Stable navigation in formation for a multi-robot system based on a constrained virtual structure
Ahmed Benzerrouk, Lounis Adouane, Philippe Martinet

To cite this version:

HAL Id: hal-01712946
https://hal.archives-ouvertes.fr/hal-01712946
Submitted on 23 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Stable Navigation in Formation for a Multi-robot System Based on a Constrained Virtual Structure

Ahmed Benzerrouk\textsuperscript{a}, Lounis Adouane\textsuperscript{a}, Philippe Martinet\textsuperscript{b}

\textsuperscript{a}Clermont Universit\'{e}, Institut Pascal UMR CNRS 6602, 63000 Clermont-Ferrand, France.
\textsuperscript{b}Ecole Centrale de Nantes, IRCCyN UMR CNRS 6597, 44321 Nantes Cedex 03, France.

Abstract
This paper deals with the navigation in formation of a group of mobile robots. A set of virtual targets (points) forms a virtual structure of the same shape as the desired formation. Hence, to join and to remain in this formation, each robot has only to track one of these targets. In order to track the chosen target, it has to be attainable by the robot despite its kinematic constraints. This paper studies then the maximum allowed dynamic of the virtual structure according to the kinematic constraints of the robots. Both linear and angular velocities of the targets are constrained. Moreover, depending on these velocities, some relative positions (targets) in the formation become unattainable. These positions are also defined. A stable control law allows to attain the generated set-points. Simulation and experimental results validate the proposed contributions.

Keywords: Cooperative multirobot system, Navigation in formation, Virtual structure, Kinematic constraints, Lyapunov-based stability.

1. Introduction

Controlling and coordinating a Multi-Robot System (MRS) is still an attractive research subject thanks to its large number of applications. Compared to one robot, a MRS offers many advantages: improvement of execution task time, execution efficiency, redundancy of sensors and actuators providing better failure tolerance, etc.

Navigation in formation is one of the most important studied issues of these systems. In fact, many tasks require the MRS to move while maintaining a desired pattern (space exploration [1], platooning [2], rescue operations [3], etc.)

In the literature, the control problem in mobile robot navigation is tackled through two methods: deliberative and reactive control (cf. Figure 1). The first approach, based on motion planning and paths, requires a prior knowledge of the environment to plan the robots’ movements [4]. This approach uses formalisms like Voronoi diagrams [5] or artificial potential functions [6], [7], while considering the overall environment knowledge [8]. Motion planning generally trade-off optimality for efficiency/reactivity. In fact, they do not scale well to a very large number of robots due to their computation complexity [9]. However, thanks to the prior knowledge of the environment, robots generally succeed in their mission with a good performance. In the second approach, the reactive method, robots act only according to their local sensor information without any other global knowledge. Behavior-based methods [10] are the perfect illustration of reactive control. In fact, the global task of the robot is divided into a set of sub-tasks (behavior patterns). According to sensor information, the control strategy applied to the robot derives from one selected behavior pattern [10], or is a merging of several weighted patterns [11]. When the application requires that robots operate in real time (e.g. in hazardous environments), it is clear that reactive methods become much more interesting than motion planning. In these cases, many research problems have not yet been resolved. To keep only the advantages of the two methods (deliberative and reactive), some hybrid control (deliberative/reactive) has been explored in the literature. The idea is to enable the suitable control according to the situation. For example, in [12] or [13], a feasible trajectory is planned before to be tracked and if a new obstacle is detected, a reactive obstacle avoidance control is locally enabled.

The proposed paper is devoted to the navigation in formation of a MRS in a fully reactive way without any motion path planning. For this kind of task, the literature highlights three main approaches: hierarchical, behavior-based, and the virtual structure strategy (cf. Figure 1) [14], [15], [16]. In the first approach, one or more robots are considered as leaders while the other robots are the followers. Generally, the leader tracks a predefined trajectory while the followers track its transformed coordinates [17], [18]. This approach is simple to perform. However, it is noticed that a leader failure causes the whole system to stop. In a distributed behavior-based approach [19], [20], there is no hierarchy between the robots. Each one has its perception and control, and a failure of a robot does not lead to a group failure [21]. Behavior-based strategy implies that
each robot has a set of behavior patterns (basic tasks) to achieve. The resulting behavior of the group emerges from the basic local interaction without any explicit model of the overall cooperative behavior. However, this approach is criticized for the way that it chooses the control for each robot. In fact, according to perception information, the control system switches between behavior patterns (e.g., competitive approach [10]), or merges several controllers (e.g., motor schema [11]). This naturally makes it hard to study the stability of the global control strategy. The virtual structure (the third approach) considers the formation as a single virtual body. The shape of the latter is the desired formation shape, and its motion is translated into the desired motion of each vehicle [22], [23]. The virtual structure is implemented in several works through potential field methods [24], [25]: thus, all the members of the formation track assigned nodes which move into the desired configuration. In these works, nodes applied an attractive field to the corresponding robot, whereas obstacles and neighbor robots apply repulsive fields. Unlike motion planning, potential functions applied for the virtual structure approach use only the instantaneous and local robots perception. The weakness of using potential functions for this last approach corresponds to the increasing complexity for controlling the fleet shape in dynamic environment. In fact, it means that the robot is submitted to a frequently-changing number/amplitude of forces leading to more local minima, oscillations, etc. Therefore, in this case, it is very difficult to demonstrate the robustness and the stability of the MRS navigation.

