



**HAL**  
open science

# Toward Fully Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures

Lounis Adouane

► **To cite this version:**

Lounis Adouane. Toward Fully Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures. 11th International Workshop on Robot Motion and Control. Plenary session, Jul 2017, Wasowo, Poland. hal-01659679

**HAL Id: hal-01659679**

**<https://hal.science/hal-01659679>**

Submitted on 26 Feb 2018

**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.

# Toward Fully Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures\*

Lounis Adouane

Institut Pascal, UCA/SIGMA UMR CNRS 6602, France

e-mail: Lounis.Adouane@uca.fr

**Abstract**—This paper makes the focus on the way to increase gradually the autonomy of single vehicle as well as multi-vehicle systems (MVS) to achieve autonomous navigation in complex environments (e.g., cluttered, uncertain and/or dynamic). Its main objective is to give an overview of the developed generic control architectures (mainly decision/action aspects), and their different components, in order to enhance the safety, flexibility and the reliability of autonomous navigation. Furthermore, among the main ideas developed in this paper are related to the potentiality of using multi-controller architectures and their hybrid nature (continuous/discrete). Indeed, using this kind of control allows us to break the complexity of the overall tasks to be carried out while using bottom-up construction. This implies the development of appropriate reliable elementary controllers (obstacle avoidance, target reaching/tracking, formation maintaining, etc.), but also the proposition of specific mechanisms to manage the controllers' interactions, which ensures the respect of different constraints and enhancing metrics/criteria linked to the overall control reliability. Although the developed concepts/methods/architectures could be applied for different domains (such as service robotics or agriculture), the transportation-related purposes remains the privileged target. Two main examples of task achievement will be highlighted in this paper: appropriate flexible and reliable navigation based on sequential target reaching, and dynamic multi-vehicle navigation in formation.

## I. INTRODUCTION

In recent years, autonomous UGVs (Unmanned Ground Vehicles) are used in a multitude of tasks/domains, for instance: area surveillance [1], mapping of unknown environments [2], human search and rescue [3], exploration [4], military [5], agriculture [6], service robotics [7] or transportation [8]. Although the big interest of scientific as well as industrial community on these UGVs, the transportation domain remains among the most challenging and strategic topic knowing its high potential impact for both futuristic smart cities and manufacture of the future. Indeed, this transportation can touch people (private car or public transport) as well as goods transportation (in warehouses or ports, for instance).

In terms of autonomous cars, among the last main events which highly contribute to draw the interest of the international community let us cite the DARPA<sup>1</sup> Grand Challenges

\*This work was supported by the French National Research Agency (ANR) and ADEME (Agence de l'Environnement et de la Maîtrise de l'Energie) through: research programs Investissements d'avenir of RobotEx Equipment of Excellence (ANR-10-EQPX-44), the IMoBS<sup>3</sup> Laboratory of Excellence (ANR-10-LABX-16-01) and Businova Evolution project.

<sup>1</sup>Defense Advanced Research Projects Agency of USA.

that happened respectively in 2004 and 2007 [9] [10] [11]. The important rate of success of these challenges opens a large spectrum to define the actual possibilities of the driverless car for mid- and long-term prospects. Autonomous mobile robotics reach a milestone and automotive industries as well as new companies increase their interest for these new applications [12, Chapter 1]. In recent years, the development of fully autonomous vehicle in the transportation field has received even more attention from different countries [13]. One of the pioneering and important work which has drawn a lot of public attention to autonomous cars have been happen respectively: in 2010 with the Google driverless car [14] (cf. Figure 1(a)) and in 2013 with the BRAiVE (cf. Figure 1(b))<sup>2</sup> car from Parma University (Italy). These two vehicles allow fully autonomous driving in different contexts (rural / free-way / urban). Nowadays several important automotive manufacturers like Mercedes, GM, Ford, Nissan/Renault, Daimler, PSA, Audi or BMW announce to sell a driverless car at the mid-term horizon (less than 10 years), but before that, important challenges must be resolved in terms of perception-decision-action aspects [12, Chapter 1]. This paper will make the focus on the decision-action phases.

It is important to notice that the driverless car is not only synonym of a car as we commonly known but with the addition just of the automation of its displacement functions. In fact, in parallel with the developments of this area by automotive industries and certain laboratories, another generation of UGVs like EZ10 (cf. Figure 1(c)) or VIPALAB (cf. Figure 1(d)) aim also to autonomously transport passengers but in a more restricted area like midtown or inside big companies, amusement parks, airports, etc. which need autonomous shuttles between their different areas. This specific autonomous navigation function is among the most important applications started at Institut Pascal since 2000. Although the environment of navigation is generally delimited and the dynamic of UGV evolution is not the same as for the Google car for instance, nevertheless an important part of the autonomous navigation issues are shared. Indeed, this kind of vehicles must, like the Google car, navigate autonomously while taking into account the different events (e.g., traffic light, obstructing objects, etc.) which could be much dense and dynamic in midtown for instance.

While laboratories and innovative companies (like Google)

<sup>2</sup><http://www.braive.vislalab.it/>, consulted in May 2017.



(a) Google<sup>®</sup> Car



(b) BRAiVE vehicle



(c) EZ10<sup>®</sup> vehicle



(d) VIPALAB<sup>®</sup> vehicles

Fig. 1. Different autonomous vehicles with several use cases.

developed a driverless car from scratch, the automotive manufacturers proceed with incremental developments using ADAS (Advanced Driver Assistance Systems). Several sophisticated and reliable ADAS have already entered the market, such as Automatic Parking, Adaptive Cruise Control, Lane Keeping Assistance or Collision avoidance system [15]. This incremental (or bottom-up approach) is close in certain manner to what we claim in this paper though the use of multi-controller architectures (MCA) and their hybrid nature (continuous/discrete) (cf. section II). Indeed, among the main ideas developed in this paper are related to the potentiality of using MCA [12] to achieve fully autonomous navigation in complex environments (e.g., cluttered or/and dynamic). This implies first, impel to components' standardization (input/output) but also the development of appropriate reliable elementary controllers (e.g., obstacle avoidance, target reaching/tracking, formation maintaining, etc.), but also the proposition of specific mechanisms to manage the controllers' in-

teractions, which ensures the respect of different constraints related to the overall control structure. This paper makes thus the focus on the way to increase gradually the autonomy of single vehicle as well as multi-vehicle systems (MVS) to achieve fully autonomous navigation. Its main objective is to give an overview of the developed generic control architectures based on MCA (decision/action aspects), and their different components, in order to enhance the safety, flexibility and the reliability of autonomous navigation. Although the developed concepts/methods/architectures could be applied for different domains (such as service robotics or agriculture), this paper will make the focus on transportation-related application.

The rest of the paper is organized as follows. Section II presents: the genesis of using MCA, the adopted overall navigation framework and also an overview of the main elementary components composing MCAs. Sections III and IV are devoted to highlight the design of appropriate MCAs to achieve respectively two complex tasks: *flexible and reliable navigation based on sequential target reaching*, and *dynamic multi-vehicle navigation in formation*. This paper ends with some conclusions and further work.

## II. FROM BEHAVIORAL TO MULTI-CONTROLLER ARCHITECTURES

There exist in animals innate behaviors which could be qualified as atomic (or elementary) in the sense that they are not reducible to simpler behaviors (directly observable). In general, all animals' motor actions (coordination of a set of muscle activities) are included in this category. These behaviors are the building blocks with which the behavior of a higher level can be built and described [16] [17]. Further, autonomous mobile robots can have to perform several tasks, for instance going to a specific target (location) in the environment while avoiding obstacles and in certain cases while maintaining a formation (as targeted notably in the work given in [8]) and so on. In addition, these sub-tasks must also be achieved generally while guaranteeing multi-objective criteria to obtain for instance reliable and smooth robot navigation. All these sub-tasks and several criteria increase considerably the complexity to attain efficient autonomous robot navigation. To address this complexity (in terms of task definition and multi-objective criteria), the control architectures can be elaborated in a modular and bottom-up way as introduced in [18] and so-called behavioral architectures [19]. Behavioral control architectures are based on the concept that a robot can achieve a global complex task while using only the coordination of several elementary behaviors [16] [18] [17]. To tackle this complexity, behavioral control architecture decomposes the global control into a set of elementary behaviors/controllers (e.g., attraction to a target, obstacle avoidance, trajectory following, etc.) to better master the overall robot behavior. Indeed, each behavior can be tested either individually or collectively with other behaviors. The goal is to verify the reliability and the efficiency of the corresponding behavior to achieve each determined sub-task [20] [21].

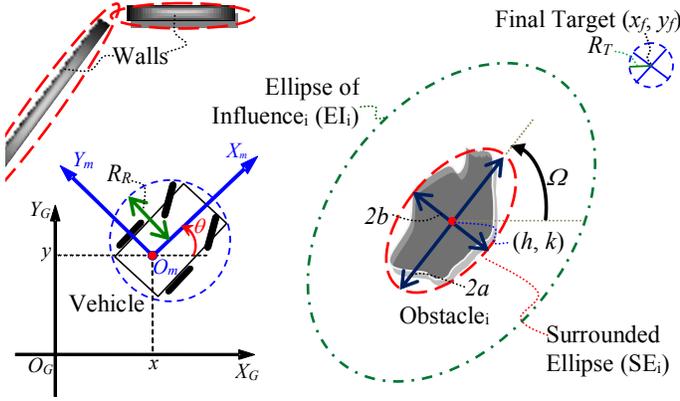


Fig. 2. Vehicle's pose and its elementary perceptions for performing reactive navigation. The parameters  $(h, k, a, b$  and  $\Omega)$  characterize the  $SE_i$  and the  $EI_i$  (cf. section II-B).

Nevertheless, several challenges remain to be addressed before obtaining an effective and reliable multi-controller architecture<sup>3</sup>. Among the main objectives of the works highlighted in this paper is to lead to stable and reliable multi-controller architectures while maintaining a high level of flexibility, necessary to tend toward fully autonomous vehicles in any complex situation.

