



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta strojní

Ústav automobilů, spalovacích motorů a kolejových vozidel

Lap Time Simulation

Výpočet teoretického času vozidla na závodním okruhu

diplomová práce

Studijní program: Strojní inženýrství
Studijní obor: Dopravní, letadlová a transportní technika

Vedoucí práce: Ing. Jan Baněček Ph.D.
Konzultant práce: Ing. Lubomír Matejko, Porsche Engineering Services s r.o.

Bc. Tomáš Novotný

Praha 2016

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Novotný** Jméno: **Tomáš** Osobní číslo: **382301**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav automobilů, spalovacích motorů a kolejových vozidel**
Studijní program: **Strojní inženýrství**
Studijní obor: **Dopravní, letadlová a transportní technika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Výpočet teoretického času vozidla na závodním okruhu.

Název diplomové práce anglicky:

Lap Time Simulation

Pokyny pro vypracování:

Vytvořte programové prostředky v prostředí Matlab/Simulink pro simulaci jízdy vozidla. Postupujte v následujících etapách:
1. Stručné zmapování postupného rozvoje problematiky závodních simulací se zhodnocením nalezených výpočtových metodik.
2. Na základě zpracované rešerše: a) vytyčení cílů po stránce stanovení šířky výpočtového modelu b) definice okrajových podmínek
3. Teoretický rozbor navrhovaného výpočtového modelu: a) způsob zpracování definované jízdní trajektorie (zadána souřadnicemi XYZ) b) algoritmus výpočtu teoretického rychlostního profilu trati pro zadanou trajektorii
4. Sestavení simulačního modelu vozidla dle požadavků plynoucích z bodů 2, 3 a realizace jízdní simulace
5. Verifikace realizovaného simulačního modelu na základě naměřených dat
6. Návrh a realizace subsystému energetické náročnosti vozidla pro různé jízdní strategie (uvažování modulace max. dostupného výkonu a bodu řazení)
7. Tvorba grafického uživatelského prostředí (Matlab GUI) pro ovládání celkové simulace s možností exportu dat do tabulkového procesoru.

Seznam doporučené literatury:

Jméno a pracoviště vedoucí(ho) diplomové práce:

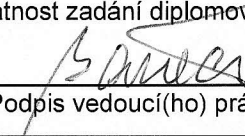
Ing. Jan Baněček Ph.D.

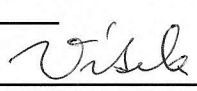
Jméno a pracoviště konzultanta(ky) diplomové práce:

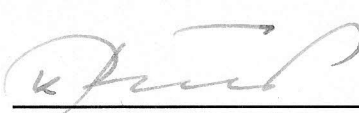
Datum zadání diplomové práce: **18.04.2016**

Termín odevzdání diplomové práce: **30.06.2016**

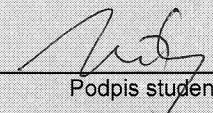
Platnost zadání diplomové práce: _____


Podpis vedoucí(ho) práce


Podpis vedoucí(ho) ústavu/katedry


Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ


22.5.2016

Datum převzetí zadání

Podpis studenta(ky)

Prohlášení


Prohlašuji, že jsem diplomovou práci s názvem

Výpočet teoretického času vozidla na závodním okruhu

vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 7.9.2016


.....
podpis

Abstrakt

Cílem této práce je tvorba nástroje pro výpočet teoretického času průjezdu vozidla definovaným okruhem v programovém prostředí MATLAB/Simulink.

Zadaný problém byl řešen pomocí dvoufázového výpočetního modelu. Na základě stanovené jízdní trajektorie a definovaných maximálních adhezních možností vozu je nejprve proveden počáteční odhad základního dráhového tachogramu. Ten je následně ve druhé fázi korigován pomocí plně specifikovaného modelu podélné dynamiky vozu.

Porovnáním výstupu simulace s naměřenými daty bylo zjištěno, že zásadní vliv na spolehlivou předpověď rychlostního profilu má definice adhezní obálky vozu. V případě správné úvodní identifikace adhezních možností vozu lze dosáhnout poměrně příznivých výsledků. U vyšetřovaných tratí se výsledná odchylka pohybovala v oblasti 0.25-1sec/km.

Hlavním přínosem této práce je tak funkční realizace zpětnovazebné regulační strategie modelu podélné dynamiky vozu na závodním okruhu. Na základě komplexity modelu podélné dynamiky je pak možné provádět další dodatečné analýzy spojené s pohonným řetězcem, případně se nabízí možnost upravovat řídicí strategie pohonných systémů pro konkrétní závodní režim.

Klíčová slova:

Lap time simulator, závodní simulace, ideální závodní trajektorie, dráhový tachogram, analýza trati, g-g diagram, model hnacího ústrojí, BSFC, Matlab GUI, Simulink

Abstract

This work is aimed to develop a MATLAB/Simulink Tool for estimation of lap time on a given racing course.

The proposed solution is based on a two-level calculation strategy. In the first phase, there is created an initial speed profile reflecting the analyzed trajectory and vehicle performance envelope defined by maximal adhesion capabilities. Finally, the initial guess obtained is used as a control input for fully-specified model of longitudinal dynamics. This model already calculates the final speed profile for particular vehicle-track combination.

It has been found out, that the successful prediction of speed profile on a race track depends especially on a precise definition of performance envelope. In case, that those adhesion boundaries are for initial calculation properly identified, the simulated results are not too far from real measurements. The overall time difference between simulation and experimental tests is varying in range of 0.25 to 1 sec/km.

The main contribution of this work represents the successful realization of closed-loop based control strategy for power train model reflecting the race drive mode on a given track. Depending on model range of power train, additional analysis or control strategies regarding to particular racing mode can be then performed.

Keywords:

Lap Time Simulation, Racing Simulation, ideal racing line, velocity profile, racing track analysis, power train model, BSFC, g-g diagram, Matlab GUI, Simulink

| | |
|--|-----------|
| Introduction | 8 |
| 1 Racing fundamentals | 9 |
| 1.1 Cornering maneuver | 10 |
| 1.2 Ideal driving lines | 12 |
| 1.3 Minimal time trajectory | 14 |
| 1.4 Time analysis of hairpin turn | 15 |
| 1.5 Quick parameter study | 18 |
| 1.6 Conclusion – final statements | 20 |
| 2 State of the art | 21 |
| 2.1 Steady-static method | 21 |
| 2.2 Quasi-static & Quasi-transient methods | 23 |
| 2.3 Transient methods | 28 |
| 2.4 Summary | 31 |
| 3 Tool development | 33 |
| 3.1 Tool specification | 33 |
| 3.2 Tool architecture | 38 |
| 3.3 Final statements | 41 |
| 4 Simulation model | 42 |
| 4.1 Simulation model – Stage I | 42 |
| 4.1.1 Track analysis | 43 |
| 4.1.2 Initial velocity profile | 48 |
| 4.1.3 Vehicle performance envelope | 55 |
| 4.2 Simulation model – Stage II | 61 |
| 4.2.1 Power train model | 62 |
| 4.2.2 Consumption analysis | 65 |
| 4.2.3 Race controller strategy..... | 67 |
| 5 Validation | 69 |
| 5.1 Test no.1 | 69 |
| 5.2 Test no.2 | 72 |
| 6 Graphical User Interface (GUI) | 74 |
| 6.1 Loading track file | 75 |
| 6.2 Loading vehicle parameters | 76 |
| 6.3 Lap time simulation | 78 |
| 6.4 Additional settings | 80 |
| 7 Conclusion | 81 |
| 8 Literature | 83 |
| 9 Appendix | 85 |

Nomenclature

| | |
|---------------------------------|---|
| ρ [1/m] | track curvature |
| R [m] | track radius |
| w [m] | track width |
| s [m] | distance |
| t [sec] | time |
| v [m/s] | speed |
| u [m/s] | longitudinal speed |
| a_x [m/s ²] | longitudinal acceleration |
| a_y [m/s ²] | lateral acceleration |
| a_{acc} [m/s ²] | vehicle acceleration |
| a_{brake} [m/s ²] | vehicle deceleration |
| ω [rad/s] | angular speed |
| Ψ [rad] | course angle |
| α_i [rad] | slip angle |
| α [rad] | climbing angle |
| β [rad] | body slip angle |
| δ [rad] | steering angle |
| r [rad/s] | yaw angular speed (yaw rate) |
| J_z [kg.m ²] | yaw moment of inertia |
| J_{red} [kg.m ²] | reduced rotation inertia |
| J_W [kg.m ²] | rotational inertia of wheel |
| J_{Dr} [kg.m ²] | rotational inertia of driveshaft |
| J_E [kg.m ²] | rotational inertia of engine |
| J_C [kg.m ²] | rotational inertia of clutch |
| J_G [kg.m ²] | rotational inertia of gearbox |
| e_i [1] | mass factor |
| m, m_v [kg] | mass vehicle |
| m_{load} [kg] | mass load |
| k_R [1] | coefficient of roll resistance |
| p_{me} [bar] | effective pressure |
| P_e [W] | effective power |
| V_H [l] | engine displacement |
| b_e [g/kWh] | brake specific fuel consumption |
| B_e [g/m] | track fuel consumption |
| n [min-1] | RPMs |
| r_{dyn} [m] | dynamic wheel radius |
| c_a [1] | aero lift coefficient |
| c_W [1] | aero drag coefficient |
| i [1] | gear ratio |
| a [m] | distance from front axle to CG position |
| b [m] | distance from rear axle to CG position |

Introduction

In highly professional motorsport teams, the racing simulations are playing very important role in supporting the decision-making process. Although the theoretical simulation results are always a bit more optimistic than a reality, the simulated minimal racing time is a great performance measure how to identify, in relatively short time period, a potentially successful design proposal or vehicle setup on a particular track. New technical trends, such as the hybridization or full-electric driveline in racing, must take into account also the energy management during the race - that can additionally expand the influence of racing simulations on adjusting the overall control strategies of power train.

The main motivation of this work is a practical realization of racing simulation that will compute the lap time by a given trajectory in Matlab/Simulink-based environment with specified power train model.

The structure of the work should reflect the way of proceeding during the whole development. The first two chapters cover the fundamentals regarding to overall racing strategies and documented methods for lap time calculations. The main development process is then started in the third chapter by specification of general simulation workflow and the proposed system architecture. After detailing the simulation model, the validation phase (fifth chapter) checks the overall functionality and evaluates the reached results against the reality. In the last chapter, the final simulation tool with user-friendly graphical interface is presented.

1 Racing fundamentals

The overall objective in the racing sport is to build a competitive car within the rule specification with the main purpose to win a racing series or championship. This „purpose-vehicle” principle is derived from airspace design, where is known under the name “mission-adequate design”.¹ The best measure of “mission success” is then a direct comparison among the other competitors during a competition on a racing track. To win a race event, not only the car performance is important, but also the driver ability for perfect car control is required and the proper choose of driving strategy, which allows to transverse a given course in minimum time, is needed.

From this point of view, the development of lap time simulation requires beside the physical interpretation of car-track system, also the knowledge about race driving strategies.

Therefore, in this chapter, the general types of driving strategies will be introduced and the first statements for lap time calculations of optimal time trajectory will be made.

“I am an artist. The track is my canvas, and the car is my brush”

– Graham Hill (F1 Race Driver)

The major point of racing strategy is a maximal utilization of the tire performance. The tire influence could be understood as a limiting element between outputted car power and maximal transferrable force on the road surface. A typical illustration of maximal longitudinal and lateral tire capability is the friction circle.² Thereby the race driver relies on the fact that even under intensive breaking or accelerating, some amount of lateral forces for adjusting the travel direction could be transferred simultaneously with longitudinal forces.

¹ J. Baněček, Lecture hand-outs; Basics of Racing Car Design

² Milliken and Milliken, Race Car Vehicle Dynamics (1995)

1.1 Cornering maneuver

The cornering maneuver is basically specified by 3 phases: braking, casting and accelerating.³ At first, the braking phase is initiated. Before starting the maneuver, the vehicle has to be slow-downed intensively.

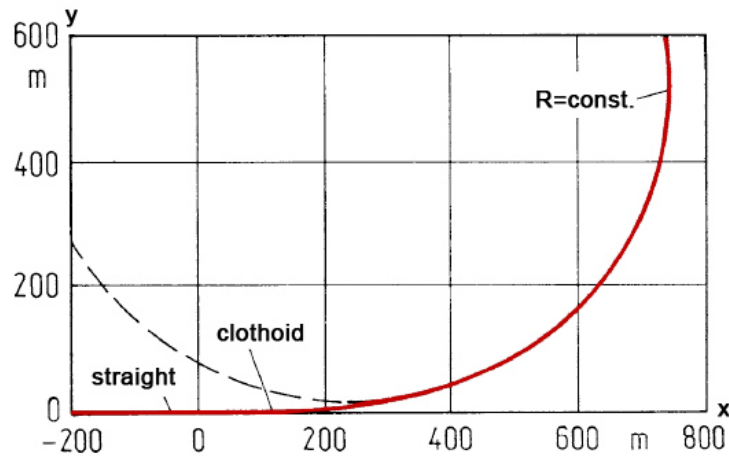


Figure 1.1: Clothoid curve³

$$\text{Clothoid curve: } \frac{1}{\rho(s)} = \frac{s}{A^2}, \quad \rho \dots \text{curvature, } A \dots \text{clothoid parameter} \quad (1.1)$$

1 – Braking: Entering the corner is characterized by continuously increasing path curvature, in *fig. 1.1* represented by clothoid whose curvature ρ grows with the distance s from the origin linearly.⁴ (*eq. 1.1*) It causes the similar gradient of lateral acceleration component – the yaw-velocity and side forces are increasing.⁵ Before the lateral dynamic is fully saturated, the rest of longitudinal component within the friction circle can be utilized for final speed adjusting. The start-breaking point can be then a bit moved subsequently.

2 – Casting: At the end of braking period, the critical corner speed is reached and a short casting interval is launched. The Yaw-velocity and forward speed are constant.⁵

3 – Accelerating: In the final part, the vehicle accelerates out of the turn again. The accelerating phase, as well as the braking phase, is non-stationary and constrained by force circle boundaries. In fact, the acceleration capacity on a good surface is limited not

³ Peter Waldmann, *Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken* (2009), S. 27, fig. 2.14

⁴ M Mitschke, H Wallentowitz, *Dynamik der Kraftfahrzeuge* (2014), S.688

⁵ J. Baněček, *Lecture hand-outs; Basics of Racing Car Design*

so much by entire friction circle, but rather by engine power. In literature⁶, there is even possible to find tables giving the information about a point from which a particular car can again accelerate under full-throttle. While reducing the steering angle towards the end of curve, the lateral tire saturation consequently falls to certain value of side force utilization, which determines the specific “full-throttle” point. For instance, for full-throttle by 150PS car, the critical lateral tire utilization must be under 95% of its maximum. By vehicle with 400PS, the point is at 80%.⁶

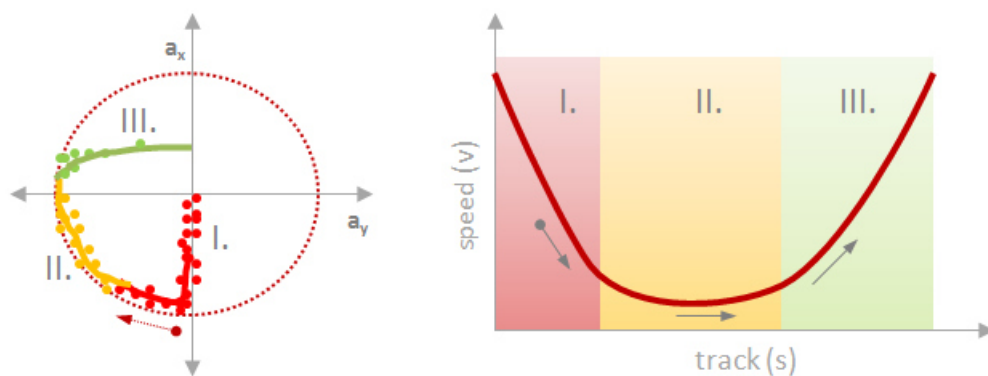


Figure 1.2: Cornering maneuver in friction circle⁷

Looking now at the final time of cornering maneuver, the critical cornering speed is playing the major role there – also significantly influencing the time period needed for braking and accelerating. While the cornering speed is firstly track-dependent (track geometry), the braking or acceleration potential is a measure of car performance. From general racing perspective, the driver should be able to use the maximal car potential permanently. Either giving gas or braking - each casting phase costs a time on the track.

Therefore, the casting period during the cornering should be as minimal as possible – this raises the question of the final shape of time-efficient cornering trajectory. To explore the relations between car-configuration, driving-path and elapsed lap time, a short analytical comparison of different driving styles will be performed in chapter 1.3.

⁶ W. Weber, *Fahrdynamik in Perfektion – Der Weg zum optimalen Fahrwerk-Setup* (2011), S.62

⁷ Peter Waldmann, *Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken* (2009), S. 27, fig. 2.14

1.2 Ideal driving lines

In general there are 3 possible concepts how the corner trajectory can look like.⁸ They are illustrated in figures below on an example of 180deg hairpin turn.

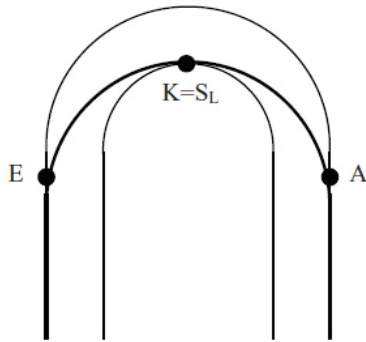


Figure 1.3: Classical ideal line⁹

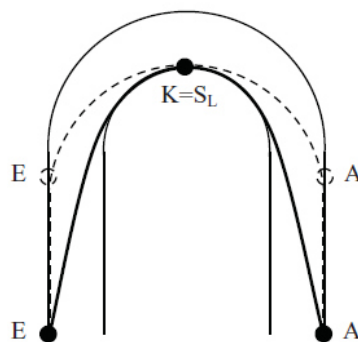


Figure 1.4: Racing/Fighting line⁹

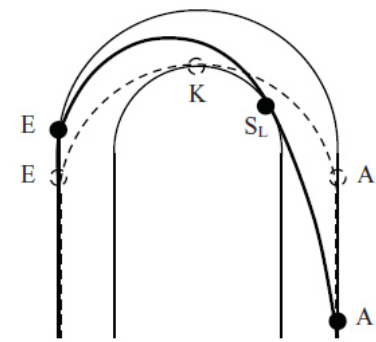


Figure 1.5: Safe line⁹

1.2.1 Classical ideal line

Probably the most effective way, how to pass a corner, is to follow the classical ideal line. (*fig.1.3*) Classical line is using the maximal track width to obtain a larger cornering radius and therefore the higher critical speed $v = \sqrt{a_y/R}$ can be reached. So far, everything seems to be good, but there is a small difficulty - the circuit racing is actually a strongly competitive sport. Using the classical line, the driver allows too much place for the other competitors in behind who could try an overtaking maneuver consequently.

1.2.2 Racing/fighting line

To minimize the probability of being overtaken, the leading driver has to adjust his driving line. For so-called racing or fighting trajectory, the smaller radiuses are typical. According to this strategy, the driver tries to avoid a situation where he could be eventually slowed-down by a car inside the corner. (*fig. 1.4*) In spite of the lower cornering speed, the late braking point is a clear advantage for initiation of an overtaking maneuver.

⁸ Cf. B. Spiegel: Die obere Hälfte des Motorrades-Über die Einheit von Fahrer und Maschine, (1999) according to: Peter Waldmann, Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken (2009), S.11

⁹ Peter Waldmann, Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken (2009), S. 27, fig. 2.14

After taking also the earlier acceleration point at the end of corner into account, there is a certain possibility that such trajectory could be even faster than the classical one. In chapter 1.5 will be this issue more detailed. The overtaking situation is graphically summarized in *fig. 1.6*.

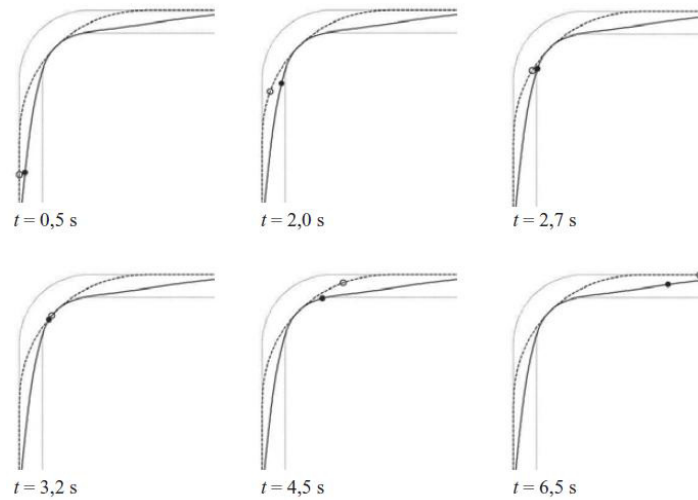


Figure 1.6: Classical vs. Racing Line (overtaking maneuver)¹⁰

1.2.3 Safe line

The last trajectory type on the list is so-called safe-line. In matter of fact, it's a not usual racing strategy. According to the motto "Slow-in, Fast-out!", the safe line is recommend as the ideal line for civil driving on public roads. Many drivers make a common fault that they are turning into the corner too early and driving direction must be then during the maneuver corrected accordingly. In contrast, the late start-point of turning gives the driver more information about curve itself.¹⁰ A disadvantage of smaller radius is compensated by a better position when leaving the corner with lower demands on lateral dynamic.

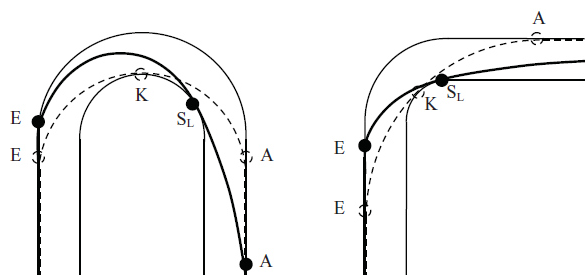


Figure 1.7: Safe line¹⁰

¹⁰ Peter Waldmann, Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken (2009)

1.3 Minimal time trajectory

Unfortunately there is no general rule for an ideal time trajectory. More or less it's a compromise between classical and racing path. In this section it will be outlined an example taken from dissertation work of Mr. Waldman¹¹ where he was designing an autonomy racing-car controller.

The typical cornering issue is illustrated on *fig.1.8*. It's a conflict between speed and trip distance. In summary, there are two options – either driving faster on the larger radius or slower inside (but on the shorter track).

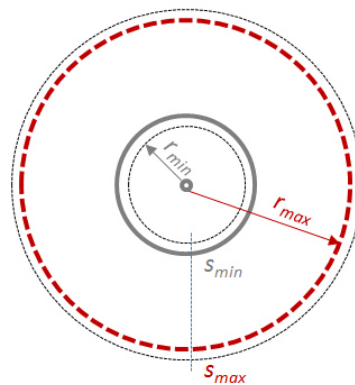


Figure 1.8: Minimal time trajectory issue¹¹

When analyzing only the arc section separately, there is a clear proportionality (*eq. 1.3*) between radius and time. The inner side is always faster (*in case $a_y = const$*).

$$s_i = 2\pi r_i \quad (1.1)$$

$$v_i = \sqrt{a_y \cdot r_i} \quad (1.2)$$

$$t_i = \frac{s_i}{v_i} \rightarrow \frac{2\pi r_i}{\sqrt{a_y \cdot r_i}} = \frac{2\pi}{\sqrt{a_y}} \sqrt{r_i} \quad (1.3)$$

As already mentioned, the whole maneuver is not only about steady circle drive, but also the breaking and accelerating phases have to be taken into account. For better illustration of all relations that are influencing the driving time, a simple hairpin (180°) corner will be introduced. (*fig.1.8*)

¹¹ Peter Waldmann, Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken (2009)

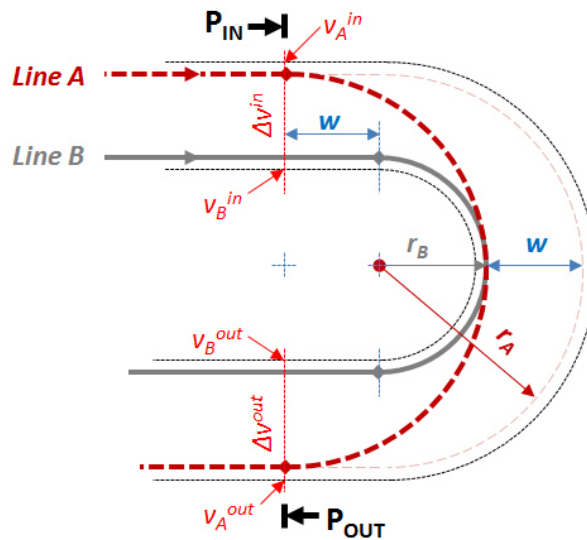


Figure 1.8: Hairpin Turn with two Driving Lines, A = Classical Line, B = Racing Line ¹²

1.4 Time analysis of hairpin turn

In this example, it will be assumed that the whole trajectory consists only of straight lines and regular corners – the longitudinal and lateral acceleration components will be now dealt separately. In contrast to real trajectory¹³, it's a quite significant simplification but the main point is to see, already on a simple calculation model, the important relations between chosen lines.

In *fig. 1.8* are marked two different paths – one as a classical line (*A*) and the second one (*B*) represents the racing line with minimal corner radius. From text above¹⁴ is already known that the inside path can spare a bit time in mid-corner phase. In order to make a relevant statement about whole cornering maneuver, the same track section for both trajectories (*A and B*) has to be compared. For this case, the referent points P_{IN} and P_{OUT} have been introduced. (See *fig. 1.8*)

1.4.1 Line A – Classical line

The time for classical line (*A*) is determined by critical cornering speed. (*eq. 1.5*) Because of constant-radius assumption, the cornering speed (*eq. 1.4*) is constant for whole maneuver distance – the calculation only has the steady component. Points P_{IN} and P_{OUT} are geometrically representing the start and the end of arc.