To overcome the drawbacks of these strategies, it was proposed to combine the virtual structure and behavior-based approaches in [26] (cf. Figure 1). The achieved task (reaching and maintaining a desired formation while avoiding collision) is divided into two basic tasks (behavior patterns): attraction to a dynamic target, and obstacle avoidance. The first basic task allows each robot to reach and follow a target from a virtual structure. The latter is elaborated through multiple virtual nodes. The second task focuses on avoiding obstacles and collisions between robots. To facilitate a possible reconfiguration of the formation with the proposed control architecture, potential field methods are avoided.

Moreover, in [26], the linear velocity of the virtual structure was constrained according to the maximum linear velocity of the robots, in order to remain attainable. However, all the nodes were considered with the same linear velocity. Experiments were then made for a straight trajectory of the virtual structure. In the proposed paper, we define possible relative positions in the virtual structure and thus, different angular and linear velocities of the targets. Moreover, the authorized angular velocity for the structure is studied according to the maximum angular speed of the robots. Experiments will be implemented using a circular formation trajectory.

The remainder of the paper is organized as follows: in section 2, the control architecture and the cooperative strategy between the robots are given. Section 3 studies the control applied to each robot, its stability and its limitations. Simulation and experimental results validate the proposed contributions in section 4. A conclusion and some prospects are given in section 5.

2. The overall control architecture based on behavior-based and virtual structure approaches

As discussed in section 1, the used control architecture [26] includes two controllers: Attraction to a Dynamic Target and Obstacle Avoidance (behavior-based part). The virtual structure is built through the Parameters of the Formation to Achieve block (cf. Figure 2).

2.1. The proposed control architecture

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 \((P_i, \theta_i)\) (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 2).

Let us recall the adopted virtual structure principle. Consider \(N\) robots with the objective of achieving and maintaining a given formation. The proposed virtual structure that must be followed by the group of robots is defined as follows:

- Define one point which is called the main dynamic target (cf. Figure 3),
- 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 \(\Phi_i\) with respect to the main target. Secondary targets defined in this way have then the same orientation \(\theta_T\). However, each target \(i\) will have its own linear velocity \(v_{ti}\). The number of these targets \(N_T\) must be \(N_T \geq N\).

It is clear that to have a complete distribution of the control, the main target can be generated by one of
Figure 2: The proposed architecture of control embedded in each robot.

2.2. Cooperative strategy between robots: dynamic target allocation

Each mobile robot should follow one of the secondary targets forming the geometric shape. It is interesting to optimize the allocation of the targets between the robots to rapidly join the formation. Information available for each robot, i.e., its configuration \((x_i, y_i, \theta_i)\), the one of the main virtual target \((x_T, y_T, \theta_T)\) and the \(D_j\) and \(\Phi_j\). \((x, y)\) refers to the position and \(\theta\) to the orientation of robots/targets. \(D_j\) and \(\Phi_j\) are the relative positions and orientations of the secondary targets \(j\) (cf. Figure 3).

In [26], each robot chooses the closest target. If one target is desired by -at least- two robots, it is allocated to the one with the highest rank. In fact, each robot has an identifier defining its hierarchic rank in the formation. We improved this work in [27]. Hence, each robot calculates a relative cost coefficient RCC for each target. It is a cost minimization function which helps each robot to choose the closest target. If this target is needed by another robot having more difficulties to find an other, it is given up to this robot (a form of altruism, where the interest of the group comes before the individual interest). The objective is to reduce time to attain the global formation while keeping a reactive and distributed control architecture. The idea of the dynamic allocation of the targets has been inspired by the auction sales activities which allow a task allocation for the MRS (exploration [28], visiting different locations [29], box pushing [30], etc.). These auction methods can be divided in three different strategies: Combinatorial methods treat all possible combinations to give the optimal distribution to the MRS [31]; Repeated parallel auctions occur every time interval to check that every robot has the suitable task [32]; Sequential mechanisms where the robot auctions each task taking into account its previous state [33]. Even if it gives the best solution, combinatorial method can not be adopted because it needs generally a central unit. This means that the overall architecture of control can not be completely distributed. Moreover, the computation complexity rapidly increases with the robot and target numbers. The proposed RCC algorithm is derived from the two other strategies. The robot wins or looses a target by computing and comparing its own RCCs for these targets. Only a minimalist communication is needed between the robots, and they decide for their targets in fully distributed and reactive way. In order to focus on the main contribution of this paper (attainability of the virtual structure), the reader can find all details about this algorithm in [27]. It is noted that experimentations made in this paper (cf. subsection 4.2) are only based on RCC algorithm.

3. Applied Robot Control

Once every robot has chosen its target, its mission is to reach this dynamic target and to track it to maintain the formation. Attraction to a Dynamic Target Controller (cf. Figure 2) gives set-points (position, angle) that, if they are followed by the robot, it converges to its dynamic chosen target (in position and orientation). In section 3.2, the Obstacle Avoidance Controller will be very briefly summarized in order to focus on the main contribution of this
paper (constrains on the virtual structure). More details can be found in [34]. Improvement of the controller to address dynamic obstacles and collision avoidance will be proposed in a future paper.

