



HAL
open science

Viability Theory and Road safety

Pierre-Olivier Vandanjon, Alex Coiret, Tristan Lorino

► **To cite this version:**

Pierre-Olivier Vandanjon, Alex Coiret, Tristan Lorino. Viability Theory and Road safety. 2013.
hal-00776297

HAL Id: hal-00776297

<https://hal.science/hal-00776297>

Preprint submitted on 15 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Viability Theory and Road safety

P-O. VANDANJON**, A. COIRET, T. LORINO
LUNAM University, Ifsttar, AME, EASE, F-44341 Bouguenais, France

v1.0 released january 2013 submitted to Vehicle System Dynamics

January 15, 2013

Abstract

Viability theory proposes geometric metaphors in addition to classical ordinary differential equation analysis. This theory is applied on a vehicle cornering. The issue is to determine if a state of the system is compatible with a soft controls strategy which leads the car to the end of the corner. The viability theory offers tools which enable to answer to this question without computing in real time these controls. The issue resumes to verify that this state belongs to a set called the viability kernel. The construction and the use of the viability kernel for a vehicle system dynamic is proposed by using Support Vector Machines algorithm. The applicability of this theory is demonstrated through experimental tests. It is the first time that the viability theory is applied to vehicle dynamics with road safety concerns.

Keywords: viability theory, dynamic model, road safety, experimental tests, calm driving.

1 Introduction

ESP and ABS system try to find optimal controls in order to track the desired trajectory. These systems are triggered in dangerous situations. Before to reach these dangerous situations, it would be interesting to warn the driver that it will no longer be possible to follow the desired trajectory with “soft” controls compatible with a calm driving.

It is difficult to compute in real time a control only to check if a “soft” control can keep the car on the road. The challenge is : how to warn the driver that there is no more soft control to maintain the car on the road without computing in real time such a control ?

The viability theory offers a theoretical frame suitable to this challenge. The viability kernel is precisely the set of vehicle states from which there exist a soft

**Corresponding author. Email : pierre-olivier.vandanjon@ifsttar.fr, Phone : 33 240845634, Fax : 33 240845993

trajectory. The viability kernel is determined a priori, i.e. before driving. To warn the driver is equivalent to verify that the vehicle no longer belongs to the viability kernel.

The first two sections set the scene by presenting the viability theory and the vehicle dynamic modeling. The next section presents the link between these two fields by describing the viability constraints for the vehicle system dynamics. From these constraints, the computation and the use of the viability kernel for the system are described in the section 5. In the last section of the paper, the applicability of this theory is demonstrated through experimental tests.

2 Viability theory

In this section, some aspects of the viability theory are presented. The reader can refer to the first book on viability [1] for a more detailed presentation and to the last publication [2] for the last developments. This paper focuses on concepts that will be used later.

Viability theory defines mathematical metaphors illustrating the evolution of a system:

- Whose dynamics is not deterministic,
- Which obeys to the viability constraints,
- Which is controlled by feedback loops.

One of the goals of the theory is to find mechanisms for selecting regulons (or controls) to regulate the system. One of the selection mechanisms is based on the principle of inertia: the controls remain constant as the viability of system is not involved.

This theory has reached a level of abstraction that allows it to offer metaphors in very different fields, from economics to biology through Image Processing and cognitive sciences. Rather than presenting the theory from a general but complicated point of view, a didactic example is presented. This example will illustrate the main concepts that will be used afterwards.

2.1 A didactic example

In the following, a classical example of a controlled system with constraints on states and controls is presented.

$$\dot{x} = f(x(t), u(t)) \tag{1}$$

$$u(t) \in U(x(t)), \quad \forall t \geq 0 \tag{2}$$

$$x(t) \in K, \quad \forall t > 0 \tag{3}$$

with
 $u(t)$ is the control of the system.

$U(x(t))$ is the set of admissible controls. This set can depend on the state $x(t)$. K is the set on states which verify constraints (they are called viability constraints).

$f(x, u)$ is the equation of the dynamic system.

Viability theory deals with set-valued map which means that the image of a point is a set. Starting from this point on view, the following set-valued map is defined :

$$F(x) = \{f(x, u) \mid u \in U(x(t))\} \quad (4)$$

Then, the initial ordinary differential equation (1) becomes a differential inclusion :

$$\dot{x} \in F(x) \quad (5)$$

Definition 2.1 A state x_0 is viable for the system F if a control $u(t) \in U(x(t))$ exists such that the trajectory starting from x_0 controlled by $u(t)$ through the equation 3 remains in K .

