Hybrid and Safe Control Architecture for Mobile Robot Navigation

Lounis Adouane
LASMEA, UBP-UMR CNRS 6602, France
Email: Lounis.Adouane@lasmea.univ-bpclermont.fr

***Abstract*— This paper deals with a multi-mode control architecture for robot navigation while using hybrid control. It presents, an adaptive and flexible mechanism of control which guarantees the stability and the smoothness of the switch between controllers. Moreover, a specific safety mode is proposed and applied on the robot which navigates very close to obstacles. The overall architecture allows to obtain very smooth trajectories while guaranteeing very safe obstacle avoidance. Many simulations on different robot configurations and cluttered environments permits to confirm the reliability and the robustness of the proposed control architecture. In addition, an appropriate indicator is proposed to quantify the trajectory smoothness.**

I. Introduction

The control of mobile robot navigation in cluttered environment is a fundamental problem that has been receiving a large amount of attention. The main issues in this field is how to obtain 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. Voronoï diagrams and visibility graphs [1] or navigation functions [2] are among these roadmap-based methods. However, the other part of the literature considers that to control a robot with safety, flexibility and reliability, it is essential to accurately take into account: robot's structural constraints (e.g., nonholonomy); avoid command discontinuities and set-point jerk, etc. Nevertheless, even in this method, there are two schools of thought, one uses the notion of planning and re-planning to reach the target, e.g., [3] and [4] and the other more reactive (without planning) like in [5], [6] or [7]. Our proposed control architecture is linked to this last approach. Therefore, where the stability of robot control is rigourously demonstrated and the overall robot behavior is constructed with modular and bottom-up approach [8].

To guarantee multi-objective criteria, control architectures can be elaborated in a modular and bottom-up way as introduced in [9] and so-called behavioral architectures [8]. 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. In this kind of control, it exists two major principles for behavior coordination: action selection and fusion of actions which lead respectively to competitive and cooperative architectures of control. In competitive architectures (action selection), the set-points sent to the robot actuators at each sample time are given by a unique behavior which has been selected among a set of active behaviors. The principle of competition can be defined by a set of fixed priorities like in the subsumption architecture [9] where a hierarchy is defined between the behaviors. The action selection can also be dynamic without any hierarchy between behaviors [10], [11]. In cooperative architectures (fusion of actions), the set-points sent to the robot actuators are the result of a compromise or a fusion between controls generated by different active behaviors. These mechanisms include fuzzy control [12] *via* the process of defuzzification, or the multi-objective techniques to merge the controls [13]. Among these cooperative architectures, schema-based principle [14], [8] is among the ones that has important diffusion in the scientific community. Moreover, it is considered in a lot of studies the investigation of the potentialities of the hybrid systems controllers [15] 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 comprised of a finite state automaton, whose states correspond to a continuous dynamic evolution, and whose transitions can be enabled by particular conditions reached by the continuous dynamics themselves. Therefore, this formalism permits a rigorous automatic control analysis of the performances of the control architecture [16].

Specifically, 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 [17]. The proposed control architecture integrates obstacle avoidance method which uses limit-cycle vector field [18], [19], [20]. Moreover, it introduces an adaptive 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.

