

## 2. Vehicular Models

Computational vehicle modeling provides a simulation environment and enables the analysis of the effect of different models on the controller's design. For example, some classical controllers work very well with a linear vehicle model that can be quite useless when applied to the control of a racing car. The linearization does not consider high speeds and aggressive attitude that an extremely nonlinear dynamic system like this is submitted to.

However, the best model is always the simplest one. Therefore, many models - with different complexities - were used in the vehicle kinematics analysis. Furthermore, the track information must also be stored, and the model used was the same as in [11]. In this section, the models used in this work are detailed and validated through individual tests.

### 2.1. Oriented Particle Vehicle Model

#### 2.1.1. Presentation and Description

The oriented particle vehicle model adopted in the optimization of a path in a predefined track is rather simple. It consists of a particle moving on the Cartesian Plane, where  $(x,y)$  symbolizes the position of the vehicle's center of mass. . The car's orientation is given by  $\theta$  and defined as the angle between the car front direction and the global  $X$ -axis. This angle lies between  $-\pi$  and  $\pi$ , which conventionally characterizes the vehicle's left side as positive and its right side as negative.

The model is graphically represented in Figure 2.1 and detailed in Equation (2.1). Its input variables are  $a_N$  and  $a_T$ , the normal and tangential accelerations respectively. The car's linear velocity is given by  $V$ , and  $\omega$  is its angular velocity. The sign of  $\omega$  defines whether the car is turning left or right. As the modeled racing car does not move backwards, the speed  $V$  assumes values equal or higher than zero.

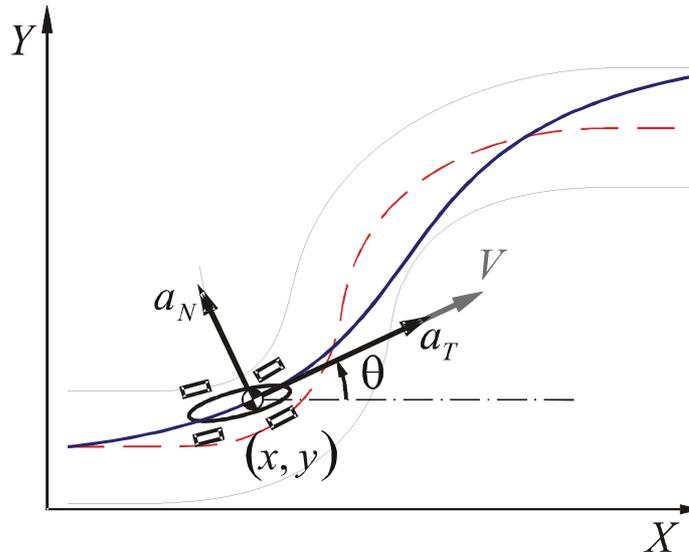
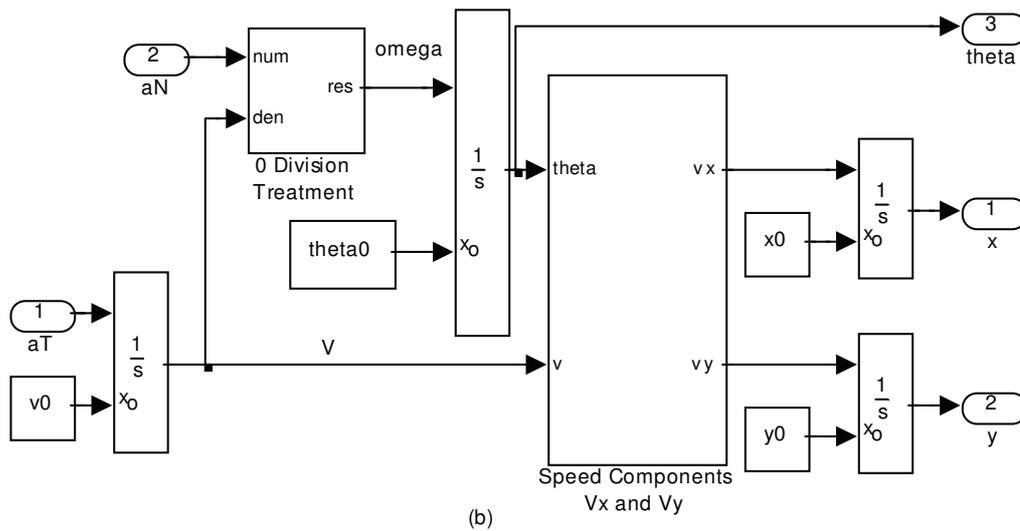
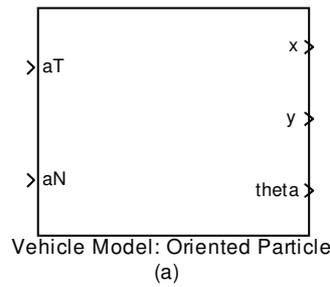


Figure 2.1 – Oriented Particle Model Representation.

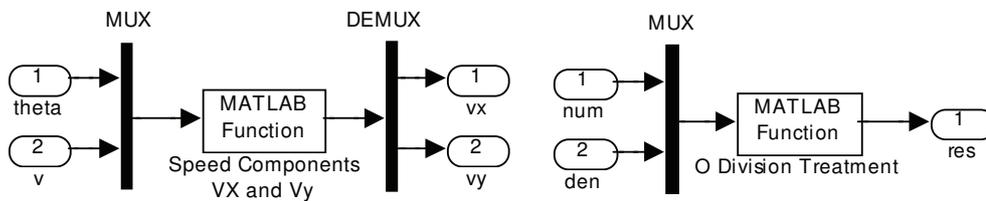
$$\left\{ \begin{array}{l} \omega = \begin{cases} \frac{1}{V} \cdot a_N & ; V > 0 \\ 0 & ; V = 0 \end{cases} \\ \dot{\theta} = \omega \\ \dot{V} = a_T \\ \dot{x} = V \cdot \cos \theta \\ \dot{y} = V \cdot \sin \theta \end{array} \right. \quad (2.1)$$

Following the modularity approach used in this work, the oriented particle vehicle model is implemented in the Simulink<sup>®</sup> environment and formatted in a single block. Figure 2.2 shows this block in its encapsulated or compact form (a) and in its extended form (b).

It should be noted that the blocks named “0 Division Treatment” and “Speed Components” are respectively the conditional and the trigonometric parts of the equations. Those calculations are better defined as *.m* files called in Simulink<sup>®</sup> by the MatLab<sup>®</sup> function blocks. Those blocks are then put inside a subsystem together with a MUX and a DEMUX ports that respectively concatenate and separate the MatLab<sup>®</sup> function’s input and output variables. Following the modularity methodology, this programming strategy is commonly used throughout the modeling; Figure 2.3 shows examples of two MatLab<sup>®</sup> functions used in the oriented particle vehicle model.



**Figure 2.2 – Simulink Block Diagram: Oriented Particle Vehicle Model. Compact Form (a) and Extended Form (b).**



**Figure 2.3 – MatLab Function Form Examples.**

Some of the simplifying hypothesis, such as the absence of mass and inertia representation, may cause the impression that the model is insufficient. However, much of the inertial behavior is implicitly considered in the acceleration profiles given as inputs. Figure 2.4 shows the profiles of the acceleration component tangent to the trajectory – vehicle’s longitudinal acceleration.

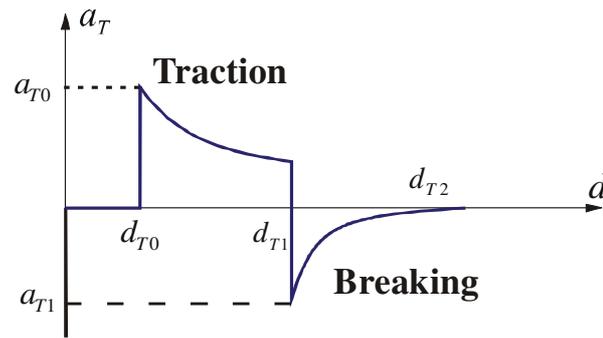


Figure 2.4 – Tangent Acceleration Profiles.

