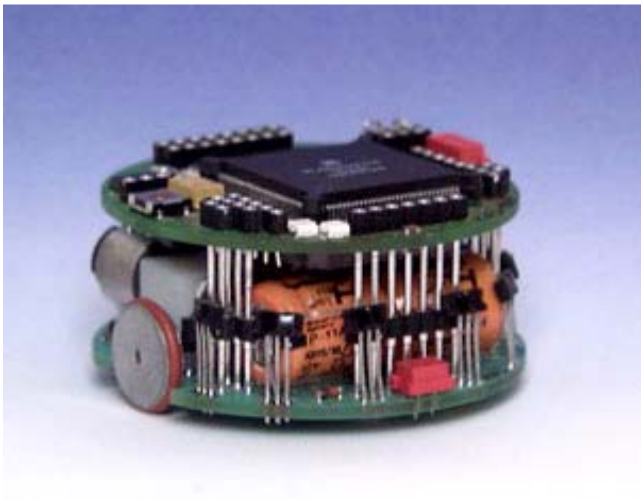
Master Project

Collective motion stabilization and recovery strategies



Author: Piyawat Kaewkerd (Microengineering, EPFL)

Assistant: Dr. Philippe Müllhaupt (EPFL), Christopher Peek (UF)

Professors: MER. Denis Gillet (EPFL), Prof. Oscar D. Crisalle (UF)

Period: October 2006- February 2007

Abstract: Collective motion handling is becoming more and more important in congested urban developed regions. Critical issues appear when failures and accidents take place. In that context, the objectives of this project are the following:

a) The first objective of this project is to adapt a recently developed controller for a single nonholonomic robot to collective motion stabilization problems (flocking). The solutions should be implemented on a simulation environment designed so as to test their performances on a virtual platoon of vehicles.

b) The second phase of the project is to elaborate a recovery strategy in case a failure occurs in one of the vehicles (for instance, steering failure, crash, etc.). A study of the state of the art along this topic will be addressed.

Preface

The master project presented in this report has been accomplished at University of Florida, Gainesville, during a one-semester exchange program. The work has been performed under the supervision of Prof. O.D. Crisalle (Chemical Engineering Department, University of Florida) and MER D. Gillet (Automatic Control Laboratory, EPFL), and with the assistance of Dr. P. Müllhaupt (EPFL) and C. Peek (University of Florida).

Table of contents

1 Introduction

Swarm intelligence has been a growing area of research these past few years due to its many possible applications, such as for victim search and rescue or in the military domain. The computing and communication capabilities newly available to mobile robots further increased interests of the scientific community on this subject.

This paper treats two basic problems in swarm intelligence. First, the flocking problem, which refers to having multiple vehicles following a trajectory in formation, is addressed by presenting several collective motion stabilization techniques, based on a novel one-vehicle controller [1]. Secondly, in order to make flocking control more robust, recovery strategies are investigated and tested. Indeed, in case a vehicle within the formation does not follow its intended trajectory anymore, due to mechanical failures for instance, or if an obstacle appears, the group has to have a strategy to avoid collision and recover the formation once there are no threats anymore.

Most of the solutions proposed by researchers for flocking control deal with the collision avoidance and formation recovery problems in an indirect manner. For instance, the methodology of [2] is based on potential functions, as commonly used in mobile robotics for obstacle avoidance problems: the vehicle is attracted to its goal, in this case a circle around a virtual leader, while repulsed by obstacles and other vehicles. By adding these virtual forces, the desired direction the vehicle should head in is computed. By moving the virtual leader in a slow enough motion, the platoon can avoid collision and go between obstacles while keeping its overall structure.

The drawbacks with this technique are that the vehicles do not use a control law that guarantees convergence to the desired position. Instead, they only compute an optimal direction they should head in, without taking into account the dynamics of the vehicles. In order to ensure that obstacles can be actually avoided, a conservative approach is taken by defining a safe zone around each vehicle. The radius of the zone is so that it is large enough for the vehicles to stop without actually touching any obstacles.

In [3], a similar approach is presented, with more insight from the automatic control point of view: the vehicles have a control law that includes a component that makes them head in the same direction as their neighbors, and another component that makes them steer away if they get too close to another particle, or steer towards each other if they are too far apart. The separation between the particles is defined by a parameter.

The collision avoidance techniques presented in this paper are inspired by these solutions, as they are based on artificial potential attraction and repulsion. Nevertheless, instead of having a global approach, flocking control and collision avoidance are treated separately by switching controllers depending on the situation. Platoon of vehicles were simulated with the resulting control laws.

2 Single-vehicle controller

Before the collective motion results and the recovery strategies are shown, the important theories and equations used throughout our work are presented and explained. The vehicle dynamics, for instance, are used to simulate the vehicles trajectories. The flatness-based control law, developed in [1], is the basis of the work on collective motion.

2.1 Vehicle dynamics

We used the following state-space model for the vehicles dynamics (which corresponds to a nonholonomic two-wheel mobile robot or "unicycle", Figure 1):

$$\begin{aligned} \dot{x}_1 &= v_1 \cos(x_3) \\ \dot{x}_2 &= v_1 \sin(x_3) \\ \dot{x}_3 &= v_2 \end{aligned} \tag{1}$$

with the inputs (also called control variables or commands) v_1 and v_2, which represent respectively the velocity and the steering. x_1 and x_2 correspond to the two coordinates of the position of the vehicle. x_3 is the angle between the x_1-axis and the heading direction of the vehicle. In our simulations, vehicles are supposed to be round, and their diameter d is the only variable needed to define their physical dimension.

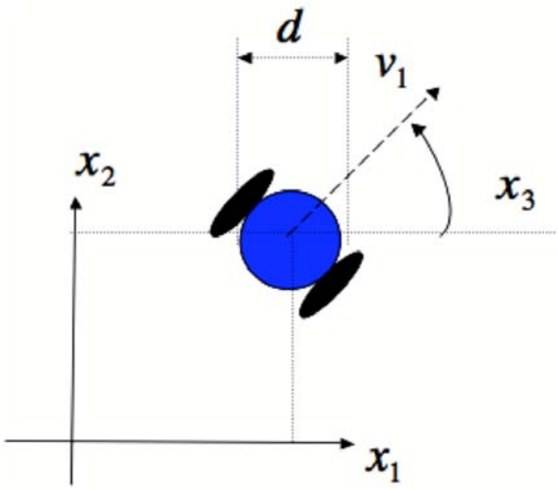


Figure 1: x_1, x_2 and x_3 are the three states of the vehicle model, and v_1 is the velocity control input. v_2, not represented here, is the first derivative of x_3. The diameter of the vehicle is d.

2.2 Flatness-based control law

In this subsection, the controller described in [1] is presented. The control law developed in this paper allows stabilization and trajectory tracking of a vehicle described by the aforementioned dynamics. It is based on the flatness property, and its performances are better than those of the classical dynamic feedback, at the expense of a second-order state extension.

2.2.1 Flatness

A system $\dot{x} = f(x,u)$ with $u \in \Re^m$ and $x \in \Re^n$ is flat if there exists an output $y \in \Re^m$ such that:

$$\begin{aligned} x &= F(y,...,y^{(p-1)}) \\ u &= P(y,...,y^{(p)}), p \in N \end{aligned} \tag{2}$$

A mobile robot that follows the dynamics in (1) is a flat system. Indeed, with x_1 and x_2 as the flat outputs, i.e. $y = (x_1, x_2)^T$, x_3 and the inputs v_1 and v_2 can be expressed as:

$$\begin{aligned} x_3 &= \arctan2(\dot{x}_1, \dot{x}_2) \\ v_1 &= \sqrt{\dot{x}_1^{\,2} + \dot{x}_2^{\,2}} \\ v_2 &= \frac{-\dot{x}_2\ddot{x}_1 + \dot{x}_1\ddot{x}_2}{\dot{x}_1^{\,2} + \dot{x}_2^{\,2}} \end{aligned} \tag{3}$$

2.2.2 Controller for stabilization

Using the flatness property, it is possible to have a correspondence between a nonlinear system and a linear equivalent one. As for dynamic feedback linearization, it requires a state extension. The novel idea of [1] is to have a second-order extension instead of a single state extension, which offers better disturbance rejection performances, and also allows the control law to be more intuitive.