It is presented in what follows first, the adopted overall navigation framework (cf. section II-A), secondly an overview of the used obstacle avoidance technique, based on limit-cycles, will be given (cf. section II-B). This latter is an important component for any autonomous navigation in complex environment. Third, an overview of the main elementary components composing MCAs will be highlighted (cf. section II-C).

#### A. Overall navigation framework definition

Autonomous vehicle navigation aims, in the proposed generic framework, to lead the vehicle from its initial configuration, to a final configuration while avoiding any obstacle (which could have different shapes, cf. Figure 2). This navigation could be done even with reactive control (while acting online according to the vehicle's local perception) or with cognitive control (while following an already planned trajectory) [22]. The desired vehicle's movement needs to be safe and smooth along all its displacement. One supposes in the setup that the vehicle and the final target to reach are surrounded by circle shapes with a radius  $R_R$  and  $R_T$  respectively (cf. Figure 2). For the obstacles/walls, it is supposed that they can be surrounded by appropriate ellipses (cf. Figure 2). Each ellipse is characterized by:  $(h, k)$  the coordinates of the ellipse's center w.r.t. global reference frame,  $(a, b)$  which correspond respectively to the major and minor elliptic semi-axes and  $\Omega$  gives the ellipse orientation.

The Surrounded Ellipse's (SE) parameters  $(h, k, a, b$  and  $\Omega)$  can be obtained by the vehicle either offline (using for

instance a road map of the static environment) or online using for example a camera positioned in the environment [23] or the vehicle's telemetric sensors [24]. Among the challenging aspects when the vehicle navigates in fully reactive way (thus with discovering online its environment), is to update smoothly and efficiently the ellipses' parameters as the vehicle discovers the entire shape of the obstacles. In order to perform this important perceptive functionality, an appropriate weighted least square method, on the range data given by telemetric sensors, has been used in [25]. An extension of this last approach while using Extended Kalman Filter and an appropriate sub-optimal heuristic method has been developed in [24] and [26].

#### B. Safe obstacle avoidance as an important component

Before to make the focus on multi-controller architectures, let us introduce in short the used "obstacle avoidance" controller. This kind of function is always an important primitive to perform autonomous vehicle navigation in complex environments. Thus, special attention should be taken for its development [27]. A multitude of methods exist in the literature to deal with obstacle avoidance, among them those based on: artificial potential field [28], Voronoï diagrams [29], visibility graphs [30], navigation functions [31], Rapidly-exploring Random Tree (RRT) [32] or Deformable Virtual Zone (DVZ) [33]. Each of them have their interests and drawbacks [27] [22]. It is used mainly in our works limit-cycles approach [34] [35] [36][37] [38] [26] [22]. The navigation methodologies based on limit-cycles have been used in the literature to perform mainly intuitive and efficient obstacle avoidance behavior. They are defined according to a circular [36], elliptic [37] or parallel to ellipse [22] periodic orbits. These periodic orbits can guarantee, if they are well-dimensioned (far enough from any obstacle) and accurately followed, to avoid any obstructing obstacle. It is to be noted that the used overall strategy for obstacle avoidance remains almost the same while using Elliptic Limit-Cycles (ELC) or Parallel ELC (PELC). Without being exhaustive and to avoid too complex developments it is given below the formulation of ELC.

1) *ELC mathematical formulation*: The differential equations of ELC are given by:

$$\begin{aligned} \dot{x}_s &= r(By_s + 0.5Cx_s) + \mu x_s(1 - Ax_s^2 - By_s^2 - Cx_s y_s) \quad (1) \\ \dot{y}_s &= -r(Ax_s + 0.5Cy_s) + \mu y_s(1 - Ax_s^2 - By_s^2 - Cx_s y_s) \quad (2) \end{aligned}$$

with  $(x_s, y_s)$  corresponds to the position of the vehicle according to the center of the ellipse;  $r = \pm 1$  according to the avoidance direction (clockwise (+) or counter-clockwise (-) respectively) (cf. Figure 3);  $\mu \in \mathbb{R}^+$  a positive constant value which allows us to modulate the convergence of the ELC. The convergence is as slow as  $\mu$  is smaller, which allows us also to obtain smoother ELC (Figure 5 highlights the influence of  $\mu$  in the case of PELC). The variables  $A$ ,  $B$  and  $C$  are given by:

$$A = (\sin(\Omega)/b_{lc})^2 + (\cos(\Omega)/a_{lc})^2 \quad (3)$$

$$B = (\cos(\Omega)/b_{lc})^2 + (\sin(\Omega)/a_{lc})^2 \quad (4)$$

$$C = (1/a_{lc}^2 - 1/b_{lc}^2) \sin(2\Omega) \quad (5)$$

<sup>3</sup>The term multi-controller will replace in what follows, the term behavioral because it has been wildly investigated in our works the use of automatic control theory to confirm among others the reliability of each controller as well as the overall multi-controller architecture [12].

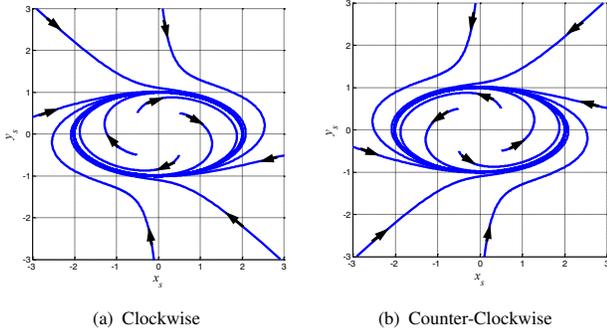


Fig. 3. Shape possibilities for the used Elliptic Limit-Cycles (ELC).

where  $a_{lc}$  and  $b_{lc}$  characterize respectively the major and minor elliptic semi-axes and  $\Omega$  gives the ellipse orientation.

Figure 3 shows that the ellipse of a major axis =  $2a_{lc} = 4$  and of minor axis =  $2b_{lc} = 2$  is a periodic orbit. Figure 3(a) and 3(b) show the shape of equations (1) and (2) when  $r = 1$  and  $-1$  respectively. They show the direction of trajectories (clockwise or counter-clockwise) according to  $(x_s, y_s)$  axes. The trajectories from all points  $(x_s, y_s)$  of  $X, Y$  reference frame, including inside the ellipse, move towards the ellipse. This can be demonstrated mathematically while using Lyapunov synthesis.

It is to be noted that the case of Parallel Elliptic Limit-Cycle (PELC) addressed in [22] lead to much complex formulation of equations (1) and (2) since it is targeted to have orbit which is always parallel to the ellipse (constant distance w.r.t. the ellipse border) which surround the obstacle. This leads to equations of 8<sup>th</sup> order. It is also important to notice that these limit-cycles could be used as local path planner (to avoid the closest obstacle) or global planner. As global planner, the aim is to find the optimal (or sub-optimal) global path leading the vehicle from its initial configuration until a final configuration while avoiding all the hindering obstacles (cf. Figure 4 for an example of global planning using PELC).

2) *References frame linked to the task achievement:* For simple and efficient description of vehicle navigation in any kind of environment, it is presented in what follows a specific reference frame assigned for each obstacle / wall / (or any object which could obstruct the vehicle's movement) inside the considered environment (or at least for each element inside the vehicle's field of view). These specific reference frames will guide the vehicle behaviors and allows the vehicle to evaluate the success of the current achieved sub-task (e.g., wall following, obstacle avoidance, etc.) [22]. Each elementary reference frame will orient thus locally the achievement of the vehicle navigation toward its final objective. A kind of analogy could be established with robot manipulator modeling. In fact, when we would like to control the movement of a robot end-effector (w.r.t. its base), it is assigned for each articulation an appropriate reference frame while using dedicated conventions [39] [40]. These local

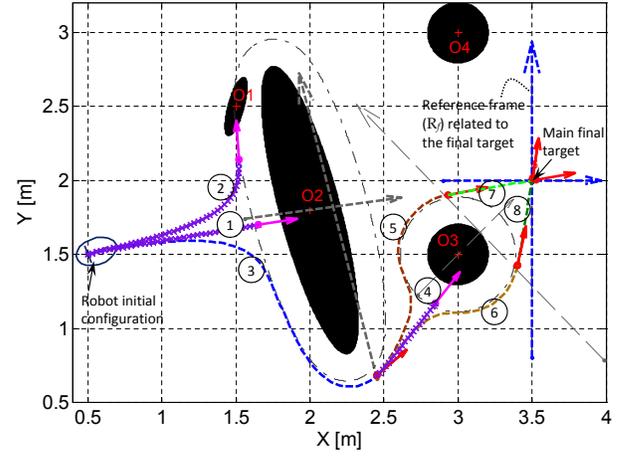


Fig. 4. Example of simple simulation highlighting the use of several PELC to reach the final assigned target. PELC paths given by “+” symbols mean invalid PELCs. The others elementary PELC in dotted colored lines show valid PELCs. The arrows given in solid lines correspond to the final PELCs' configurations. The numbering correspond to the successive computed PELCs in order to obtain the global optimized path based on PELC [22].

reference frames are mainly used to express simply the local elementary articulations' movements (translation / rotations) in order to obtain the desired final end-effector movement. The context of vehicle navigation is obviously different but the proposed reference frames will similarly lead to make a reasoning on the efficiency of the vehicle movements in order to reach its final objective.

The reference frame ( $\mathfrak{R}_{OT}$ ) attributed to each obstacle will allow to evaluate the obstacle avoidance sub-task achievement while knowing the relative vehicle's localization according to it.  $\mathfrak{R}_{OT}$  is obtained with a simple geometric construction and has the following features [22] (cf. Figure 5):

- $X_{OT}$  axis connects the center of the obstacle  $(x_O, y_O)$  to the center of the final Target  $(x_f, y_f)$ . This axis is oriented toward this target.
- the  $Y_{OT}$  axis is defined by two points  $PT_1$  and  $PT_2$ , which correspond to the tangent points between the two straight lines coming from the final target  $(x_f, y_f)$  and the Parallel Ellipse of Influence (PEI).  $Y_{OT}$  axis is oriented while following trigonometric convention.

