Multi Lyapunov Function Theorem Applied to a Mobile Robot Tracking a Trajectory in Presence of Obstacles

Ahmed Benzerrouk, Lounis Adouane, Philippe Martinet and Nicolas Andreff

`firstname.lastname@lasmea.univ-bpclermont.fr`

Abstract— In this paper, a reactive control architecture based on hybrid systems (continuous/discrete) is used to control a unicycle mobile robot tracking a given trajectory while avoiding obstacles. The main motivation of using hybrid systems is the possibility to define the overall control scheme as a combination of several elementary controllers (trajectory tracking, obstacle avoidance) that stability can be easily proved. However, there is a serious risk of oscillatory switching or even instability caused by random switch between these two elementary controllers. The contribution of this paper is to use the multiple Lyapunov functions (MLF) theorem to prove the global stability of a trajectory tracking task in presence of obstacles. To satisfy the MLF conditions, we propose to introduce a third controller in the architecture of control: the go-to-goal controller. Its role is to satisfy the second and the most difficult condition of MLF in a finite time. The approach is validated by numerical simulation.

I. INTRODUCTION

Controlling a nonholonomic mobile robot to follow a desired trajectory is a large topic of research investigated since a long time (see [10], [5], for instance). However, when controllers are developed for this specific task, it is important to take into account the environment variations. Indeed, it's obvious that these controllers become easily useless and lead to collision when an obstacle appears on the robot's way since they were only designed to track trajectory. However, the obstacle avoidance is not a problem anymore since it is widely investigated in the literature. Khatib [8] assumes that the robot moves in a potential field considering the objective to reach as an attractive point whereas the obstacle surfaces are repulsive fields. To minimize its local minima problems, circular potential fields were used [11]. Zapata and al [13] use a deformable virtual zone (DVZ) surrounding the robot thanks to proximity sensors: if an obstacle is detected, it will deform the DVZ and the approach is to minimize this deformation by modifying the control vector. These methods improve obstacle avoidance task but local minima problems were not completely solved. To perform the obstacle avoidance behavior, limit-cycle navigation proposed by Kim and Kim [9] can be used.

The unknown nature of the robot's environment leads up to develop control architectures which guarantee a desired and a safe navigation in presence of obstacles. In fact, the intuitive idea to make a mobile robot able to avoid obstacles while tracking the desired trajectory is to have two simple controllers and to switch from one to another according to the robot's relative position to the obstacle. Brooks [3] proposes a behavior based architecture where each layer accomplishes an elementary task. Therefore, it

becomes possible and important to study each controller and examine it independently from the whole control system. Theorems for hybrid systems seem consequently the most suitable to combine elementary controllers without loosing the global stability. Branicky in [2], shows that random switches between any stable systems do not guarantee the stability of the overall control. He consequently imposes restrictions on switching via multiple Lyapunov function. Moreover, with automata approach where each node gives the control law applied to the robot, hard switch may lead to the Zeno phenomenon [7] that exhibits an infinite number of discrete transitions between controllers in finite time. It potentially appears when the robot is on the boundary where the discrete event actuating switch becomes true. Effects caused by this phenomenon were shown by Egerstedt [6]. The latter regularizes its automaton by adding a node to overcome these undesirable effects: this node contains the sliding dynamics that is defined on the boundary between the two controllers. It comes to use more than one controller to control simultaneously the robot. The advantage of having each controller in a distinct node is then lost. Therefore, sliding dynamics seem not to be the optimal solution for robotics application. In this paper, it is proposed to apply theoretic study of multiple Lyapunov function to guarantee restrictions on switching. Adouane in [1], proposes to avoid this oscillatory switching between controllers commands while introducing a specific adaptation of each controller law.

Here, our idea is to introduce a third controller (go-to-goal) which leads the robot on its trajectory after the obstacle avoidance step. This one allows to verify the multiple Lyapunov function theorem. Moreover, the used automaton is regularized by adding a third node corresponding to this controller: the go-to-goal node. Thus, undesirable effects are avoided without using sliding mode. The proposed automaton has then only one controller in each node. It will be proved that this controller achieves the desired task in a finite time.

The rest of this paper is organized as follows. In section II, we give the used mobile robot model and Individual controllers. Details on the multiple Lyapunov function theorem with its application in the proposed control architecture is given in section III. Convergence of the proposed architecture is proved in section IV. Simulation results, are given in section V. We conclude and give some prospects in section VI.

