Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Mobile Robot Navigation in Cluttered Environment using Reactive Elliptic Trajectories

Lounis Adouane, Ahmed Benzerrouk and Philippe Martinet

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

Abstract: Reactive navigation in very cluttered environment while insuring maximum safety and task efficiency is a challenging subject. This paper proposes online and adaptive elliptic trajectories to perform smooth and safe mobile robot navigation. These trajectories use limit-cycle principle already applied in the literature but with the difference that the applied limit-cycles are now elliptic (not circular) and are more generic and flexible to perform navigation in environments with different kinds of obstacles shape. The set points given to the robot are generated while following reactive obstacle avoidance algorithm embedded in a multi-controller architecture (*Obstacle avoidance* and *Attraction to the target* controllers). This algorithm uses specific reference frame which gives accurate indication of robot situation. The robot knows thus if it must avoid the obstacle in clockwise or counter-clockwise direction and prevent robot from local minima, dead ends and oscillations. The stability of the proposed bottom-up control architecture is proved according to Lyapunov synthesis. Simulations and experiments in different environments are performed to demonstrate the efficiency and the reliability of the proposed control architecture.

Keywords: Mobile robot navigation, Multi-controller architecture, Reactive control, Obstacle avoidance, Elliptic limit-cycles, Lyapunov synthesis.

1. INTRODUCTION

An important issue for successful mobile robot navigation is obstacle avoidance. In fact, this function permits to prevent robot collision and insure thus robot safety. One part of the literature considers that the robot control is entirely based on path planning methods while involving the total knowledge of its environment. Voronoi diagrams and visibility graphs Latombe (1991) or Artificial potential fields functions Rimon and Koditschek (Oct. 1992) are among these methods. All obstacles configurations are taken thus into account in the planning step. In these methods, it is possible also to deal with changing environment while regularly replanning the robot's path Fraichard (1999), Jur-Van-Den and Overmars (2005). However, planning and replanning require a significant computational time and complexity.

The other community is more interested by reactive methods where only local sensors information are used rather than a prior knowledge of the environment Egerstedt and Hu (2002), Toibero et al. (2007), Adouane (2009a). In Khatib (1986) the author proposes a real-time obstacle avoidance approach based on the principle of artificial potential fields. He assumes that the robot actions are guided by the sum of attractive and repulsive fields. In Arkin (1989) author extends Khatib's approach while proposing specific schema motors for mobile robots navigation. Another interesting approach, based on a reflex behavior reaction, uses the Deformable Virtual Zone (DVZ) concept, in which a robot movement depends on risk zone surrounds the robot Zapata et al. (2004). If an obstacle is detected, it will deform the DVZ and the approach consists of minimizing this deformation by modifying the control vector. This method deals with any obstacle shape, however, it suffers as schema motors from local minima problem. In general, reactive methods do not require high computational complexities since robot's actions must be given in real-time according to the perception Arkin (1998).

The proposed paper deals with this last community, where the control is reactive. An obstacle avoidance algorithm is proposed and uses orbital trajectories described by limit-cycle differential equations Khalil (2002). These methods have already been used for obstacle avoidance with mobile robots Kim and Kim (2003), Jie et al. (2006) or in Adouane (2009b). Unlike potential field methods Khatib (1986), only disturbing obstacles affect the robot trajectory and direction. However, in existing works, only circular orbits were studied. In this paper, elliptical trajectories are used and stability proofs of the controllers and set points are given using Lyapunov functions. More generic and efficient obstacle avoidance is thus performed and this even with specific obstacle shapes, for instance long walls. In fact, elliptic trajectories fit better this kind of obstacles than cylindrical one.

The rest of the paper is organized as follows. Section 2 gives the specification of the task to achieve. The details of the proposed control architecture are given in section 3. It presents the model of the considered robot and the implemented elementary controllers laws. Section 4 gives in details the proposed obstacle avoidance algorithm whereas section 5 is devoted to the description and analysis of simulation and experiment results. This paper ends with some conclusions and further work.

2. MOBILE ROBOT NAVIGATION IN CLUTTERED ENVIRONMENT

The objective of this task is to lead a mobile robot towards a specific target in an unstructured environment. This task must be doing while avoiding statical and dynamical obstacles which

can have different shapes. One supposes in the setup that robot and obstacles are surrounded by respectively cylindrical and elliptical boxes. The cylindrical box (the robot) is characterized by R_R radius and elliptical boxes (obstacles) are given by (cf. equation 1 and figure 1):

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

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 1). When $a = b$ equation 1 becomes a circle equation (Ω will do not gives thus any more information).