In the case of the stabilizing controller, the two additional states χ_1 and χ_2 provide the ideal velocity vector for the vehicle to head to the origin ($x_1 = 0$ and $x_2 = 0$). The scheduler updates that ideal velocity vector, according to the current position of the vehicle.

$$\begin{aligned} \dot{\chi}_1 &= k_1 x_1 + k_2 \chi_1 \\ \dot{\chi}_2 &= k_3 x_2 + k_4 \chi_2 \end{aligned} \tag{4}$$

The velocity input v_1 is the ideal velocity according to χ_1 and χ_2, and the steering input is the output of a simple proportional regulator that makes x_3 converge to the ideal angle of $\arctan2(\chi_1, \chi_2)$. These relationships clearly come from the equations that stem from the flatness property (3).

$$\begin{aligned} v_1 &= \sqrt{\chi_1^{\,2} + \chi_2^{\,2}} \\ v_2 &= \frac{-\chi_2\dot{\chi}_1 + \chi_1\dot{\chi}_2}{\chi_1^{\,2} + \chi_2^{\,2}} - k_p(x_3 - \arctan2(\chi_1, \chi_2)) \end{aligned} \tag{5}$$

2.2.3 Controller for trajectory tracking

Adapting the controller for trajectory tracking is straightforward. The control inputs are unchanged, only the two additional states update equations are different. This can be explained by the fact that the ideal velocity vector is not directed to the origin anymore, but to the current point of the trajectory the vehicle needs to track.

x_{1ref} and x_{2ref} are function of time and are respectively the x_1 and the x_2 coordinate of the reference trajectory. The proportional gains k_1 and k_3 in the differential equations of the two additional states (6) make the actual position of the vehicle converge to the current point of the reference trajectory, whereas the gains k_2 and k_4 make the ideal velocity vector converge to the desired velocity vector (tangent to the reference trajectory). If the vehicle follows the reference trajectory perfectly, its acceleration should match the acceleration of the trajectory, represented by the second-order derivatives $\ddot{x}_{1ref}$ and $\ddot{x}_{2ref}$.

$$\begin{aligned} \dot{\chi}_1 &= k_1(x_1 - x_{1ref}) + k_2(\chi_1 - \dot{x}_{1ref}) + \ddot{x}_{1ref} \\ \dot{\chi}_2 &= k_3(x_2 - x_{2ref}) + k_4(\chi_2 - \dot{x}_{2ref}) + \ddot{x}_{2ref} \end{aligned} \tag{6}$$

It is easily verifiable that the controller for stabilization (4) is a special case of the controller for trajectory tracking, with $x_{1ref} = x_{2ref} = \dot{x}_{1ref} = \dot{x}_{2ref} = \ddot{x}_{1ref} = \ddot{x}_{2ref} = 0$.

3 Collective motion stabilization

The control law presented above has been proven to provide asymptotic convergence to a given trajectory (or position, in the case of stabilization) for a single vehicle. The next step was to generalize it for collective motion problems. Some changes have to be made to adapt the control law to allow multiple vehicles to flock in a given formation.

The multi-agent flocking problem consists in having several vehicles stay in a given formation and follow a given path, using similar control laws for each of the agents. The formation and reference trajectory information can be for instance broadcast wirelessly to all the vehicles by a central unit. Robustness against perturbations such as obstacles and failure of one of the agents is an area of undergoing research and will be addressed in section 4.

The formations can either rotate to follow a leader and trail it, or be invariant with respect to the heading direction. In the first case, the vehicles will need to follow different trajectories (with different radii of curvature) to track their respective spots behind the leader. In the automatic control point of view, this is problematic, as only the leader's reference trajectory should be calculated. In the second case, all the vehicles will have identical trajectories, but shifted in space, which facilitates the stabilization problem. However, it is less practical, because the formation may need to be rotated in order to go through a narrow area, for instance.

In the case where the formation rotates behind the leader, there are two ways to solve this problem using the control law described in 2.2. One possible solution is to implement the same control law to every agent, but with a different reference trajectory for every one of them. This solution is clearly not adequate for several reasons. First, there would be as many reference trajectories to define and to compute as there are vehicles, which could represent a significant burden for the central unit, especially as the first two derivatives of each trajectory have to be computed. Furthermore, a complex path-planning regeneration would be necessary for the resulting system to be able to handle obstacles. Hence, this approach will not be discussed below.

The second solution is to have a leader that tracks the desired reference trajectory, and to have the other vehicles follow it with a predefined offset in space. Each vehicle has a different offset, which defines the overall formation. The leader does not need to be real, as a virtual leader can be used as well depending on the application.

In the following subsections, the different solutions for collective motion stabilization, or flocking control, are presented, either for rotating and non-rotating formation. The simulations were all done with the same control gains $k_1 = k_3 = -100$, $k_2 = k_4 = -20$ and $k_p = 2$.

3.1 Formation characterization

In order to define a flocking formation, each vehicle is assigned with a pair of offsets $\delta_{i,1}$ and $\delta_{i,2}$ that determines the position it should keep with respect to the leader (Figure 2).

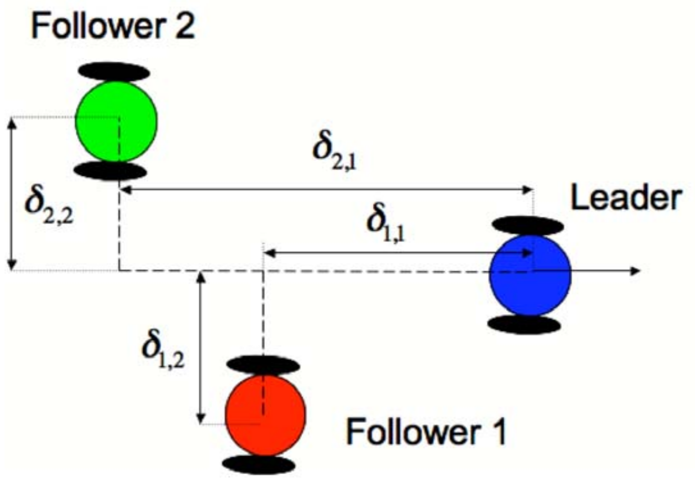


Figure 2: Every follower has a different pair of offsets that defines its position in the formation

The offsets are either defined in the world reference frame, in the case of non-rotating formations, or in the leader's reference frame, in the case of rotating ones, as explained in sections 3.2 and 3.3.

3.2 Non-rotating formation with estimation of the leader's trajectory

The control law of the leader is the same as for the one-vehicle case and is defined by the equations (5) and (6). The path that the platoon is supposed to follow is its reference trajectory.

The control law of the followers (7) is similar, except for the reference trajectory: instead of a predefined trajectory function, the reference trajectory is updated at every time step according to the current position of the leader. Thus, x_{1ref} and x_{2ref} are the coordinates of the leader plus a specified distance or offset, δ_1 and δ_2, respectively, so that each vehicle's target is not the leader itself, but a point in its vicinity (Figure 3). The first and second time-derivatives of x_{1ref} and x_{2ref} used in the control law are estimated using a backward Euler approximation, as shown in equations (8) and (9).

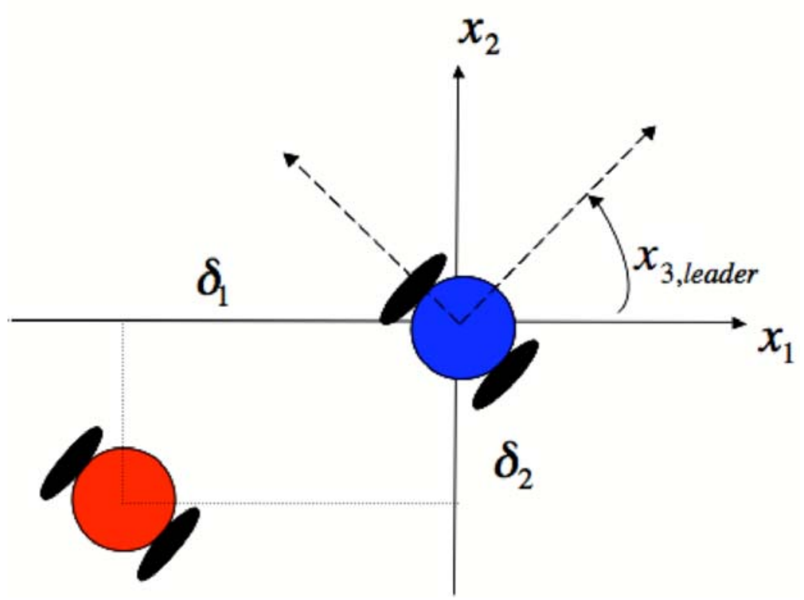


Figure 3: The offset of the follower (red) is defined with respect to the leader (blue), but in the world reference frame. Thus, the formation does not rotate and is invariant to $x_{3,leader}$.

$$\begin{aligned}\dot{\chi}_1 &= k_1(x_1 - x_{1,leader} - \delta_1) + k_2(\chi_1 - \hat{\dot{x}}_{1,leader}) + \hat{\ddot{x}}_{1,leader} \\ \dot{\chi}_2 &= k_3(x_2 - x_{2,leader} - \delta_2) + k_4(\chi_2 - \hat{\dot{x}}_{2,leader}) + \hat{\ddot{x}}_{2,leader}\end{aligned} \tag{7}$$

$$\begin{aligned}\hat{\dot{x}}_{1,leader}(t) &= \frac{x_{1,leader}(t) - x_{1,leader}(t-h)}{h} \\ \hat{\dot{x}}_{2,leader}(t) &= \frac{x_{2,leader}(t) - x_{2,leader}(t-h)}{h}\end{aligned} \tag{8}$$

$$\begin{aligned}\hat{\ddot{x}}_{1,leader}(t) &= \frac{\hat{\dot{x}}_{1,leader}(t) - \hat{\dot{x}}_{1,leader}(t-h)}{h} \\ \hat{\ddot{x}}_{2,leader}(t) &= \frac{\hat{\dot{x}}_{2,leader}(t) - \hat{\dot{x}}_{2,leader}(t-h)}{h}\end{aligned} \tag{9}$$

With (7), the offsets are defined in the world reference frame, not in the leader's one. If observed from a global way, the formation will thus only move by translation, and will not rotate according to the heading direction of the leader (Figure 4 and Figure 5).