II. ROBOT MODEL AND ELEMENTARY CONTROLLERS

A. Robot model

Considering the unicycle mobile robot (cf. Figure 1), let s,e and $\tilde{\theta}$ be the state variables where $s \in R$ and $e \in R$ are the Frenet frame coordinates (curvilinear and lateral coordinates respectively) of the center of the wheels axle, $\tilde{\theta} \in]-\pi, \pi]$ is the robot orientation with respect to the X_r axis of the Frenet frame. Linear and angular velocities of the robot are respectively noted v and ω. The kinematic model of the unicycle can be described by the well-known equations (cf. Equation 1).

$$\begin{aligned} \dot{s} &= \frac{v.cos(\tilde{\theta})}{1-ec(s)} \\ \dot{e} &= v.sin(\tilde{\theta}) \\ \dot{\tilde{\theta}} &= \omega - \dot{s}c(s) \end{aligned} \quad (1)$$

where $\frac{1}{c(s)}$ is the curvature radius in the point of coordinate s.

To accomplish a trajectory tracking task in presence of obstacles, a classical architecture of control has to contain a controller responsible of obstacle avoidance. Two main controllers are then requested.

B. Trajectory tracking controller

Consider the lateral and the angular errors of the robot noted e and $\tilde{\theta}$ respectively (cf. Figure 1). Tracking a reference trajectory with a stable law means that e and $\tilde{\theta}$ decrease always to 0. The following controller based on the Lyapunov stability allows that. It is developed in [4] and is expressed as follows:

$$\begin{aligned} v &= K \\ \omega &= -k_1.v.e.\frac{sin\tilde{\theta}}{\tilde{\theta}} - k_2.|v|.\tilde{\theta} + \frac{c(s)vcos\tilde{\theta}}{1-ec(s)} \end{aligned} \quad (2)$$

where K, k_1 and k_2 are positive constants. Its candidate Lyapunov function $V_{TT} = k_1.\frac{e^2}{2} + \frac{\tilde{\theta}^2}{2}$ has a decreasing time derivative. This controller asymptotically stabilizes $(e = 0, \tilde{\theta} = 0)$ provided that the robot is not on the singular point $e = \frac{1}{c(s)}$. Full demonstration is available in [4].
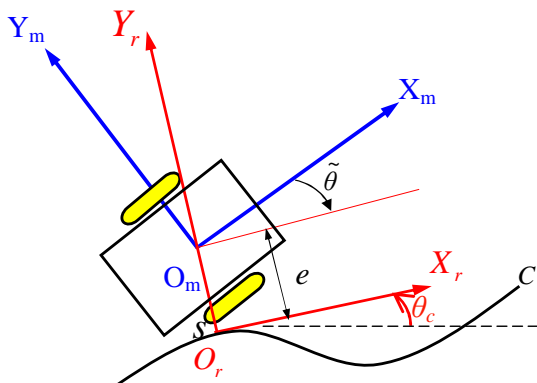

Fig. 1. The mobile robot on the Frenet frame.

C. Obstacle avoidance controller

Limit cycle navigation method allows deciding in which direction and how far the robot avoids the obstacle. The limit cycle is considered to be the circle characterized by R_i radius which is the radius of the hindering obstacle plus a safe margin. In order to focus the attention only on the proposed architecture of control, accurate details about this method are available in [9].

III. THE PROPOSED CONTROL ARCHITECTURE

The proposed hybrid architecture is applied to a mobile robot which uses basic perceptual and decisional capabilities. Therefore, according to the robot's sensors information, decision to apply the convenient controller is made. The robot has to track trajectory while avoiding obstacles. Consequently, the architecture contains at least two controllers: trajectory tracking and obstacle avoidance. However, hard switches between these one may lead to instability even if each controller is individually stable. Therefore, more restrictions are needed to control these switches. Multiple Lyapunov function theorem [2] gives sufficient theoretic conditions to guarantee stability of an overall hybrid system. The proposed control architecture is based on it.

