Toward Smooth and Stable Reactive Mobile Robot Navigation using On-line Control Set-points

Lounis Adouane
Institut Pascal, UBP/IFMA UMR CNRS 6602, France
e-mail: Lounis.Adouane@univ-bpclermont.fr

***Abstract*— This paper deals with the challenging issue of on-line mobile robot navigation in cluttered environment. Indeed, it is considered in this work, a mobile robot discovering the environment during its navigation, it should thus, to react to unexpected events (e.g., obstacles to avoid) while guaranteeing to reach its objective. Nevertheless, in addition to avoid safely and on-line these obstacles, it is proposed to enhance the smoothness of the obtained robot trajectories. Otherwise, to quantify this smoothness, suitable indicators were used. Specifically, this paper proposes to appropriately link on-line set-points defined using elliptic limit-cycle trajectories with a multi-controller architecture which guarantees the stability (according to Lyapunov synthesis) and the smoothness of the switch between controllers. Moreover, a comparison between fully reactive mode (the aim of this paper) and planned mode is given through the proposed control architecture which could exhibits the two aspects. Many simulations in cluttered environments permit to confirm the reliability and the robustness of the overall proposed reactive control.**

I. INTRODUCTION

Among the main challenges to obtain a fully autonomous mobile robot navigation, is the ability for the robot to react on-line to unpredictable events encountered in its environment. The asked question is thus *how to navigate toward a goal in a cluttered environment when obstacles are discovered in real time?* [13]. Nevertheless, it is not sufficient to avoid these obstacles. In fact, robot should also guarantee a smooth navigation [7] for the comfort, for example of the passengers. In [8], the author characterizes this smooth navigation while using a cost functional J that reflects the trade-off between the travel time and the integral of acceleration (which characterizes the amount of jerking of angular and linear robot velocity). All these criterion are concatenated in one and modulated by weights which give thus the priority for each one.

To obtain on-line, accurate, flexible and reliable navigation, one part of the literature in this domain considers that the robot is fully actuated with no control bound and focuses the attention on path planning and re-planning. Voronoï diagrams and visibility graphs [12], navigation functions [18] or planning based grid checking and trajectory generation [17] are among these road-map-based methods. However, the other part of the literature considers that to control a robot with the above criterion, it is essential to accurately take into account: robot structural constraints (e.g., nonholonomy); avoid command discontinuities and set-point jerk, etc. Our proposed control architecture is linked to this last approach, thus where the control stability is rigorously demonstrated.

It is commonly used in the literature a pre-planned reference trajectory, which means that it was appropriately planned or selected before robot movement [14]. However, in real motion conditions where the environment can to be very cluttered and with high dynamic, these methods could not be very efficient due, among others, to time consuming to obtain the new re-planed trajectory [13]. Otherwise, a large class of model-based techniques use optimization to choose between a set of admissible trajectories [5], [16]. In the proposed paper, it is defined a fully reactive mobile robot navigation. Indeed, at each sample time, the robot should follows defined set-points, according to local robot perceptions and objectives.

To guarantee multi-objective criteria, control architectures can be elaborated in a modular and bottom-up way as introduced in [6] and so-called behavioral architectures [3]. These techniques are based on the concept that a robot can achieve a complex global task while using only the coordination of several elementary behaviors. In fact, to tackle this complexity, behavioral control architecture decompose the global controller into a set of elementary behavior/controller (e.g., attraction to the objective, obstacle avoidance, trajectory following, etc.) to master better the overall robot behavior. Moreover, it is considered in a lot of studies the investigation of the potentialities of the hybrid systems controllers [21] to provide a formal framework to demonstrate the robustness and the stability of such architecture. In their most simple description, hybrid systems are dynamical systems modeled as a finite state automaton. These states correspond to a continuous dynamic evolution, and the transitions can be enabled by particular conditions reached by the continuous part. This formalism permits a rigorous automatic control analysis of the performances of the control architecture [4].

Among controllers which can make up a behavioral control architecture, obstacle avoidance controllers play a large role to achieve autonomously and safely the navigation of mobile robots in a cluttered and unstructured environments. An interesting overview of obstacle avoidance methods is accurately given in [13]. The proposed control architecture integrates obstacle avoidance method which uses limit-cycle vector field [10], [11], [1]. Moreover, it introduces an adap-

*This work was supported by the French National Research Agency (ANR) through the Safeplatoon project.