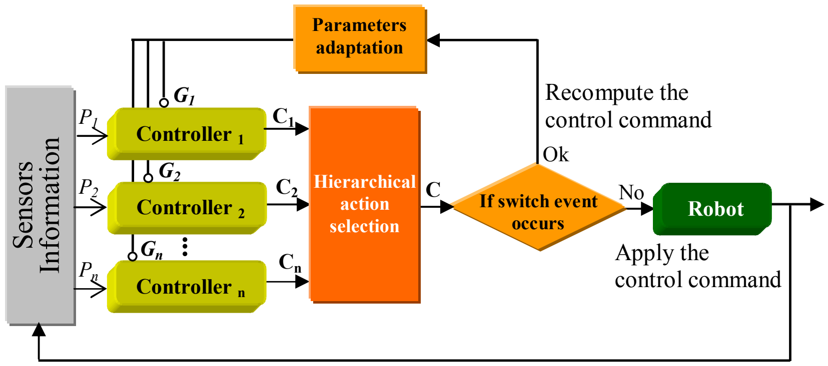


Fig. 1. The proposed hybrid control architecture for mobile robot navigation

II. Control Architecture

The proposed control architecture (cf. Figure 1) is dedicated for mobile robots navigation in presence of obstacles. It permits to manage the interactions between different elementary controllers while guaranteing the stability and the smoothness of the overall control. Moreover, a specific "safety mode" is proposed to avoid undesirable robot behaviors. The robot can therefore have very smooth trajectories while guaranteeing safe obstacle avoidance. This control architecture permits for example to an autonomous applications of travelers transportation [21] to have more comfortable displacements while guaranteeing the security of passengers. The specific blocks composing this control are detailed below. Concrete control architecture applied in real task is proposed in section III.

A. Hierarchical action selection

The activation of one controller in favor of another is achieved completely with a hierarchical manner like the principle of the subsumption proposed initially by Brooks in [9]. Therefore, specific stimuli perceived by the robot (e.g., the robot-obstacle distance) are responsible to trigger the switch between controllers behaviors.

B. Controllers

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

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

with:

- P_i perceptions useful to the controller "i",
- S_i set-points given 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:

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

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

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

where Δt represents the sampling time between two control set-points.

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

$$G_i(P_i, S_i, T) = 0 \tag{4}$$

The function of adaptation $G_i(P_i, S_i, t)$ is updated by the "Parameters adaptation" block every time a hard control switch concerning the "i" controller occurs (cf. Section II-C) (cf. Figure 1). The main challenge introduced by this kind of control structure is to guarantee the stability of the updated control law (cf. Equation 2) during the period where $G_i(P_i, S_i, t) \neq 0$.

C. Parameters adaptation

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 "adaptation 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-Δt" 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 an unstructured environment task has as objective to lead the robot to reach specific position in its environment (the target) while avoiding obstacles (cf. Figure 2).

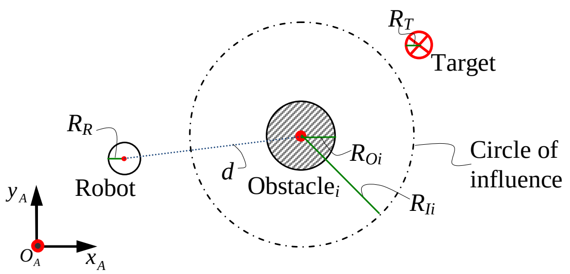


Fig. 2. The used perceptions for mobile robot navigation

The robot trajectory need to be safe, smooth and fast. One supposes in the setup that obstacles and the robot are surrounded by bounding cylindrical boxes with respectively R_O and R_R radii [22]. The target to reach is also characterized by a circle of R_T radius. Several perceptions are also necessary for the robot navigation (cf. Figure 2):

- d distance between the robot and the obstacle "i",
- R_{Oi} radius of the obstacle "i" to avoid,
- For each detected obstacle we define a *circle of influence* with a radius of $R_{Ii} = R_R + R_{Oi} + Margin$. *Margin* corresponds to a safety tolerance which includes: perception incertitude, control reliability and accuracy, etc.

A. Model of the used robot

Before proposing appropriate elementary controllers to achieve the considered task, it is important to know the robot model. Its model is given by the kinetic model of a unicycle robot which is given by (cf. Figure 3):

$$\dot{\xi} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & -l_2\cos\theta - l_1\sin\theta \\ \sin\theta & -l_2\sin\theta + l_1\cos\theta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \quad (5)$$

with:

- x, y, θ: configuration state of the unicycle at the point "P_t" of abscissa and ordinate (l_1, l_2) according to the mobile reference frame (X_m, Y_m),
- v: linear velocity of the robot at the point "P_t",
- w: angular velocity of the robot at the point "P_t".