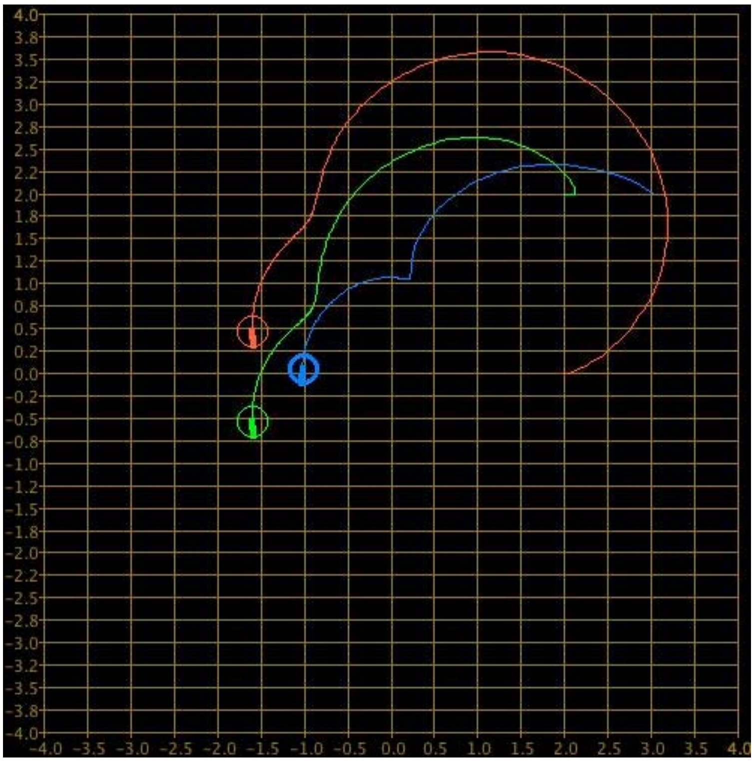


Figure 4: The lines show the trajectories of each of the vehicles, which are represented by a circle and a bold line that indicates their heading direction. The leader (blue) tracks a circular reference trajectory, and the other vehicles follow it with a fixed offset.

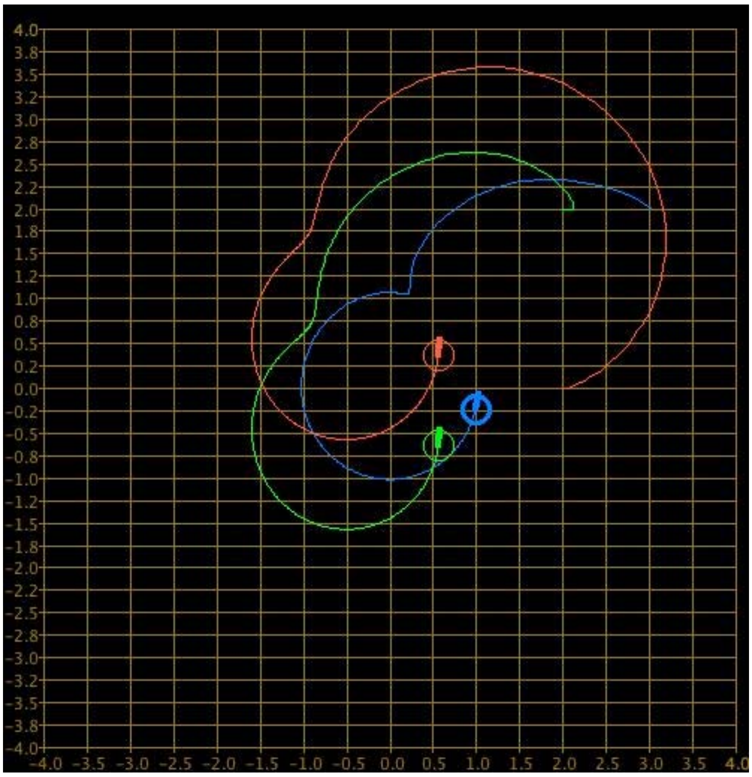


Figure 5: The positions of the red and green vehicles are not constant in the blue vehicle's reference frame. Their trajectories have the same radius of curvature.

3.3 Rotating formation with estimation of the leader's trajectory

The behavior presented above may not be desirable, especially if the formation is asymmetric, as the width of the formation will differ depending on the heading direction of the leader, viewed from the vehicles' perspective, which can be troublesome if the platoon has to go between two obstacles. In order to have the vehicles actually follow the leader, a change has to be made to the followers' control law. Instead of having the offsets δ_1 and δ_2 defined in the x_1 and x_2 coordinates, they are defined in the leader's reference frame. The offsets are then projected on the inertial reference frame (Figure 6).

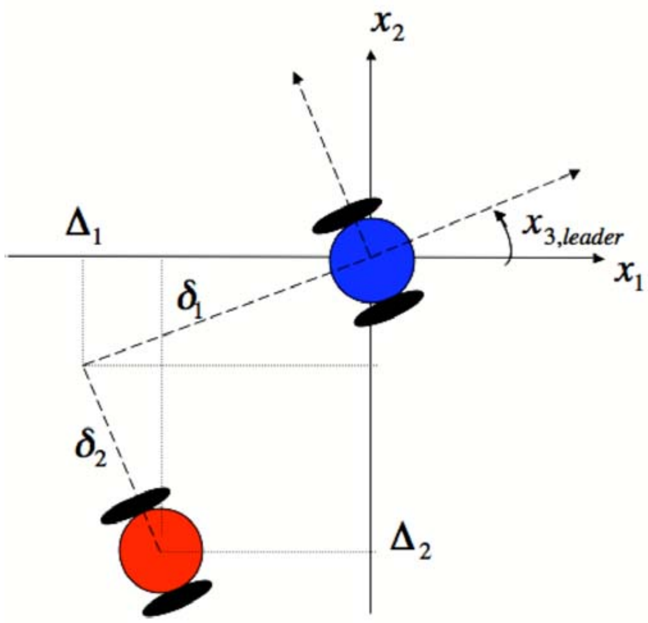


Figure 6: The offset of the follower (red) is defined in the leader's (blue) reference frame, which is rotated by $x_{3,leader}$ with respect to the inertial reference frame. For the control law, these offsets need to be converted (projected) back to the inertial reference frame.

The update equations of the additional states are thus similar to (7), except that the offsets $\Delta_1(t)$ and $\Delta_2(t)$ are not constant anymore, as, in the inertial reference frame, they change depending on the heading angle of the leader $x_{3,leader}$, as shown in the equations (10) and (11). The velocity and acceleration of the leader are estimated in the same manner, using equations (8) and (9).

$$\begin{aligned} \dot{\chi}_1 &= k_1[x_1 - (x_{1,leader} + \Delta_1(t))] + k_2(\chi_1 - \hat{\dot{x}}_{1,leader}) + \hat{\ddot{x}}_{1,leader} \\ \dot{\chi}_2 &= k_3[x_2 - (x_{2,leader} + \Delta_2(t))] + k_4(\chi_2 - \hat{\dot{x}}_{2,leader}) + \hat{\ddot{x}}_{2,leader} \end{aligned} \tag{10}$$

$$\begin{aligned} \Delta_1(t) &= \delta_1 \cdot \cos(x_{3,leader}) - \delta_2 \cdot \sin(x_{3,leader}) \\ \Delta_2(t) &= \delta_2 \cdot \cos(x_{3,leader}) + \delta_1 \cdot \sin(x_{3,leader}) \end{aligned} \tag{11}$$

The drawback with this method is that the leader's trajectory first and second derivatives estimates $\hat{\dot{x}}_{i,leader}$ and $\hat{\ddot{x}}_{i,leader}$ are used in the control law of the followers, even though their trajectories are different. Indeed, the curvatures of their trajectories, and thus the first and second derivatives of their trajectories, vary depending on their offsets (Figure 7 and Figure 8) because of the fact that the formation rotates with the leader, and the control law is therefore inaccurate.

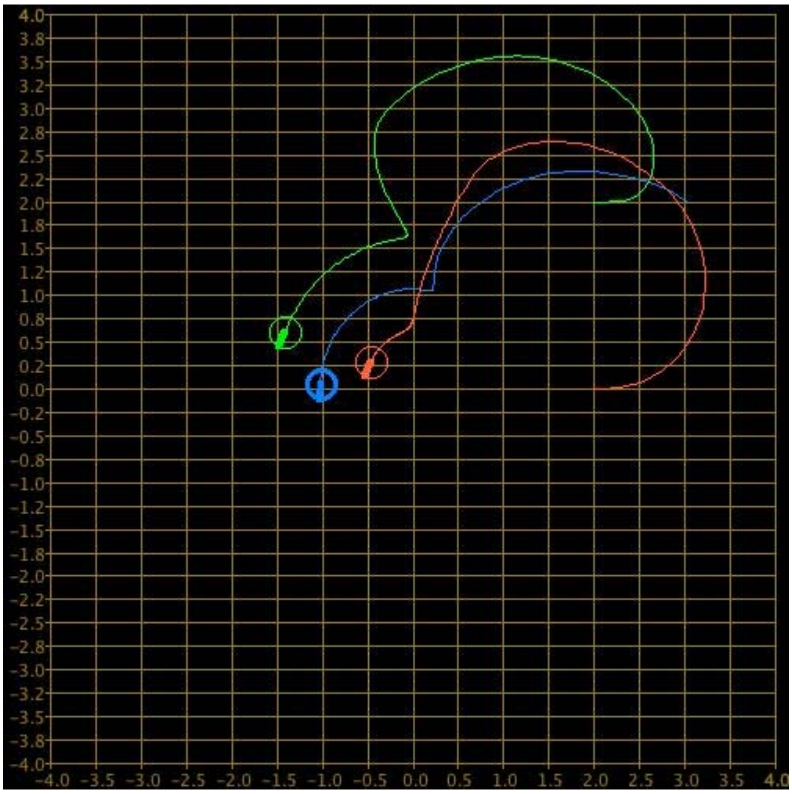


Figure 7: The red and green vehicles follow the leader (blue), which track a circular reference trajectory.