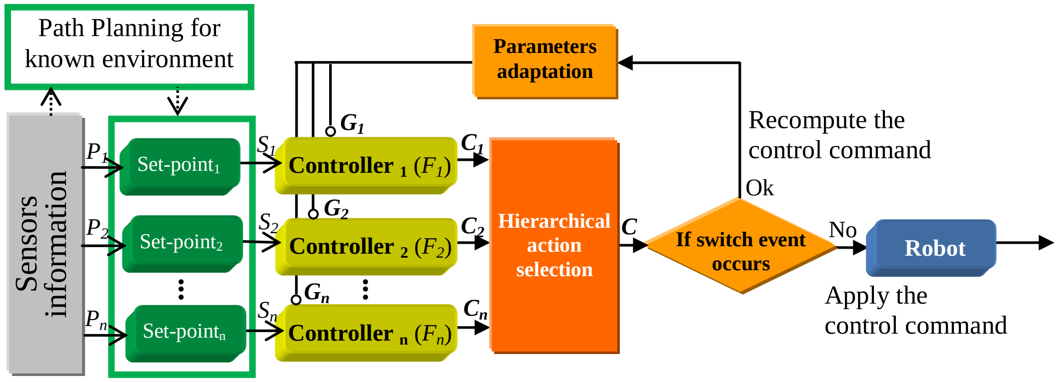


Fig. 1. The proposed hybrid control architecture for mobile robot navigation

tive and flexible mechanism of control which guarantees the stability and the smoothness of the switch between controllers.

The rest of the paper is organized as follows. Section II gives the specificities of the proposed control architecture. In section III, the control architecture is applied to the task of navigation in the presence of obstacles. It presents the model of the considered robot and the different modules constituting the proposed control architecture. Section IV deals with safety mode mechanism. Section V is devoted to the description and analysis of the simulation results. This paper ends with some conclusions and further work.

II. Control Architecture

The proposed control architecture (cf. Figure 1) is dedicated for the general framework of the navigation of mobile robots in cluttered environments. It permits to manage the interactions between different elementary controllers while guaranteeing the stability and the smoothness of the overall control. Moreover, a specific "safety mode" is proposed in section IV to avoid undesirable robot behaviors (oscillations, abrupt movement, etc.). The robot can therefore have very smooth trajectories while guaranteeing safe obstacle avoidance. The specific blocks composing this generic overall control architecture are detailed below. In section III a concrete control architecture applied for a real task is given.

A. Path planning for known environment

This path planner (and re-planner) block is activated only if the entire mission is well known or when the navigation is achieved in relatively low dynamic environment. The aim of the proposed paper is to make the focus only around reactive mobile robot navigation (where the environment is discovered on-line). Therefore, the used path planner and its interaction with low level control are not addressed here. This part of the control was studied in [15] and will be the subject of a future developments.

B. Set-points blocks

These blocks, which have as input the perceptions P_i, are responsible to give for each dedicated controller block (e.g., obstacle avoidance, target to reach, etc.) the set-points useful for its working (e.g., for attraction to target controller, the relative position of the target to reach).

C. Controllers blocks

Every controller F_i is characterized by a stable nominal law which is represented by the function:

$$F_i(S_i, t) = \eta_i(S_i, t) \tag{1}$$

with S_i is the set-point sent to the controller "i". Otherwise, in order to avoid the important controls jumps at the time for example of the switch between controllers (e.g., from the controller "j" toward the controller "i" at the instant t_0), an adaptation of the nominal law is proposed, F_i becomes thus:

$$F_i(S_i, t) = \eta_i(S_i, t) + G_i(S_i, t) \tag{2}$$

with $G_i(S_i, t)$ (cf. Equation 3) is a strictly monotonous function that tends to zero after a certain amount of time "$T = H_i(P_i, S_i)$". The value of this time depends on the criticality of the controller$_i$ to join as quickly as possible the nominal law $\eta_i(S_i, t)$. It constitutes thus the controller safety mode (cf. Section IV for a specific example for obstacle avoidance controller).

$$G_i(S_i, t_0) = F_j(S_j, t_0 - \Delta t) - \eta_i(S_i, t_0) \tag{3}$$

where Δt represents the sampling time between two control set-points and t_0 is the time of abrupt change in S_i.

The definition of $G_i(S_i, t)$ allows to guarantee that the control law (cf. Equation 2) tends toward the nominal control law after a certain time T, thus:

$$G_i(S_i, T) = \varepsilon \tag{4}$$

Where ε very small constant value $\approx$ 0. The adaptive function $G_i(S_i, t)$ is updated by the "Parameters adaptation" block every time a hard control switch concerning the "i" controller occurs (cf. Section II-D) (cf. Figure 1). The main challenge introduced by this kind of control is to guarantee the stability of the updated control law (cf. Equation 2) even during the period where $|G_i(S_i, t)| \gg \varepsilon$.