This representation can be modeled as two conditional exponential functions [11]. In Equation (2.2) this acceleration profile is detailed, where:

- $a_{T0}$  and  $a_{T1}$  are the possible peak tractions and breaking accelerations;
- $d_{FT}$  is the distance where the maximum speed is achieved;
- $d_{FB}$  is the minimum breaking distance possible;
- $d_{T0,1,2}$  are the characteristic distances of the tangent acceleration profile;
- $d$  is the dynamic traveled track distance which it is the profiles' dependent variable;
- $\tau$  is a decay constant of the exponential.

$$a_T = \begin{cases} a_{T0} \cdot e^{\left(-\tau \cdot \frac{d-d_{T0}}{d_{FT}-d_{T0}}\right)} & ; d_{T0} \leq d < d_{T1} \\ -a_{T1} \cdot e^{\left(-\tau \cdot \frac{d-d_{T0}}{d_{FB}-d_{T1}}\right)} & ; d_{T1} \leq d < d_{T2} \end{cases} \quad (2.2)$$

Figure 2.5 shows the profiles of the acceleration component normal to the trajectory – in this model the lateral acceleration of the vehicle. The direction convention is again left as positive and right as negative.

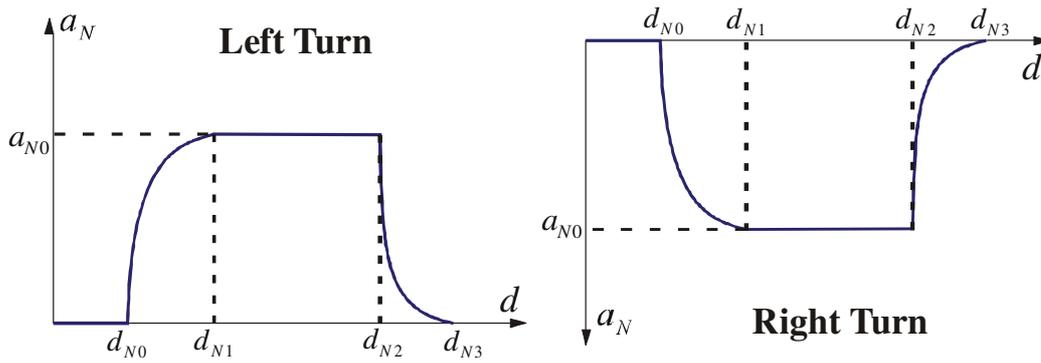


Figure 2.5 – Normal Acceleration Profiles.

Equation (2.3) details the conditional equations that define  $a_N$ . The desired turn acceleration is  $a_{N0}$ , and  $d_{N0,1,2}$  is the characteristic distance of the normal acceleration profile.

$$a_N = \begin{cases} a_{N0} - a_{N0} \cdot e^{\left(-\tau \cdot \frac{d-d_{N0}}{d_{N1}-d_{N0}}\right)} & ; d_{N0} \leq d < d_{N1} \\ a_{N0} & ; d_{N1} \leq d < d_{N2} \\ a_{N0} \cdot e^{\left(-\tau \cdot \frac{d-d_{N2}}{d_{N3}-d_{N2}}\right)} & ; d_{N2} \leq d < d_{N3} \end{cases} \quad (2.3)$$

The Friction Ellipse introduces the vehicles' responses to those accelerations into the model. This graph, as detailed in [9], points out the acceleration limits a specific vehicle can support before it slides or spins. Those limits vary according to some features, such as tires' composition, engine power and the center of mass position. Each car presents a different Friction Ellipse and only three acceleration values are necessary to define it. These values are the maximum traction, lateral and braking accelerations, respectively named  $a_{TM_{ax}}$ ,  $a_{LM_{ax}}$  and  $a_{BM_{ax}}$ . Figure 2.6 shows an example of Fiction Ellipse and the dynamic acceleration values in two time instants ( $t_1$  and  $t_2$ ) that are analyzed to attend the restriction imposed by the three components of Maximum Acceleration.

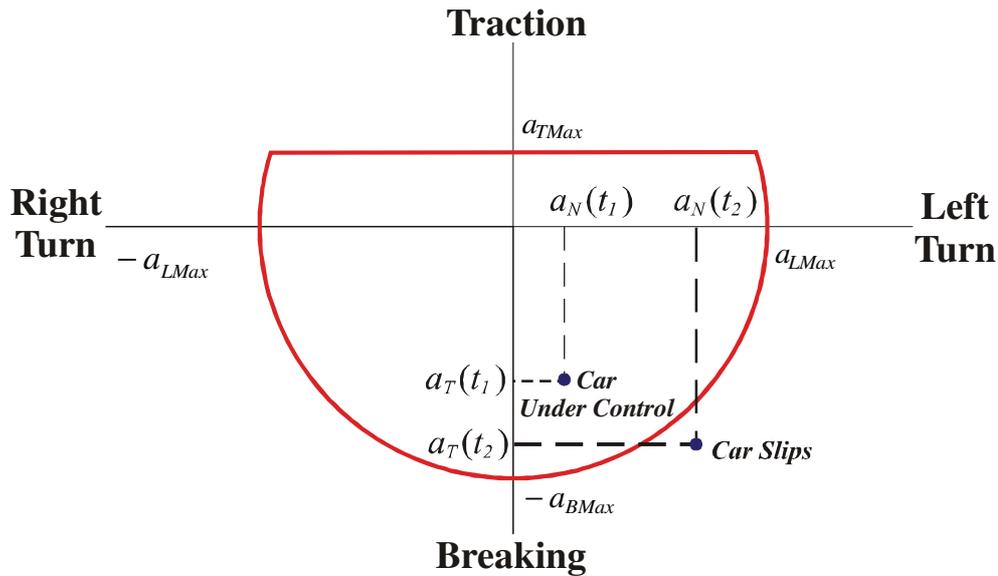


Figure 2.6 – Friction Ellipse.

### 2.1.2. Validation Tests

The tests for this model consist of applying some acceleration inputs and observe how the particle moves through the Cartesian Plan. The first test has the simple purpose of evaluating if the oriented particle vehicle model responds correctly to the given accelerations. As shown in Figure 2.7, the inputs to the model are pulses (see Figure 2.8) representing an initial speed increase followed by two left turns.

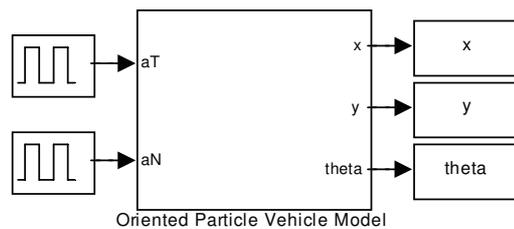
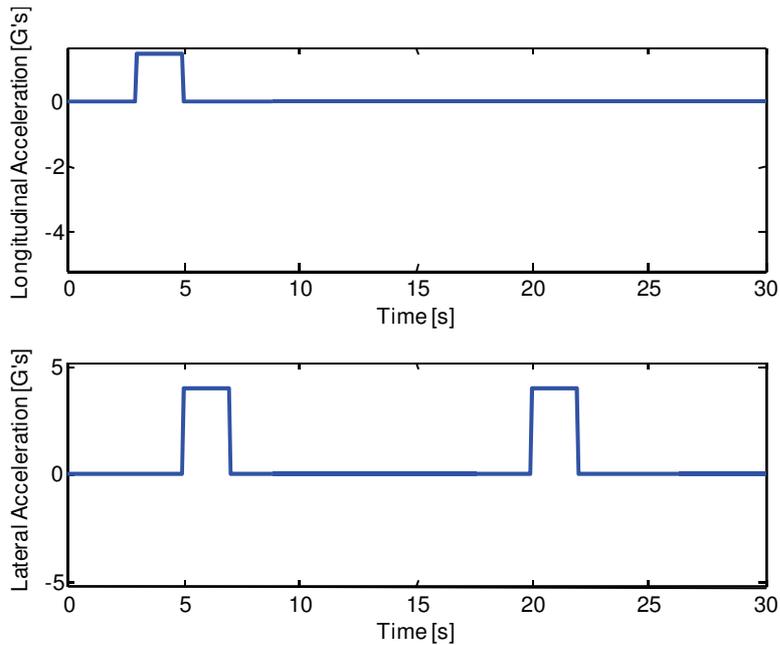


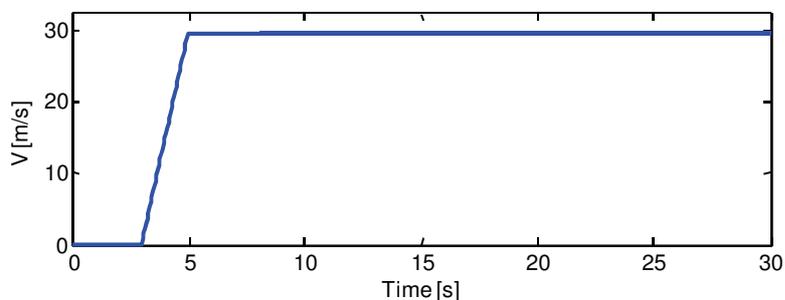
Figure 2.7 – Oriented Particle Model 1<sup>st</sup> Validation Test: Simulink Block Diagram.