¹² Peter Waldmann, Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken (2009)

¹³ See Chapter 1.1

¹⁴ See Chapter 1.3

$$v_A^{in} = \sqrt{a_y \cdot r_A} \quad (1.4)$$

$$t_A = \frac{\pi}{\sqrt{a_y}} \sqrt{r_A} \quad (1.5)$$

1.4.2 Line B – Racing line

In case of racing line, the situation is a bit complicated. To have a fair comparison of those two racing strategies, the same reference points P_{IN} and P_{OUT} have to be considered. (see *fig. 1.8*) The time component of steady circle drive (*mid-corner phase*) is calculated in the same way as in the case above. (see *eq. 1.5*) The only difference is that the straight segments must be calculated additionally in order to reach given reference points. (*eq. 1.14*)

Let's go through the sequence of all important points on the track and find out the differences to the classical line. Let's start in reference point P_{IN} , although indeed the real start point of cornering phase is shifted backward through the required distance as a length w . (see *fig.1.8*). It also explains the velocity difference at P_{IN} . (*eq. 1.7*)

$$v_B^{in} = \sqrt{a_y r_B + 2w a_{brake}}, \quad (a_{brake} > 0) \quad (1.6)$$

$$\Delta v^{in} = v_A^{in} - v_B^{in} = \sqrt{a_y r_A} - \sqrt{a_y r_B + 2w a_{brake}} \quad (1.7)$$

$$w = r_A - r_B \quad (1.8)$$

$$\Delta v_{(r_A, w, a_{brake})}^{in} = \sqrt{a_y r_A} - \sqrt{a_y (r_A - w) + 2w a_{brake}} \quad (1.9)$$

As a next step, the cornering speed (mid-phase) will be calculated.

$$v_B^{mid} = \sqrt{a_y \cdot r_B} \quad (1.10)$$

And finally there is the end-phase where the car is accelerating again. In order to be able compare the time advantage of particular racing strategy, the same reference point P_{OUT} must be considered. Very important fact – in this point, both cars have to full-accelerate again, otherwise would be the comparison not valid.

$$v_B^{out} = \sqrt{a_y r_B + 2w a_{acc}} \quad (1.11)$$

$$\Delta v_{(r_A, w, a_{acc})}^{out} = \sqrt{a_y r_A} - \sqrt{a_y (r_A - w) + 2w a_{acc}} \quad \text{with (1.8)} \quad (1.12)$$

For race-line (B), the full maneuvering time consist of 3 particular components. (eq. 1.14)
Beside the time of steady corner drive (see eq. 1.5), the braking and acceleration parts must be taken into account. Both time components are derived from track functions for steady acceleration. – Example in (eq. 1.13)

$$s(t) = v \cdot t + \frac{1}{2} a_x t^2 \quad \rightarrow \quad t(s) = \frac{1}{a_x} (-v \pm \sqrt{v^2 + 2a_x s}) \quad (1.13)$$

$$(1.14)$$

$$t_B = \underbrace{\frac{1}{a_{brake}} \left(-v_B^{in} + \sqrt{(v_B^{in})^2 + 2w a_{brake}} \right)}_{\text{Braking}} + \underbrace{\frac{\pi}{\sqrt{a_y}} \sqrt{r_B}}_{\text{Cornering}} + \underbrace{\frac{1}{a_{acc}} \left(-v_B^{out} + \sqrt{(v_B^{out})^2 + 2w a_{acc}} \right)}_{\text{Accelerating}}$$

1.4.3 Time evaluation of ideal- and racing- line

In velocity diagram on fig. 1.9, the whole situation is illustrated graphically again. The crucial part is the *Phase II* with different positions of start-braking points for classical line (A) and racing line (B).

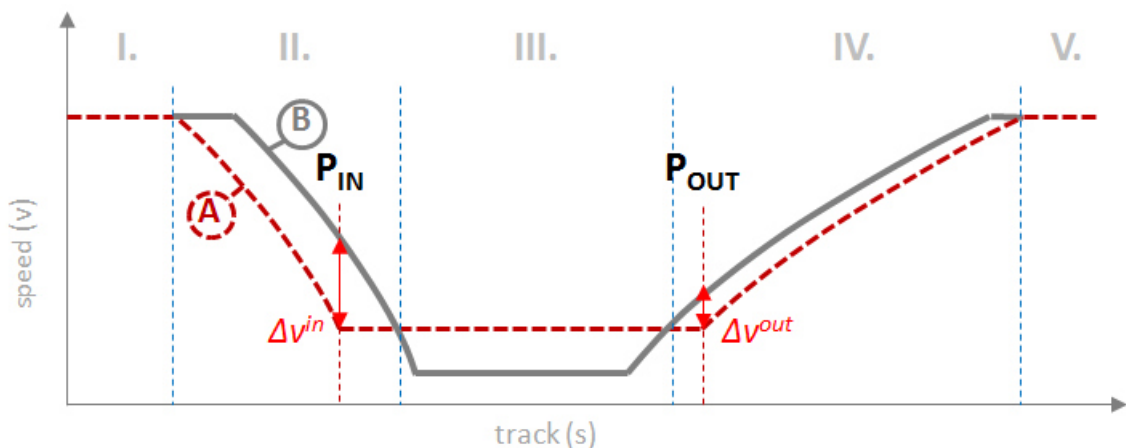


Figure 1.9: Speed diagram for Line A and Line B ¹⁵

¹⁵ Peter Waldmann, Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken (2009)

Although the classical line (A) should be in general the most time-efficient one, the driver on this line must start to brake earlier. This induces a speed difference Δv^{in} against the race line (B) and generates so a temporary time disadvantage for classical line (A), which could be used by an opponent on race-line (B) for an overtaking maneuver. The lost-time in slow mid-corner section (*line B, phase III*) can be partially reduced due to earlier acceleration out of the corner. (*phase IV*)

The overall time difference of both trajectories between points P_{IN} and P_{OUT} is than represented by subtraction (*eq. 1.5*) and (*eq. 1.14*) in (*eq. 1.15*)

$$\Delta t_{(r_A, w, a_{acc}, a_{brake}, a_y)} = t_A - t_B \quad (1.15)$$

$$\Delta t_{(r_A, w, a_{acc}, a_{brake}, a_y)} = \frac{\pi}{\sqrt{a_y}} \sqrt{r_A} - \frac{1}{a_{brake}} \left(-v_B^{in} + \sqrt{(v_B^{in})^2 + 2w a_{brake}} \right) -$$

$$- \frac{\pi}{\sqrt{a_y}} \sqrt{r_A - w} - \frac{1}{a_{acc}} \left(-v_B^{out} + \sqrt{(v_B^{out})^2 + 2w a_{acc}} \right)$$

1.5 Quick parameter study

On the basis of equations above (*eq. 1.9, 1.12, 1.15*), the functional relations between maneuver time and following parameters have been found:

- **Track geometry:** max. radius (r_{max}) and track width (w)
- **Friction coefficient:** here related to max. lateral acceleration (a_y)
- **Vehicle performance:** related to max. longitudinal accelerations (a_{acc}, a_{brake})

Listed analytic equations are also proof of hypothesis¹⁶, that there is no general minimal-time trajectory which could be applied for diverse vehicles. The number of variables makes from this issue quite challenging optimization problem. In order to get better inside view, how the individual parameters from *eq. 1.12* and *1.15* will interact with maneuver time, a short parameter study was performed. The time difference (*eq. 1.16*) and end-speed difference (*eq. 1.17*) were selected as a direct measure influencing the choice of driving line.

¹⁶ See chapter 1.3

Equations below show a simple relation between classical and racing line. Time value with positive sign should be interpreted as a time advantage for racing line (B). Positive sign by velocity difference than means advantage of higher end-speed for classical ideal line (A).

$$\Delta t = t_A - t_B \quad (1.16)$$

$$\Delta v^{\text{out}} = v_A - v_B \quad (1.17)$$

1.5.1 Engine power vs. Trajectory shape

In *fig 1.10*, the relation between driving line (r_i) and car performance (a_{acc}) has been examined. The other parameters representing friction coefficient (a_y, a_{brms}) and track width (w) stayed fixed as constants.

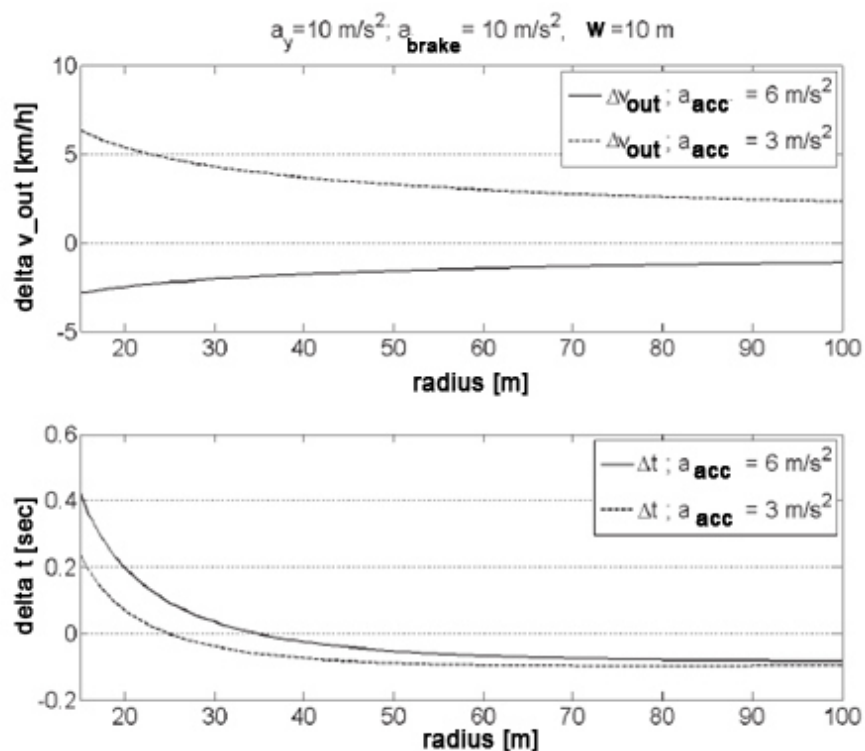


Figure 1.10: Parameter Study¹⁷

¹⁷ Peter Waldmann, Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken (2009)

The information about spared time in corner section (eq. 1.16) has to be judged with respect to reached speed (eq. 1.17) at the end of section. With other words: there would be no point in saving some tenths of seconds in corner and finally lose them because of low speed on the beginning of straight sector - except the case, that instead of a straight sector, next corner is following.

1.5.2. Findings of parameter study

The conclusion coming from graph 2 confirms the trend that going through slow corners on shorter racing line spares the section time. The intersection of the graph curve and zero time level clarifies then for this way of interpretation the term „slow corner“. Faster corners are to pass on the classical line consequentially.

Graph 1 is indicating that speed reached on the end of curve section depends on the car performance significantly. That's why the pure racing line strategy could be profitable only for more powerful cars. Weaker cars are tending to pass corners on the widest possible radius same as lower-skilled drivers.

1.6 Conclusion: Final statements

After analyzing of introduced hairpin turn, the following statement can be written:

Statement: *„Despite the highest curve speed by the classical ideal line, this line cannot be generally considered as the absolutely fastest one for every single situation. The optimal time trajectory depends on track geometry, friction circle, car performance and of course on driver's skills.“¹⁸*

The same conclusion, that driving trajectory depends on vehicle specification, did T. Gustafson in his work¹⁹ His lap time simulator is directly based on optimization task²⁰ and therefore is able to suggest also the optimal trajectory. Further description of lap-simulator strategies is covered in the next chapter.

¹⁸ Peter Waldmann, Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken (2009)

¹⁹ Thomas Gustafsson, Computing the ideal racing line using optimal control (2008), S.55

²⁰ See Chapter 1.5

2 State of the art

Probably the first experience with Lap Time Simulation has been performed by Mercedes-Benz Racing Department in 30's. The very first estimation of track velocity profile was based on simplified steady-state calculation investigating the vehicle speed on racing track only composed of straight sections and regular corners with constant radius.²¹

Observing the current situation, nowadays, there is an enormous potential of computation capacities, and thus the simple steady-state task has become a sophisticated non-linear problem trying to cover the large spectrum of car behavior on the track. The objective of new racing simulation methods is additionally focused on parameter optimization in order to help by looking for ideal vehicle configuration intended for particular racing track.

This chapter introduces a different simulation approaches regarding to implemented calculation philosophy. In first section, the steady-state method will be briefly outlined, which helps to see basic relations of vehicle-track system. On basis of methods presented in next sections, it will be tried to extend the basic calculation framework over specific phenomena in order to found out a relevant range of calculation model.

2.1 Steady-static method

As mentioned above, there is the track described as a set of straights and regular corners. In the first step (*fig.1.2 – step 1*), the critical speeds for every single turn on a course have to be found out.

$$v_{R(i)} = \sqrt{\frac{a_{ymax}}{R(i)}} \quad (1.1)$$

²¹ Cf. Moss and Pomeroy, *Design and Behaviour of the Racing Car* (1963) according to Casanova, "On Minimum Time Vehicle Manoeuvring: The Theoretical Optimal Lap" (2000), S. 1-2.

Where in Equation 1.1, the critical cornering speed $v_{R(i)}$ is estimated by maximal lateral capacity $a_{y,max}$ respecting the adhesion capacity between tires and track. Influence of aero down forces is in this case neglected. Because of assumption of the specific track splitting (*fig. 1.1*), the lateral and longitudinal behavior of car is treated separately. (i.e. the vehicle brakes and then turns in). Thus, only the lateral acceleration performance of the vehicle is taken into account during cornering.²²

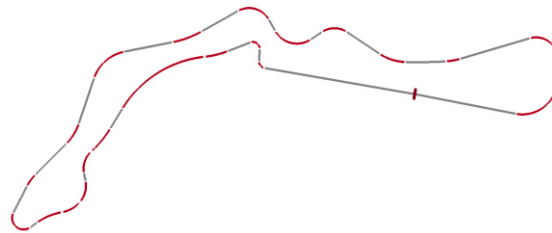


Figure 2.1: Splitting the Race Track (Autodrom Most)

Therefore the main calculation task in second step (*fig.1.2 – step 2*) contains only the longitudinal motion on straight sections. In order to reach final form of speed profile for sections in between of two corners, basically two driving modes must be considered – *accelerating and braking phase*. Both states are graphically represented in *fig. 2.2* in form of accelerating and braking curve. Final form and gradient of both characteristics is given by vehicle performance and tire capacity.

Finally, the whole calculation process of second step is sketched graphically on *fig. 1.2* by the help of performing the forward (acceleration) and backward (braking) speed examination towards already known critical cornering speeds defining the boundary conditions. The intersection of these two curves stated above initiates the braking phase (red blocks).

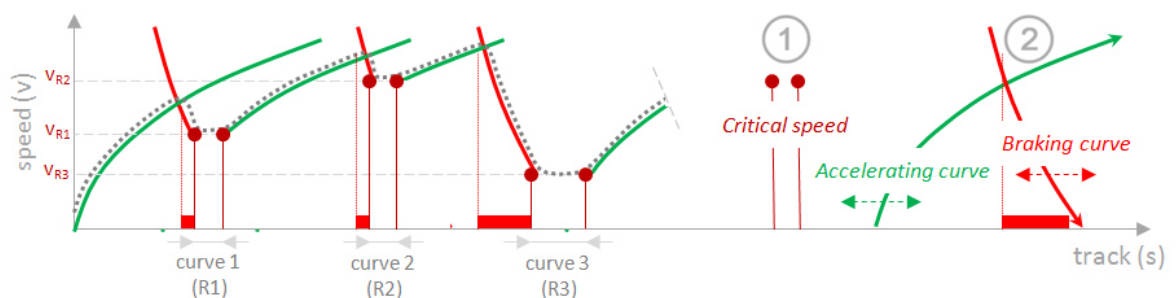


Figure 2.2: Construction of speed diagram²³

²² B. Siegler, A. Deaking, D. Crolla, Lap Time Simulation: Comparison of Steady State, Quasi-Static and Transient Racing Car Cornering Strategies (2000), S. 3

²³ J. Baněček, Lecture hand-outs; Basics of Racing Car Design

Both calculation steps are constrained by given car-performance measures. Firstly, there is a maximal lateral and longitudinal capability respecting the friction coefficient of tires and, secondly, the acceleration limits restricted by a power-train specification are also taken into account.²⁴ There is considered the whole car model as a point mass, which makes the task suitable for calculations by hand without any higher effort.²⁵

2.2 Quasi-static & Quasi-transient methods

On basis of comparing the steady-state strategy against real-cornering concept, it is obvious, that the static working calculations from chapter 2.1 have only a very low ability to give accurate results for racing time estimation. Main reason for the bad convergence to real measurement on the track is hidden in the enormous simplification of track description which leads to unrealistic cornering maneuver ($R=const.$). In order to receive more satisfactory results, the regular arc segment must be replaced against a better track description corresponding to the real cornering trajectory described in previous chapter²⁶ and basically characterized by 3 phases – *braking, casting and acceleration*.²⁷ Therefore the trajectory must be expressed with a sufficient number of points, path-segments, reflecting the real form of cornering path.

2.2.1 Iterative quasi-static method:

Considering the race driving strategy²⁷, the main objective of driving on the limit is to use during cornering the whole friction circle and maximizing so the utilization of vehicle/tire performance. A car that accelerates, brakes and corners in a smooth fashion can be modeled approximately by joining together a series of static equilibrium or steady-state maneuvers. The condition that must be met, the changes in vehicle states between path segments occur slowly. Every path segment is then represented by a certain path radius with an adequate step size.²⁸

²⁴ Timo Völkl, *Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen* (2013), S. 9.

²⁵ See the chapter 1.3

²⁶ See the chapter 1.4

²⁷ Peter Waldmann, *Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken* (2009), S. 27, fig. 2.14

²⁸ D.L. Brayshaw and M.F. Harrison, *A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location* (2005), S.729

Finally, the maximal performance peak for every track step is found iteratively (Newton Raphson technique) at the same time during the velocity profile calculation. According to the physical model complexity, the ratio between result-accuracy and calculation-efficiency is determined; whereas this method provides already quite good correlation between simulated results and real vehicle testing. There is mostly used single- or two-track model with non-linear tire description. In general, the lateral tire force is found on the basis of friction circle approach using a combined Pacejka Magic Tire Formula. The remaining tire force then helps to find the longitudinal acceleration of vehicle.²⁹ Lot's of commercial offered SW is based on this method.³⁰

Siegler und Crolla³¹, using this calculation approach, have tried to establish a „rank list“ of the most significant parameters influencing the minimal lap time. The three most important parameters according to [31] are:

- **Friction coefficient of tires**
- **Engine power**
- **Aerodynamic drag force**

Followed by: *Height of COG, wheel gauge, car-weight, aerodynamic lift force, roll-stiffness distribution and weight distribution*

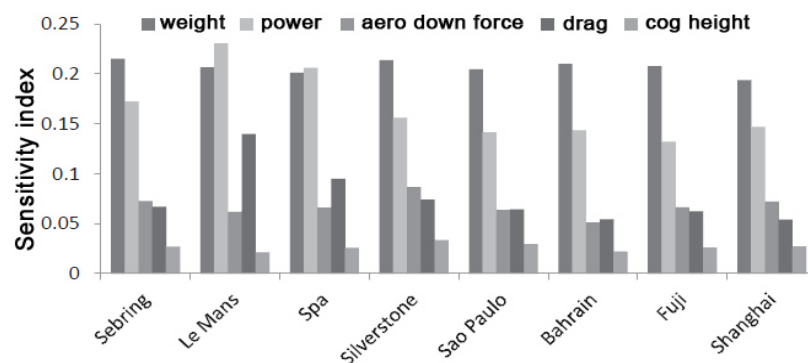


Figure 2.3: Lap time sensitivity for all WEC race track by T. Völkl [4]

The findings achieved in dissertation work by Timo Völkl³⁰ extend this raking by scaling the parameter importance for particular racing track (*fig.1.3*). Basically, he reached in his work the same conclusions with a clear statement, that the major role for a reliable time

²⁹ B. Siegler, A. Deaking, D. Crolla, Lap Time Simulation: Comparison of Steady State, Quasi-Static and Transient Racing Car Cornering Strategies (2000), S. 3

³⁰ Timo Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013), S. 17, 166.

³¹ Siegler und Crolla, Lap Time Simulation for Racing Car Design (2002).

prediction has a valid tire model. Steady-state approximation of tire behavior by Pacejka has been in [31] replaced by the semi-empirical model TMeasy³² reflecting the transient and thermal characteristics more accurately (*see fig. 1.4*). Transient effects, in general, are then objective of chapter 2.3.

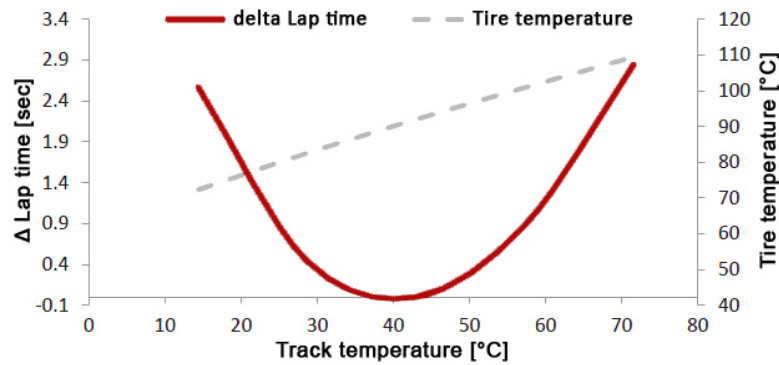


Figure 2.4: Influence on Racing Time by Tire/Track Temperature [4]

2.2.2 Quasi-static method with pre-calculated g-g diagram

The g-g diagram (*fig. 1.5*) illustrates a performance envelope by supplying all information about the combined acceleration limit of the vehicle in the longitudinal and lateral directions. Using the g-g diagram and a driven line, an optimal speed profile for the minimum lap time can be directly created.³³ For calculation of particular g-g diagram, the non-linear tire characteristics, linear suspension model with known spring rates, and

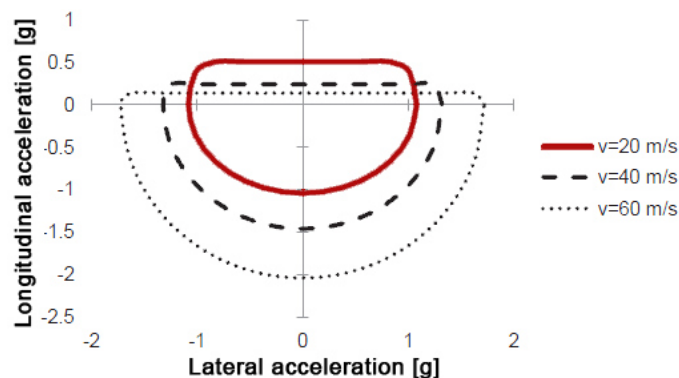


Figure 2.5: The g-g diagram ³¹

³² Cf. Rill, Simulation von Kraftfahrzeugen (2007) according to Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013), S. 52

³¹ Timo Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013), S. 93, 15.

³³ Chris Patton, Development of Vehicle Dynamics Tool for Motorsport (2013), S. 62

aerodynamic lift effects have to be determined.³⁴ The influence of aero down force makes the performance envelope velocity-dependent, just as it can be seen on *fig. 1.5*.

The calculation of speed profile is indeed very similar to concept which was introduced in steady-state methodic on *fig. 1.2*. The program identifies in the trajectory curvature data the corner sections with peaks representing the apexes of corners and between a pair of them calculates the maximum acceleration possible from apex i up to apex $i+1$ (*fig. 1.6*). This procedure is carried out for all apex pairs encountered over the entire lap. Then in second step, the same calculation is performed for maximum braking deceleration in opposite direction, backwards from $i+1$ to i . Crossover point is than switching point from the acceleration data to the braking data.³⁵

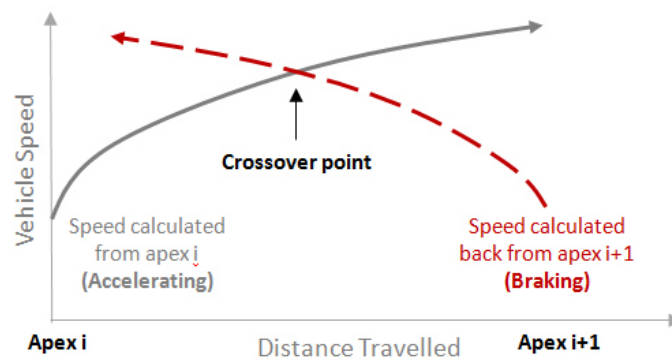


Figure 2.6: Quasi-static simulation principle

Research work in [35] shows, that despite the easier calculation approach, the simulation-result accuracy was not significantly influenced. The main difficulty is that this method is very sensitive on smoothness of trajectory dataset on input.

2.2.3 Quasi-transient method with pre-calculated MMM Diagram

All methods that have been so far listed are working under the assumption, that they are completely ignoring the time dependent phenomena of vehicle dynamics. Taking into account so-called transient effects, reflecting the response time on driver input, would be so the next logical step to increase the simulation precision.

³⁴ Cf. Candelpergher, Gadola und Vetturi, Developments of a method for lap time simulation (2000) according to Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013), S. 16

³⁵ D.L. Brayshaw and M.F. Harrison, A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location (2005), S.729, 730

There are several studies examining the effects of different calculation approaches on the final time results. One of the interesting findings³⁶ is the statement that considering the transient effects, such as, for instance, the influence of yaw inertia has only marginal effect on overall simulated time, but, on the other hand, could be significant when looking at the stability of the vehicle because the different yaw inertia affects the driver input consequently. Finally, a good car handling with adequate yaw stability is for the driver main performance-measure in order to be able to control the car on its limit.³⁵

Probably the easiest implementation of transient effects into lap time simulation is the slightly extension of quasi-static method originally based on g-g diagram. For taking the transient effects into account, the g-g will be replaced by a three dimensional Limit Acceleration Surface (LAS) that represents the performance envelope reflecting the combined lateral, longitudinal, and yaw acceleration.³⁶ LAS surface is then created from various MMM Diagrams³⁷ evaluating the vehicle handling by taking into account the steering angle and the tire slip angle. The major contribution of this method is in giving additional information about the stability and controllability of the vehicle on racing course.

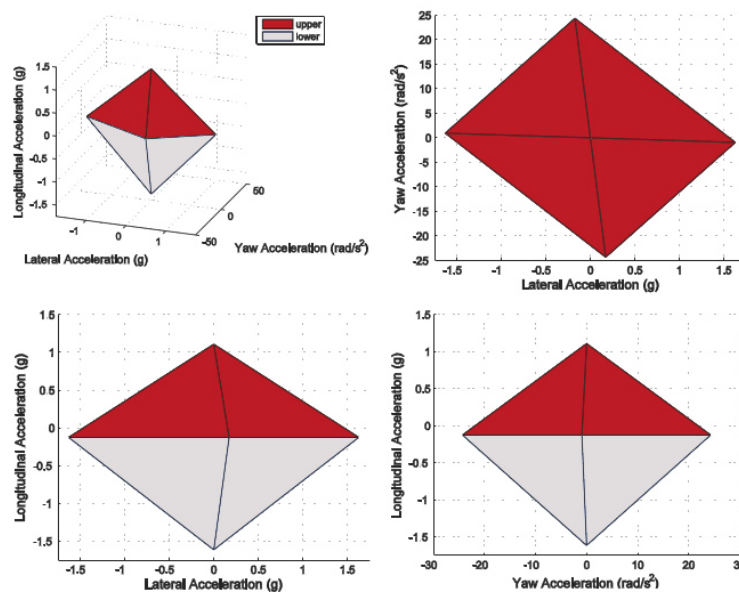


Figure 2.7: Limit Acceleration Surface (LAS)

³⁶ B. Siegler, A. Deaking, D. Crolla, Lap Time Simulation: Comparison of Steady State, Quasi-Static and Transient Racing Car Cornering Strategies (2000), S. 3

³⁵ Timo Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013), S. 121.