D. Parameters adaptation block

This block has as input the "conditional block" (cf. Figure 1) that verifies if specific control switch event occurs. So, if it is the case then it must update "Adaptive Function" corresponding to the future active controller (cf. Equation 3). The different configurations which need the activation of parameters adaptation block are given below:

1) When a controller which should be active at the current "t" instant is different than the one which was active at the "t-Δ t" instant,
2) When an abrupt transition in the set-points S_i of the controller$_i$ is encountered.

III. Navigation in presence of obstacles task

The navigation in a cluttered environment aims here to lead the robot to reach a target-position while avoiding obstacles (cf. Figure 2). The robot movement needs to be fast and smooth while avoiding statical and dynamical obstacles which could have different shapes.

One supposes in the setup that robot and obstacles are surrounded by respectively cylindrical and elliptical boxes (cf. Figure 2). The cylindrical box (the robot) is characterized by R_R radius and elliptical boxes (obstacles) are given by:

$$a(x-h)^2 + b(y-k)^2 + c(x-h)(y-k) = 1 \qquad (5)$$

With:

- $h, k \in \mathbb{R}$, give the coordinate of the center of the ellipse,
- $a \in \mathbb{R}^+$, permits to give the half length $A = 1/\sqrt{a}$ of the longer side (major axis) of the ellipse,
- $b \in \mathbb{R}^+$, permits to give the half length $B = 1/\sqrt{b}$ of the shorter side (minor axis) of the ellipse (thus $b > a$),
- $c \in \mathbb{R}$, permits to give the ellipse orientation $\Omega = 0.5arctan(c/(b-a))$ (cf. Figure 2). When $a = b$ equation 5 becomes a circle equation (Ω will do not gives thus any more information).

The choice of ellipse box rather than circle as used in [11], [9] or [1] is to have one more generic and flexible mean to surround and fit accurately different kind of obstacles shapes (specifically longitudinal shapes [2]).

The surrounded ellipse parameters (h, k, A, B and Ω) (cf. equation 5 and figure 2) can be obtained on-line, while using an appropriate weighted least square method on the data range given by the robot infrared sensors [19]. An extension of this approach while using Extended Kalman Filter and an appropriate heuristic is given in [20].

A. Mobile robot model

Before proposing appropriate elementary controllers to achieve the considered task, it is important to know the robot model. Its model is given by the well known kinetic model of a unicycle robot (cf. Figure 2):

$$\dot{\xi} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \qquad (6)$$

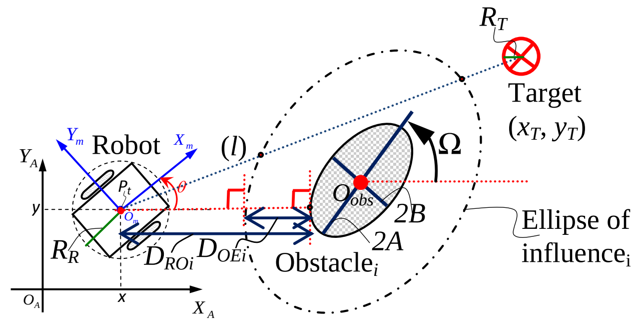


Fig. 2. Robot pose and the used perceptions for the navigation.

With x, y, θ correspond to configuration state of the unicycle and v and w correspond respectively to linear and angular velocity of the robot at the point "P_t".

Knowing the model of the robot as well as the task to achieve, we present below the controller of *Attraction to the target* and the *Obstacle avoidance* which are necessary to the mobile robot navigation in cluttered environment. In section III-C the control law used for the two controllers is presented.

B. The used controllers

1) Attraction to the target controller: This controller leads the robot toward the target to reach. This target is represented by a circle of (x_T, y_T) center and R_T radius (cf. Figure 2).

2) Obstacle avoidance controller: The objective of this controller is to avoid obstacles which hinder the robot movement toward the target. In what follows we will give only few details about the overall obstacle avoidance algorithm in order to make more the focus on the proposed mechanisms of control which can guarantee at the same time: the stability and the smoothness of the switch between controllers. More details about the proposed obstacle avoidance algorithm are given in [1] and [2].