Knowing the model of the robot as well as the task to achieve, one presents below the controller of *Attraction to the*

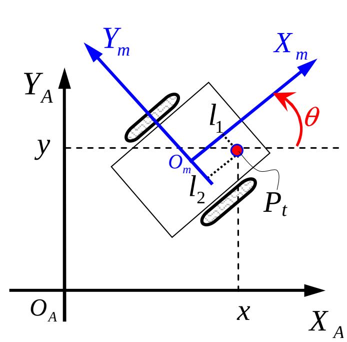


Fig. 3. Robot configuration in a cartesian reference frame

target and the *Obstacle avoidance* controller which are necessary to the mobile robot navigation in presence of obstacles. The set of these controllers will be synthesized while using the Lyapunov theorem.

B. Attraction to the target controller

This controller guides the robot toward the target which is represented by a circle of center (x_T, y_T) and of R_T radius (cf. Figure 2). The used control law is a control of position at the point P_t = (l_1, 0) (cf. Figure 3). As we consider a circular target with R_T radius, thus, to guarantee that the center of robot axis reaches the target with asymptotical convergence, l_1 must be $\leq R_T$.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos\theta & -l_1\sin\theta \\ \sin\theta & l_1\cos\theta \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = M \begin{pmatrix} v \\ w \end{pmatrix} \quad (6)$$

with M invertible matrix.

The errors of position are: $\begin{cases} e_x = x - x_T \\ e_y = y - y_T \end{cases}$

The position of the target is invariable according to the absolute reference frame (cf. Figure 2) $\Rightarrow \begin{cases} \dot{e}_x = \dot{x} \\ \dot{e}_y = \dot{y} \end{cases}$

Classical techniques of linear system stabilization can be used to asymptotically stabilize the error to zero [23]. We use a simple proportional controller which is given by:

$$\begin{pmatrix} v \\ w \end{pmatrix} = -K \begin{pmatrix} \cos\theta & -l_1\sin\theta \\ \sin\theta & l_1\cos\theta \end{pmatrix}^{-1} e = - K \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta/_{l_1} & \cos\theta/_{l_1} \end{pmatrix} \begin{pmatrix} e_x \\ e_y \end{pmatrix} \quad (7)$$

with $K > 0$ and $l_1 \neq 0$ (cf. Figure 3).

To guarantee the right transition between controllers as described in section (II-B), the modification of the controller law (7) must be done, it becomes thus:

$$\begin{pmatrix} v \\ w \end{pmatrix} = -K \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta/_{l_1} & \cos\theta/_{l_1} \end{pmatrix} \begin{pmatrix} e_x \\ e_y \end{pmatrix} + \begin{pmatrix} G_{A\_v}(t) \\ G_{A\_w}(t) \end{pmatrix} \quad (8)$$

Let's consider the following Lyapunov function

$$V_1 = \tfrac{1}{2}d^2 \quad (9)$$

with $d = \sqrt{e_x^2 + e_y^2}$ (distance robot-target). The proposed controller is asymptotically stable if $\dot{V}_1 < 0$. After some simplification we can deduce that:

$$K > \frac{-(G_{A\_v}(t)e_x + G_{A\_w}(t)e_y)}{e_x^2 + e_y^2} \quad (10)$$

As we said above $G_{A\_v}(t)$ and $G_{A\_w}(t)$ functions which must be chosen with respect to the constraints given in section (II-B). In fact, the absolute value of these functions must be monotonically decreasing according to the time "t", they will be equal to zero after a certain time "T". Therefore, in order to have always bounded K, we must have: $-(G_{A\_v}(t)e_x + G_{A\_w}(t)e_y) \leq e_x^2 + e_y^2$. Thus, to guarantee this assertion, it is sufficient to impose that $G_{A\_v}(t)$ decreases more quickly to zero than e_x and also that $G_{A\_w}(t)$ decreases more quickly to zero than e_y.

C. Obstacle avoidance controller

The objective of this controller is to avoid obstacles which hinder the robot movement toward the objective. In what follows we will give only few details about the overall obstacle avoidance algorithm in order to focus the attention only around the proposed mechanisms of control which can guarantee at the same time: the stability and the smoothness of the switch between controllers. Accurate details about the proposed obstacle avoidance algorithm is given in [20].