To guide the vehicle's future movements it is important to define its localization w.r.t.  $\mathfrak{R}_{OT}$ . One needs, therefore, to make a transformation from the global reference frame  $X_G Y_G$  to the local reference frame  $X_{OT} Y_{OT}$  [22]. Once the transformation is obtained, it is enough for instance to check the sign of the vehicle's localization according to the axis  $X_{OT}$  to assign the vehicle appropriate behavior. For instance, if the sign of  $x_{RO}$  is negative, the vehicle must follows the defined limit-cycle (to avoid the obstacle) and if positive the vehicle can consider that the obstacle is not a hindering obstacle and can go thus straight toward its final target (cf. Figure 5). At the condition obviously that there is no other constrained obstacle; if not, the process will be reiterated. In

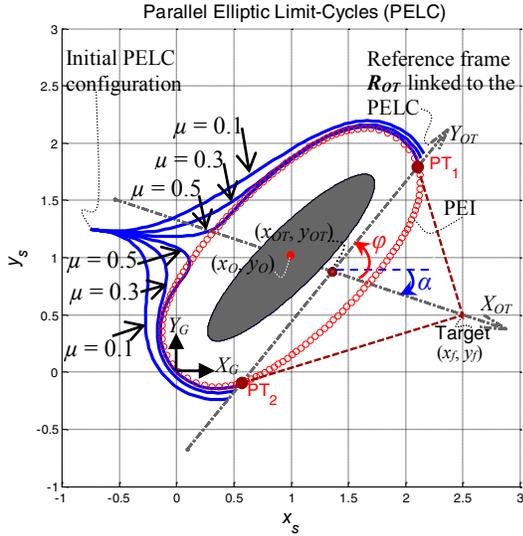


Fig. 5. Reference frame linked to each PELC and different PELC shapes according to the value of  $\mu$ ,  $\alpha$  and  $\varphi$  characterize the reference frame linked to each obstacle [22].

previous work given in [36] the sign of  $y_{RO}$  (ordinate of the robot in  $\mathcal{R}_{OT}$ ) has been used to determine the right direction to avoid the obstacle. If  $y_{RO} \geq 0$  then apply clockwise limit-cycle direction else apply counter-clockwise direction. This simple rule allows to reduce the length of robot trajectory in the case where the obstacle is surrounded by a circle. In the work given in [22] the presented reference frame has been served as an important component to determine the best direction and shape of local planning using PELC and global planning using an optimal sequence of PELC to reach the final target.

After introducing the global idea to obtain the limit-cycles which should be followed by the vehicles to avoid obstacles, let us highlight the main structures characterizing MCAs and their main components to achieve autonomous vehicle navigation.

### C. Multi-controller architectures' main structures

The main aim of multi-controller architectures is to have a bottom-up construction of the principal functionalities of autonomous vehicle navigation, while maintaining a high level of flexibility and reliability of the achieved complex tasks [12]. It will be presented, in what follows, two simple multi-controller architectures (cf. Figure 6) which could serve as basic structures to highlight the main components to perform flexible and reliable navigation in cluttered environments.

Before highlighting the main difference between the two control structures depicted in Figure 6, it is important to give the definition of the robot's elementary behavior (controller). In what follows each robot's controller is constituted by a dedicated set-point and stable control law blocks which allow us to achieve safely and reliably the desired robot's behavior. As shown in Figures 6(a) and 6(b) the two structures 1 and 2, are different in terms of the used control laws. In fact, structure 1 has two distinct control laws whereas structure

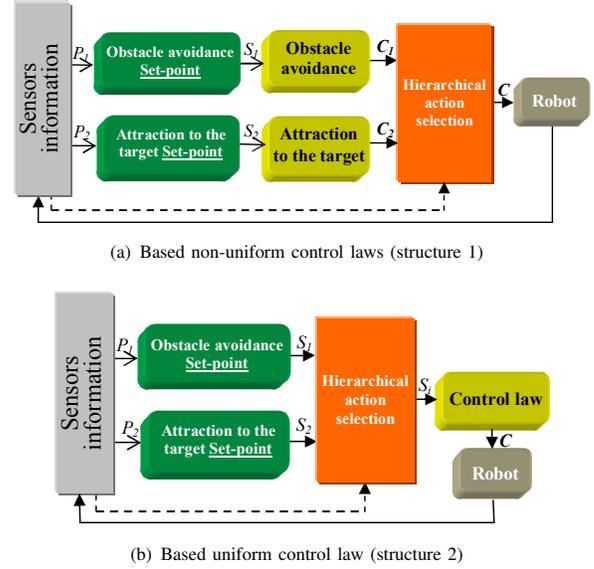


Fig. 6. The two main used multi-controller structures.

2 has only one common control law shared by the two set-point blocks. It is interesting to notice that structure 1 is the less restrictive architecture, in the sense that both controllers can have completely different set-points and control laws definition. It is enough to have two already stable elementary controllers to integrate them in this structure, without any harmonization of the used set-points or control laws. The possible drawback of this kind of multi-controller architecture corresponds to its difficulty in having a simple analysis of the overall control architecture stability (since it could use non-uniform control laws) [12, Chapter 3].

The two simple multi-controller architectures depicted in Figure 6 allow us to manage the interaction between different elementary blocks. The main features of each block composing these architectures are detailed below.

1) *Sensor information block*: While using robot's sensors and any already known data on the environment (using a road-map for instance), this block is in charge of detecting / localizing / characterizing any important features in the environment. Mainly this block, in the case of the basic architectures given in Figure 6, must provide the list of all perceived obstacles and the relative final target localization w.r.t. the robot (cf. Figure 2).

2) *Controllers' coordination block*: The coordination of the several controllers constituting a multi-controller architecture is among the most important aspect to master in order to obtain coherent and reliable architecture. Generally once the stability of each elementary controller is proved, they are gathered in specific multi-controller architectures to perform complex tasks. The objective is to ensure in addition, the stability and the smoothness of the overall control. This can be done mainly if the coordination between the elementary controllers is mastered [12, Chapter 3].

There exist two major principles of controller coordination: *action selection* and *fusion of actions*. Even if *fusion of actions* process gives very interesting robot behaviors, as

it has been shown in [21] (using a kind of schema motor principle); in [41] (using Multi-Agent System) or in [42] (using Fuzzy Logic principles), nevertheless the stability of the overall control architecture remains very complex, even impossible to demonstrate. However, control architectures based on the *action selection* process are relatively much easier to demonstrate even when switches between controllers occur [43] [44] [45]. However, the challenge is, in addition to the overall stability, to guarantee control smoothness. In fact, during mainly the phase of switching between controllers, the robot's set-points or the control could be subject to jerking/discontinuities/oscillations, the objective of the Controllers' coordination block is therefore to avoid (or at least minimize) these drawbacks to obtain finally reliable and smooth robot navigation [46] [23] [47] [48]. More details on the hybrid behavior (continuous/discrete) of the overall multi-controller architectures in order to guarantee at the same time, the overall control stability and the smoothness of the switch between controllers is emphasized in [12, Chapter 3].

### 3) Set-point blocks and their homogeneous definition:

Set-point blocks (cf. Figure 6), which have as input the perceptions  $P_i$  coming from the sensor information block, are responsible to give for each dedicated controller, the appropriate set-points for its operation. The design of multi-controller architecture aims to decompose the overall complex task into a multitude of sub-tasks to achieve (e.g., target reaching, wall following, obstacle avoidance, etc.) (cf. section II). According to these elementary sub-tasks, performed in a reactive or cognitive way [22], it has been noted in general that the robot must follow/track a path/trajectory or reach/track a specific target. This section aims to propose a homogeneous set-points definition for the multitude of the robot's navigation sub-tasks in order to simplify the design of control architectures.

It will be described in sections III and IV the use of static and/or dynamic targets to lead to a much more flexible way to define the robot's sub-tasks. It is promoted therefore in what follows the use of target set-points, defined by a pose  $(x_T, y_T, \theta_T)$  and a velocity  $v_T$ . The following subsections (II-C.3.a to II-C.3.c) will highlight the fact that this set-point formulation is generic enough to define an important number of the robot's behaviors. It is to be noted that, once the set-points are defined, at each sample time, it is important to have reliable control laws to reach/track these assigned set-points. To do that, one of the reliable control laws defined in section II-C.4 will be used to stabilize the errors to zero. This corresponds thus to the scheme of control given by structure 2 (cf. Figure 6(b)).

*a) Target tracking set-points based on global planned path:* The first identified case corresponds to the one where a global path is already defined using for instance a PELC [22]. In fact, in certain situations (e.g., static environment) it is enough for the robot to follow the path as accurately as possible without modifying its initial planning. In that situation, a Frenet reference frame is used [49] to extract the robot's set-points. The target set-point, at each sample time,

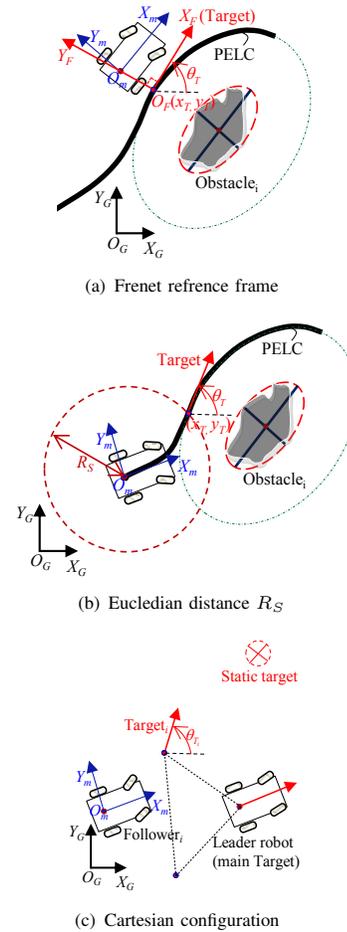


Fig. 7. Set-points definition based on (a) global planned path, (b) local planned path, (c) general static/dynamic target.

is given by (cf. Figure 7(a)):

- A position  $(x_T, y_T)$  corresponding to the closest position in the pre-planned path w.r.t. the origin of the reference frame  $X_m Y_m$ .  $(x_T, y_T)$  point corresponds to the origin of Frenet reference frame  $X_F Y_F$ .
- An orientation  $\theta_T$  corresponding to the tangent of the path w.r.t.  $X_G Y_G$  reference frame.
- A velocity  $v_T$  which could be constant or variable indifferently.

