Hybrid and Multi-controller Architecture for Autonomous System: Application to the navigation of a mobile robot

Amani Azzabi[1], Marwa Regaieg[1], Lounis Adouane[2], Othman Nasri[1]

[1]*Research Unit SAGE, ENISo, University of Sousse, BP 264 Sousse Erriadh 4023, Tunisia*

[2]*Institut Pascal, Blaise Pascal University - UMR CNRS 6602, Clermont-Ferrand, France*

Ameni.Azzabi@gmail.com, MarwaRegaiegrm@gmail.com, Lounis.Adouane@univ-bpclermont.fr, Nasri.Othman@gmail.com

Keywords: Control architecture, Mobile robotics, Stability and Reachability analysis, Hybrid dynamic system, Hybrid automata, Interval analysis.

Abstract: This paper deals with the problem of unicycle mobile robot navigation in cluttered environments. It presents in particular an approach which permits to verify the stability of the control architecture of mobile robot using the reachability analysis. To perform this analysis, we consider the robot as a hybrid dynamic system. The latter is modeled by an hybrid automata in order to verify the reachability property by using the interval analysis. The simulation results validate the proposed control architecture.

1 INTRODUCTION

The navigation control of a mobile robot in cluttered environment is a determining problem and is identified as among the priority field of research in the robotics community. The main issues in this field is how to obtain accurate, flexible and reliable navigation? In the proposed set-up, the mobile robot has for mission to reach its target while avoiding any annoying obstacles. Thus, its main behavior is the attraction toward the target and the Obstacle avoidance.

In the literature, a part of the community supposes that the mobile robots use methods of path planning. This means that the environment where it navigates is totally or partially known. Thus, the robot leans on a model of the world which is generally a map supporting a planning. Among these methods, Voronoï diagrams, visibility graphs or artificial potential functions include all the information about the task to be achieved and the environment features (Santiago Garrido and Jurewicz, 2011). The other part of the community admit that robot's navigation is based only on the capacity of the robot to answer to the perceived stimuli using appropriate control law (called reactive) which takes into account the robot's constraints as well as the local state of the environment (Luciano C. A. Pimenta1 and Campos, 2006), (Egerstedt and Hu, 2002).

Several control architectures based reactive modes are proposed in the literature. The conception of the latter, called also behavioral architectures, is based on several elementary controllers/behaviors to be coordinated: Selection of the actions (competitive architectures) and the fusion of the actions (cooperative architectures), (Adouane and Le-Fort-Piat, 2006), (Brooks, 1986). The work proposed in this paper studies the stability of reactive control architecture, proposed in (Adouane, 2009), for a unicycle robot.

Using the considered architecture, it is necessary to guarantee the robot capacity to accomplish its mission while avoiding the obstacles. One of the solutions consists in considering it as as hybrid dynamical system whose behavior is modeled by a hybrid automaton. As shown in (Luciano C. A. Pimenta1 and Campos, 2006), this approach of hybrid control allows the coordination of the action of several mobile robots and checks the reachability property. This analysis consists in determining if a position or a configuration can be reached by the system. Thus it can check that no unwanted behavior of the system will occur. To verify this property for dynamic systems, several approaches are proposed in the literature (J. Toibero and Kuchen, 2007). Among these approaches, we adopt the reachability analysis using the interval analysis (Ramdani et al., 2009). This approach, based on a hybrid automatons, allows finding all the minimal and maximal trajectories of the system, governed by Ordinary Differential Equations (ODE), and to check according to these later if the unwanted system configuration could occurs.

The rest of the paper is organized as follows: we present briefly, in the section 2, the control architecture for the navigation of unicycle mobile robot in a cluttered environment. After having modeled the robot behavior by a hybrid automaton, we shall study in section 3 the control architecture while considering the reachability analysis using the interval analysis. We present in section 4 the simulation results. Section 5 concludes this paper with some prospects.

2 CONTROL ARCHITECTURE FOR REACTIVE NAVIGATION IN A CLUTTERED ENVIRONMENT

We present in this section, the control/command architecture of mobile robot developed in (Adouane, 2009).

2.1 Navigation in a cluttered environment

The mobile robot has for mission to reach its target while avoiding the obstacles met during its navigation. Therefore, its behavior is mainly the attraction toward the target and obstacle avoidance. Furthermore, the navigation of the robot has to be safe, smooth and fast.