To implement the obstacle avoidance behavior, limit-cycles was used [18], [24], [20]. The differential equations giving these desired robot trajectories are given by two differential equations:

- For the clockwise trajectory motion (cf. Figure 4(a)):

$$\begin{aligned} \dot{x}_s &= y_s + x_s(R_c^2 - x_s^2 - y_s^2) \\ \dot{y}_s &= -x_s + y_s(R_c^2 - x_s^2 - y_s^2) \end{aligned} \quad (11)$$

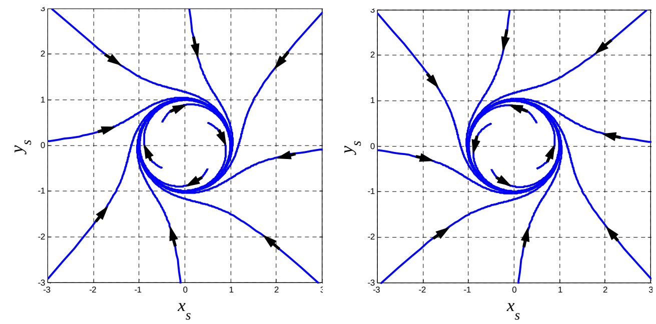
- For the counter-clockwise trajectory motion (cf. Figure 4(b)):

$$\begin{aligned} \dot{x}_s &= -y_s + x_s(R_c^2 - x_s^2 - y_s^2) \\ \dot{y}_s &= x_s + y_s(R_c^2 - x_s^2 - y_s^2) \end{aligned} \quad (12)$$

where $(x_s,\ y_s)$ corresponds to the position of the robot according to the center of the convergence circle which is characterized by an R_c radius. Figure 4 shows that the circle of "$R_c = 1$" is a periodic orbit. This periodic orbit is called a limit-cycle. Figure 4(a) and 4(b) show the shape of equations (11) and (12) respectively. 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) including inside the circle, move towards the circle.

Summarily, the obstacle avoidance algorithm [20] follow these steps:

- Detect the most disturbing obstacle which avoids the robot to reach the target (cf. Figure 2). (x_{Oi}, y_{Oi}) and R_{Ii} are respectively, the position and the radius of influence circle of corresponding obstacle$_i$. (x_{Oi}, y_{Oi}) constitutes the center of the limit-cycle.
- According to specific stimuli, the direction of avoidance (clockwise or counter-clockwise) is obtained,
- Robot go into the orbit of the obstacle$_i$ to avoid (**Attractive phase**). The radius of the limit cycle to follow is given by $R_c = R_{Ii} - \xi$, , with ξ a small constant value as $\xi \ll$ *Margin* (cf. Section III) [20].



(a) Clockwise (b) Counter-Clockwise

Fig. 4. Shape possibilities for the used limit-cycles

- Robot go out the orbit of the obstacle$_i$ (**Repulsive phase**). The radius of the limit cycle to follow is given by $R_c = R_c + \xi$.

Controller law definition: The proposed control law which permits to follow these trajectories is an orientation control, the robot is controlled according to the center of its axle, i.e., while taking $(l_1,\ l_2)$ = (0, 0) (cf. Figure 3). The desired robot orientation θ_d is given by the differential equation of the limit-cycle (11) or (12) as:

$$\theta_d = \arctan(\frac{\dot{y}_s}{\dot{x}_s}) \quad (13)$$

and the error by

$$\theta_e = \theta_d - \theta \quad (14)$$

We control the robot to move to the desired orientation by using the following nominal control law:

$$w = \dot{\theta}_d + K_p\theta_e \quad (15)$$

with K_p a constant > 0 and $\dot{\theta}_e$ is given by:

$$\dot{\theta}_e = -K_p\theta_e \quad (16)$$

To guarantee the right transition between controllers as described in section (II-B), the modification of the controller law (7) must be done, it becomes thus:

$$w = \dot{\theta}_d + K_p\theta_e + G_O(t) \quad (17)$$

where $G_O(t)$ the adaptive function.