To implement the obstacle avoidance behavior, limit-cycles was used [10], [1]. The differential equations giving elliptic limit-cycles are:

- For the clockwise trajectory motion (cf. Figure 3(a)):

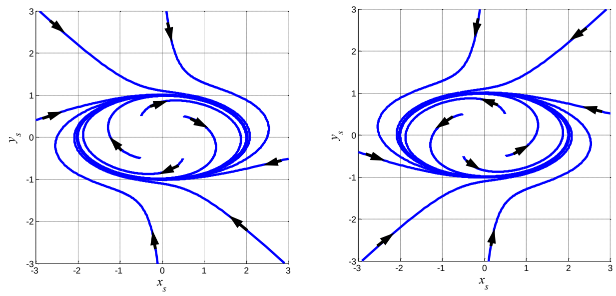
$$\begin{aligned} \dot{x}_s &= y_s + x_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_sy_s) \\ \dot{y}_s &= -x_s + y_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_sy_s) \end{aligned} \qquad (7)$$

- For the counter-clockwise trajectory motion (cf. Figure 3(b)):

$$\begin{aligned} \dot{x}_s &= -y_s + x_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_sy_s) \\ \dot{y}_s &= x_s + y_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_sy_s) \end{aligned} \qquad (8)$$

where (x_s, y_s) corresponds to the position of the robot according to the center of the ellipse; A_{lc} and B_{lc} characterize respectively major and minor elliptic axis (cf. Figure 2); c if $\neq 0$ gives the Ω ellipse angle (cf. Section III).

Figure 3 shows that the ellipse of a major axis = $2A_{lc}$ = 4 and of minor axis = $2B_{lc}$ = 2 is a periodic orbit. This periodic orbit is called a limit-cycle [10]. Figure 3(a) and 3(b) show the shape of equations (7) and (8) respectively.



(a) Clockwise (b) Counter-Clockwise

Fig. 3. Shape possibilities for the used elliptic limit-cycles for different initial conditions (x_0, y_0).

They show the direction of trajectories (clockwise or counter-clockwise) according to (x_s, y_s) axis. The trajectories from all points (x_s, y_s) of X, Y reference frame, including inside the ellipse, move towards the ellipse.

Summarily, the obstacle avoidance algorithm used in the paper follow these steps [2]:

- Detect the most disturbing obstacle which avoids the robot to reach the target, i.e., it is enough here to know if it exists an intersect points between the line "l" and the *Ellipse of influence* (cf. Figure 2). In fact, it is defined for each perceived obstacle an *Ellipse of influence* which has the following features:
  - The same center (h, k) and tilt angle Ω as the ellipse which surround the obstacle,
  - The value of its major axis is $2A_{lc}$ with $A_{lc} = A + R_R + \mathit{Margin}$,
  - The value of its minor axis is $2B_{lc}$ with $B_{lc} = B + R_R + \mathit{Margin}$.

  Where *Margin* corresponds to a safety tolerance which includes: perception uncertainty, control reliability and accuracy, etc.
- According to the relative position of the robot with regard to the disturbing obstacle and to the target to reach, the direction of avoidance (clockwise or counter-clockwise) is taken,
- The robot passes after by two steps, go into the orbit of the obstacle$_i$ to avoid (*Attractive phase*) and after go out the orbit of the obstacle$_i$ (*Repulsive phase*).

C. The used control law

To make a focus specifically around the efficiency of the proposed adaptive control mechanism a simple control law is used:

$$v = v_{\max} e^{-K_v/d} \cos(\theta_e) \tag{9a}$$
$$w = \dot{\theta}_d + K_p \theta_e \tag{9b}$$

where $v_{\max}$ is the robot maximum linear velocity, K_v and K_p are constant values $\in \mathbb{R}^+$. and d is the distance between the robot and the target when the *attraction to the target* controller is activated, and d is equal to D_{ROi} (cf. Figure 2) if the *obstacle avoidance* is activated. The robot reaches the target when $0 < d \leq R_T$ (cf. Figure 2). θ_e is the angular error given by:

$$\theta_e = \theta_d - \theta \tag{10}$$

The desired robot orientation θ_d is given according the following two cases:

1)
$$\theta_d = \arctan(\frac{y_T - y}{x_T - x}) \tag{11}$$
Where (x, y) and (x_T, y_T) correspond respectively to the position of the robot and the target (cf. Figure 2) in the case of the activation of *attraction to the target* controller.
2) and it is equal to:
$$\theta_d = \arctan(\frac{\dot{y}_s}{\dot{x}_s}) \tag{12}$$
Where $\dot{x}_s$ and $\dot{y}_s$ are given by differential equation of the limit-cycle (7) or (8)) in the case of the activation of *obstacle avoidance* controller.