The obstacles met during the robot navigation can have different forms. In order that the robot can avoid them, these obstacles are generally encompassed by simple forms of circular or elliptic type (Adouane, 2009), (Adouane et al., 2011). After that, the obstacle avoidance behavior based on limit-cycles can be achieved (Kim and Kim, 2003), (Adouane, 2009), (Adouane et al., 2011).

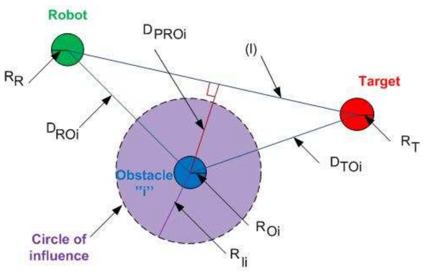

Figure 1: The used perceptions for mobile robot navigation.

In this work, the limit-cycles will be considered as circular shape. In order to avoid collision, the robot and the target will also be characterized or encompassed by circles. Several perceptions are necessary for the navigation of the robot (cf. Fig. 1):

- D_{ROi} distance between the robot and the obstacle "*i*",
- D_{PROi} perpendicular distance between the line (l) and the obstacle "*i*",
- D_{TOi} is the distance between the obstacle i and the target.
- R_R, R_T and R_{Oi} are respectively the radius of the robot, the target and the Obstacle$_i$,
- for each detected obstacle$_i$ we define a circle of influence with a radius of

  $$R_{Ii} = R_R + R_{Oi} + margin$$

  The *margin* corresponds to a safety tolerance which includes: the uncertainty of the perception, control reliability and accuracy, etc.

2.2 Control architecture

The proposed control architecture is represented by the figure 2. This control architecture uses a procedure for selecting a hierarchical action to manage the switching between the controllers according to the environment perception and ensure the stability of the overall control. Its objective is also to ensure safe, smooth and fast robot navigation (Adouane, 2009).

The mechanism of selection activates the "*obstacle avoidance*" controller if at least one obstacle is detected. In order to understand the implemented hybrid architecture of control to check the unicycle mobile robot navigation, it is important to know its kinematic model.

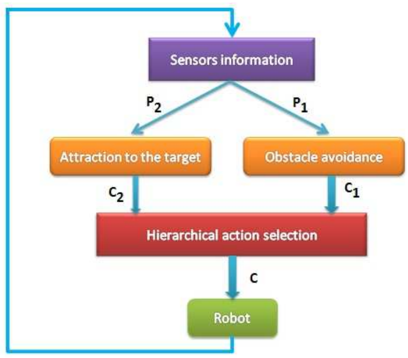

Figure 2: Control architecture for mobile robot navigation.

2.3 Model of the used unicycle robot

The well known kinetic model of a unicycle robot, in a Cartesian reference frame (cf. Fig. 3), is given below:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -l_2\cos\theta - l_1\sin\theta \\ \sin\theta & -l_2\cos\theta + l_1\sin\theta \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

with :

- x, y, θ: configuration state of the unicycle robot at the point P_t of abscissa and ordinate (l_1, l_2) according to the mobile reference frame (O_m, X_m, Y_m) associated to the robot center.
- v: the robot's linear velocity at the point "P_t".
- ω : the robot's angular velocity at the point "P_t".
- (O_A, X_A, Y_A) is the absolute reference.

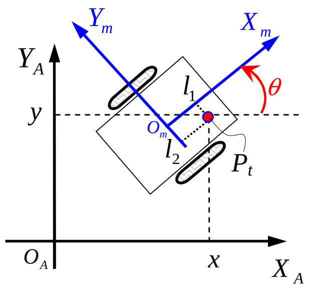

Figure 3: Robot configuration in a Cartesian reference frame.

Knowing the model of the robot and the task to achieve, we present the both controller "*Attraction to the target*" and "*Obstacle avoidance* ". The set of these controllers will be synthesized using the Lyapunov theorem (Benzerrouk et al., 2010).

2.3.1 Attraction to the Target Controller

This controller guides the robot toward the target position represented by a circle of center (x_T, y_T) and of R_T radius. As detailed in (Adouane, 2009), the used control law is a control of position at the point P_t of coordinates $(l_1, 0)$. It is based on the configuration of the robot position relative to the target, represented by the errors e_x and e_y.

To guarantee that the center of the robot axis reaches the target with asymptotic convergence, the distance $d = \sqrt{e_x^2 + e_y^2}$ must be smaller than R_T.