A. Multiple Lyapunov function theorem

Multiple Lyapunov function theorem (MLF)
Given N dynamical subsystems $\sigma_1, \sigma_2,...,\sigma_N$, each with an equilibrium point at the origin, and N candidate Lyapunov functions, $V_1, V_2,...,V_N$. For each subsystem σ_i, let $t_1, t_2, ..., t_m, ..., t_k$ be the switching moments in this subsystem (only one subsystem is active at a time).
***If** V_i decreases when σ_i is active and*
$V_i(t_m) \leq V_i(t_{m-1})$
***Then** the hybrid system is Lyapunov stable.*

The theorem is illustrated (cf. Figure 2) for a simple subsystem σ_i. Thus, when σ_i is active (phases I and III), its Lyapunov function decreases. When the control switches to another subsystem (phases II and IV), V_{σ_i} may increase. However, to insure the global stability, this subsystem (and so as for all the other subsystems) must be reactivated only if its Lyapunov function takes a smaller value than the last time the system switches in.
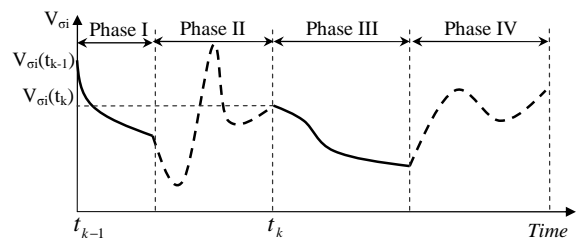

Fig. 2. Variation of the Lyapunov function for the subsystem σ_i. Solid lines indicate that σ_i is active, dashed inactive.

In this paper, we prove that it is possible to use this theorem to guarantee stability of the robot control during all its navigation. Indeed, since the robot is controlled by a set of simple controllers whose stability is proved by the classical Lyapunov theorem, only the second condition of the above theorem needs to be satisfied.

If we assume that the mobile robot is able to reach its trajectory after achieving avoidance of each obstacle (which is not a heavy assumption since overlapped obstacles can be seen as one obstacle), the switching steps (Trajectory tracking → Obstacle avoidance →Trajectory tracking) form a new cycle for each obstacle. Then, we have to satisfy the second condition of MLF theorem for the trajectory tracking controller only since it is the only controller which appears two times in a cycle. Our idea is to use a specific "Go-to-goal" controller. This one is activated once the obstacle is avoided. Its role is to lead the robot on the reference trajectory until the MLF condition for the trajectory tracking becomes true. The proposed automaton is given (cf. Figure 3). $D_{robot-obstacle}$ is the distance between the robot and the obstacle, while R_i is defined in section (II-C). It is obvious that undesirable effects due to fast switches between trajectory tracking and obstacle avoidance controller are removed. Indeed, when switching from obstacle avoidance in trajectory tracking, control has to go through go-to-goal.

In practice, it is not possible to satisfy the MLF condition all the time. For example, when the robot is already on the reference trajectory, $V_{TT} \to 0$. When it meets an obstacle, V_{TT} naturally increases since the robot leaves the reference trajectory to avoid the obstacle. Go-to-goal can lead the robot on the reference trajectory again but its convergence is guaranteed only when time tends to infinity. In this paper, the go-to-goal controller designed by Toibero [12] is used. Here, we prove that this controller converges in a finite time, which is more interesting for practical application. Moreover, we will release the second constraint of MLF theorem in a
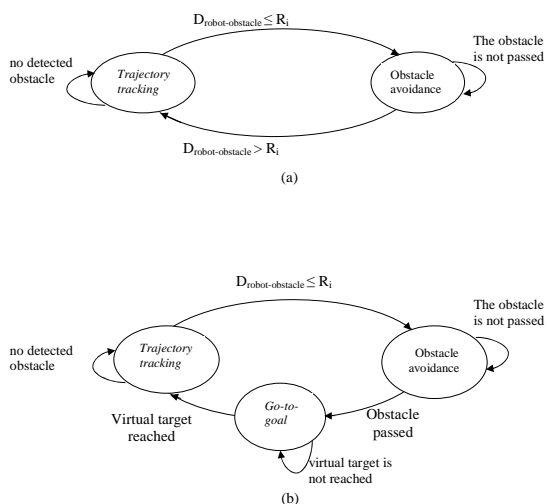
suitable way such that we allow the control to switch in trajectory tracking for the kth time, when $V_{TT}(t_{k-1}) \approx 0$.

B. Go to goal controller

The task to accomplish with this controller is to reach a desired target. Here, $(d, \tilde{\theta}_R)$ tend to $(0,0)$, where d is the robot-target distance and $\tilde{\theta}_R$ is the robot orientation in the relative robot frame (cf. Figure 4). control inputs are expressed as follows [12]

$$v = \frac{v_{max}}{1+|d|}d\cos(\tilde{\theta}_R)$$
$$\omega = \frac{v_{max}}{1+|d|}\cos(\tilde{\theta}_R)\sin(\tilde{\theta}_R) + k_2 tanh(k_3\tilde{\theta}_R) \quad (3)$$

where v and ω are linear and angular velocities respectively.