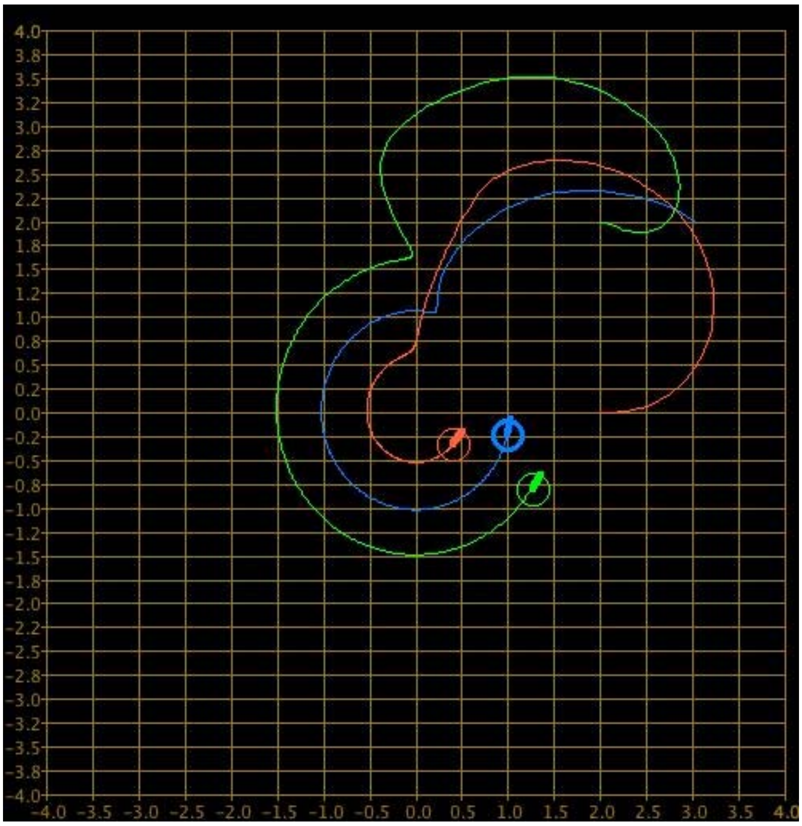


Figure 8: The red and green vehicles stay behind the leader (blue), and their positions remain constant in the leader's reference frame.

This inaccuracy leads to positioning errors, as it can be seen by considering (10) in the steady-state case (12).

$$
\begin{aligned}
Err_1 &= -\frac{1}{k_1}[k_2(\dot{x}_{1,traj} - \hat{\dot{x}}_{1,leader}) + \hat{\ddot{x}}_{1,leader} - \ddot{x}_{1,traj}] \\
Err_2 &= -\frac{1}{k_3}[k_4(\dot{x}_{2,traj} - \hat{\dot{x}}_{2,leader}) + \hat{\ddot{x}}_{2,leader} - \ddot{x}_{2,traj}]
\end{aligned} \tag{12}
$$

Err_1 and Err_2 are the positioning errors along x_1 and x_2, and $\dot{x}_{1,traj}$, $\dot{x}_{2,traj}$, $\ddot{x}_{1,traj}$ and $\ddot{x}_{2,traj}$ are the first and second derivatives of the follower's actual trajectory. Thus, beside the control parameters, the magnitude of the error depends on the difference between the actual trajectory of the vehicle and the leader's trajectory. It can be made more acceptable if this difference is reduced, by decreasing the size of the platoon (or more exactly the distance between the followers and the leader) and by making the leader follow a trajectory that has bigger radii of curvature.

3.4 Control law using the leader's reference trajectory

Instead of estimating the leader's velocity and acceleration (using backward Euler approximations), the vehicles that follow the leader can also use the same reference trajectory. Indeed, with the hypothesis that the leader is correctly following the intended trajectory, the reference trajectory will be close enough from the actual trajectory of the leader, and thus, the behavior of the vehicles should be similar to what is described in sections 3.2 and 3.3. This has the advantage of not having to estimate the leader's motion at every time step, which provides smoother control input. Furthermore, if the time steps used for estimating the derivatives are too large, the vehicles will have trouble keeping track of the leader because of the difference between the actual and the estimated values. This problem will be less critical if all the vehicles know the reference trajectory, as it can be stored in their memories and need not be computed at every time step.

It is possible for all the vehicles to have the same control law with the same reference trajectory, except for the position, which is modified by a different offset for each vehicle (13). With this kind of update equations, an actual leader is not needed anymore, as the reference trajectory can be considered as a virtual leader. The offsets can either be constant, as in section 3.2, or projected to the virtual leader's reference frame (14), to have results similar to section 3.3. In this case, the angle that the offsets need to be projected on is found by computing the tangent to the trajectory, which is the reference velocity vector. A simple inverse tangent operation converts the vector coordinates to the angle ϑ (15).

$$
\begin{aligned}
\dot{\chi}_1 &= k_1[x_1 - (x_{1ref} + \Delta_1(t))] + k_2(\chi_1 - \dot{x}_{1ref}) + \ddot{x}_{1ref} \\
\dot{\chi}_2 &= k_3[x_2 - (x_{2ref} + \Delta_2(t))] + k_4(\chi_2 - \dot{x}_{2ref}) + \ddot{x}_{2ref}
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
\Delta_1(t) &= \delta_1 \cdot \cos(\vartheta) - \delta_2 \cdot \sin(\vartheta) \\
\Delta_2(t) &= \delta_2 \cdot \cos(\vartheta) + \delta_1 \cdot \sin(\vartheta)
\end{aligned} \tag{14}
$$

$$
\vartheta = a\tan 2(\dot{x}_{2ref}, \dot{x}_{1ref}) \tag{15}
$$

As shown before, having the offsets defined in the leader's reference frame yields different trajectories with different first and second derivatives for each of the vehicles. This is not an issue if the size of the platoon is not too large compared to the curvature of the reference trajectory. However, if the platoon becomes too large, the reference trajectory will not be close to the actual trajectory of the vehicles, especially the ones on the outer edge of the formation, and the difference in the trajectory parameters can lead to tracking inaccuracies.

This method is not reliable for applications in which the vehicles have to follow an actual leader. Indeed, the leader's position or trajectory is not in the control law, and the other vehicles will not follow it if it drifts away from the reference trajectory because of some external disturbances. To some extent, it is possible to make them follow the leader without estimating its velocity and acceleration, but some inaccuracies will appear if the leader's trajectory differs too much from the reference trajectory. Hence, with (16), the vehicles will follow the leader more robustly than with (13), as long as the leader's trajectory is nearly parallel to the reference trajectory.

$$\begin{aligned} \dot{\chi}_1 &= k_1[x_1 - (x_{1,leader} + \Delta_1(t))] + k_2(\chi_1 - \dot{x}_{1ref}) + \ddot{x}_{1ref} \\ \dot{\chi}_2 &= k_3[x_2 - (x_{2,leader} + \Delta_2(t))] + k_4(\chi_2 - \dot{x}_{2ref}) + \ddot{x}_{2ref} \end{aligned} \tag{16}$$

However, if the leader totally drifts away from the reference trajectory, the vehicles will follow it less accurately because of the information that do not match (Figure 9). If the mismatch is too large, collisions can occur among the formation, due to the fact that the vehicles will no longer converge to their predestinated spots.

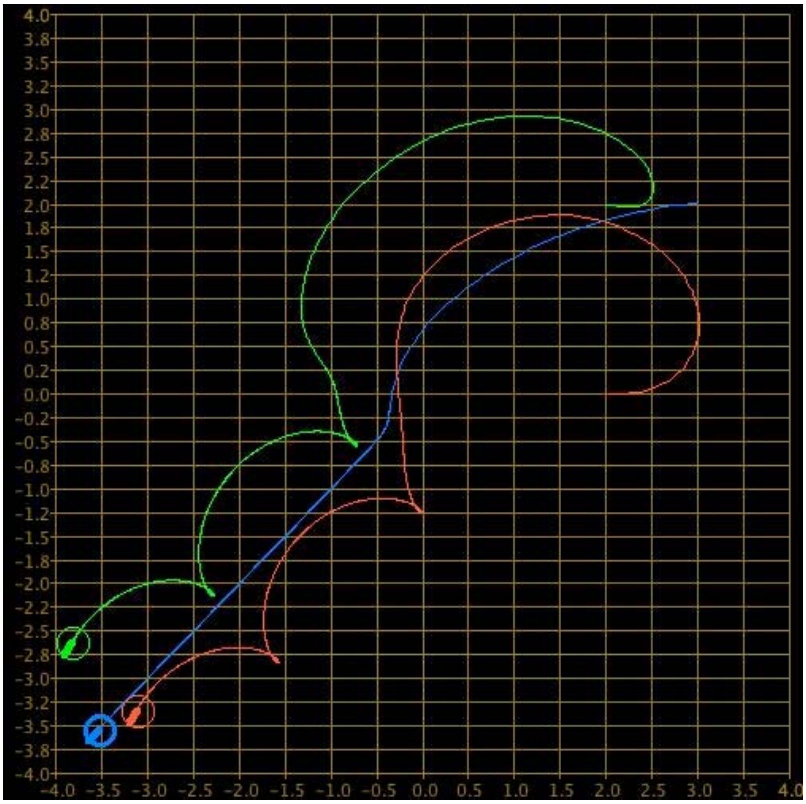


Figure 9: The green and red vehicles do not follow the leader (blue) accurately when it stops tracking the reference trajectory. The cyclic behaviors are due to the circular reference trajectory.

