Obstacle Avoidance Controller Generating Attainable Set-points for the Navigation of Multi-Robot System

Ahmed Benzerrouk, Lounis Adouane, Philippe Martinet

Obstacle Avoidance Controller Generating Attainable Set-points for the Navigation of Multi-Robot System

A. Benzerrouk[1], L. Adouane[1] and P. Martinet[2]

[1] Clermont Université, Université Blaise Pascal, BP 10448, 63000 Clermont-Ferrand, France
[2] IRCCYN, Ecole Centrale de Nantes, 1 rue de la Noé,
BP 92101, 44321 Nantes Cedex 03, France
Ahmed.BENZERROUK@lasmea.univ-bpclermont.fr

***Abstract*— This paper considers the navigation in formation of a mobile Multi-Robot System (MRS) in presence of obstacles. In such areas, the collision avoidance between the robots themselves and with other obstacles (static and dynamic) is a challenging issue. To deal with it, a reactive and a distributed control architecture is built. The navigation in formation of the MRS is ensured while *tracking a global virtual structure* (first controller). Limit-cycle principle is used to compute the set-point of the *obstacle avoidance* task (second controller). In this paper, kinematic constraints of the robot are taken into account in order to generate an attainable set-point. The objective is to guarantee safety of the mobile robots with respect to their maximum velocities. Simulation and experimental results validate the proposed contributions.**

I. INTRODUCTION

Navigation of multiple mobile robots is a recurrent research subject due to a large amount of the met issues. Safety of the robots in cluttered environment is among the most important ones. Collision avoidance is then widely investigated in the literature for multi-robot systems. It is tackled through two main approaches. The first one considers the robots control entirely based on path planning methods, which involve the prior knowledge of the robots environment. The objective is to find the best path to all the robots in order to avoid all the obstacles and each other while minimizing a cost function [1], [2]. This first method requires a significant computational complexity, especially when the environment is highly dynamic. In fact, the robot has to frequently replan its path.

Rather than a prior knowledge of the environment, reactive methods are based on local robots sensors information. At each sample time, robot's control is computed according to its perceived environment. Potential field [3] and the Deformable Virtual Zone (DVZ) [4] are a good illustration of reactive approaches. The reactive methods given above suffer from local minima problems when, for instance, the sum of potential forces is null, or the deformation of the DVZ is symmetric (as in the U shape obstacle case). Generally, reactive methods do not require high computational complexities, since robots actions must be given in real-time according to the perception.

The distributed architecture of control, that we developed [5], deals with this last kind of methods. The studied task is the navigation in formation. The formation is considered as a virtual structure (rigid body) and the control law for each robot is derived by defining the dynamics of this body. Virtual structure approach is often associated to potential field applications since they are simple and allow collision avoidance [6], [7]. However, potential forces are limited, especially when the formation shape needs to be frequently reconfigured. In fact, it means that the robot is submitted to a frequently-changing number/amplitude of forces leading to more local minima, oscillations, etc. Hence, it was proposed that the robots track a virtual body without using potential forces. Since collision avoidance must stay possible despite the absence of potential fields, behavior-based concept [8], [9] was introduced. This allows to divide the task into two different behaviors (controllers): *Attraction to Dynamic Target*, and *Obstacle Avoidance* (cf. Figure 1). The latter was based on limit-cycle differential equations [10]. Limit-cycle navigation was already used for obstacle avoidance [11], [12]. It allows to choose the obstacle avoidance direction (clockwise or counterclockwise) in order to rapidly join the assigned target. In [13], it is proposed to extend this method to dynamic obstacles and to robots of the same system without loosing the control reactivity. Unlike most of algorithms addressing dynamic obstacles, no communication is required among the robots to accomplish the task. Avoidance is based only on the local perception of each robot. As in [11], [12], the idea is to find the best direction of avoidance. It was proved that only the velocity vector of the obstacle is sufficient to deduce this direction. In this paper, our architecture is enriched by constraining the set-point generated by obstacle avoidance controller: this set-point has to be attainable despite the maximum velocities of the robot and dimensions of the obstacles in order to guarantee the robot's safety. New parameters are then introduced to the set-point formula to prevent the robot from collision.