³⁶ Chris Patton, Development of Vehicle Dynamics Tool for Motorsport (2013), S. 62

³⁷ See Milliken Moment Method: Milliken, Race Car Vehicle Dynamics, S. 293

2.3 Transient method

The (quasi-)steady state methods, reviewed in the previous paragraphs, are working under the major assumption, that the tire lateral forces are always utilized on maximum of their availability to counteract the vehicle lateral acceleration. In reality, there must be counted with a certain time response of system that causes a bit delay till the maximal lateral force is reached.³⁸ *Fig.1.8* visualizes comparison of 3 different methods for simple right turn corner.³⁹

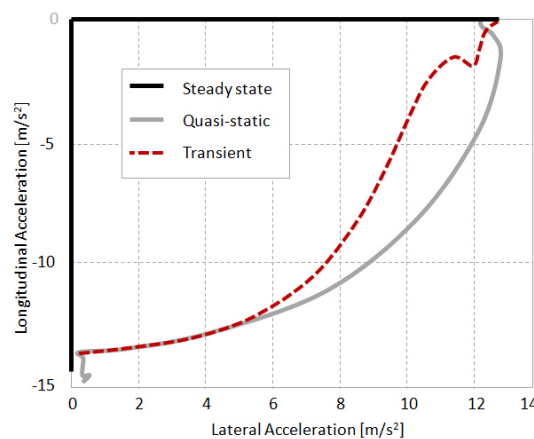


Figure 2.8: Method Comparism

Taking into account the various transient effects is currently the highest stage of model complexity in order to describe the real vehicle handling. The term “transient effects” refers firstly to yaw moments (See chapter 2.2.3), however, there are also another arts of transient behavior such as, for instance, the tire characteristic, already mentioned in chapter 2.2.1. On *fig.1.9* is illustrated the effect of mentioned TMeasy model and influence of temperature on speed performance while cornering.³⁸

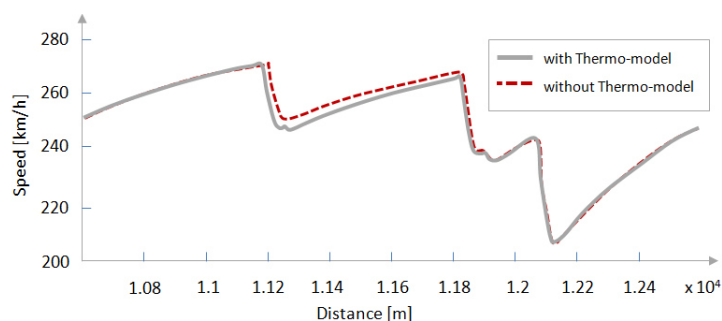


Figure 2.9: Thermo Model [4]

³⁸ D. Casanova, On minimum time vehicle maneuvering: The theoretical optimal lap (2000), S.6

³⁹ B. Siegler, A. Deaking, D. Crolla, Lap Time Simulation: Comparison of Steady State, Quasi-Static and Transient Racing Car Cornering Strategies (2000), S. 7 [4] Timo Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013), S. 93, 15.

2.3.1 Commercial applications - ADAMS

There are many computer programs such as ADAMS that use a transient simulation to show the time-varying response of a vehicle. In this case, the task is formulated in following way: For the purpose of performing the transient lap simulation, the control parameters of the vehicle model (steering, throttle/brake input) have to be found.

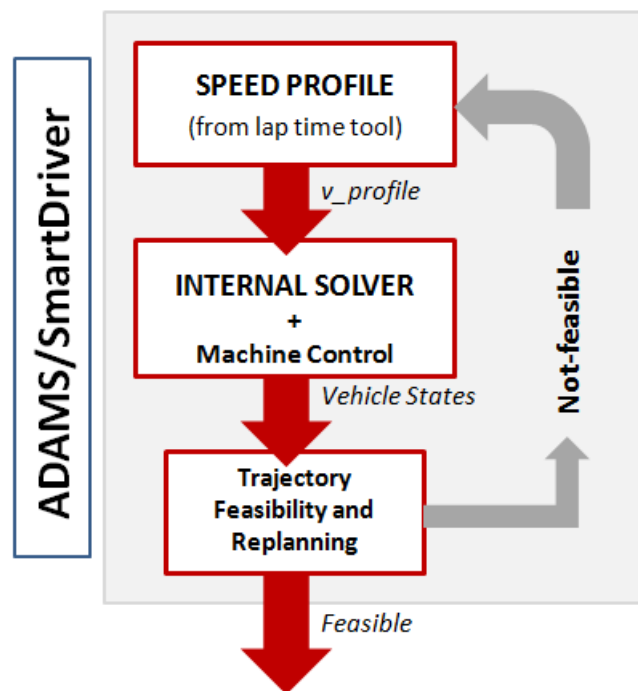


Figure 2.10: Simulation Stack – ADAMS [15]

The referred ADAMS system can be extended with an extra SmartDrive add-on module which is able to solve driving tasks for a given path line. The add-on module architecture is based on 3 layers, whereas the first one is simple quasi-static calculation of an initial speed profile, in a second step, it comes to forward integration of simplified car model taking into account inertial effects, load transfer, aerodynamics and driver demands. And finally a full analysis is performed in the last step by searching for path locations that cannot be traveled during the full dynamic simulation. If SmartDriver finds an unfeasible point, it notifies the quasi-static simulation, which modifies the speed profile and repeats the entire procedure until all feasible points are found.⁴⁰

⁴⁰ ADAMS/SmartDriver - Datasheet

2.3.2 Optimal control approach

In chapter 1.6, it has been found out, that minimal lap time is function of entire trajectory which depends on number of parameters.⁴¹ It is clear that any change in the vehicle configuration implies a different utilization of the tire forces and results in a different optimal racing line consequently.⁴²

The task is then formulated as an optimality problem and solved with the help of the optimal control strategy. Optimal control approach is characterized by a cost function which should be minimized (lap time). The main goal is to find a control history (steering angle, throttle/brake) of the dynamic system (vehicle model) for a set of constraints (road boundaries). A rigorous theoretical basis of optimal control methods is provided in ^[42] or ^[43]. In this short review, it will be mentioned only the key features and important characteristics.

The solution of optimality in this application area is mostly found with direct numerical methods. Because of functional connection between trajectory, speed and maximal vehicle performance, the parallel calculation of minimal-time trajectory as well as speed profile causes a huge numerical effort.⁴⁴ The calculation costs of one circle on track in Monza are then within the range of 24 hours.⁴² The biggest threat for calculation process is a strong non-linear behavior of a car driving on the limit, which can lead to height gradients restricting the car controllability eventually.⁴⁴

The bad numerical stability is possible to improve with various stability criteria or by coupling the simulation with an additional drive controller guaranteeing the desired stability⁴⁴. With drive controller, the optimal control methods are also able to reflect for simulation different driving strategies such as, for instance, trail-braking or pendulum-turn.⁴⁵

⁴¹ See Equation 1.15 in Chapter 1.4.3

⁴² D. Casanova, On minimum time vehicle maneuvering: The theoretical optimal lap (2000), S.5

⁴⁴ Timo Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013), S. 93, 15.

⁴³ Gustafsson, Computing The Ideal Racing Line Using Optimal Control (2008)

⁴⁵ Velenis, Tsiotras, Lu, Modeling Aggressive Maneuvers on Loose Surfaces: The Cases of Trail-Braking and Pendulum-Turn (2007)

2.4 Summary

A previous text shows a stepwise application of various methods for lap time-based simulations. It covers the very early experiments and then step by step is trying to achieve a maximal possible accuracy between calculated simulation-result and reality on the racing track. As a first important statement is the fact, that all approaches are more or less based on those first analytical calculations from 50's (see fig. 1.2). Even the last one transient approach needs a kind of initial guess, which is efficiently performed by a quasi-static application.

2.4.1 Overall evaluation of methods

Practically there are two most significant sources of uncertainties - the chosen parameter depth and calculation method itself. In paragraph 1.2.1 - the friction coefficient was mentioned on the first place on the ratings of most time-influencing parameters. This implies for parameterization two statements – *“Firstly, the precise description of the tire model and, secondly, the perfectly mapped racing track is needed.”*

Type of calculation method, as a second significant source of uncertainty, is constrained by a given assumptions – mostly in order to reduce calculation time. It has been already mentioned various works comparing simulation methods with or without involvement of the transient effects.⁴⁶ Results of those reports give a hope for generally quite a good convergence of both methods.

More detailed evaluation of calculation methods did quite recently Timo Völkl in his dissertation.⁴⁷ He performed on this place a kind of strength-weakness analysis, whereas he defined following criteria for method evaluation: *precision, accuracy and robustness*.

On the basis of his report, the transient approach can give results with very high accuracy, but under the major threat, that calculation process could converge in a wrong direction.

⁴⁶ B. Siegler, A. Deaking, D. Crolla, Lap Time Simulation: Comparison of Steady State, Quasi-Static and Transient Racing Car Cornering Strategies (2000)

⁴⁷ Timo Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013)

That's why the robustness (reliability of Algorithm) is in by this method significantly reduced. On the opposite side of reliability spectrum are standing the (quasi-)steady-state methods. - Working with static equilibriums ensures the stability and generate so a better precision. The precision as a measure of high repeating accuracy is then important property for performing of diverse parameter studies.

2.4.2 Final statements

The final message – In development of a lap time simulation is always good to respect the top-down rule. Already a good basis with an easy quasi-static model can give very satisfactory results.

That's also the strategy that will be followed in this work. The main goal is to prepare a reasonable first version of lap time simulation on basis of given data. Application of advanced calculation techniques would require enormous quantity on specifications to identify the model and find out the relations in between.

3 Tool development

The following sections cover the creation of simulation tool.

At first, the requirements and needs are collected for general tool specification. On this basis, the overall strategy will be formulated in form of proposed system architecture. The most demanding job is than the further detailing and implementation of individual components for simulation tool. After having the first running release, the test and validation phase can be started.

The whole process, as outlined above, is based on typical V-Model methodology describing the software development lifecycle.

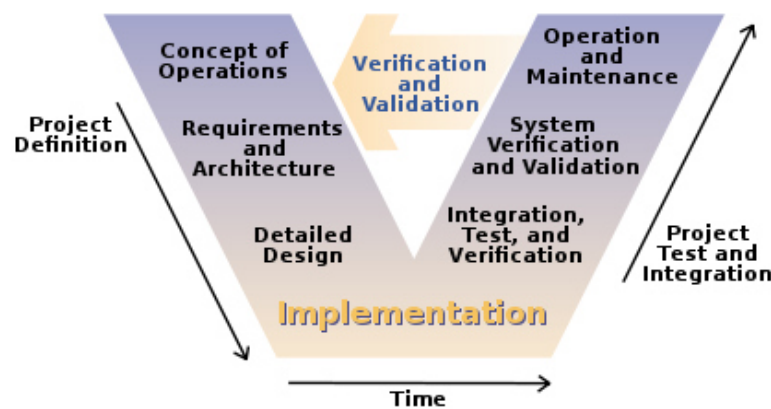


Figure 3.1: V-Model ⁴⁸

3.1 Tool specification

Development of lap time simulation tool was initiated by power train department in Porsche Engineering Services s.r.o. in order to have an own tool running on MATLAB/Simulink platform which would be able to evaluate vehicle performance on particular track in form of elapsed time and fuel consumption.

⁴⁸ source: <https://en.wikipedia.org/wiki/V-Model>

3.1.1 Requirements on developer software

Simulink, as a part of Matlab numerical computing package, is a graphical programming environment for modeling, simulating and analyzing dynamic systems.⁴⁹ Model-based development perfectly fits to V-Model strategy (modeling, analyzing, simulating and integration)⁵⁰ and thereby is preferred as ideal way for programming of embedded⁵¹ automotive software.

The main target of using MATLAB/Simulink for this lap time simulation is to have a fast evaluation tool based on the same background environment as for development of driveline and its control functions. In the end, it should spare problems with data-format incompatibilities and allow easy integration of new functionalities eventually.

3.1.2 Considerations about calculation Method

From first investigations is already known, that racing time as a function of trajectory depends on variety of factors⁵² which leads to quite complicated optimization task. Solving of these tasks in MATLAB Package is possible, but mostly it requires an extra commercial Toolbox. For instance Modelica/Optimica Add-Ons,⁵³ implemented in work by T. Gustafson.⁵⁴

To avoid using of any 3rd party software (major requirement of submitter), it has been decided to find out a reasonable calculation approach involving no optimization procedures, but still providing high-quality results. The quasi-steady state methods⁵⁵ perfectly fit into this concept – They are fast, easy to implement and, according to published works,⁵⁶ accurate enough in compare to real measured laps or results of another simulation methods.

⁴⁹ "Simulink," Wikipedia, last modified July 22, 2016, <https://en.wikipedia.org/wiki/Simulink>.

⁵⁰ "Model-based desing," Wikipedia, last modified June 23, 2016, https://en.wikipedia.org/wiki/Model-based_design.

⁵¹ Computer software written to control machines or devices that are not typically thought of as computers. (Wikipedia.org)

⁵² See Chapter 1.5

⁵³ See <http://www.modelon.com/industries/automotive/motorsports/>

⁵⁴ Thomas Gustafsson, Computing the ideal racing line using optimal control (2008)

⁵⁵ See Chapter 2.2

⁵⁶ B. Siegler, A. Deaking, D. Crolla, Lap Time Simulation: Comparison of Steady State, Quasi-Static and Transient Racing Car Cornering Strategies (2000)

The whole simplification of quasi-steady state approach is based on the assumption that the driving path is already pre-defined. The number of variables is hereby reduced quite substantially - The racing time becomes so a direct function of main vehicle properties and therefore the racing line must be no more re-calculated for each change of vehicle set-up. Next point of using this approach is that there are also quite high requirements on track description in case of using trajectory optimization techniques. In connection with this, there would be a danger that high time sensitivity of trajectory form could re-write the real time influence of set-up change.⁵⁷ The high-repeating accuracy mentioned in former chapter⁵⁸ couldn't be so guaranteed.

The findings of chapter 2 and the research works⁵⁹ confirm this strategy as relevant, under the condition that vehicle, used for recording the trajectory file, is conceptually and parametrically not too far from the simulation model:

*"It was found out that, despite the high variation of parameters with a significant time influence, the driving line has been modified only slightly. This leads to assumption that for parameter variations with moderate racing time impact, there is no need of any additional adjusting the driving trajectory."*⁵⁹

In original:⁵⁹ „... Da sich die Fahrlinie, auch bei Änderungen von Parametern mit großem Rundenzeiteinfluss, nur wenig verändert wird angenommen, dass es nicht nötig ist, die Fahrlinie bei Rundenzeiteinflüssen im mittleren Bereich anzupassen. “

3.1.3 Concept of implementation into model-based system

Simulink is a software package that enables to model, simulate, and analyze systems whose outputs change over time.⁶⁰ Such systems are referred to as dynamic systems and mathematically described in form of differential equations. The simulation is then performed by numerical integration methods (e.g. Euler) from a specified start time to a specified stop time.

⁵⁷ Timo Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013), S. 151.

⁵⁸ See Chapter 2.4.1

⁵⁹ Cf. Mühlmeier und Müller, "Optimization of the driving line on a race track" (2002) according to Timo Völkl, Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen (2013), S. 151.

⁶⁰ "How Simulink Works," MathWorks Documentation, last modified February 22, 2016, http://www.mathworks.com/help/simulink/ug/introduction_f7-5734.html.

From this point of view, the time line could be understood as a kind of control input. However, the problem is that in case of racing simulation, the time value is also unknown variable. In order to solve this issue, suitable transformation procedure has to be found.

Knowing the fact, that simulation workflow is based on continuous time integration, let's consider the velocity as a first derivation of the track after the time. In opposite, it means that to get information about driven distance, the velocity has to be integrated over the specified time period which should have been indeed the main objective of lap time simulation. In order to reach this period and hereby the real racing time, the integration process has to be stopped in right moment. In this case, the stop-point is specified by the overall circuit length. When this milestone is reached, the final time can be read. (fig.3.2)

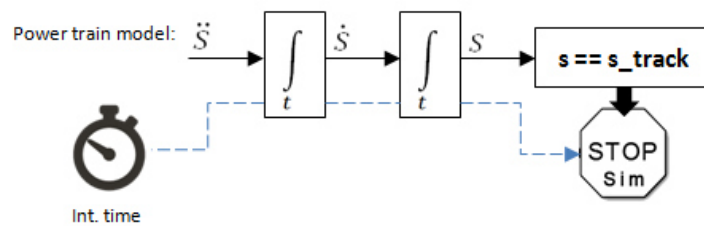


Figure 3.2: Transposition method

On this basis, the originally time based input can be easily transposed to the track based input. In other words, it means that the controller of vehicle model is able to operate on the basis of track information.⁶¹ For instance, if there is a curve with critical corner speed 50km/h on the track milestone at distance 500m from start, the controller initiate braking maneuver automatically, when the particular distance is reached. However, for this kind of speed controller realization is important to perform at beginning a track analysis in order to create a speed prescription as a control input. Because of assumption of „frozen” trajectory line, this step can be done quite quickly by the help of a script calculating so-called initial speed diagram. In mother of fact, it's nothing else than a quasi-steady state analysis of trajectory constrained by maximal adhesion capabilities of vehicle in longitudinal and lateral direction building the control boundaries for longitudinal Simulink model.

⁶¹ See Chapter 4.2.3

The detailed description of this two-stage simulation approach will be given in chapter 4. For better understanding of whole functionality, there is a graphic diagram (fig.3.3) illustrating the complex simulation stack with detailed simulation layer.

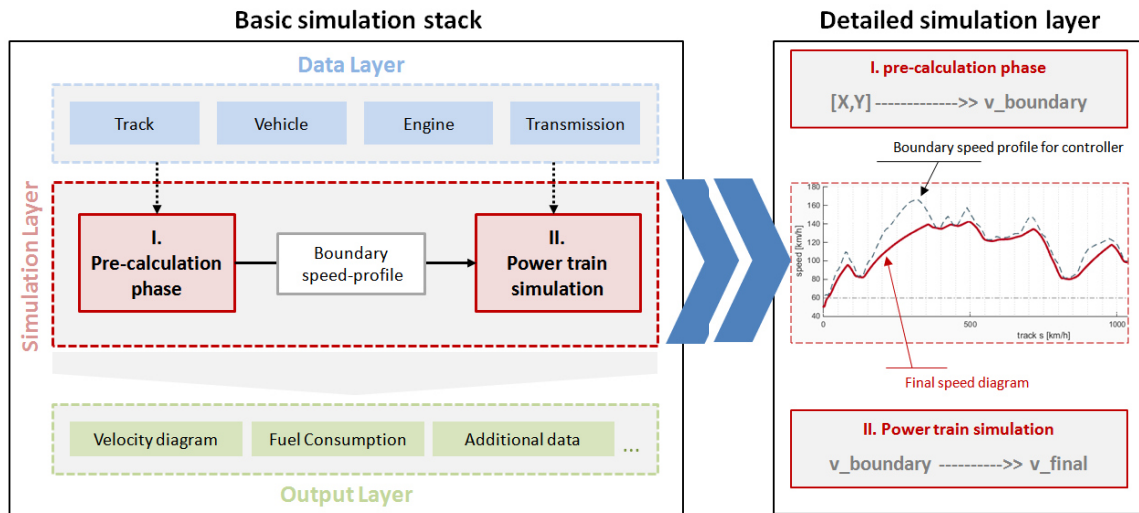


Figure 3.3: Basic Simulation Stack with detailed Simulation Layer

3.1.4 Summary of conceptual simulation strategy

After introducing of basic key points of calculation methodology, there will be given a brief summary outlining the concept of proposed simulation workflow.

Statement 1: „The overall computing strategy is proposed as a two-stage simulation.”

Because of reason that simulation is primary focused on power train system evaluation, it has been decided to use the simulink environment only for longitudinal vehicle model. Two-stage simulation concept should solve the issue about control strategy of simulink model. At this place, the preferred control approach is the speed adjusting on basis of pre-calculated velocity diagram.

Statement 2: „Before starting the model-based simulation (Stage II), its control input has to be calculated in a first step. On the basis of given trajectory in xy coordinates, the control input as a initial velocity profile will be built. It represents maximal adhesion capabilities of vehicle on the track.”

Initial or boundary speed profile for particular trajectory is obtained from quasi-static analysis of g-g diagram utilization.⁶² By considering of only maximal friction relations, the algorithm is spared of complicated iteration process in order to look for maximal acceleration capabilities for different transmission ratios over the engine rev range. The necessary input track data are obtained from company intern database collecting the real driven trajectories and telemetry data of various vehicles.

Statement 3: *“The boundary velocity profile specifies the operation area for speed controller of model based drive line. The final velocity profile is in the end not only result of friction capabilities but also the drive train characteristics.”*

The very complex models describing the power train functionalities are in property of Porsche Engineering Services s.r.o.. For objective of this thesis, it will be introduced own model of conventional driveline using publicly released datasets. Optionally should be possible to replace this sample model for an advanced-one including, for instance, the hybrid functions. For this reason, the perfect description of model input/output interfaces is obligatory. (Specified in chapter 3.2)

3.2 Tool architecture

In the last paragraph, there have been formulated the basic ideas and principals of Matlab/Simulink-based lap time simulation. In order to have a better overview of whole realization process, it will be in the next part specified the complete system of simulation architecture. This should help to clarify all the relations among system elements and set a clear workflow for calculation sequence. The simulation range is specified additionally in table below. Fundamentally as a basis for architecture model the simulation stack introduced in *fig.3.3* has been taken.

⁶² See Chapter 2.2.2

3.2.1 Simulation workflow

The structured simulation flow on *fig. 3.4* is the first step for detailed description of proposed calculation model. According the colored blocks in workflow, there is the same functional splitting as in diagram showing the simulation stack. (*fig. 3.3*) The data layer (=input information) is blue, the calculation blocks are red, and results are outlined with green color. The simulation flow is than shown in arrow direction and all entities are again listed in table 3.1.

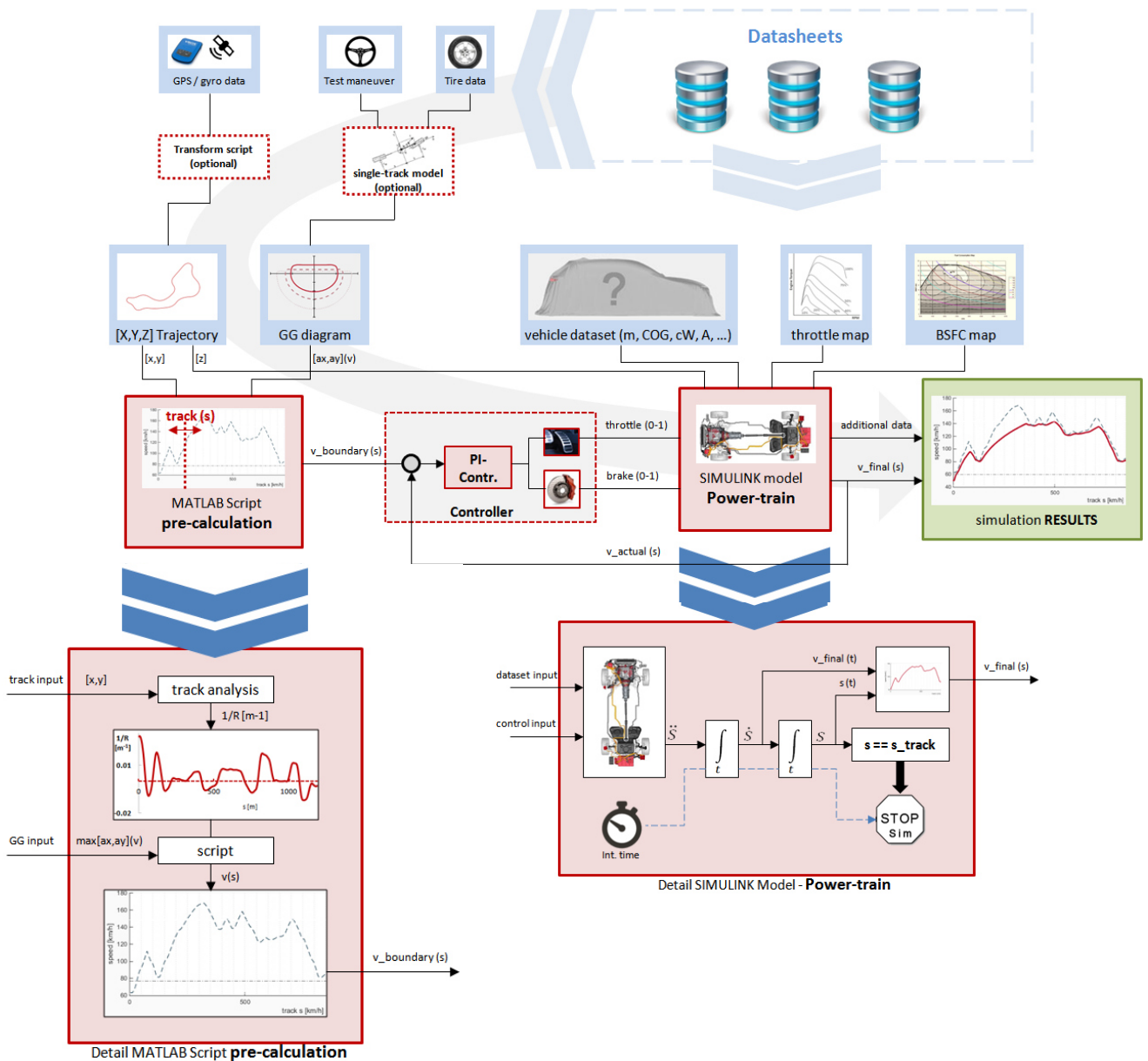








Figure 3.4: Workflow Lap Time Simulator

| Data Layer | | | | |
|------------|--|---|--|-----------|
| No. | Input name | Variable name | source | |
| 1 | Drive trajectory | {x; y; z} | intern | |
| |  GPS Trajectory  Telemetry data | {lat.; long.; hor.} {ax; ay} | GPS .dat file (when no raw xy data) | |
| 2 | GG Diagram | {ax_max; ay_max} | intern (when no raw data) | |
| |  Tire data | {pac2002} | | |
| |  Single Track model | {lf;lr} | | |
| |  Aero down force | {A; C _{af} ; C _{ar} } | | |
| |  Test maneuver input | {steer_in} | | |
| 3 | Vehicle data | <i>Curb weight</i> | {m _v } | datasheet |
| | | <i>Load weight</i> | {m _{load} } | |
| | | <i>Wheel radius</i> | {r _{dyn} } | |
| | | <i>Aero. drag</i> | {c _W } | |
| | | <i>Front Area</i> | {A} | |
| 4 | Engine data | <i>Throttle map</i> | {engine map} | datasheet |
| | | <i>BSFC map</i> | {bsfc map} | |
| 5 | Transmission | <i>Shift ratios</i> | {i _{1,2,...,n} } | datasheet |
| | | <i>Diff ratio</i> | {i _{diff} } | |
| | | <i>RPM to shift</i> | {n _{up} ; n _{down} } | |

| Simulation Layer | | | | |
|------------------|--|-------------------------------|--|--------------|
| No. | Calculation method | Input | Output | Data layer |
| 1 | Stage I Pre-calculation | [x,y] | -> [v_{boundary}] | 1,2 |
| | | - Track analysis | [x,y] -> [1/R(s)] | |
| | | - Script boundary vel. | [1/R(s)] -> [v _{boundary} (s)] | |
| 2 | Stage II Power train simulation | [v_{boundary}] | -> Results | 3,4,5 |
| | | - Controller | [v _{boundary} (s)] -> [throttle/brake(t)] | |
| | | - Power train model | [throttle/brake(t)] -> [x_dotdot] | |
| | | - subsystems | | |

| Output Layer | |
|--|-------------------------|
| Output variables | Dependent on... |
| Velocity profile, Fuel Consumption, RPM, Gear, Accelerations, ... | Track & Time |

Table 3.1: Simulation Layers

3.3 Final statements:

In this section about Tool development, the basic functional principle of simulation has been proposed. Finally, the simulation workflow is sketched in fig. 3.4. Additionally, there is also the table (tab. 3.1) with all entities that are for development of simulation tool needed.