The errors of position are given by the following equations:

$$\begin{cases} e_x = x - x_T \\ e_y = y - y_T \end{cases} \quad (2)$$

Thus:

$$\begin{cases} \dot{e}_x = \dot{x} \\ \dot{e}_y = \dot{y} \end{cases} \quad (3)$$

For the stabilization of the error towards zero, a proportional controller was used (Adouane, 2009):

$$\begin{bmatrix} v \\ w \end{bmatrix} = -K \begin{bmatrix} \cos\theta & -l_1\sin\theta \\ \sin\theta & l_1\sin\theta \end{bmatrix}^{-1} \times \begin{bmatrix} e_x \\ e_y \end{bmatrix} \quad (4)$$

with $K > 0$.

To study the asymptotic stability of the proposed controller, let us consider the following Lyapunov function: $V_1 = \frac{1}{2}d^2$. Therefore, to guarantee this stability in the sense of Lyapunov, it is necessary that: $\dot{V}_1 < 0$, so $d\dot{d} < 0$, what is easily proven as long as $d \neq 0$.

2.3.2 Obstacle Avoidance Controller

During the activation of this controller, the robot follows limit-cycle vector fields given by two differential equations:

- For the clockwise trajectory motion:(cf. Fig 4(a))
  $\dot{x}_s = y_s + x_s(R_c^2 - x_s^2 - y_s^2)$
  $\dot{y}_s = -x_s + y_s(R_c^2 - x_s^2 - y_s^2)$
- For the counter-clockwise trajectory motion:(cf. Fig 4(b))
  $\dot{x}_s = -y_s + x_s(R_c^2 - x_s^2 - y_s^2)$
  $\dot{y}_s = x_s + y_s(R_c^2 - x_s^2 - y_s^2)$

where (x_s, y_s) corresponds to the robot position according to the center of the convergence circle (characterized by an R_c radius). These equations show the direction of trajectories according to axes (x_s, y_s).

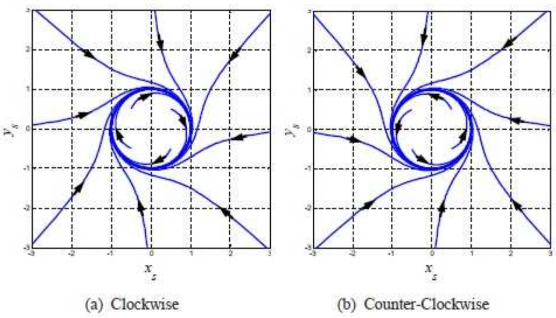

Figure 4: Shape possibilities for the used limit-cycles.

The control law associated with this controller allows the robot to follow the trajectories of the limit-cycles. The robot will be controlled according to its center of coordinates $(l_1, l_2) = (0, 0)$. The desired robot orientation θ_d is:

$$\theta_d = \arctan\left(\frac{\dot{y}_s}{\dot{x}_s}\right) \quad (5)$$

The error of orientation θ_e is given by:

$$\theta_e = \theta_d - \theta \quad (6)$$

The desired orientation is reached using the following control law:

$$w = \dot{\theta}_d + K_p \theta_e \qquad (7)$$

with $K_p > 0$.

The orientation error is given by the following differential equation: $\dot{\theta}_e = -K_p \theta_e$. The asymptotic stability of the controller is verified through the following Lyapunov function: $V_2 = \frac{1}{2}\theta_e^2$.
$\dot{V}_2$ is equal then to $\theta_e \dot{\theta}_e = -K_p \theta_e^2$ which is always strictly negative (thus, asymptotically stable).

2.3.3 Hierarchical Action Selection Block

The block corresponding the selection of action manages the switching between the controllers. It takes a decision thanks to environment's information, collected by the sensors. The selection of action to be made ("*Attraction to the target*" or "*Obstacle avoidance* ") depends on the distance between the perpendicular distance to the line (l) and the distance between the obstacle and the circle of influence (cf. Fig 1).

- If ($D_{PROi} \leq R_{Ii}$) then the controller "*Obstacle avoidance* " is activated.
- Else the controller "*Attraction to the target* " remain always active.

3 STABILITY OF THE CONTROL ARCHITECTURE USING REACHABILITY BY INTERVAL ANALYSIS

In this section, we study the stability of the considered reactive control architecture by checking the reachability property. That's why, it is necessary to model the robot behavior by a hybrid automaton.