$\dot{\theta}_e$ is given then by:

$$\dot{\theta}_e = -K_p\theta_e - G_O(t) \quad (18)$$

Let's consider the following Lyapunov function

$$V_2 = \tfrac{1}{2}\theta_e^2 \quad (19)$$

$\dot{V}_2$ is equal then to $\theta_e\dot{\theta}_e = -K_p\theta_e^2 - G_O(t)\theta_e$. To guarantee that the proposed controller is asymptotically stable we must have $\dot{V}_2 < 0$, so:

$$K_p > -\frac{G_O(t)}{\theta_e} \quad (20)$$

where $G_O(t)$ function is chosen with respect to constraints given in section II-C and to the fact that it decreases more quickly to zero than θ_e.

D. Hierarchical action selection block

The activation of a controller in favor to another is achieved according to complete hierarchy as given below:

if *It exists at least one constrained obstacle.* $\{$*i.e.,* $d \leq R_{Ii}$ *(cf. Figure 2)* $\}$ **then**
| Activate obstacle avoidance controller
else
| Activate the attraction to the target controller
end

Algorithm 1: Hierarchical action selection

E. Parameters adaptation block

In the applied navigation, 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 an other obstacle to avoid,
- the "obstacle avoidance" controller switch from attractive phase to the repulsive phase (cf. Section III-C).

IV. Obstacle avoidance safety mode

The adaptive function $G_O(t)$ (cf. Equation 17) permits mainly to obtain smooth control when a switch event occurs. However, during "T" time (cf. Section II-B) the obstacle avoidance controller is far from its nominal law (given when $G_O(t) \neq 0$) and the robot can collide with obstacles [7]. Therefore, to insure the smoothness of the control without neglecting the robot safety, G_O will be parameterized according to the robot-obstacle distance "d" (cf. Figure 2), G_O becomes thus:

$$G_O(t, d) = A.e^{Bt} \tag{21}$$

where:

- A value of the control difference between the control at the instants "$t - \delta t$" and "t" (cf. Equation 3),
- $B = Log\left(\varepsilon/|A|\right) 1/T(d)$

  with:

  - ε very small constant value ≈ 0,
  - $\begin{cases} T(d) = T_{max} & \text{if } d > R_{Ii} \\ T(d) = c.d + e & \text{if } R_{Ii} \geq d \geq R_{Ii} - (p.Margine) \\ T(d) = 0 & \text{if } d < R_{Ii} - (p.Margine) \end{cases}$

  where:

  - $Margine$ defined in section III,
  - 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,
  - $c = \left[T_{max}/p.Margine\right]$
  - $e = \left[T_{max}(Margine - R_{Ii}/p)\right]/Margine$

Therefore, $T(d)$ goes from T_{max} until 0 while following a linear decrease. If the robot is out of R_{Ii} than $T = T_{max}$ and decrease linearly to become 0 when $d < R_{Ii} - (p.Margine)$. This function permits thus, when $d < R_{Ii} - (p.Margine)$, 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

Figure 5 shows the smoothness of the robot trajectory when the proposed control architecture is applied in cluttered environment. It shows also the clockwise and counter-clockwise robot obstacle avoidance. Figure 6 shows the progress of v and w controls when the adaptive functions are used. These controls are thus less abrupt and smoother than those obtained without adaptive functions (cf. Figure 7).

Moreover, to quantify the smoothness of the control set-points, we propose this two indicators:

$I_v = \int_0^{T_{Simulation}} |v'|dt$ and $I_w = \int_0^{T_{Simulation}} |w'|dt$

where v' and w' are the derivative functions of v and w. According to these indicators we can observe a significant gain in smoothness of v and w controls which are equal respectively to 6% and 50%.

The seconde step of simulations permits to demonstrate the relevance of the proposed safety mode specially when the robot navigate very close to obstacles. Figure 8 shows the case where obstacle avoidance controller apply and do not apply the safety mode (cf. Section IV). When it do not apply it, the robot hit the obstacle (cf. Figure 8(a)).

Figure 9 gives the progress of adaptive function when the safety mode is applied (cf. Figure 9(b)) or not (cf. Figure 9(a)). We observe in figure 9(b) that the maximal time T_{max} to achieve the interpolation decreases every time that the robot moves dangerously closer to the obstacle.