The simulation program is designed under following assumptions:

- Program works with pre-defined trajectory in form of [xyz] dataset.
- The initial velocity profile is calculated with quasi-static method using the g-g diagram
- The g-g diagram boundaries are estimated by the help of non-linear single-track model, additionally, it should be possible to adjust the g-g diagram manually from extern dataset.
- Power train model is working with given throttle and BSFC map
- Diverse driving strategies are realized by adjusting the shifting RPMs and maximal throttle position.

4 Simulation model

In conceptual part, basically focused on model complexity, it was tried to identify the real model range by investigating the relations between particular calculation steps. The resulted simulation complexity is than summarized in table 3.1 with all important characteristics that cover all three functional layers (*fig.3.3*). After knowing all constrains and having the basic clue about simulation workflow (*fig.3.4*), it can be started with detailed designing of single functionalities.

The structure of this chapter will be similar to the introduced workflow - generally divided into two main parts reflecting the layout of simulation layer (Stage I & Stage II). The main purpose of two-stage based simulation is the demand on appropriate control strategy for Simulink model.⁶³

4.1 Simulation model – Stage I

The aim of “Stage One” is to pre-calculate so-called initial velocity profile. It’s considered as theoretical speed boundary when only adhesion limits are taken into account. In this part, it won’t be reflected any performance limitation caused by power-train, only limits between track and maximal tire capability will be investigated.

In accordance with the simulation workflow (*fig.3.4*), at first, the track or, more precisely, the trajectory is analyzed. Then, it comes the main phase with calculation of initial speed profile. Additionally, it will be mentioned an alternative approach how to get position data and finally, at the end, the adhesion limits in form of vehicle performance envelope will be discussed. Complete structure of chapter is noted here: ^[64]

⁶³ See Chapter 4.2

⁶⁴ Structure of following chapter 4.1:

4.1.1 Track analysis

- 4.1.1.1 - Functional realization
- 4.1.1.2 - Alternative approaches

4.1.2 Initial velocity profile

- 4.1.2.1 - Functional realization
- 4.1.2.2 - Functional validation

4.1.3 Vehicle performance envelope

- 4.1.3.1 - idea of g-g diagram modification
- 4.1.3.2 - Practical realization

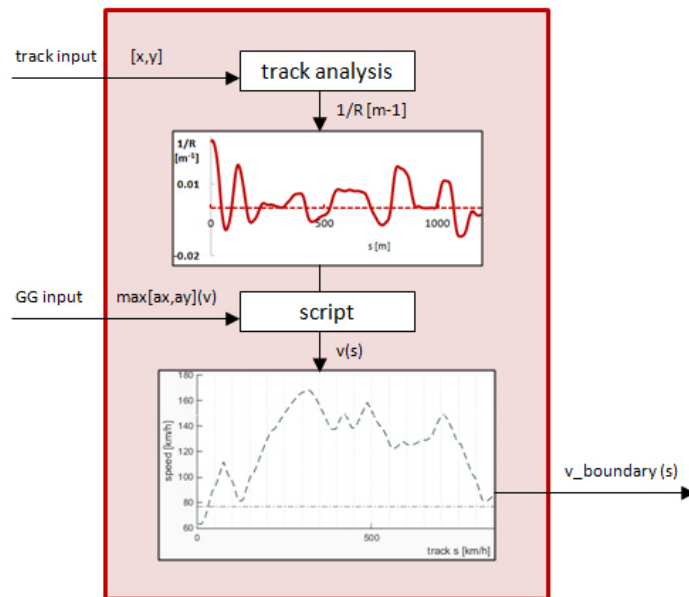


Figure 4.1: Simulation Model – Stage I

4.1.1 Track analysis

The basic input dataset comes in form of XY respectively XYZ coordinates. Explicit knowledge of coordinates is good for direct visualizing the track form, but on the other hand, the pure XY coordinates do not offer any qualitative description that could be used for calculation of initial velocity profile directly.

The main aim of track analysis is thereby focused on trajectory description which could be efficiently used for next processing.

From loaded position dataset [xy], it will be at first calculated the course angle ψ and then the track curvature ρ .⁶⁵ The whole theory is formulated by the help of equations 4.1 and 4.2. Precision of this procedure depends on density of trajectory points and chosen step size delta Δ .

$$\psi = \text{atan} \left(\frac{\Delta y}{\Delta x} \right) \quad (4.1)$$

$$\rho = \frac{\Delta \psi}{\Delta s} \quad (4.2)$$

⁶⁵ Peter Waldmann, Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken (2009), S. 24

4.1.1.1 Track analysis - functional realization

The functionality of track analysis will be explained on the path section between points P_i and $P_{i+\Delta}$ that are both building the trajectory line. Fundamentally, the method is based on geometrical properties of red and blue highlighted triangles in *fig. 4.2*. Both of them are determined by step delta Δ and the functional meaning of them will be explained together with a description of algorithm analyzing the whole trajectory in the next text. Realized script code is attached in appendix 1.

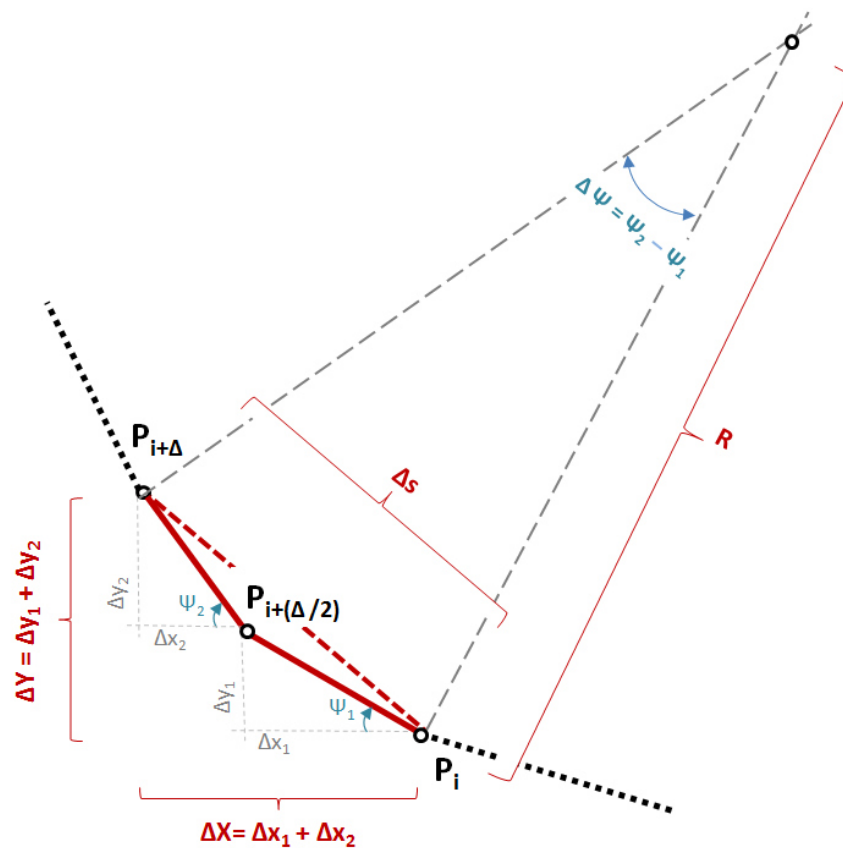


Figure 4.2: Track Analysis

1, Definition of step size delta (Δ)

In order to get current track radius R , the blue triangle defined with the side Δs and angle $\Delta \psi$ is necessary. The change of course angle $\Delta \psi$ comes geometrically from difference between ψ_1 and ψ_2 . Therefore, the step size Δ must be defined only as an odd number higher than 2 in order to guarantee a point in the middle for construction of two same-sized elements that allow the identification of difference $\Delta \psi$.

Because of input datasets with various data qualities, it has been decided to declare the step size delta also as an input variable to give the end-user a chance to adjust the step size value manually.⁶⁶ In general, it is recommended to choose delta-size Δ from interval $\langle 6;12 \rangle$. Too low values are generating high oscillations, in opposite, higher delta values could rewrite smooth alternations of the trajectory form.

2, Identification of $P_i - P_{i+\Delta}$ section geometry

Knowing the delta step size, the critical parameters ($\Delta\psi, \Delta s$) for curvature estimation ρ can be obtained in very efficient way according to already explained relations in *fig 4.2* and described by basic equations 4.1 and 4.2.

3, Definition of datasets and final data processing

Thus the curvature of a circle is defined to be the reciprocal of the radius ($1/R$).⁶⁷ The main advantage of this concept is the fact that the curvature value ($\rho=1/R$) of straight line tends automatically towards zero. From this point of view, it is the ideal way how to reliably describe a continuously varying trajectory form on relatively small value range with preserving quite high sensibility on direction variability at the same time.

Assuming that the typical civil car has turning circle of $12m$, the general curvature limit of $0.08m^{-1}$ can be expected for next data processing.

On this basis, the final filtering procedure was developed. Designed data filters are functionally divided into two groups and practically presented on the highlighted part of driving trajectory in *fig.4.2*.⁶⁸

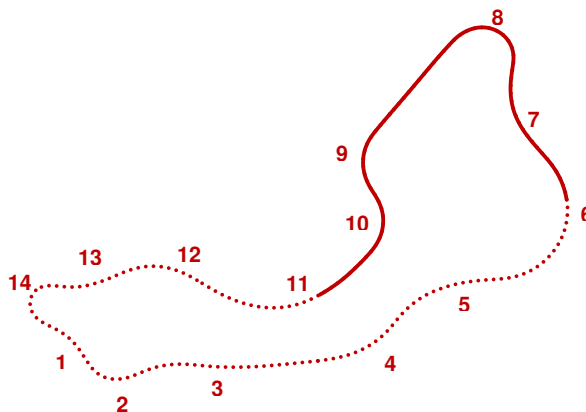


Figure 4.3: Test Track [6]

⁶⁶ See Chapter 6 - GUI

⁶⁷ "Curvature of plane curves" Wikipedia, last modified June 30, 2016, <https://en.wikipedia.org/wiki/Curvature>.

⁶⁸ Idiada Test Track, plotted trajectory

3a, Array adjustment

The calculated data are plotted in *fig. 4.4* with gray lines (raw data). In a first step, all points have to be continuously connected (red line), as well as the identified out-of-range peaks must disappear promptly (gray line - *fig.4.4*).

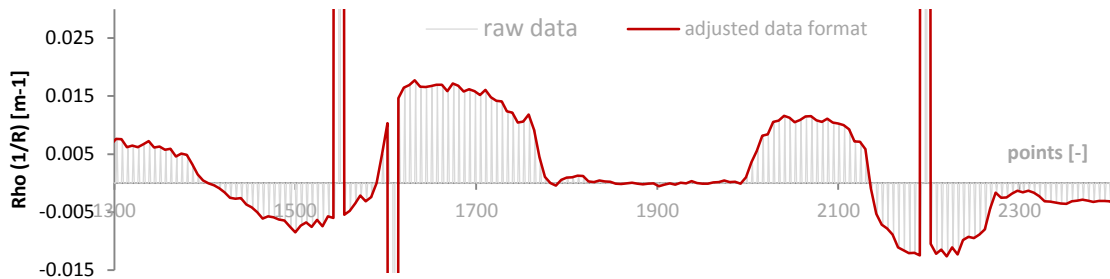


Figure 4.4: Data Filtering – Step 1

3b, Final smoothening

On such prepared curve profile, the Smooth function from Curve Fitting Toolbox⁶⁹ can be finally applied. This routine works with additionally specified x-axis data (=track distances), just for the case that recorded data are not uniform spaced. The parameter of smoothening strength will be again specified as an individual user-input for GUI.⁷⁰

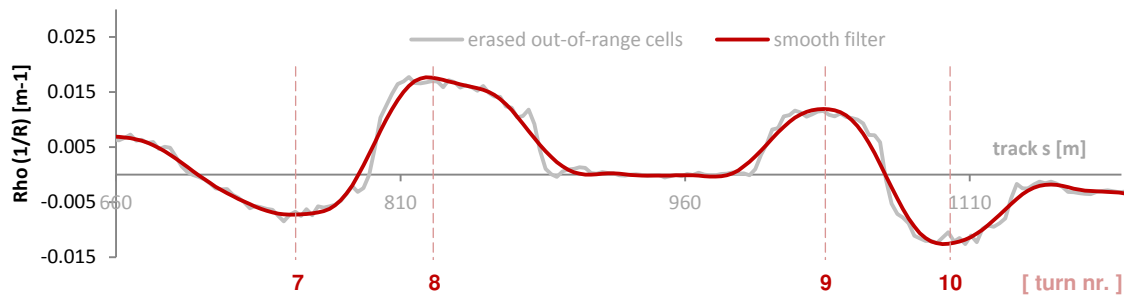


Figure 4.5: Data Filtering – Step 2

4.1.1.2 Track analysis - alternative approaches

In case, there is no prepared [x,y] dataset, it's possible to obtain position data by importing them from a telemetry device – just as it was proposed in workflow on *fig. 3.4*. In general, depending on the art of device, there are two ways how to get the track description:

⁶⁹ See: <http://www.mathworks.com/help/curvefit/smooth.html>

⁷⁰ See: Chapter 6.1

a, Converting the GPS position data [long./lat.]

Basically, the GPS device is working with ellipsoidal interpretation of earth globe, which most closely approximates the shape of the Earth. The common way of stating terrestrial position is with two angles, latitude and longitude.⁷¹ In order to get position in Cartesian coordinates, the ellipsoidal position must be transformed. For this procedure, there is a special Mapping Toolbox in Matlab⁷² with command *geodetic2enu* using the following syntax:

$$[xEast, yNorth, zUp] = geodetic2enu(lat, lon, h) \quad (4.3)$$

b, Integration of acceleration data [ax,ay]

Alternatively, there is option to obtain curvature profile directly based on processing the recorded acceleration data. The basic workflow is shown on *fig. 4.6* working on principle of blocks that integrate the acceleration data recorded on the track with telemetry device. As shown on *fig. 4.7*, in comparison between original calculated curvature by telemetry device and Simulink output, it's clear that calculation in telemetry device is working on the same principal as in *fig. 4.6*.

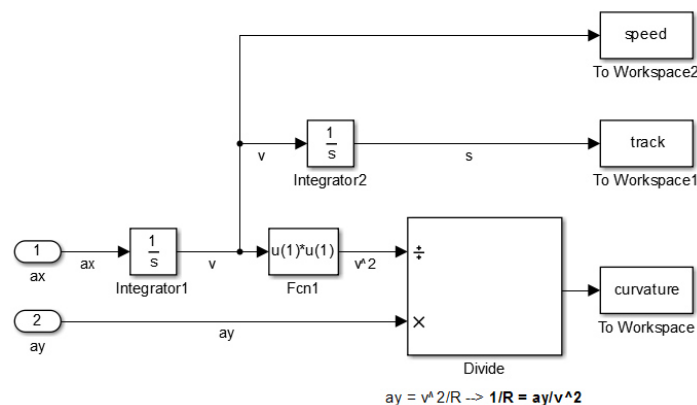


Figure 4.6: Calculation of Curvature (Simulink)

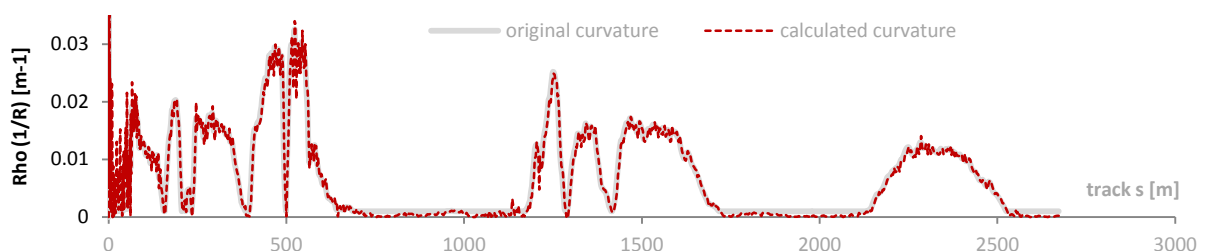


Figure 4.7: Original vs. Calculated Curvature

⁷¹ Ordnance Survey, A guide to coordinate systems in Great Britain, (2015), S.9, <https://badc.nerc.ac.uk/help/coordinates/OSGB.pdf>

⁷² See: <http://www.mathworks.com/help/map/ref/geodetic2enu.html>

4.1.2 Initial velocity profile

As described in chapter 3.1.3, the pre-calculated velocity profile is fundamental part of controller strategy for power train model in Simulink. Considering that the power train model actually represents only longitudinal part of dynamic, there must be found a way how to take into account also the lateral dynamic. In other words, the control input for longitudinal model has to already include very precise information about interactions in vehicle-track system. The whole operation principle is technically based on closed-loop of speed adjusting, which was conceptually already sketched in simulation layer on *fig. 3.3*.

Speed profile generated in this part so represents the maximal feasible velocity that satisfies the adhesion limitation. This speed profile will be denoted as initial. Proposed calculation method comes from quasi-static approach investigating the maximal allowed speed for every track segment by the help of predefined vehicle performance envelope - the g-g diagram.⁷³ Realized script code is attached in appendix 2.

By considering only adhesion limits, the applied performance envelope can be substantially simplified – the limitation of realistic accelerations will be simulated later on by simulink model. Basically, for initial speed profile, there are only two information needed – maximal lateral and longitudinal capability. Because of taking the aerodynamic down force into account, both acceleration values must be speed depended. More about investigation of performance envelope for particular vehicle will be given in next chapter 4.1.3. On this place, there is just assumed that for calculation of initial speed profile, the elliptical performance envelope defined for each speed level with $a_{y_{\max}(v)}$ and $a_{x_{\max}(v)}$ will be used. As per every quasi-static approach, time dependant phenomena are ignored.

4.1.2.1 Initial velocity profile - functional realization

The calculation method will be presented in form of case study comparing two car concepts with different influence on aerodynamic lift forces. The first case with neglected aerodynamic will be used for explaining of elementary relations for computing model of initial speed diagram. Every step will be then extended with approach reflecting the down forces produced in higher speeds.

⁷³ See: Chapter 2.2.2

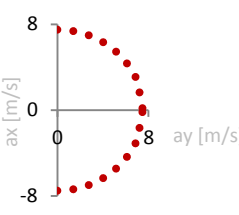
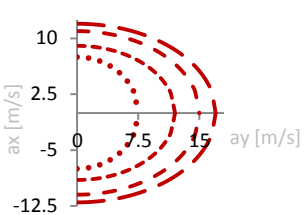
| Use case no.1 = Civil car | | Use case no.2 = Sports car | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|----------------------------|----|----|------|------------------|------------------|--------|-----|-----|--|--|---|----|----|------|------------------|------------------|----|-----|-----|-----|---|----|-----|----|----|-----|----|----|
|  | <table border="1"> <thead> <tr> <th>v</th> <th>ax</th> <th>ay</th> </tr> <tr> <th>km/h</th> <th>m/s²</th> <th>m/s²</th> </tr> </thead> <tbody> <tr> <td>const.</td> <td>7.5</td> <td>7.5</td> </tr> </tbody> </table> | v | ax | ay | km/h | m/s ² | m/s ² | const. | 7.5 | 7.5 |  | <table border="1"> <thead> <tr> <th>v</th> <th>ax</th> <th>ay</th> </tr> <tr> <th>km/h</th> <th>m/s²</th> <th>m/s²</th> </tr> </thead> <tbody> <tr> <td>50</td> <td>7.5</td> <td>7.5</td> </tr> <tr> <td>100</td> <td>9</td> <td>12</td> </tr> <tr> <td>150</td> <td>11</td> <td>15</td> </tr> <tr> <td>200</td> <td>12</td> <td>17</td> </tr> </tbody> </table> | v | ax | ay | km/h | m/s ² | m/s ² | 50 | 7.5 | 7.5 | 100 | 9 | 12 | 150 | 11 | 15 | 200 | 12 | 17 |
| | v | ax | ay | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| km/h | m/s ² | m/s ² | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| const. | 7.5 | 7.5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| v | ax | ay | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| km/h | m/s ² | m/s ² | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 50 | 7.5 | 7.5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 100 | 9 | 12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 150 | 11 | 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 200 | 12 | 17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 4.1: Two Car Concepts for Case Study

The parameters of performance envelopes for this case study are chosen fictively with main purpose to demonstrate the algorithm functionality. As test environment, the test track from *fig.4.3* is selected.

Step 1 – Initial guess

In a first step, the maximal cornering speed is calculated for every point N of driving trajectory which is specified by curvature value ρ .⁷⁴ The speed profile depends only on maximal lateral capacity $a_{y,max}$ and curvature ρ . (*fig.4.8*). The acceleration or braking phase is in this step not considered yet. Maximal velocity is constrained by parameter v_{max} .

$$v_{step1(n)} = \min\left(v_{max}, \sqrt{\frac{a_{y,max}}{\rho(n)}}\right), \quad n = 1 \dots N \quad (4.4)$$

In addition, the original algorithm (*Eq. 4.4*) has been appropriately modified in order to include the effect of aero down force causing the increase of lateral acceleration capacity $a_{y,max}$. The change evoked by an aero down force is marked with Δv_n and depends on interpolated data from extended g-g diagram (*Table 4.1 – Use Case 2*). The lateral acceleration limit $a_{y,max}$ becomes so a velocity dependent variable. To get the final end speed, the lateral acceleration limit for every track point n must be iteratively recalculated.

The functionality is proved by displayed speed difference between Δv_3 and Δv_4 on *fig.4.8*. The increase of lateral capacity in slower corners is lower than in faster passages where the higher speed generates more down force.

⁷⁴ See: Chapter 4.1.1

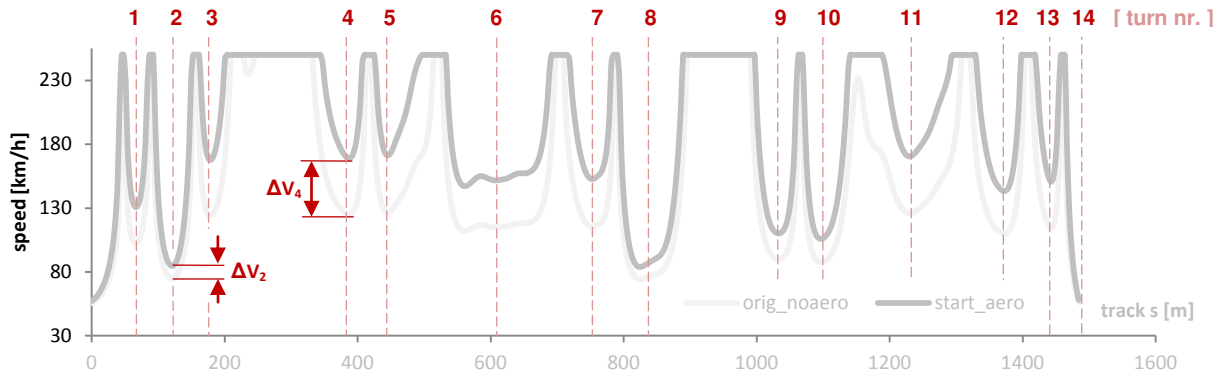


Figure 4.8: Initial Speed-Guess (Test Track from *fig.4.3*) – Step 1

Step 2 – Braking phase

Regarding the chapter 2.1, where the basic steady-state strategy, with the accelerating and braking curves, have been outlined, the same conceptual idea is now applied in this calculation model.

In order to correct extremely high speed gradients of initial guess (*fig.4.8*), in a second step, the velocity profile will be adjusted accordingly the maximal available braking capacity a_x . In general, the whole algorithm is running backwards and according to actual operating position in g-g diagram, the adjusting process incrementally calculates the new speed level on base of available a_x for particular point n .

For particular algorithm realization (*see fig. 4.9*), at first, the local minimum representing the critical cornering speed has to be found (*point $s_{(i)}$*). The saturation of lateral capacity a_y in this point is used on maximum level ($a_y = a_{y,max}$). This is the reason why in this point any longitudinal acceleration a_x can't be performed. The same thing, the utilization of lateral capacity, will be checked for next point $s_{(i-1)}$ (eq. 4.5) but under the assumption of fixed speed level $v_{(i)}$ ($v_{(i)} = v_{(i-1)}$). Because of lower track curvature ($\rho_{(i-1)} < \rho_{(i)}$), the lateral capacity for same speed level is not fully saturated and an available longitudinal capacity a_x (eq. 4.6) can be found that is directly used for intended braking maneuver from $v_{(i-1)}$ to $v_{(i)}$. The last step than represents the final calculation of new velocity value $v_{(i-1)}$ for point $s_{(i-1)}$ (eq. 4.7). In *fig.4.9* is the complete algorithm function showed graphically.