The target to reach is also characterized by a circle of R_T radius. Several perceptions are also necessary for the proposed robot navigation (cf. Figure 1):

- D_{ROi} minimal distance between the robot and the obstacle "i" Eberly (2008),
- Detect constrained obstacles, i.e., in our case it is enough to know if it exists an intersect points between the line "l" and the *Ellipse of influence* (cf. Figure 1). 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 $A_{lc} = A + R_R + Margin$,
  - The value of its minor axis is $2B_{lc}$ with $B_{lc} = B + R_R + Margin$.

  Where *Margin* corresponds to a safety tolerance which includes: perception incertitude, control reliability and accuracy, etc.

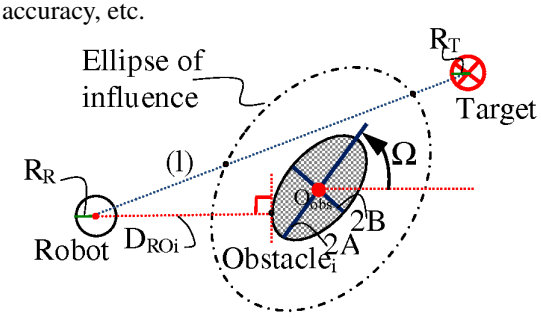


Fig. 1. The used perceptions for mobile robot navigation

The choice of ellipse box rather than circle as used in Kim and Kim (2003), Jie et al. (2006) or Adouane (2009b) is to have one more generic and flexible mean to surround and fit accurately different kind of obstacles shapes. Among examples of shapes which can be appropriately fitted by an ellipse and less by a circle is a wall (or in general, longitudinal shapes). Figure 2 shows this kind of configuration. In fact if we would like to surround this wall by an appropriate circle, this one will have a large radius which will induces more robot path distance to avoid safely the obstacle Kim and Kim (2003) (cf. Figure 2(a)). Figure 2(b) shows that the ellipse fits better the dimension of the obstacle. This figure shows also uncertain perceptions taken by infrared sensor in one side of the wall (left side). The surrounded ellipse parameters (h, k, A, B and Ω) (cf. equation 1 and figure 1) are obtained while using weighted least square (WLS) method Gander et al. (1994). More the distance between the robot and the measured point is large less is the weight of this measure to obtain the final estimated parameters. It is to note also that the used WLS method needs at less 5 measured points to work. A navigation in cluttered environment using this parameter estimation is given in Figure 7.

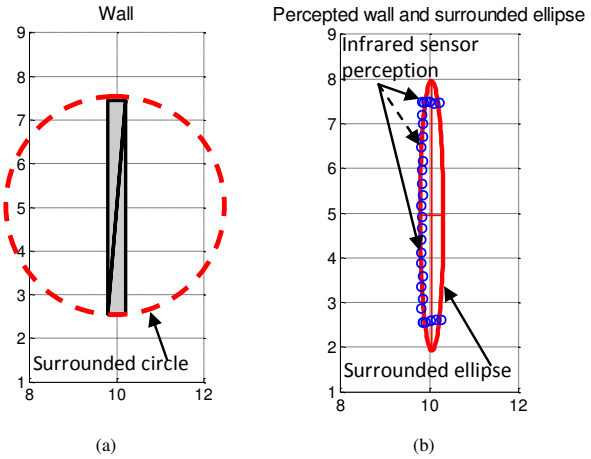


Fig. 2. Interpolated wall using a circle and ellipse shapes

3. CONTROL ARCHITECTURE

The proposed structure of control (cf. Figure 3) aims to manage the interactions between elementary controllers while guaranteing the stability of the overall control as proposed in Adouane (2009a). Its objective is also to obtain safe, smooth and fast robot navigation. It will permit for example to an autonomous application of travelers transportation Pradalier et al. (2005) to have more comfortable displacements of the passengers. The specific blocks composing this control are detailed below.

3.1 Hierarchical action selection

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 Brooks (1986) or Adouane and Le Fort-Piat (2006).