*b) Target tracking set-points based on local planned path:* Here, the set-point configurations are taken within the generated PELC trajectories but the same principle could be used for any other online local generated trajectory obtained from local planners.

When the environment is not very well known or dynamic, it is better to navigate reactively. In that situation, the current PELC takes as initial configuration, and at each sample time, the current robot configuration. The target set-point is given by (cf. Figure 7(b)):

- A position  $(x_T, y_T)$  corresponding to the intersection between the circle (which has as origin the origin of the reference frame  $X_m Y_m$  and as radius  $R_S$ ) and the planned PELC.

- An orientation  $\theta_T$  corresponding to the tangent to the PELC w.r.t.  $X_G Y_G$  reference frame at the intersection point  $(x_T, y_T)$ . If  $R_S = 0$ , the robot has to apply only an orientation control. Indeed, since the robot is already on the current computed PELC, the robot has only to control its heading w.r.t.  $\theta_T$ . This simple control has been used in [36] and [37].
- A velocity  $v_T$  which could be constant or variable indifferently.

c) *General target reaching/tracking set-points*: The last identified case (cf. Figure 7(c)) corresponds to the general situation where the robot must reach/track a static/dynamic target  $(x_T, y_T, \theta_T, v_T)$ . The sub-tasks which can deal with this kind of target definition correspond to all the cases where the set-points are not restricted to evolve inside a specific path. For instance, let us cite:

- For a static target, the set-points could correspond to the final robot destination as given in Figure 2. They could also correspond to an appropriate waypoints in the environment through which the robot must cross sequentially [8] (section III will give an example showing this navigation strategy).
- For a dynamic target, this kind of target set-point can serve for the Follower robot (as depicted in Figure 7(c)) to track a secondary target referenced w.r.t. the Leader. Section IV, focused on multi-robot systems, will highlight better this kind of target set-point definition.

4) *Uniform used control law*: Before presenting the used control law, it is important to know the robot's model. The used robot corresponds to a tricycle vehicle [50] modeled according to the well-known kinematics model given by equation 6.

$$\begin{cases} \dot{x} = v \cos(\theta) \\ \dot{y} = v \sin(\theta) \\ \dot{\theta} = v \tan(\gamma)/l_b \end{cases} \quad (6)$$

where  $(x, y, \theta)$  is the posture (configuration state) of the vehicle at the point  $O_m$  (origin of the local reference frame  $X_m Y_m$  linked to the vehicle (cf. Figure 8)),  $\gamma$  is the orientation of the equivalent front wheel (cf. Figure 8),  $v$  is the linear velocity of the vehicle at  $O_m$  and  $l_b$  is the vehicle's wheelbase.  $v$  and  $\gamma$  are the two control inputs of the vehicle (cf. equations 10 and 11 respectively). According to Figure 8,  $w_b$  corresponds to the track width of the vehicle and  $I_{cc}$  the instantaneous center of curvature of the vehicle trajectory. The radius of curvature  $r_c$  is given by:

$$r_c = l_b / \tan(\gamma) \quad (7)$$

and  $c_c = 1/r_c$  is the curvature of the vehicle trajectory.

The used control law [8] aims to drive the vehicle toward specific targets (static or dynamic) in the environment. At each sample time the tracked target is defined by a posture  $(x_T, y_T, \theta_T)$  and a velocity  $v_T$  (this velocity could be  $= 0$  if the target is static). As it was shown in subsection II-C.3, different vehicles behaviors are described with a uniform way where the vehicle has to reach/follow/track a specific target set-points. In order that this paper becomes at maximum

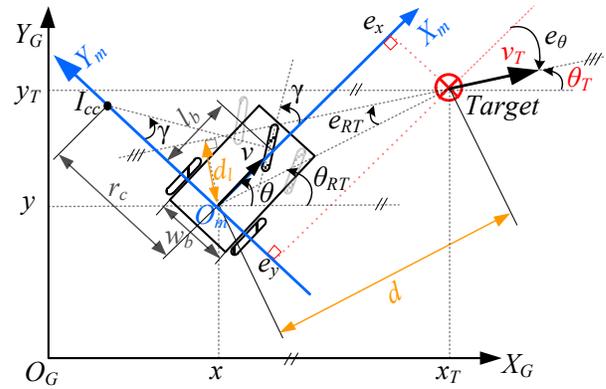


Fig. 8. Vehicle and target configuration in Global and Local reference frames. Control variables according to Lyapunov synthesis are also shown.

self-contained, let us give in summary the main elements of synthesis, using a Lyapunov formulation [51], the used control law [8]. The adopted Lyapunov function  $V$  is given by equation (8) (cf. Figure 8):

$$\begin{aligned} V &= \frac{1}{2} K_d d^2 + \frac{1}{2} K_l d_l^2 + K_o [1 - \cos(e_\theta)] \\ &= \frac{1}{2} K_d d^2 + \frac{1}{2} K_l d^2 \sin^2(e_{RT}) + K_o [1 - \cos(e_\theta)] \end{aligned} \quad (8)$$

where the initial values of  $e_{RT}$  and  $e_\theta$  must satisfy the following initial conditions:

$$e_{RT} \in ] -\pi/2, \pi/2[ \quad \text{and} \quad e_\theta \in ] -\pi/2, \pi/2[ \quad (9)$$

The Lyapunov function (8) is therefore a function of three parameters which depend on: the distance  $d$  between the target and vehicle's position; the distance  $d_l$  from the vehicle to the target line (line that passes through the target position with an orientation equal to the target orientation), this term is related to the Line of Sight and Flight of the target [52]; and the orientation error  $e_\theta$  between the vehicle and the target.

The desired linear velocity  $v$  and the front wheel orientation  $\gamma$  of the vehicle which allows to asymptotically stabilize the error vector  $(e_x, e_y, e_\theta, (v - v_T))$  toward zero (permitting therefore to have  $\dot{V} < 0$ ) are given by:

$$v = v_T \cos(e_\theta) + v_b \quad (10)$$

$$\gamma = \arctan(l_b c_c) \quad (11)$$

where  $v_b$  and  $c_c$  are defined by:

$$v_b = K_x [K_d e_x + K_l d \sin(e_{RT}) \sin(e_\theta) + K_o \sin(e_\theta) c_c] \quad (12)$$

with:

$$\begin{aligned} c_c &= \frac{1}{r_{cT} \cos(e_\theta)} + \frac{d^2 K_l \sin(e_{RT}) \cos(e_{RT})}{r_{cT} K_o \sin(e_\theta) \cos(e_\theta)} + K_\theta \tan(e_\theta) \\ &\quad + \frac{K_d e_y - K_l d \sin(e_{RT}) \cos(e_\theta)}{K_o \cos(e_\theta)} + \frac{K_{RT} \sin^2(e_{RT})}{\sin(e_\theta) \cos(e_\theta)} \end{aligned} \quad (13)$$

$\mathbf{K} = (K_d, K_l, K_o, K_x, K_\theta, K_{RT})$  is a vector of positive constants defined by the designer. Accurate analysis of this stable and efficient control law is given in [53] and [8].

Knowing the used common control law permitting to reach any static or dynamic target with stable way, let us present in what follows, how to perform different navigation sub-tasks using appropriate multi-controller architectures.

### III. FLEXIBLE AND RELIABLE NAVIGATION BASED ON SEQUENTIAL TARGET REACHING

This section emphasizes in summary the fact that it is not absolutely mandatory (as commonly admitted and broadly used in the literature) to have a predetermined trajectory to be followed by a robot to perform reliable and safe navigation in an urban and/or cluttered environment [12, Chapter 1]. It is presented in what follows the idea to use only a set of waypoints, appropriately disposed in the environment, to perform such navigation. The use of only a discrete number of waypoints in the environment will allow us even more flexibility of the vehicle's movements, since it is allowed to perform more maneuvers between waypoints, while remaining obviously safe (non-collision of the vehicle w.r.t. the road limits or any obstructing obstacle). Hence, navigation using only waypoints allows us to avoid any path/trajectory planning which could be time-consuming and complex, mainly in cluttered and dynamic environments. Moreover, this kind of navigation does not require knowledge of the pose of the closest point to the followed trajectory (w.r.t. the robot configuration) and/or the value of the curvature at this point [54]. Consequently, the navigation problem is simplified to a waypoint-reaching problem, i.e., the vehicle is guided by waypoints instead of following a specific fixed path [8]. Moreover, it is important to notice that if the successive waypoints are closer to each other, then the vehicle tends to perform a path-following navigation. The proposed technique tends therefore to gather the different navigation techniques. In addition, the use of only waypoints to control the vehicle instead of a fixed trajectory, allows the robot to carry out local operations (to avoid such obstacle) while maintaining overall stability of the used hybrid multi-controller architectures [12, Chapter 3].

#### A. Problem statement and Task modeling

The proposed navigation strategy uses a sequence of  $N$  sorted waypoints appropriately disposed in the environment. The aim of this sequence is to guarantee safe and flexible robot navigation. Each waypoint  $T_j = (x_{T_j}, y_{T_j}, \theta_{T_j}, v_{T_j})$  corresponds to a specific key configuration in the environment (cf. Figure 9).  $T_j$  is characterized by:

- A position  $(x_{T_j}, y_{T_j})$ .
- An orientation  $\theta_{T_j}$  such as:

$$\theta_{T_j} = \arctan((y_{T_{j+1}} - y_{T_j}) / (x_{T_{j+1}} - x_{T_j})) \quad (14)$$

where:  $(x_{T_{j+1}}, y_{T_{j+1}})$  corresponds to the position of the next target  $T_{j+1}$ .  $T_j$  orientation is therefore always oriented toward the waypoint  $T_{j+1}$ .

