Decentralized Multi-Agent Systems: Collusion or Collision

A study into the effect of communication and sensing when controlling a decentralized system of autonomous agents.

Scott M. Kyle

April 20, 2018

1 Problem Introduction & Motivation

Suppose there exists an arena, $\mathcal{A} \subseteq \mathbb{R}^m$, with $n > 0$ agents, $a_i, i \in \{1, \ldots, n\}$. Now suppose that each a_i is designed to achieve a task, Γ_i, that has an associated "variable" cost, J_i, to complete (i.e., time, distance, energy consumption etc.).

Logically, the goal for each agent is to minimize its own cost. In turn, this will correspond to a planned trajectory, $\gamma_i : \mathcal{A} \to \mathcal{A}$, from some initial location $X_i^0 \in \mathcal{A}$, to some destination, $X_i^f \in \mathcal{A}$ (and without loss of generality, assume that for any $t \in \mathbb{R}_{>0}$, $X_i(t) \in \mathcal{A}$), as seen in Figure 1.

Real-world applications range from mines, to airports (and airspace), to storage warehouses. Currently, agents are organized through some centralized hub, controlled by a number of "fixed" cost decision makers (human, or computer). For the sake of order and safety, this essentially restricts the paths of agents to a combination of pre-determined paths i.e., roads, lanes, channels etc. meaning that the overall "variable" cost of the task is not generally minimized.

Now suppose you want to make the system decentralized i.e., there is no central decision maker, and agents (or vehicles in this case) make decisions based solely on local information. The justification of implementing a decentralized system is twofold, as it:

- eliminates, or reduces, the "fixed" costs i.e., central controller; and
- allows agents to minimize their "variable" cost function;

all while ensuring the overall system meets (and exceeds) the current functional and performance requirements.

While this project does not focus on optimization and graph theory, we will assume each agent's cost can be minimized (or have a known trajectory, or final location) using some prescribed technique (e.g., as described in [1]). As such, will only deal with the control aspect of the problem—which can be broken into two fundamental problems:

1. A global trajectory (or path) tracking problem: given an initial location $X_i^0 \in \mathcal{A}$, can each agent, a_i, be steered to to some destination, $X_i^f \in \mathcal{A}$?
2. A local avoidance problem: is the system capable of avoiding other agents, objects, and disturbances?