Definition 2.2 K is viable for the system F if $\forall x \in K$, x is viable.

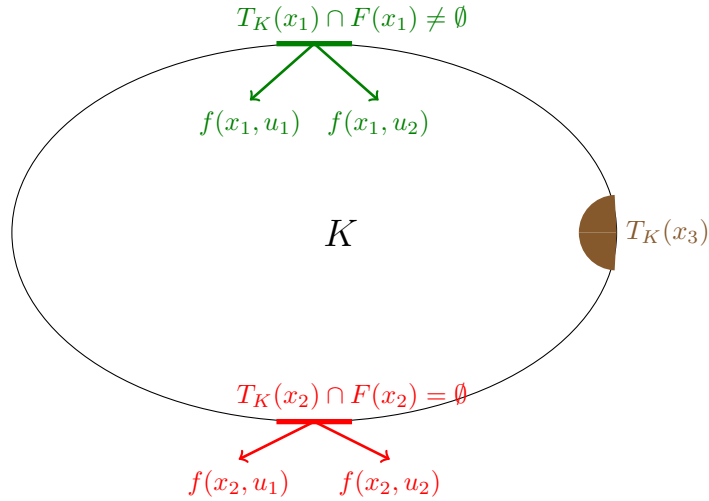


Figure 1: viability constraints, contingent cones, viability domain

One of the issues of viability theory is to verify that K is viable without the need to solve for each point x an ordinary differential equation. A geometric intuition is that to check that K is viable, it seems sufficient to focus on the

boundary of K . In fact, while the trajectory is in the interior of K , there is no problem of viability. A problem emerges when the trajectory reaches the boundary of K . At this point, the viability is ensured if a control enabling the system to return in K exists.

This intuition is illustrated in the figure 1 where viability constraints are represented by an ellipse. In this figure, the green and red arrows indicate the directions that can take the system starting from the origins of the arrows. Green directions point from the border to the interior of K , so the starting point is viable. In contrast, red directions do not return to K . If these directions are all possible commands, point to the origin of these arrows is not viable.

This geometric intuition is formalized mathematically by defining the tangent of a set. In viability, it is adopted the definition of Bouligand which introduced the concept of contingent cone.

Definition 2.3 *The contingent cone $T_K(x)$ from K at x is defined by :*

$$T_K(x) = \{v \in X \mid \liminf_{h \rightarrow 0^+} \frac{d_K(x + hv)}{h} = 0\} \quad (6)$$

- $d_K(x)$ is the distance between the set K and the point x . A deeper mathematical analysis of this distance is out of the scope of this paper.

The brown cone on the figure 1 is a graphical example of the contingent cone.

The regulatory function $R_K(x)$ is defined as the set of viable controls, i.e. leading to a direction belonging to the contingent cone. This function is a set-valued map (the image of a point is a set).

Definition 2.4 *The Regulation map $R_K(x)$ from K in x is defined by :*

$$R_K(x) = \{u \in U(x) \mid f(x, u) \in T_K(x)\} \quad (7)$$

K is a viability domain if for all x in K , the regulation map $R_K(x)$ is not empty, i.e. a control u exists such that $f(x, u)$ belongs to $T_K(x)$.

Definition 2.5 *K is a viability domain if $\forall x \in K, \exists u \in U(x) \mid f(x, u) \in T_K(x)$*

or shortly : K is a *viability domain* if $T_{Viab(K)}(x) \cap F(x) \neq \emptyset$

Main theorems of viability are about the minimal assumptions for an equivalence between :

- K is a viability domain,
- and K is viable.

The interest of these theorems is that they replace a property on the dynamics, the viability of a system, by a geometrical property, the viability domain.

In many applications, the set K is not viable. Then, the viability kernel $Viab(K)$ is defined as the set of viable states.

Definition 2.6 *The viability kernel $Viab(K)$ is defined by :*

$$Viab(K) = \{x_0 \in K \mid \exists u(t), x(t) \in K, \forall t\} \quad (8)$$

The “anatomy” of K is studied by breaking it down between the viability kernel and the complement of its kernel.

In the section 5, an algorithm for building a viability kernel is proposed. This algorithm is based on the property that a viability kernel is necessary a viability domain.