3.5 Rotating formation with estimation of every vehicle's trajectory

A more accurate but also more computing intensive solution is for each vehicle to estimate its own desired trajectory, instead of the leader's trajectory. With this solution, every vehicle of a platoon of any size will have information about its own trajectory, and will thus track more accurately its spot in the formation. To do that, the estimation should not be done on the leader's velocity and acceleration, but on the vehicle's predestined position in the formation instead. Hence, the control law is similar to the one shown in section 3.3, with a difference in the backward Euler estimation equations:

$$\begin{aligned}\dot{\chi}_1 &= k_1[x_1 - (x_{1,leader} + \Delta_1(t))] + k_2(\chi_1 - \hat{\dot{x}}_1) + \hat{\ddot{x}}_1 \\ \dot{\chi}_2 &= k_3[x_2 - (x_{2,leader} + \Delta_2(t))] + k_4(\chi_2 - \hat{\dot{x}}_2) + \hat{\ddot{x}}_2\end{aligned} \tag{17}$$

$$\begin{aligned}\hat{\dot{x}}_1(t) &= \frac{[x_{1,leader}(t) + \Delta_1(t)] - [x_{1,leader}(t-h) + \Delta_1(t-h)]}{h} \\ \hat{\dot{x}}_2(t) &= \frac{[x_{2,leader}(t) + \Delta_2(t)] - [x_{2,leader}(t-h) + \Delta_2(t-h)]}{h}\end{aligned} \tag{18}$$

$$\begin{aligned}\hat{\ddot{x}}_1(t) &= \frac{\hat{\dot{x}}_1(t) - \hat{\dot{x}}_1(t-h)}{h} \\ \hat{\ddot{x}}_2(t) &= \frac{\hat{\dot{x}}_2(t) - \hat{\dot{x}}_2(t-h)}{h}\end{aligned} \tag{19}$$

$$\begin{aligned}\Delta_1(t) &= \delta_1 \cdot \cos(x_{3,leader}) - \delta_2 \cdot \sin(x_{3,leader}) \\ \Delta_2(t) &= \delta_2 \cdot \cos(x_{3,leader}) + \delta_1 \cdot \sin(x_{3,leader})\end{aligned} \tag{20}$$

The additional states update equations are identical to (10), except that the trajectory of the position in the formation for each specific vehicle (defined by their offset) is estimated and tracked instead (18). The acceleration is simply found by estimating the derivative of the velocity (19).

The drawback of this method is that each vehicle's ideal trajectory has to be estimated. In section 3.3, only the leader's trajectory had to be estimated, and that could be done by the central computing unit or even by the leader itself, before it broadcast it to every vehicles. It is not adequate here anymore, as the number of trajectories that need to be estimated is proportional to the number of vehicles, and a larger number of agents should not require a better central computing unit for scaling purposes, according to the swarm intelligence philosophy. Thus, each vehicle should estimate its own trajectory itself, and this additional computation can be a burden for smaller mobile robots, especially those without built-in trigonometric functions. However, this method should be supported easily by most of the processors used in robotics.

3.6 Considerations about applications

Depending on the application, some of the aforementioned control laws can be more appropriate than others. When formations do not need to be rotated, such as in open areas (for unmanned aerial vehicles (UAVs) applications for instance), it is possible to use the same reference trajectory for each vehicle, with a fixed offset, which has the advantage of being very simple.

For most applications, a virtual leader can be used, using the control law described by (13) for example, which makes recovery strategies easier (see section 4.3). However, there are some applications where a real leader needs to be followed, as in the case of the reference trajectory being unknown to the followers, when the leader is manually piloted for instance. The leader's trajectory needs to be estimated (3.3. and 3.5) in these cases.

In ground applications, the vehicles will most of the time have to actually trail the leader, and rotating formations are thus needed. The most appropriate control law will depend on the size of the platoon, the desired accuracy of the formation stabilization and also the computing power of the vehicles. If a small platoon of mobile robots with limited computing power is expected to move in a predictable environment and on a smooth surface, it is possible to use the same reference trajectory for each vehicle (section 3.4), the inaccuracies being limited by the small size of the platoon.

For more demanding real-world applications, the leader cannot be expected to follow the desired trajectory at all times. For small platoons, it is possible for all the vehicles to use only one reference trajectory, estimated from the leader's trajectory as in section 3.3. However, this can cause some inaccuracies, especially during sharp turns or if some vehicles are too far away from the leader. The most robust and accurate controller in this case is the one presented in 3.5.

4 Recovery strategies

Due to the unpredictability of the environment in which mobile robots may evolve, such as obstacles or uneven grounds, the actual trajectories of autonomous vehicles may differ from their theoretical path. Malfunction on a vehicle's motors can also make it fail to follow its desired trajectory, by causing it to spin or even to simply stop. Collisions can also occur when the vehicles try to simply get into formation from random positions. To deal with these unwanted behaviors, recovery strategies are needed. First, collision must be avoided with the obstacle or the uncontrollable vehicle while keeping cohesion among the platoon, and secondly, the formation has to be restored once the group is out of danger.

4.1 Potential-based discontinuous approach

In order to have a collision avoidance feature with the collective motion stabilization solutions presented in section 3, a potential-based approach was taken. Unlike [2] and [3], in which the potential descent is enforced at all times, a switching technique is used instead, so that the collision avoidance strategy is only used when an obstacle is within a certain radius of the vehicle. The advantage is that the good dynamic and convergence characteristics provided by the controller can be kept when there are no collision risks. The switching between the two modes is performed at a definite distance around the vehicles, which is why this approach is called discontinuous.

4.1.1 Control law

Instead of using a totally new control law when an obstacle has to be avoided, the target that the vehicle tries to reach, represented by the reference trajectory, is modified accordingly. When obstacles are inside a vehicle's "safe zone", defined by the radius R, a vector $vect_{avoid,i}$ is added for each obstacle it has to avoid to the original position it was trying to reach (Figure 10). The magnitude of these vectors is inversely proportional to the distance between the obstacles and the vehicle. The position resulting from the addition of all these vectors define the new position the vehicle has to try to reach. This position gives (x_{1ref}, x_{2ref}) in the new reference trajectory, with all the other derivatives being unchanged. The reason these derivatives are not changed to zero (to have a stabilization controller to (x_{1ref}, x_{2ref})) is that, if the obstacle does not change the trajectory too much, these derivatives will be close to the derivatives of the actual collision avoidance trajectory, and will thus give a quicker response than a stabilization controller.

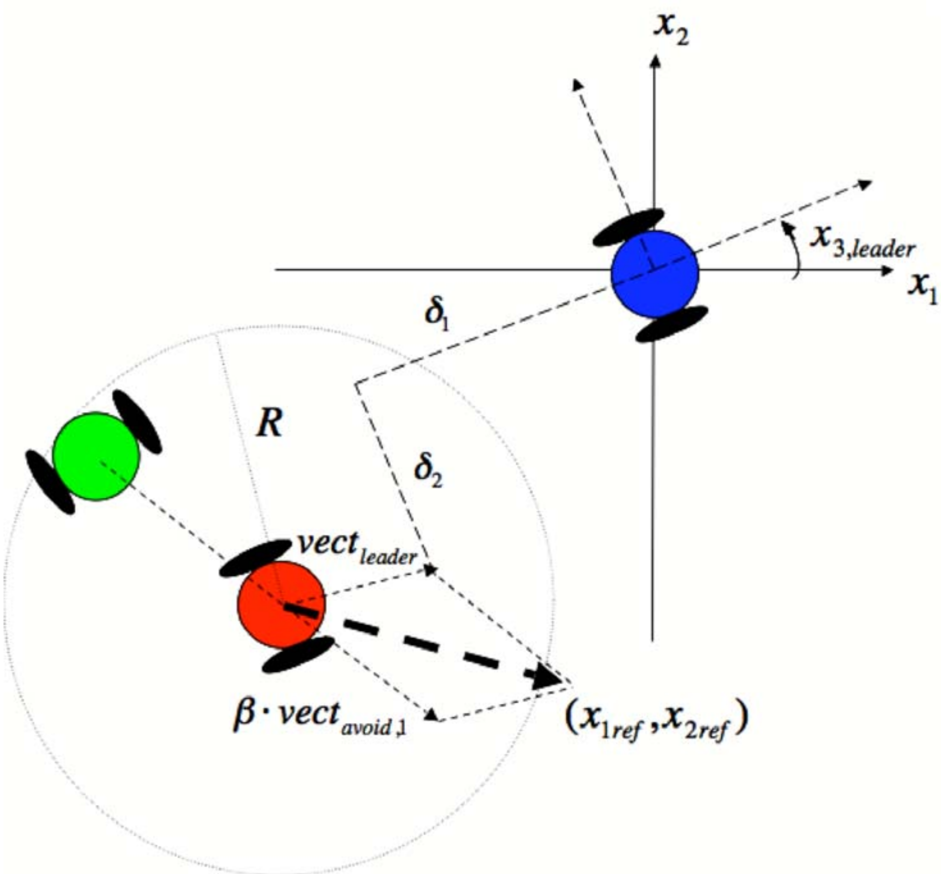


Figure 10: The red follower is attracted to its intended position in the formation (defined by the two offsets) and repulsed by the green follower in his alert zone. The resulting vector defines the point (x_{1ref}, x_{2ref}) it will try to reach.