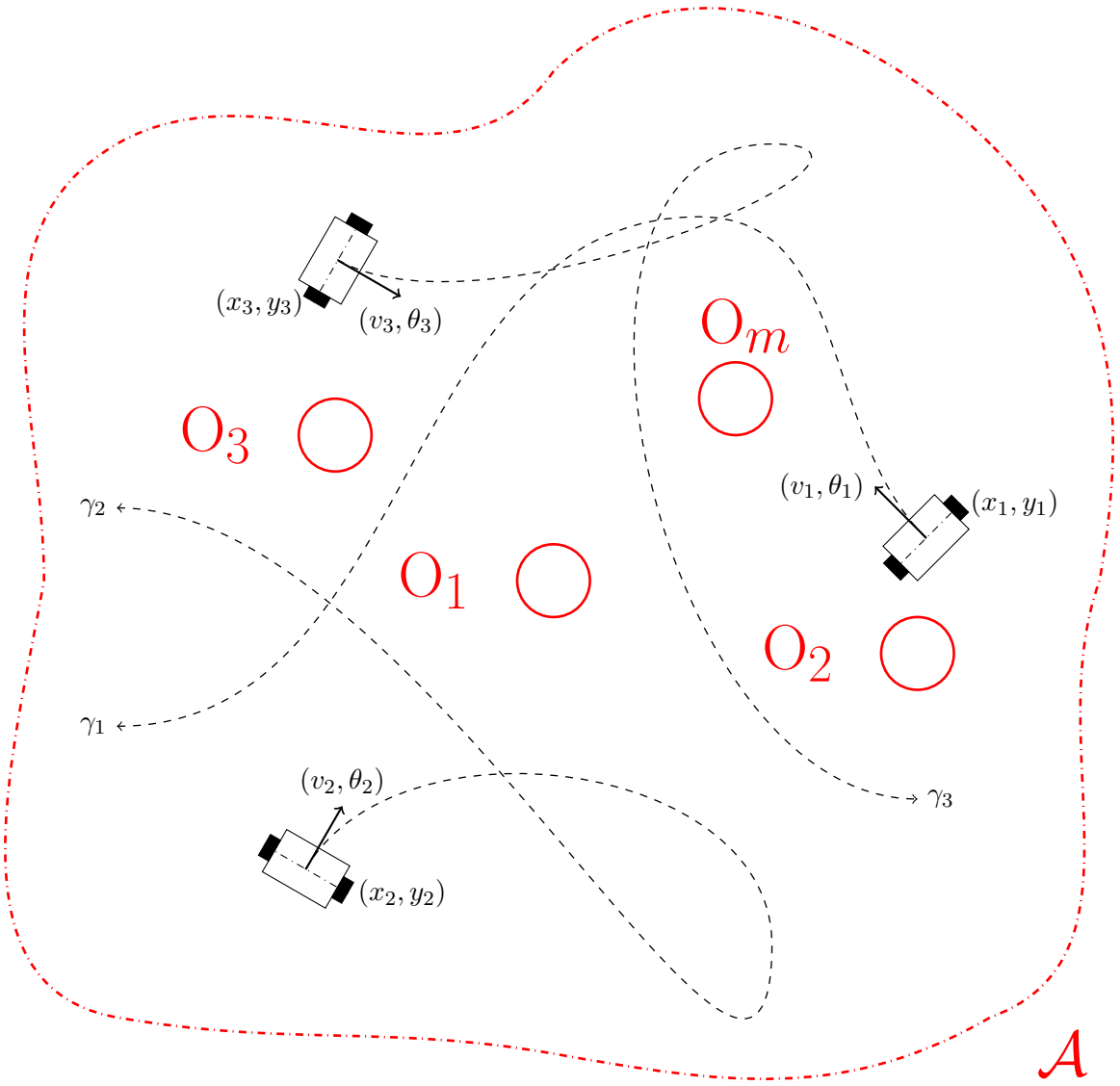


Figure 1: An arena, $\mathcal{A}$, with three agents and m objects.

The first problem is trivial for one agent, or even two or three. But as the number of agents increases, how do system constraints, dynamics, and disturbances from part two impact the overall performance of the system?

The constraints on the system are what make this problem interesting as they impact each agent's Guidance, Navigation, and/or Control capabilities, and include (but are not limited to):

- Kinematics: How can an agent move (holonomic or non-holonomic)? Are there velocity bounds?
- Global Trajectory: Where can each agent go and why? What information does the agent know about the arena?
- Sensors: What state information does each agent have access to? How reliable is it? How can agents best fuse data from different sources? How does this impact controllability and observability?
- Communication: How can each agent communicate? How far, how fast and with whom i.e., how *centralized* is the system?
- Microdynamics: How quickly can an agent respond to the required input decided by the macrosystem's algorithm (macrodynamics)?

The main constraints considered in this projects are the kinematic constraints of the vehicle and the communication and sensing radii of each agent. This is outlined in section 3.2.

The macrosystem is modelled and controlled as per the methods outlined in [2], [3], which utilizes reference vector fields. These reference vector fields are composed of two parts:

- Attractive Vector Fields: Help solve the global path tracking problem. From the agent's initial location, a vector field is generated that steers the agent to its final destination.
- Repulsive Vector Fields: Help solve the local avoidance problem. For an object at a known location i.e., an obstacle or vehicle (from prior knowledge, communicated, or sensed), a vector field is generated that steers the agent away from this object.

As this was used as a basis for system modelling and control, only the major steps will be identified in this report. Refer to [2] and [3] for further detail.

Current literature, from [1], [4], [5] and [6], assesses this problem in various ways, yielding a spectrum of results. However, what they have in common is that all agents are attempting to coordinate a common task, i.e., rendezvous or flocking.

The goal of this research is to study the interaction of agents attempting different tasks. The desired outcome is to have the number of agents in the arena increase, while decreasing some "crash" or "unsafe operating distance" metric. Further, an accurate block diagram of the system has been created, meaning future work can explore and account for vehicle dynamics, actuator dynamics, communication and sensing profiles, communication delays, state estimation etc.

2 Assumptions, Simplifications and Constraints

Before modelling, controlling or experimenting on the system, assumptions, simplifications and constraints of the system need to be identified.

The assumptions include:

- Each agent knows its initial location, final location, and own map (which is a function of these two properties and the location of known obstacles).
- Communication among agents, and sensing of other agents or obstacles, is instantaneous.

The simplifications and constraints include:

- Vehicles are modelled as unicycles (a non-holonomic constraint) with the i-th vehicle's state $q_i = (x_i, y_i, \theta_i)$ and control inputs v_i, and ω_i, the forward velocity, and steering rate respectively.
- The communication & sensing radii are modelled as circles of radius R_c and R_s, from the centre of the vehicle. R_c and R_s are not necessarily the same for each vehicle.

3 Modelling

3.1 Vehicle Motion

Each vehicle, a_i, is modelled as a unicycle, and hence its motion is governed by:

$$\underbrace{\begin{bmatrix}\dot{x}_i\\ \dot{y}_i\\ \dot{\theta}_i\end{bmatrix}}_{\dot{q}} = \underbrace{\begin{bmatrix}\cos\theta_i & 0\\ \sin\theta_i & 0\\ 0 & 1\end{bmatrix}}_{G(q)} \underbrace{\begin{bmatrix}v_i\\ \omega_i\end{bmatrix}}_{\mathbf{u}}.$$

3.2 Communication and Sensors

The communication and sensing radii for each vehicle, a_i, are modelled as circles of radius R_c^i and R_s^i, about (x_i, y_i), as per figure 2.

In practice, each vehicle would have a communication and sensing "profile", which would be dependent on the hardware used and set-up configuration.

3.3 Arena Mapping Using Reference Vector Fields

The bulk of the modelling process deals with determining each agent's map. After extensive research, it was determined to use the method of Reference Vectors as developed and discussed in [2].

This method, while mathematically rigorous, provides a novel solution to both the global path tracking (or long-range planning) problem and local avoidance (short-range planning) problem, and can be done from prior arena knowledge, **and/or** in real time.

3.3.1 Reference Vector Fields

The following section gives a brief introduction to the family of two-dimensional analytic vector fields. For further reading on the fundamental properties of vector fields, refer to [7] or [8].