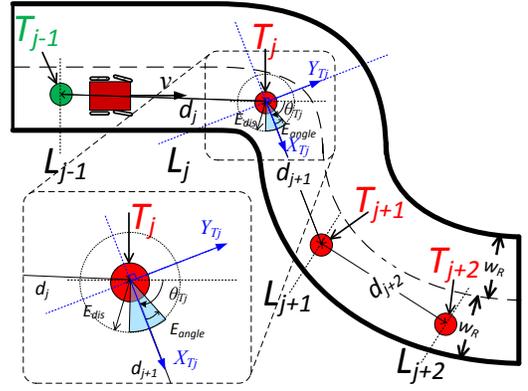


Fig. 9. Description of waypoints assignment.

- A velocity  $v_{T_j}$ . It is important to mention that to perform the proposed navigation by reaching sequential waypoints (targets), it is mandatory to reach each target (except the final waypoint) with a velocity  $v_{T_j} \neq 0$  to not have a jerky vehicle movement, at the starting and the arrival phase for each waypoint. The overall vehicle navigation becomes therefore smoother without oscillations in terms of linear velocity.

Different methods to obtain the appropriate set of waypoints (target set-points  $((x_{T_j}, y_{T_j}, \theta_{T_j}, v_{T_j}) |_{j=1 \dots N})$ ) are presented in [55]. They are based either on a heuristic method or on optimal multi-criteria optimization.

To define the robot's navigation strategy between the successive waypoints (cf. subsection III-B), an orthogonal reference frame  $X_{T_j} Y_{T_j}$  (cf. Figure 9) is attributed to each waypoint, where:

- the  $X_{T_j}$  axis connects the position of  $T_j$  to the following waypoint  $T_{j+1}$  and oriented toward  $T_{j+1}$ , and
- the  $Y_{T_j}$  axis is perpendicular to  $X_{T_j}$  and is oriented while following trigonometric convention.

This reference frame will be used in subsection III-B to perform the target assignment process. In addition, to ensure safe robot navigation between successive waypoints, each waypoint is assigned upper error bounds defined by  $E_{dis}$  and  $E_{angle}$  (cf. Figure 9). They correspond respectively to the maximal distance  $d$  and angle  $e_\theta$  errors between the robot and the target when it crosses the axis  $Y_{T_j}$ . Further,  $E_{dis}$  and  $E_{angle}$  correspond to a kind of maximal error tolerance when the robot reaches the target  $T_j$ . This tolerance is notably related to the inaccuracies of the robot localization and/or to the performance of the used control law. The maximum authorized values of  $E_{dis}$  and  $E_{angle}$  allow us to keep reliable robot navigation toward the target  $T_j$  (cf. Figure 9) while guaranteeing the appropriate robot configuration to reach the next target  $T_{j+1}$  [8], and so on.

#### B. Proposed MCA and Experimental validation

To perform the navigation based on sequential target reaching, the multi-controller architecture depicted in Figure 10 is used. This architecture is composed of several blocks:

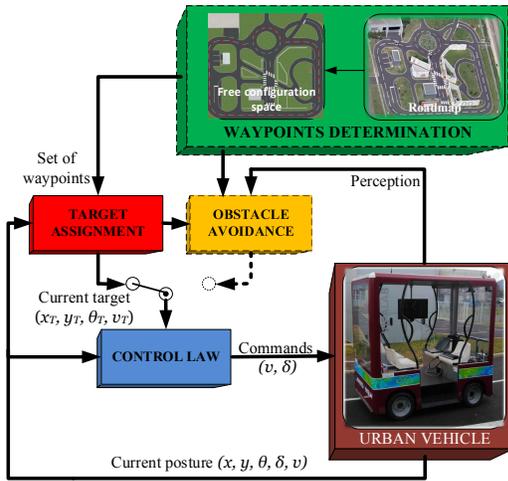


Fig. 10. Proposed multi-controller architecture to perform autonomous vehicle navigation based on sequential target reaching [8].

- The “Waypoint determination” block (dashed green box in Figure 10) obtains the set of appropriate waypoints configuration [55].
- The “Obstacle avoidance” block is activated when an obstacle obstructs the robot’s movement toward its current assigned waypoint. The used obstacle avoidance is based on limit-cycle technique (as given in subsection II-B) and allows us to avoid locally any obstructing obstacle.
- The “Control law” block ensures asymptotic stability to reach the current assigned waypoint  $T_j(x_{T_j}, y_{T_j}, \theta_{T_j}, v_{T_j})$ , as given in subsection II-C.4, where the control law can perform either static as well as dynamic target reaching/tracking.
- The “Target assignment” block lets us obtain, at each sample time, the current waypoint (target) to reach. This block is detailed in what follows.

*Sequential target assignment:* The strategy to assign, at each sample time, the waypoint to reach by the vehicle is shown in Algorithm 1. The stable and reliable control law has to reach each assigned waypoint while ensuring that the vehicle’s trajectory is always within the road boundaries.

The error conditions,  $E_{dis}$  and  $E_{angle}$ , are used to switch to the next waypoint when the vehicle’s position is inside a circle given by the center  $(x_{T_j}, y_{T_j})$  and a radius  $E_{dis}$ . Hence, the current waypoint index is updated with the next waypoint and the vehicle has to thereafter adapt its movement according to this new target. If the vehicle does not satisfy the distance and orientation error conditions (the errors  $d$  and  $e_\theta >$  than  $E_{dis}$  and  $E_{angle}$  respectively) when crossing the  $Y_{T_j}$  axis (cf. Figure 9), then the vehicle must nevertheless switch to the next waypoint. Obviously, this situation should not occur if the environment is accurately modeled/identified and the control law well settled. Despite all these aspects, if this situation happens, then the value of the maximal distance and angular errors can be used to decide if the vehicle could or not continue its navigation. This fault detection/diagnosis

---

### Algorithm 1: Sequential target assignment

---

- Require:** Vehicle pose, current target  $T_j$  and a set of  $N$  sorted waypoints
- Ensure:** Reaching  $T_j$  while guaranteeing to the vehicle to be in the best configuration to reach after the next waypoint  $T_{j+1}$ .
- 1: **if** ( $(d \leq E_{dis}$  **and**  $e_\theta \leq E_{angle}$ ) **or**  $(x^{T_j} \geq 0)$ )  
 $\{x^{T_j}$  is the coordinate of the vehicle in the local Target frame  $X_{T_j}Y_{T_j}$  (cf. Figure 9)**}** **then**
  - 2: Switch from the current target  $T_j$  to the next sequential waypoint  $T_{j+1}$
  - 3: **else**
  - 4: Keep going to waypoint  $T_j$
  - 5: **end if**
- 

is not addressed in this paper.

It is also interesting to mention that the definition of  $Y_{T_j}$  axis, as in section II-B, guides the task achievement. In fact, it is used in the cited section to perform elementary obstacle avoidance. This axis is used here as a mean to decide when to switch to the next waypoint.

*Experimental validation:* The presented scenario was built to show different situations in urban environment, such as multi-vehicle navigating in platoon (based on Leader-Follower formulation (cf. section IV)), static and dynamic target-reaching and obstacle-avoidance situation. The experiment was done using VIPALAB vehicles (cf. Figure 1(d)) in PAVIN platform (*Plate-forme d’Auvergne pour Véhicules INtelligents*). A metric map of PAVIN [56] is used to plan the optimal configuration of geo-referenced waypoints with optimal configurations [55]. In this experiment, each vehicle uses a combination of RTK-GPS and gyrometer to estimate its current position and orientation at a sample time of  $T_s = 0.1$  s. The vehicles have a range sensor (LIDAR) with a maximum detected range equal to 10 m. Moreover, the vehicles communicate by Wi-Fi, enabling the transmission of the Leader’s pose data.

The presented experiment highlights the performance of the proposed control law and target assignment strategy using waypoint selection. The Leader vehicle has to successively reach static waypoints. Moreover, the proposed control law (cf. section II-C.4) was implemented in another vehicle (Follower) which takes the first vehicle (Leader) as a dynamic target to track at a curvilinear distance equal to 5 m (behind the Leader). The tracking of the dynamic target allows us to apply the proposed control law to multi-vehicle systems where the dynamic set-point is given by the leader and the desired geometric formation shape [57]. The configuration of the dynamic target is sent by the Leader to the Follower via Wi-Fi. This experiment can be found online.<sup>4</sup> Furthermore, to exhibit the flexibility of the proposed navigation strategy, an obstacle is placed between the waypoints. Therefore, the vehicle can perform different maneuvers between waypoints, in this case obstacle avoidance, without the use of any trajectory replanning method. The obstacle avoidance is activated as soon as the vehicle detects at least one obstacle

<sup>4</sup><https://goo.gl/gVgsIa>

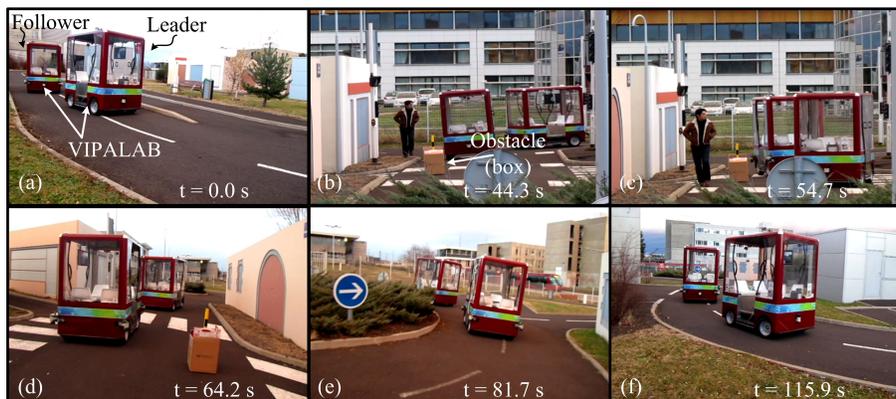


Fig. 11. Some images from the performed navigation in urban environment.

