performance in a preview phase of the optimization (e.g. maximal attainable longitudinal and lateral accelerations), or the processes that achieve these boundaries online with the simulation.

Steady-State Based Simulations

The very first lap time simulation approaches started using SS methods. Under this assumption, a manoeuvre that involves linear motion or cornering at constant speed defines the limits of the steady-state performance of the vehicle. To simplify calculations, no transient effects due to the accelerations are taken into account. The car model is considered to be in equilibrium, and the time dependent variables are always set to zero (Steady-State). Most of the SS-based investigations split the track corners into a series of constant radius corners [3], known as Quasi Steady-State simulations (QSS), as represented in Figure 1.

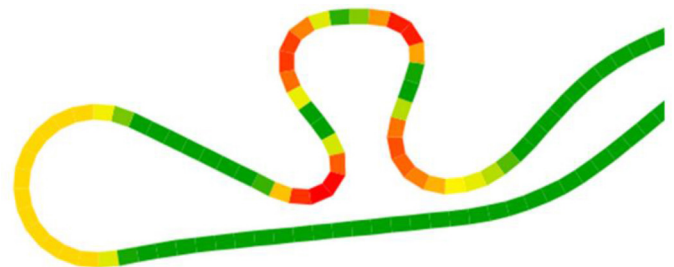


Figure 1. Schematic representation of the different maximum attainable lateral acceleration in a QSS simulation.

These methods usually work with predefined paths or trajectories. A proven benefit of not optimizing the optimal trajectory is the ability to separate the improvements of the vehicle performance from the progresses that could bring us a new optimized racing line [1, 4].

Some of the QSS (quasi steady-state simulations) use a pre-calculated vehicle maximum performance, with most of the methods using performance maps or diagrams called G-G diagrams [1, 5, 6, 7]. The solution proposed in [3, 8] is capable of computing the maximum achievable vehicle performance at the same time as running the lap simulation, avoiding unnecessary calculation or interpolation of unused states of the car, as well as the work presented in [9, 10]. Nevertheless, a high computational effort is needed.

The main disadvantage of the SS-based methods is that they are not capable of simulating phenomena as “turn-in” behaviour or other transient reactions, like kerb riding or the polar inertia associated effects. These methods assume that the vehicle can use the tyre maximum capacity in each moment, and they are not able to evaluate vehicle frequency-dependent responses.

The algorithm developed in [9, 10], named Extended Quasi Steady-State simulation (EQSS) is one of the latest QSS-based approaches. It works with a pre-defined race line, by means of a double-track vehicle model and a QSS simulation. Using a real-time procedure to obtain information on the transient states of the vehicle, the algorithm first calculates the normal QSS performance. After that, the information delivered by the speed profile is taken as an input by the transient state software. This software predicts the said transient effects of the vehicle in an independent form. Then, the results provided by the independent model are taken by the principal simulator and overlaid to the first values. This routine runs using an iterative process, until the overlaid outputs from the transient effects simulator converge to the outputs of the speed profile maker. The method also offers an optimization of different vehicle parameters with the aim of attaining the best lap time, thanks to a Sequential Quadratic Programming (SQP) solving method, calculating the optimal set-up or car configuration according to the elected race track. The accuracy of the algorithm varies when comparing the simulated lap times with the best laps in the race season, with a maximal difference of around 2% in tracks like Sao Paulo, Bahrain and Sebring and values below 0.6% in other circuits like La Sarthe, Spa or Silverstone. The procedure is computationally efficient, taking around 12 minutes to complete one track optimization. The method proved to perform better compared than a normal QSS in all contemplated measures except computational time and robustness.

In other investigations [2] the QSS method has been compared to an OC solution, obtaining a lap time with an error within 3% of the OC solution and reducing the computing time from several hours to a few minutes, displaying its effectiveness. The most significant advantage of the QSS algorithm proposed in [5] is the ability to characterize the stability and controllability of the vehicle.

The simplicity and poor accuracy attained by the simple SS methods are the main reasons for not continuing with their development. No actual research has been conducted in this area. The main discussions relate to the election of QSS or OC processes. Several studies have compared the approaches provided by the SS, QSS and OC methods in the lap time optimization [2, 3, 4, 8].