3.1 Hybrid Automaton for the Robot's Control

An hybrid automaton handles a set of continuous differential equations, models a dynamic system and describes its behavior. As mentioned in (Ramdani et al., 2009), a hybrid automaton used to find the envelope which frames, in a guaranteed way, the real robot's path, is generally considered as being one 6-uplet $H = (Q, X, P, F, T, RI)$ defined by:

- ◇ Q is a discrete set of modes or situations;
- ◇ X represents the continuous state space (of dimension n), the state vector is noted x and the initial state is noted x_0;
- ◇ $P = [p] = [\underline{p}, \overline{p}]$ represents the parameters' admissible bounded domain;
- ◇ $F = \{(\underline{f}_q, \overline{f_q}), q \in Q\}$, f is the collection of vector fields that surround all the possible dynamics;
- ◇ $T = \{t_e, e \in E\}$ is the collection of the transitions' moments where $E \subseteq Q \times Q$ is the set of transitions;
- ◇ $RI = R_e, e \in E$ is the collection of the functions of update.

According to the case of our studied model, we have:

- ◇ $Q = \{q_1, q_2\}$={ *Attraction to the target* , *Obstacle avoidance*} : the set of discrete states;
- ◇ $X = R^3$, $z = (x, y, \theta) \in X$ is the state vector;
- ◇ $P = \{p_{q_1}, p_{q_2}\}$ is the admissible bounded domain of parameters;
- ◇ $T = \{t_{q_1 q_2}, t_{q_2 q_1}\}$ where
$$t_{q_1 q_2} = (q_1, D_{PROi} \leq R_{Ii}, Id, q_2)$$
and
$$t_{q_2 q_1} = (q_2, D_{PROi} > R_{Ii}, Id, q_1)$$
- ◇ $F = (\underline{f}_q, \overline{f_q}), q \in Q$ where f is the collection of vector fields that surround all the possible dynamics.

The unicycle robot behavior in a cluttered environment can be then modeled by the hybrid automaton of the figure 5.

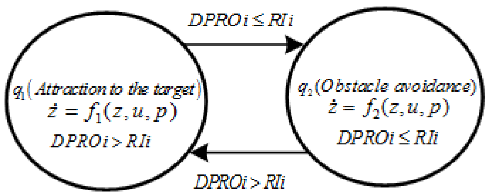

Figure 5: Hybrid automaton of the navigation task to be performed by the robot.

The continuous dynamics of the robot in the discrete state q_1 at the point P_t of coordinates (l_1, 0) is defined by:

$$\dot{z} = f_{q_1}(z, u, p_{q_1}) = \begin{cases} \dot{x} = v\cos\theta - wl_1 \sin\theta \\ \dot{y} = v\sin\theta - wl_1 \cos\theta \\ \dot{\theta} = w \end{cases} \qquad (8)$$

with $z = (x, y, \theta)$; $u = (v, w)$ and $p_{q_1} = (l_1, K)$.

The continuous dynamics of the robot in the discrete state q_2 which is controlled by its center of coordinates $(l_1, l_2) = (0,0)$ is defined by:

$$\dot{z} = f_{q_2}(z, u, p_{q_2}) = \begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = w \end{cases} \tag{9}$$

with $z = (x, y, \theta)$, $u = (v, w)$ and $p_{q_2} = (K_p)$.

The invariant regions associated with both discrete states q_1 and q_2 are respectively $(D_{PROi} > R_{Ii})$ and $(D_{PROi} \leq R_{Ii})$. The transition from a state to another requires that the position and the orientation of the robot verify the conditions of guard $((D_{PROi} > R_{Ii})$ and $(D_{PROi} \leq R_{Ii}))$. The update functions *RI* are consider as the identity functions because the robot dynamics doesn't present discontinuity during the crossing of the transitions between the discrete states.

After modeling the robot behavior by an hybrid automaton, we will verify the stability of the control architecture considered using the reachability analysis.

3.2 Principle of the Hybrid Bounding Method

As indicated in (Ramdani et al., 2009), The method of hybrid bounding is an approach of reachability analysis. It allows thus to find the envelope enclosing all possible trajectories of the robot in order to study its stability. For that purpose, we will detail the algorithm used to calculate an over-approximation of the reachable space.