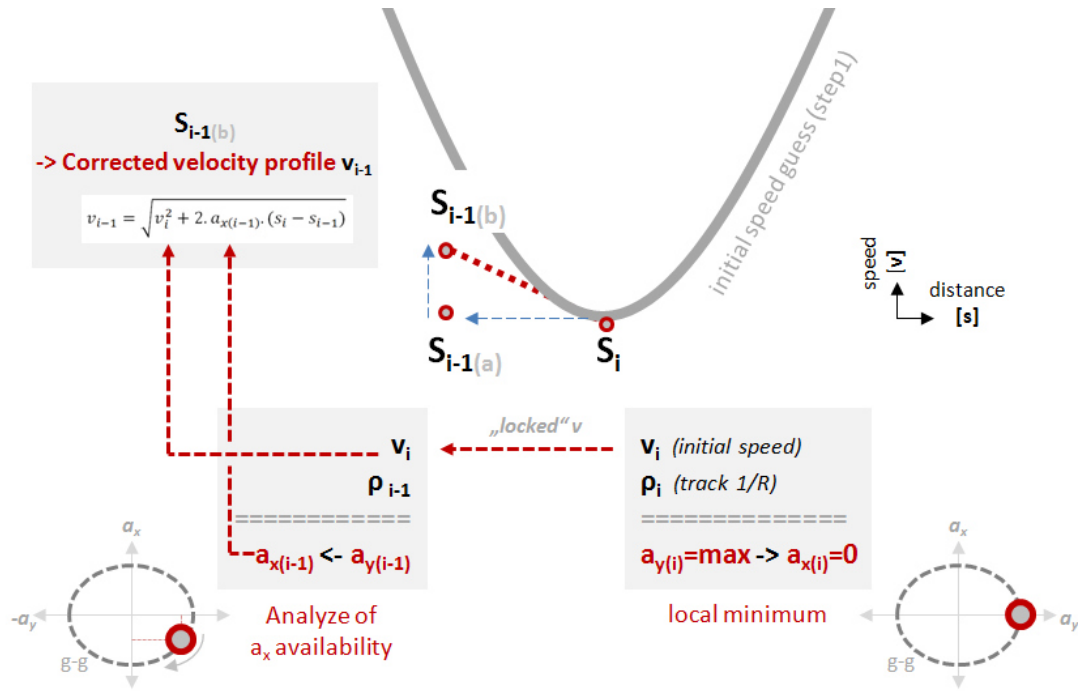


Figure 4.9: Braking Phase (Step 2) - Algorithm Realization

for $i = 1 \dots N$

$$a_{y(i-1)} = v_{step1(i)}^2 \cdot \rho(i) \quad (4.5)$$

$$a_{x_{brake}(i-1)} = \sqrt{\left(1 - \frac{a_{y(i-1)}^2}{a_{y_{max}}^2}\right) \cdot a_{x_{max}}^2} \quad (4.6)^{75}$$

$$v_{(i-1)} = \sqrt{v_{step1(i)}^2 + 2 \cdot a_{x_{brake}(i-1)} \cdot \Delta s} \quad (4.7)$$

Same as in the first step, in Matlab script, the whole calculation process is modified for varying influence of aero down forces. Practically it means that before the whole investigation of particular acceleration capacity a_x for calculation of new speed level $(i-1)$ can be launched, the appropriate g-g diagram in accordance with the last known velocity level has to be loaded. This tactic requires short steps between the track points. The test data have already quite good track density, however, in case of need, an additional routine for track interpolation would be proposed in order to minimize spacing gaps.

⁷⁵ In Equation 4.6 is used the circle approximation instead of ellipse approx. in code

In *fig.4.10* is visualized the corrected speed profile for braking maneuver. To demonstrate the difference, both vehicle concepts are plotted. Because of aero influence on a_x limit, the braking curves are not just displaced horizontally like on *fig.4.8*, but also the curve gradient has changed a bit.

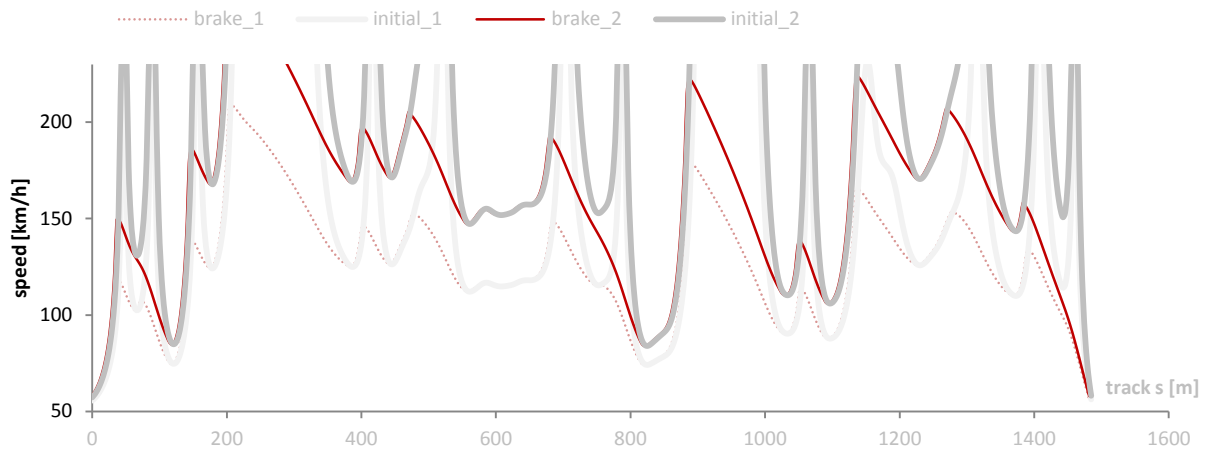


Figure 4.10: Corrected Speed Guess (Braking Phase) – Step 2

Step 3 – Accelerating phase

Correction of initial guess profile for acceleration maneuver is based on the same principal as for the braking maneuver in step 2. The only difference is that instead of backwards calculation, this type of correction is processed by running the algorithm in forward direction.

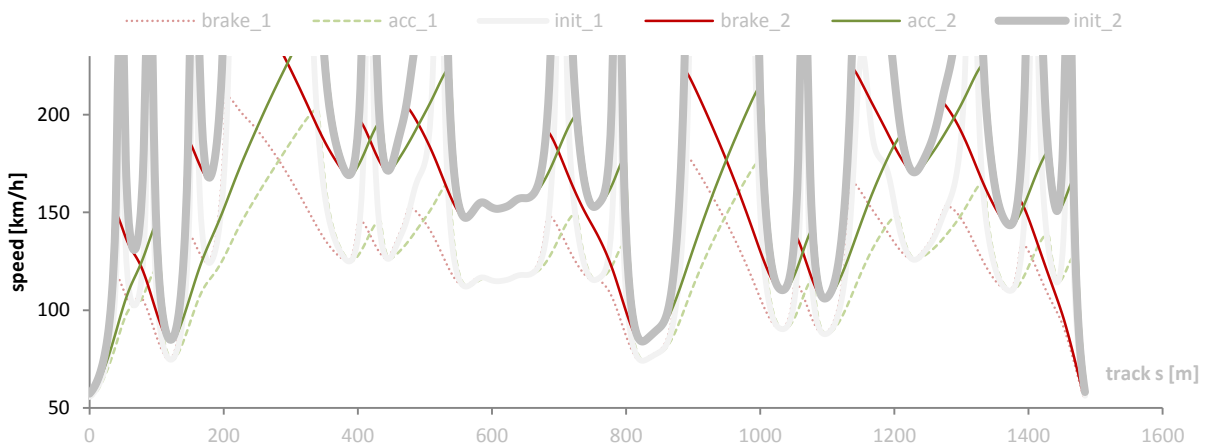


Figure 4.11: Corrected Speed Guess (Braking + Acceleration) – Step 3

Because of adapting the velocity profile to adhesion boundaries without any performance constraints given by power train configuration, the same g-g boundaries can be used. That's why the red and green lines in *fig.4.11* have the identical gradients. The final restriction of forward acceleration capacity will be done later on by Simulink model.

Step 4 – Final corrected velocity profile

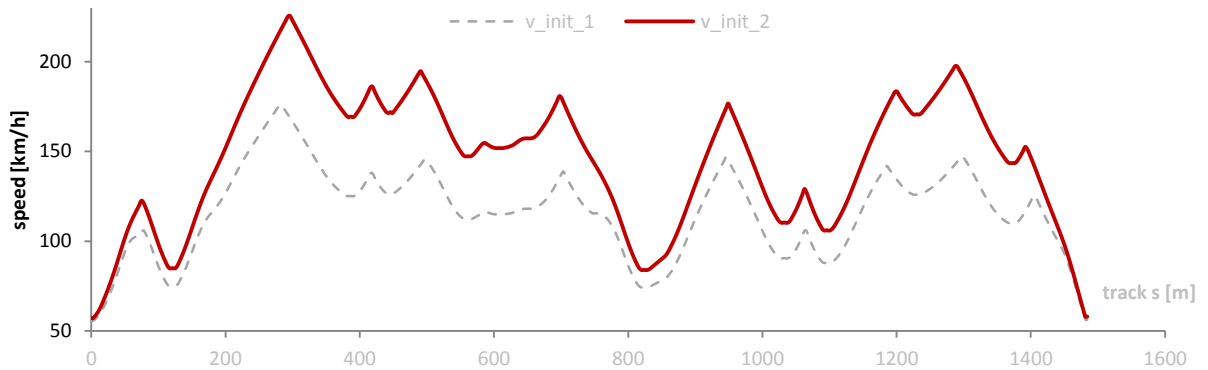


Figure 4.12: Initial (Boundary) Speed Profile – Step 4

The last operation in script is the final connection of particular speed profiles obtained in steps 2 and 3. The final corrected version of initiate speed profile is then shown on *fig.4.12* and illustrates the form of operation area for longitudinal speed controller.

4.1.2.2 Initial velocity profile - functional validation

According to *fig. 4.12*, every track point has its own allowed speed level which guarantees the maximal utilization of friction limits described by the help of g-g diagram boundaries (table 4.1).

Reverse visualization of all calculated operation states (n) along the driven track, plotted against a_x and a_y axis, should restore the original form of g-g limits in the same way as they were defined at the beginning. Regarding two introduced car concepts, both of them will be tested (table 4.1).

The results presented in *fig. 4.13* and *fig.4.14* illustrates the clear difference between these two car concepts. While the civil car (Use Case 1) works exactly within given boundary ellipse (*fig. 4.13*), the sport car (Use Case 2) is effectively using the aerodynamic effects in order to extend the operation area (*fig. 4.14*).

For verification, if the cornering maneuver⁷⁶ was calculated correctly, operating points are highlighted with different colors, depending on the actual speed. Fast passages are represented by points with lighter tones of green and yellow, the slower sections are respectively in dark blue.

Especially in *fig.4.14* with blue tones is good documented, that not only the longitudinal acceleration capacity is during cornering restricted, but also the total lateral capability is lower due to speed decrease. In overall, both diagrams are kind of proof, that developed calculation of initial speed profile works fine and can be implemented into the lap time simulation.

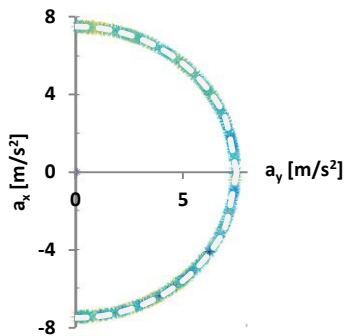


Figure 4.13: G-G Diagram Test (Case 1)

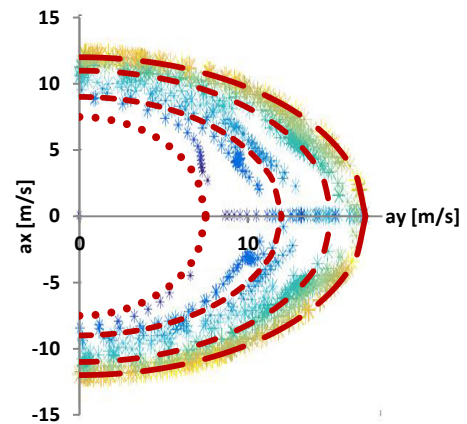


Figure 4.14: G-G Diagram Test (Case 2)

⁷⁶ See Chapter 1.1

4.1.3 Vehicle performance envelope

Calculation strategy of initial speed profile is in chapter 4.1.2 using a bit modified g-g diagram which is basically built on approximation of ellipse defined by maximal lateral and longitudinal vehicle capability. The main modification came from the basic assumption that the initial simulation phase is not taking into account the restrictions given by power limitation. That's why the form of diagram can be assumed as a regular circle or ellipse. (Depends if $a_y = a_x$.)

This part gives a short theoretical review about g-g diagrams, the meaning, construction and finally will be introduced the practical simplification of how the modified g-g diagram for this simulation can be constructed and under what assumptions the theoretical vehicle limit has been examined.

4.1.3.1 Theoretical basis of g-g diagram

Quite idealized concept claims, that tire horizontal force is independent of direction and the maximum on force that the tire can produce is limited by the tire/road friction coefficient times the load on the wheel. It means that graphically it can be represented by friction circle with radius determined by load and friction coefficient.⁷⁷

More accurate friction diagram is obtained from combination of experimentally measured longitudinal and lateral forces for given set of operating conditions (load, surface, temperature, etc.) (See fig. 4.15-6).⁷⁷ The working point in plot is than given by slip angle and traction/braking slip ratio. Depending on the outer form of diagram, the outside envelope can be approximated as circle or ellipse. The boundary envelope is than the most interesting diagram area for race driving.

Basically the friction circle describes only the separate tire behavior. In order to get information about vehicle limits, the whole vehicle configuration with 4 tires have to be summarized into so called g-g diagram describing the overall vehicle capability which is able to take into account the load transfers and other effects.⁷⁷

⁷⁷ Milliken, Race Car Vehicle Dynamics, chapt.9

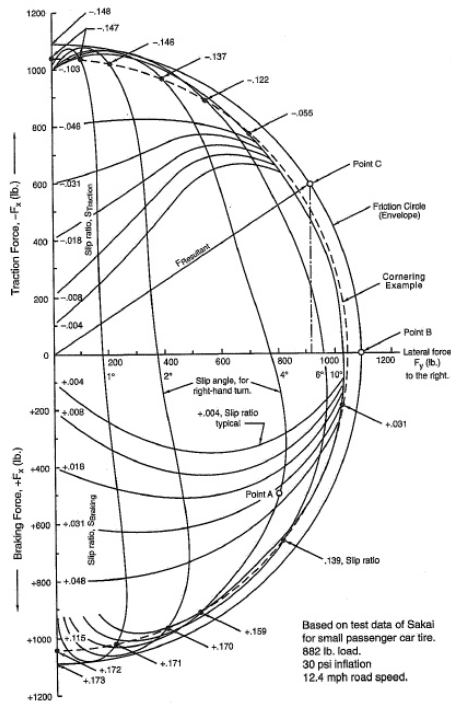


Figure 4.15: Friction Circle Diagram ⁷⁷

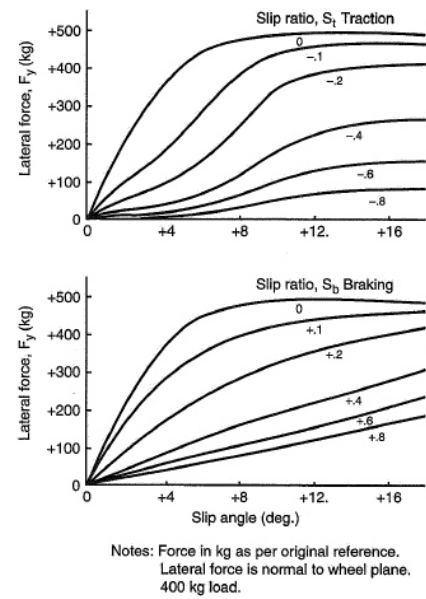


Figure 4.16: Lateral Force (slip angle/ratio) ⁷⁷

There are two variables that affect the g-g boundaries at most - the aero down force and drag. Both of them are speed dependent and directly related to vehicle aerodynamic design. As speed increase, the aero down force improve cornering while drag reduce forward acceleration but improves braking. Finally, the g-g diagram can be plotted as three dimensional figure where the third axis is speed⁷⁷. Basically, there are two options how to get real vehicle performance envelope (g-g boundaries) – by practical examination or from theoretical calculation model.

In Milliken⁷⁷ is outlined the basic method of accurately calculation the g-g boundary for real vehicles. It's based on Moment Method simulation where the steady-state cornering maneuver while accelerating or braking is simulated. The same steady-state maneuver is performed at a constant speed for different levels of drive torque or brake line pressure. Individual simulation steps are finally plotted against longitudinal and lateral acceleration axis in form of series of basically horizontal lines representing specific vehicle capability for given drive torque or brake pressure. The overall results for RWD Car of such constructed g-g diagram are showed in *fig. 4.17*. – This particular diagram is constructed for specific speed level.

⁷⁷ Milliken, Race Car Vehicle Dynamics, chapt.9

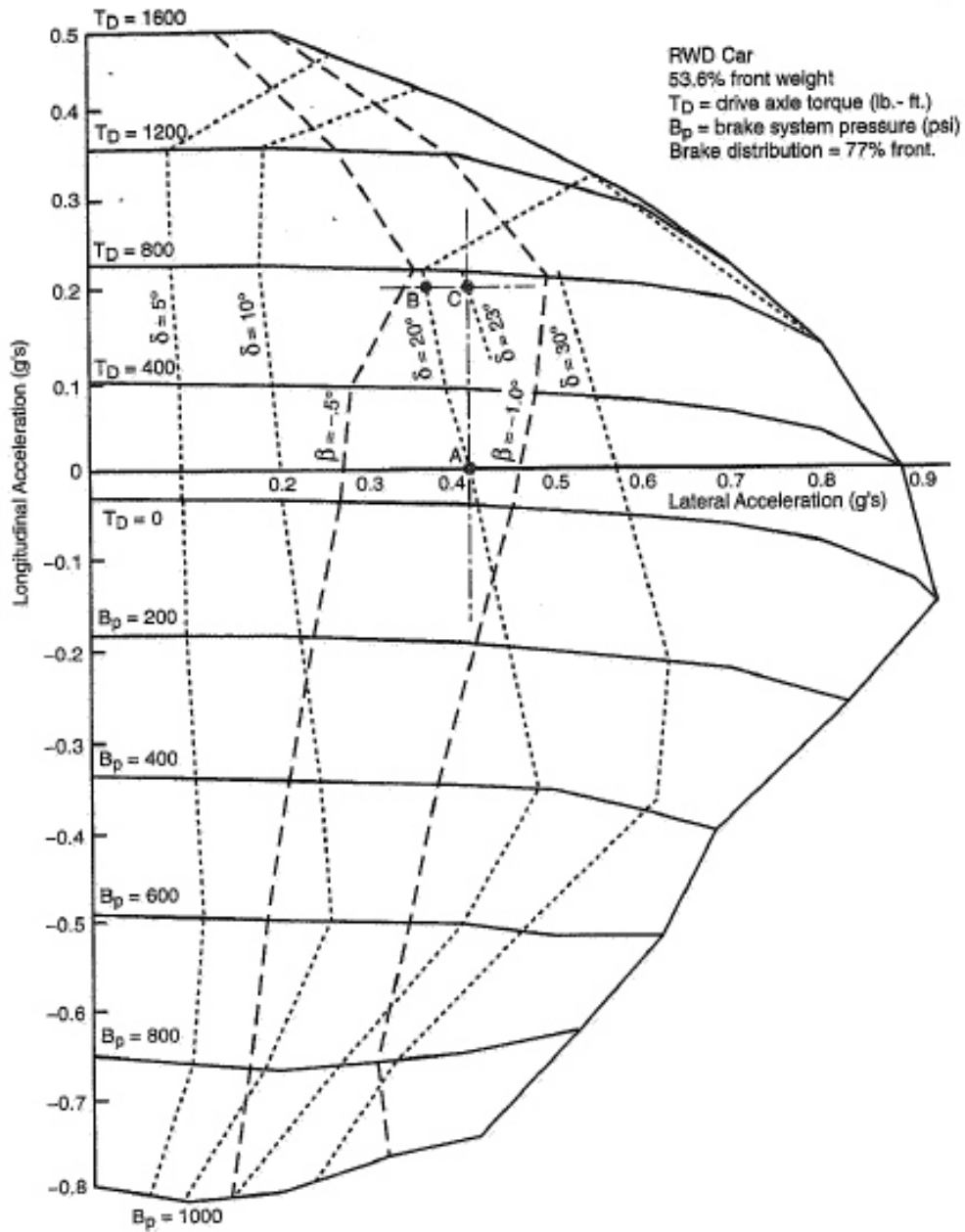


Figure 4.17: G-G Diagram, Rear-Wheel Drive ⁷⁷

⁷⁷ Milliken, Race Car Vehicle Dynamics, chapt.9

4.1.3.2 Idea of g-g diagram modification

Regarding the short Recherche, it's obvious that construction of real g-g diagram costs quite a lot calculation time and requires huge complexity of vehicle model to take into account maximal number of factors as traction limitations, load transfer effects, suspension effects, brake balance, etc.

Already at the beginning, during the first conceptual phase, it was clear, that the very first release of lap time simulator must be based on comprehensive background. That's the reason why the calculation strategy with a simplified g-g concept has been initiated. On this basis, it was decided to create a fast routine evaluating the vehicle adhesion capability in corners.

For simplified vehicle performance envelope, the following assumptions are formulated:

Vehicle model:

The whole vehicle is summarized on level of single-track model respecting:

- Static weight distribution
- Implementation of aero down forces
- Non-linear description of tire behavior

Evaluating strategy of performance envelope:

Maximal lateral tire-capability is, according to finding of chapter 2.2.1, one of the most important parameters influencing the racing time. The introduced single-track model is intended to provide simple simulations of steady-state cornering maneuver in order to get maximal value of lateral acceleration for different speed levels.

On the basis of friction circle approach, the whole performance envelope will be also assumed as a circle with radius defined by maximal lateral capability for particular speed level. Because of neglecting the traction forces in single-track model, a bit optimistic results can be expected – see the $T_d=0$ in *fig.4.17*. (While cornering increase the roll resistance)

4.1.3.3 Vehicle performance envelope - practical realization:

The complete simulation stack can be found on CD in `..\Program\Tools-GGV_tester\Lateral_dynamics.m`. On this place, it will be outlined just basic relations declaring the range of model complexity.

Single-Track model:

The equations of motion for single-track model:

$$m(\dot{v} + u \cdot r) = F_{y1} + F_{y2} \quad (4.8)$$

$$J_z \cdot \dot{r} = a \cdot F_{y1} - b \cdot F_{y2} \quad (4.9)$$

=====

Where the horizontal forces $F_{y1,2}$ are function of:

$$\text{Slip angle } \alpha_{1,2}: \quad \alpha_1 = \delta - \frac{v+a \cdot r}{u} \quad \& \quad \alpha_2 = -\frac{v-b \cdot r}{u} \quad (4.10-1)$$

$$\text{Normal force } F_{z1,2_stat}: \quad F_{z1_stat} = m \cdot g \cdot \frac{b}{l} \quad \& \quad F_{z2_stat} = m \cdot g \cdot \frac{a}{l} \quad (4.12-3)$$

$$\text{Aero down force } F_{z1,2_aero}: \quad F_{z1_aero} = A \cdot c_{a1} \cdot \frac{\rho}{2} \cdot u^2 \quad \& \quad F_{z2_aero} = A \cdot c_{a2} \cdot \frac{\rho}{2} \cdot u^2 \quad (4.14-5)$$

Aero down force F_{z12_aero}

The influence of vehicle aerodynamics is characterized by changing the normal load F_z . Eq. 4.14-5 shows the additional load force on axle (front and rear) caused by velocity increase. The magnitude and direction of this additional load is characterized by down force coefficient $c_{a(1,2)}$. The positive coefficient respectively positive force F_{Zaero} means opposite z-direction in vehicle coordinate system defined by ISO 8855:2011. The typical values of c_a coefficient: $c_a=0.2$ (civil cars); $c_a=0$ (sport car); $c_a=-1$ (DTM); $c_a=-1.5$ (F1).⁷⁹

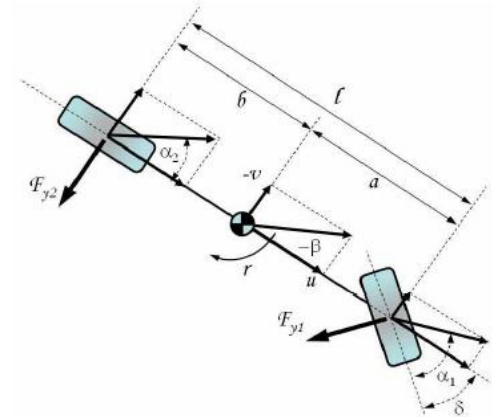


Figure 4.18: Single-Track Model ⁷⁸

⁷⁸ J.P. Pauwelussen, Vehicle modelling and behaviour assessment, HAN University (2010)

⁷⁹ T. Schütz, Hucho - Aerodynamic des Automobils, (2013)

Horizontal forces F_{Y12}

Because of specific non-linear tire behavior, for simulation of cornering maneuver, is necessary to use a non-linear tire model reflecting the variables as load function and slip angle. In introduced single track model was implemented PAC2002 Tire model working on base of Pacejka's Magic Formula⁸⁰. All equations and model parameters are listed in Appendix 3.

Simulation of cornering maneuver:

The cornering maneuver for estimation of maximal lateral acceleration a_{y_max} has been inspired by NHTSA's routine called SIS (Slowly Increasing Steer) maneuver⁸¹. While originally is test defined by SAE J266 with constant speed and variable steer, for this case it was slightly modified to match specific criteria of lap time simulation – the gray line.

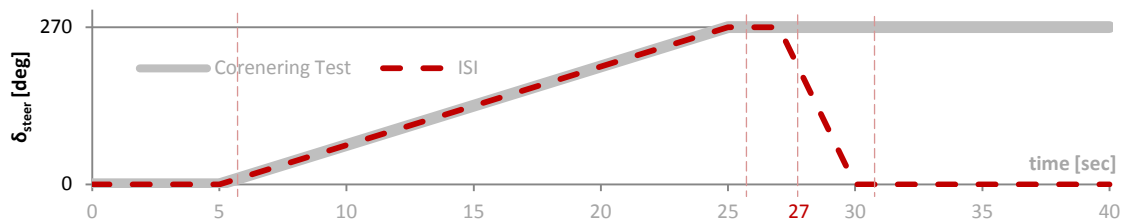


Figure 4.19: Test Maneuver

Every testing loop is predefined by specific test speed that must be during the test hold. Increasing steer angle δ induce subsequently the slip angle and under specific load F_z is finally produced the side force F_y responsible for reached lateral acceleration $a_{y(v)}$. Final GG diagram, constructed under assumption of friction circle approach, is then showed on *fig. 4.20 - right*.

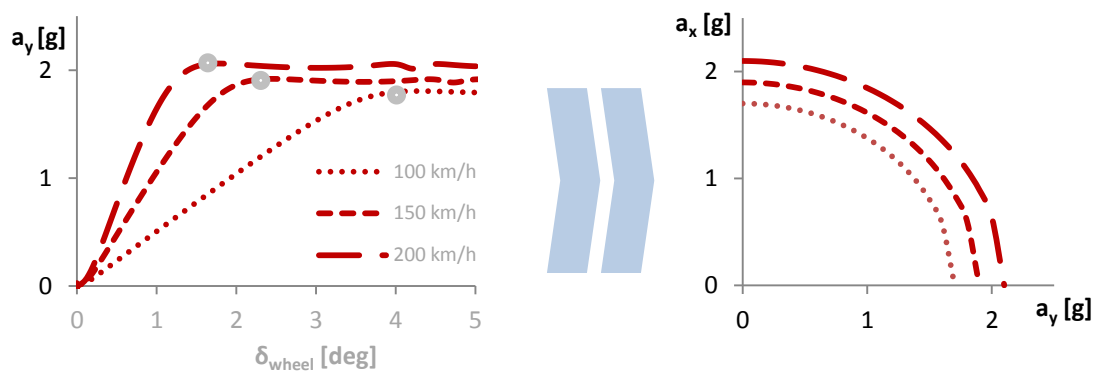


Figure 4.20: Modified G-G Diagram

⁸⁰ Pacejka, Tire and Vehicle Dynamics (2002)

⁸¹ source: <http://www.nhtsa.gov/DOT/NHTSA/NRD/Multimedia/PDFs/VRTC/ca/capubs/PhaseIVRollover03SAE.pdf>

4.2 Simulation model – Stage II

The main aim of “Stage Two” is to calculate by the help of model based simulation the final racing time for the track analyzed in Stage I and at the same time to perform an additional estimation of consumed fuel during the simulated drive on the track.