The remainder of the paper is organized as follows. Section II gives the principle of the navigation in formation and the general control architecture. Basic controllers and the control law are reminded in this section. In section III, the set-point generated by obstacle avoidance is modified to deal with each robot according to its maximum velocity and to the obstacle dimensions. Section IV validates the proposed contribution with experimental results. Finally, we conclude and give some perspectives in section V.

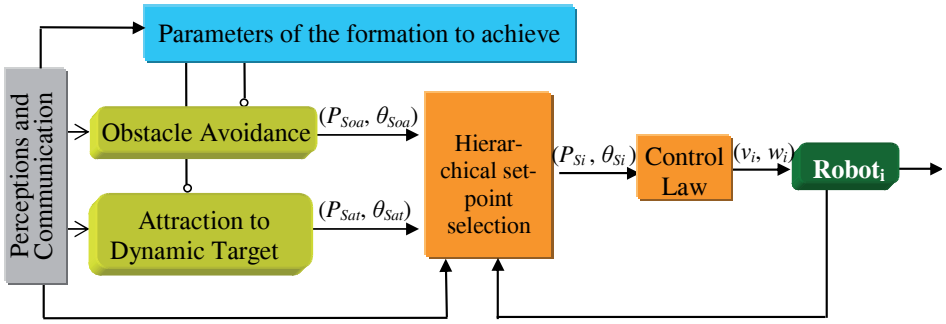


Fig. 1. The proposed architecture of control embedded in each robot.

II. Control Architecture

The used control architecture includes two controllers: *Attraction to Dynamic Target* and *Obstacle Avoidance*. The virtual structure is built through the *Parameters of the Formation to Achieve* block (cf. Figure 1).

According to environment information collected by the *Perceptions and Communication* block (sensors) and the robot's current state, one controller is chosen thanks to the *Hierarchical Set-Point Selection* block.

The corresponding set-points (P_{S_i}, θ_{S_i}) (position and orientation) are then sent to the *Control Law* block which calculates the linear and angular velocities noted v_i and w_i respectively (cf. Figure 1).

A. Parameters of the Formation to Achieve block

This subsection briefly describes the adopted virtual structure principle. Consider N robots with the objective of reaching and maintaining them in a given formation. The proposed virtual structure that must be followed by the group of robots is defined as follow:

- Define one point which is called the main dynamic target (cf. Figure 2),
- Define the virtual structure to follow by defining N_T nodes (virtual targets) to obtain the desired geometry. Each node i is called a secondary target and is defined according to a specific distance D_i and angle Φ_i with respect to the main target. The number of these targets N_T must be $N_T \geq N$.

Each robot i has to track a predefined target i. An exemple to get a triangular formation is given in figure 2.

B. Attraction to Dynamic Target controller

To remind the *Attraction to Dynamic Target* controller which allows to reach and to keep the formation, consider a robot i with (x_i, y_i, θ_i) pose. This robot has to track its secondary dynamic target. To simplify notations in the following, the same subscript of the robot is given to its target. The latter is then noted $T_i(x_{T_i}, y_{T_i}, \theta_T)$ (cf. Figure 3) and the variation of its position can be described by

$$\begin{cases} \dot{x}_{T_i} = v_{T_i}.cos(\theta_T) \\ \dot{y}_{T_i} = v_{T_i}.sin(\theta_T) \end{cases} \quad (1)$$

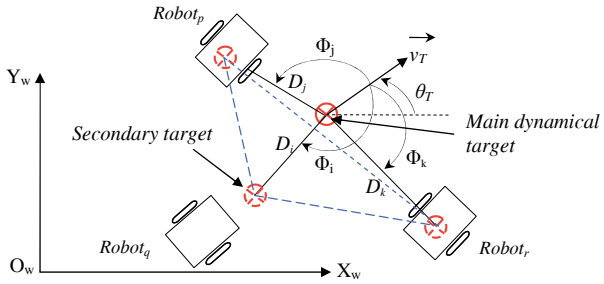


Fig. 2. Keeping a triangular formation by defining a virtual geometrical structure.