3.1. Attraction to a Dynamic Target Controller

Attraction to a Dynamic Target Controller allows the formation to be maintained, since it leads each robot to follow its target. To detail this controller, consider a robot \( i (x_i, y_i, \theta_i) \) and its secondary dynamic target \( T_i(x_T, y_T, \theta_T) \). Note that to simplify notations in the following, the same subscript as the robot is given to its target(cf. Figure 4). The variation of the target position can be described by

\[
\begin{cases}
\dot{x}_T = v_T \cos(\theta_T) \\
\dot{y}_T = v_T \sin(\theta_T)
\end{cases}
\]  

Let us also introduce the robot used model (cf. Figure 4). The proposed study focuses on unicycle mobile robots corresponding to the used robots for simulation and experimental part (cf. Section 4). Their kinematic model can be described by the well-known equations (cf. Equation 2).

\[
\begin{cases}
\dot{x}_i = v_i \cos(\theta_i) \\
\dot{y}_i = v_i \sin(\theta_i) \\
\dot{\theta}_i = \omega_i
\end{cases}
\]  

where \( v_T \) is the target linear velocity, \( v_i \) and \( \omega_i \) are respectively the robot linear and angular velocities. It is to be noted also that \( \dot{x} \) corresponds to the derivative of \( x \) with respect to time.

Figure (4) allows to define position errors as

\[
\begin{cases}
e_{x_i} = (x_T - x_i) = d_{S_i} \cos(\gamma_i) \\
e_{y_i} = (y_T - y_i) = d_{S_i} \sin(\gamma_i)
\end{cases}
\]  

The current distance between robot \( i \) and its target \( T_i \), noted \( d_{S_i} \), can then be expressed as

\[
d_{S_i} = \sqrt{e_{x_i}^2 + e_{y_i}^2}
\]  

Its derivative is

\[
d_{S_i}' = \frac{e_{x_i} \dot{e}_{x_i} + e_{y_i} \dot{e}_{y_i}}{d_{S_i}}
\]  

By using equations (1) and (2), \( \dot{e}_x \) and \( \dot{e}_y \) are then given by

\[
\begin{cases}
\dot{e}_{x_i} = (\dot{x}_T - \dot{x}_i) = v_T \cos(\theta_T) - v_i \cos(\theta_i) \\
\dot{e}_{y_i} = (\dot{y}_T - \dot{y}_i) = v_T \sin(\theta_T) - v_i \sin(\theta_i)
\end{cases}
\]  

We then obtain

\[
d_{S_i}' = v_T \cos(\gamma_i - \theta_T) - v_i \cos(\gamma_i - \theta_i)
\]  

Similarly, the current angle of the robot according to its dynamic target is noted \( \gamma_i \) (cf. Figure 4) and is calculated as

\[
\gamma_i = \arctan\left(\frac{e_{y_i}}{e_{x_i}}\right)
\]  

Its derivative is

\[
\dot{\gamma}_i = \frac{e_{x_i} / e_{x_i}}{1 + (e_{y_i} / e_{x_i})^2}
\]  

To obtain the set-point angle \( \theta_{S_{xt}} \) applied to the robot in order to reach its dynamic target, our idea is to keep \( \gamma_i \) constant. In other words, we would like to have \( \dot{\gamma}_i = 0 \). Under this constraint, we show that the defined set-point angle leads the robot to its target. Developing equation (9) thus enables us to write:

\[
\frac{v_T \sin(\theta_T - \gamma_i) - v_i \sin(\theta_i - \gamma_i)}{d_{S_i}} = 0
\]  

The set-point angle that the robot must follow to satisfy the constraint expressed by equation (10) and to reach its dynamic target is then given by

\[
\theta_{S_{xt}} = \arcsin\left(\frac{\gamma_i}{\gamma_i} \sin(\theta_T - \gamma_i)\right) + \gamma_i
\]  

In the following, it is noted \( b = \frac{v_T}{v_i} \).

To prove that the robot always reaches its target, we have to prove that \( d_{S_i} \) is continually decreasing. To do this, it is sufficient to prove that \( d_{S_i}' < 0 \). Before giving the proof, it is stipulated that the linear velocity of the robot will be elaborated satisfying the constraint \( v_i \geq v_T \), (the expression of \( v_i \) is given later in equation (17a)). It is natural that the robot moves faster than the target to reach it, especially when the latter is escaping. Therefore, we always have \( b = \frac{v_T}{v_i} \leq 1 \).

By reaching the target, the robot velocity has then to satisfy \( v_i \geq v_T \), to keep \( d_{S_i} \rightarrow 0 \) (cf. Section 3.3). Hence, we will obtain \( b = 1 \).

In addition, the trajectory of the target is assumed to be smooth.
To prove that while \(d_{Si} \neq 0, \dot{d}_{Si} \text{ is always } d_{Si} < 0 \) if the robot follows the set-point angle given by equation 11, the following properties are recalled

\[
\arcsin(x) \in [-\frac{\pi}{2}, \frac{\pi}{2}], \forall x \in [-1, 1]
\]

\[
\arcsin(\sin(x)) = \begin{cases} 
  x & \text{if } x \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\
  \pi - x & \text{if } x \in [\frac{\pi}{2}, \pi]
\end{cases}
\]

Consider equation (7). The following transformation is considered

\[
cos(\theta_T - \gamma_i) = \pm \sqrt{1 - (\sin(\theta_T - \gamma_i))^2}
\]