3 Vehicle dynamic modeling

3.1 Modeling

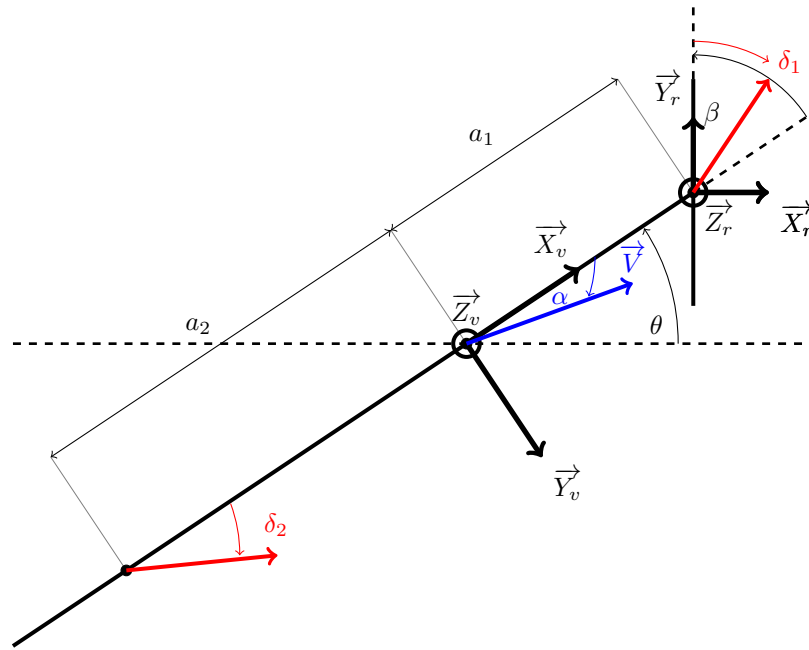


Figure 2: Two-wheels vehicle model in a cornering maneuver

In this part, the model of vehicle used for the application of the viability theory to road safety is presented. The application of this theory is not specific to this model and can be adapted to others models as a more realistic 4 wheels model.

A common model : the two-wheels vehicle model in a cornering maneuver was chosen. A not symmetric braking system is added to study the instability.

The figure 2 presents a top view of the model's geometry. This model comes from [3, p. 27] . The geometric variables are the following :

- θ is the yaw angle. The angle between the x abscissa of the road frame \vec{X}_r and the longitudinal axis of the vehicle \vec{X}_v .
- α is the vehicle sideslip angle. The angle between the longitudinal axis of the vehicle and the velocity vector of the centre of gravity \vec{V} . In the following, this angle is supposed to be small (for clarity's sake, small angles are exaggerated on the figure)
- δ_1 (resp. δ_2) is the front (resp. rear) wheel sideslip angle. $\tan(\delta_1)$ is the ratio of the lateral and the forward velocity of the front wheel. This angle depends on the transversal elasticity of the tyre.
- β is the front steer angle. This angle is supposed to be the same for the two front wheels. This angle is controlled by the driver.

The geometric parameters are :

- a_1 (resp. a_2) is the distance from the front wheels (rear wheels) to the centre of gravity G . $L (= a_1 + a_2)$ is the wheelbase of the vehicle.

Model's equations are given below. The complete explanation is given in [4]. The so-called " small angle " approximation is applied, i.e, sinus of small angle are replaced by the angle itself, cosinus of small angle are approximated by 1.

$$MV(\dot{\theta} + \dot{\alpha}) = -D_1\delta_1 - D_2\delta_2 \quad (9)$$

$$I_z\ddot{\theta} = -D_1a_1\delta_1 + D_2a_2\delta_2 - \frac{b}{2}iM\gamma \quad (10)$$

$$(11)$$

- M is the mass of the vehicle ;
- D_1 and D_2 are the cornering stiffness of tyres.
- V is the velocity, it is controlled by the driver according to the following equation :

$$\dot{V} = \gamma \quad (12)$$

- γ is controlled by the driver by the braking system.
- The expression $\frac{b}{2}iM\gamma$ is the irregularity momentum due to longitudinal actions.
- b is the front track width.

- i_r is the dissymmetrical coefficient. It represents the unbalance of the baking system between the right wheels and the left wheels. In the case of acceleration, this coefficient is set to zero.

For clarity's sake, a simple model of tyre is presented. In fact, the cornering stiffnesses D_1 et D_2 vary during braking, they cannot be supposed to be constant. In the algorithm implementation (section 6), the transversal force delivered by the Pacejka model ([5]) is used.