Its time derivative $\dot{V}_{GG}$ is negative for all $(d, \tilde{\theta}_R) \neq (0, 0)$. This controller was shown globally asymptotically stable using the Lyapunov function $V_{GG} = \frac{\tilde{\theta}_R^2}{2} + \frac{d^2}{2}$ [12]. Note that it is possible to correct the final orientation of the robot when this one arrives to its goal simply by using the particular cas of this control law with $v_{max} = 0$ (the robot corrects its orientation without linear velocity) and the new control law becomes

$$v = 0, \omega = k_2 tanh(k_3\tilde{\theta}_R) \quad (4)$$

C. Proof of convergence according to MLF theorem

Since it is assumed that after achieving each obstacle avoidance, the robot is able to reach its reference trajectory before eventually meeting an other obstacle, we have to prove that this task is achieved in a stable manner and in a finite time. It means that it is important to prove that switching from obstacle avoidance to trajectory tracking through go-to-goal controller arrives in a finite time. Trajectory tracking controller is asymptotically stable. By noting $x_{TT} = (e, \tilde{\theta})$ (cf. Figure 1) and $\|x_{TT}\|$ the Euclidian norm of x_{TT}, and by definition of asymptotic stability

$$\exists \delta_{TT} > 0 : \|x_{TT}(0)\| < \delta_1 \Rightarrow \|x_{TT}(t)\| \overset{t \to \infty}{\to} 0$$

If an obstacle is met, and once avoided, Go-to-goal controller leads the robot to a point P_t on the reference trajectory

(a)

(b)

Fig. 3. (a) Hard switches between controllers. (b)The proposed regularized automaton with go-to-goal controller.
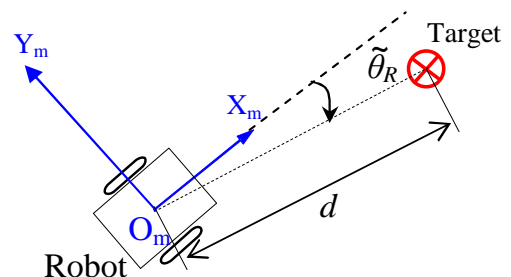

Fig. 4. Go-to-goal controller.

(cf. Figure 5) until MLF condition is satisfied. P_t is defined as the intersection of the trajectory with a virtual circle of radius $(R_i + \epsilon)$ where R_i is the circle of influence of the obstacle. The latter is considered as avoided when the robot is on the point P_e. P_e is the intersection of the influence circle with its tangent going through P_t (chosen according to the avoidance direction) (cf. Figure 5).

Go-to-goal is globally asymptotically stable (cf. Subsection III-B). By noting $x_2 = (d, \tilde{\theta}_R)$ (cf. Figure 4), we have

$$\forall \delta_{GG} > 0 : \|x_{GG}(0)\| < \delta_{GG} \Rightarrow \|x_{GG}(t)\| \overset{t \to \infty}{\to} 0$$

Note that asymptotic stability cited above is insured when $t \to \infty$. In practice, if the proposed architecture seems allowing the robot to achieve obstacle avoidance and switching in a stable manner, we have to prove that it is also done in a finite time. In general, switching in trajectory tracking controller for the kth time t_k occurs when $V_{TT}(t_k) \le V_{TT}(t_{k-1})$.

However, the worst case is when $V_{TT}(t_{k-1})$ is already near to 0 ($V_{TT}(t_{k-1}) \to 0$): How to decrease V_{TT} again to have $V_{TT}(t_k) \le V_{TT}(t_{k-1})$? In this case, it comes to lead up the robot until $\|x_{TT}\| \le \delta_{TT}$. Hence, the robot is in the convergence area of the trajectory tracking controller. Switching can then occur and stability is insured. Moreover, note that $\|x_{TT}(t_k)\| \le V_{TT}$ (for every $k_1 \ge 1$ (cf. Subsection II-B), and the switch condition is $\|x_{TT}(t_k)\| \le \delta$, the constraint $V_{TT}(t_k) \le V_{TT}(t_{k-1})$ can then be released. This case takes the maximum of convergence time since $\|x_{TT}\|$ have to decrease until $\|x_{TT}\| \le \delta_{TT}$ whereas the general case is $\|x_{TT}\| \le V_{TT}(t_k) \le V_{TT}(t_{k-1})$. We will thus study this case proving that this time is still finite.