Figure 10 shows that the overall proposed structure of control is stable, and that the Lyapunov function attributed to each controller $V_i|_{i=1..2}$ decreases always asymptotically to

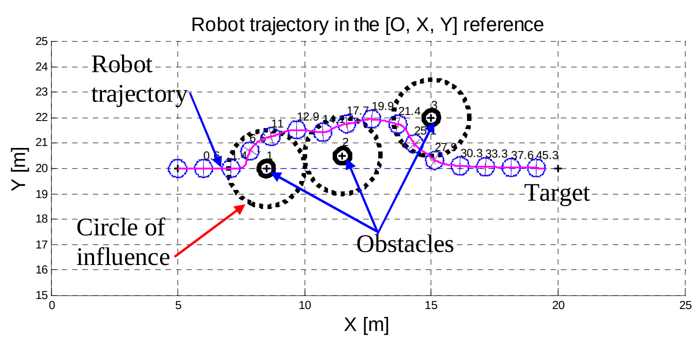


Fig. 5. Smooth robot trajectory obtained with the proposed control architecture

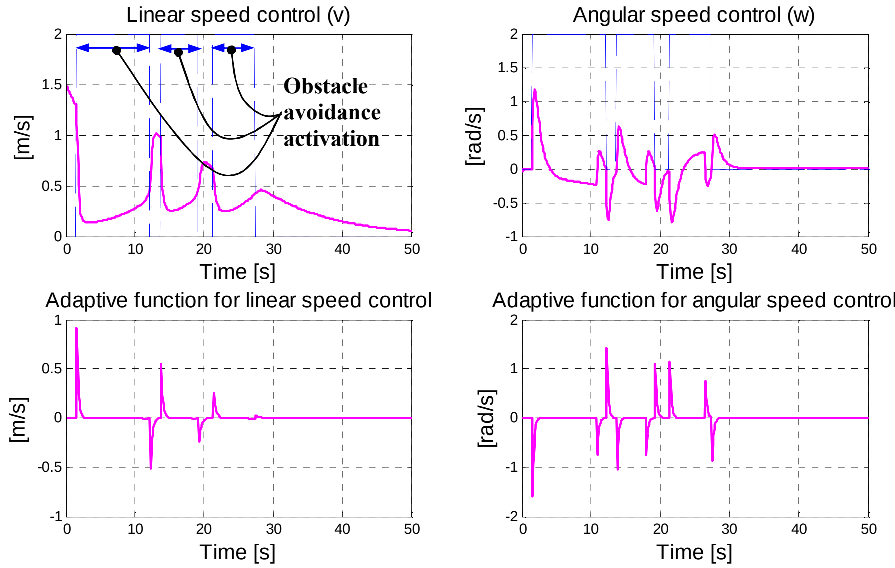


Fig. 6. Control with adaptive mechanism

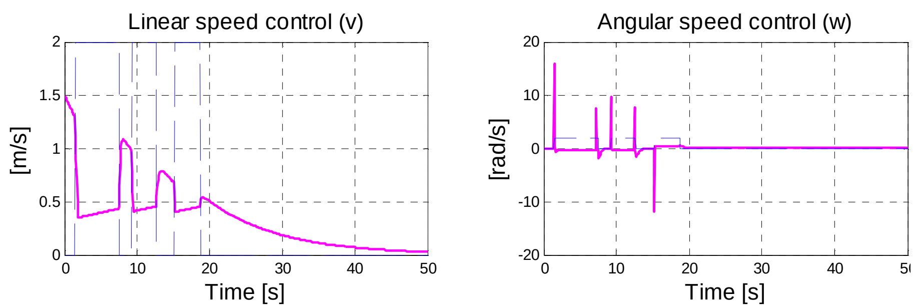


Fig. 7. Control without adaptive mechanism

the equilibrium point even when the adaptive safety mode is applied.