**Figure 2.8 – Oriented Particle Model 1<sup>st</sup> Validation Test: Inputs.**

It should be emphasized that the accelerations do not necessarily correspond to the ones actually experimented by a vehicle. However, this kind of test intends to verify the coherence of the model's responses. Moreover, submitting the model to simple inputs makes it easier to recognize expected outputs and validate the model.

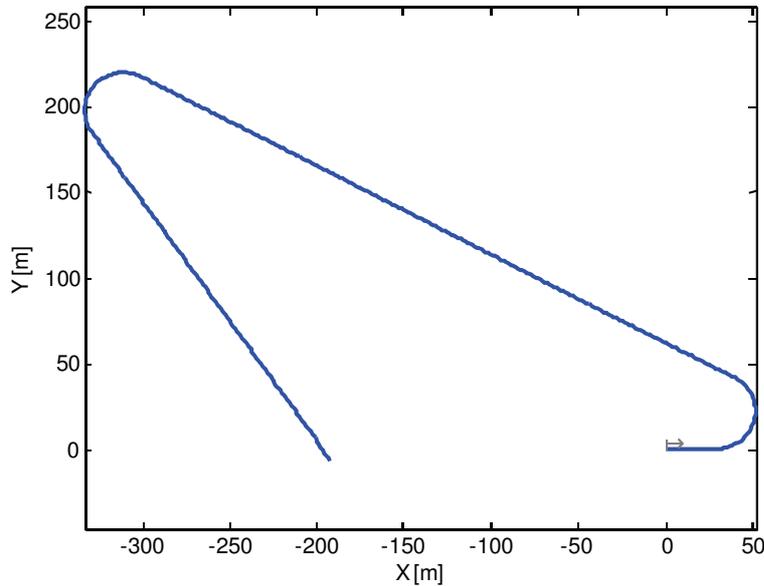
By analyzing the first input signal, it can be seen that about the third simulation second the vehicle receives a positive longitudinal acceleration, which is maintained constant until the fifth second. This acceleration should naturally increase the car speed to a certain level and then become constant, as no other longitudinal input is given afterwards. Figure 2.9 shows the dynamic speed variable and how the expected behavior is attained.



**Figure 2.9 – Oriented Particle Model 1<sup>st</sup> Validation Test: Vehicle Speed.**

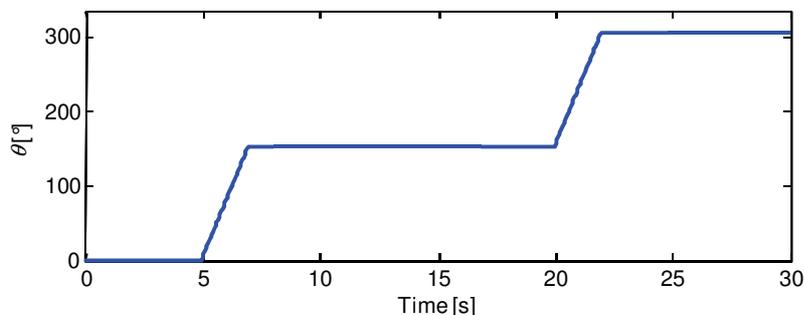
In addition to the longitudinal acceleration, the second input represents the lateral acceleration. This input signal corresponds to a couple of two-second

positive steps in the fifth and twentieth simulation seconds. The expected model response to design of two curves to the left, as in Figure 2.10. To generate this trajectory image, both output variables ( $x,y$ ) are plotted together. The small gray arrow indicates the beginning of the displacement.



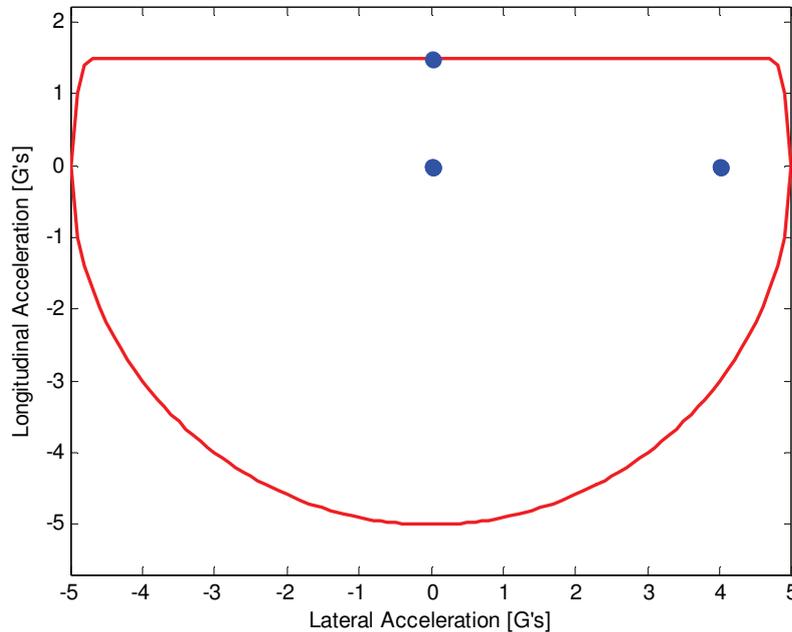
**Figure 2.10 – Oriented Particle Model 1<sup>st</sup> Validation Test: Performed Trajectory.**

An important observation is that the obtained curves have the same radius due to the fact that the car was moving with constant speed and that the lateral acceleration had the same amplitude in both pulses. The orientation output can also be visualized in Figure 2.11. Comparing the trajectory with  $\theta$ , it is possible to validate the orientation calculation. Both turns shown in Figure 2.10 have about a hundred and fifty degrees, easily identified by the two different levels of  $\theta$  in Figure 2.11.



**Figure 2.11 – Oriented Particle Model 1<sup>st</sup> Validation Test: Orientation Output.**

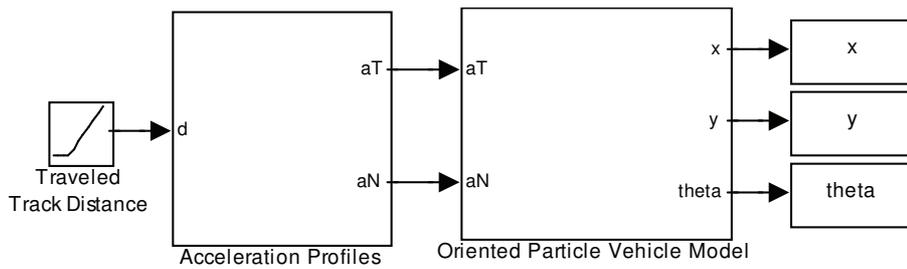
This type of test establishes constant levels of accelerations. As shown in Figure 2.12, the acceleration levels plots some points together in the Friction Ellipse.



**Figure 2.12 – Oriented Particle Model 1<sup>st</sup> Validation Test: Friction Ellipse.**

The dots in the center of the ellipse are the acceleration representation at the moment the car was stopped or on a straight line with constant speed. Moreover, the dots on the right represent the acceleration experimented by the vehicle when turning. Finally, the higher dots represent the initial traction acceleration that takes the car out of rest.