It is interesting to note that only one control law is applied to the robot even if its control architecture contains two (or more) different controllers. Only the set-points change according to the applied controller.

In what follows, a study is given to use the adaptive control mechanism on the nominal angular control law (9b). While using (9b), it is straightforward to demonstrate that the evolution of $\dot{\theta}_e$ will be given by:

$$\dot{\theta}_e = -K_p \theta_e \tag{13}$$

To guarantee the right transition between controllers as described in section (II-C), the modification of the controller law must be done, it becomes thus:

$$w = \dot{\theta}_d + K_p \theta_e + G(t) \tag{14}$$

where $G(t)$ the adaptive function. $\dot{\theta}_e = \dot{\theta}_d - \dot{\theta}$ will be given now by:

$$\dot{\theta}_e = -K_p \theta_e - G(t) \tag{15}$$

Let's consider the following Lyapunov function

$$V = \tfrac{1}{2}\theta_e^2 \tag{16}$$

$\dot{V}$ is equal then to $\theta_e \dot{\theta}_e = -K_p \theta_e^2 - G(t)\theta_e$. To guarantee that the proposed controller is asymptotically stable, we must always have $\dot{V} < 0$, thus:

$$K_p > -\frac{G(t)}{\theta_e} \tag{17}$$

Where $G(t)$ is a function chosen with respect to the constraints given in sections (II-C and II-D) and to the fact that it decreases more quickly to zero than θ_e.

D. Hierarchical action selection block

The proposed control architecture uses a hierarchical action selection mechanism to manage the switch, between two or even more controllers, according to environment perception. Obstacle avoidance strategy is integrated in a more global multi-controller architecture. Otherwise, the controllers' activations are achieved in a reactive way as in [6]. The proposed algorithm 1 activates the obstacle avoidance controller as soon as it exists at least one obstacle which can obstruct the future robot movement toward its target.

if *It exists at least one constrained obstacle* $\{$*i.e., it exists at least one intersect point between the line "l" and the ellipse of influence (cf. Figure 2)*$\}$ **then**
  Activate *Obstacle avoidance* controller
else
  Activate *Attraction to the target* controller
end

Algorithm 1: Hierarchical action selection

E. Parameters adaptation block

In the applied navigation task, the "conditional" block activate the "parameters adaptation" block (cf. Figure 1) when at least one of the following switch events occurs:

- the "Hierarchical action selection" block chose to switch from one controller to another,
- the "obstacle avoidance" algorithm chose another obstacle to avoid,
- the "obstacle avoidance" controller switch from attractive phase to the repulsive phase (cf. Section III-B.2).

IV. Obstacle avoidance safety mode

The adaptive function $G(t)$ (cf. Equation 14) permits mainly to obtain smooth control when a switch event occurs. However, during "T" time (cf. Section II-C) the obstacle avoidance controller is far from its nominal law (given when $G(t) \neq 0$) and the robot can collide with obstacles. Therefore, to insure the smoothness of the control without neglecting the robot safety, G will be parameterized according to the robot-obstacle distance "$d = D_{RO_i}$" (cf. Figure 2), G becomes thus:

$$G(t,d) = Ae^{Bt} \tag{18}$$

Where:

- A value of the control difference between the control at the instants "$t - \Delta t$" and "t" (cf. Equation 3),
- $B = log\left(\varepsilon/|A|\right)/T(d)$ with:
  - ε very small constant value ≈ 0 (cf. Equation 4),
  - $\begin{cases} T(d) = T_{max} & \text{if } d > D_{OEi} \\ T(d) = c.d + e & \text{if } D_{OEi} \geq d \geq D_{OEi} - p.Margin \\ T(d) = \varepsilon & \text{if } d < D_{OEi} - p.Margin \end{cases}$

    Where:
    - D_{OEi} corresponds to the distance Obstacle-Ellipse of Influence (cf. Figure 2),
    - $Margin$ defined in sub-section III-B.2,
    - p positive constant < 1 which allows to adapt the maximum distance "d" where the adaptive function must be resetting to zero. As small as p is, more the priority is given to the safety behavior instead to the smoothness of controllers switch,
    - $c = T_{\max}/p.Margin$
    - $e = T_{max}(1 - D_{OEi}/p.Margin)$

Therefore, $T(d)$ goes from T_{max} until 0 while following a linear decrease. If the robot is out of D_{OEi} than $T = T_{max}$ and decrease linearly to become 0 when $d < D_{OEi} - (p.Margin)$. This function permits thus, when $d < D_{OEi} - (p.Margin)$, to remove completely the effect of adaptive control (which promote the smoothness of control) and insures thus the complete safety of the robot navigation.