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 (cf. Algorithm 1). This permits to anticipate the activation of obstacle avoidance controller unlike what is proposed in Huang et al. (2006) or Zhang et al. (2006), which wait until the robot is in the

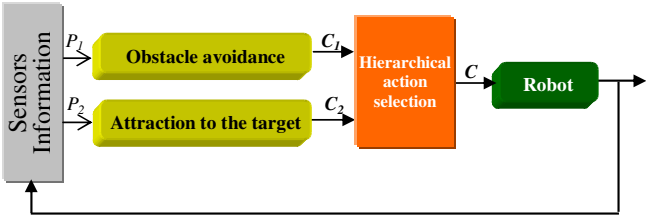


Fig. 3. Control architecture for mobile robot navigation

immediate vicinity of the obstacle (i.e. $D_{ROi} \leq R$ "a certain radius value"). Algorithm 1 permits thus to decrease the time to reach the target, especially in very cluttered environments (cf. Section 5).

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 1)* } **then**
  Activate *Obstacle avoidance* controller
else
  Activate *Attraction to the target* controller
end

Algorithm 1: Hierarchical action selection

3.2 Elementary controllers

Each controller composing the control architecture (cf. Figure 3) is characterized by a stable nominal law. These laws are synthesized according to Lyapunov theorem. We will present here only some details about the stability demonstration of the used laws. More details are given in Adouane (2009a). Before describing each elementary controller, let's show the used kinematic robot model (cf. Figure 4):

$$\dot{\xi} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & -l_2\cos\theta - l_1\sin\theta \\ \sin\theta & -l_2\sin\theta + l_1\cos\theta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \quad (2)$$

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".

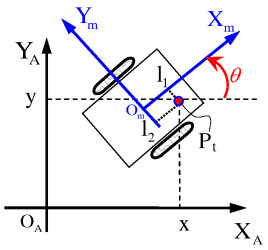


Fig. 4. Robot configuration in a cartesian reference frame

Attraction to the target controller This controller guides the robot toward the target which is represented by a circle of (x_T, y_T) center and of R_T radius (cf. Figure 1). The used control law is a control of position at the point P_t = (l_1, 0) (cf. Figure 4). As we consider a circular target with R_T radius, therefore, to guarantee that the center of robot axis reaches the target with asymptotical convergence, l_1 must be $\leq R_T$ (cf. Figure 4).

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos\theta & -l_1\sin\theta \\ \sin\theta & l_1\cos\theta \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = M \begin{pmatrix} v \\ w \end{pmatrix} \quad (3)$$

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 6) $\Rightarrow \begin{cases} \dot{e}_x = \dot{x} \\ \dot{e}_y = \dot{y} \end{cases}$

Classical techniques of linear system stabilization can be used to asymptotically stabilize the error to zero Laumond (2001) [Chapter 2]. We use a simple proportional controller which is given by:

$$\begin{pmatrix} v \\ w \end{pmatrix} = -KM^{-1} \begin{pmatrix} e_x \\ e_y \end{pmatrix} \quad (4)$$

with $K > 0$. Let's consider the following Lyapunov function

$$V_1 = \tfrac{1}{2}d^2 \quad (5)$$

with $d = \sqrt{e_x^2 + e_y^2}$ (distance robot-target).

Therefore, to guarantee the asymptotical stability of the proposed controller, $\dot{V}_1$ must be strictly negative definite, so, $d\dot{d} < 0$, what is easily proven as long as $d \neq 0$.

Obstacle avoidance controller To perform the obstacle avoidance behavior, the robot needs to fellow accurately limit-cycle trajectories as what is given in Khalil (2002), Kim and Kim (2003), Jie et al. (2006) or Adouane (2009b). In these works authors use a circular limit-cycle characterized by a circle of influence of R_I radius. In this paper, we propose to extend this methodology for more flexible limit-cycle shape (an ellipse) (cf. Section 2). In fact, this shape is the generalization of circle shape, it is enough to choose $a = b$ in equation 1 to obtain a circle equation.