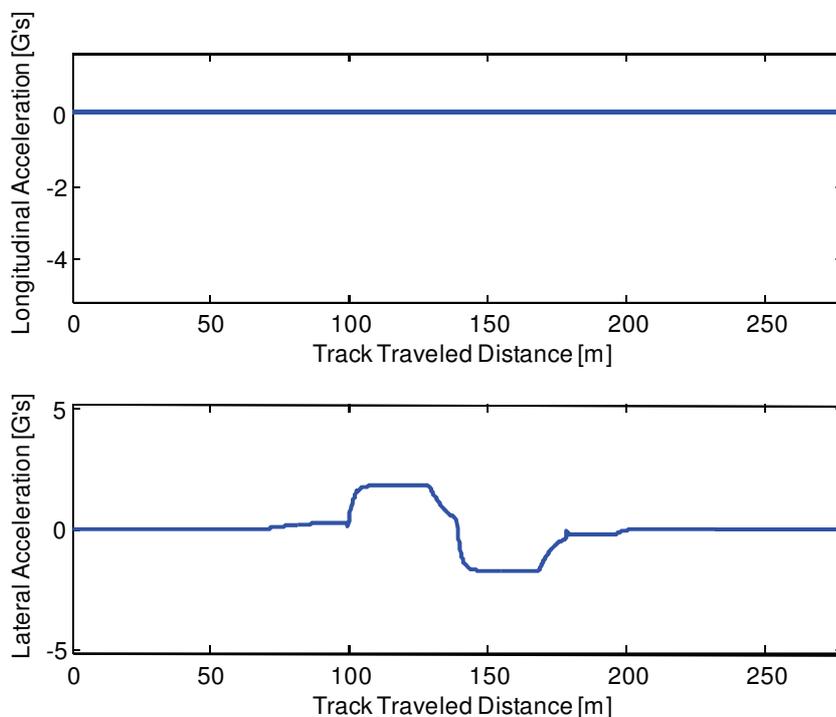
The next tests introduce the acceleration profiles as the oriented particle model inputs. Those profiles, defined in Equations (2.2) and (2.3), were assembled in a MatLab<sup>®</sup> function, which is called by a Simulink<sup>®</sup> Block. The Simulink<sup>®</sup> Block Diagram shown in Figure 2.13 is used in the next three following tests. The traveled track distance, detailed in Chapter 3, becomes the test input. Although a ramp is used as the time varying input distance, in future simulations that value is taken as a feedback of the global position concerning a determined track.



**Figure 2.13 – Acceleration Profiles Validation Tests: Simulink Block Diagram.**

Considering that the acceleration inputs of the model are now written as distance functions, it is natural that they are plotted against travelled track distance instead of time. For a better comparison, all the other variables are also plotted against distance. As the travelled track distance is completely dependent on the track, these tests will show the trajectory within the track limits.

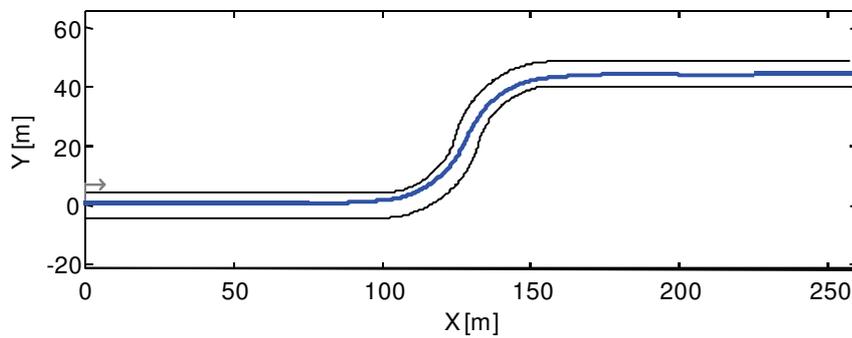
The acceleration profiles defined for this second test intent to submit the model to a set of accelerations equivalent to those experienced by a car while on “S” turn with constant speed. As shown in Figure 2.14, the longitudinal acceleration remains zero along the track travelled distance; the lateral one represents the driving commands on the steering wheel in order to complete the desired path.



**Figure 2.14 – Oriented Particle Model 2<sup>nd</sup> Validation Test: Inputs.**

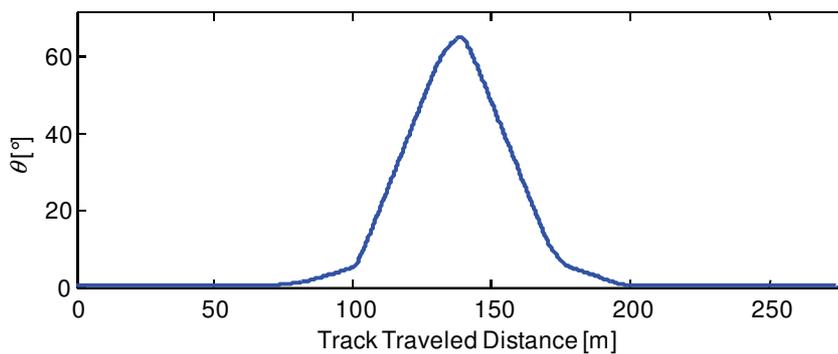
Both axes of this figure are limited by the acceleration limits established as the Friction Ellipse characteristic values for the tests. Those values correspond to the average values for racing cars and are shown in Equation (2.4). The resulting trajectory of those inputs is shown in Figure 2.15, where the gray arrow indicates the direction of movement.

$$\vec{a}_{Max} = \begin{bmatrix} a_{TMax} \\ a_{LMax} \\ a_{BMax} \end{bmatrix} = \begin{bmatrix} 1.5 \\ 5 \\ 5 \end{bmatrix} \quad (2.4)$$

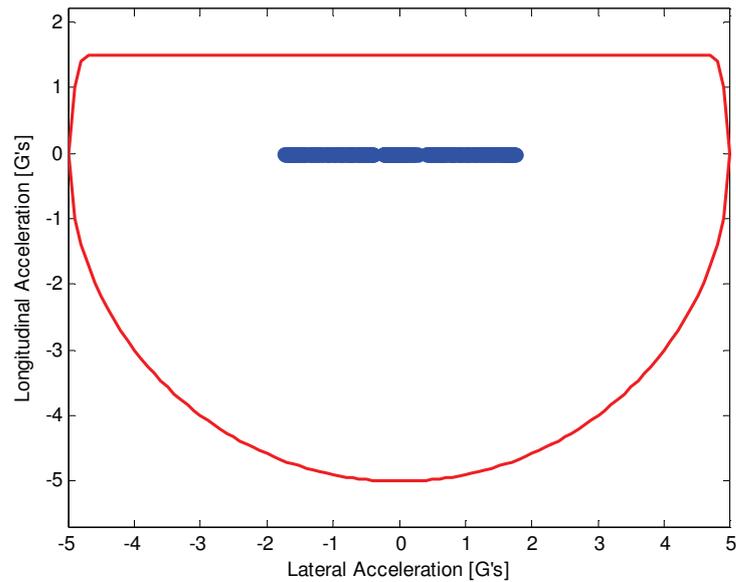


**Figure 2.15 – Oriented Particle Model 2<sup>nd</sup> Validation Test: Trajectory.**

The orientation output,  $\theta$ , is shown in Figure 2.16. By comparing it to the trajectory, it is possible to identify their coherence. Finally, the Friction Ellipse with all the dots along the trajectory is represented in Figure 2.17.



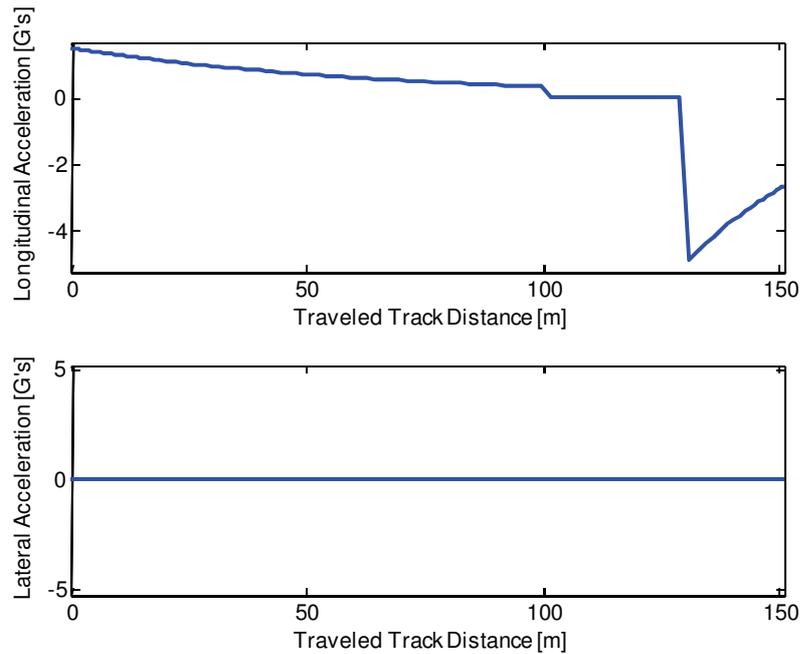
**Figure 2.16 – Oriented Particle Model 2<sup>nd</sup> Validation Test: Vehicle Orientation.**