The algorithm 1 of the hybrid bounding method looks first for the initial discrete mode q through the function Initialization "*Initialization*". This function identifies the initial discrete state according to the signs of all the partial derivatives of the vector fields relative to the uncertain parameters. In our case, these parameters are θ, x and y. Then, as long as the time t doesn't reach the final value t_{nT}, a guarantee integration of the current mode is given by the algorithm 2 "*Integer-one-step*" which makes a step of integration noted h. Next, according to the signs of partial derivatives, the function "switching" verifies if there is a transition towards another mode on the interval $[x_j, x_{j+1}]$ and calculates the new mode q'. If it is the case, then the boolean variable *jump*, that indicates the existence of a transition, sets the boolean value to 1 and if not it takes the value 0. The switching from a mode to another depends on the current mode. If it is equal to 0 then it is enough to turn to the new mode $q' \neq 0$ and continues the integration. Else ($q \neq 0$),

Algorithm 1 Main algorithm

1 ◇ **Inputs** : t_0, t_{nT}, F, $[x_0]$, $[p]$
2 ◇ **Outputs** : $[\tilde{x}_0]$, $[\tilde{x}_1]$,..., $[\tilde{x}_{nT}]$, $[x_1]$,..., $[x_{nT}]$
3 **Begin**
4 $j := 0$;
5 q := Initialization (f, $[x_0]$, $[p]$) ;
6 while($j < nT$) do
7 $(h_j, [x_{j+1}], [\tilde{x}_j])$:= Integer-one-step$(q, f, F, t_j, [x_j], [p])$;
8 $(jump, q_0)$:= switching $(q, f, [\tilde{x}_j])$;
9 if $(jump)$ then
10 if $(q = 0)$ then
11 q := q_0;
12 j := $j+1$
13 else
14 $q := 0$
15 end if
16 else
17 j:=j+1
18 end if
19 **End**

it is necessary to repeat the integration on the mode $q = 0$ to save the guarantee of the frame for the transition crossing. Besides, the algorithm 2 "*Integer-one-step*" detailed below calculates on each step of integration, a protected frame from solutions of uncertain differential equation. In fact, for the case of the mode $q = 0$ the digital integration of the uncertain differential equation is given through the method of Taylor Intervalle (MTI)in the line 5. This method contains essentially two steps:

- a step of prediction which verify the existence and the uniqueness of the solution.
- a step of correction which calculate the solution $[x_{j+1}]$ at the moment $t_{j+1} = t_j + h_j$.

However, for the modes $q \neq 0$, firstly we select the ODE bounding in the line 7 and the initial conditions at the moment $t_j(\omega(t_j), \Omega(t_j))$ are fixed to lines 8 and 9. Then, the digital integration of this EDO bounding is executed in the line 10. Finally, to guarantee numerically the obtained results, we use as in the line 5 the integration methods based on the models of Taylor intervals to solve the ODE.

4 SIMULATION RESULTS

The stability of the proposed architecture is verified using the hybrid bounding method. For that purpose,

Algorithm 2 Integer-one-step

◇ **Inputs**: q, F, t_j , $[x_j]$, $[p]$

◇ **Outputs**: h_j , $[x_{j+1}]$, $[\widetilde{x}_j]$

Begin

if q := 0 then

$(h_j, [x_{j+1}], [\widetilde{x}_j])$:= MTI($f, [x_j], [p], t_j$);

else

$(\underline{f}_q, \overline{f}_q)$:= select frame(q, F) ;

$[\omega_j]$:= $[\underline{x}_j]$;

$[\Omega_j]$:= $[\overline{x}_j]$;

$(h_j, [\omega_{j+1}], [\Omega_{j+1}], [\widetilde{\Omega}_j], [\widetilde{\omega}_j])$:= MTI($\underline{f}_q$, $\overline{f}_q$, $[\omega_j]$, $[\Omega_j]$, $[\underline{p}]$, $[\overline{p}]$, t_j) ;

$[\widetilde{x}_j]$:= $[\widetilde{\omega}_j , \widetilde{\Omega}_j]$;

$[x_{j+1}]$:= $[\omega_{j+1}, \Omega_{j+1}]$;

end if

End

we used the MATLAB toolbox INTLAB and we suppose that the data received from the robot sensors are uncertain. Thus, the state vector components are represented by intervals translating this uncertainty on the position (x et y) and on the orientation θ. The execution of the hybrid bounding algorithm over the period of simulation $[0,5]s$ gave the results of figures 6, 7 and 8.

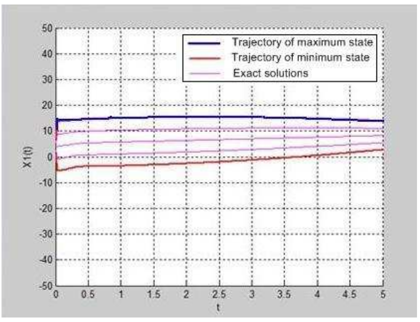

Figure 6: Frame of all the possible solutions of $x(t)$ due to the hybrid bounding method.