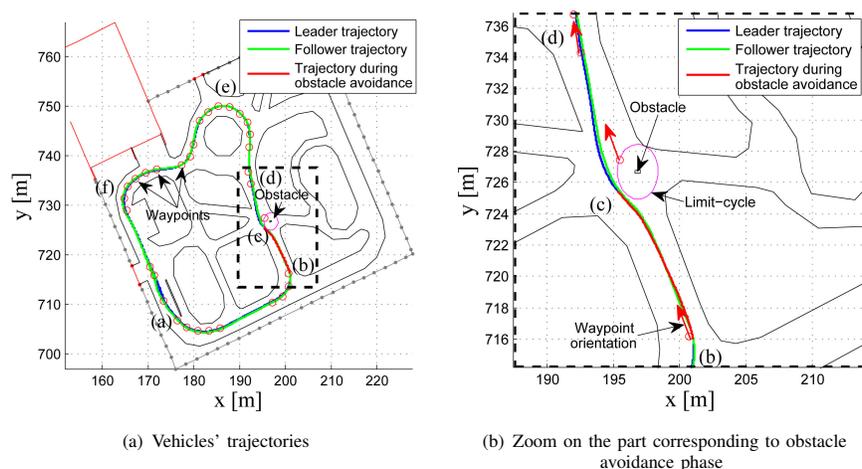


Fig. 12. Vehicle trajectories obtained using GPS and a set of waypoints positioned in the environment.

which can hinder the future vehicle movements toward the current assigned waypoint. It can be seen in Figure 12(a) that the Leader accurately reaches the successive assigned static waypoints and the Follower also accurately tracks the dynamic target (Leader). Moreover, the Follower trajectory using the proposed control law is close to the Leader trajectory (cf. Figure 12(a)). Figure 12(b) focuses on the vehicles' trajectories when the obstacle avoidance is activated. The Leader detects the obstacle between the waypoints and it applies local obstacle avoidance based on limit-cycles (cf. section II-B). The Follower also avoids the obstacle since it accurately tracks the Leader trajectory. It can be noted that the proposed navigation strategy allows flexible and smooth movements between the waypoints and performance of different behaviors, such as obstacle avoidance, emergency stop or waypoint reassignment.

#### IV. DYNAMIC MULTI-VEHICLE NAVIGATION IN FORMATION

The navigation in formation is addressed in this section while using the Leader-follower approach. This approach has been adopted and applied on VIPALAB vehicles for dynamic reconfiguration of a fleet of vehicles [57]. Each follower tracks the instantaneous state (pose and velocity) of

its assigned virtual targets, given w.r.t. the Leader dynamic.

##### A. Problem statement and Task modeling

The Leader-follower approach enables us to maintain a rigid geometric shape (e.g., a triangle in Figure 13). The formation is defined w.r.t. the Cartesian frame (local frame of the leader), as follows:

- A Leader (UGV<sub>L</sub> in Fig. 13); its pose  $(x_L, y_L, \theta_L)$  and its linear velocity  $v_L$  determine the dynamic of the formation.
- The formation structure is defined with as many nodes as necessary to obtain the desired formation shape. Each node  $i$  is a virtual dynamic target ( $T_{d_i}$ ). The formation is defined as  $\mathbf{F} = \{\mathbf{f}_i, i = 1 \dots N\}$ , where  $\mathbf{f}_i$  are the coordinates  $(h_i, l_i)^T$  of the dynamic target  $T_{d_i}$  w.r.t. the leader local reference frame.

The position and orientation of each node (virtual target) are computed from the leader configuration. The leader position determines the node positions according to the formation shape. The instantaneous center of curvature  $I_{ccl}$  of the formation is determined by the leader according to its movements (cf. Figure 13).  $I_{ccl}$  allows us to compute the desired orientation of the nodes according to the formation

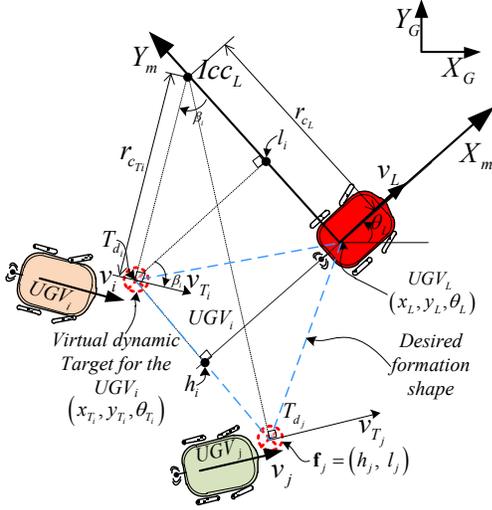


Fig. 13. Maintaining a triangular formation by defining set-points according to a mobile reference frame linked to the Leader (Leader-follower approach).

shape. The leader turns around  $I_{cCL}$  (positioned perpendicularly to its rear wheels), then the other target set-points  $T_{d_i}$  must also turn around  $I_{cCL}$  to maintain a rigid formation. Thus, the target velocity  $v_{T_i}$  must be tangent to the circle which has  $I_{cCL}$  as center and the distance between  $T_{d_i}$  and  $I_{cCL}$  as radius  $r_{cT_i}$ .

The idea behind this strategy is to eliminate the dependency of each vehicle to a global reference frame. A straightforward transformation can be applied to obtain the set-point w.r.t. a local reference frame attached to the leader. The polar coordinates  $(r_i, \Phi_i)$  can also be used by applying a straightforward transformation. An important advantage of the used Leader-follower approach is that it does not depend here on any reference trajectory and the formation is fully defined by the instantaneous dynamic of the leader. Furthermore, the presented approach is more reactive in the sense that it takes at each sample time only the current configuration and velocity of the Leader, instead of using the trajectory of the Leader as a reference for the formation [58], [59].

An important consideration to take into account to achieve the presented formation strategy, is that the followers have to know, as accurately as possible, the leader state (pose and velocity). It is assumed in what follows that the Leader sends its state by stable Wi-Fi communication without latency. However, cameras and/or LIDAR sensors embedded in each follower, can be used to estimate the leader state [60] [61] [8]. In the sequel,  $\mathbf{f}_i$  is given in a global Cartesian frame to homogenize the notation of the equations [57]. The pose of the virtual target  $T_{d_i}$  w.r.t. the leader pose in the global reference frame can be written as (cf. Figure 13):

$$\begin{cases} x_{T_i} &= x_L + h_i \cos(\theta_L) - l_i \sin(\theta_L) \\ y_{T_i} &= y_L + h_i \sin(\theta_L) + l_i \cos(\theta_L) \\ \theta_{T_i} &= \theta_L + \beta_i \end{cases} \quad (15)$$

where  $(x_L, y_L, \theta_L)$  is the current pose of the leader and  $\beta_i$  is the  $T_{d_i}$  orientation w.r.t. the leader pose. It is given by:

$$\beta_i = \arctan(h_i / (r_{cL} - l_i)) \quad (16)$$

where  $r_{cL}$  is the radius of curvature of the leader. Differentiating equation (15), the velocities of each  $T_{d_i}$  are given thus by:

$$v_{T_i} = \sqrt{(v_L - l_i \omega_L)^2 + (h_i \omega_L)^2} \quad (17)$$

$$\omega_{T_i} = \omega_L + \dot{\beta}_i \quad (18)$$

where  $v_L$  and  $\omega_L$  are respectively the linear and angular velocities of the leader,  $\dot{\beta}_i$  is computed as:

$$\dot{\beta}_i = -h_i \dot{r}_{cL} / ((r_{cL} - l_i)^2 + (h_i)^2). \quad (19)$$

One can note from equation (19) that when  $\dot{\beta}_i$  is equal to zero, the formation has a constant radius of curvature  $r_{cL}$  and the angular velocities of the virtual targets are equal to the angular velocity of the leader ( $\omega_{T_i} = \omega_L$ ) (18).

*Dynamic and smooth formation reconfiguration:* It is presented summarily in what follows a new Strategy for Formation Reconfiguration (SFR) [57] based on suitable smooth switches between different virtual target configurations. It is considered in the following a deterministic target assignment, and a label  $H_i$  of the virtual target  $T_{d_i}$  is assigned to UGV $_i$  at the beginning of the experiments. This label is kept by each UGV in the reconfiguration phase. It is important to notice that the new virtual targets (defined on the new formation shape) must be ahead of the UGVs to guarantee the stability of the overall system (the vehicle must not go back to reach the new virtual target). If this condition is not satisfied, then the former formation will be adapted by increasing smoothly and contentiously the longitudinal coordinates  $h_i$  until all UGVs are positioned in the right configuration. The error between the coordinates of the former and the new formation  $\mathbf{e}_{f_i}(e_{h_i}, e_{l_i})$  is defined as:

$$\mathbf{e}_{f_i} = \mathbf{f}_i^n - \mathbf{f}_i^f \quad (20)$$

where  $\mathbf{f}_i^f(h_i^f, l_i^f)$  and  $\mathbf{f}_i^n(h_i^n, l_i^n)$  are respectively the coordinates of the former formation and the new desired formation (cf. Figures 13).