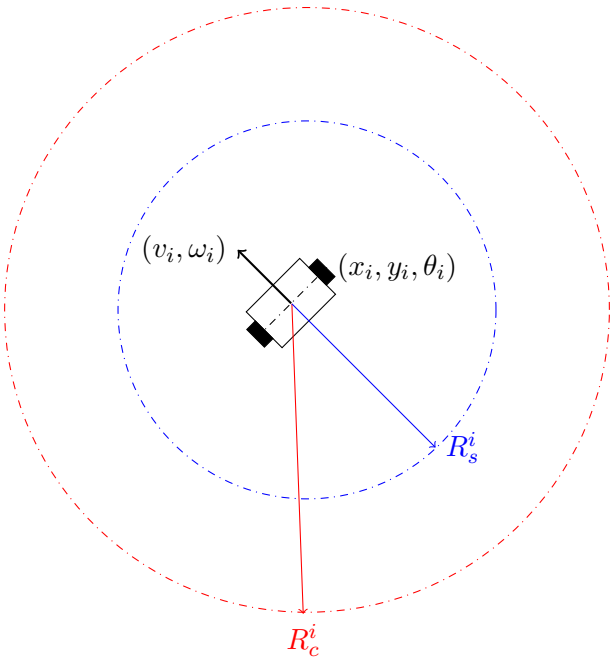


Figure 2: The simplified communication and sensing profile of an agent.

For a deeper understanding of this particular family of vector fields for this application, refer to [2] and [3].

The family of vector fields used in this method is of the form

$$F(r) = \lambda \left(p^T r\right) r - p \left(r^T r\right), \tag{1}$$

where $r \in \mathbb{R}^2$ is the coordinate of the vector field being computed, $p \in \mathbb{R}^2$ is the "orientation vector" that satisfies the non-holonomic constraints of the vehicle (and hence vector field) from r to the agent's final position. This essentially orients the vector field in the direction of the final coordinate. The parameter $\lambda = 0, 1, 2$, determines if the vector field is *attractive* or *repulsive*.

With $r = \begin{bmatrix} x & y \end{bmatrix}^T$ and $p = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$, equation 1 can be expressed in matrix form as

$$F = \begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} (\lambda - 1)\, x^2 - y^2 & \lambda xy \\ \lambda xy & (\lambda - 1)\, y^2 - x^2 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}. \tag{2}$$

It is clear that this vector field is non-zero everywhere except for the origin $r = (0, 0)$. This origin can be placed in each obstacle, agent or final coordinate reference frame by a change in coordinates, hence the agent positions (initial and final) and obstacle locations must be known. This transformation is demonstrated at the end of section 3.3.3.

3.3.2 Attractive Vector Fields

An attractive vector field arises when $\lambda = 2$ is substituted into equation 2, yielding

$$F_A = \begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} x^2 - y^2 & 2xy \\ 2xy & y^2 - x^2 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}, \tag{3}$$

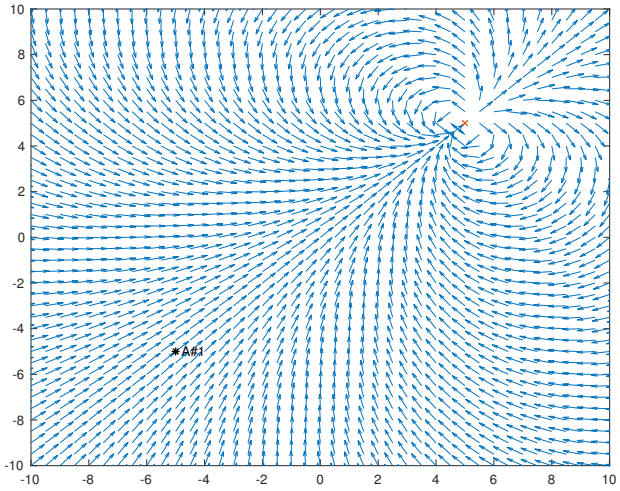


Figure 3: An attractive vector field. $X^0 = (-5, -5)$, and $X^f = (5, 5)$

and is used to steer the vehicle from its initial coordinate to its final destination i.e., for long-range planning.

The attractive field is generated by sweeping all coordinates $(x, y) \in \mathcal{A}$.

With initial location $X^0 = \left(x^0, y^0\right) \in \mathcal{A}$, and final destination, $X^f = \left(x^f, y^f\right) \in \mathcal{A}$, the vector $p = \begin{bmatrix} p_x & p_y \end{bmatrix}^T = \begin{bmatrix} \cos\phi & \sin\phi \end{bmatrix}^T$ where $\phi = \arctan\left(\frac{y^f - y^0}{x^f - x^0}\right)$. This is done for each agent.

An example attractive field, for one agent, is given in figure 3.

3.3.3 Repulsive Vector Fields

Determining the repulsive vector field for an obstacle is slightly more involved, as it factors in whether or not the agent is "before" or "past" the object based on their current position and final destination.

The geometry and ultimately the determination of if the vehicle is "before" ($\lambda = 1$) or "past" ($\lambda = 0$) an obstacle can be reviewed in [3].