Cinematic considerations enable to compute δ_1 and δ_2 from the geometric variables α , β , $\dot{\theta}$ and V :

$$\delta_1 = \alpha + \frac{a_1 \dot{\theta}}{V} - \beta \quad (13)$$

$$\delta_2 = \alpha - \frac{a_2 \dot{\theta}}{V} \quad (14)$$

3.2 State space representation

In control engineering, dynamic systems are commonly described according to a state space representation. This representation is the following.

$$\begin{pmatrix} \dot{\alpha} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -\frac{D_1+D_2}{MV} & -(1 + \frac{D_1 a_1 - D_2 a_2}{MV^2}) \\ -\frac{D_1 a_1 - D_2 a_2}{I_{zz}} & -\frac{D_1 a_1^2 + D_2 a_2^2}{VI_{zz}} \end{pmatrix} \begin{pmatrix} \alpha \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} -\frac{D_1 \beta}{MV} \\ \frac{D_1 \beta a_1}{I_{zz}} + \frac{biM\gamma}{I_{zz}} \end{pmatrix}$$

$$\dot{V} = -\gamma \quad (15)$$

The controls of the driver are β , the steering angle, γ , the acceleration (or deceleration).

The previous representation is rewritten in order to emphasize the controls u :

$$\dot{x} = A(x) \cdot x + B \cdot u \quad (16)$$

The state x is :

$$x = \begin{pmatrix} V \\ \alpha \\ \dot{\theta} \end{pmatrix} \quad (17)$$

The matrix A is :

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\frac{D_1+D_2}{MV} & -(1 + \frac{D_1 a_1 - D_2 a_2}{MV^2}) \\ 0 & -\frac{D_1 a_1 - D_2 a_2}{I_{zz}} & -\frac{D_1 a_1^2 + D_2 a_2^2}{VI_{zz}} \end{pmatrix} \quad (18)$$

The matrix B is :

$$B = \begin{pmatrix} 0 & 1 \\ -\frac{D_1}{MV} & 0 \\ \frac{D_1 a_1}{I_{zz}} & -\frac{biM}{I_{zz}} \end{pmatrix} \quad (19)$$

It is not a linear model. The matrix A vary according to the state x : some coefficients of the matrix vary with V . Moreover, the cornering stiffnesses D_1 and D_2 vary with x .

The controls of the driver are :

$$u = \begin{pmatrix} \beta \\ \gamma \end{pmatrix} \quad (20)$$

4 Constrains

The vehicle dynamic equation 17 can be interpreted as the ordinary differential equation 1 of the viability theory. Now, the different constrains on the vehicle dynamic which defines a calm driving are explored. These constrains are directly interpreted in the viability theory as :

- constrains on the states : the set K in the equation 3
- constrains on the controls : the set $U(x)$ in the equation 2.

4.1 Constrains on the controls

4.1.1 Constrains on the steering velocity

Apart mechanically constrains on steering, a calm driving implies constrains on steering velocity.

$$|\dot{\beta}| < \dot{\beta}_{max} \quad (21)$$

4.1.2 Constrains on the acceleration and jerk

$$\gamma_{min} < \gamma < \gamma_{max} \quad (22)$$

$$|\dot{\gamma}| < \dot{\gamma}_{max} \quad (23)$$

4.2 Direct constrains on the state

The speed is, obviously, limited :

$$V_{min} < V < V_{max} \quad (24)$$

4.3 Indirect constrains on the state

4.3.1 Road

One obvious constrain is to avoid lane departure. This constrain is tackled by computing the Cartesian coordinates, x_1 , y_1 , and orientation (the yaw) θ from the Eulerian variables : α , the vehicle sideslip angle, and V , the velocity. Then, the belonging of these Cartesian coordinates to the road is verified. For the algorithm implementation, these Cartesian and orientation coordinates are added to the state x .

4.3.2 Stability

Another constrain is that the grip needed by the car have to be compatible with the grip offered by the road. These constrains are directly taken into account in the Pacejka model in the sense that the car slips and does not stay on road.

A more subtle constrain is the stability of the system. According to [6], the stability constrain when driver brakes cornering is :

$$\frac{L}{V^2} + K > 0 \quad (25)$$

- L is the wheelbase ;
- $K = \frac{M_1}{2D_1} - \frac{M_2}{2D_2}$, the understeer gradient ;
- M_1 (resp. M_2) is the static mass on the front (resp. rear) axle ;

$$M_1 = M \frac{a_2}{a_1 + a_2} \quad ; \quad M_2 = M \frac{a_1}{a_1 + a_2} \quad (26)$$

4.3.3 Remarks

- The viability set K includes the following constrains : the car does not slip, and the stability is ensured. These two conditions are also required for the modeling. The proposed model is no more valid when these conditions are violated. For example, the “small angle approximation ” is no more valid. The viability constrains ensure that the model is still valid.
- Some constrains can be added, others can be modified or cancelled according to the objectives of the application. The viability theory takes naturally into account constrains on the state of the system.