**Figure 2.17 – Oriented Particle Model 2<sup>nd</sup> Validation Test: Friction Ellipse.**

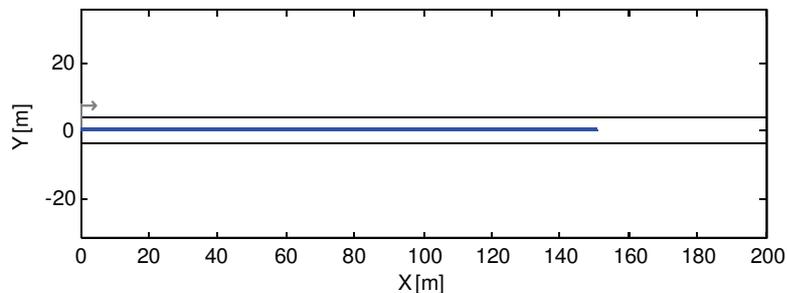
The results of this test enable the acceleration model validation; and especially the hypothesis that lateral acceleration profiles can represent the immediate consequence of the steering command. Hence, the performed trajectory reproduces the expected displacement. Furthermore, the Friction Ellipse does not show any variation regarding longitudinal acceleration and shows the same lateral accelerations amplitude as the inputs. This perfectly illustrates the constant speed with an “S” turn test.

The lateral accelerations input remains as zero throughout the travelled distance. As shown in Figure 2.18, the longitudinal acceleration profile establishes that the car accelerates during the first hundred meters, remains with the same speed for thirty meters, and then stops.



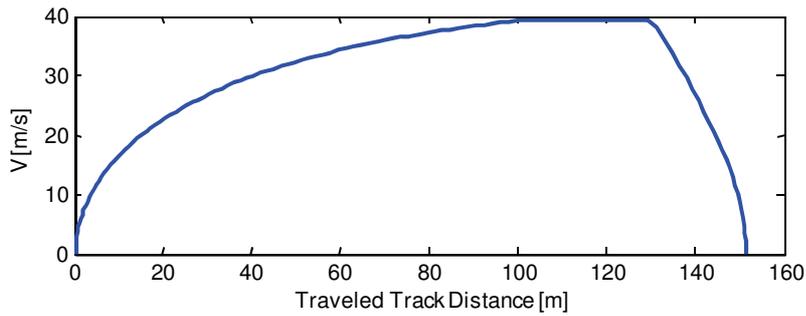
**Figure 2.18 – Oriented Particle Model 3<sup>rd</sup> Validation Test: Inputs.**

The trajectory could not be different from the straight line, since there is no lateral acceleration. Figure 2.19 shows this trajectory, also with a small gray arrow indicating the displacement origin and direction. It can be seen that the car stops before the end of the track.



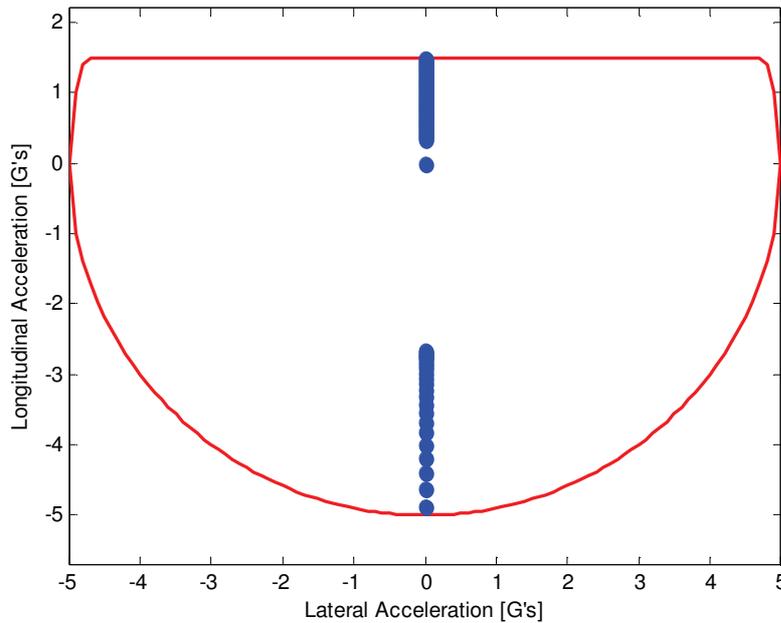
**Figure 2.19 – Oriented Particle Model 3<sup>rd</sup> Validation Test: Realized Trajectory.**

The displacement behavior mentioned before can be clearly observed by analyzing the vehicle speed graph. Moreover, Figure 2.20 shows how the acceleration profiles attribute to the particle model an asymptotic speed response, usually characteristic of inertial models.



**Figure 2.20 – Oriented Particle Model 3rd Validation Test: Vehicle Speed.**

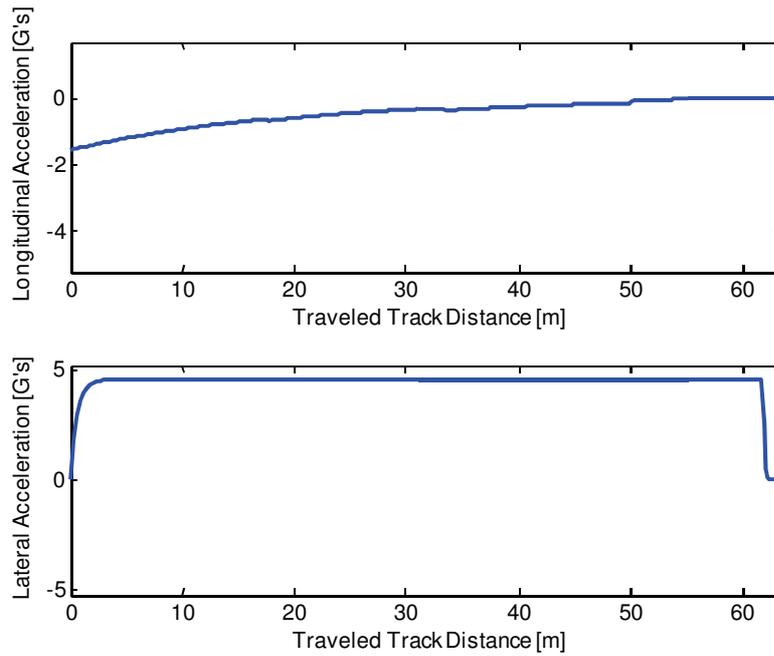
The Friction Ellipse plot, shown in Figure 2.21, completes the longitudinal validation analysis. The orientation output is not plotted in this test, as the trajectory is a straight line where  $\theta$  remains zero throughout the travelled track distance.



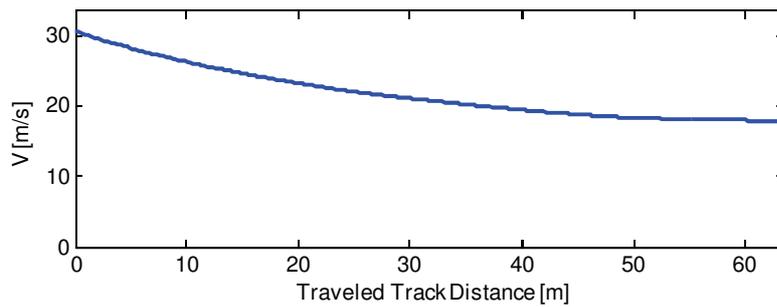
**Figure 2.21 – Oriented Particle Model 3rd Validation Test: Friction Ellipse.**

The final validation test has the purpose of analyzing both acceleration profiles together. When submitted to a constant lateral acceleration, the trajectory should be a curve whose radius is a quadratic function of the car's speed.

The test inputs are defined as illustrated in Figure 2.22. The longitudinal acceleration input represents a soft braking command that culminates in a decreasing speed, as seen in Figure 2.23.