Thus, the control law for the collision avoidance mode is similar to the collective motion stabilization controller, but with the target at (x_{1ref}, x_{2ref}) instead of $(x_{1,leader} + \Delta_1(t), x_{2,leader} + \Delta_2(t))$. Equation (21) shows the collision avoidance controller corresponding to the controller (10), but other collective motion control laws can also be used.

$$\begin{aligned} \dot{\chi}_1 &= k_1(x_1 - x_{1,ref}) + k_2(\chi_1 - \hat{\dot{x}}_{1,leader}) + \hat{\ddot{x}}_{1,leader} \\ \dot{\chi}_2 &= k_3(x_2 - x_{2,ref}) + k_4(\chi_2 - \hat{\dot{x}}_{2,leader}) + \hat{\ddot{x}}_{2,leader} \end{aligned} \tag{21}$$

(x_{1ref}, x_{2ref}) is found by adding the position the vehicle had to reach originally with the collision avoidance vectors $vect_{avoid,i}$, weighted by a constant β that determine how aggressively the vehicle turns away from the obstacles (22).

$$\begin{pmatrix} x_{1ref} \\ x_{2ref} \end{pmatrix} = \begin{pmatrix} x_{1,leader} + \Delta_1(t) \\ x_{2,leader} + \Delta_2(t) \end{pmatrix} + \beta \cdot \sum_i vect_{avoid,i} \tag{22}$$

The collision avoidance $vect_{avoid,i}$ for the ith obstacle (23) is directed away from the obstacle, along the unit vector u_i (24), and its magnitude is inversely proportional by the distance $dist_i$ between the obstacle and the vehicle (25).

$$vect_{avoid,i} = \frac{1}{dist_i} u_i \tag{23}$$

$$u_i = \frac{1}{dist_i}\left[\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} x_{1,i} \\ x_{2,i} \end{pmatrix}\right] \tag{24}$$

$$dist_i = \sqrt{(x_1 - x_{1,i})^2 + (x_2 - x_{2,i})^2} \tag{25}$$

4.2 Potential-based continuous approach

Stability analysis of switched systems, or more generally of hybrid systems, is a current domain of research (see [4] and [5]). Indeed, switching control laws can cause unwanted behaviors and even destabilize a system. For instance, an ill-suited coefficient β or radius R can make a vehicle oscillate. In order to lessen these disadvantages, switching can be performed in a smoother manner, by having a continuous transition between collision avoidance and collective motion stabilization modes.

4.2.1 Control law

The smoothening is done by defining two radii R and r, with R>r (see Figure 11). The vehicle is purely in collision avoidance mode when an obstacle is within a distance r of it, and in collective motion stabilization mode when no obstacle is within a distance R of it. If an obstacle is between a distance r and R of it, a coefficient is multiplied to the collision avoidance vector, so that it does not steer away too aggressively (26). Instead of a step function, a ramp is used to make the transition between the two modes (27).

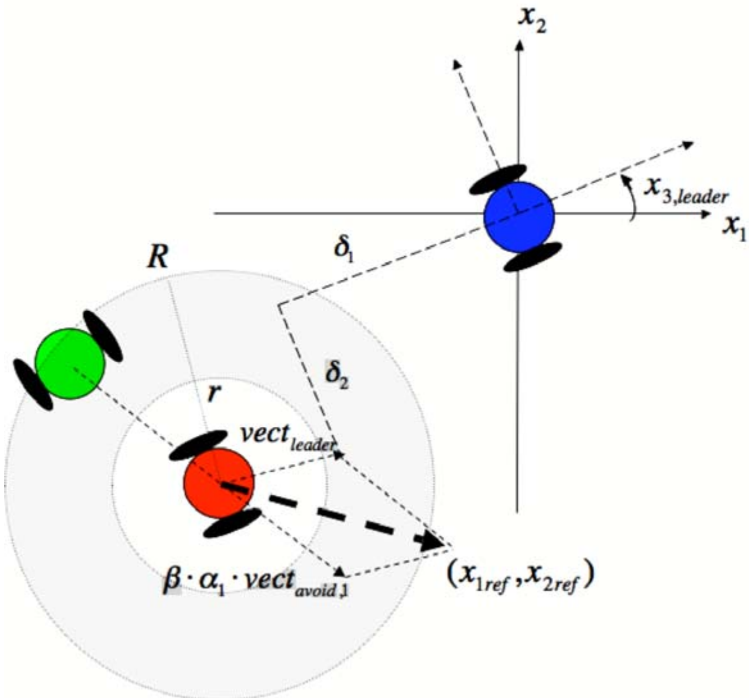


Figure 11: As in the discontinuous case, the red follower's target is modified by the presence of the green follower in his alert zone. However, the coefficient α_1 makes the repulsion less significant in the grey zone between R and r.

$$\begin{pmatrix} x_{1ref} \\ x_{2ref} \end{pmatrix} = \begin{pmatrix} x_{1,leader} + \Delta_1(t) \\ x_{2,leader} + \Delta_2(t) \end{pmatrix} + \beta \cdot \sum_i \alpha_i \cdot vect_{avoid,i} \tag{26}$$

$$\alpha_i = \begin{cases} 0 & dist_i > R \\ 1 - \dfrac{dist_i - r}{R - r} & r < dist_i < R \\ 1 & dist_i < r \end{cases} \tag{27}$$

4.3 Simulations

In the following simulations of a platoon of three vehicles, one can see the necessity of having a recovery strategy, the effect of β on the vehicles behavior, and also the improvement that the continuous approach brings. The collective motion control law used here is the one for rotating formations, described by (10) and (11). This brings an extra difficulty, as all the vehicles try to follow a real leader, which should also avoid collisions. This explains the three different strategies presented in the subsequent sections. First, simulations of platoons that do not use any collision avoidance strategy are presented. Then, platoons where all vehicles but the leader use either the discontinuous or the continuous approach are simulated. Finally, simulations of platoons where all the vehicles use the same collision avoidance strategy are shown. In all those simulations, the initial conditions are the same to make comparison easier.

4.3.1 No recovery strategy

When no collision avoidance technique is used, collisions can occur when the vehicles try to reach their formation position, depending on their initial conditions (Figure 12).

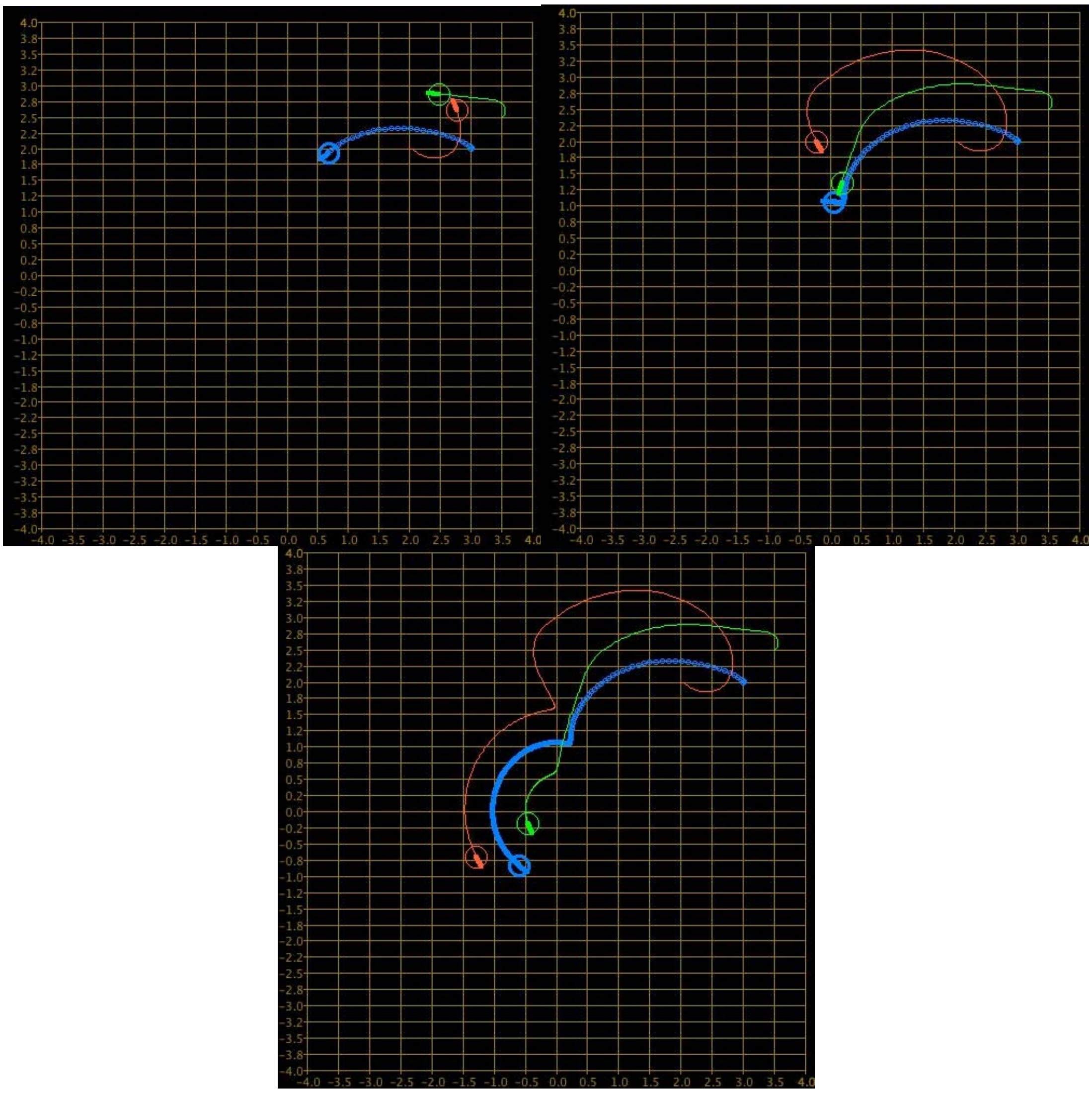


Figure 12: If no collision avoidance technique is used with this set of initial conditions, two collisions occur before the vehicles reach their position in the formation.