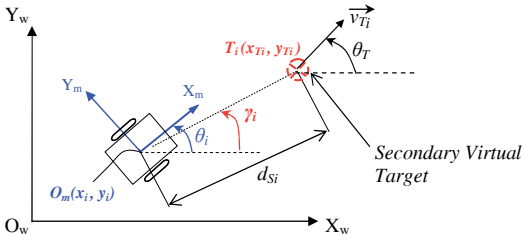


Fig. 3. Attraction to Dynamic Target.

Let's also introduce the used robot model (cf. Figure 3). Experimental results are made on Khepera robots, which are unicycle mobile robots. Their kinematic model can be described by the well-known equations (cf. Equation 2).

$$\begin{cases} \dot{x}_i = v_i.cos(\theta_i) \\ \dot{y}_i = v_i.sin(\theta_i) \\ \dot{\theta}_i = \omega_i \end{cases} \quad (2)$$

where θ_i, v_i and ω_i are respectively the robot orientation, the linear and angular velocities.

The set-point angle that the robot must follow, to reach its dynamic target, is given by

$$\theta_{S_{ati}} = \arcsin(b \sin(\theta_T - \gamma_i)) + \gamma_i \quad (3)$$

Where $b = \frac{v_{T_i}}{v_i}$. γ_i is the angle that the robot would have if it was directed to its target (cf. Figure 3). This set-point

has been obtained by keeping γ_i constant. More details and proofs are available in [5].

The corresponding set-points (P_{S_i}, θ_{S_i}) (cf. Figure 1) given by the *Attraction to Dynamic Target* controller are composed by:

- $(P_{S_i} = (x_{T_i}, y_{T_i}))$: the current position of the dynamic target (cf. Figure 3),
- $(\theta_{S_i} = \theta_{S_{ati}})$ given by equation (3).

C. Obstacle Avoidance controller

A particular attention is given to this controller since the objective of the paper is to make its set-point attainable despite the kinematic constraints of the robots. As cited in section I, the task is performed through the limit cycle methods. The robot follows the limit cycle vector fields described by the following differential equations:

$$\begin{aligned} \dot{x}_s &= (sign) y_s + \mu x_s (R_c^2 - x_s^2 - y_s^2) \\ \dot{y}_s &= -(sign) x_s + \mu y_s (R_c^2 - x_s^2 - y_s^2) \end{aligned} \quad (4)$$

where $(x_s,\ y_s)$ corresponds to the relative position of the robot according to the center of the convergence circle (characterized by an R_c radius).

The function $sign$ allows to define the direction of the trajectories described by these equations. Hence, two cases are possible:

- $sign = 1$, the motion is clockwise.
- $sign = -1$, the motion is counterclockwise.

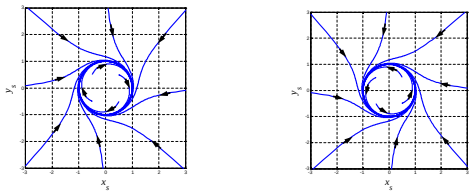
Figure 4 shows the limit cycles with a radius $R_c = 1$. The Obstacle is then covered by a circle, which is itself surrounded by an other virtual circle of influence with R_c radius (cf. Figure 6). The latter is chosen as the sum of the obstacle radius, the robot radius and a safety margin. μ is a positive constant. Figure 5 illustrates its influence on the limit-cycle trajectory. The choice of this constant will be rigorously discussed in section III to generate an attainable set-point.