5 Viability kernel

Starting from a state of the vehicle, i.e. the vector x of the equation 17, the issue is : is-it possible to drive along the corner in a calm way or, in a more mathematical sense, does one smooth trajectory exists *without computing explicitly this trajectory* ? The viability theory assets that the existence of this trajectory is

equivalent to check that the state belongs to the viability kernel. This property replaces the resolution of a differential equation with the belonging of a state to a set.

This section presents algorithms in order to set up and to use a viability kernel. Before this presentation, the system has to be discretized.

5.1 Discretization

5.1.1 In time

The discretization of the equation 16 gives :

$$x_{n+1} = x_n + (A(x_n) \cdot x_n + B \cdot u_n) \Delta T \quad (27)$$

- ΔT is the sampling time.

5.1.2 In space

Every dimension of the state space are discretized. The state space is 3 (see the length of x in equation 17), but 3 dimensions for the spatial localization are added, and the two dimension of the control space are discretized too. Finally, the dimension of the proper space is 8.

The implementation of the viability theory faces the dimensionality curse : each additional variable, discretized in, for example, ten values, multiply by ten the number of states to be evaluated. This point is one of the current limit of this theory.

5.2 Algorithm Principle

The algorithm and the proof of its convergence were initially presented in [7]. It is a backward recurrence algorithm. It starts from the end of the trajectory (called the “target”). The “target” of a viable trajectory corresponds to the car at the end of the corner with an adapted speed, and a proper direction. The states of this target belong to the viability kernel. Then, states belonging to K from which, it is possible to reach the viability kernel with a constant control during ΔT , are added to the viability kernel. This step is applied recursively. This algorithm is based on the link between viability kernel and viability domain which is described in the section 2.

The figure 3 illustrates this procedure in two dimension (the number of dimension is 8 for the given application). Road limits are plotted by black arc circles. Blue points belong to the viability kernel, red points do not belong to it.

The red car does not belong to the viability kernel. Indeed, driver control which leads the vehicle into the viability kernel does not exist :

- either the control brings the car outside the road, so the constraints on the state are violated,

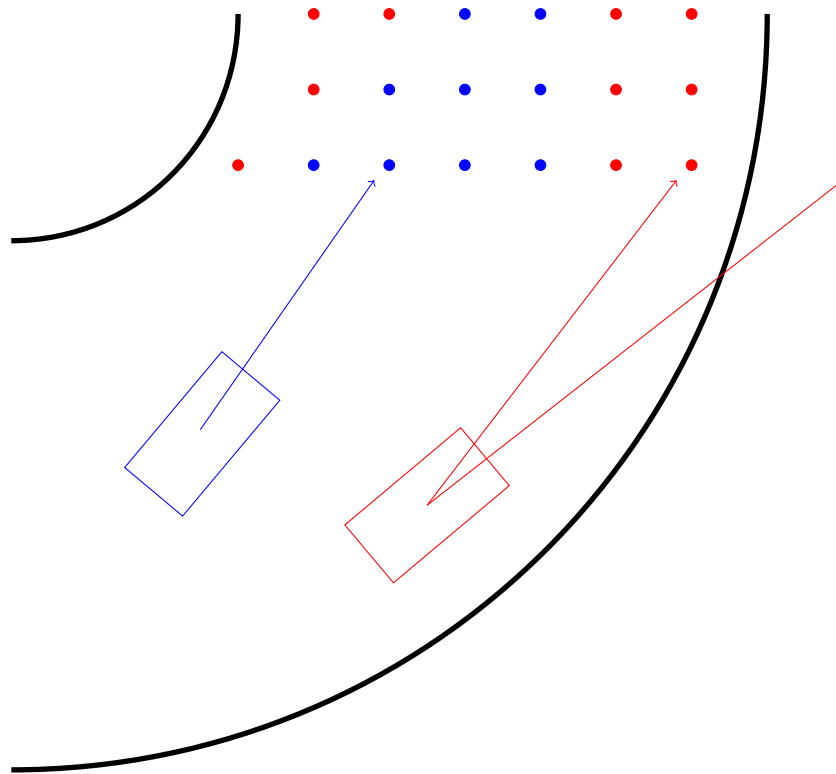


Figure 3: Algorithm principle

- or the control leads the car into a red point, or this point represents a state which does not belong to the viability kernel.