We saw that trajectory tracking controller is asymptotically stable provided that the lateral error satisfies $e < \frac{1}{c(s)}$. Since $\|x_{TT}\| = (e, \tilde{\theta})$, we can define the convergence area of trajectory tracking controller as $\delta_{TT} = (e_\delta, \tilde{\theta}_\delta)$ where $e_\delta = inf(\frac{1}{c(s)}), \tilde{\theta}_\delta \approx 0$. $inf(\frac{1}{c(s)})$ is the smallest curvature radius of the reference trajectory. We have then to prove that Go-to-goal controller allows to arrive in this area in a finite time.

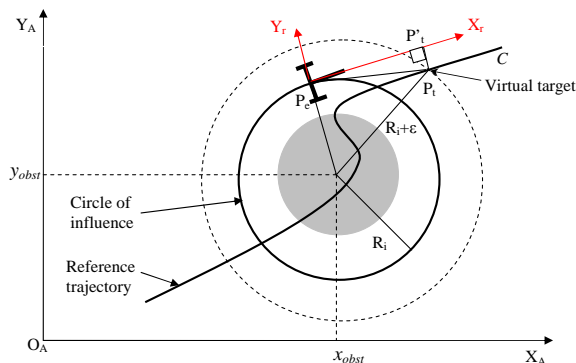According to (cf. Figure 4) we get the simple equation of

Fig. 5. Virtual target to reach before reactivating trajectory tracking controller.

the derivative $\dot{\tilde{\theta}}_R$

$$\dot{\tilde{\theta}}_R = v \frac{sin(\tilde{\theta}_R)}{d} - \omega \tag{5}$$

Since the objective is to lead the robot to δ_1, $d \approx 0$ but $d \ne 0$. $\dot{\tilde{\theta}}_R$ is then defined.

By replacing v and ω from (3) in (5) we get:

$$\dot{\tilde{\theta}}_R = -k_2 tanh(k_3 \tilde{\theta}_R) \tag{6}$$

It can be noticed from (6) that the variation of $\tilde{\theta}_R$ is independent from d. Thus, if time of convergence of the Go-to-goal controller is noted T_f, its maximum is obtained if first $\tilde{\theta}_R$ decreases , and only when $\tilde{\theta}_R$ converges, the distance robot-target d starts decreasing. It can be concluded that

$$T_f \le (\Delta t_1 + \Delta t_2)$$

where Δt_1 is the convergence time of $\tilde{\theta}$ to 0 (it is the time of convergence of $\tilde{\theta}_R$ to 0 plus the time of convergence of $\tilde{\theta}$ to 0 knowing $\tilde{\theta}_R = 0$) and Δt_2 is the convergence time of d (it will be shown later that it means decreasing of e to e_δ).

Hence, if Δt_1 and Δt_2 are finite, T_f is finite, too.

1) Δt_1 is finite: The angle $\tilde{\theta}_R$ is such that $\tilde{\theta}_R \in]-\pi, \pi]$. We decompose the interval into two parts: $\tilde{\theta}_R \ge 0$ and $\tilde{\theta}_R \le 0$. Let us study the case $\tilde{\theta}_R \ge 0$:

from (6) we get: $\dot{\tilde{\theta}}_R \le 0$. It means that $\tilde{\theta}_R$ is decreasing.

It can be noticed that $max(\Delta t_1)$ is when $\tilde{\theta}_R \to \pi$ (since $max(\tilde{\theta}_R) = \pi$).

An important property of $tanh(k\tilde{\theta}_R)$ is that $tanh(k\tilde{\theta}_R) \to 1$ for every angle $\tilde{\theta}_R >> 0$ and with a well chosen k. Let us consider that $\tilde{\theta}_R >> 0$ if $\tilde{\theta}_R > 10°$. Thus, by replacing in (6) we get $\dot{\tilde{\theta}}_R \approx -k_2(1 - \epsilon_1)$ where $\epsilon_1 << 1$.