The reconfiguration process between the different formation shapes is given by:

$$\mathbf{f}_i = \begin{cases} h_i = h_i^n - e_{h_i} e^{-k_r(t-t_r)}, & l_i = l_i^n; & \text{if } e_{h_i} < 0 \\ h_i = h_i^f, & l_i = l_i^f; & \text{if } e_{h_i} \geq 0 \end{cases} \quad (21)$$

where  $\mathbf{f}_i(h_i, l_i)$  are the coordinates of the current virtual target  $T_{d_i}$  to be tracked by the follower UGV $_i$ .  $e_{h_i}$  is the longitudinal coordinate of  $\mathbf{e}_{f_i}$  that enables to detect if the virtual target is ahead of its corresponding follower ( $e_{h_i} \geq 0$ ). The adaptation function when  $e_{h_i} < 0$  (virtual target behind to follower $_i$ ) is set as proportional to the error between formation shapes, where  $k_r$  is a real positive constant designed according to the dynamic of the leader and  $t_r > 0$  is the initial time for the reconfiguration process.

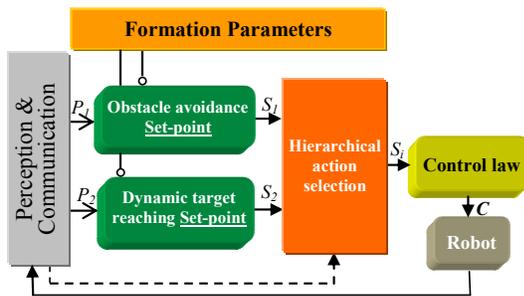


Fig. 14. Multi-controller architecture embedded in each robot in the formation.

### B. Proposed MCA and Experimental validation

This section presents the proposed multi-controller architecture (cf. Figure 14) to obtain safe and smooth robot navigation in formation. A basic structure of type 2 (cf. Figure 6(b)) is used with notably the addition of a Formation Parameters block which determines the desired multi-robot configurations. An overview of the different blocks composing this architecture is briefly presented below while emphasizing the new blocks/features.

- “Perceptions & communication” block: as emphasized in section II-C.1, this block is in charge of all the local and/or global robot perceptions. Furthermore, knowing that several robots have to coordinate their movements, it is important to have reliable and low-latency communication between the robots.
- “Hierarchical action selection” block: it aims to manage the switches between the two elementary controllers, *Obstacle avoidance* and *Dynamic target reaching* blocks, according to the formation parameters and environment perception. It activates the *Obstacle avoidance* controller as soon as it detects at least one obstacle which can hinder the robot’s future movement toward its assigned dynamic virtual target.
- In terms of set-point blocks, they are harmonized as motivated in section II-C.3. Indeed, they are always defined according to an appropriate target pose  $(x_T, y_T, \theta_T)$  and a linear velocity  $v_T$ .
  - “Obstacle avoidance set-point” block: This well detailed block, in section II-B, allows to each elementary robot to avoid reactively, and in a safe and reliable way, any obstructing obstacle.
  - “Dynamic target reaching set-point” block: These set-points are defined according to the assigned formation shape (e.g., triangle, line, etc.). All the robots (except the Leader) have to track their assigned dynamic target (given according to the desired formation).
- “Control law” block: A stable and generic common control law for target tracking, as given in section II-C.4 is used.

*Experimental validation:* An experimentation was made using 3 VIPALABs. The objective is to validate the pro-

posed strategies based on the Leader-follower approach and reconfiguration mechanism. Figure 15(a) shows the sequence of the MVS evolution, from the beginning with an initial triangular formation to a linear one, when the Leader detects an obstacle, and once the last follower detects the end of the obstacle, the formation returns to the triangular formation in a smooth way. Figure 15(b) shows the trajectories of the 3 VIPALABs, and it attests on the safety and the smoothness of the performed navigation in formation and its reconfiguration. This experiment can be found online.<sup>5</sup>

More details about the proposed strategy which deals with the navigation in formation of a group of autonomous vehicles are given in [57] and [62, Chapter 5]. These references highlight also the way to guarantee that the assigned target’s set-points are always attainable by the Followers, which allows thus to guarantee the reliability of the overall multi-vehicle navigation in formation.

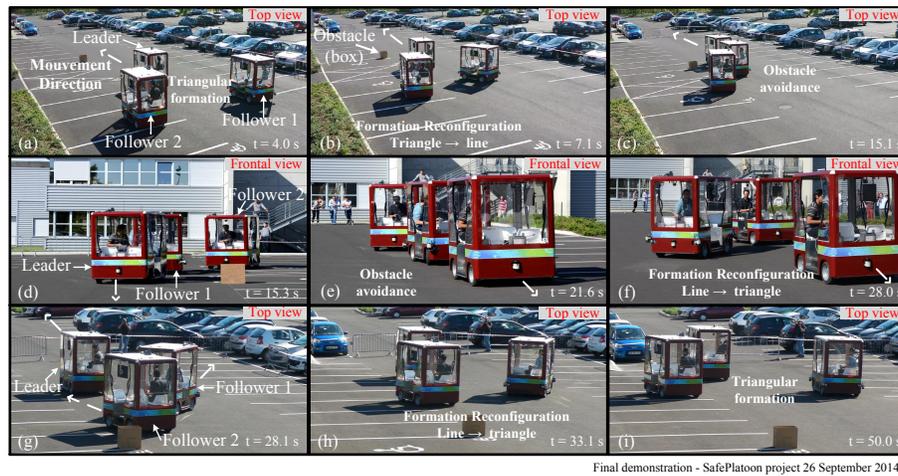
### V. CONCLUSION AND FURTHER WORK

This paper emphasizes the use of homogeneous Multi-Controller Architectures (MCAs) in order to increase gradually the autonomy of single vehicle as well as multi-vehicle systems to achieve complex navigation in cluttered and/or dynamic environments. Indeed, the obtainment of fully autonomous vehicle could be reached only if the overall used control architecture (perception-decision-action) can guarantee the safety and reliability of the navigation while allowing to maintain a high level of flexibility and adaptability to achieve different assigned tasks. This paper made a focus on the decision/action aspects, while emphasizing the importance to have generic and bottom-up developments of the MCA. Three closely related elements have been highlighted concerning MCA: (i) Importance of flexible and reliable obstacle avoidance controller (ii) Homogenization and standardization of task modeling: in terms of set-points definition and their corresponding stable control law; appropriate reference frames for task modeling/achievement. These specific reference frames guide for instance the vehicle’s behaviors and allows us to evaluate the success of the current achieved sub-task (e.g., obstacle avoidance, targets reaching/tracking, etc.). Therefore, each elementary reference frame orientates locally the vehicle navigation. (iii) Navigation sub-tasks definition while using only appropriate attraction/tracking toward static/dynamic targets. This has been highlighted with two examples: flexible and reliable navigation based on sequential target reaching, and dynamic multi-vehicle navigation and formation reconfiguration. As main future work to enhance the features of MCAs, several works have been initiated to prove the stability and reliability of the overall MCA even in the presence of uncertainties (due for instance to the perception or to the vehicle modeling).

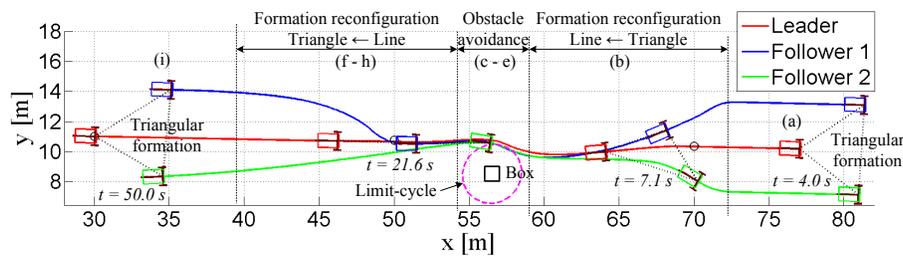
### REFERENCES

- [1] W. A. Memon, M. Bilal, Fully autonomous flammable gases (Methane-Gas) sensing and surveillance robot, in: International Conference on Artificial Intelligence, Energy and Manufacturing Engineering, Dubai, 2015.

<sup>5</sup><https://goo.gl/nUPnt9>



(a)



(b)

Fig. 15. Final demonstration given in the context of the SafePlatoon project (reference ANR-10-VPTT-0011).