For this car, “soft”controls, which drive the car to the end of the corner, do not exist

On the contrary, a control drives the blue car to a blue point which belongs to the viability kernel. Therefore, it exist controls which drive the car to the target in a calm way. The blue car belongs to the viability kernel.

This paper focuses on the existence of at least one control to maintain the car on the road in a way compatible with calm driving. It could be of interest to count the number of different strategies of “soft” control. Higher is this number, more robust will be the given state.

5.3 Practical use : classification of a state

Knowing a state x_0 belonging to K , the issue of this subsection is to determine if x_0 belongs to the viability kernel of $K : Viab(K)$. $Viab(K)$ had been discretized with the algorithm described in the previous subsection. The previous algorithm gives two discretized sets. A set of states which belongs to the viability kernel (blue points) and a set of states which do not belong to the viability kernel (red points). To know if x_0 belongs to a set or to its complement is typically the phase of generalization for the machine learning methods.

One of the most fruitful and, relatively new, technique in this field is the Support Vector Machines (SVM). Support Vector Machines (SVM) were issued from the Vapnik Chervonenkis theory (the initial article is [8]). Due to their generalization performance on classification problems, SVM have obtained much success in a variety of domain. Unlike other machine learning methods, generalization errors of SVMs are not related to the input dimensionality of the problem, but to the margin with which a SVM separates data. The application of SVM to build viability kernel had been initially carried out in the ecology field ([9] and [10]). In the following, the principles of the method are recalled.

The viability kernel algorithm, described in the previous section, gives a set of example (x_i, y_i) ; where x_i is a state of K and $y_i = 1$ if $x_i \in Viab(K)$ and $y_i = -1$ if $x_i \notin Viab(K)$.

From this list, SVM define the decision function $d(\cdot)$ which separates the given examples :

$$d(x) = \sum_{i=1}^n \alpha_i y_i k(x_i, x) + b \quad (28)$$

- $k(y, z)$ is a kernel function which defines a scalar product between the projection of y and z in an other space than the initial space to which y and z belongs ;
- b is a constant.
- α_i is a scalar which is non null only for the vectors on the margins of the set. These vectors are called the Support Vector.



Figure 4: Controllability corner

The resolution of this problem introduces a regularization parameters C . This parameter is used to tune the trade-off between the number of misclassified examples and the number of examples defining the support vectors (i.e. examples x_i which are associated with non null α_i).

According with this decision function, a new state x_0 belongs to the viability kernel if $d(x_0) \geq 0$

6 Application

This section aims at proving the applicability of the viability theory to a real example in road safety. The scenario is the following : a car is in cornering maneuver at two different speeds. At the middle of the clothoid, the driver turns suddenly the wheel in order to simulate an inappropriate reaction. At low speed, this reaction is still compatible with a calm driving. At high speed, this reaction is no more compatible and the driver will not be able to maintain the car on the lane with soft controls.

In the following, the geometry of the infrastructure, the test vehicle, the maneuver and the algorithm implementation are described. The last subsection is devoted to the analysis of this maneuver with viability tool.

6.1 Infrastructure

The application is done on a real corner which is illustrated by the figure 4. This corner, called the “controllability curb”, is a part of the Ifsttar experimental track (located in Nantes, France). It is consisted of :

- a straight alignment (its length is 50m)
- a first clothoid (87m)
- an arc circle (Radius : 110m, its length is 134m)

M (kg)	I_z (kg · m ²)	i_r	b (m)	a_1 (m)	a_2 (m)	D_{10} (N)	D_{20} (N)
1485	2570	0.1	1.46	1.09	1.49	131500	117000

Table 1: Vehicle parameters

- a second clothoid which is the symmetric of the first one,
- a second straight alignment (50m)

6.2 Vehicle

The vehicle used during our trials is a sedan. It is a front wheel drive car equipped with a V12 engine coupled with a 5-speed automatic gear box. Ground link is composed by a pseudo Mac Pherson at the front and a flexible H-form torsion beam with trailing arms at the rear. The tires are classical summer tires.

Parameters of the vehicle needed by the algorithm are given in the table 1. D_{10} and D_{20} are the “nominal” tyre lateral stiffness. These figures are given in order to have a physical understanding of the system.

6.3 Maneuvers

1. the first maneuver consist in driving along the controllability curb at a constant speed of 60 km/h (37 mph) ;
2. the second maneuver is based on the first one, except that the driver turns the wheel suddenly along the first clothoid ; after this action, he has to maintain his vehicle in the lane ;
3. the third maneuver consist in driving along the controllability curb at a constant speed of 80 km/h (50 mph);
4. the fourth maneuver is a repetition of the second one at a speed of 80 km/h. In particular, the driver has to turn the wheel with the same angle.