The simulation results show the guarantee frame of all the temporal trajectories of the robot. Indeed, there is no intersection with hazardous areas and all solutions do not leave the envelope during the period of simulation. Therefore, these simulations prove the stability of the robot and demonstrate the efficiency of the reactive control architecture presented in section 2.

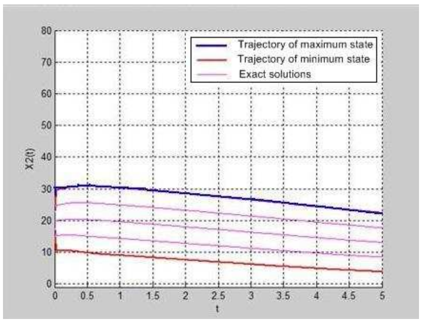

Figure 7: Frame of all the possible solutions of $y(t)$ due to the hybrid bounding method.

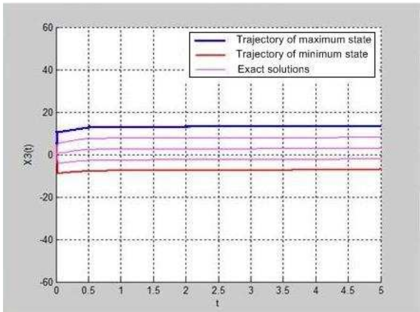

Figure 8: Frame of all the possible solutions of $\theta(t)$ due to the hybrid bounding method.

5 CONCLUSION

In this work, we were interested in the problem of mobile robot navigation in a cluttered environment. In particular, the stability of the dynamics of the robot using reactive control architecture was studied. At first, we modeled the set(continuous dynamics of the robot and the discrete change of the used control laws) by a hybrid automaton. Thereafter, we used an approach of reachability analysis based on the interval method in order to verify the stability of the dynamic system even in presence of uncertainty. This approach is based on the calculation of an over-approximation of the reachable space set of the robot (which correspond to all the points of the robot's trajectory). The idea was to find a guarantee frame of all the trajectories possible taking into account the uncertainties on the data received by the sensors. Thus, the reachability analysis is handled by calling on to the interval analysis. Finally, we validated the proposed solution by simulations. The obtained result show that the considered control architecture allows the robot to have a smooth behavior while avoiding the obstacles met throughout its navigation toward the target. Future work will aims to test in real experimentation the obtained results.

ACKNOWLEDGEMENTS

This work was supported by LABEX IMobS3 Innovative Mobility: Smart and Sustainable Solutions.

REFERENCES

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

Adouane, L., Benzerrouk, A., and Martinet, P. (2011). Mobile robot navigation in cluttered environment using reactive elliptic trajectories. In *18th IFAC World Congress*, Milano-Italy.

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

Benzerrouk, A., Adouane, L., and Martinet, P. (2010). Lyapunov global stability for a reactive mobile robot navigation in presence of obstacles. In *ICRA'10 International Workshop on Robotics and Intelligent Transportation System, RITS10*, Anchorage-Alaska.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(10).

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

Guguen, H., Lefebvre, M.-A., Zaytoon, J., and Nasri, O. (2009). Safety verification and reachability analysis for hybrid systems. *Annual Reviews in Control*, 33(1):25 – 36.

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

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.

Luciano C. A. Pimenta1, Alexandre R. Fonseca1, G. A. S. P. R. C. M. E. J. S. W. M. C. and Campos, M. F. M. (2006). Robot navigation based on electrostatic field computation. *IEEE Transactions on Magnetics*, 42(4).

Ramdani, N., Meslem, N., and Candau, Y. (2009). A Hybrid Bounding Method for Computing an Over-Approximation for the Reachable Set of Uncertain Nonlinear Systems. *IEEE Transactions on Automatic Control*, 54(10):2352 – 2364.

Santiago Garrido, Luis Moreno, D. B. and Jurewicz, P. (2011). Path planning for mobile robot navigation using voronoi diagram and fast marching. *International Journal of Robotics and Automation (IJRA)*, 2(1).

Vilca, J.-M., Adouane, L., and Mezouar, Y. (2013). Reactive navigation of mobile robot using elliptic trajectories and effective on-line obstacle detection. *Gyroscopy and Navigation Journal*, 4(1).