4.3.2 Leader without collision avoidance mode

In all of the following figures, the leader does not use any collision avoidance technique. This can be applicable under the assumptions that the other vehicles will be able to avoid it and that the reference trajectory was designed so that no obstacle is in the direct path of the leader. A clear leader's path can be ensured by continuously creating a map of the environment, using the leader's sensors and the SLAM (Simultaneous Localization and Mapping) algorithm for instance [6], and having a trajectory regeneration scheme if an obstacle appears on the original trajectory.

If all the vehicles except the leader use the potential-based discontinuous recovery strategy with appropriate parameters, the formation can be established without any collisions (Figure 13).

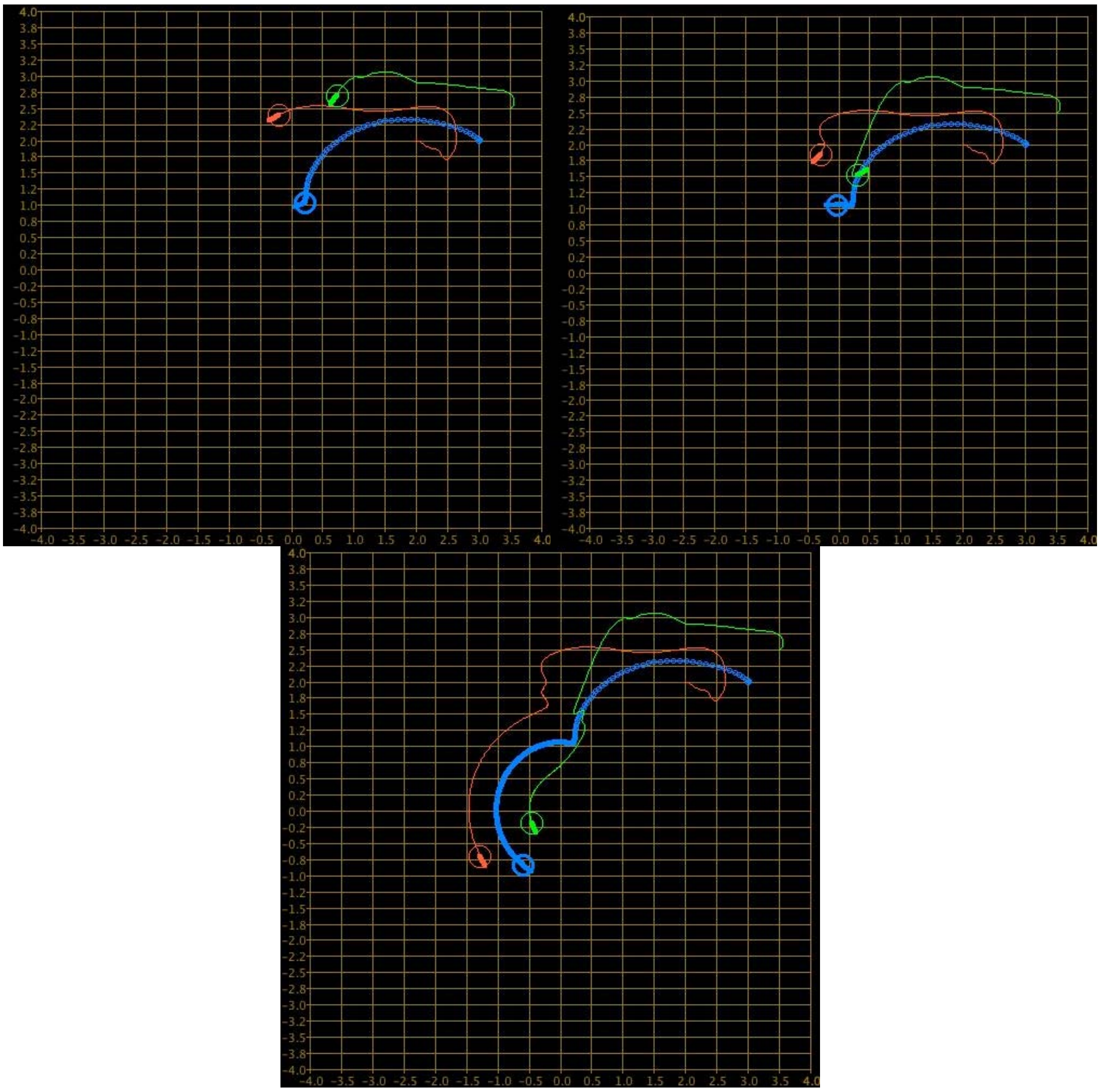


Figure 13: When the discontinuous approach is used by the red and green followers with parameters $\beta = 1$ and $R = 3d$ (d is the diameter of the vehicles), collisions are avoided before the formation is established.

The parameters selection is important, as coefficients that are too small may not be able to prevent certain collisions, whereas too large ones may create unnecessarily aggressive control inputs, which increase the formation settling time, mechanical wear on the vehicles motors, and can even destabilize the system. The latter case arises when the difference between the two control laws is too big, which causes the vehicle to repetitively switch between the two modes. Indeed, the collision avoidance controller makes the vehicle go farther away from its desired position in the formation, which, when it goes back into the collective motion stabilization mode, triggers a bigger control input that can make it overshoot and enter the collision avoidance mode again (Figure 14).

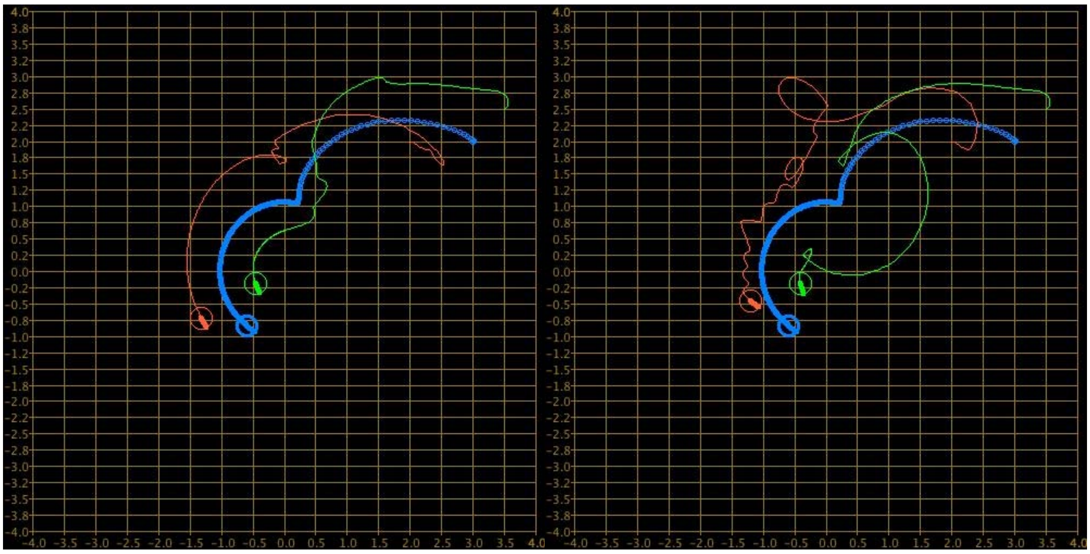


Figure 14: With $\beta = 2.5$ (left figure), the trajectories of the red and green vehicles are a lot less smooth than with $\beta = 1$. With $\beta = 5$ (right figure), the collision avoidance controller becomes too aggressive and destabilizes the red follower.

The continuous approach has been specifically designed to avoid this kind of behavior caused by switching. The following simulations (Figure 15) show that collision is avoided in a smoother manner with this technique.

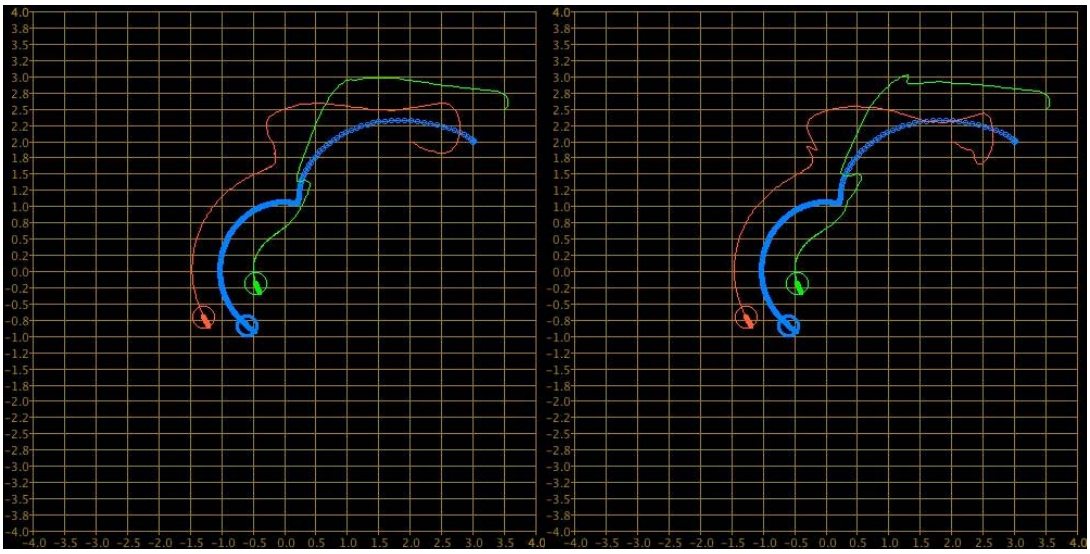


Figure 15: Using the continuous approach with $R = 3d$ and $r = \frac{R}{2}$, the vehicles trajectories are smoother, while collision avoidance is still ensured with $\beta = 1$ (left figure). Compared to the discontinuous approach, it stabilizes the case with $\beta = 5$ (right figure).