The figure 5 illustrates the state of the test vehicle during these maneuvers.

6.4 Algorithm implementation

The code has been written with the Free and Open Source software scilab ([11]).

6.4.1 Viability kernel construction

The algorithm described in the subsection 5.2 has been applied to the “control-lability” curb. The direct constrains are the followings :

- the minimal velocity is 60 km/h, the maximal velocity is 90 km/h ;



Figure 5: Photos of the different maneuvers

- the acceleration has to be included between $-0.1 g$ and $0.1 g$ (it is the frame of a smooth driving) ;
- the maximal steering velocity is $\pm 3 \text{ deg/s}$;
- the maximal jerk $\pm 0.2 g/s$.

The discretization is

- $4 m$ along the curvilinear abscissa ;
- $0.5 m$ for the transversal axis ;
- $0.2 s$ for the time

It takes 51 mn to build the viability kernel (the computer is a classical old PC scoring a performance of 10.3 gigaflops according [12]) . 759 states belong to the viability kernel and 291471 do not belong to the viability kernel. Thus, about 3 % of the states belongs to the viability kernel. It includes small variation around “nominal” trajectories, i.e. trajectories at constant speed in the middle of the road. Every states with large deviation with these “nominal” trajectories belong to K but not to $Viab(K)$.

This application shows that applying Viability theory on real case needs computing power. It has been one of the barriers for its application on a real road until now.

The algorithm can be parallelized. Therefore, by using Graphics Processing Unit (GPU), this computing time could be dramatically reduced or the addressed model could be more complex with the same computing time.

This computation has to be done a priori, i.e. before the practical use (i.e. on board) of the viability kernel.

6.4.2 Viability kernel classification

Among the different available algorithms for SVM, the algorithm described in [13] is chosen because it is suitable to large data set problem. The kernel function is the Gaussian function which is a classical choice when the problem is nonlinear.

It takes 27 mn to get the decision function (the computer performance is 10.3 gigaflops). The regularization parameters C is set up in order to get 19 examples which are misclassified (19 states of the viability kernel are considered as not belonging to the viability kernel and all the states of the complementary of the viability kernel are well classified), and to get 1482 support vectors.

It means that to know if a state belongs to the viability kernel, it is enough to compute the equation 28 with $n = 1482$. The results is immediate with a modern computer.

6.5 Results

According to the feeling of the driver, the maneuvers 1, 2 and 3 are compatibles with calm driving but not the maneuver 4.

During the maneuver, the instant when the driver is turning the wheel is considered.

These four situations are modeled by the states given in the table 2 for the maneuvers at 60 km/h and table 3 for those corresponding to 80 km/h. The interpretation of this table is the following.

- The first three columns : x, y, θ are the Cartesian and orientation coordinates.
- The difference between the two states of the same speed is, mainly, the steering angle β .
- The last column is the value of the decision function which classes a state according to the equation 28. The sign of the value indicates if the state belongs or does not belong to the viability kernel.

According to this last information, the maneuvers 1,2,3 are detected as viable by the algorithm but not the maneuver 4. These results are coherent with the driver's feeling.

As expected, it means that the tolerance to an inappropriate control in the steering angle is more important at a low speed than at a higher speed. It implies that the frontiers of the viability kernel is nearer to the nominal trajectory at high speed than at lower speed.

	x (m)	y (m)	θ (rd)	α (rd)	β (rd)	$\dot{\theta}$ ($\frac{rd}{s}$)	V ($\frac{km}{h}$)	γ ($\frac{m}{s^2}$)	$d()$
viable	74.9	-1.7	-0.14	0	-0.017	-0.10	60	0	2.4E-6
viable	74.9	-1.7	-0.14	0	0.01	-0.10	60	0	4E-8

Table 2: Comparison of two states at 60 km/h

	x (m)	y (m)	θ (rd)	α (rd)	β (rd)	$\dot{\theta}$ ($\frac{rd}{s}$)	V ($\frac{km}{h}$)	γ ($\frac{m}{s^2}$)	$d()$
viable	74.9	-1.7	-0.14	0.01	-0.018	-0.13	80	0	1.5E-6
not viable	74.9	-1.7	-0.14	0.01	0.01	-0.13	80	0	-7E-8

Table 3: Comparison of two states at 80 km/h

This first investigation of the viability kernel shows that the study of the shape of the viability kernel is fruitful for analysing the road-driver-vehicle system. Critical states belongs to parts of the viability kernel which are “narrow” in a geometrical sense to the boundary. Viability theory replaces analysis of differential equation by analysis of a set. It replaces differential equations analysis by geometry analysis.