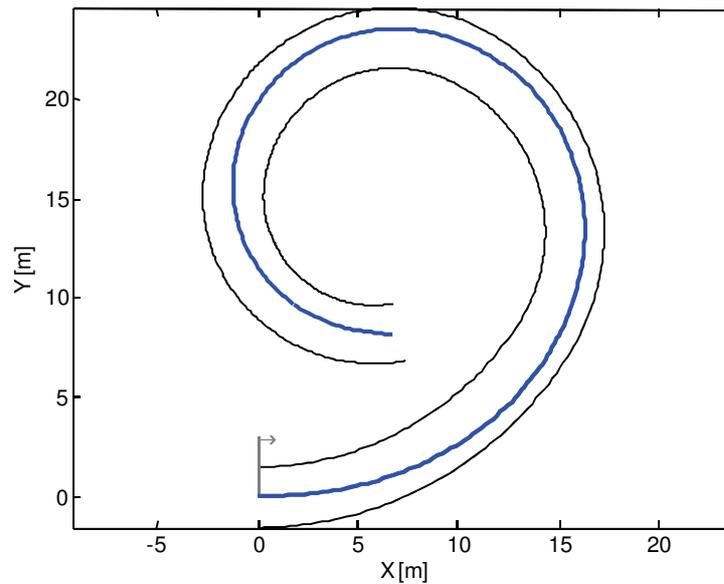


**Figure 2.22 – Oriented Particle Model 4th Validation Test: Inputs.**



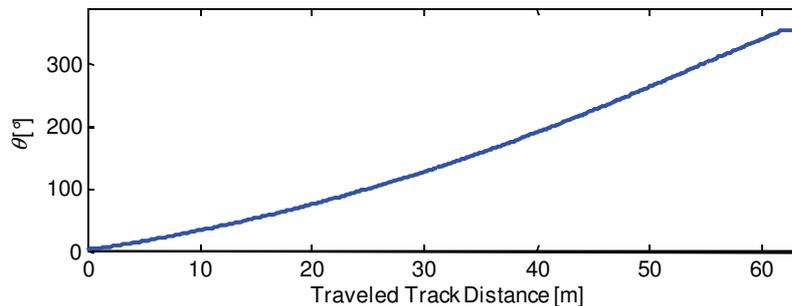
**Figure 2.23 – Oriented Particle Model 4th Validation Test: Vehicle Speed.**

Despite some initial and final transients, the lateral acceleration describes a constant wheel's angle. The trajectory response is also as expected: starting at the small gray arrow, it is curve with a decreasing radius. The spiral trajectory is shown in Figure 2.24.



**Figure 2.24 – Oriented Particle Model 4th Validation Test: Realized Trajectory.**

Figure 2.23 shows the decreasing speed that causes the spiral behavior, and Figure 2.25 shows the vehicle orientation  $\theta$ . Since the car completes a full turn, the orientation increases about three hundred and sixty degrees.



**Figure 2.25 – Oriented Particle Model 4th Validation Test: Orientation.**

Finally, to complete the model validation analysis, the Friction Ellipse associated to the spiral trajectory is shown in Figure 2.26. A relevant observation is that no acceleration was too close to the limits given by  $\vec{a}_{Max}$ . However, due to the elliptical shape of the graph's boundaries, the car almost reaches a critical condition.

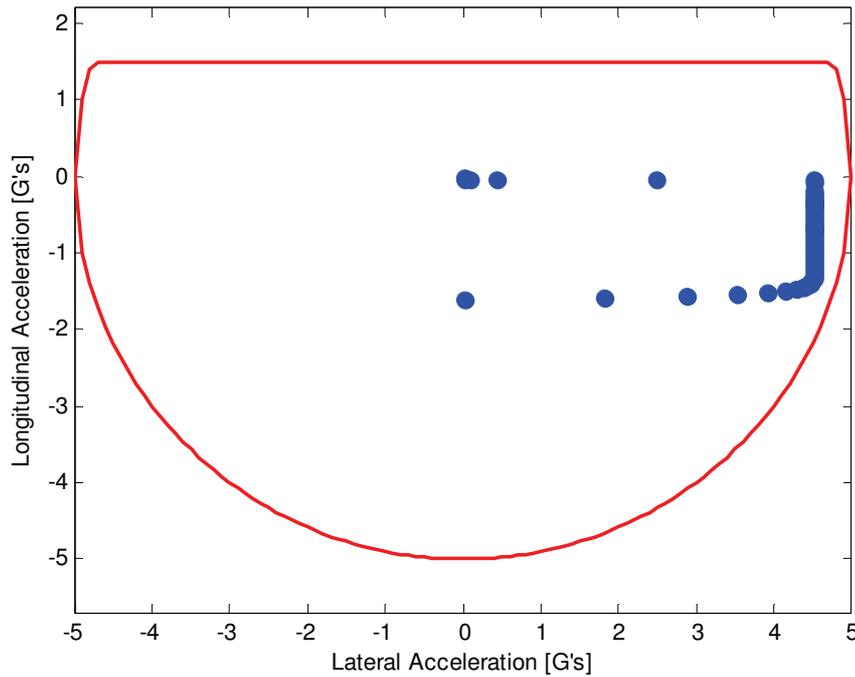


Figure 2.26 – Oriented Particle Model 4th Validation Test: Friction Ellipse.

## 2.2. Stationary Kinematical Vehicle Model

### 2.2.1. Presentation and Description

In addition to the optimization trajectory, this paper proposes the design of a mimetic human controller capable of tracking and maneuvering a car through that trajectory. As this controller uses the error in position and orientation of the vehicle to determine its steering wheel angle, any model that does not consider the Ackerman Geometry or any other element of the steering system is inadequate. Due to it, the stationary kinematical model, also used in [12], replaces the previous one. This model and the car's complete direction system are detailed below.

First, the vehicle's body kinematics is modeled again as a planar moving particle. However, a local Cartesian reference system is fixed in the car's center of mass. Due to it, the speed  $V$  is free to make an angle with the  $x$  vehicle axis.

The car's angular velocity is given by  $\omega$ , and the attack angle between the local  $X$ -axis of the car and its speed is  $\alpha_v$ . Figure 2.27 illustrates the first part of this modeling.

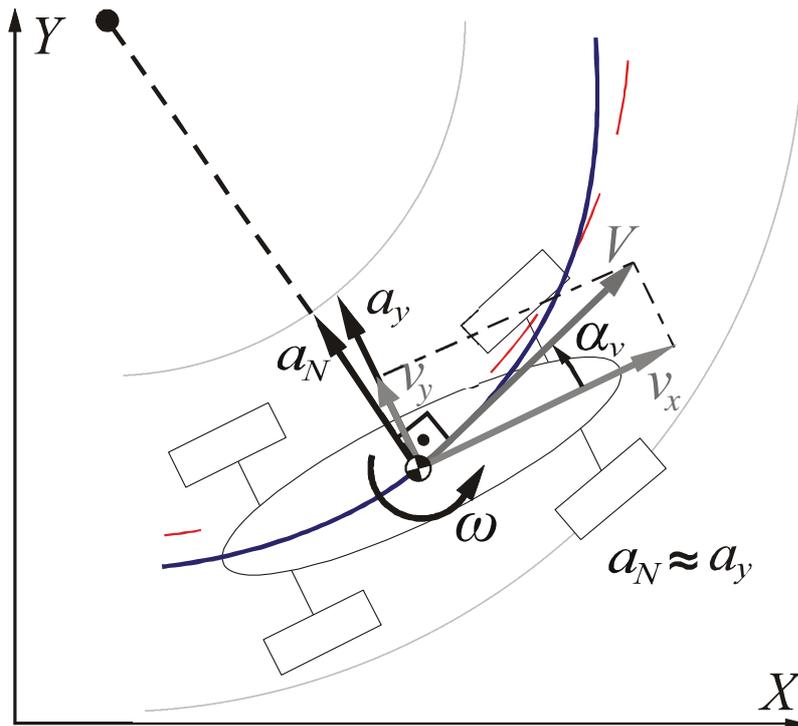


Figure 2.27 – Kinematical Model Representation.

As mentioned before, this model assumes the stationary condition, that is, the car's speed is constant. Another simplifying hypothesis is that the normal acceleration can be approximated by the  $y$  component of the acceleration referred to the vehicle's local reference system.