In the case when $\lambda = 1$, equation 2 becomes

$$F_R = \begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} -y^2 & xy \\ xy & -x^2 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}, \tag{4}$$

and when $\lambda = 0$, equation 2 becomes

$$F_R = \begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} -x^2 - y^2 & 0 \\ 0 & -y^2 - x^2 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}. \tag{5}$$

The repulsive field is generated by sweeping through all coordinates $(x, y) \in \mathcal{A}$. In practice (and in simulations) this is unnecessary as there are parameters that make the contribution of the repulsive

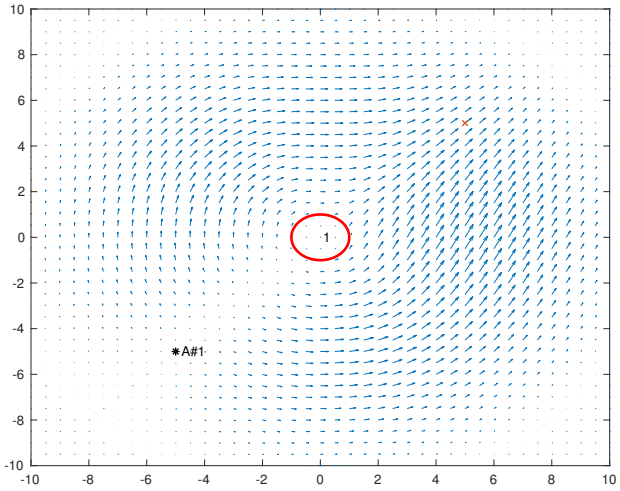


Figure 4: A repulsive vector field. $X^0 = (-5, -5)$, $X^f = (5, 5)$, and $p_0 = (0, 0)$.

field identically zero far enough away from the obstacle. Additionally, the value calculated is divided by the distance (Euclidian norm) from the obstacle, and as such, converge to zero.

The initial position does not impact the repulsive field as the value at each r is calculated as if the vehicle is at that specific location, i.e., $r = (x, y)$—the vehicles state. As such, $\phi = \arctan\left(\frac{y^f - y}{x^f - x}\right)$ and p is as above. Essentially ϕ is the orientation of the vehicle and therefore, the vector p satisfies the non-holonomic constraint.

See, for example, figure 3, where $X^0 = (-5, -5)$, $X^f = (5, 5)$, and with an obstacle of radius, $d_r = 1$ at $p_o = (0, 0)$, the repulsive field generated is shown in figure 4.

Note that the above example has the repulsive field (i.e., the obstacle) located at $(0, 0)$, however the above equations hold for an obstacle located at any $p_o = (x_o, y_o)$ per the transformation $F(r - p_o) = \lambda\, p^T (r - p_o)(r - p_o) - p(r - p_o)^T (r - p_o)$.

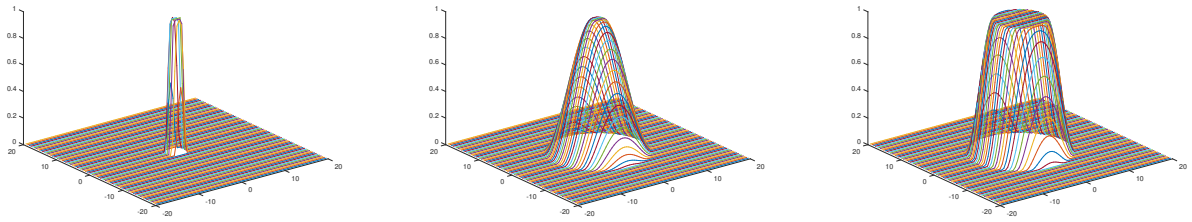
3.3.4 Blending Vector Fields

An agent a_i, has a blended vector field (or map), F_i^*, given by

$$F_i^* = \sum_{i=1}^{m} \sigma_i F_{R_i} + (1 - \sigma_j) F_{A_i}, \tag{6}$$

where F_A and F_R are as above and σ_i is the bump function, which weights the contribution of the repulsive and attractive fields at a given point. This value is calculated at the same time as the repulsive field, and is defined (as per [3]) as

$$\sigma_i = \begin{cases} 1, & 0 < d_N < d_r; \\ ad_N^3 + bd_N^2 + cd_N + d, & d_r < d_N < d_c; \\ 0, & d_N > d_c; \end{cases} \tag{7}$$



(a) $d_r = 1, d_c = 2$ (b) $d_r = 1, d_c = 10$ (c) $d_r = 5, d_c = 10$

Figure 5: Various sigma-meshes for an obstacle at the origin.

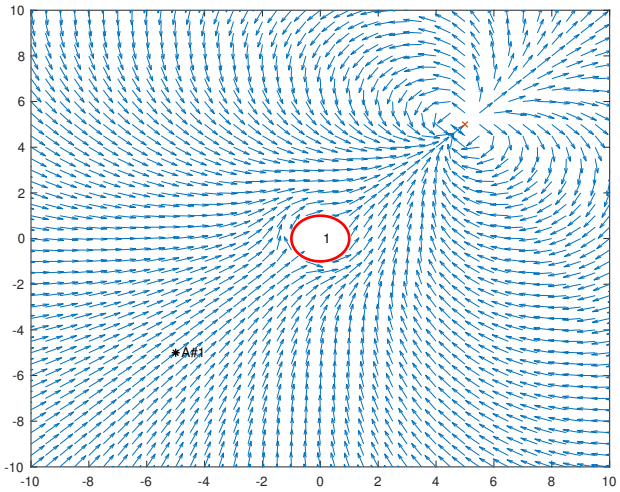


Figure 6: A blended vector field. $X^0 = (-5, -5)$, $X^f = (5, 5)$, $p_0 = (0, 0)$, $d_r = 1$ and $d_c = 5$.

where $d_N = ||r - p_o||$ is the distance of the coordinate from the centre of the object, d_r is the radius of the object, and d_c is the radius of *contribution*, i.e., how far away from the object does it affect the vector field. Sigma values can be calculated for all obstacles. The values a, b, c, d are functions of d_r and d_c. To determine the sigma-mesh, the highest σ value at the coordinate is used. Examples of various sigma-meshes for an obstacle location at the origin are given in figure 5.