In this example, the value of the decision function $d(.)$ ranks the situation in a way coherent with our intuition. This suggests that this function can be analyzed as a distance to the frontiers of the viability kernel so as a ranking function. Thus, this function could be used to warn gradually the driver.

This algorithm can be implemented in roadside system camera. In this case, the dynamic model has to be simplified because the system can not access to the vehicle parameters, different viability kernels have to be computed according to weather conditions and viability kernels have to evolve with the road in order to take into account the road polishing.

It can also be implemented directly on the vehicle. In this case, the dynamic model can be richer by using all available information on CAN-bus. However, as the available grip between the tyre and the road cannot be known accurately by classical embedded measurement ([14]), this model cannot be very accurate.

The most efficient solution is to mix the two systems. The car updates its viability kernel thanks to information delivered by road systems.

7 Conclusion - Perspective

A viability kernel has been built for a road-car-driver system. This set includes trajectories compatible with a calm driving style. The system is modelled as a classical two-wheels model in a cornering maneuver including braking. A decision function is build by using the Support Vector Machine (SVM) technique. This decision function indicates if a state of the vehicle belongs or not to the viability kernel. The calculation of the viability kernel and the construction

of the decision function are done before to drive. During driving, the decision function verify if the state of the vehicle still belongs to the viability kernel. This operation consumes little computing time and can be computed on board. Experimental tests were carried out to prove the applicability of the proposed method.

It is the first time that the viability theory is applied to vehicle dynamics with road safety concern. This was possible through the evolution of computation power and recent progresses in classification techniques. Development of computation with Graphics Processing Units will make applications of this theory easier.

Beyond this immediate application, viability theory provides a new point of view on road safety: the issue is not to calculate optimal or robust controls but it is to know if theses controls exist, without calculating them in real time.

Perspectives are multiple. The decision function is used in a binary mode but it could also be analyzed as a distance to the frontier of the viability kernel and, thus, helps to deliver gradually warnings to the driver. The number of driving strategy to maintain the car on the road can be computed, this defines a system robustness. The most promising issue is the anatomy of the viability kernel. The shape of the viability kernel would define the critical states.

References

- [1] J.P. Aubin *Viability Theory*, Birkhauser, Boston, 1991.
- [2] J.P. Aubin, A. Bayen, and P. Saint-Pierre *Viability Theory: New Directions*, Springer, 2011.
- [3] H.B. Pacejka *Tyres and vehicle dynamics*, Butterworth-Heinemann, 2002.
- [4] J.R. Ellis *Vehicle Handling Dynamics*, Wiley-Blackwell, 1994.
- [5] H. Pacejka and E. Bakker, *Magic formula tyre model*, Vehicle System Dynamics 21 (1993), pp. 1 – 18.
- [6] J.P. Brossard *Dynamique du freinage*, PPUR presses polytechniques, 2009.
- [7] P. Saint-Pierre, *Approximation of the viability kernel*, Applied Mathematics & Optimization 29 (1994), pp. 187–209.
- [8] C. Cortes and V. Vapnik, *Support-vector networks*, Machine Learning 20 (1995), pp. 273–297.
- [9] L. Chapel, G. Deffuant, S. Martin, and C. Mullon, *Defining yield policies in a viability approach*, Ecological Modelling 212 (2008), pp. 10–15.
- [10] L. Chapel, *Maintenir la viabilité ou la résilience d'un système : les machines à vecteurs de support pour rompre la malédiction de la dimensionnalité ?*, Université Blaise Pascal - Clermont-Ferrand II, 2007.

-
- [11] S. Enterprises *Scilab: Free and Open Source software for numerical computation*, Orsay, France, 2012.
 - [12] J.J. Dongarra, *Performance of Various Computers Using Standard Linear Equations Software, (Linpac Benchmark Report)*, CS-89-85, University of Tennessee, USA, 2011.
 - [13] M. Tohme and R. Lengelle, *F-SVC: A simple and fast training algorithm soft margin Support Vector Classification*, in *IEEE Workshop on Machine Learning for Signal Processing, 2008. MLSP 2008*, Oct., , 2008, pp. 339–344.
 - [14] A. Andrieux, P.O. Vandanjon, R. Lengelle, and C. Chabanon, *New results on the relation between tyre–road longitudinal stiffness and maximum available grip for motor car*, *Vehicle System Dynamics* 48 (2010), pp. 1511–1533.