Instead of receiving accelerations as inputs, this model obtains them from the curve radius of the rear shaft,  $R_t$ , the speed  $V$  and the vehicle geometric configuration given by the distances between the rear and front axles and the center of mass,  $l_t$  and  $l_d$ , respectively. The second part of the kinematical model is the Ackerman geometry consideration. This geometry [9] states that, to avoid sliding in any wheel, the center of curvature must be in the interception of the extensions of all the axles of the vehicle. This configuration is easily understood through the graphical representation shown in Figure 2.28. The variables  $b_t$  and  $b_d$  are respectively the rear and front axles size. The angles  $\delta_e$  and  $\delta_d$  are changes of the steering wheel angle,  $\delta$ , for the left and right wheels respectively.

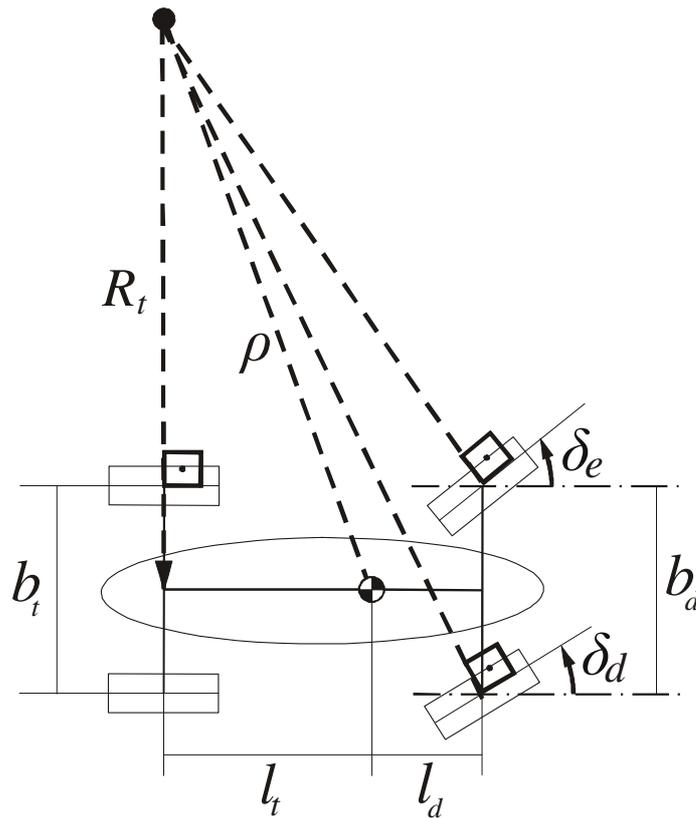


Figure 2.28 – Ackerman Geometry Definitions.

The simplifying hypotheses of the constant car speed (stationary condition) and the approximation of the normal acceleration by the  $y$  acceleration component should be recalled. Considering all the geometry shown on Figure 2.27 and Figure 2.28, the complete mathematical description of the considered model is given by Equations (2.5).

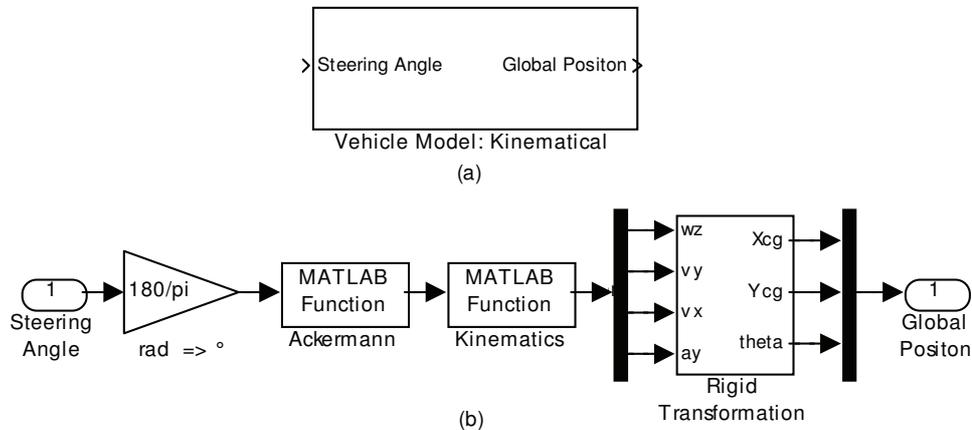
$$\left\{ \begin{array}{l}
\alpha_v = \begin{cases} \arctan\left(\frac{l_t}{R_t}\right) & ; R_t > 0 \text{ :. Left Curve} \\
-\arctan\left(\frac{l_t}{R_t}\right) & ; R_t < 0 \text{ :. Right Curve} \end{cases} \\
V = \frac{v_x}{\cos \alpha_v}; \\
v_y = v_x \cdot \tan \alpha_v \\
\rho = \sqrt{R_t^2 + l_t^2} \\
\omega = \begin{cases} \frac{V}{\rho} & ; R_t > 0 \text{ :. Left Curve} \\
-\frac{V}{\rho} & ; R_t < 0 \text{ :. Right Curve} \end{cases} \\
a_N = a_y = v_x \cdot \omega
\end{array} \right. \quad (2.5)$$

It can be seen that the condition of infinite radius – which corresponds to no curve – is modeled on  $R_t=0$ . This strategy is not only simpler to implement, but also computationally cheaper. The same procedure is repeated for the direction system modeling.

Some of the steering system parameters also influence on how the steering angle is translated on accelerations. The gain between the steering wheel angle and the present wheel angle is given by the variable  $K_d$ . Moreover, the gap that occurs when the steering wheel turns slightly but the wheels do not move is given by  $d_f$ . Those considerations on the steering system and the final  $R_t$  definition are mathematically detailed in Equation (2.6).

$$\left\{ \begin{array}{l}
\delta_e = K_d \cdot \delta \\
R_t = + \left[ \left( \frac{l_t + l_d}{\tan \delta_e} \right) + \frac{b_d}{2} \right] ; \delta > d_f \text{ :. Left Curve} \\
\delta_d = K_d \cdot \delta \\
R_t = - \left[ \left( \frac{l_t + l_d}{\tan \delta_d} \right) - \frac{b_d}{2} \right] ; \delta < -d_f \text{ :. Right Curve}
\end{array} \right. \quad (2.6)$$

Once implemented, the model's block receives the steering wheel angle command from the driver and responds with the curvature radius. The Simulink® Block Diagram of the model is shown in Figure 2.29.

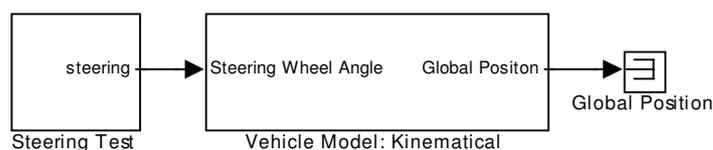


**Figure 2.29 – Simulink Block Diagram: Kinematical Vehicle Model. Compact Form (a) and Extended Form (b).**

A relevant observation is that this model has a non-inertial reference system; thus, a coordinate transformation is necessary. Usually, due to the modularity strategy, the Rigid Transformation Block that rewrites the state variables on the global reference system could be set outside the model block. However, as the controller to be developed should not use any information besides the global position and orientation, this structure works properly.

## 2.2.2. Validation Tests

As in the previous section, it is necessary to design a test scenario on the Simulink® environment. Figure 2.30 shows the Simulink® Block Diagram used in several tests, including the one below:



**Figure 2.30 – Kinematical Model Validation Tests: Simulink Block Diagram.**

The structure of the test scenarios is rather simple: an input block, the model to be tested and a visualization block. The Steering Test block is composed of a signal generator and connected to the input of the model block.

The Global Position block is no more than a scope probe to collect the model output data.

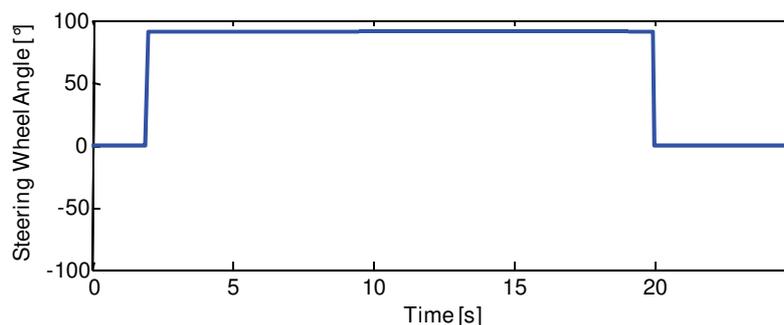
Differently from the previous model presented, the stationary kinematical model does not need any correlation to the track or the travelled distance. Separating the kinematical part from the Ackerman geometry does not make sense here. Therefore, all the variables are plotted against time, and only two validation tests are necessary to analyze the model responses.