The set-point angle $\theta_{S_{oa}}$ of the *Obstacle Avoidance* controller is given by the following relation

$$\theta_{S_{oa}} = \arctan(\frac{\dot{y}_s}{\dot{x}_s}) \quad (5)$$

The corresponding set-points (P_{S_i}, θ_{S_i}), -when the *Obstacle Avoidance* controller is chosen by *Hierarchical Set-Point Selection block* (cf. Figure 1)-, are defined such that

- $(P_{S_{oa}} = (x_o, y_o))$ corresponds to the center position of the obstacle,



(a) Clockwise $(sign = 1)$ (b) Counter-Clockwise $(sign = -1)$

Fig. 4. Possible trajectories of the limit-cycles

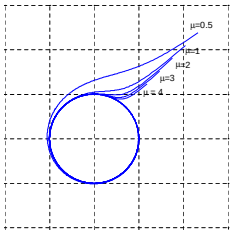


Fig. 5. Influence of μ on the limit-cycle trajectory smoothness.

- $(\theta_S = \theta_{S_{oa}})$.

It is noticed that previous works on limit-cycle methods applied to obstacle avoidance [11], [12] do not consider dynamic obstacles. Here, it is proposed to extend this reactive method to deal with them.

According to the nature of the obstacle, three cases are considered: static obstacles, dynamic obstacles, and robots of the same system. These strategies are briefly reminded in the next paragraphs. More details are available in [13].

1) static obstacles,
2) dynamic obstacles,
3) robots of the same system.

1) Static obstacles: The same strategy proposed in [12] is maintained. Summarily, the value of $sign$ is specified by the ordinate of the robot y_s in the relative obstacle's frame $(O_o X_o Y_o)$ (cf. Figure 6). The X_o axis of this orthonormal frame is defined thanks to two points: the center of the obstacle (which makes the origin of the frame) and the target to reach.

$$sign = \begin{cases} 1 & \text{if } y_s \geq 0 \text{ (clockwise avoidance)} \\ -1 & \text{if } y_s < 0 \text{ (counterclockwise avoidance)} \end{cases} \quad (6)$$

The chosen direction by this strategy allows then to join the target by the side offering the smallest covered distance.

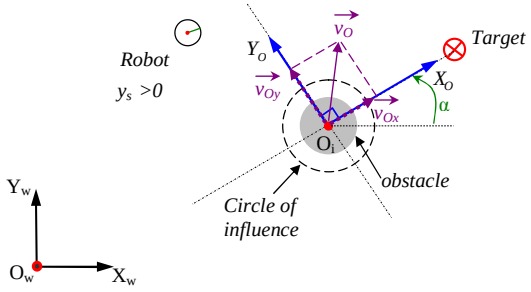


Fig. 6. Avoiding an obstacle. Static obstacle: the ordinate y_s is analyzed, dynamic one: projection of $\vec{v}_O$ is analyzed.

2) Dynamic obstacles: Rather than analyzing the sign of y_s, it is proposed that the robot uses the obstacle's vector velocity $\vec{v}_O$. The idea is to project this vector on the Y_o axis of the relative frame $(O_o X_o Y_o)$ defined in paragraph II-C.1.

The function $sign$ (cf. Equation 4) is then defined according to v_{O_y} as follows:

$$sign = \begin{cases} 1 & \text{if } v_{O_y} \leq 0 \text{ (clockwise avoidance)} \\ -1 & \text{if } v_{O_y} > 0 \text{ (counterclockwise avoidance)} \end{cases} \quad (7)$$

By using the projection v_{O_y} of the obstacle velocity, the obstacle is always avoided round the back, such that the robot never cuts off the obstacle's trajectory.

3) Robots of the same system: One can consider that every robot of the MRS is treated as a dynamic obstacle and projects its velocity vector to deduce the side of avoidance (cf. Equation 7). However, a conflict problem could appear when, for instance, two robots have to avoid each other in opposite directions calculated by velocity vector projections.

To deal with this kind of conflicts, and assuming that each robot is able to identify those of the same system, it is proposed to impose one reference direction for all the system. Hence, when one robot detects a disturbing robot of the same group, it always avoids it counterclockwise.