The impact of d_r and d_c on system performance are explored later in this project.

Continuing the earlier example, the attractive and repulsive fields from figures 3 and 4 are blended in figure 6. Note that the field F_i^* is normalized at each coordinate in the arena as it is used for directional purposes only.

This process can be repeated for the n agents and m known obstacles in the arena to obtain the *a priori* map for each agent.

3.3.5 Real-Time Mapping

A real-time map can be calculated at each simulation time-step using the *a priori* map as a basis. If a vehicle or obstacle is within an agent's R_c or R_s, then the *a priori* map is updated by placing repulsive vector fields at the now known locations of the other vehicles or obstacles.

The impact of R_c and R_s on system performance are explored later in this project.

4 Vehicle Guidance and Control

4.1 Guidance

The "optimal" paths, or trajectories, are not explicitly calculated using reference vector fields, which might be seen as a drawback of this approach. That said, the *a priori* map and the prescribed control law forces the system to flow along the vector field to the final destination, meaning optimality can be achieved by tuning the controller parameters appropriately. The main advantage of this method is that it allows the vehicle to respond in real-time to unknown obstacles and agents (assuming there are communication and sensing capabilities).

4.2 Control

The control law governing each agent's forward velocity v_i, and the steering rate ω_i is given by

$$v_i = -k_v \tanh \left|\left| X_i - X_i^f \right|\right|; \tag{8}$$

$$\omega_i = -k_\omega \left(\theta_i - \phi_i\right) + \dot{\phi}_i; \tag{9}$$

where

k_v is the forward velocity control gain;

k_ω is the steering rate control gain;

X_i is the agent's current position;

X_i^f is the agent's final destination;

θ_i is the agent's current orientation;

$\phi_i = \arctan\left(\dfrac{F_{i_y}^*}{F_{i_x}^*}\right)$ is the orientation of the reference vector field at X_i; and,

$$\dot{\phi}_i = \left(\left(\frac{\partial F_{i_y}^*}{\partial x}\cos\theta + \frac{\partial F_{i_y}^*}{\partial y}\sin\theta_i\right)F_{i_x}^* - \left(\frac{\partial F_{i_x}^*}{\partial x}\cos\theta_i + \frac{\partial F_{i_x}^*}{\partial y}\sin\theta_i\right)F_{i_y}^*\right)v_i.$$

Notice that the forward velocity is not a function of the vector field.

The steering rate is a function of the difference in vehicle and vector field orientations. Ideally, this is zero, as it means the vehicle is travelling in the same direction as the flow of the reference vector field. The $\dot{\phi}$ term accounts for any changes in the direction of the vector field at the current location, and is the term that allows agents to avoid both obstacles and other agents.

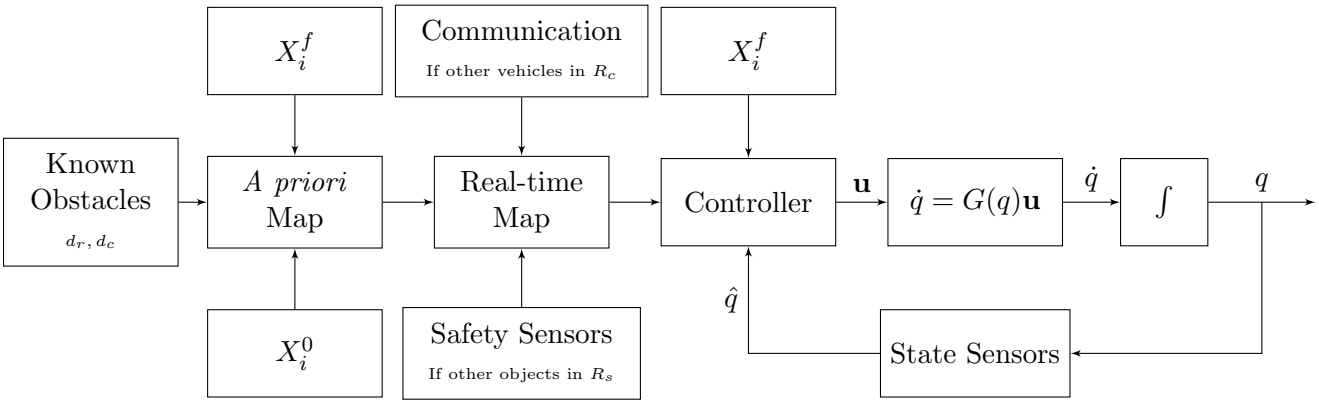


Figure 7: System architecture for one vehicle.