By replacing \(\theta_i\) in (7) with the set-point angle that the robot must follow (cf. Equation 11), we also obtain

\[
cos(\gamma_i - \theta_i) = \pm \sqrt{1 - \sin((\arcsin(b\sin(\theta_T - \gamma_i)))^2)
\]

two cases are then possible:

1. \((\theta_T - \gamma_i) \in [-\frac{\pi}{4}, \frac{\pi}{4}]\) (escaping target (cf. Figure 5(a)))

   this leads to

   \[
   \dot{d}_{Si} = v_T \sqrt{1 - (\sin(\theta_T - \gamma_i))^2} - v_i \sqrt{1 - (b\sin(\theta_T - \gamma_i))^2} \tag{12}
   \]

   However, while the robot has not yet reached the target, we have \(b < 1\) since \(v_T < v_i\) as discussed above. This means that

   \[
v_T \sqrt{1 - (\sin(\theta_T - \gamma_i))^2} < v_i \sqrt{1 - (b\sin(\theta_T - \gamma_i))^2}
   \]

   thus

   \[
   \dot{d}_{Si} < 0
   \]

   (a) Escaping target

   \[
   \dot{d}_{Si} = v_T \sqrt{1 - (\sin(\theta_T - \gamma_i))^2} - v_i \sqrt{1 - (b\sin(\theta_T - \gamma_i))^2}
   \tag{13}
   \]

   It can then immediately be deduced that \(\dot{d}_{Si} < 0\).

   It is interesting to note that the proposed set-point enables convergence to \(\theta_T\) as \(d_{Si} \to 0\). In fact, since \(b = 1\) when \(d_{Si} \to 0\) as discussed in the last proof, two cases are again possible:

   1. \((\theta_T - \gamma_i) \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right]\) (approaching target (cf. Figure 5(b)))

   \[
   \dot{d}_{Si} = -v_T \sqrt{1 - (\sin(\theta_T - \gamma_i))^2} - v_i \sqrt{1 - (b\sin(\theta_T - \gamma_i))^2}
   \tag{14}
   \]

   (b) Approaching target

   \[
   \theta_{Sat} = \arcsin(\sin(\theta_T - \gamma_i)) + \gamma_i
   \]

   \[
   \theta_{Sat} = \gamma_i + \gamma_i = \gamma_i
   \tag{15}
   \]

   The set-point angle tends directly to the target direction.

   2. \((\theta_T - \gamma_i) \in \left[\frac{\pi}{2}, \frac{3\pi}{2}\right]\) (approaching target (cf. Figure 5(b)))

   \[
   \theta_{Sat} = \pi - (\theta_T - \gamma_i) + \gamma_i
   \]

   \[
   \theta_{Sat} = \pi + 2\gamma_i - \theta_T
   \tag{16}
   \]

   However, the robot still reaches the target, but with this set-point angle it goes past it once reached. The robot is then behind the target and tries to join it again. Therefore, \(\gamma_i\) is recalculated. Since the target trajectory is assumed to be smooth (constrained in section 3.3), the new calculated \(\gamma_i\) thus verifies case 1 (the robot is now behind the target, which then becomes an escaping target).

   The proposed set-point angle allows each robot to converge to its target by decreasing the position and orientation error. The set-points \((P_{Sat}, \theta_{Sat})\) corresponding to this controller (cf. Section 2.1) are then

   \[
   (P_{Sat}, \theta_{Sat}) = (x_T, y_T, \theta_{Sat})
   \]

3.2. Obstacle avoidance controller

The aim of this controller is to allow the robot to avoid obstacles that hinder its attraction to the target. It is based on the limit cycle methods [34], [35], [36]. The differential equations representing the desired trajectory of the robot are given by the following system

\[
\begin{align*}
\dot{x}_r &= a y_r + x_r(R_e^2 - x_r^2 - y_r^2) \\
\dot{y}_r &= -a x_r + x_r(R_e^2 - x_r^2 - y_r^2)
\end{align*}
\tag{16}
\]

With \(a = \pm 1\) according to the optimal direction of avoidance (clockwise or counterclockwise). \((x_r, y_r)\) are the relative robot coordinates with respect to the obstacle. This latter is characterized by a circle of radius \(R_d = R_o + R_e + \epsilon\) where: \(R_o\) is the obstacle radius, \(R_e\) is the robot radius and \(\epsilon\) is a safety margin (cf. Figure 6).

The obstacle avoidance algorithm [34] is summarized as follows:
Figure 6: Obstacle avoidance controller [34].

- The nearest hindering obstacle is detected.
- The direction of avoidance is chosen according to the sensor information.
- The robot avoids the obstacle while following a limit cycle which has a radius $R_e = R_{cl} - \xi$ (attraction phase).
- The robot avoids the obstacle while following a limit cycle which has a radius $R_e = R_{cl} + \xi$ (repulsive phase) (cf. Figure 6). Where $\xi$ is a small value and ($\xi \ll \epsilon$).

This controller then generates the following set-points:

$$(P_{S_{oa}}, \theta_{S_{oa}}) = ((x_i, y_i), \tan^{-1}(\frac{y_i}{x_i}))$$

To deal with dynamic obstacles, this controller has been enriched. Hence, collision with dynamic obstacles and robots of the same system can be avoided. Moreover, for a higher safety, a penalty function adapts the robots velocities if they are too close from each other. More details are available in [37].