A simple integration gives

$$\delta t_1 = \frac{\tilde{\theta}_R(t) - \tilde{\theta}_R(t_0)}{-k_2(1 - \epsilon_1)}$$

After a time δt_1, $\tilde{\theta}_R$ becomes small, we can therefore make the assumption

$$tanh(k_3 \tilde{\theta}_R) = k_3 \tilde{\theta}_R$$

thanks to a Taylor development of the function $tanh$ of order 1. when we replace in (6) we get:

$$\tilde{\theta}_R(t_2) = \tilde{\theta}_R(t_1) e^{-k_2 k_3 (t_2 - t_1)}$$

It means that $\tilde{\theta}_R$ is exponentially decreasing. In addition, $\tilde{\theta}_R(t_1)$ is already small. Consequently it rapidly decreases and we can make the approximation that: $\Delta t_1 = \delta t_1 + \delta$ where δ is time of convergence of $\tilde{\theta}$ to 0 knowing $\tilde{\theta}_R = 0$. In other words, when the robot arrives to the virtual target, it has to correct its final orientation to have $\tilde{\theta} \le \tilde{\theta}_\delta$. This is accomplished thanks to the controller (cf. Equation 4) when the linear velocity $v = 0$ as cited. Details will not be given since it is done in the same way as for $\tilde{\theta}_R$. Demonstration is also given only for one interval. For the second interval ($\tilde{\theta}_R \le 0$), it is done exactly in the same way.

2) Δt_2 is finite: The variation of d can be written

$$\dot{d} = -v\cos(\tilde{\theta}_R) = -\frac{v_{max}}{1+d}d\cos^2(\tilde{\theta}_R) \qquad (7)$$

After Δt_1, $\tilde{\theta}_R$ is sufficiently small to write $\cos^2(\tilde{\theta}_R) = 1 - \epsilon_2$ ($\epsilon_2 << 1$).

By replacing in (7), we get

$$\dot{d} = -\frac{v_{max}}{1+d}d(1-\epsilon_2)$$

where ($\dot{d} = \frac{dd}{dt}$)

The solution of this differential equation gives

$$\Delta t_2 = \frac{\ln(d(t)) + d(t) - \ln(d(t_0)) - d(t_0)}{-v_{max}(1-\epsilon_2)} \qquad (8)$$

Note also that the relation between the lateral error e and the distance d can be easily deduced (cf. Figure 5). Indeed, when switching in trajectory tracking, lateral error according to P_t (cf. Figure 5) is

$$e = d\sin(\phi)$$

where $\phi = P_t\hat{P}_eP_t'$, where P_t' is the projection of P_t on the X_r axis (of the Frenet frame).

The worst case to switch in trajectory tracking for the kth time t_k is when $V_{TT}(t_{k-1}) \approx 0$ and the curvature radius in P_t is $inf(\frac{1}{c(s)})$. $\max(\Delta t_2)$ can then be calculated as:

$$\Delta t_2 = \frac{\ln(\frac{inf(\frac{1}{c(s)})}{\sin(\Phi)}) + \frac{inf(\frac{1}{c(s)})}{\sin(\Phi)} - \ln(d(\Delta t_1)) - d(\Delta t_1)}{-v_{max}(1-\epsilon_2)}$$

Note that $c(s) \neq 0$ (otherwise, the curvature radius $inf(\frac{1}{c(s)}) = \infty$ which means in practice that there is no singularity, and switching in trajectory tracking can occur everywhere). Δt_2 is then finite too. Since we saw above that T_f, the whole time of convergence is $T_f < \Delta t_1 + \Delta t_2$ T_f is then finite, too. In next section, numerical simulation confirms the stability of the proposed control architecture.

IV. SIMULATION RESULTS

To estimate the relevance of the proposed hybrid architecture, we compare it with results given by hard switches between the two controllers: trajectory tracking and obstacle avoidance. Results are shown in (Fig. 6). We can notice that when detecting the obstacle, the robot switches effectively in obstacle avoidance. However, when the obstacle is passed, the robot switches in trajectory tracking controller and oscillations are observed. Indeed, the robot (which is now controlled by trajectory tracking command) falls in the circle of influence again and tries to avoid it (switch in obstacle avoidance again). Variations of Lyapunov functions are illustrated in (Fig. 7) with switching control indicator. The latter indicates which controller is active (its values are 1 and 2 for trajectory tracking and obstacle avoidance respectively). We can notice that undesirable switches are occurring and MLF condition is not satisfied. Indeed, unless geometrical constraints (distance of the robot to the obstacle), there is no rules managing these switches. Moreover, the Lyapunov
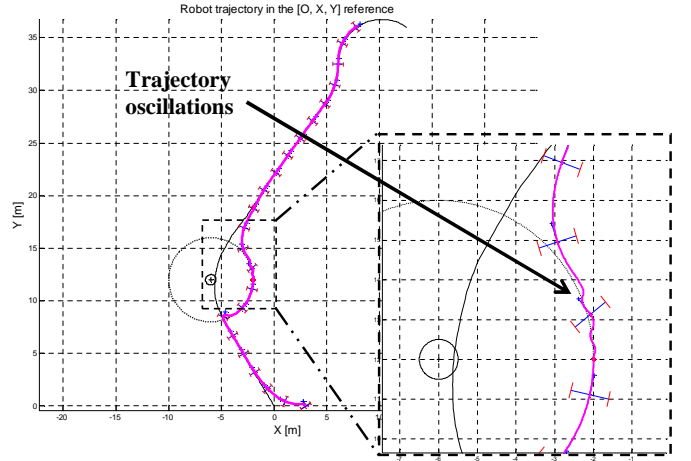