D. The control law block

This block allows for the robot i to converge to its set-point given by the *Hierarchical set-point selection* block (cf. Figure 1). It is expressed as

$$v_i = v_{max} - (v_{max} - v_T)e^{-(d_{S_i}^2/\sigma^2)} \tag{8a}$$

$$\omega_i = \omega_{S_i} + k\tilde{\theta}_i \tag{8b}$$

where

- v_{max} is the maximum linear speed of the robot,
- σ, k are positive constants,
- v_i and ω_i are linear and angular velocities of the robot. $\omega_{S_i} = \dot{\theta}_{S_i}$,
- and $\tilde{\theta}_i = \theta_{S_i} - \theta_i$ is the error orientation.

θ_{S_i} is the set-point angle according to the active controller (cf. Equation (3), (5)). Asymptotic stability of the control law is demonstrated in [13]. In fact, it can be easily deduced from equation (8b) that the error orientation exponentially converges.

It is also noticed that linear velocity v_i is made so that $v_i \leq v_{max}$ is always verified. Naturally, for target following case, it is imposed that $v_T < v_{max}$ to attain the virtual target. Next section, the main contribution of this paper, prevents saturation of the angular velocity from occurring despite the robot's kinematic constraints.

III. Obstacle avoidance with respect to kinematic robot constraints

Now, we are interested in the maximum angular velocity of the robots ω_{max}, such that the variation of the angular set-point $\dot{\theta}_{S_{oai}}$ remains attainable (i for the i^{th} robot) and safety of the robot guaranteed. As previously explained, we are interested in the obstacle avoidance case. *Attraction to Dynamic Target* study is subject of a future paper.

It is clear that the angular velocity applied to the robot has to verify

$$|\omega_i| \leq \omega_{max} \tag{9}$$

where $\omega_{max} > 0$. By replacing (8b) in (9), we have

$$\left|\omega_{S_i} + k\tilde{\theta}_i\right| \leq \omega_{max} \tag{10}$$

knowing that $\left|\omega_{S_i} + k\tilde{\theta}_i\right| \leq |\omega_{S_i}| + \left|k\tilde{\theta}_i\right|$

To find the values of ω_{S_i} which verify (10), it is proposed to use

$$|\omega_{S_i}| + k\left|\tilde{\theta}_i\right| \leq \omega_{max} \tag{11}$$

To be always verified, the latter relation then becomes

$$|\omega_{S_i}| \leq \min_{|\tilde{\theta}_i|}(\omega_{max} - k\left|\tilde{\theta}_i\right|) \tag{12}$$

Which leads to

$$|\omega_{S_i}| \leq (\omega_{max} - k\ max\left|\tilde{\theta}_i\right|) \tag{13}$$

Since the proposed control law is asymptotically stable (cf. Section II-D), and the orientation error is exponentially decreasing, the following relation is easily deduced

$$\max\left|\tilde{\theta}_i\right| = \left|\tilde{\theta}_i(t_s)\right| \tag{14}$$

where t_s is the switching moment to the *Obstacle Avoidance* controller.

Since $\omega_{S_i} = \dot{\theta}_{S_{oai}}$, let us compute $\dot{\theta}_{S_{oai}}$ according to equation (5)

$$\dot{\theta}_{S_{oa}} = \frac{\frac{d}{dt}(\frac{\dot{y}_s}{\dot{x}_s})}{(1+(\frac{\dot{y}_s}{\dot{x}_s})^2)} \tag{15}$$

To develop $\dot{\theta}_{S_{oa}}$, we note

$$A = R_c^2 - x_s^2 - y_s^2 \tag{16}$$

Using equation (4), (15) leads to

$$\dot{\theta}_{S_{oa}} = -sign - 2sign\mu^2 A(x_s^2 + y_s^2)/(sign^2 + \mu^2 A^2) \tag{17}$$

Replacing in (13), we obtain

$$\left|1 + 2\mu^2 A\frac{(x_s^2 + y_s^2)}{(1 + \mu^2 A^2)}\right| \leq \omega_{max} - k\left|\tilde{\theta}_i(t_s)\right| \tag{18}$$

We can use the following relation

$$1 + \left|2\mu^2 A\frac{(x_s^2 + y_s^2)}{(1 + \mu^2 A^2)}\right| \leq \omega_{max} - k\left|\tilde{\theta}_i(t_s)\right| \tag{19}$$