Optimal Control Simulations

In the area of OC methods, the investigations carried out by [1, 11] are some of the most advanced researches in this area. Most of the OC methods use direct optimization routines to solve the numerical optimization problem [1, 8, 11]. On the other hand, indirect methods are less used [12, 13]. In favor of the direct methods is the greater freedom in defining and including state and control constraints in the optimization problem, which in constrained optimization problems with indirect methods need specially designed algorithms or penalty functions, in order to contain the constraints in the objective function. The indirect methods face difficulties in solving the boundary value problem, and therefore they need to work with oversimplified vehicle models and sections of the entire racetrack. The method developed in [11] used automatic differentiation techniques to solve the non-linear programming problem. Instead of time being the independent variable used to solve the model equations, the distance is used, with a time-to-distance scaling factor applied afterwards. The method optimizes both the race line and the vehicle controls. The election of the number of control points and length of the segments was one of the drawbacks of this method. The simulation times exceed 24 hours for a complete simulation around a race track. Inspired by this work, the OC method developed in [1] improved the computational efforts, reducing the simulation time to around 8 hours.

Driver Model Based Simulations

Most of the driver model-based LTS use QSS or TO methods as a starting solution to determine the speed and trajectory profile [2, 14, 15, 16]. After the determination of the speed profile, the driver model tries to follow the track at the defined speed and trajectory. Using a driver model based simulation simplifies the optimization task, because the complexities and nonlinearities of the car are controlled by the driver model. The research explained in [14] was based on an OC method that optimized the driver controls (steering angle and throttle/brake control) in order to minimize the lap time. Using a normal QSS simulation as start solution, the software needed around 4 hours to optimize a lap around the Jerez racetrack, using a Sequential Quadratic Programming as solver for the optimization problem. An analysis of the convenience of using a QSS or TO as start solution for a driver model method has been researched in [16], concluding the TO method provides the most accurate results. In these “pre-solution” methods, the first approach has been calculated with simple vehicle models and then it has been simulated with the whole model. This creates inaccuracies in the results, due to the different performances provided by the simple and the complete vehicle model. The method presented in [16] first approached the

start solution via optimal control methods. Afterwards, it optimized the target speed history of the vehicle, reaching similar levels of accuracy and robustness as those offered by a pure OC process. However this is also highly computationally expensive, requiring up to 7 hours to reach the minimum time in the complete race track, with differences of up to 5% in lap time depending on the optimization algorithm used. The research clarified in [18] also offered a good answer to the problem of the “presolution” methods. It suggested a driver model capable of calculating the maximum acceleration at each moment during the whole vehicle simulation. To achieve this, the algorithm varied the braking points and used a PI controller to regulate the tyre slip, saturating the driver controls when the tyres were overloaded. It achieved more realistic driver model inputs, not requiring to iterate to attain the optimal solution, and achieving the solution in around 40 minutes. The researches of these authors are some of the latest added to the driver model-based LTS [16, 17, 18].

Referring to the vehicle parameter optimization, only a few publications reviewed this in combination with a LTS simulator. In the field of vehicle parameter optimization solely oriented to the dynamic behavior, much more investigations have been found, the more relevant ones being those in [19, 20, 21, 22]. If the automated and combined optimization of certain vehicle parameters related to a minimum LTS is considered, the research explained in [9, 10] sits amongst the best in this field. Optimizing parameters such as active damping controls or stability systems cannot be executed with SS based methods, due to the non-evaluated transient effects and vehicle frequency-related singularities. The EQSS or OC methods could be a patch to some of these problems.

The driver model based simulations have a high adaptability within current simulation environments, both in-house developed and commercially sold. They are capable of providing accurate results compared to the OC and the most advanced QSS methods. The QSS methods tend to be the faster, being more used in trackside simulations.

Table 1 summarizes the different advantages of the various types of lap time simulators.