As analytical expressions for the attractive and reference vector fields have been established, the partial derivatives of each vector field can be computed and blended in a similar manner to section 3.3.4. This relationship (shown for x, but the same argument applies for y) is given by

$$\begin{aligned}\frac{\partial F_i^*}{\partial x} &= \frac{\partial}{\partial x}\left(\sum_{i=1}^{m} \sigma_i F_{R_i} + (1-\sigma_j)\, F_{A_i}\right) \\ &= \sum_{i=1}^{m} \sigma_i \frac{\partial}{\partial x} F_{R_i} + (1-\sigma_j)\, \frac{\partial}{\partial x} F_{A_i} \\ &= \sum_{i=1}^{m} \sigma_i \frac{\partial F_{R_i}}{\partial x} + (1-\sigma_j)\, \frac{\partial F_{A_i}}{\partial x}. \end{aligned} \tag{10}$$

The derivative reference vector fields are computed in parallel with the *a priori* and real-time maps.

The control gains are calibrated in section 5.

5 Implementation and Results

The overall system architecture is given in figure 7, and dictates the general implementation algorithm for each vehicle.

The implementation goal is to observe the impact of an agent's radii of communication and sensing (R_c and R_s), and an obstacle's (or other agent's) radius and radius of contribution (d_r and d_c), on the overall performance of the system.

5.1 Testing Metrics

Testing metrics include the minimum passing distance between any two agents, $D_{i,j}$, and the total distance travelled for an agent compared to the distance from its initial location to its final

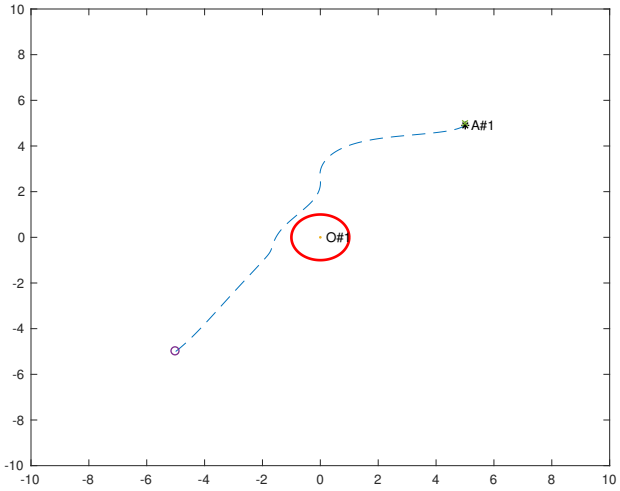


Figure 8: The path followed by an agent with parameters in table 1.

destination, E_i, represented by equations 11 and 12, respectively.

$$D_{i,j} = \min ||X_i - X_j|| \tag{11}$$

$$E_i = \frac{\int \gamma_i(t)\, dt}{\left|\left|X_i^0 - X_i^f\right|\right|} \tag{12}$$

5.2 Controller Tuning

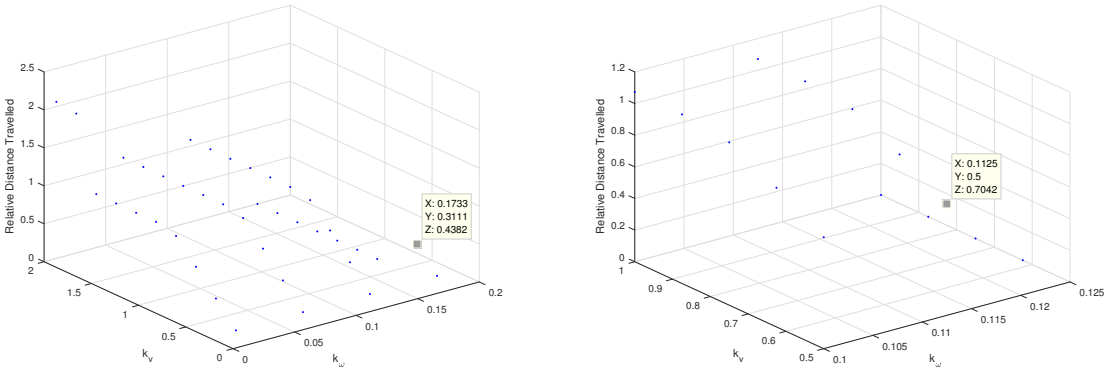
Next, the controller gains k_v and k_ω need to be tuned. The simulation conditions to tune the controller are given in table 1. An iterative approach was taken using one vehicle, with $k_v \in (0, 5]$ and $k_\omega \in (0, 1)$.

Figure 9 demonstrates this combination of control parameters. For a value of $k_\omega > 0.12$, the system did not "react" in time to the obstacle. Additionally, a value of $k_v < 0.7$, meant the vehicle did not "reach" the final destination in the simulation time, and a value of $k_v > 2$ caused the vehicle to "over-react" and travel further in order to reach the final destination.

Values of $k_v = 1.5$ and $k_\omega = 0.1$ allowed the vehicle to reach the destination and react to the obstacle appropriately. These values will be used in the simulations moving forward, as seen in figure 8. Note the path *after* the object—this phenomenon will be explored below in section 5.3.1.