In fact, values of μ verifying (19), verify also (18).
On the other side, it is clear that (cf. Equation 18)

$$0 \leq \left|2\mu^2 A\frac{(x_s^2 + y_s^2)}{(1 + \mu^2 A^2)}\right| \leq \omega_{max} - k\left|\tilde{\theta}_i(t_s)\right| - 1 \tag{20}$$

The gain k has then to verify

$$\omega_{max} - k\left|\tilde{\theta}_i(t_s)\right| - 1 > 0 \tag{21}$$

and the allowed values of k are

$$k < \frac{\omega_{max} - 1}{\left|\tilde{\theta}_i(t_s)\right|} \tag{22}$$

To deal with the worst possible configurations, k is chosen such that

$$k < \frac{\omega_{max} - 1}{\pi} \tag{23}$$

In fact, the maximum value of $\left|\tilde{\theta}_i(t_s)\right| = \pi$, since the maximal possible orientation error corresponds to the case where the robot orientation is in the opposite of the set-point angle.

The left member of the inequation (18) can be bounded as follows

$$1 + \left| 2\mu^2 A \frac{(x_s^2 + y_s^2)}{(1 + \mu^2 A^2)} \right| \leq 1 + \left| 2\mu^2 A (x_s^2 + y_s^2) \right| \quad (24)$$

In fact, using the right member of (24) is simpler to find values of μ verifying the condition (20). Hence, to find these values, the following relation is used

$$2\mu^2 |A| (x_s^2 + y_s^2) \leq \omega_{max} - k \left| \tilde{\theta}_i(t_s) \right| - 1 \quad (25)$$

In what follows, we note $P_{oa} = \omega_{max} - k \left| \tilde{\theta}_i(t_s) \right| - 1$. To always generate a reachable set-point angle $\dot{\theta}_{S_{oa}}$ of the obstacle avoidance controller, μ has then to be chosen as

$$\mu^2 \leq \frac{P_{oa}}{2 |A| (x_s^2 + y_s^2)} \quad (26)$$

The distance of the robot to the obstacle noted d_{RO} can be introduced to the last relation (26) which becomes (replacing A defined in (16))

$$\mu^2 \leq \frac{P_{oa}}{2 |R_c^2 - d_{RO}^2| d_{RO}^2} \quad (27)$$

To find a least upper bound of μ regardless of d_{RO}, it is proposed to compute the minimum of the right member of (27) (which corresponds to the maximum of its denominator). When the obstacle avoidance is activated, two cases can be distinguished :

1) $d_{RO} < R_c$ (the robot is inside the limit-cycle) this gives

$$Den = (R_c^2 - d_{RO}^2) d_{RO}^2$$

Its derivative with respect to d_{RO} is

$$\frac{\partial Den}{\partial d_{RO}} = 2 d_{RO} (R_c^2 - 2 d_{RO}^2) \quad (28)$$

Roots corresponding to the maximum of Den are $\pm \frac{R_c}{\sqrt{2}}$. (the solution $d_{RO} = 0$ is rejected since it means that the distance between the robot and the obstacle centers is null, which is impossible). Replacing in (27), μ has to satisfy

$$\mu \leq \frac{1}{R_c^2} \sqrt{2 P_{oa}} \quad (29)$$

2) $d_{RO} > R_c$ (the robot is outside the limit-cycle) Den becomes

$$Den = -(R_c^2 - d_{RO}^2) d_{RO}^2$$

its derivative is

$$\frac{\partial Den}{\partial d_{RO}} = -2 d_{RO} (R_c^2 - 2 d_{RO}^2)$$

There is no solution satisfying the condition $d_{RO} > R_c$ (the Den domain of definition corresponding to the second case). In addition, Den is always increasing and $max(Den)$ is attained when $d_{RO} \rightarrow \infty$. In practice, the robot is continuously approaching the obstacle (the robot is outside the limit-cycle in this case) and the maximum considered distance d_{RO} can be chosen when the obstacle avoidance controller is activated. It is noted d_{RO0}.