Regarding the simulation flow presented in *fig. 4.21*, the power-train model is controlled by output dataset coming from Stage I. As mentioned before, the control strategy is based on speed adjusting, whereas the calculated initial speed profile is compared as a reference value to actual speed value reflecting the dynamical capabilities of power train model. On basis of measured difference in closed-loop system, the controller block sends an appropriate control signal to power train with intention to adjust the actual speed level towards the reference level.

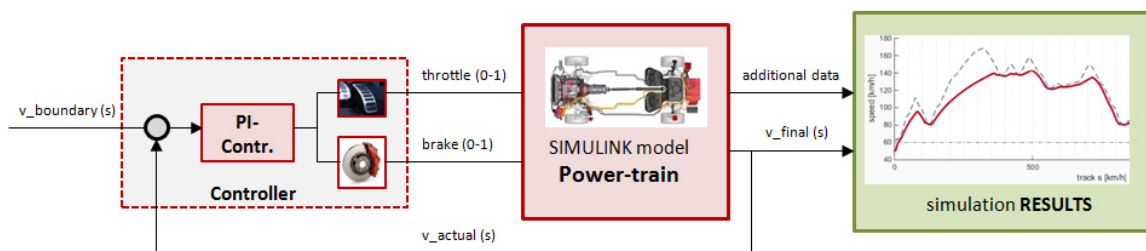


Figure 4.21: Simulation Model – Stage II

In the next text,⁸² at first, the power train model and additional functionalities working as subsystems with main engine model are introduced. Knowing the functions of all elements, the complex control strategy is going to be implemented and finally, with completed model will be possible to perform first tests for proving the system functionality.

⁸² Structure of following chapter 4.2:

4.2.1 Power train model

- 4.2.1.1 - Control strategy for power train model
- 4.2.1.2 - Gearbox

4.2.2 Fuel consumption analysis

- 4.2.2.1 - Functional validation

4.2.3 Race controller strategy

- 4.2.3.1 - Inputs/Outputs

4.2.1 Power train model

Simulation of power train is built as a rigid model of longitudinal dynamic standing on principal that moment value is sequentially forwarded over the given path connecting the individual blocks in Simulink model (fig.4.22). The end-value of traction torque comes from equilibrium with all driving resistances referred in fig.4.24. The basic formula for calculating the total power train force F_{reg} required to maintain a vehicle's state of motion is given in Equation 4.16 below:⁸³

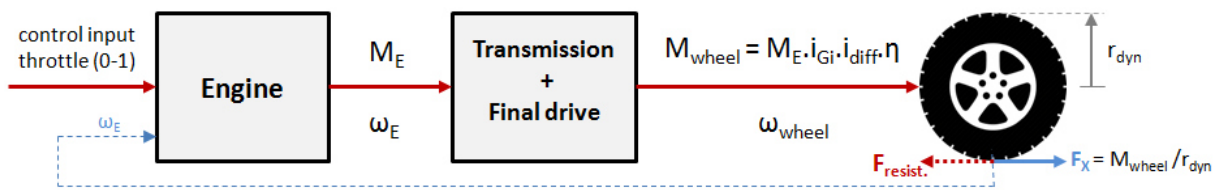


Figure 4.22: Simulink Model⁸⁴

$$F_{Reg} = F_R + F_C + F_A + F_I \quad (4.16)$$

The sum consist of the total rolling resistance F_R of all four wheels, the climbing resistance F_C , the air resistance F_A and the inertial resistance F_I . The same equation is written in itemized form below:

$$F_{Reg} = k_R \cdot F_z + m_v \cdot g \cdot \sin(\alpha_{sl}) + c_w \cdot A \cdot \frac{\rho}{2} \cdot v^2 + (e_i \cdot m_v + m_{load}) \cdot a_x \quad (4.17)$$

Where the last term F_I is stated in short form using the mass factor e_i . However, the simulation model works with traditional form (eq. 4.18) respecting the complete dynamical system on fig. 4.23 which is further simplified by defining the reduced rotational inertia J_{red} for engaged gear i_g .

$$F_I = F_{I,trans} + F_{I,rot} = \left(m_v + \frac{J_{red,i_g}}{r_{dyn}^2} \right) \cdot a_x \quad (4.18)$$

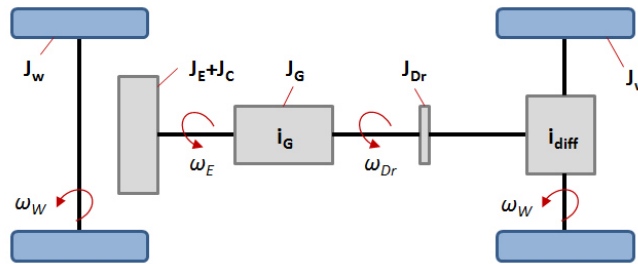
$$J_{red,i_g} = 4 \cdot J_W + i_{diff}^2 \cdot J_{Dr} + i_{g(i)}^2 \cdot i_{diff}^2 \cdot (J_E + J_C + J_G) \quad (4.19)$$

Additionally, the mass factor e_i can be for equation 4.17 defined in following way:

$$e_i = \frac{J_{red,i_g}}{m_v \cdot r_{dyn}^2} + 1 \quad (4.20)$$

⁸³ B. Heißing, Chassis Handbook, (2011), S.49

⁸⁴ H.E. Scherf, Modellbildung und Simulation dynamischer Systeme, (2007), S.33

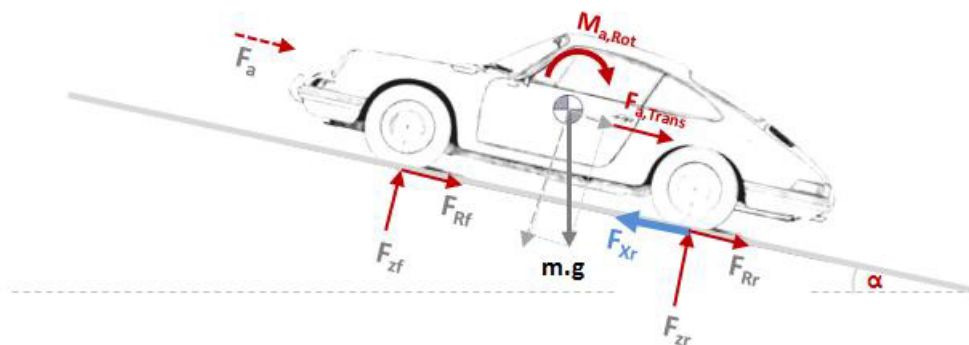
Figure 4.23: Power Train Model ⁸⁵

The simulation of longitudinal dynamics with simulink is based on methods of numerical solvers that integrate the listed differential equation of motion (eq. 4.16) in specified step-size time. In order to get desired output signal, for instance the speed, the equations have to be in simulink model formulated in following way (eq. 4.21-3) to allow the solver start the appropriate integration routine.

$$a_x = \dot{s}_x = \frac{1}{m_{red}} * (F_{Reg} - F_r - F_A - F_T) \quad (4.21)$$

$$a_x = \dot{s}_x = \frac{1}{m_{red}} * (F_{Reg} - F_r - F_A - F_T) \quad (4.22)$$

$$v_x = \dot{s}_x = \int \frac{1}{m_{red}} * (F_{Reg} - F_r - F_A - F_T) . dt \quad (4.23)$$

Figure 4.24: Driving Resistances ⁸⁶

4.2.1.1 Power train model - control strategy:

The simulated system is in general controlled with throttle input scaled as a value within the interval range $\langle 0;1 \rangle$. In settings is possible to restrict the upper range limit in order to simulate also moderate drive strategies for evaluation of fuel consumption. (more about fuel consumption in chapter 4.2.3) The look-up table of throttle map interpolates

⁸⁵ Mitschke, Wallentowitz, Dynamik der Kraftfahrzeuge, (2014), S.80

⁸⁶ Ecksein L., IKA RWTH, Vorlesungsumdruck Fahrzeugtechnik I., (2012)

then the actual torque request for specific combination of throttle input and actual engine speed. Because of intention to use particular power train model for different vehicles and power train configurations, the specific control strategies, for instance as gear shifting, have been not fully detailed in order to keep the high level of model universality.

4.2.1.2 Power train model – Gearbox:

The gearbox logic is therefore to be simulated in very efficient way by the help of state-flow diagrams. State-flow is special feature that enables to implement the action logic in Simulink model. On this basis, the shifting procedure is in this case modeled only as a pure parameter change – no mechanical interactions or additional components typical for gear change are modeled. On *fig.4.25* is presented the state-flow chart defining the simple shifting strategy for sample 3-speed gearbox.

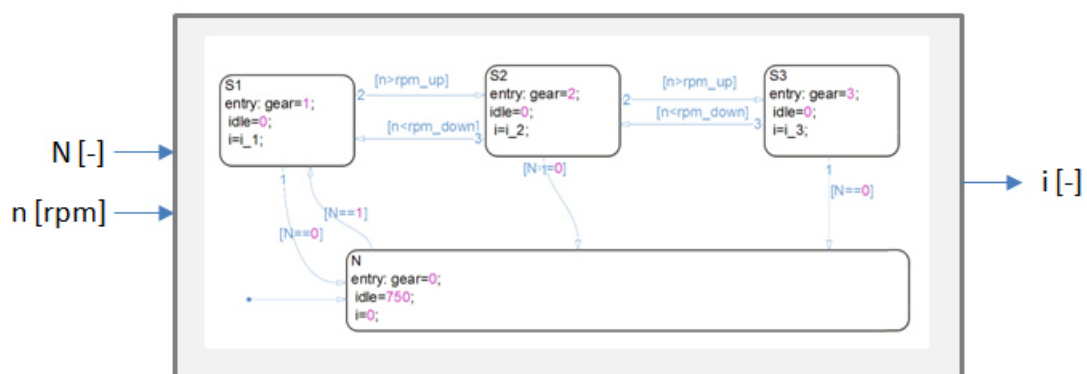


Figure 4.25: Stateflow Gearbox

In this case, the state chart is a separate Simulink block with own input and output ports. The blocks inside the diagram represent individual states (*S1, S2, ...*) respectively the speed gears of gearbox. Depending on actual state, the corresponding value of chosen gear ratio appears on output *i*. Every junction between two blocks is signed with a rule. When the given condition is fulfilled, new gear can be engaged. Control input *N* is there just for switching between drive and idle state.

Control strategy for gearbox in particular case is based on pre-defined shifting points initiating the gear ratio change. In GUI⁸⁷ will be possible to define the shifting RPMs manually.

⁸⁷ See Chapter 6

4.2.2 Fuel consumption analysis

Tool evaluating the fuel consumption is very close connected to power train model introduced in chapter 4.2.1. Principally, there is a so-called BSFC map evaluating the specific fuel consumption for particular engine. Knowing the actual working point of engine (n, p_e) , it's possible to estimate actual specific fuel consumption b_e in g/kWh from given diagram. At the end, the integral of all state points over the whole time interval calculates the total fuel consumption.

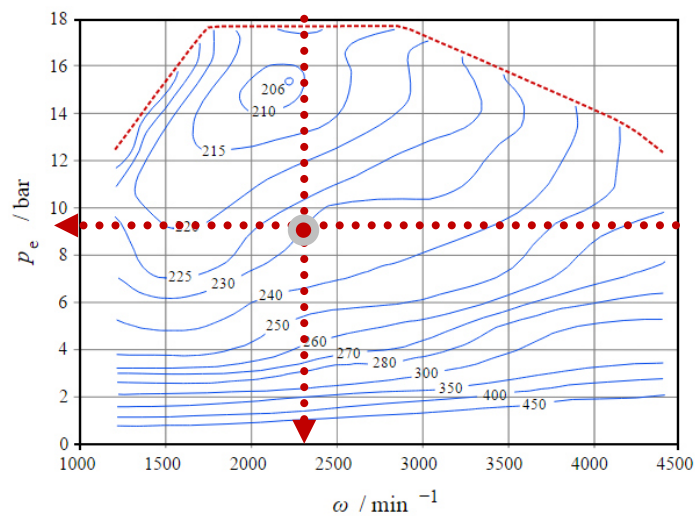


Figure 4.26: BSFC Map⁸⁸

In general, the consumption analysis is working in following steps:

a, identification of working points $\{p_e, n\} \rightarrow b_e [g/kWh]$

Basically both values estimating the working point of engine are possible to calculate with output signals from power train model. Equation 4.24 gives a brief insight into how to reach the effective pressure p_e (MEP).⁸⁹

$$p_{me} = \frac{P_e}{i \cdot n \cdot V_H} [bar], \text{ where } i = 0.5 \text{ (4 stroke engines), } V_H = \text{engine displacement} \quad (4.24)$$

b, track based consumption B_e

In equation 4.25 is sketched the basic simulation workflow of consumed fuel on track.

$$B_e = \frac{\int b_e \cdot P_e \cdot dt}{\int v \cdot dt} [g/m] \quad (4.25)$$

⁸⁸ source: https://en.wikipedia.org/wiki/Brake_specific_fuel_consumption

⁸⁹ Ecksein L., IKA RWTH, Vorlesungsumdruck Fahrzeugtechnik I., (2012)

4.2.2.1 Fuel consumption analysis - functional validation

Function of proposed simulation chain, where the power train model is linked directly with subsystem evaluating the fuel consumption, was tested on real vehicle in various traffic situations. During the test drives, all relevant data have been logged through the connected OBDII dongle, subsequently the simulink power train model was fed with measured velocity profile additionally coupled with RPM datasets in order not to lose the information about engaged gear.

Simulated fuel consumption values were compared against to real values calculated from refilled amount of petrol. On this basis, the following corrections and modifications in calculation subsystem have been implemented:

- *Restriction of negative values of P_e caused during the deceleration (braking force)*
 - Corrected the error of temporally increasing the fuel level in tank
- *Identification of idling states*
 - Switch on predefined idle consumption
- *Identification of engine braking*
 - Consumption is zero

Performed validation was tested on Škoda Octavia 1U with gasoline engine 1.6mpi (BFQ). On *fig.4.27* is showed a sample test cycle in moderate city traffic. The calculated fuel consumption in simulation is 6.4l/100km, the real one about 6.5l/100km.

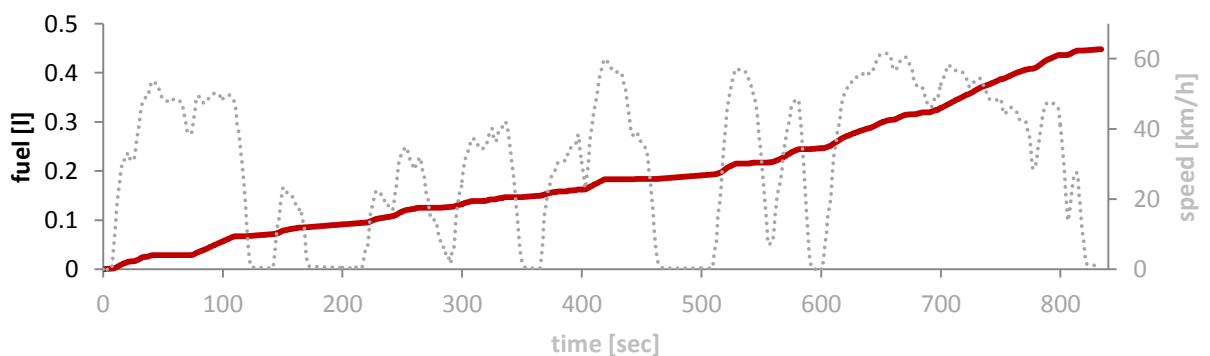


Figure 4.27: Fuel Consumption, Simulation

4.2.3 Race controller strategy

The proposed control strategy of overall simulation layer has already been mentioned and theoretically discussed several times – see chapter 3 or 4. After knowing the basic strategy behind this closed-loop concept, the sketched workflow on *fig.4.28* tries to clarify all relations between pre-calculated initial solution of $v_boundary$ (initial speed) on the left side and model based power train simulation on the right side.

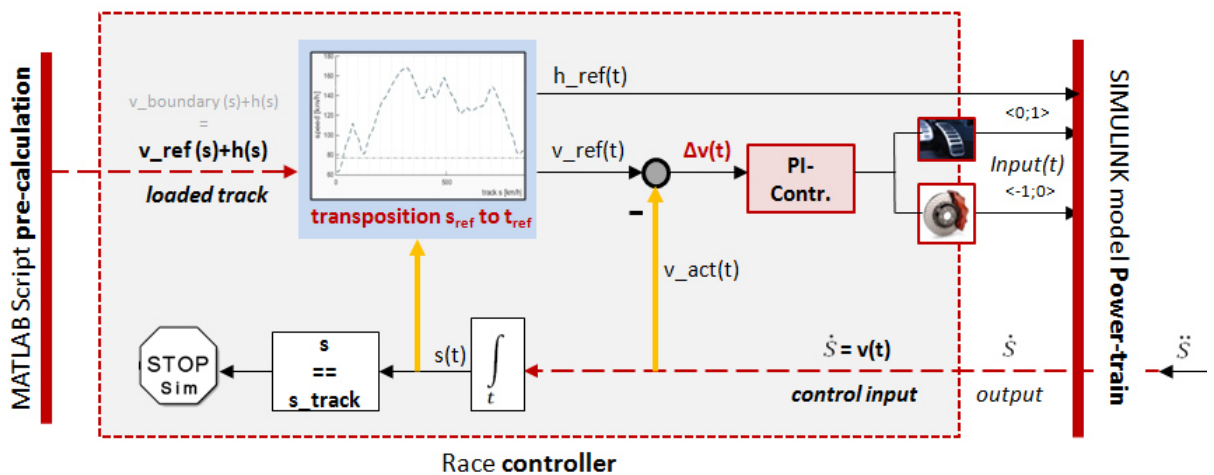


Figure 4.28: Race Controller, Realization

Because of fact that whole simulation is basically driven by signed integration time,⁹⁰ the real control principle must have been transposed into another reference value to ensure that controller works with right couples of signed references (v_ref) and feedback (v_act) values. Parallel to original control time-line (t) with a regular step size given by solver clock, there is a control track line(s) where the step size is function of actual velocity (v_act). Both lines are highlighted in orange.

4.2.3.1 Race controller strategy – Input/Outputs

Input – xyz respectively the initial velocity profile

As it was mentioned, the controller input for race controller represents the analyzed track. The initial velocity profile contains for this application already enough information, to estimate the velocity profile on the track.

⁹⁰ See Chapter 3.1.3

Output - horizontal profile h

Additional input for simulink model is a kind of disturbance in form of horizontal profile which is again loaded in form of $h(s)$ originally coupled with xy coordinates but for control output, it must be again transposed into the time-referenced value $h(t)$. Power train model then contains special block for conversion of horizontal difference Δh into climbing angle α respectively the appropriate resistance force as sinus function of vehicle load noted as F_C .

Output - throttle/brake

Control outputs respectively the control inputs for power train model were already mentioned in corresponding chapter 4.2.1.1. Basically the controller outputs are signal values within interval $\langle -1; 1 \rangle$, where the positive values represent throttle inputs and the negative values are then responsible for braking. Actually the brakes are also part of longitudinal dynamic model in simulink, modeled as linear braking force element scaled on basis of control input.

The applied controller itself is based on proportional and integral controller term (PI Controller). The working principle of Race Controller will be evaluated during the validation phase in the next chapter.

5 Validation

According to V-Model⁹¹ methodology, the testing phase is started with the first running release. Although the single components of simulation model have been already during development more or less tested,⁹² the main aim of this section is the complex validation of simulation layer with proposed race controller. The comparison with real measured data gives the important feedback for identification of potential problem zones in simulation model. The whole evaluation will be performed exactly, as the simulation process, in two phases. At first, the pre-calculated speed boundaries and then the simulated speed profile will be compared with the real measured data from the track. In total, two cars on two different tracks have been tested.

5.1 Test no.1

Vehicle: Porsche Cayenne Turbo V8

Track: Test Track in Nardò (Lap Time: 2.40min)

Unfortunately, there were no relevant data regarding to tires or aero lift coefficients for this car. Thus, it was decided to tune the simulation input (modified g-g diagram) exactly from acceleration data recorded by telemetry.⁹³

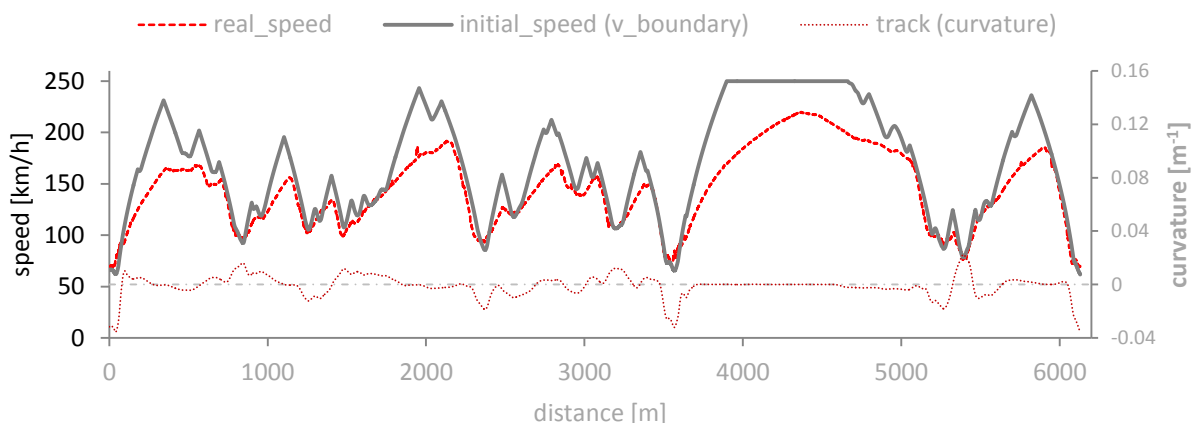


Figure 5.1: Initial Speed Profile (Porsche)

⁹¹ See: Chapter 3

⁹² See: 4.1.2.2 (initial speed profile), 4.1.3.3 (performance envelope), 4.2.2.1 (Fuel Consumption)

⁹³ See chapter 5.2, fig.5.7

The result of pre-calculated initial speed profile in *figure 5.1* seems to be very optimistic. The initial profile should build boundaries along the real speed profile, what the script really does. The only problematic area is the high-speed section with a low curvature – for instance, in the region around 5000m distance, the real car starts to slow-down quite early and in relatively smooth fashion, which raises the question whether the driver has really reached the vehicle limit.

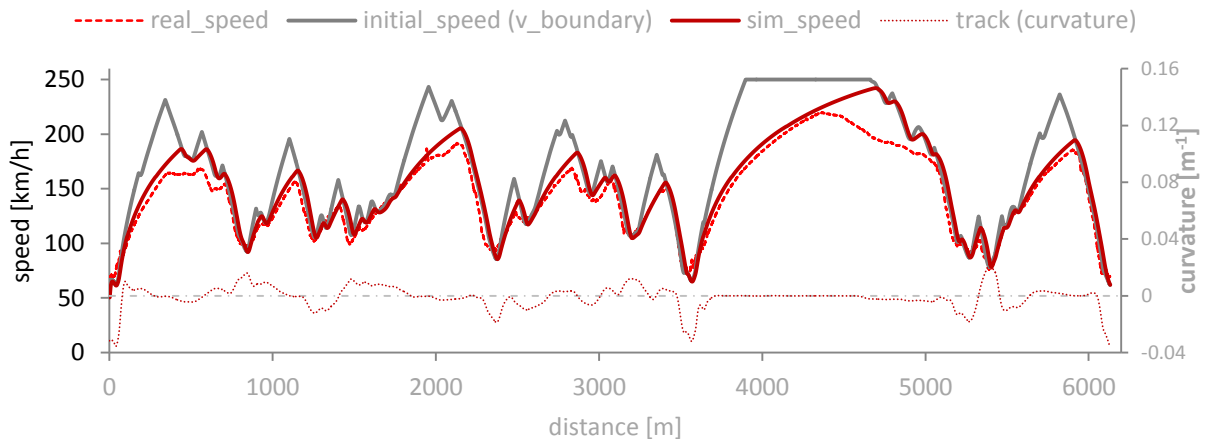


Figure 5.2: Simulation Results (Porsche)

Figure 5.2 then shows the final calculated speed profile with final time *2.36min*. Simulated acceleration curves are in compare to reality quite the same, only the above-mentioned high-speed sections converge not exactly. On the basis of detailed analysis of the speed profile (*See fig.5.3*), it was found out, that the braking curves tend to be delayed. The delay on *fig.5.3* seems actually not significantly, but on the other hand, looking back at *fig.5.2*, the real driver has logically the opposite tendency – to brake earlier.

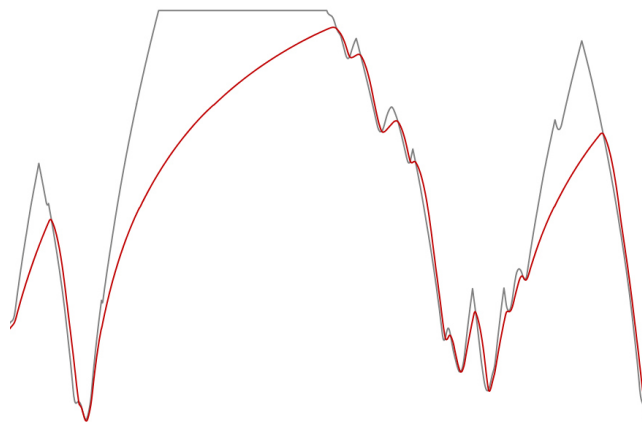


Figure 5.3: Detailed View of final Simulation Profiles

Hence, it was proposed a slight improvement of race controller with additional option of moving the braking boundary according to which (distance in meters)⁹⁴ a bit forward, exactly as shown on *fig.5.4*. It seems to be relatively efficient way how to initiate the braking maneuver also in simulation a bit earlier. The estimated time has been changed to *2.41min*.

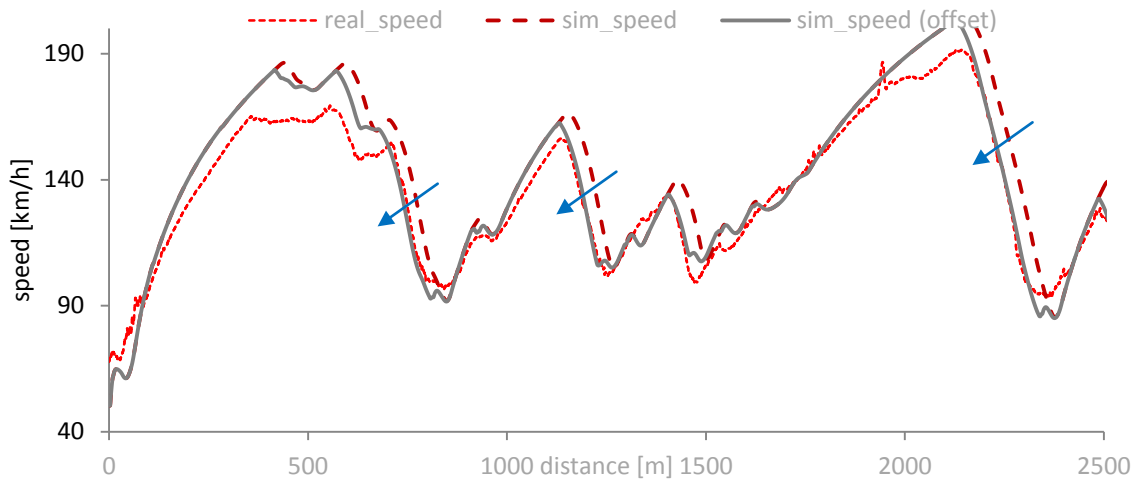


Figure 5.4: Adjusting the Braking Boundary

Finally, on the basis of simulated vehicle states, the g-g diagram has been visualized in order to prove, that vehicle system maintains within the ellipsoidal area and the model is able to utilize the whole performance envelope.