VI. Conclusion and Further Work

In this paper, a hybrid and safe multi-controller architecture is proposed and applied to the navigation of mobile robot in cluttered environments. The stability and the smoothness of the switching between these multi-control's modes are guaranteed according to a specific adaptive mechanism. Moreover, to obtain safer robot navigation an appropriate safety mode is proposed and experimented in cluttered environment. The robot can therefore have very smooth trajectories while guaranteeing obstacle avoidance. Many simulations confirm the robustness of the proposed control architecture. Future work will first test the proposed control architecture on the CyCab vehicle [21]. The second step is to adapt the proposed control structure to more complex tasks like navigation in highly dynamical environments.

References

[1] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

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

[3] C. Belta, V. Isler, and G. J. Pappas, "Discrete abstractions for robot motion planning and control in polygonal environments," *IEEE Transactions on Robotics*, vol. 21(5), pp. 864–874, Oct. 2005.

[4] D. C. Conner, H. Choset, and A. Rizzi, "Integrated planning and control for convex-bodied nonholonomic systems using local feedback," in *Proceedings of Robotics: Science and Systems II*. Philadelphia, PA: MIT Press, August 2006, pp. 57–64.

[5] M. Egerstedt and X. Hu, "A hybrid control approach to action coordination for mobile robots," *Automatica*, vol. 38(1), pp. 125–130, 2002.

[6] J. Toibero, R. Carelli, and B. Kuchen, "Switching control of mobile robots for autonomous navigation in unknown environments," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 1974–1979.

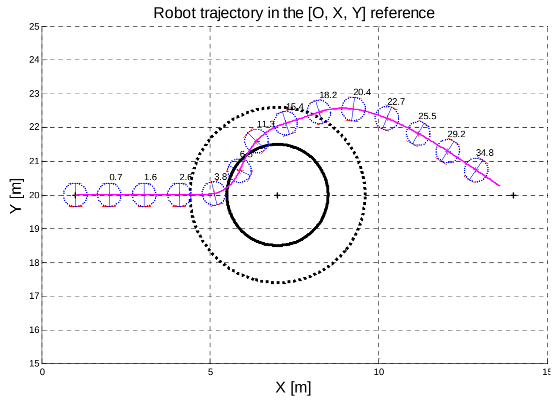
[7] L. Adouane, "An adaptive multi-controller architecture for mobile robot navigation," in *10th IAS, Intelligent Autonomous Systems*, Baden-Baden, Germany, July 23-25 2008, pp. 342–347.

[8] L. Adouane and N. Le Fort-Piat, "Behavioral and distributed control architecture of control for minimalist mobile robots," *Journal Européen des Systèmes Automatisés*, vol. 40, no. 2, pp. pp.177–196, 2006.

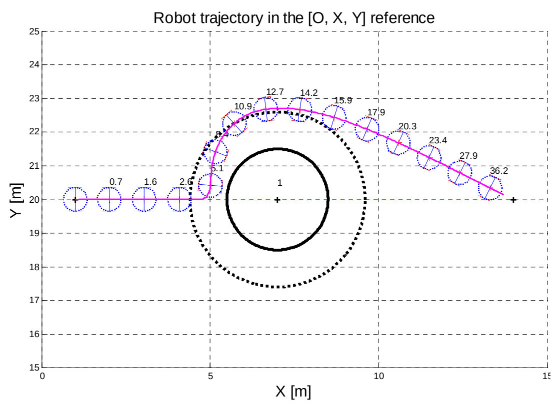
[9] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. RA-2, pp. pp.14–23, March 1986.

[10] P. Maes, "The dynamics of action selection," in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI)*, Detroit, 1989, pp. 991–97.