5.3 General Implementation

Table 2 outlines the values that need to be defined for each agent in each simulation. The radius of the obstacle cannot be "controlled" in a real application, and as such a relationship between d_r and d_c will be determined (denoted as $\alpha \in \mathbb{R}_{>1}$ and $\beta \in \mathbb{R}_{>1}$), essentially providing the safety margin for an object.



(a) $k_v \in (0, 2]$ and $k_\omega \in (0, 0.2)$ (b) $k_v \in (0, 1]$ and $k_\omega \in (0, 0.12)$

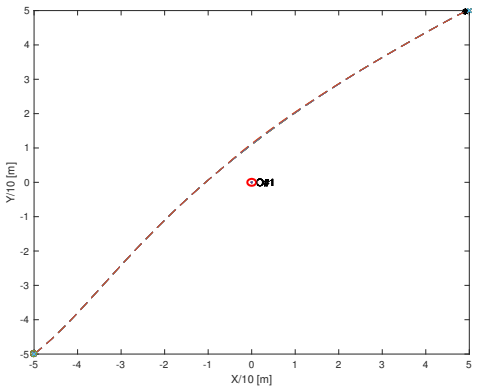
Figure 9: Relative distance travelled, E_i, for various controller gains.

Table 1: Simulation Parameters for controller tuning.

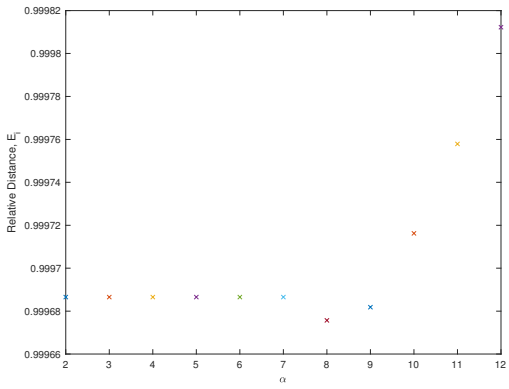
Parameter	Description	Simulation Value
T	Simulation Time	20 s
δt	Simulation Step	0.05 s
D	Arena Size	100×100 m^2
δx	Arena Discretization	0.1 m
n	Number of Agents	1
X^0	Initial State	$(-5, -5)$ or $(-50$ m$, -50$ m$)$
X^f	Final Destination	$(5, 5)$ or $(50$ m$, 50$ m$)$
L_v	Vehicle Length	2.0 m
R_c	Vehicle Radius of Communication	0 m
R_s	Vehicle Radius of Sensing	0 m
m	Number of Obstacles	1
d_{r_o}	Obstacle Radius	10 m
d_{c_o}	Obstacle Radius of Contribution	50 m

Table 2: Simulation Parameters.

Parameter	Description	Simulation Value
X^0	Initial State	Trial Dependent
X^f	Final Destination	Trial Dependent
L_v	Vehicle Length	2.0 m
d_{r_v}	Vehicle Radius	$1.5L_v$ m
d_{c_v}	Vehicle Radius of Contribution	βd_{r_c} m
d_{r_o}	Obstacle Radius	1.0 m
d_{c_o}	Obstacle Radius of Contribution	αd_{r_o} m

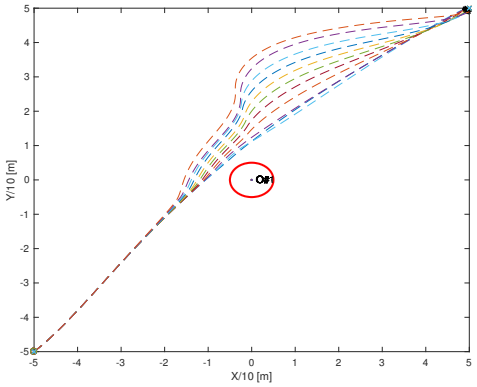


(a) Paths travelled.

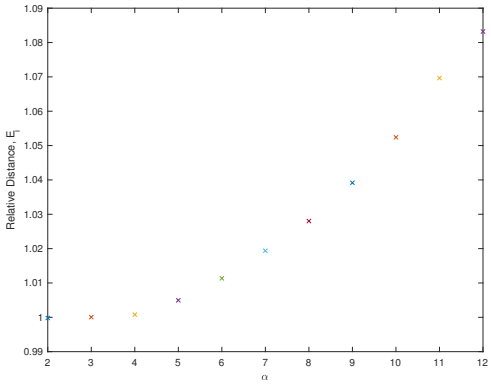


(b) Relative distance travelled, E

Figure 10: One vehicle, one obstacle, at various α values. $d_r = 1$m.



(a) Paths travelled.



(b) Relative distance travelled, E

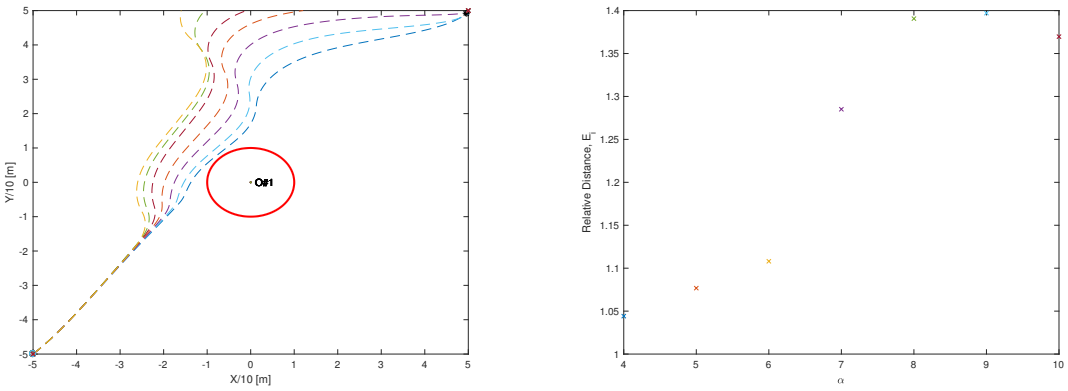
Figure 11: One vehicle, one obstacle, at various α values. $d_r = 5$m.