V. Simulation results

In this section, many simulations on different robot configurations and cluttered environments will permits to confirm the reliability and the robustness of the proposed control architecture (cf. Figure 1). Figure 4 shows the smoothness of the obtained robot trajectories. It shows also the clockwise and counter-clockwise obstacle avoidance using on-line set-point based elliptic limit-cycle. In figure 4(a), it is showed the tracks of "limit-cycle planned path", which is not really followed by the robot, in fact, at each sample time, the robot computes the new control set-points given by equations (7) and (8). The showed planned track corresponds to the limit-cycle path obtained the first time that the robot see the obstacle to avoid, this trajectory do not take into account the robot constraints, e.g., its nonholonomy (cf. Equation 6).

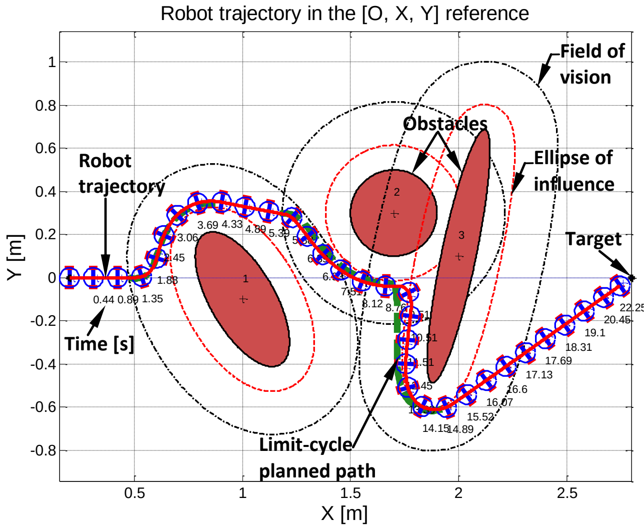
Figures 5 (c) and (d) shows respectively the progress of v and w robot velocities when the adaptive functions are used (cf. Equation 3). These controls are thus less abrupt and smoother than those obtained without adaptive functions (cf. Figures 5 (a) and (b)). In addition, to demonstrate the real smoothness enhancement of the obtained trajectories, a statistical survey was made while doing a large number of simulations in different cluttered environments and with for each one, a navigation with and without adaptive function is performed. We did specifically 1000 simulations with every time, 10 obstacles with different random positions in the environment (cf. figure 4(b) for an example of trajectory). Otherwise, to quantify the smoothness of the obtained robot trajectories [7], [8], it is proposed to use these two indicators:

$$I_v = \int_0^{T_T} |\dot{v}| dt \tag{19}$$

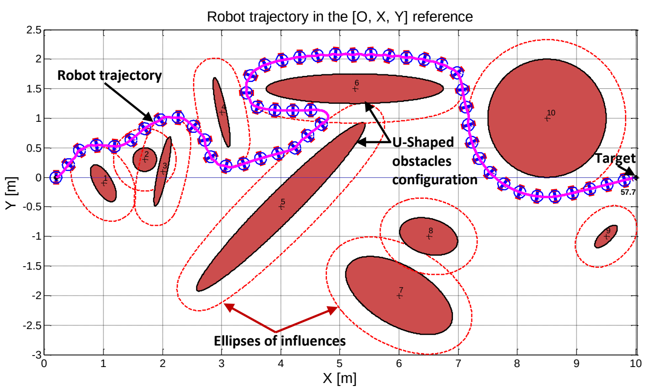
and

$$I_w = \int_0^{T_T} |\dot{w}| dt \tag{20}$$

Where $\dot{v}$ and $\dot{w}$ correspond respectively to linear and angular robot acceleration, and T_T is the necessary time, for



(a) Environment with 3 obstacles



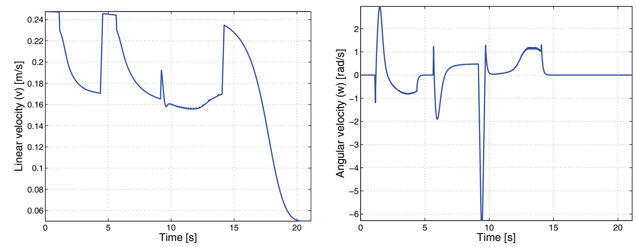
(b) Environment with 10 obstacles

Fig. 4. Some smooth robot trajectories obtained with the proposed on-line control architecture.