Fig. 6. Real robot's path controlled by hard switches. The surrounded area is enlarged on the right.
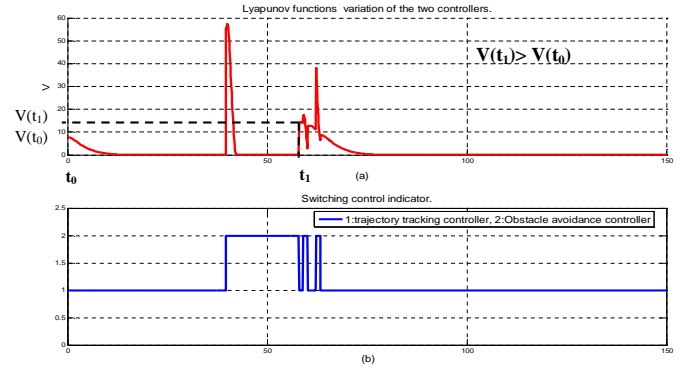

Fig. 7. Lyapunov function variations of the two controllers.

function V_{TT} is decreasing every time the controller is active, however the second MLF condition is not satisfied.

On the other side, our architecture of control (cf. Section III) allows avoiding these useless switches thanks to go-to-goal controller. The latter prevents hard switches from occurring before that the obstacle is passed. The real path followed by the robot is shown in (Fig. 8) where no oscillation is observed. The robot has to avoid two obstacles. According to their positions, It avoids the first counter clockwise while the second is avoided in a clockwise direction. Switch moments (cf. Figure 9) with Lyapunov functions progress of the active controller are given in (a). It can be noticed that the Lyapunov function of each controller is decreasing when this one is active (first condition of MLF). Lyapunov function of the trajectory tracking controller is given in (b). Switch moments are illustrated. It is noticed that switch occurs at the moments t_2 and t_1 where $V_{TT}(t_2) < V_{TT}(t_1) < V_{TT}(t_0 = 0)$, which is due to the second condition of MLF. Note that in (c), the value 0 of the switching control indicator refers to the added controller Go-to-goal. Finally, lateral and angular errors are given in (d) and (e) respectively: It is shown that $T_f << \Delta t_1 + \Delta t_2$. However, note that due to the MLF condition,
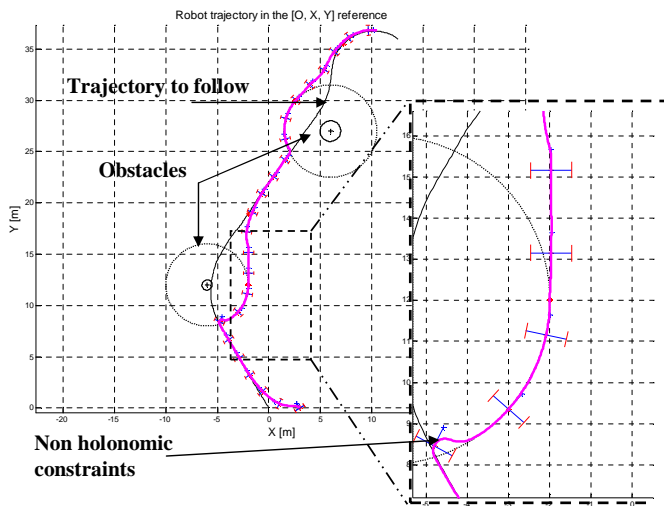
Fig. 8. Real robot path when controlled with the proposed control architecture. The surrounded area is enlarged on the right.

there was no need to maintain Go-to-goal controller until $e < e_\delta$ and $\tilde{\theta} \approx 0$.