The differential equations giving elliptic limit-cycles are:

- For the clockwise trajectory motion (cf. Figure 5(a)):

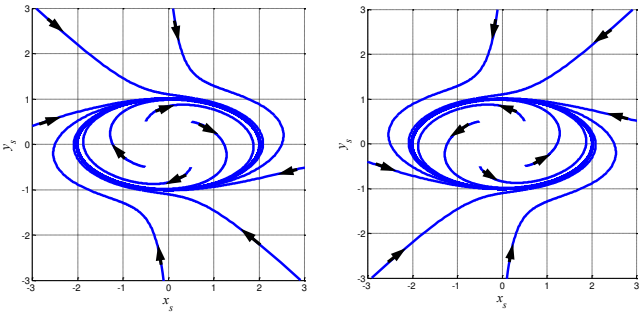
$$\begin{aligned} \dot{x}_s &= y_s + \mu x_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_sy_s) \\ \dot{y}_s &= -x_s + \mu y_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_sy_s) \end{aligned} \quad (6)$$

- For the counter-clockwise trajectory motion (cf. Figure 5(b)):

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

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 1); c if $\neq 0$ gives the Ω ellipse angle (cf. Section 2) and μ is a positive constant value.

Figure 5 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 Khalil (2002). Figure 5(a) and 5(b) show the shape of equations (6) and (7) respectively. They show the direction of trajectories (clockwise or counter-clockwise) according to (x_s, y_s) axis. The trajectories from all



(a) Clockwise (b) Counter-Clockwise

Fig. 5. Shape possibilities for the used elliptic limit-cycles

points (x_s, y_s) of X, Y reference frame, including inside the ellipse, move towards the ellipse. This can be demonstrated mathematically while taken the following lyapunov function:

$$V(x) = (1/2)(x_s^2 + y_s^2) \tag{8}$$

The derivative of $V(x)$ along the trajectories of the system is given by $\dot{V}(x) = x_s\dot{x}_s + y_s\dot{y}_s$ and after some simplifications we obtain finally:

$$\dot{V}(x) = \mu V(x)(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_sy_s) \tag{9}$$

The derivative of $\dot{V}(x)$ is:

- negative if $(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_sy_s) < 0$, thus if the initial condition (x_{s0}, y_{s0}) is inside the ellipse (given by equation $x_s^2/A_{lc}^2 + y_s^2/B_{lc}^2 + cx_sy_s = 1$),
- and is positive if the initial condition is outside the ellipse.

Therefore, the ellipse given by $x_s^2/A_{lc}^2 + y_s^2/B_{lc}^2 + cx_sy_s = 1$ is a periodic orbit and is called the elliptic limit-cycle. We can also say that, more the value of μ is large, more is the velocity of the limit-cycle to converge toward its periodical orbit (the opposite is true).

To follow with reactive way the set points given by the elliptic limit cycles, we use 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 4). The desired robot orientation θ_d is given thus by the differential equation of the limit-cycle (6) or (7) as:

$$\theta_d = \arctan(\frac{\dot{y}_s}{\dot{x}_s}) \tag{10}$$

and the error by

$$\theta_e = \theta_d - \theta \tag{11}$$

We control the robot to move to the desired orientation by using this standard control law:

$$w = \dot{\theta}_d + K_p\theta_e \tag{12}$$

with K_p a constant ¿ 0, $\dot{\theta}_e$ is given then by:

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

Let's consider the following Lyapunov function

$$V_2 = \tfrac{1}{2}\theta_e^2 \tag{14}$$

$\dot{V}_2$ is equal then to $\theta_e\dot{\theta}_e = -K_p\theta_e^2$ which is always strictly negative (so, asymptotically stable). It is to note that the nominal velocity of the robot v when this controller is active is a constant.

4. REACTIVE OBSTACLE AVOIDANCE ALGORITHM

In what follows, the overall methodology to achieve the proposed obstacle avoidance algorithm will be given. The algorithm is developed according to stimuli-response principle. To implement this kind of behavior it is important to:

- Detect the obstacle to avoid (cf. Section 2),
- Give the direction of the avoidance (clockwise or counter-clockwise),
- Define an escape criterion which defines if the obstacle is completely avoided or not yet.

All these different steps must be followed and applied while guaranteeing that: the robot trajectory is safe, smooth and avoid undesirable situations as deadlocks or local minima ; and that the stability of the applied control law is guaranteed. The necessary steps to carry out the obstacle avoidance algorithm (2) are given below:

(1) For each sample time, obtain the distance D_{ROi} for each potentially disturbing obstacle "i" (cf. Figure 1),
(2) Among the set of disturbing obstacles (which can constrain the robot to reach the target), choose the closer to the robot (the smallest D_{ROi} (cf. Figure 1)). This specific obstacle has the following features: (x_{obst}, y_{obst}) center position and $2A$ as major axis and $2B$ as minor axis,
(3) After the determination of the closest constrained obstacle, we need to obtain four specific areas (cf. Figure 6) which give the robot behavior: clockwise or counter-clockwise obstacle avoidance ; repulsive or attractive phase (cf. Algorithm 2). To distinguish between these 4 areas we need to:
   - define a specific reference frame which has the following features (cf. Figure 6):
     · the X_O axis connects the center of the obstacle (x_{obst}, y_{obst}) to the center of the target. This axis is oriented towards the target,
     · the Y_O axis is perpendicular to the X_O axis and it is oriented while following trigonometric convention.
   - apply the reference frame change of the position robot coordinate $(x, y)_A$ (given in absolute reference frame) towards the reference frame linked to the obstacle $(x, y)_O$. The transformation is achieved while using the following homogeneous transformation:

$$\begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_O = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & x_{obst} \\ \sin\alpha & \cos\alpha & 0 & y_{obst} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_A \tag{15}$$

Once all necessary perceptions are obtained, one can apply the proposed reactive obstacle avoidance strategy given by Algorithm 2. To obtain the robot set points, it is necessary to obtain the value of A_{lc} and B_{lc} (cf. Section 2) of the orbital ellipse and the direction "clockwise or counter-clockwise" of the limit-cycle to follow. The position (x_O, y_O) gives the configuration (x, y) of the robot according to obstacle reference frame. The definition of this specific reference frame gives an accurate mean to the robot to know what it must do. In fact, the sign of x_O gives the kind of behavior which must be taken by the robot (attraction or repulsion).

In repulsive phase, the limit-cycle takes an increase value of A'_{lc} and B'_{lc} values to guarantee the trajectory smoothness. The sign of y_O gives the right direction to avoid the obstacle. In fact, if $y_O \geq 0$ then apply clockwise limit-cycle direction else apply counter-clockwise direction. This choice permits to optimize the length of robot trajectory to avoid obstacles. Nevertheless, this direction is forced to the direction taken just before if the obstacle avoidance controller was already active at $(t - \delta T)$ instant and this to avoid local minima and dead-end Adouane (2009b).

The good performance of proposed algorithm 2 need to manage some conflicting situations which are due to local minima or dead ends. These specific local and reactive rules are detailed in Adouane (2009b).

Input: All the features of the closest constrained obstacle ; value of μ
Output: Features of the limit-cycle trajectory to follow
//I) Obtaining the values of A'_{lc} and B'_{lc} of the limit-cycle to follow
if $x_O \leq 0$ **then**
  $\begin{cases} A'_{lc} = A_{lc} - \xi \\ B'_{lc} = B_{lc} - \xi \end{cases}$ (**Attractive phase**)
  {*with ξ a small constant value as $\xi \ll$ Margin which guarantees that the robot do not navigate very closely to the obstacle (cf. Section 2).*}
else
  {*Escape criterion: go out of the obstacle ellipse of influence with smooth way*}
  $\begin{cases} A'_{lc} = A'_{lc} + \xi \\ B'_{lc} = B'_{lc} + \xi \end{cases}$ (**Repulsive phase**)
end
//II) Obtaining the limit-cycle direction
if *obstacle avoidance controller was active at $(t - \delta T)$ instant* **then**
  Apply the same direction already used, equation (6) or (7) is thus applied.
  {*This will permit to avoid oscillations and several conflicting situations Adouane (2009b)*}
else
  {*The limit-cycle set-point is given by:*}
  $\dot{x} = sign(y_O)y + \mu x(1 - x^2/{A'_{lc}}^2 - y^2/{B'_{lc}}^2 - cxy)$
  $\dot{y} = -sign(y_O)x + \mu y(1 - x^2/{A'_{lc}}^2 - y^2/{B'_{lc}}^2 - cxy)$
end