μ must then satisfy the following condition

$$\mu \leq \sqrt{\frac{P_{oa}}{2 \left| R_c^2 - d_{RO_0}^2 \right| d_{RO_0}^2}} \quad (30)$$

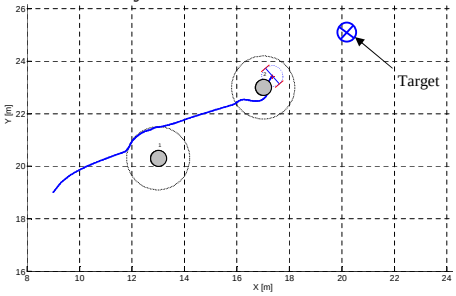
Finally, note that the case where $d_{RO} = R_c$ means that the robot is on the limit-cycle. According to relation (27), any value of μ can then be accepted. In fact, figure 5 shows that μ does not affect the trajectory smoothness on the limit-cycle but only when converging to it.

Next section illustrates how the choice of μ directly influences the safety of the robots.

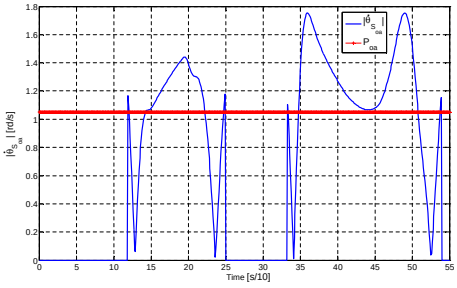
IV. Simulation and Experimental Results

A. Simulation results

It is proposed to show how the proposed bound of the parameter μ constrains the set-point angle $\dot{\theta}_{S_{oa}}$ and then guarantees a safe navigation. A mobile robot going toward a static target $(v_T = 0)$ in presence of obstacles is simulated. We set $\omega_{max} = 3rd/s$, and $k = 0.6$ (cf. Equation 23). First, the simulation is accomplished using $(\mu = 1)$ which assumes that the classic equations of limit-cycles are used (cf. Equation 4) (without μ). Figure 7(a) shows that the robot avoids the first obstacle but fails to avoid the second one. Figure 7(b) shows the variation of the set-point angle $\dot{\theta}_{S_{oa}}$: it increases and becomes higher than the authorized value P_{oa} imposed by the maximum angular velocity of the robot ω_{max}. It means that the robot's dynamic can not follow this variation and then may collide with the obstacle.



(a) Trajectory of the robot

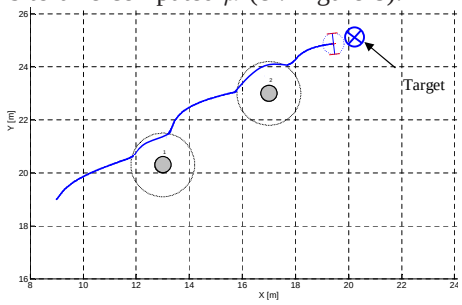


(b) Variation of the set-point angle

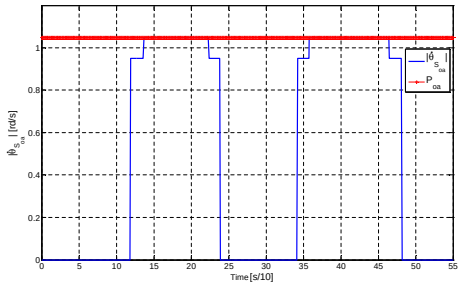
Fig. 7. A mobile robot avoiding two obstacles (constant $\mu = 1$).

Now, simulation is run again in the same environment (position and dimension of the obstacles, initial conditions of the robot) by replacing μ with its constrained value (cf.

Equation 30) (the robot is outside the obstacles). It is noticed that this time, the robot succeeds to avoid the two obstacles (cf. Figure 8). The variation of the set-point can not exceed P_{oa} thanks to a re-computed μ (cf. Figure 8).



(a) Trajectory of the robot



(b) Variation of the set-point angle

Fig. 8. Dynamic of the mobile robot avoiding two obstacles (μ recomputed for each one).