Table 1. Comparison of the different lap time simulators performance. ++ Very good, + good, - poor, -- very poor

LTS	QSS based methods	OC methods	Driver model based methods
Computational time	++	--	-
Accuracy	+	++	+
Possibility of parameter optimization	-	++	+
Modularity and transferability	-	--	++

The present work will approach the problem by developing a lap time simulator based on a given driver model for an established vehicle dynamics simulation environment. Therefore, proven vehicle, tyre, driver and road models will be used. The proposed solution couples

all these parts together to produce a method able to minimize the time needed for a certain vehicle to complete a given track. The driver model can accurately follow a given trajectory, using a given speed profile or approximate a non-optimal SS speed profile, the road model defines a certain track and the vehicle model reacts to all inputs given by the driver.

The generation of the ideal trajectory will not be researched in this investigation. Instead, specialized software will be used to determine a near-optimal race line. Removing the trajectory optimization from the main LTS algorithm, can be helpful to ensure that the time improvement comes from a better exploitation of the vehicle capabilities following the given line, and not from the path optimization.

Thus, the information that needs to be optimized and provided to the driver model, will be the target speed profile. With this data, the driver model should precisely follow the pre-defined ideal trajectory, trying to reach the optimum target speed profile, thus allowing us to attain the maximum performance.

In the next sections, the different parts of the suggested method are presented, explaining the characteristics of the vehicle, driver and road models used, and finally the developed optimization routine. The discussed approach compares the results against real track tests, and sensitivity studies and parameter optimization capabilities are considered. To conclude, the advantages and disadvantages of the method are highlighted.

PROPOSED SOLUTION

The present research has, as a starting point, an advanced vehicle dynamics simulation environment. It contains a complex vehicle model, a driver model, and a road model.

Simulation Environment

Vehicle Model

This research is based on a non-linear double track vehicle model [23], as represented in Figure 2. The vehicle is represented as 5 rigid bodies, the vehicle body and the 4 wheels, with 16 degrees of freedom.

The tyre dynamics are calculated using the Magic Formula model [24]. The complete simulation environment is embedded in Matlab/Simulink. The vehicle model uses wheel motion plots to increase the accuracy of the simulations. Thanks to this, the basic characteristics and elastic kinematics of the suspension are modelled, describing the behaviour of the suspension subsystem. Different types of control systems can be attached to the vehicle model due to its modular construction, so the influence of new components and systems like stability controls, active steering, active dampers or anti-lock braking can also be simulated. Completely different types of vehicle architectures, such as cars equipped with automatic or manual gearbox, front, rear or all-wheel-drive transmissions and even hybrid or pure electrical cars can be studied with this software again due to its modularity, giving a unique advantage during the early-phase of vehicle dynamics design.



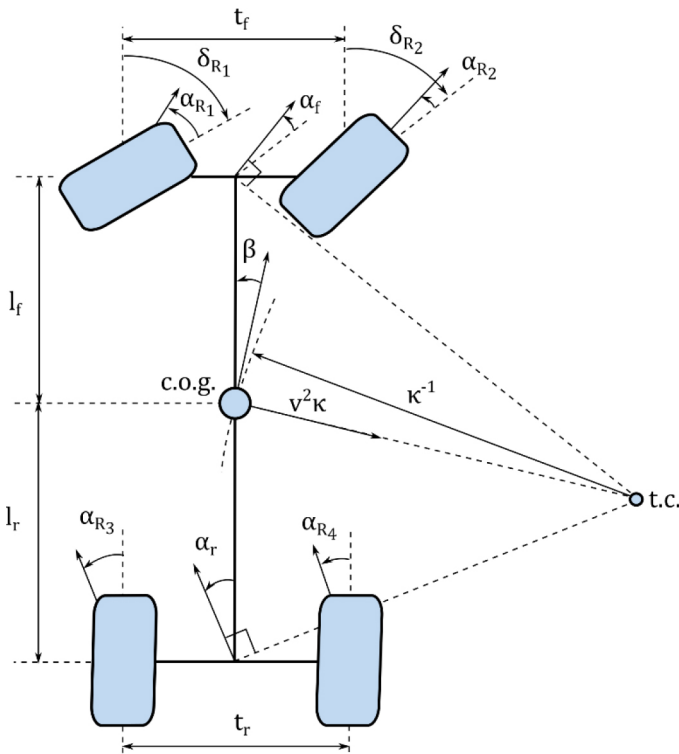


Figure 2. Double track vehicle model.