Algorithm 2: Obstacle avoidance algorithm

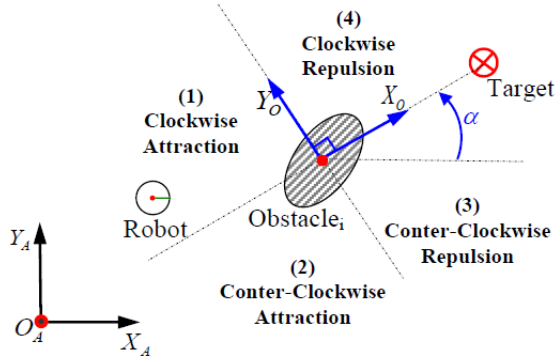


Fig. 6. The 4 specific areas surrounding the obstacle to avoid

5. SIMULATIONS AND EXPERIMENTAL RESULTS

To demonstrate the efficiency of the proposed obstacle avoidance algorithm, a statistical survey was made while doing a large number of simulations in different cluttered and unstructured environments (cf. figure 7 for an example of simulation). We did specifically 1000 simulations with every time, 40 obstacles with different random positions in the environment. It is to note that each surrounded ellipse (obstacle) is submitted to parameters uncertainty. This is due mainly to an amount of noise introduced in the simulations which represents the inaccuracy of robot's infrared perception. 97% of the performed simulations are succeed and permit to the robot to reach the target with smooth way and in finite time, thus, while avoiding local minima and dead end (cf. Figure 7). This is an encouraging result comparing to the lot of constraints imposed by reactive navigation as: no planning step, no global information around the environment, etc. 3% of the simulations that do no't succeed are due mainly to some specific obstacle configurations (no free path solution between the robot and the target) and to the amount of introduced noise.

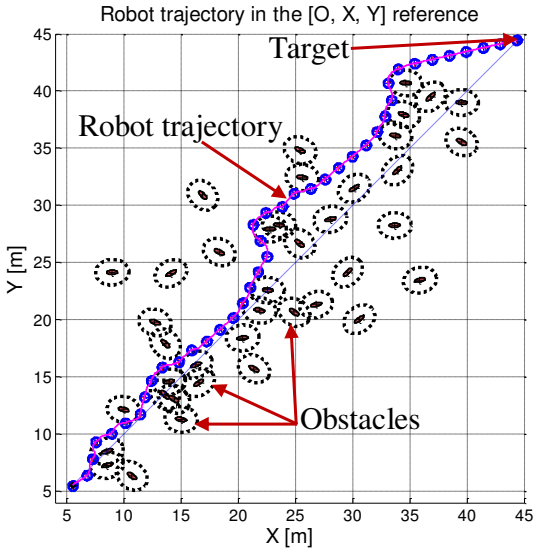


Fig. 7. Smooth robot trajectory obtained with the proposed reactive limit-cycles algorithm

Figure 8 shows the use of elliptic limit-cycle for a specific case where the obstacle to avoid is a wall (a longitudinal object). Otherwise, figure 9 shows the progress value of Lyapunov functions attributed to each controller $V_i|_{i=1..2}$ (cf. Figure 3) when the navigation is performed. These functions decrease asymptotically to the equilibrium point. More details about the stability demonstration of the overall proposed structure of control are given in Adouane (2009a).

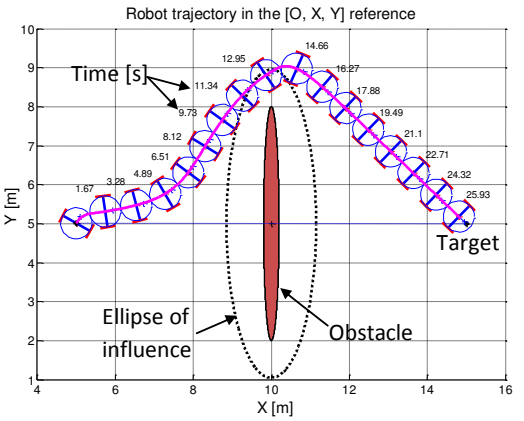


Fig. 8. Wall avoidance strategy using elliptical trajectory

Experimentations are implemented using Khepera III robots. As a very first tests, navigation is achieved on a platform equipped with a camera in the top which gives positions and orientations of the robots and the obstacles to avoid. The camera will be replaced by local infrared sensors in future works to demonstrate the efficiency of the proposed reactive obstacle avoidance algorithm. The real trajectory of the robot avoiding two obstacles is given in figure 10. It can be seen that the robot successfully converges to its target at moment t_e after

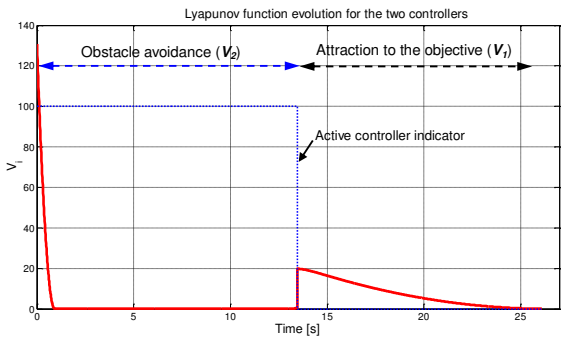


Fig. 9. Evolution of Lyapunov functions for the two used controllers during the robot navigation

avoiding two elliptical obstacles (surrounded with two ellipses of influence).