- [2] J.-A. Fernández-Madrigal, J. L. B. Claraco, Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods, IGI Global, Hershey, PA, USA, 2013.
- [3] A. Simpkins, C. Simpkins, Rescue robotics [on the shelf], Robotics Automation Magazine, IEEE 21 (4) (2014) 108–109.
- [4] G. Lozenguez, L. Adouane, A. Beynier, A. I. Mouaddib, P. Martinet, Punctual versus continuous auction coordination for multi-robot and multi-task topological navigation, Autonomous Robots - Springer 40 (4) (2016) 599–613.
- [5] P. J. Springer, Military Robots and Drones: A Reference Handbook, ABC-CLIO, 2013.
- [6] X.-Y. Lu, S. Shladover, Automated truck platoon control and field test, in: G. Meyer, S. Beiker (Eds.), Road Vehicle Automation, Lecture Notes in Mobility, Springer International Publishing, 2014, pp. 247–261.
- [7] J. Güttler, C. Georgoulas, T. Linner, T. Bock, Towards a future robotic home environment: A survey, Gerontology (International Journal of Experimental, Clinical, Behavioural and Technological Gerontology).
- [8] J.-M. Vilca, L. Adouane, Y. Mezouar, A novel safe and flexible control strategy based on target reaching for the navigation of urban vehicles, Robotics and Autonomous Systems (RAS) 70 (2015) 215–226.
- [9] DARPA, Grand Challenge, [https://fr.wikipedia.org/wiki/DARPA\\_Grand\\_Challenge](https://fr.wikipedia.org/wiki/DARPA_Grand_Challenge), consulted January 2015 (2015).
- [10] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al., Stanley: The robot that won the DARPA Grand Challenge, in: The 2005 DARPA Grand Challenge, Springer, 2007, pp. 1–43.
- [11] M. Buehler, K. Iagnemma, S. Singh, The DARPA urban challenge: Autonomous vehicles in city traffic, Vol. 56, Springer Science & Business Media, 2009.
- [12] L. Adouane, Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures, Taylor & Francis CRC Press, ISBN: 9781498715584, 228 pages, 2016.
- [13] L. D. Burns, Sustainable mobility: A vision of our transport future, Nature 497 (7448) (2013) 181–182.
- [14] S. Thrun, IEEE Spectrum, How Google's Self-Driving Car Works, <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works> (2011).
- [15] ADAS, Advanced Driver Assistance Systems, [http://en.wikipedia.org/wiki/Advanced\\_driver\\_assistance\\_systems](http://en.wikipedia.org/wiki/Advanced_driver_assistance_systems), consulted January 2015 (2015).
- [16] M. A. Arbib, Perceptual structures and distributed motor control, In Handbook of Physiology, Section 2: The Nervous System, II, Motor Control, Part 1 (1981) 1449–1480.
- [17] T. Anderson, M. Donath, Animal behavior as a paradigm for developing robot autonomy, Robotics and Autonomous Systems 6 (1990) 145–168.
- [18] R. A. Brooks, A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation RA-2 (1986) 14–23.
- [19] R. C. Arkin, Behavior-Based Robotics, The MIT Press, 1998.
- [20] L. Adouane, N. Le-Fort-Piat, Hybrid behavioral control architecture for the cooperation of minimalist mobile robots, in: ICRA'04, International Conference On Robotics And Automation, New Orleans-USA, 2004, pp. 3735–3740.
- [21] L. Adouane, N. Le-Fort-Piat, Behavioral and distributed control architecture of control for minimalist mobile robots, Journal European des Systemes Automatises 40 (2) (2006) pp.177–196.
- [22] L. Adouane, Reactive versus Cognitive Vehicle Navigation based on Optimal Local and Global PELC, Robotics and Autonomous Systems (RAS) 88 (2017) 51–70. doi:<http://dx.doi.org/10.1016/j.robot.2016.11.006>.
- [23] A. Benzerrouk, L. Adouane, P. Martinet, N. Andreff, Multi lyapunov function theorem applied to a mobile robot tracking a trajectory in presence of obstacles, in: European Conference on Mobile Robots (ECMR 2009), Milini/Dubrovnik Croatia, 2009.
- [24] J.-M. Vilca, L. Adouane, Y. Mezouar, Robust on-line obstacle detection using data range for reactive navigation, in: 10th International IFAC Symposium on Robot Control (SYROCO'12), Dubrovnik - Croatia, 2012.
- [25] J.-M. Vilca, L. Adouane, Y. Mezouar, On-line obstacle detection using data range for reactive obstacle avoidance, in: 12th International

- Conference on Intelligent Autonomous System (IAS-12), Jeju Island - Korea, 2012, published after in *Advances in Intelligent Systems and Computing*, 193, 2013, 3-13.
- [26] J. Vilca, L. Adouane, Y. Mezouar, Reactive navigation of mobile robot using elliptic trajectories and effective on-line obstacle detection, *GyroscoPy and Navigation*, Ed. Springer, Russia ISSN 2075 1087 4 (1) (2013) 14–25.
- [27] J. Minguez, F. Lamiroux, J.-P. Laumond, *Handbook of Robotics*, 2008, Ch. Motion Planning and Obstacle Avoidance, pp. 827–852.
- [28] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *The International Journal of Robotics Research* 5 (1986) pp.90–99.
- [29] F. Aurenhammer, Voronoi diagrams – a survey of a fundamental geometric data structure, *ACM COMPUTING SURVEYS* 23 (3) (1991) 345–405.
- [30] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, MA, 1991.
- [31] E. Rimon, D. E. Koditschek, Exact robot navigation using artificial potential fields, *IEEE Transactions on Robotics and Automation* 8(5) (1992) 501–518.
- [32] S. M. Lavalle, Rapidly-exploring random trees: A new tool for path planning, Tech. rep., Computer Science Dept., Iowa State University (1998).
- [33] R. Zapata, A. Cacitti, P. Lepinay, DVZ-based collision avoidance control of non-holonomic mobile manipulators, *JESA, European Journal of Automated Systems* 38(5) (2004) 559–588.
- [34] D.-H. Kim, J.-H. Kim, A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer, *Robotics and Autonomous Systems* 42(1) (2003) 17–30.
- [35] M. S. Jie, J. H. Baek, Y. S. Hong, K. W. Lee, Real time obstacle avoidance for mobile robot using limit-cycle and vector field method, *Knowledge-Based Intelligent Information and Engineering Systems*.
- [36] L. Adouane, Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation, in: 9th Conference on Autonomous Robot Systems and Competitions, Selected for publication in Portuguese Journal *Robotica* N79, automacao, controlo and instrumentacao, Portugal, 2009.
- [37] L. Adouane, A. Benzerrouk, P. Martinet, Mobile robot navigation in cluttered environment using reactive elliptic trajectories, in: 18th IFAC World Congress, Milano-Italy, 2011.
- [38] R. A. Soltan, H. Ashrafioun, K. R. Muske, Ode-based obstacle avoidance and trajectory planning for unmanned surface vessels, *Robotica* 29 (2011) 691–703. doi:10.1017/S0263574710000585.
- [39] J. Denavit, R. S. Hartenberg, A kinematic notation for lower-pair mechanisms based on matrices, *Trans. ASME E, Journal of Applied Mechanics* 22 (1955) 215–221.
- [40] W. Khalil, E. Dombre, *Modeling, identification and control of robots*, Hermes Penton, 2004.
- [41] B. Dafflon, J. Vilca, F. Gechter, L. Adouane, Adaptive autonomous navigation using reactive multi-agent system for control laws merging, *Procedia Computer Science* Selected paper from the International Conference on Computational Science ICCS'15 (Reykjavík, Iceland).
- [42] F. Boufera, F. Debbat, L. Adouane, K. M. Khelifi Faycal, Mobile robot navigation using fuzzy limit-cycles in cluttered environment, *International Journal of Intelligent Systems and Applications* (7) (2014) 12 – 21.
- [43] M. S. Branicky, Multiple Lyapunov functions and other analysis tools for switched and hybrid systems, *IEEE Transaction on Automatic Control* 43(4) (1998) 475–482.
- [44] M. Zefran, J. W. Burdick, Design of switching controllers for systems with changing dynamics, in: *IEEE Conference on Decision and Control CDC'98*, Vol. 2, FL, USA, 1998, pp. 2113–2118. doi:10.1109/CDC.1998.758647.
- [45] D. Liberzon, *Switching in Systems and Control*, Birkhauser, 2003.
- [46] L. Adouane, Hybrid and safe control architecture for mobile robot navigation, in: 9th Conference on Autonomous Robot Systems and Competitions, Portugal, 2009.
- [47] A. Benzerrouk, L. Adouane, Z. Al-Barakeh, P. Martinet, Stabilité globale pour la navigation réactive d'un robot mobile en présence d'obstacles, in: *CIFA 2010, Sixieme Conference Internationale Francophone d'Automatique*, Nancy-France, 2010.
- [48] L. Adouane, Towards smooth and stable reactive mobile robot navigation using on-line control set-points, in: *IEEE/RSJ, IROS'13, 5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, Tokyo-Japan, 2013.
- [49] C. Samson, Control of chained systems: Application to path following and time-varying point-stabilization of mobile robots, *IEEE Transactions on Automatic Control* 40 (1) (1995) 64–77.
- [50] A. D. Luca, G. Oriolo, C. Samson, Feedback control of a nonholonomic car-like robot, in: J.-P. Laumond (Ed.), *Robot Motion Planning and Control*, Springer-Verlag, 1998, pp. 171–253.
- [51] H. K. Khalil, *Frequency domain analysis of feedback systems, Non-linear Systems: Chapter 7*, 3 edition, 2002.
- [52] G. M. Siouris, *Missile Guidance and Control Systems*, Springer-Verlag, 2004.
- [53] J.-M. Vilca, L. Adouane, Y. Mezouar, P. Lebraly, An overall control strategy based on target reaching for the navigation of a urban electric vehicle, in: *IEEE/RSJ, IROS'13, International Conference on Intelligent Robots and Systems*, Tokyo-Japan, 2013.
- [54] T. Gu, J. M. Dolan, On-road motion planning for autonomous vehicles, in: C.-Y. Su, S. Rakheja, H. Liu (Eds.), *Intelligent Robotics and Applications*, Vol. 7508, Springer Berlin Heidelberg, 2012.
- [55] J.-M. Vilca, L. Adouane, Y. Mezouar, Optimal multi-criteria waypoint selection for autonomous vehicle navigation in structured environment, *Journal of Intelligent & Robotic Systems (JIRS)* 4 (1) (2016) 301–324.
- [56] S. IP.Data.Sets, <http://ipds.univ-bpclermont.fr> (March 2015).
- [57] J.-M. Vilca, L. Adouane, Y. Mezouar, Adaptive leader-follower formation in cluttered environment using dynamic target reconfiguration, in: *Springer Tracts in Advanced Robotics*, from International Symposium on Distributed Autonomous Robotic Systems, DARS 2014, Daejeon - Korea, 2014.
- [58] X. Chen, Y. Li, Smooth formation navigation of multiple mobile robots for avoiding moving obstacles, *International Journal of Control, Automation* 4 (4) (2006) 466–479.
- [59] I. Shames, M. Deghat, B. Anderson, Safe formation control with obstacle avoidance, in: *IFAC World Congress*, Milan - Italy, 2011.
- [60] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, C. Taylor, A vision-based formation control framework, *IEEE Transaction on Robotics and Automation* 18 (5) (2002) 813–825.
- [61] M. El-Zaher, J.-M. Contet, F. Gechter, A. Koukam, Echelon platoon organisation: A distributed approach based on 2-spring virtual links, in: *Proceeding of the 15th International Conference on Artificial Intelligence: Methodology, Systems, Applications*, Germany, 2012.
- [62] J. M. Vilca Ventura, Safe and flexible hybrid control architecture for the navigation in formation of a group of vehicles, Ph.D. thesis, Balise Pascal University (October 2015).