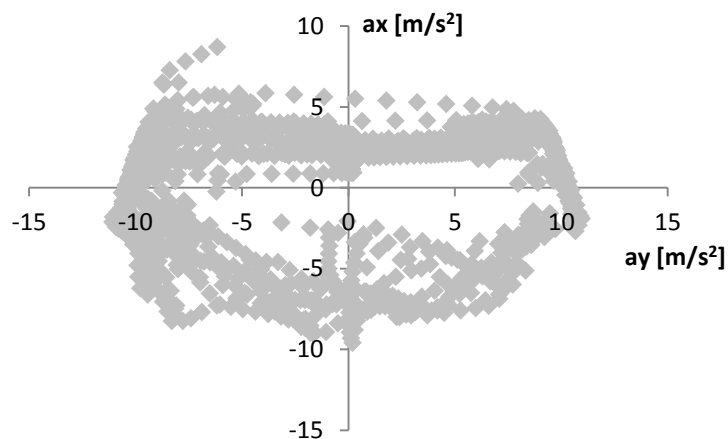


Figure 5.5: G-G Diagram of simulated Race Lap

⁹⁴ See: Chapter 6.4

5.2 Test no.2

Vehicle: BMW M3 3.2L (e36)

Track: EuroSpeedway Lausitz (Lap Time: 2:12.9min)

The speed diagram on *fig.5.6* for second car has been calculated in the same way as in the first case. The diagram on *fig.5.7* and data in Table 5.1 show the general strategy, how to obtain from telemetry log the speed-depended values of lateral accelerations. As it can be seen, the lateral capacity has a decreasing tendency – it’s a typical effect among normal civil cars that are producing no real aero down force. The lift coefficient (C_a) of tested BMW saloon car (e36) is about 0.2 for rear as well for front axis.⁹⁵ It means that at higher speeds, the car is getting lighter, and thus the maximal lateral capacity is reduced.⁹⁶ Data for column with longitudinal accelerations in Table 5.1 are then investigated in the same way.

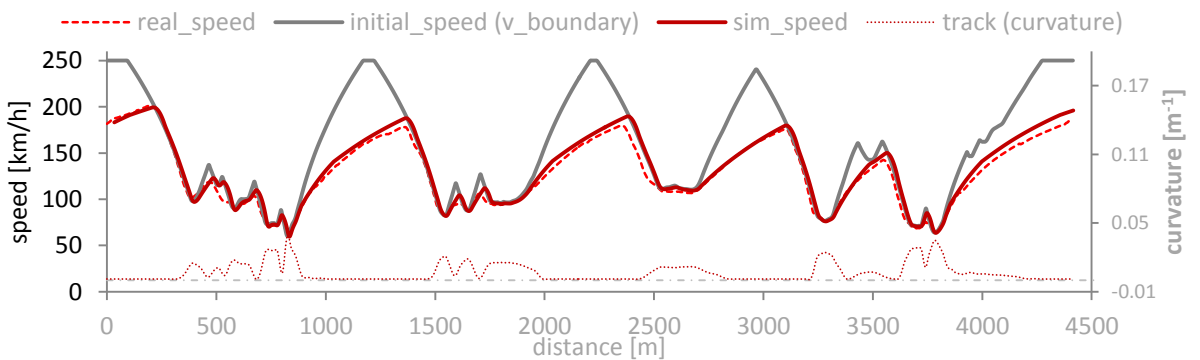


Figure 5.6: Simulation Results (BMW)

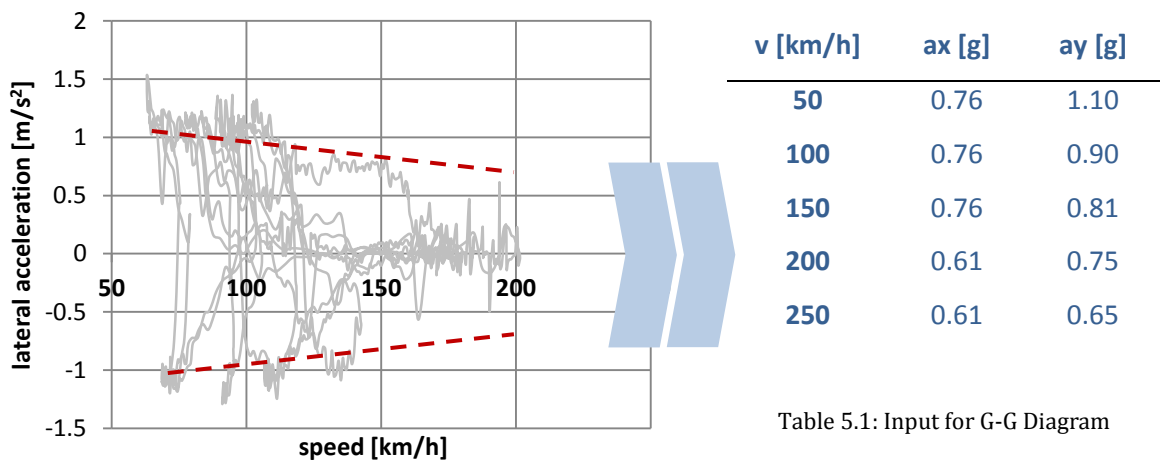


Table 5.1: Input for G-G Diagram

Figure 5.7: Lateral Acceleration vs. Speed Level

⁹⁵ T. Schütz, Hucho - Aerodynamic des Automobils, (2013)

⁹⁶ See: 4.1.3.3, equation 4.14-5

The same track test was performed once more again, but in second run with the performance envelope defined by test routine proposed in chapter 3.1.3 and based on the idealized strategy, where the maximal lateral vehicle capacity is simulated by the help of single-track model. According to Table 5.1, five speed levels have been tested on maximal lateral capability in order to create the performance envelope for initial calculation phase.⁹⁷ For particular test, there is applied a non-linear model for lateral forces based on Magic Tire Formula with parameters representing the civil tire with dimensions *205/55 R16*.⁹⁸

With simplified single-track model, it has been achieved significantly higher lateral performance operating in range from $1.3g$ to $1.5g$, which also explains the over-optimistic results of lap time simulation (*fig. 5.8*).

Basically, there were identified two phenomena responsible for too high values of lateral accelerations:

- *Neglected load transfers*

This comes from nature of single-track model where the centre of gravity is placed in zero height.

- *No traction forces in single-track model*

In a matter of fact, during cornering maneuver it comes to higher deformation of tire which causes also higher rolling resistance. More power is needed in order to keep car rolling with constant speed. This state of combined slip should be also reflected in simulation model to reach more realistic results.

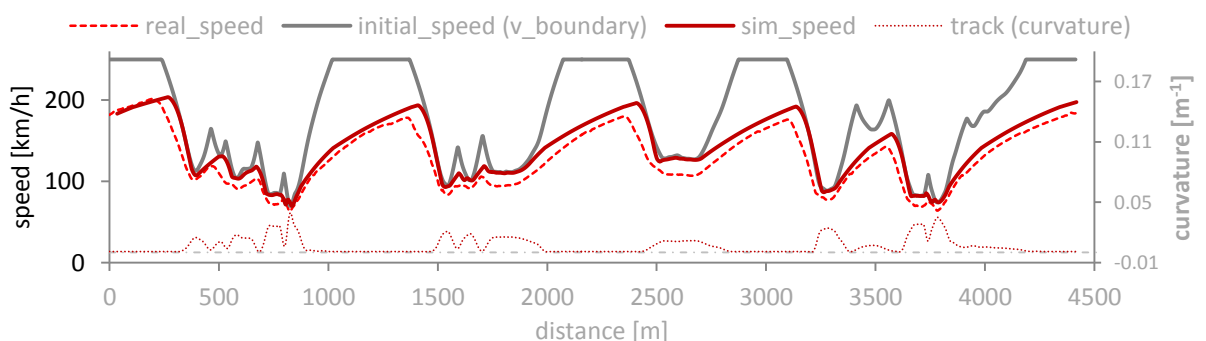


Figure 5.8: Simulation Results with pre-calculated Performance Envelope

⁹⁷ T. Schütz, Hucho - Aerodynamic des Automobils, (2013)

⁹⁸ See: Appendix 3

6 Graphical User Interface (GUI)

The last-but-one chapter of this work will introduce the graphical interface of designed simulation tool. As it can be seen from complex simulation flow on *fig.3.4*, the proposed program is relatively wide and offers quite a large number of possible set-up options. For end-user who didn't participate on developer phase, it would be nearly impossible to get familiar with all functions and set-up options in relatively short time. That's why the GUI was created - to reduce system complexity and enhance so the user comfort. Basically, the GUI connects all simulation parts noted in simulation model and on-click is able to perform the whole calculation. The program can be started directly from Matlab Command line - once the *Laptime_Simulator.m* file is loaded,⁹⁹ the main program window should be displayed. The GUI was developed and tested in Matlab R2014b.

The general idea of UI design follows the main workflow of simulation. This means that on the left side, all parameters are stored, while the space on right is used for presenting the results. In next sections, the complete GUI will be presented step-by-step on an example.

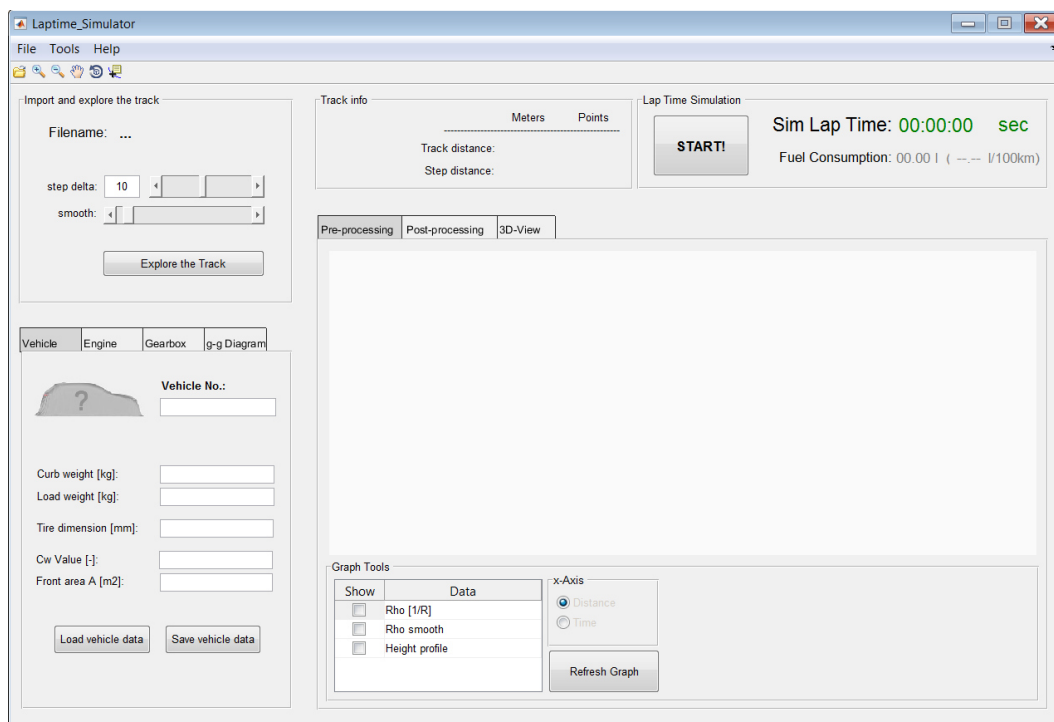


Figure 6.1: Main Program Window

⁹⁹ File path on CD: ..\Program\LapTime_Simulator\Laptime_Simulator.m

6.1 Loading track (trajectory) file

The track can be loaded in very simple way either by using a menu command: **File – Import Track File – Import *.mat** or just by clicking on the first icon in toolbox menu.

Both actions call up the File Explorer from where the particular track file can be chosen. The Track File must be in format .mat containing all three components – x,y,z. It's also possible to load the track from .xls file – the main condition: x,y,z coordinates in excel sheet are defined as the first, second and third column.

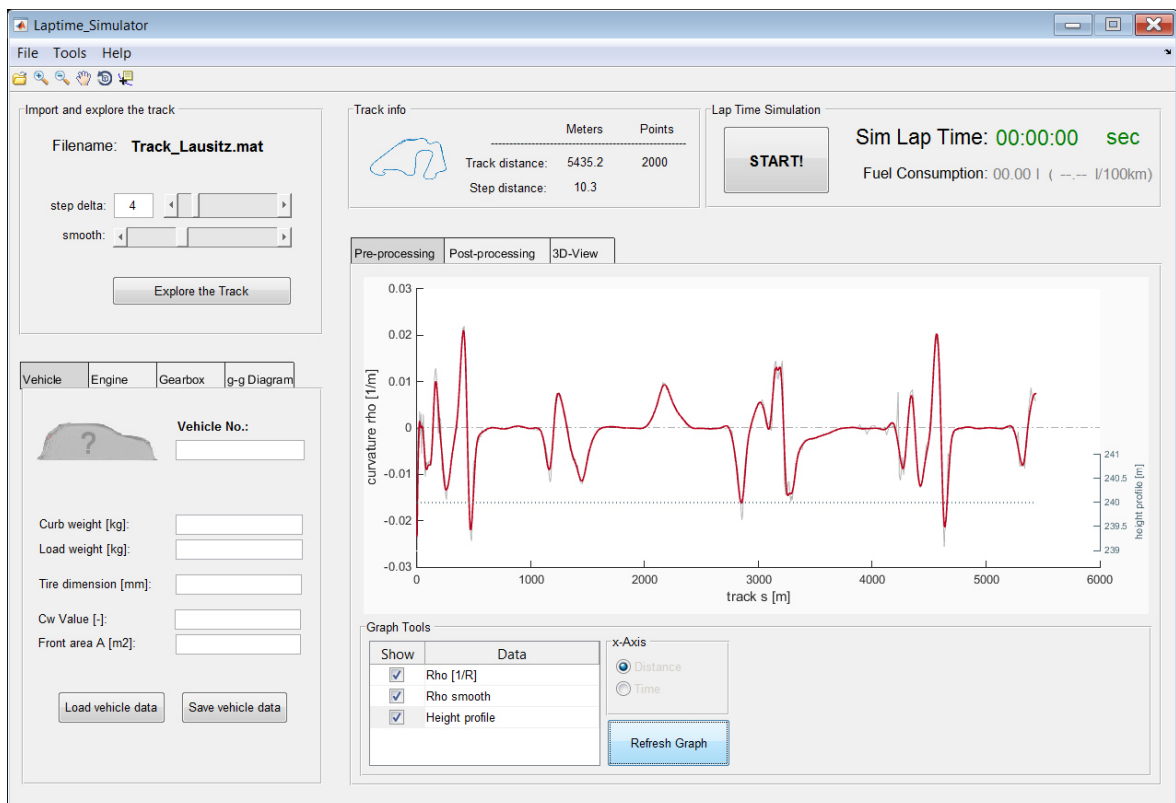


Figure 6.2: Track Processing

At first, short track info with general track visualization is displayed. To get relevant track description,¹⁰⁰ there is a button „**Explore the Track**”. In order to reach qualitatively good result of track description, the analysis can be adjusted in two steps, exactly as it has been described in the simulation model¹⁰⁰ (Step delta and parameter smooth). Both steps are also visualized in graph window where the red curve represents then the final result after smoothening.

¹⁰⁰ See: Chapter 4.1.1

6.2 Loading of vehicle parameters

In the box exactly under the panel with track import, there are in total four tabs where the car is fully specified. The complete vehicle profile can be saved as a *.mat file and next time just loaded without any time-consuming typing of single values into boxes.

In general, the tabs contain these settings for:

- Vehicle (chassis)
- Engine (torque map, BSFC map)
- Gearbox (gear ratios, shifting strategy)
- g-g diagram (performance envelope)

The engine data are saved in form of look-up tables, where for every throttle position is defined the full RPMs-range of torque values. After loading the engine map, there are displayed basic engine characteristics in form of maximal power and torque (*fig.6.4*).

| | $n=300$ | $n=800$ | $n=1000$ | $n=2000$ | $n=3000$ | $n=4000$ | $n=5000$ | $n=6500$ |
|-------------------|---------|---------|----------|----------|----------|----------|----------|----------|
| $\alpha=90^\circ$ | 110 | 110 | 125 | 140 | 170 | 185 | 200 | 190 |
| $\alpha=40^\circ$ | 110 | 106 | 115 | 140 | 160 | 165 | 175 | 155 |
| $\alpha=30^\circ$ | 110 | 105 | 110 | 130 | 150 | 155 | 160 | 140 |
| $\alpha=20^\circ$ | 110 | 100 | 105 | 125 | 110 | 95 | 70 | 40 |
| $\alpha=15^\circ$ | 110 | 95 | 100 | 100 | 75 | 50 | 25 | -20 |
| $\alpha=10^\circ$ | 110 | 90 | 80 | 65 | 35 | 20 | -10 | -40 |
| $\alpha=5^\circ$ | 110 | 55 | 50 | 20 | -25 | -40 | -50 | -60 |
| $\alpha=0^\circ$ | 110 | 27 | 0 | -32 | -51 | -60 | -63 | -65 |

Table 6.1 Engine Throttle Map ¹⁰¹

The tab with the data about a gearbox contains also the basic information regarding to driving style – there are defined the RPM levels for shifting up and down.

The particular driving style is then especially given by vehicle performance envelope loaded in form of g-g diagram (*fig.6.6*). Because of generally not-satisfactory validation results of simulation-based investigation of lateral capacity, this part was in final GUI not implemented. The g-g boundaries are entered into table *on fig.6.6*. For estimation of values that should be entered in g-g table, the mentioned simulation routine saved in ...\\Program\\Tools-GGV_tester\\Lateral_dynamics.slx can be used.

¹⁰¹ H.E. Scherf, Modellbildung und Simulation dynamischer Systeme, (2007), S.32

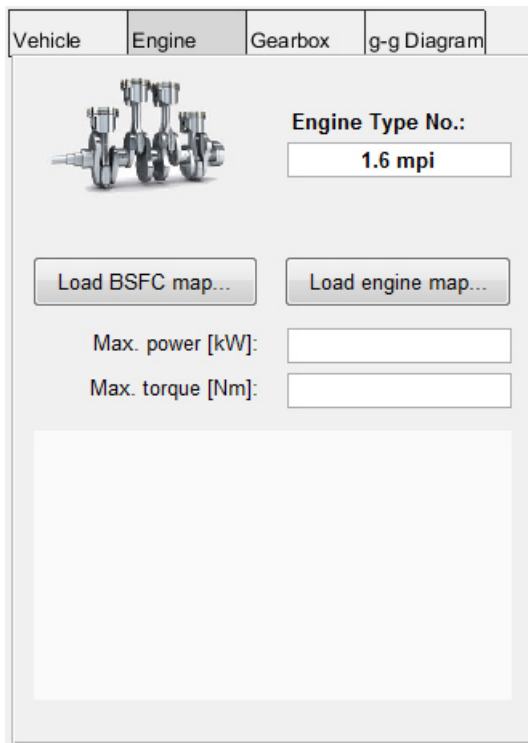


Figure 6.3: Engine Tab

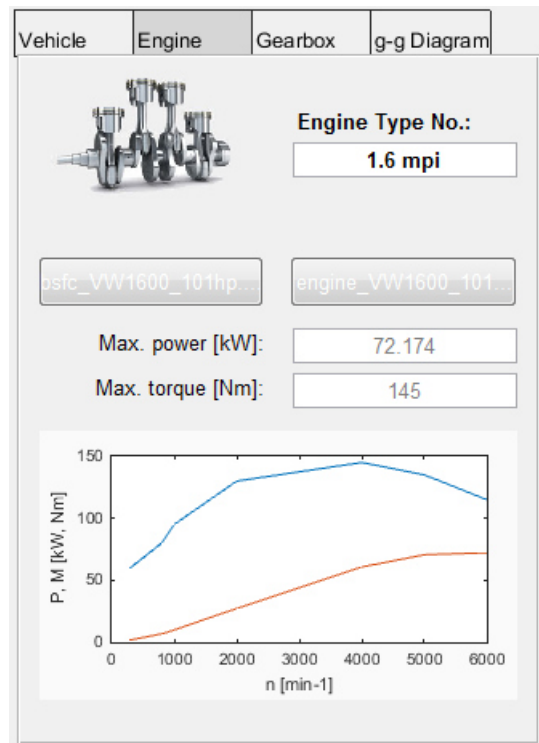


Figure 6.4: Loaded Engine Profile

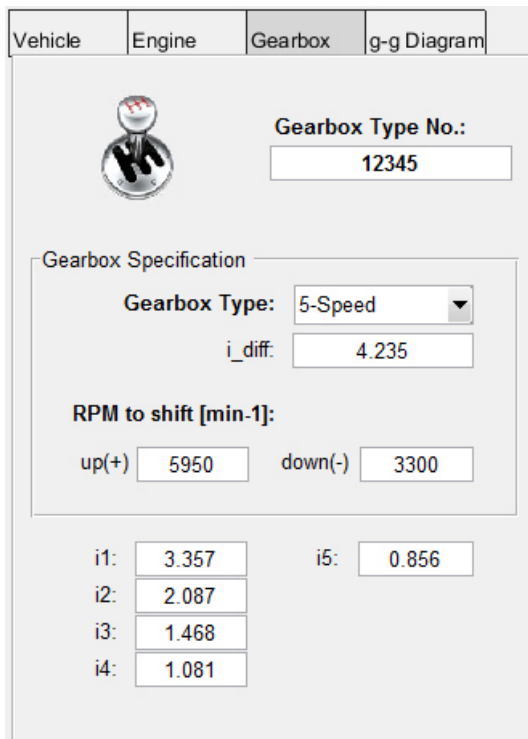


Figure 6.5: Gearbox Tab

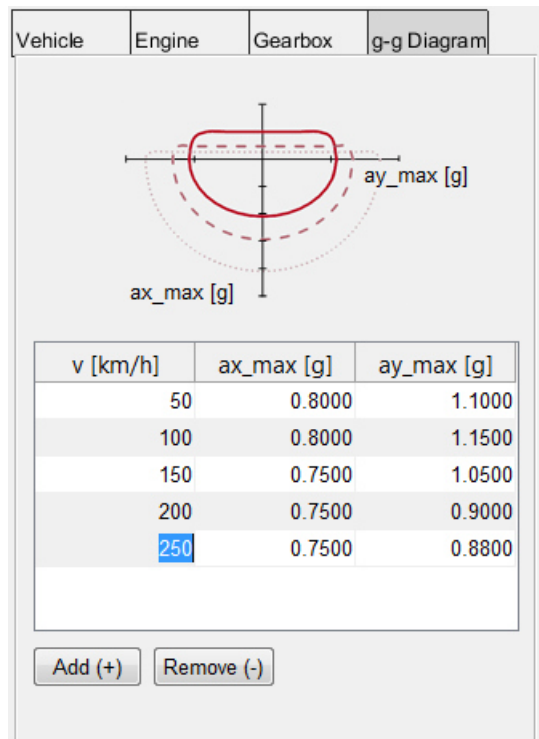


Figure 6.6: Definition of g-g Diagram

6.3 Lap time simulation

After all settings in the simulation data layer have been done, the final lap time calculation can be started by clicking on the **START** button. The first results, after the simulation has ended, are presented in the post-processing tab. The final lap time and fuel consumption are then stated in upper panel next to the start button (fig. 6.7).

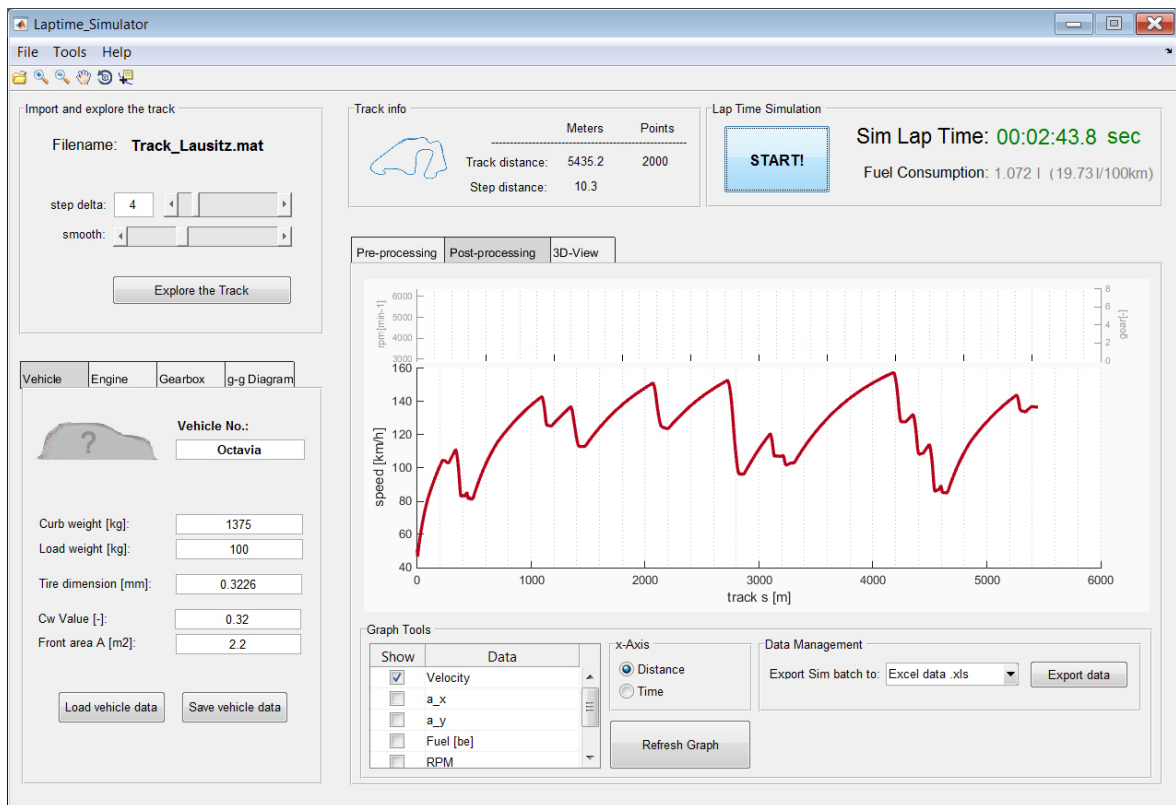


Figure 6.7: First Simulation Results

Additionally, it's possible to show more data by selecting the specific data layers from the list positioned in the panel with Graph Tools (see fig. 6.8).