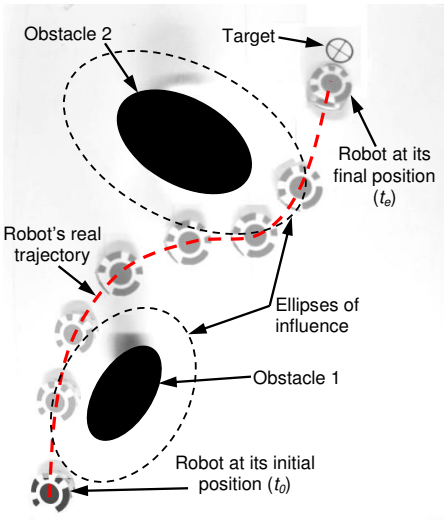


Fig. 10. Top view of the robot trajectory in the platform

6. CONCLUSION AND FURTHER WORK

This paper proposes online and adaptive elliptic trajectories to perform smooth and safe mobile robot navigation. These trajectories use limit-cycle principle to obtain generic and flexible navigation in very unstructured environments. The proposed reactive navigation was embedded in multi-controller architecture and permits for a mobile robot to efficiently navigate in environments with different obstacles shapes. Otherwise, the stability proof of the overall control architecture is given. Simulations and experiments in different environments was done and prove the efficiency and the flexibility of the proposed control architecture. Future works aim, first to embedded all the perception capabilities (as localization and obstacle detection) in the robot and secondly, to adapt the proposed control structure to more complex tasks (as the navigation in highly dynamical environment).

REFERENCES

Adouane, L. (2009a). Hybrid and safe control architecture for mobile robot navigation. In *9th Conference on Autonomous Robot Systems and Competitions*. Portugal.

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

Adouane, L. and Le Fort-Piat, N. (2006). Behavioral and distributed control architecture of control for minimalist mobile robots. *Journal Europen des Systmes Automatiss*, 40(2), pp.177–196.

Arkin, R.C. (1989). Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4), pp.92–112.

Arkin, R.C. (1998). *Behavior-Based Robotics*. The MIT Press.

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

Eberly, D. (2008). Distance from a point to an ellipse in 2d. In *Geometric Tools, LLC*. http://www.geometrictools.com/.

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

Fraichard, T. (1999). Trajectory planning in a dynamic workspace: a state time approach. *Advanced Robotics*, 13(1), 75–94.

Gander, W., Golub, G.H., and Strebel, R. (1994). Fitting of circles and ellipses, bit. 34, 558–578.

Huang, W.H., Fajen, B.R., Fink, J.R., and Warren, W.H. (2006). Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems*, 54(4), 288–299.

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

Jur-Van-Den, B. and Overmars, M. (2005). Roadmap-based motion planning in dynamic environments. *IEEE Transactions on Robotics*, 21(5), 885–897.

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

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5, pp.90–99.

Kim, D.H. and Kim, J.H. (2003). 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.

Latombe, J.C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA.

Laumond, J.P. (2001). *La robotique mobile*. Herms.

Pradalier, C., Hermosillo, J., Koike, C., Braillon, C., Bessire, P., and Laugier, C. (2005). The cycab: a car-like robot navigating autonomously and safely among pedestrians. *Robotics and Autonomous Systems*, 50(1), 51–68.

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

Toibero, J., Carelli, R., and Kuchen, B. (2007). Switching control of mobile robots for autonomous navigation in unknown environments. In *IEEE International Conference on Robotics and Automation*, 1974–1979.

Zapata, R., Cacitti, A., and Lepinay, P. (2004). Dvz-based collision avoidance control of non-holonomic mobile manipulators. *JESA, European Journal of Automated Systems*, 38(5), 559–588.

Zhang, H., Liu, S., and Yang, S.X. (2006). A hybrid robot navigation approach based on partial planning and emotion-based behavior coordination. In *International Conference Intelligent Robots and Systems*, 1183–1188. Beijing, China.