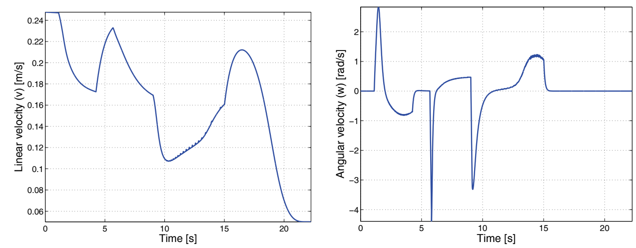
the robot, to reach the target. According to these indicators we can observe a significant gain in the smoothness of v and w controls which are equal respectively to 30% and 35%.

The second step of simulations permits to demonstrate the relevance of the proposed safety mode, especially when the robot navigates very close to obstacles. Figure 6 shows the case where obstacle avoidance controller apply or not the safety mode (cf. Section IV). When it do not apply it, the robot hit the obstacle (cf. Figure 6(a)). Figures 7 (a) and (b) give the evolution of adaptive functions when the safety mode is applied. We observe in these figures that the maximal time T_{max} to achieve the interpolation (=3s in the simulation (cf. Section IV)) decreases every time that the robot moves dangerously closer to the obstacle (cf. Figure 6(b)). Figure 7(c) shows that the overall proposed structure of control is always stable even when the adaptive safety mode is applied.

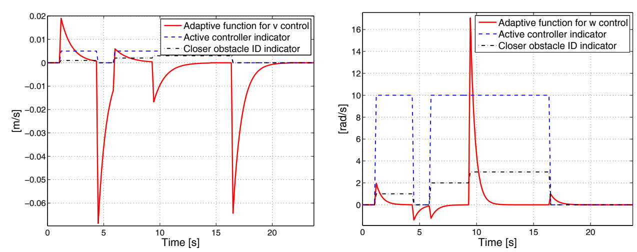
The two peaks shown in Figure 7(c) correspond respectively to the phase of the attraction toward the elliptic limit-cycle and the repulsion from this one. The applied algorithm is accurately explained in [1] and [2].



(a) v without AF (b) w without AF

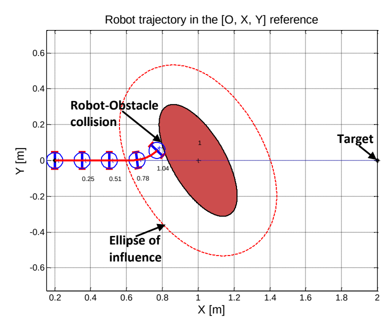


(c) v with AF (d) w with AF

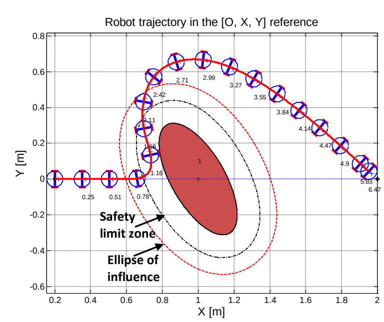


(e) AF evolution for v (f) AF evolution for w

Fig. 5. Adaptive Function (AF) influence on the v and w robot velocities (cf. Section III-C).

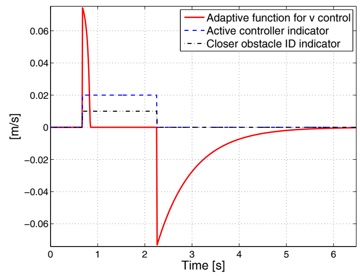


(a) Without safety mode

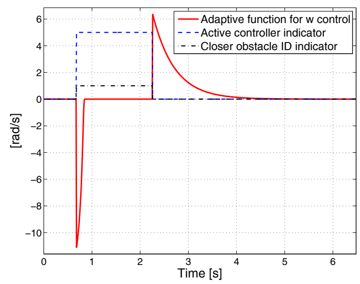


(b) With safety mode

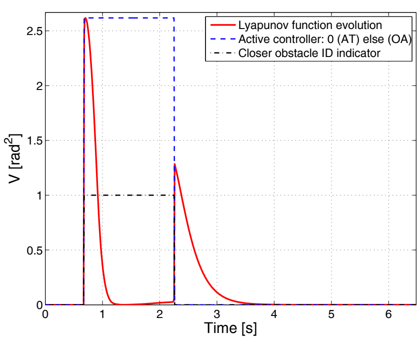
Fig. 6. Robot trajectories with and without safety mode.