There is the complete list of all items that can be displayed:

- Velocity
- Longitudinal and lateral accelerations
- Fuel
- RPMs
- Engaged gear
- Initial speed profile (boundary speed)

Generally, it's possible to plot all datasets against track (default) or time.

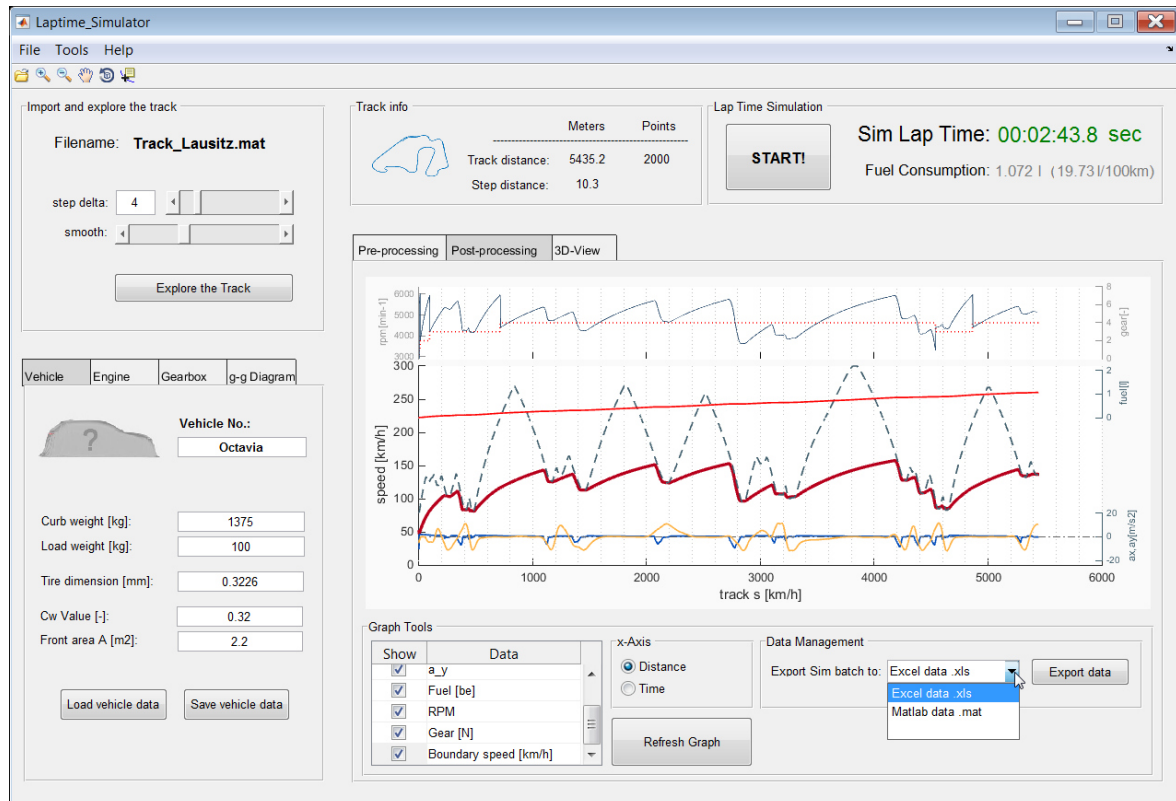


Figure 6.8: All simulated Data Layers

All data from simulation phase can be saved in form of default *.mat file or directly exported into selected directory in form of excel sheet with pre-defined header (fig.6.9).

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|------------|-----------|------------|------------|----------|-------------------------|-------------------------|----------|---------------|---|
| 1 | | | | | | | | | | | |
| 2 | | time [sec] | track [m] | speed [km] | RPM [min-] | Gear [-] | a_x [m/s ²] | a_y [m/s ²] | fuel [l] | fuel [100/km] | |
| 3 | | 0 | 0 | 50.4 | 750 | 0 | -9.92287 | -4.68064 | 0.0001 | 65535 | |
| 4 | | 0.1 | 1.350399 | 46.82917 | 0 | 1 | 0.901757 | -3.56214 | 0.0001 | 7.405221 | |
| 5 | | 0.2 | 2.655717 | 47.15371 | 5514.998 | 1 | 2.487441 | -3.16963 | 0.000398 | 14.997 | |
| 6 | | 0.3 | 3.977903 | 48.04099 | 5618.773 | 1 | 2.442044 | -2.85086 | 0.001466 | 36.84654 | |
| 7 | | 0.4 | 5.32451 | 48.91208 | 5720.653 | 1 | 2.397452 | -2.53078 | 0.002535 | 47.60622 | |
| 8 | | 0.5 | 6.695093 | 49.76726 | 5820.673 | 1 | 2.353651 | -2.20624 | 0.003605 | 53.84311 | |
| 9 | | 0.6 | 8.089212 | 50.6068 | 5918.865 | 1 | 2.310628 | -1.86835 | 0.004675 | 57.79587 | |

Figure 6.9: Export in Excel

The last option of data visualization is finally to see on fig.6.10. The 3-D View shows the calculated velocity profile in context with trajectory form. The red highlighted areas represent the lateral accelerations. Not only velocity profile can be displayed, but it's practically possible to select all calculated parameters. The track is, in this case, plotted just in form of x,y coordinates - the z-coordinate of height profile would act as disturbing element restricting the comprehensive visualization of other parameters.

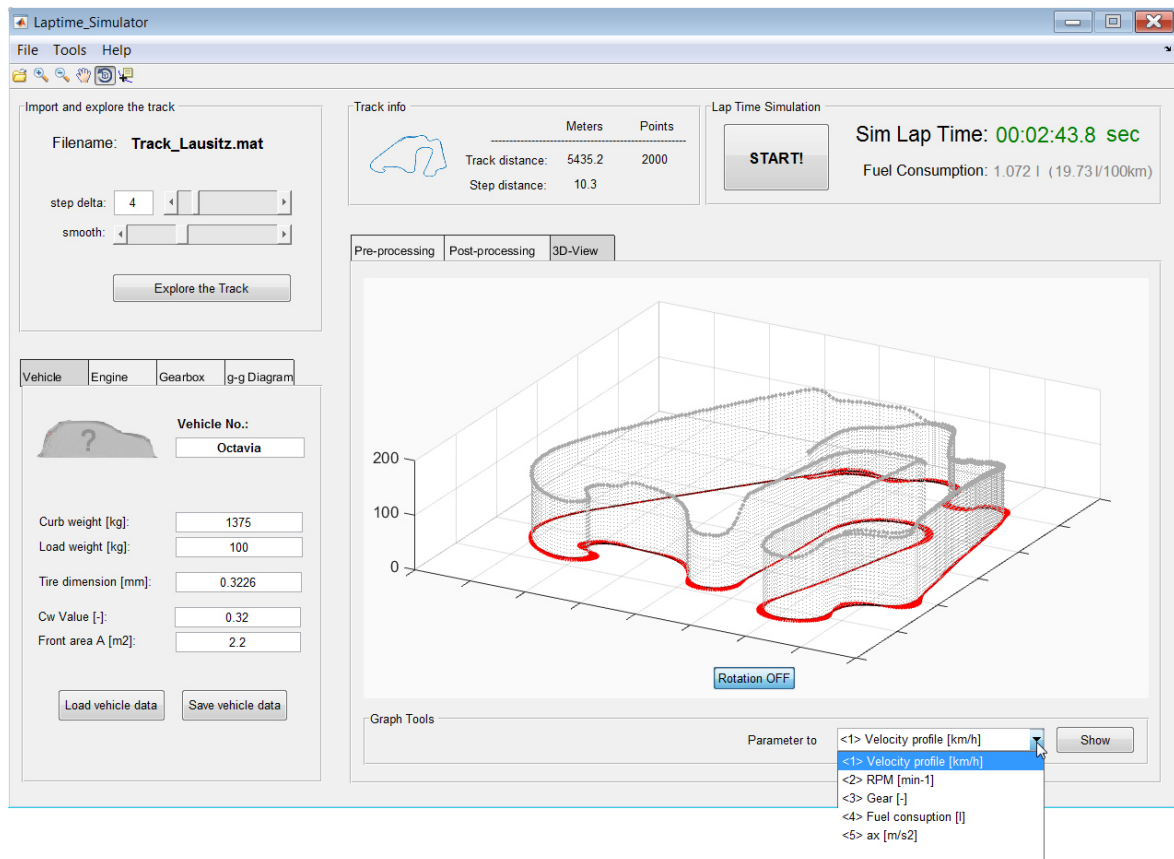


Figure 6.10: 3D View

6.4 Additional settings

All additional settings are possible to find in menu under **Tools – Settings**. On this place can be specified the remaining options regarding the driving strategy as modulation of maximal throttle-position or adjusting the braking maneuver on basis of findings in validation phase.¹⁰²

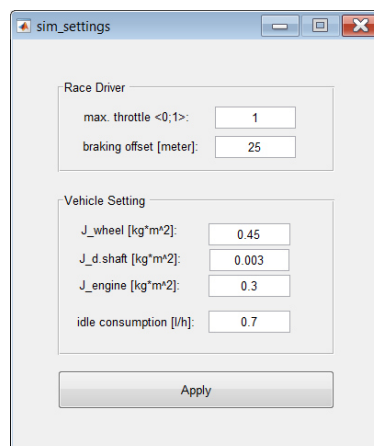


Figure 6.11: Additional Settings

¹⁰² See Chapter 5

7 Conclusion

The main objective of submitted work was the development of lap time simulation with additional functionality of evaluating the fuel consumption on the track. The work has been divided into six parts reflecting the whole development cycle.

The first part was intended as strictly opening chapter making a brief review of racing objectives and overall strategies. The three general racing trajectories have been introduced and evaluated on sample hairpin turn. On basis of simplified calculation model was found out that the time-ideal trajectory is not always represented by classical ideal racing line, but in fact the form of trajectory depends on relatively high number of parameters reflecting also the vehicle configuration, which leads to optimality based task.

The aim of second chapter is then brief summary of already known approaches for the lap time calculation. It was tried to create compact overview of all relevant strategies and their comparison of usability for intended calculation model. The main message of this report was that relatively unsophisticated quasi-static methods are already able to return quite accurate results. On the other hand, despite the enormous numerical demand of sophisticated approaches solving the optimality task, they cannot ensure the basic requirements regarding to result reproducibility. The stability of optimization process is very sensitive towards the non-linear behavior of vehicle during the racing drive. The perfect description of vehicle and track environment is obligatory.

Finally, having the basic clue about substance of particular simulation task, it has been started with development of own simulation tool. On the basis of stated requirements in third chapter, the two-staged simulation strategy is proposed. This concept works on detailed track analysis calculating in the first stage so-called initial speed profile reflecting only adhesion limits of the car. The second-stage with longitudinal vehicle model then makes the correction of pre-calculated initial guess regarding to specific power limitation of vehicle model.

The whole simulation model is then detailed in fourth chapter which is regarding the proposed strategy divided into two parts. At the end of each section are all simulation units functionally verified that they are working properly.

The real functionality of proposed simulation system is finally checked in validation phase where two vehicles have been tested and their simulation results then compared with real measured track data. Obtained results confirmed the importance of properly identified vehicle model. The proposed single-track model has been recognized as not sufficient for investigation of performance envelope necessary for initial profile calculation. On the other hand it was proved that, in case the calculation is fed with real data from telemetry, the simulation results are not too far from reality. This means that proposed algorithm calculating initial velocity is working fine.

The objective of last section was the development of GUI interface in order to simplify the work with simulation tool. This chapter covers practically the whole simulation workflow from viewpoint of normal end-user. All options and settings are clearly described. The simulation output can in the end be exported into .xls format for next data processing.

Literature

- [1] ADAMS/SmartDriver - Datasheet
- [2] Brayshaw, D. und Harrison, M. F.: "A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location". In: Proceedings IMechE, Part D: J. Automobile Engineering 219 (2005), S. 725–739.
- [3] Candelpergher, A., Gadola, M. und Vetturi, D.: Developments of a method for lap time simulation. SAE technical paper 2000-01-3562. 2000.
- [4] Casanova, D.: "On Minimum Time Vehicle Manoeuvring: The Theoretical Optimal Lap". Diss. School of Engineering, Cranfield University, 2000.
- [5] Eckstein L., IKA RWTH, Hrsg. Verkehrssystem Kraftfahrzeug, Kräfte am Fahrzeug, Antriebstrang, Bremsen, Fahrleistungen und Verbrauch ; Vorlesungsumdruck Fahrzeugtechnik I. 6. Auflage Aachen: Forschungsges. Kraftfahrwesen, 2012.
- [6] Gustafsson, T.: Computing The Ideal Racing Line Using Optimal Control. Masterthesis, Department of Electrical Engineering, Linköpings Universitat, LiTH-ISY-EX-08/4074-SE. 2008.
- [7] Heissing B. and Ersoy M.: Fahrwerkhandbuch Grundlagen, Fahrdynamik, Komponenten, Systeme, Mechatronik, Perspektiven. 3. Aufl. Wiesbaden: Springer Fachmedien Wiesbaden, 2011.
- [8] Milliken, W. F. and Milliken, D. L.: Race Car Vehicle Dynamics. 1. Auflage. Haynes Verlag, 1995.
- [9] Mitschke, M. and Wallentowitz, H.: Dynamik der Kraftfahrzeuge, Springer Verlag, Heidelberg, 2004.
- [10] Moss, S. and Pomeroy, L.: Design and Behaviour of the Racing Car. William Kimber, 1963.
- [11] Muhlmeier, M. und Muller, N.: "Optimization of the driving line on a race track". In: Proceedings of the 2002 SAE Motorsports Engineering Conference and Exposition (2002). SAE technical paper 2002-01-3339.
- [12] Pacejka, H. B. Tire and Vehicle Dynamics. 1. Auflage. Butterworth Heinemann, 2002
- [13] Patton, Ch.: Development of vehicle dynamics tools for motorsport. Diss. Oregon State University, 2013.

- [14] Velenis, E., Tsiotras, P. und Lu, J.: "Modeling Aggressive Maneuvers on Loose Surfaces: The Cases of Trail-Braking and Pendulum-Turn". In: Proceedings of the 2007 European Control Conference (2007).
- [15] Völkl T.: Erweiterte quasistatische Simulation zur Bestimmung des Einflusses transienten Fahrzeugverhaltens auf die Rundenzeit von Rennfahrzeugen. Diss. TU Darmstadt, 2013.
- [16] Schütz T., Herausgeber. Hucho - Aerodynamik des Automobils Strömungsmechanik, Wärmetechnik, Fahrdynamik, Komfort. 6., vollst. überarb. u. erw. Aufl. 2013. Wiesbaden: Springer Fachmedien Wiesbaden, 2013.
- [17] Siegler, B. und Crolla, D.: Lap Time Simulation for Racing Car Design. SAE technical paper 2002-01-0567. 2002.
- [18] Siegler, B., Deakin, A. und Crolla, D.: Lap Time Simulation: Comparison of Steady State, Quasi- Static and Transient Racing Car Cornering Strategies. SAE technical paper 2000-01-3563. 2000.
- [19] Spiegel, B.: Die obere Hälfte des Motorrades-Über die Einheit von Fahrer und Maschine, Heinrich Vogel Verlag, 3. Auflage München, 1999.
- [20] Waldmann, P.: Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken, Shaker Verlag, Aachen, 2009.
- [21] Weber, W.: Fahrdynamik in Perfektion Der Weg zum optimalen Fahrwerk-Setup, Motorbuch, Stuttgart, 2011.
- [22] Rill, G.: Simulation von Kraftfahrzeugen. 1. Auflage. Vieweg, 2007.

Appendix 1:

Track analysis

```

%
% Track Analysis -----
% =====
% v1.4 - 20160814 - Tomas Novotny
% function: convert [x,y]dataset --> rho(s)curvature over the track distance
%
%     data_inputs: xy coordinates
%     data_outpus: rho_v_filter1; s_v
%
%     GUI Controls: delta(line34); smooth_nr(line117)
% -----

% Step no.1 - INPUT TRACK DATA - coordinates [x,y] -----
%
x_in=x.';           %convert the format layout
y_in=y.';

%     Step no.2 - TRACK GEOMETRY DETECTION -----
%
% -----> 2.1 intilization of new data arrays
%
s=0;               % initial track distance
rho_v = zeros(1, length(x_in)); % data array curcature
s_v = zeros(1, length(x_in)); % data array track distance

x_new=x.';        % dataset for 3D graph
y_new=y.';        % (must be the same length as simulation input rho_v)

% -----> 2.2 adjusting parameter step size delta between points i and i+1
%
delta = 6;        % It's GUI PARAMETR no.1! (can be adjusted)

% -----> 2.3 triangle geometry
%
for i=1:delta:(length(x_in)-delta)

    dx1=x_in(i+0.5*delta)-x_in(i); % triangle t1
    dy1=y_in(i+0.5*delta)-y_in(i);
    alfa1=atan(dy1/dx1);           % course angle alfa1

    dx2=x_in(i+delta)-x_in(i+0.5*delta); % triangle t2
    dy2=y_in(i+delta)-y_in(i+0.5*delta);
    alfa2=atan(dy2/dx2);           % course angle alfa2
end

```

```

    ds=sqrt((dx1+dx2)^2+(dy1+dy2)^2);      % length of ds (stepsize)
    s_v(i)=s;                               % incremental track size in steps

    s=(s+ds);                               % overall track length (overwrite of initial
value)

    alfa=alfa2-alfa1;                       % trajectory course angle (tecny uhel k
trajektorii)

    rho=alfa/ds;
    rho_v(i)=rho;                           % incremental track curvature in steps (1/R)
end

%      Step no.3 - FORMATING OF DATA ARRAYS -----
%      fixing of data arrays lenghts -> ereasing of zero cells caused by step delta
%
%
% -----> 3.1 Differentiation of zero cells caused only by delta step
%      real zero curvature values have to be kept

for i=1:delta:length(rho_v)
    for ii=1:1:delta-1                      % odd numbers restriction
        s_v(i+ii)=10;                       % differentiation by an out-of-range value (exp. 10)
        rho_v(i+ii)=10;
        x_new(i+ii)=10;
        y_new(i+ii)=10;
    end
end

% -----> 3.2 Ereasing of marked points (with value 10)

s_v(s_v == 10) = [];
rho_v(rho_v == 10) = [];
x_new(x_new == 10) = [];
y_new(y_new == 10) = [];

% -----> 3.3 Ereasing of last array cell, because of delta-1 on the line 66

s_v(length(s_v)) = [];
rho_v(length(rho_v)) = [];
x_new(length(x_new)) = [];
y_new(length(y_new)) = [];

%      Step no.4 - DATA FILTRATION -----
%
%
% -----> 4.1 - Filter no.1 - removing of out-of-range values => rewrite to zero
%
for i=1:length(rho_v)
    if (rho_v(i)>=0.1) || (rho_v(i)<=-0.1)
        rho_v(i)=0;
    end
end
end

```

```
% -----> 4.2 - Filter no.2 - smoothing of zero places from step 4.1
%
for i=1:1:length(rho_v)-2
    if (rho_v(i)~=0) && (rho_v(i+1)==0) && (rho_v(i+2)~=0)
        rho_v(i+1)=(rho_v(i)+rho_v(i+2))/2;
    end
end

% -----> 4.3 - Filter no.3 - final curve smoothing
%
% tool: http://www.mathworks.com/help/curvefit/smoothing-
% data.html
%
rho_v_filter1 = smooth(s_v, rho_v, 0.06, 'loess'); % smooth_nr is GUI PARAMETR no.2!
```

Published with MATLAB® R2014b

Appendix 2:

Initial velocity profile

```

%
% Control velocity profile -----
% =====
% v2.1 - 20160816 - Tomas Novotny
% function: calculation of boundary velocity profile
%
% data_inputs: rho_v_filter1; s_v
% data_outpus: v_new_km (boundary profile for control input)
%
% GUI Controls: ggTable(line33); top_speed(line 48)
% -----

% DATA ARRAYS INTIALIZATION -----
% =====

N=length(rho_v_filter1); % N ... declaration of array length (depends on track
length)

v_base_ay=zeros(1, N); % v_base_ay ... basic velocity profile (only ay)
v_new=zeros(1, N); % v_new ... final velocity profile (ay & ax)

ay_n=zeros(1, N); % ay_n ... data array for lateral acc. while braking
ay_p=zeros(1, N); % ay_p ... data array for lateral acc. while accelerating
ax_neg=zeros(1, N); % ax_neg ... data array for neg. acc.
ax_pos=zeros(1, N); % ax_pos ... data array for pos. acc.

ay_aero_actual=zeros(1, N);
v_basic_ay=zeros(1, N);

% DATA INPUTS -----
% =====

ggInput = [ % ggTable ... data for of gg-diagram ---> GUI Parameter No.1
50 1.5 1.1 % 1.column = velocity level (v)
100 1.3 1.1 % 2.column = max. acc. ax(v)
150 0.8 1.2 % 3.Column = max. acc. ay(v)
200 0.7 1.3
250 0.6 1.3];

ggTable=[0,ggInput(1,2),ggInput(1,3); ggInput];

v_aero = (ggTable(:,1))/3.6; % velocity array
ax_aero = ggTable(:,2); % ax_acc fce(v)
ay_aero = ggTable(:,3); % ay_acc fce(v)
ay_basic = min(ggTable(1,3)); % ay_acc intial
ay_max = ggTable(6,3);

v_max=250/3.6; % top speed ---> GUI Parameter No.

```



```

% CALCULATION MODEL -----
% =====

% Step no.1 - Calculation of basic speed profile according ay=v^2/R
% -----
%
%
% ---- Step 1.1: v_base = sqrt(ay*R) ... for ay = konst.
%
for i=1:1:N
v_basic_ay(i)=min(v_max, sqrt((ay_basic)/abs(rho_v_filter1(i))));
end

% ---- Step 1.2: correction of v_base profile for ay = fce(v)
%
for i=1:1:N
    ay_aero_actual(i)=interp1(v_aero, ay_aero, v_basic_ay(i),'spline'); % interpolation of gg
data
    v_base_ay(i)=min(v_max, sqrt(ay_aero_actual(i)/abs(rho_v_filter1(i))));
end

% Step no.2 - Calculation of braking maneuvers ... backwards loop (N->1)
% -----
%
%
v_actual1=v_base_ay;          % definition of extra speed array for this step

for i=N:-1:2

    ay_n(i-1)=min(ay_max, v_actual1(i)^2*abs(rho_v_filter1(i-1)));
    % real lateral acc. for v_base_ay

    ay_aero_actual=interp1(v_aero, ay_aero, v_actual1(i));
    % max. lateral acc for v_actual

    ax_aero_actual=interp1(v_aero, ax_aero, v_actual1(i));
    % max. longitudinal acc for v_actual

    p=ax_aero_actual^2/ay_aero_actual^2;
    % ellipse parameter

    ax_neg(i-1)=real(sqrt(p*(ay_aero_actual)^2 - p*(ay_n(i-1))^2));
    % actual ax for braking (ellipse)

    v_actual1(i-1)=min(v_base_ay(i), sqrt(v_actual1(i)^2+2*ax_neg(i-1)*(s_v(i)-s_v(i-1))));
    % final speed iteration, backwards loop (N->1)

end

```

```

% Step no.3 - Calculation of acceleration maneuvers ... forwards loop (1->N)
% -----
%
%
v_actual2 = v_base_ay;           % definition of extra speed array for this step

for i=1:1:N-1

    ay_p(i+1)=min(ay_max,v_actual2(i)^2*abs(rho_v_filter1(i+1)));
    % real lateral acc. for v_base_ay

    ay_aero_actual=interp1(v_aero, ay_aero, v_actual2(i));
    % max. lateral acc for v_actual

    ax_aero_actual=interp1(v_aero, ax_aero, v_actual2(i));
    % max. longitudinal acc for v_actual

    p=ax_aero_actual^2/ay_aero_actual^2;
    % ellipse parameter

    ax_pos(i+1)=real(sqrt(p*(ay_aero_actual)^2 - p*(ay_p(i+1))^2));
    % actual ax for accelerating (ellipse)

    v_actual2(i+1)=min(v_base_ay(i), sqrt(v_actual2(i)^2+2*ax_pos(i+1)*(s_v(i+1)-s_v(i))));
    % final speed iteration, forwards loop (1->N)

end

% Step no.4 - Addition of both arrays -> Generate the final speed profile
% -----
%
%
for i=1:1:N
    v_new(i) = min(real(v_actual1(i)), real(v_actual2(i)));
end

v_new_km = v_new*3.6; % convert to km/h; boundary profile for control input!

```

Published with MATLAB® R2014b

Appendix 3:

Lateral force at pure slip for sample tire 205/55 R16

```

function Fy1 = fcn(Fz,ga,alf)

FNOMIN = 4700;

PCY1 = 1.26750770709;
PDY1 = 0.900306094598;
PDY2 = -0.167479289311;
PDY3 = -0.431843698162;
PEY1 = -0.346197273355;
PEY2 = -0.103742794757;
PEY3 = 0.115058178269;
PEY4 = -6.95357120308;
PKY1 = -25.7371397371;
PKY2 = 3.27019793551;
PKY3 = -0.00536363421643;
PHY1 = 0.00311146581545;
PHY2 = 2.08186325307e-005;
PHY3 = -0.0370286277317;
PVY1 = 0.00649325487186;
PVY2 = -0.00520414365481;
PVY3 = 0.0126232741011;
PVY4 = -0.00668823390518;

LCY = 1;
LMUY = 1;
LEY = 1;
LFZO = 1;
LGAY = 1;
LKY = 1;
LHY = 1;
LVY = 1;

Fzo = FNOMIN * LFZO; %1
dfz = (Fz-Fzo)/Fzo; %2
gay = ga * LGAY; %3
SHy = (PHY1 + PHY2*dfz)*LHY + PHY3*gay; %4
alfy = alf + SHy; %5
Kyo = PKY1 * FNOMIN * sin(2*atan(Fz/(PKY2 * FNOMIN * LFZO))) * LFZO * LKY; %6
Ky = Kyo * (1 - PKY3 * abs(gay)); %7
Cy = PCY1*LCY; %8
MUY = (PDY1 + PDY2 *dfz) * (1 - PDY3 * (gay)^2)*LMUY; %9
Dy = MUY * Fz; %10
By = Ky/(Cy*Dy); %11
Ey = (PEY1 + PEY2 * dfz) * (1-(PEY3 + PEY4 * gay) * sign(alfy)) * LEY; %12
Svy = Fz * ((PVY1 + PVY2 * dfz) * LVY + (PVY3 + PVY4 * dfz) * gay) * LMUY; %1

```

```
Fy1 = Dy * sin(Cy * atan(By * alfy - Ey * (By * alfy - atan(By * alfy)))) + SVy;
```

Published with MATLAB® R2014b

Source: Adams/Tire Documentation - Using the PAC2002Tire Model