B. Attaining a formation while avoiding collision between the robots

Experimentations are made on Khepera III robots and illustrate a navigation in formation of the robots while avoiding each other. A central camera, at the top of the platform gives positions of all the robots and the obstacles thanks to circular bar codes installed on them. The objective, in short term horizon, is to use the local sensors of the robots in order to get a completely decentralized architecture.

The scenario illustrates three robots which have to join a triangular virtual structure. The latter moves along a circular trajectory (cf. Figure 9). Robots are put in their initial conditions so that they must avoid each other before joining the formation. It is observed that the collision avoidance is successfully accomplished for all the robots. Moreover, no conflict was observed since avoidance is done in one direction (robots of the same system)(cf. Section II-C.3). The formation is attained as shown in figure 9 illustrating the trajectories of the three robots.

V. Conclusion

The proposed control architecture devoted to the navigation in formation in presence of obstacles must be enriched to generate only attainable set-points. In fact, the proposed control law is theoretically stable. However, in practice, additional constraints must be taken into account. In our case, kinematic constraints (maximum velocities) of the robot imposes to define maximum authorized set-points. It is then proposed to study the obstacle avoidance controller case.

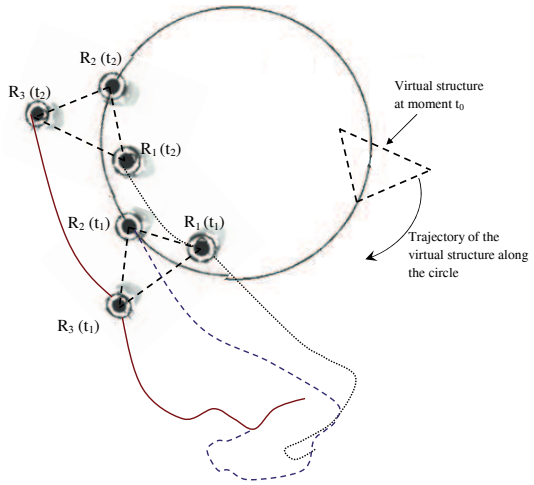


Fig. 9. Trajectories of the robots attaining the formation.

Usually, its set-point depends on the obstacle characteristics (dimensions, shape, etc.). These parameters depends on the environment and can not be directly modified. A new parameter is then added to adapt the set-point according to these characteristics. Saturation of the velocities are avoided while ensuring safety of the robot. Future works will tackle the *Attraction to Dynamic Target* controller constraints. The objective is to define the allowed dynamic of the virtual structure to stay attainable.

References

[1] S. J. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey. Clearpath: Highly parallel collision avoidance for multi-agent simulation. In *ACM SIGGRAPH Eurographics symposium on computer animation*, 2009.

[2] A. Pongpunwattana and R. Rysdyk. Real-time planning for multiple autonomous vehicles in dynamic uncertain environments. *Journal of Aerospace Computing, Information and Communication*, 1:580–604, 2004.

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

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

[5] A. Benzerrouk, L. Adouane, L. Lequievre, and P. Martinet. Navigation of multi-robot formation in unstructured environment using dynamical virtual structures. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[6] P. Ogren, E. Fiorelli, and Leonard N. E. Formations with a mission: Stable coordination of vehicle group maneuvers. In *15th International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[7] K. D. Do. Formation tracking control of unicycle-type mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 527–538, 2007.

[8] T. Balch and R.C. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Autmation*, 1999.

[9] G. Antonelli, F. Arrichiello, and S. Chiaverini. The nsb control: a behavior-based approach for multi-robot systems. *PALADYN Journal of Behavioral Robotics*, 1:48–56, 2010.

[10] H.K. Khalil. *Frequency domain analysis of feedback systems (chapter 7)*. 2002.

[11] 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.

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

[13] A. Benzerrouk, L. Adouane, and P. Martinet. Dynamic obstacle avoidance strategies using limit cycle for the navigation of multi-robot system. *4th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (IROS '12)*, 2012.