V. CONCLUSIONS AND FUTURE WORKS

We applied hybrid control architecture on a mobile robot to obtain a stable trajectory tracking while avoiding obstacles. Indeed, the robot is controlled by elementary continuous controllers according to the sub-tasks to accomplish (trajectory tracking, obstacle avoidance) and switching from a controller to an other is done referring to discrete events. We saw that hard switches are not efficient to insure global stability. Therefore, we propose to design a stable hybrid control architecture. In addition to elementary stable controllers for the two main sub-tasks, we introduce a third controller which overcomes different constraints (second condition of the multiple Lyapunov function theorem, singularities of trajectory tracking controller). Simulations show that our architecture prevents useless switches, guaranteeing thus a suitable navigation for the robot. Application to multi-robot systems navigating in formation will be done. The objective is to make each robot able to avoid an obstacle before regaining the formation.

REFERENCES

[1] L. Adouane. An adaptive multi-controller architecture for the navigation of mobile robot. In *10th IAS, Intelligent Autonomous Systems*, pages 342–347, Baden-Baden, Germany, July 2008.

[2] M. S. Branicky. Stability of switched and hybrid systems. In *33rd IEEE Conference on Decision Control*, pages 3498–3503, 1993.

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

[4] C. Canudas de Wit, B. Siciliano, and G. Bastin. *Theory of Robot Control*. Springer Verlag Edition, 1996.

[5] B. D Andrea-Novel, G. Campion, and G. Bastin. Control of nonholonomic wheeled mobile robots by state feedback linearization. *International Journal of Robotics Research*, 14(6):543559, 1995.

[6] M. Egerstedt, K. H. Johansson, J. Lygeros, and S. Sastry. Behavior based robotics using regularized hybrid automata. *Computer Science ISSN 0302-9743*, 1790:103–116, 2000.
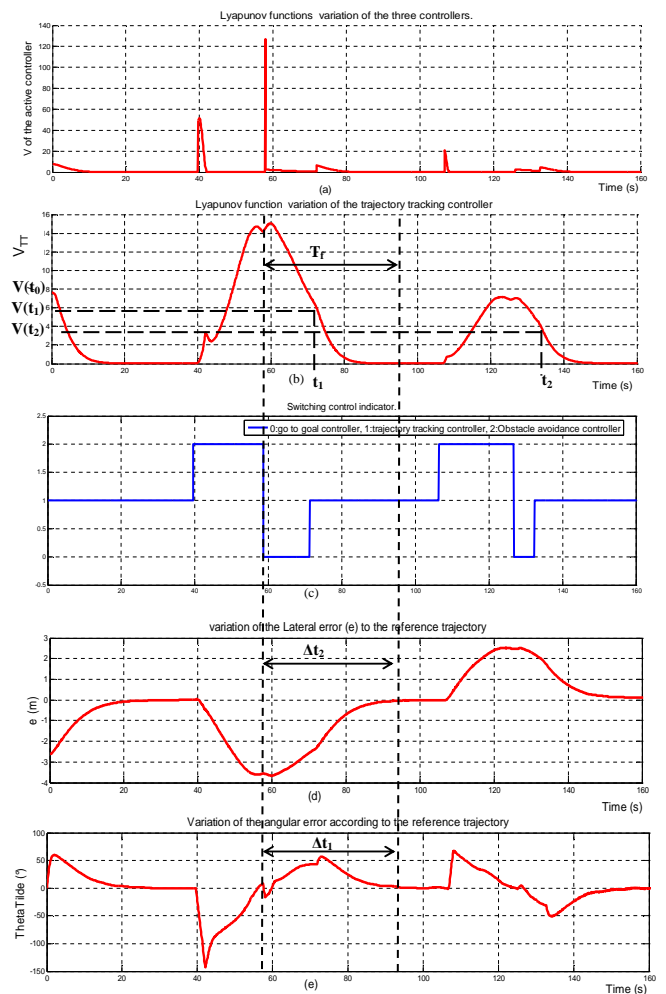
[7] K. H. Johansson, M. Egerstedt, J. Lygeros, and Sastry S. On the regularization of zeno hybrid automata. *Systems & Control Letters*, 38:141–150, 1999.

[8] O. Khatib. Real time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5:90–99, 1986.

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

[10] 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):64–77, 1995.

[11] M.G. Slack. Navigation template: mediating qualitative guidance and quantitative control in mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, 23(2):452466, 1993.

[12] J.M. Toibero, R. Carelli, and B. Kuchen. Switching control of mobile robot for autonomous navigation in unkown environments. *IEEE International Conference on Robotics and Automation*, pages 1974–1979, 2007.

[13] R. Zapata and P. Lepinay. Reactive behaviors of fast mobile robots. *Journal of Robotics Systems*, 11(1):13–20, 1994.

Fig. 9. Lyapunov variation functions of the proposed control architecture.