(a) AF for v with SM, T_{max} = 3s



(b) AF for w with SM, T_{max} = 3s



(c) Lyapunov function

Fig. 7. Adaptive Functions (AF) with Safety Mode (SM) and Lyapunov function evolution.

VI. Conclusion and further work

This paper proposes to link, on-line control set-points using elliptic limit-cycle trajectories and a multi-mode control architecture which uses an adaptive mechanism to guarantee at the same time, the stability (according to Lyapunov synthesis) and the smoothness of the switch between controllers. Therefore, in addition to safe robot navigation, the robot trajectories become also smoother. Otherwise, appropriate indicators are used to quantify the trajectories smoothness. Moreover, to obtain safer robot navigation, an appropriate safety mode is proposed and experimented in cluttered environment. Many simulations confirm the reliability and the robustness of the proposed control architecture. Future works aim, first to apply this control architecture in real robots and secondly, to propose control architecture which find the right balance between reactive and cognitive aspects (planned).

References

[1] Adouane, L.: Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation. In: 9th Conference on Autonomous Robot Systems and Competitions. Portugal (2009)

[2] Adouane, L., Benzerrouk, A., Martinet, P.: Mobile robot navigation in cluttered environment using reactive elliptic trajectories. In: 18th IFAC World Congress. Italy (2011)

[3] Arkin, R.C.: Behavior-Based Robotics. The MIT Press (1998)

[4] Branicky, M.S.: Multiple lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Transaction on Automatic Control **43(4)**, 475–482 (1998)

[5] Brock, O., Khatib, O.: High-speed navigation using the global dynamic window approach. In: ICRA, pp. 341–346 (1999)

[6] Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation **RA-2**, pp.14–23 (1986)

[7] Fleury, S., Souères, P., Laumond, J.P., Chatila, R.: Primitives for smoothing mobile robot trajectories. In: ICRA (1), pp. 832–839 (1993)

[8] Gulati, S.: A framework for characterization and planning of safe, comfortable, and customizable motion of assistive mobile robots. Ph.D. thesis, The University of Texas at Austin (2011)

[9] Jie, M.S., Baek, J.H., Hong, Y.S., Lee, K.W.: Real time obstacle avoidance for mobile robot using limit-cycle and vector field method. Knowledge-Based Intelligent Information and Engineering Systems pp. 866–873 (2006)

[10] Khalil, H.K.: Frequency domain analysis of feedback systems. Nonlinear Systems: Chapter7, 3 edition (2002)

[11] Kim, D.H., Kim, J.H.: A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer. Robotics and Autonomous Systems **42(1)**, 17–30 (2003)

[12] Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Boston, MA (1991)

[13] Minguez, J., Lamiraux, F., Laumond, J.P.: Handbook of Robotics, chap. Motion Planning and Obstacle Avoidance, pp. pp.827–852 (2008)

[14] Morin, P., Samson, C.: Control of nonholonomic mobile robots based on the transverse function approach. Trans. Rob. **25**, 1058–1073 (2009)

[15] Mouad, M., Adouane, L., Khadraoui, D., Martinet, P.: Mobile robot navigation and obstacles avoidance based on planning and re-planning algorithm. In: 10th International IFAC Symposium on Robot Control (SYROCO12). Dubrovnik - Croatia (2012)

[16] Ogren, P., Leonard, N.E.: A convergent dynamic window approach to obstacle avoidance. IEEE Transactions on Robotics **21**(2), 188–195 (2005)

[17] Pivtoraiko, M., Kelly, A.: Fast and feasible deliberative motion planner for dynamic environments. In: International Conference on Robotics and Automation (2009)

[18] Rimon, E., Koditschek, D.E.: Exact robot navigation using artficial potential fields. IEEE Transactions on Robotics and Automation **8(5)**, 501–518 (1992)

[19] Vilca, M., Adouane, L., Mezouar, Y.: On-line obstacle detection using data range for reactive obstacle avoidance. In: 12th International Conference on Intelligent Autonomous System (IAS'12). Korea (2012)

[20] Vilca, M., Adouane, L., Mezouar, Y.: Robust on-line obstacle detection using data range for reactive navigation. In: 10th International IFAC Symposium on Robot Control (SYROCO'12). Dubrovnik - Croatia (2012)

[21] Zefran, M., Burdick, J.W.: Design of switching controllers for systems with changing dynamics. In: IEEE Conference on Decision and Control CDC'98, Tampa, FL, pp. 2113–2118 (1998)