One advantage of the proposed control architecture is that we have the same control law for Attraction to dynamic target and for Obstacle avoidance controllers (cf. Figure 2). An accurate Lyapunov-based stability demonstration is given in [26]. This demonstration is briefly reminded in next section 3.3. According to it, the error between the current robot’s configuration $(x_i, y_i, \theta_i)$ and the desired static/dynamic configuration $(P_S, \theta_S)$, is always steady and converge to 0.

### 3.3. Control law

The control law used, which was proposed in [26], allows each robot $i$ to converge to the set-point generated by the chosen controller.

$$v_i = v_{max} - (v_{max} - v_{Ti})e^{-(d_{Si}^2/\sigma^2)}$$

$$\omega_i = \omega_{Si} + k_3 \hat{\theta}_i$$

where

- $\omega_{Si} = \dot{\theta}_{Si}$, for the attraction to a dynamic target controller, and $\omega_{Si} = \dot{\theta}_{Soa}$ for the obstacle avoidance controller.
- $v_{max}$ is the maximum linear speed of the robot. Naturally, $v_{Ti}$ has to be such that $v_{Ti} \leq v_{max}$
- $\sigma, k$ are positive constants,
- $\hat{\theta}_i$ is the orientation error such that $\dot{\theta}_i = \theta_{Si} - \theta_i$ ($\dot{\theta}_i = \theta_{Soa} - \theta_i$ for the obstacle avoidance) which gives $\dot{\hat{\theta}}_i = \dot{\theta}_i = \omega_{Si} - \omega_i$.
- $d_{Si}$ is the distance between the current robot, position and its attributed target. It is to be noted that for obstacle avoidance controller, the set-point position is equal to $(x_i, y_i)$, thus $d_{Si} = 0$ in this case (cf. Section 3.2).

Lyapunov-based stability enabled the convergence of the robot with its target to be proved [26]. Convergence of the whole multi-robot system to the set-point virtual structure can then be derived by studying the following Lyapunov function

$$V = \sum_{k=1}^{N} V_k$$

where $V_k$ is the Lyapunov function associated with robot $k$. This function was defined as

$$V_k = \frac{1}{2} \dot{\theta}_k^2$$

It has been proved that $\dot{V}_k < 0$ (when $\dot{\theta}_k \neq 0$) [26]. Therefore, it can be easily deduced that

$$\dot{V} = \sum_{k=1}^{N} \dot{V}_k < 0$$

The system is then asymptotically stable. Moreover, it can be noticed that the applied angular velocity $\omega_i$ enables the exponential convergence of the orientation error toward 0. The control law is common to the two controllers (attraction to a dynamic target and obstacle avoidance) and stability is then valid for the obstacle avoidance.

However, this theoretical convergence is applied to non-holonomic mobile robots. This means that stability will be ensured only if the angular set-point is attainable by the robot, considering its kinematic constraints (maximum velocities). In the next paragraph, we then propose to define the permitted dynamic of the target, taking into account the corresponding robot constraints.

### 3.4. Target attainability with respect to kinematic robot constraints

#### 3.4.1. Linear velocity constraints

In section 3.1 it was shown that the robot always converges to its target if

$$v_{Ti} \leq v_i \iff b \leq 1$$

...
According to equation (17a), the linear velocity of the robot verifies the condition given by inequation (21), and takes into account its maximum linear velocity. However, it is clear that the linear velocity of the secondary targets depends on their relative positions in the virtual structure (cf. Figure 7). This figure shows the different trajectories of the targets according to their relative position in the virtual structure. The choice of \( D_i \) and angle \( \Phi_i \) thus affects \( v_{T_i} \). Each secondary target \( i \) has coordinates \((x_{T_i}, y_{T_i})\) expressed as

\[
\begin{align*}
    x_{T_i} &= x_T + D_i \cos(\Phi_i + \theta_T) \\
    y_{T_i} &= y_T + D_i \sin(\Phi_i + \theta_T)
\end{align*}
\]

Their derivatives are then (only rigid virtual structures are considered)

\[
\begin{align*}
    \dot{x}_{T_i} &= \dot{x}_T - D_i \dot{\theta}_T \sin(\Phi_i + \theta_T) \\
    \dot{y}_{T_i} &= \dot{y}_T + D_i \dot{\theta}_T \cos(\Phi_i + \theta_T)
\end{align*}
\]

and linear velocity of the virtual target \( i \) can be written as

\[
v_{T_i} = \sqrt{\dot{x}_{T_i}^2 + \dot{y}_{T_i}^2}
\]

By replacing with (23)

\[
v_{T_i} = \sqrt{\dot{x}_T^2 + D_i^2 \dot{\theta}_T^2 + 2D_i \dot{\theta}_T F}
\]

where

\[
F = (\dot{y}_T \cos(\Phi_i + \theta_T) - \dot{x}_T \sin(\Phi_i + \theta_T))
\]

It is clear that

\[
\dot{y}_T \cos(\Phi_i + \theta_T) - \dot{x}_T \sin(\Phi_i + \theta_T) \leq \sqrt{\dot{x}_T^2 + \dot{y}_T^2}
\]

then

\[
\dot{y}_T \cos(\Phi_i + \theta_T) - \dot{x}_T \sin(\Phi_i + \theta_T) \leq v_T
\]

Hence, according to equation (25)

\[
v_{T_i} \leq \sqrt{(v_T + D_i \dot{\theta}_T)^2}
\]

Robots must be able to move faster than their targets (cf. Equation 21). This means

\[
|v_T + D_i \dot{\theta}_T| < v_{max}
\]

The relative distance of each secondary target has then to be

\[
D_i < \frac{v_{max} - |v_T|}{|\dot{\theta}_T|}
\]

Note that \(|\dot{\theta}_T|\) is bounded in the next paragraph.