Driver Model

A privately developed driver model [25] is used in this study. Like most of the driver models, it is based on the preview control theory and on the 2-level control theory [26], which divides both the longitudinal and lateral control into the anticipation and the compensation level, using PID controllers. A schematic representation of the 2-level model is represented in Figure 3.

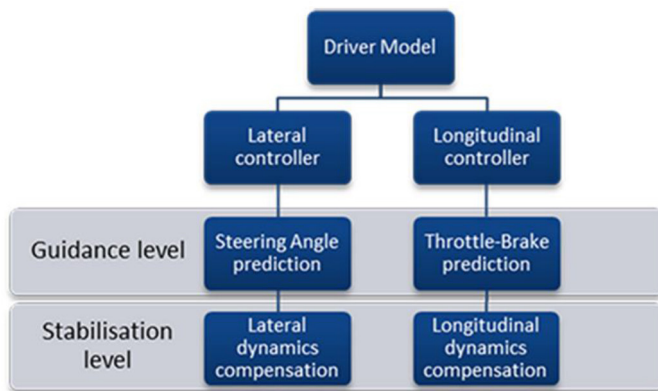


Figure 3. Driver model based on the 2-level control theory.

This driver model tries to follow a given path or trajectory, following a given speed profile. At each point of time, given the vehicle speed, the sideslip angle, the preview time and the preview distance, the model is able to produce the necessary steering angle to compensate the path error at each point of the track. The driver uses a yaw rate controller to

mitigate the un-modelled behaviour and effects, comparing the differences between the vehicle current yaw angle and the yaw angle of the pre-defined trajectory, generating the error which determines the corrections to be applied. With this scheme, the controller is capable of manoeuvring on the highly non-linear limit zone, where the tyre forces are saturated and reaching their boundaries. The steering inputs are generated based on the preview control theory. The preview distance represents the difference between the vehicle and the look-ahead point projected on the reference line, *i. e.*, the point where the vehicle plans to achieve the trajectory. Its value is very important in order to regulate and improve or deteriorate the vehicle behaviour and the path-following accuracy. The driver model uses an automated method to adapt the look-ahead distance, correcting it dynamically relative to the vehicle speed. This driver model needs some expertise to be adjusted to achieve the best performance.

This driver model can follow a given speed trace, or it can provide a first guess of the optimal speed profile. It can create a non-optimal speed profile during the simulation, calculating the maximum achievable speed in the next points, thanks to a pre-defined G-G performance diagram. Therefore, it works as a normal QSS simulator, but the used vehicle model that, at the end, performs the lap is the complete model, thus being affected from all transient effects. Due to the non-optimality of this speed trace, this first approach will only be considered later as start-solution for the main problem.

Road Model

The used road model has been developed under the standards of the OpenCRG software packages [27]. A schematic representation can be seen in Figure 4.

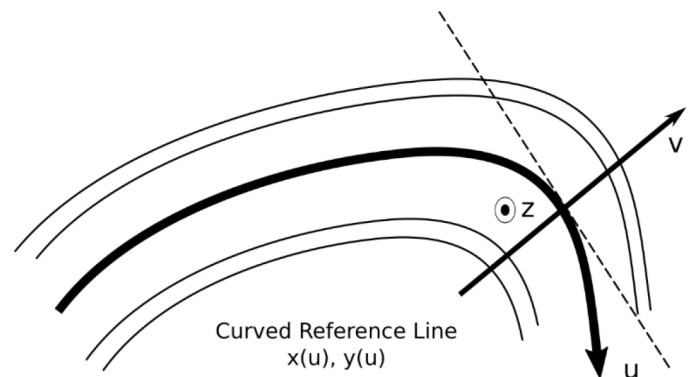


Figure 4. OpenCRG track representation

The reference line data can either be measured with adapted vehicles, or it can be mathematically modelled. The obtained $\{x, y\}$ data needs to be transformed into the coordinate system $\{u, v\}$. The variable u stands for the transverse reference line of the road, while v matches with the orthogonal coordinate of u . The vector v allows the definition of a variable or a constant road width, creating a grid around the reference path in which different characteristics can be defined, such as the elevation or the friction coefficient.