To study the influence of the vehicle speed  $V$  on the output variables, three different values were tested. The car's geometry and the steering system parameters are detailed in Table 2.1.

**Table 2.1 – Kinematical Model 1st Validation Test: Vehicle Parameters**

Parameter	Variable	Value
Steering Loose Gap	$d_f$ [°]	0.1
Rear Axle Length	$b_t$ [m]	1.2
Front Axle Length	$b_d$ [m]	1.2
Distance between the Axles	$l$ [m]	2
Distance from the Center of Mass to the Rear Axles	$l_t$ [m]	1.2
Distance from the Center of Mass to the Front Axles	$l_d$ [m]	0.8
Steering Gain	$K_d$	0.015

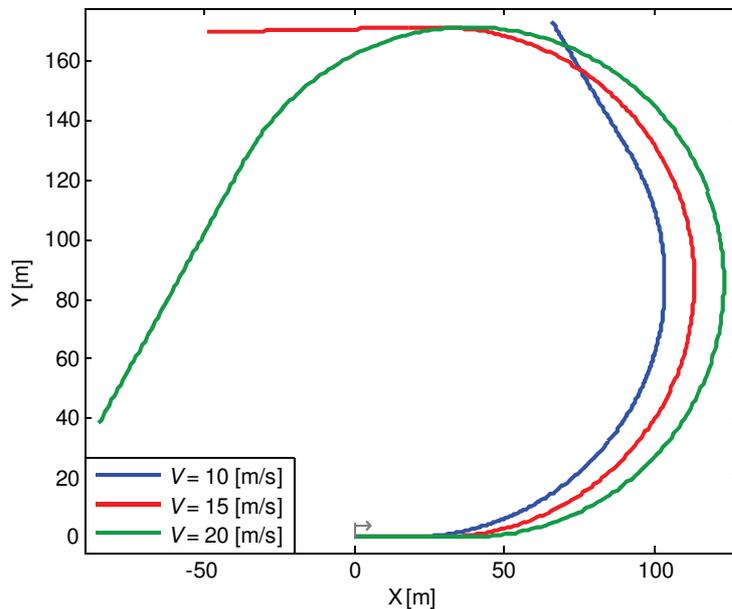
The first test consists of applying a pulse to the steering wheel angle input. That input should imply a constant lateral acceleration, causing the car to go through a “U” turn. The test input is shown in Figure 2.31.



**Figure 2.31 – Kinematical Model 1st Validation Test: Input.**

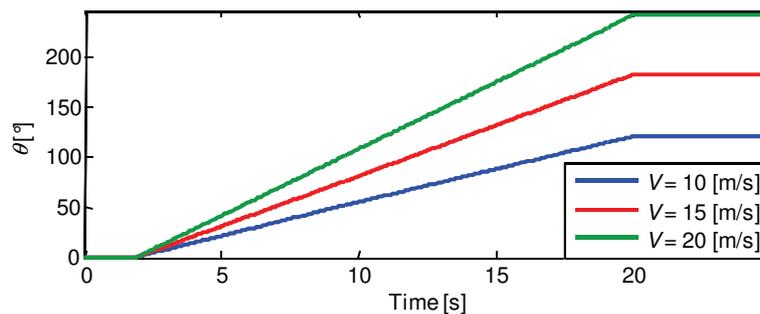
The three trajectories with different speeds are plotted together in Figure 2.32. The expected characteristics of the trajectories are observed for all the tested values of  $V$ , although only for  $V=15\text{m/s}$  the “U” turn is perfectly obtained.

The higher the speed, the later the car enters the curve. Moreover, for a constant simulation time, the lower the testing speeds, the smaller the car displacement.



**Figure 2.32 – Kinematical Model 1<sup>st</sup> Validation Test: Trajectory Comparison.**

The orientation output,  $\theta$ , is also plotted for the three tested speeds in Figure 2.33. By analyzing it together with the trajectories, it is possible to observe that the difference in inclinations means that the three trajectories are not only delayed in phase, but also have different curve's radius.



**Figure 2.33 – Kinematical Model 1<sup>st</sup> Validation Test: Orientation Comparison.**

The second test has the purpose of evaluating the model behavior when submitted to an oscillatory input. The steering wheel is initially turned  $90^\circ$  to the left and, with a sinusoidal movement, moves to  $90^\circ$  to the right and back in a period of about eight seconds. As seen in Figure 2.34, a driver executes the same control in an “S” turn.

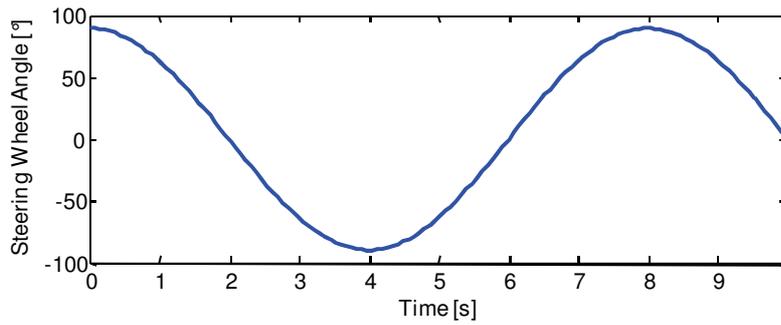


Figure 2.34 – Kinematical Model 2<sup>nd</sup> Validation Test: Input.

This input is tested again with three different speeds:  $V=10\text{m/s}$ ,  $V=15\text{m/s}$  and  $V=20\text{m/s}$ . Figure 2.35 shows the corresponding trajectories. It can be seen that the highest speeds not only cause the car to go further along the  $X$ -axis, but also increase the  $Y$  coordinate reached.

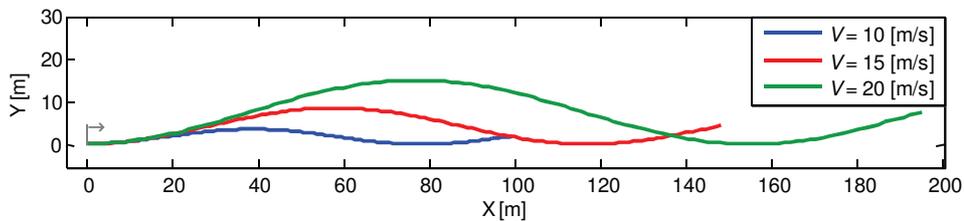


Figure 2.35 – Kinematical Model 2<sup>nd</sup> Validation Test: Realized Trajectories Comparison.

As in the previous test, the next step is to analyze the car orientation output. Figure 2.36 shows a comparison of the  $\theta$  graphs for the different speeds tested. The response corresponds to the model's expectation and to the previously observed trajectories. At higher speeds, the car attains larger angular variations.

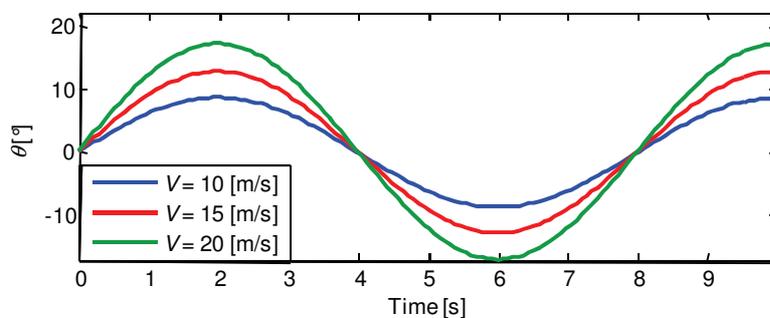


Figure 2.36 – Kinematical Model 2<sup>nd</sup> Validation Test: Orientation Comparison.

As all the presented vehicular models were tested and validated, they can now be used to study the vehicle behavior, to design controllers or specific car components, or even for training the driver. In the next sections these models are used as objective functions in the optimization of the vehicle trajectory and as benchmarks for the controller design.