3.4.2. Angular velocity constraints

For now, we are interested in the maximum angular velocity of the robots, \( \omega_{max} \), such that the variation of the angular set-point \( \dot{\theta}_{S_{ati}} \) remains attainable. Indeed, the angular speed applied to the robot has to verify

\[
|\omega_i| \leq \omega_{max}
\]

where \( \omega_{max} > 0 \). By replacing (17b) in (31), we have

\[
|\omega_{S_i} + k \dot{\theta}_i| \leq \omega_{max}
\]

knowing that

\[
|\omega_{S_i} + k \dot{\theta}_i| \leq |\omega_{S_i}| + |k \dot{\theta}_i|
\]

To find the values of \( \omega_{S_i} \) which verify (32), it is proposed to use

\[
|\omega_{S_i}| + k |\dot{\theta}_i| \leq \omega_{max}
\]

These values then verify

\[
|\omega_{S_i}| \leq \omega_{max} - k |\dot{\theta}_i|
\]

It is clear that \(|\omega_{S_i}|\) has to verify this condition (inequation 34) for every \( \dot{\theta}_i \). The latter relation then becomes

\[
|\omega_{S_i}| \leq \min(\omega_{max} - k |\dot{\theta}_i|)
\]

\(\min(\omega_{max} - k |\dot{\theta}_i|)\) is obtained when \(|\dot{\theta}_i|\) is maximum. The maximum orientation error is when the robot has an opposite orientation compared with the set-point angle. This means that \(\max(|\dot{\theta}_i|) = \pi\).

To remain attainable, the angular variation of the set-point \( \omega_{S_i} = \dot{\theta}_{S_{ati}} \) has thus to verify

\[
|\omega_{S_i}| \leq \omega_{max} - k \pi
\]
Let us compute $\omega_S = \dot{\theta}_{Sat}$, according to equation (11)

$$\dot{\theta}_{Sat} = \frac{\frac{d}{2} \left[ b \sin(\theta_T - \gamma_i) \right]}{\sqrt{1 - (b \sin(\theta_T - \gamma_i))^2}} + \gamma_i$$  \hspace{1cm} (37)

As noted in section 3.1, the set-point was deduced by keeping $\gamma_i$ constant, which means that $\dot{\gamma_i} = 0$. It can also be noted that $\dot{\theta}_{Sat}$ is not defined if we have simultaneously

$$b = 1$$

(38)

$b = 1$ is true only if $d_S \rightarrow 0$ (cf. Equation 17a), which means that this singularity may occur only when the robot is on its target. To avoid this in practice, we propose to redefine $\theta_{Sat}$ according to equation (11), by considering a virtual circle of radius $\rho$ in the neighborhood of $d_S \approx 0$. Considering the cases of approaching and escaping targets, $\theta_{Sat}$ then becomes

$$\theta_{Sat} = \begin{cases} \arcsin(b \sin(\theta_T - \gamma_i) + \gamma_i) & \text{if } d_S \geq \rho \\ \theta_T & \text{if } d_S < \rho \text{ and } (\theta_T - \gamma_i) \in \left[\frac{\pi}{2}, \frac{3\pi}{2}\right] \\ \pi + 2\gamma_i - \theta_T & \text{if } d_S < \rho \text{ and } (\theta_T - \gamma_i) \in \left[\frac{-\pi}{2}, \frac{-3\pi}{2}\right] \end{cases}$$

(39)

As $\theta_{Sat}$ is redefined, its derivative becomes ($\gamma_i$ constant)

$$\dot{\theta}_{Sat} = \begin{cases} \frac{\frac{1}{b} \left[ (b \sin(\theta_T - \gamma_i) + \gamma_i) \right]}{\sqrt{1 - (b \sin(\theta_T - \gamma_i))^2}} & \text{if } d_S \geq \rho \\
\dot{\theta}_T & \text{if } d_S < \rho \text{ and } (\theta_T - \gamma_i) \in \left[\frac{\pi}{2}, \frac{3\pi}{2}\right] \\ -\dot{\theta}_T & \text{if } d_S < \rho \text{ and } (\theta_T - \gamma_i) \in \left[\frac{-\pi}{2}, \frac{-3\pi}{2}\right] \end{cases}$$

(40)

Robots which are in the neighborhood of their targets then have

$$|\dot{\theta}_{Sat}| = |\dot{\theta}_T|$$

The permitted dynamic of these targets can then be easily deduced as (cf. Equation 36)

$$|\dot{\theta}_T| \leq \omega_{max} - k\pi$$

(41)

Let us study now the case of robots which are not in the neighborhood of their targets. Beforehand, we recall the following properties:

1. $\gamma_i$ is constant $\Rightarrow \dot{\gamma}_i = 0$,
2. $b = \frac{v_T}{v_i} \Rightarrow \dot{b} = \frac{v_i}{v_T} (\frac{\dot{v}_T v_i - v_T \dot{v}_i}{v_T})$,
3. $|a \cos(\alpha) + b \sin(\alpha)| \leq \sqrt{a^2 + b^2} \hspace{1cm} \forall a, b, \alpha \in R$,
4. $d_S$ is bounded (cf. Equation 7).