In these simulations, the parameters β, R and r were chosen on an experimental way. The best set of parameters for a certain application depends on the velocity of the vehicles, the distance between them in the formation and the safety margin requirements.

4.3.3 Leader with collision avoidance mode

The previous simulations were done with a leader that did not have any collision avoidance technique. This could be acceptable for most of the situations, as when vehicles try to get in formation starting from random initial conditions, or when a possible collision might occur within the formation, but not in the proximity of the leader. Nevertheless, this method is not robust if a follower becomes uncontrollable and runs into the leader, or if any other kind of unexpected obstacle collides with the leader. The following simulations (Figure 16) show the problems that can arise if the leader has a collision avoidance scheme.

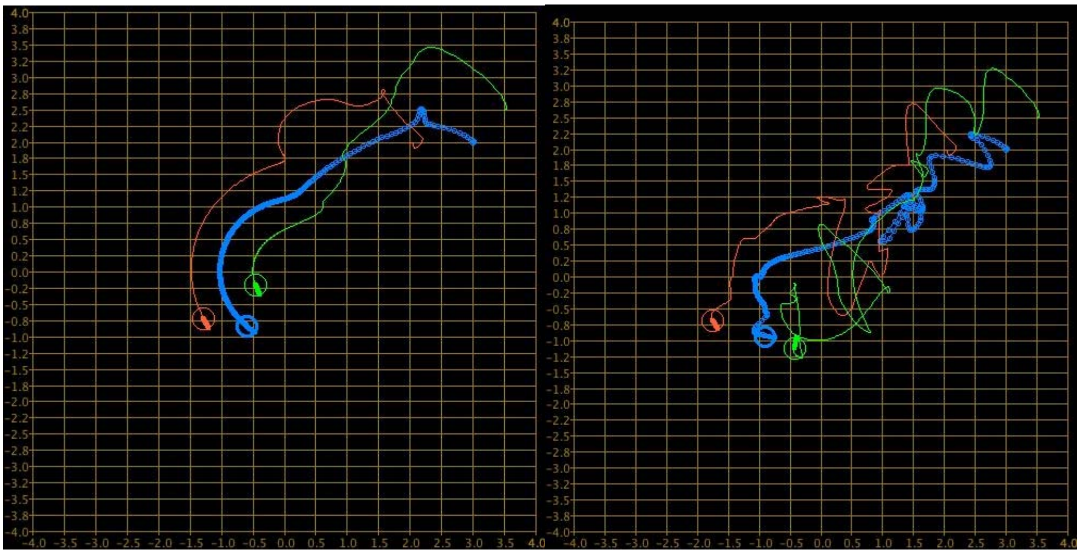


Figure 16: With the discontinuous approach and $\beta = 1$ (left figure), the platoon needs more time and control effort to stabilize than in the case where the leader does not try to avoid collisions at all. When β is increased to 2, the platoon does not converge to its desired formation anymore.

The fact that the followers try to move toward the leader, while it tries to avoid them when they come too close, causes instability. The platoon is thus more sensitive to the value of the parameter β. The discontinuous approach reduces this sensitivity (Figure 17).

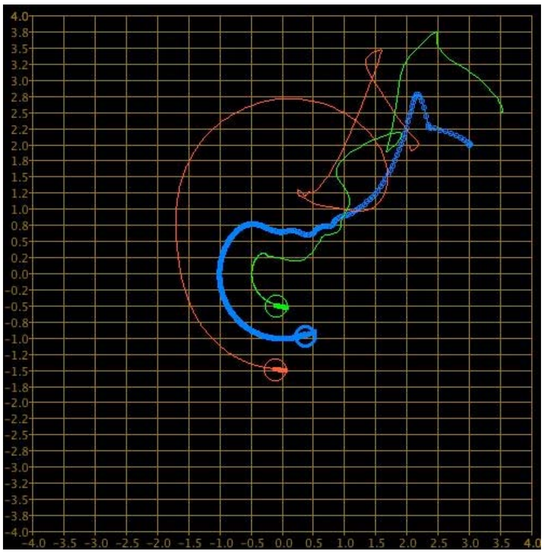


Figure 17: If all the vehicles use the continuous approach with $R = 3d$ and $r = \frac{R}{2}$, the platoon can still stabilize itself with $\beta = 5$.

If a collective motion stabilization control law that has a virtual leader is used instead, these instability problems will not appear, as the leader would not have to avoid collisions and lead the platoon at the same time. The results would be similar to those shown in section 4.3.2, where the leader did not have any collision avoidance mode and the dependence on the parameter β was less critical.

5 Conclusion

Several solutions, all based on the one-vehicle flatness-based controller [1], were proposed for the flocking problem and can be used for different applications, depending on the size and the available computing power of the platoon. The collision avoidance and the recovery problems were addressed at once by introducing a switching strategy between the flocking control mode and a potential-based collision avoidance mode. A smoother switching technique was also tested to reduce the likelihood of unwanted oscillations between the two modes. The resulting controller has been simulated, showed good results, and can thus be used for collective motion control.

In order to improve the recovery strategy proposed in this paper, a smooth function can be used for the transition between the two modes, a hyperbolic tangent function for instance, instead of the ramp function in section 4.2. This would make pathological switching behaviors less likely to appear and stability analysis easier to perform.

Another possible improvement in the collision avoidance technique would be to change the first and second derivatives in the trajectory-tracking controller. Indeed, in the solution proposed here, the target the vehicle tries to reach changes according to the obstacles and the leader's states, but the trajectory of this point over time is not taken into account. Using Euler's backward approximations, it is possible to estimate the first and second derivatives of the trajectory of this point, and use them as the reference trajectory that the vehicle should follow. This would improve the vehicle's dynamic behavior, as the extra information given by the derivatives will make it go more quickly and accurately to its desired trajectory.

6 References

[1] D. Buccieri, D. Perritaz, P. Mullhaupt, Z.P. Jiang and D. Bonvin, “Velocity Scheduling Controller for a Nonholonomic Mobile Robot: Theoretical and Experimental Results,” in *IEEE Transactions on Robotics and Automation*, 2006

[2] G.H. Elkaim and R.J. Kelbley, “A Lightweight Formation Control Methodology for a Swarm of Non-Holonomic Vehicles,” in *IEEE Aerospace Conference*, 2006

[3] E.W. Justh and P.S. Krishnaprasad, “Equilibria and steering laws for planar formations,” in *Systems & Control Letters 52*, 25-38, 2004

[4] J. Daafouz, P. Riedinger and C. Iung, “Stability Analysis and Control Synthesis for Switched Systems: A Switched Lyapunov Function Approach,” in *IEEE Transactions on Automatic Control*, vol. 47, no. 11, 2002

[5] D. Liberzon, “Switching in Systems and Control,” Springer, 2003

[6] V.T. Nguyen, A. Harati, N. Tomatis, A. Martinelli and R. Siegwart, “Orthogonal SLAM: a Step toward Lightweight Indoor Autonomous Navigation,” in *Proc. of the IEEE/RSJ Intenational Conference on Intelligent Robots and Systems*, 2006

[7] H.G. Tanner, G.J. Pappas and V. Kumar, “Leader-to-Formation Stability,” in *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, 2004

[8] S. Hettiarachchi and W. Spears, “Moving Swarm Formations Through Obstacle Fields,” in *International Conference on Artificial Intelligence*. CSREA, Press 1, 97-103, 2005

[9] D. Paley, N.E. Leonard and R. Sepulchre, “Collective Motion: Bistability and Trajectory Tracking,” in *Proc. of the 43rd IEEE Conference on Decision and Control*, 2004

[10] J. Jeanne, N.E. Leonard and D. Paley, “Collective Motion of Ring-Coupled Planar Particles,” in *Proc. of the 44th IEEE Conference on Decision and Control*, 2005

[11] W.B. Dunbar and R.M Murray, “Receding horizon control of multi-vehicle formations: a distributed implementation,” in *Proc. of the 43rd IEEE Conference on Decision and Control*, 2004

[12] D. Paley, N.E. Leonard and R. Sepulchre, “Oscillator Models and Collective Motion: Splay State Stabilization of Self-Propelled Particles,” in *Proc. of the 44th IEEE Conference on Decision and Control*, 2005

[13] A. Ganguli, S. Susca, S. Martinez, F. Bullo and J. Cortés, “On collective motion in sensor networks: sample problems and distributed algorithms,” in *Proc. of the 44th IEEE Conference on Decision and Control*, 2005

[14] S. Hettiarachchi and W. Spears, “DAEDALUS for Agents with Obstructed Perception,” in *IEEE Mountain Workshop on Adaptive and Learning Systems*, 2006

[15] A. Engwirda, T. Engwirda, L. Vlacic and A. Sattar, “Cooperative Autonomous Robot Colonies,” in *Proc. of IEEE International Conference on Industrial Technology*, 2000