[11] M. J. Matarict', M. Nilsson, and K. Simsarian, "Cooperative multi-robots

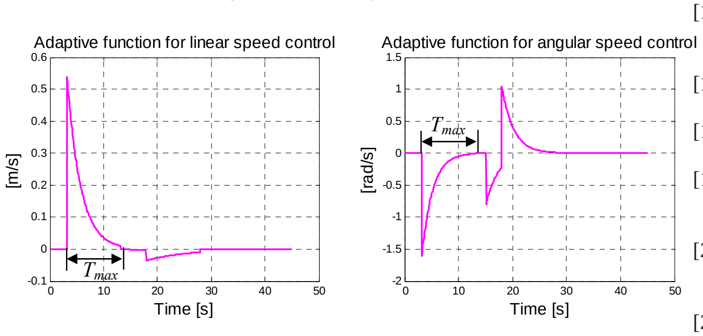


(a) Without safety mode

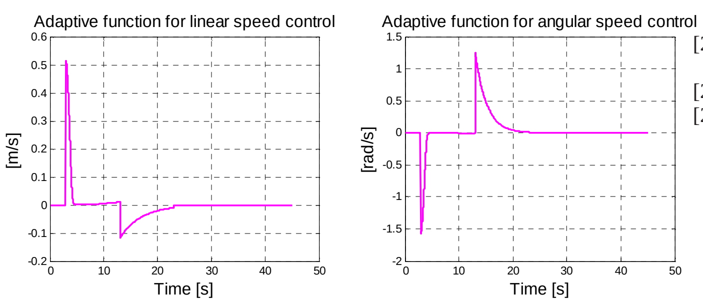


(b) With safety mode

Fig. 8. Robot trajectories



(a) Without safety mode



(b) With safety mode

Fig. 9. Adaptive function progress

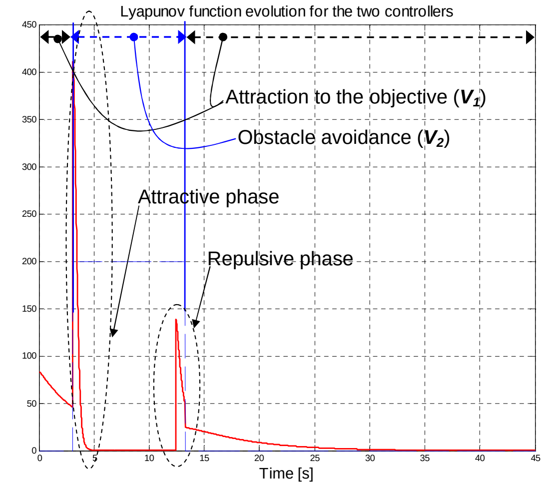


Fig. 10. V_1 and V_2 Lyapunov functions evolution when the safety mode is used

box-pushing," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, 1995, pp. 556–561.

[12] M. Wang and J. N. Liua, "Fuzzy logic-based real-time robot navigation in unknown environment with dead ends," *Robotics and Autonomous Systems*, vol. 56, no. 7, pp. pp.625–643, 2008.

[13] P. Pirjanian, "Multiple objective behavior-based control," *Journal of Robotics and Autonomous Systems*, vol. 31, no. 1, pp. pp.53–60, April 2000.

[14] R. C. Arkin, "Motor schema-based mobile robot navigation," *International Journal of Robotics Research*, vol. 8, no. 4, pp. pp.92–112, 1989.

[15] M. Zefran and J. W. Burdick, "Design of switching controllers for systems with changing dynamics," in *IEEE Conference on Decision and Control CDC'98, Tampa, FL*, 1998, pp. 2113–2118.

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

[17] J. Minguez, F. Lamiraux, and J.-P. Laumond, *Handbook of Robotics*, 2008, ch. Motion Planning and Obstacle Avoidance, pp. pp.827–852.

[18] H. K. Khalil, *Frequency domain analysis of feedback systems*, P. Hall, Ed. Nonlinear Systems: Chapter7, 3 edition, 2002.

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

[20] L. Adouane, "Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation," in *9th Conference on Autonomous Robot Systems and Competitions*, Portugal, May 2009.

[21] C. Pradalier, J. Hermosillo, C. Koike, C. Braillon, P. Bessière, and C. Laugier, "The cycab: a car-like robot navigating autonomously and safely among pedestrians," *Robotics and Autonomous Systems*, vol. 50, no. 1, pp. 51–68, 2005.

[22] M. Yerry and M. Shephard, "A modified quadtree approach to finite element mesh generation," *Computer, Graphics and Applications*, 1983.

[23] J.-P. Laumond, *La robotique mobile*. Hermès, 2001.

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