By replacing equation (40) when $d_S \geq \rho$ in (36), and using points 1) and 2), we obtain

$$b \sin(\theta_T - \gamma_i) + b\dot{\theta}_T \cos(\theta_T - \gamma_i) \leq \sqrt{(\omega_{max} - k\pi)^2 - (b \sin(\theta_T - \gamma_i))^2}$$

(42)

To find the permitted variation of target $\dot{\theta}_T$, we use the upper bound of the left member of relation (42) according to point 3). We then have

$$\sqrt{(b)^2 + (b\dot{\theta}_T)^2} \leq (\omega_{max} - k\pi) \sqrt{1 - (b \sin(\theta_T - \gamma_i))^2}$$

(43)

In fact, the values of $\dot{\theta}_T$, verifying the relation (43), verify also (42). $\dot{\theta}_T$ can then be expressed as

$$(\dot{\theta}_T)^2 \leq \left[ (\omega_{max} - k\pi)^2 (1 - (b \sin(\theta_T - \gamma_i))^2) - (b)^2 \right] / b^2$$

(44)

To remain attainable, the target must have an angular velocity $\dot{\theta}_T$ verifying relation (44) for all the robots, in other words for all possible linear velocities. Thus, it has to be

$$(\dot{\theta}_T)^2 \leq \min\left( (\omega_{max} - k\pi)^2 (1 - (b \sin(\theta_T - \gamma_i))^2) - (b)^2 \right) / b^2$$

(45)

A necessary condition to reach the minimum of the right member, noted (R), of this relation is that $\frac{1}{b} \rightarrow 1$, since $b \leq 1$ (cf. Equation 21). However this condition occurs only when $d_S \approx 0$. The latter case $d_S < \rho$ has already been addressed in equation (41).

Since $v_i$ decreases as the robot approaches its target (cf. Equation 17a), this condition occurs as $d_S \approx \rho$.

First, let us calculate $\dot{v}_i$ using relation (17a)

$$\dot{v}_i = v_T e^{-\frac{a^2}{2\sigma^2}} - \frac{2d_S \dot{d}_S}{\sigma^2} e^{-\frac{a^2}{2\sigma^2}}$$

(46)

If $\rho$ is considered as being sufficiently small, and $\dot{d}_S$ is bounded (point 4)), relation (46) becomes

$$\dot{v}_i \approx \dot{v}_T$$

In the neighborhood of $\rho$, relation (44) becomes

$$(\dot{\theta}_T)^2 \leq \left[ (\omega_{max} - k\pi)^2 (1 - (v_T \sin^2(\theta_T - \gamma_i))) \right] \frac{v_i(\rho)}{v_T}$$

(47)

Note that we can also have

$$1 - \left[ (\frac{v_T}{v_i(\rho)} \sin^2(\theta_T - \gamma_i)) \right] = 0$$

(48)

Hence, $R = 0$.

To remain attainable, the dynamic of the virtual structure has to follow two phases:

1. a transitional phase, where the robots have not yet achieved the formation. In this phase, $\dot{\theta}_T$ is constrained such that $\dot{\theta}_T = 0$.
2. once the formation is achieved, the virtual structure can vary according to relation (41).
4. Simulation and Experimental Results

First, the behavior of a robot with respect to its dynamic target motion is observed through simulation. Next, experimentation is made with three robots to attain and to maintain a triangular formation.

4.1. Variation of the virtual structure angular velocity

This section shows the importance of bounding the angular velocity of the virtual structure $\dot{\theta}_T$ according to the kinematic constraints of the robots. Hence, a mobile robot reaching a virtual target is simulated. The maximum angular velocity of the robot is $\omega_{\text{max}} = 3\text{rd/s}$. We choose $k = 0.6s^{-1}$.

According to relation (41), and to simplify notation on figures, we propose to note $P = \omega_{\text{max}} - k\pi$. Based on the chosen values of $\omega_{\text{max}}$ and $k$, we find $P = 1.1$. First, it is proposed to show the importance of the transitional phase where the variation of $\dot{\theta}_T$ must be set to 0 (cf. Equation 47). Hence, in figure 8.(b), we can see that $\dot{\theta}_T$ increases at the beginning of the simulation (from 0.1s) and the target trajectory follows immediately a significant curve (cf. Figure 8.(a)). Consequently, we observe oscillations in the trajectory of the robot. The robot correctly attains the target only when this one has a straight trajectory ($\dot{\theta}_T = 0$). Figure 8.(b) confirms that. In fact, even if $\dot{\theta}_T$ satisfies the condition described in equation 44, oscillations may appear if the transitional phase is not imposed. Naturally, the distance $d_S$ is oscillating in this case (cf. Figure 8.(c)).

The Lyapunov function is also oscillating and the control law is not stable (cf. Figure 8.(d)).

Figure 9 shows the importance of satisfying the condition described in relation 41 after the transitional phase. Once the target is attained ($\dot{\theta}_T = 0$ until the moment 0.5s), the condition 41 is also satisfied. It can be seen that the robot goes toward the target. Even if it increases, the variation of $P$ is such that $\dot{\theta}_T < P$ (cf. Figure 9.(b)). In this interval, the robot correctly tracks its target (cf. Figure 9.(a)). The distance $d_S$, separating them is $d_{S_{k}} = 0$ (cf. Figure 9.(c)). The Lyapunov function also decreases and then remains equal to 0 (cf. Figure 9.(d)). After 0.5s, we remove the constraint (41) such that $\dot{\theta}_T$ can be $\dot{\theta}_T > P$. It can be seen that the robot cannot track the target. The oscillation of distance $d_S$ and $V$ confirms this (cf. Figures 9.(c) and (d)).

4.2. Experimental results: 3 robots with a attainable virtual structure

Experiments were performed using Khepera III robots (cf. Figure 10). For the first tests, only the perception of the MRS was centralized. Hence, navigation was achieved on a platform equipped with a camera giving positions and orientations of the robots by detecting the bar code associated with each one (cf. Figure 10). This information was sent to the robots by a computer through a Wi-Fi network.

In [26], the virtual structure has a straight trajectory. Here, it is proposed to extend to circular motion such that all the targets remain attainable by all the robots despite their kinematic constraints. Knowing that the dynamic of the virtual structure has to follow relation (41), the radius $R_{v_s}$ of the circular motion formed by the main target $T_1$ (cf. Figure 11(a)) verify

$$R_{v_s} = \frac{v_T}{\dot{\theta}_T} > \frac{v_T}{\omega_{\text{max}} - k\pi}$$

(48)

with $v_T$ constant and $v_T \ll v_{\text{max}}$.

First, a clockwise motion is considered (cf. Figure 11(a)). It is observed that the robots converge to the virtual structure even without passing the transitional phase. The reason is that $R_{v_s}$ is big enough and initial conditions
of the robots are far from critical situations described in section 3.4.2. In fact, equation 47 imposes constraints for even the maximal orientation errors corresponding to $\pi$ (cf. Equation 36). Here, initial positions of the robots do not correspond to this critical configuration and the highest authorized bound is higher than the one given in equation 47. At time $t_2 + \Delta t$, a jump in the virtual structure state was produced (cf. Figure 11(b)). The dynamic of the virtual structure was also changed so that its motion became counter-clockwise. Note that robots change dynamically their targets at each set-points jump to rapidly reach the new formation using RCC algorithm [27]. The distances between the robots and their targets are given in figure 12. They decreased to 0, which confirms that the formation was reached and maintained. When the virtual structure dynamic was changed, the robots were far from their targets, which explains the observed jumps. The same observations were noticed on the global Lyapunov function (cf. Figure 13).


[24] P. Ogren, E. Fiorelli, L. N. E., Formations with a mission: Sta-


27. A. Benzerrouk, L. Adouane, P. Martinet, Altruistic distributed target allocation for stable navigation in formation of multi-robot system, 10th International IFAC Symposium on Robot Control.


Biography

Ahmed Benzerrouk received his Ph.D. in Robotics and Vision from Blaise Pascal University, France in 2011. He was a Postdoctoral Associate in Institut Pascal. Currently, he is a teacher fellow at Institut Francais de Mecanique Avancee. His research interests include: Mobile robotics control, Cooperative robotics, Lyapunov-based stability.

Lounis Adouane received his Master of sciences in 2001 from IRCCyN - ECN Nantes (France), where he worked on the control of legged mobile robotics. In 2005 he obtained the Ph.D. in Automatic Control from LAB - UFC Besanon. During his PhD Lounis Adouane has deeply investigated the field of multi-robot systems, especially those relying to reactive control architectures. After that, he joined in 2006 LAI - INSA Lyon and he studied the hybrid architecture of control applied to cooperative mobile arms robots. Since 2006, he is an Associate Professor at Institut Pascal - Polytech Clermont-Ferrand. His research interests include: Mobile robotics control, Cooperative robotics, Artificial intelligence, Behavioral/Hybrid control architectures, Multi-robot simulation.

Philipphe Martinet graduated from the CUST, Clermont-Ferrand, France, in 1985 and received the Ph.D. degree in electronics science from the Blaise Pascal University, Clermont-Ferrand, France, in 1987. From 1990 until 2000, he was an assistant Professor with CUST in the Electrical Engineering Department, Clermont-Ferrand. From 2000 until 2011, he has been a Professor with Institut Francais de Macronique Avance (IFMA), Clermont-Ferrand. He was performing research at the Robotics and Vision Group of Institut Pascal - CNRS, Clermont-Ferrand. In 2006, he spent one year as a visiting professor in ISRC at the Sungkyunkwan university in Suwon, South Korea. He was the leader of the group GRAVIR (over 74 person) from 2001 til 2006. From 1997 until 2011, he led the Robotic and Autonomous Complex System team (over 20 persons). From 2008 until 2011, he co-lead a Joint Unit of Technology called "Robo- tization in meet Industry", and the Korea France Joint Research Center on Cognitive Personal Transport Service Robot in Suwon (South Korea). Since September 2011, he moves to Ecole Centrale de Nantes and make his research at IRCCYN in the robotics team. He stays associated researcher to Pascal Institute in Clermont-Ferrand. His research interests include Visual Servoing of Robot Visual Servoing, Autonomous Guided Vehicle Control, Modeling/Identification and Control of complex MACHines, Redundancy Control and Autonomy of Humanoid, Active Vision and Sensor Integration, Visual Tracking, and Parallel Architecture for Visual Servoing Applications. From 1990, he is author and co-authors of more than two hundred thirty references.