5.3.1 The effect of α—how an object impacts performance

Figure 10 shows the agent path and relative distance travelled for $d_r = 0.1$ $(1m)$, and $\alpha = 2, 4, \ldots, 12$. The agent avoided the obstacle in all trials, and the difference in relative distance travelled was marginal, implying α has no impact on the performance of the system for a small obstacle.

To see if α does indeed contribute, the same simulation was run with $d_r = 0.5$ $(5m)$, and $\alpha = 2, 4, \ldots, 12$, as above. Figure 11 shows the agent path and relative distance travelled, and shows that α does impact the performance of the system. As α increases, the vehicle travels further. An interesting observation from this simulation is that the path is significantly affected *after* the obstacle—with $\alpha = 4, 5$ appearing optimal, implying a minimum radius of contribution of 20 metres.

To observe this relationship further, the simulation was run with $d_r = 1$ (10m), and $\alpha = 4, 6, \ldots, 10$ (For $d_r > 1$ (10m), $\alpha > 4$ to avoid the obstacle). The results, shown in figure 12, exhibit the same characteristics after the obstacle, however, the effects after the obstacle are even more pronounced.



(a) Paths travelled.

(b) Relative distance travelled, E

Figure 12: One vehicle, one obstacle, at various α values. $d_r = 10$m.

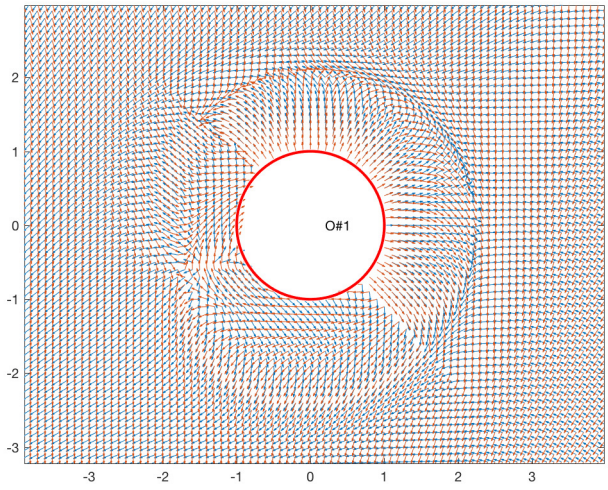


Figure 13: Partial vector fields in x and y direction, showing behaviour at d_c.

This again demonstrates that $\alpha = 4, 5$ gives desirable performance, implying a minimum radius of contribution of 40 metres.

One possible explanation for this behaviour is that the derivative vector fields are inconsistent (and almost discontinuous) along the boundary of contribution for each obstacle, as seen in figure 13. As such, if the vehicle enters this area, it will likely leave it "past" the obstacle, meaning the behaviour due to the control law is undesirable. Another explanation might have to do with the normalization of the vector fields in the algorithm.

For conservatism, the minimum radius of contribution for an obstacle will be the larger of $d_c = 20$ m, and $d_c = 5d_r$ m.

Table 3: Simulation Parameters for β-test.

Parameter	Description	Simulation Value
T	Simulation Time	20 s
δt	Simulation Step	0.05 s
D	Arena Size	100×100 m^2
δx	Arena Discretization	0.1 m
n	Number of Agents	2
X^0	Initial State	$(\pm 5, \pm 5)$ or $(\pm 50$ m$, \pm 50$ m$)$
X^f	Final Destination	$(\pm 5, \pm 5)$ or $(\pm 50$ m$, \pm 50$ m$)$
L_v	Vehicle Length	2.0 m
R_c	Vehicle Radius of Communication	20 m
R_s	Vehicle Radius of Sensing	0 m
d_{r_v}	Vehicle Radius	$1.5L_v$ m
d_{c_v}	Vehicle Radius of Contribution	βd_{r_c} m
m	Number of Obstacles	0
d_{r_o}	Obstacle Radius	–
d_{c_o}	Obstacle Radius of Contribution	–

5.3.2 The effect of β—how another vehicle impacts performance

After gaining insight into how an agent will react to an obstacle, the next test is examining how two vehicles interact with one another based on the β value. To do so, the vehicle must be allowed to communicate. Conservatively, this will be set at $R_c = 20$m. The simulation parameters for this test are outlined in table 3, and $\beta \in [0, 20]$. For performance, both metrics will be assessed.

As seen in figure 14, higher values of β only impact the path of the vehicle *after* they have passed, and cause the agent to travel further. At small β values, there is a marginal increase in the minimum passing distance between two agents, and when $\beta > 10$, it appears to be a polynomial increase. As a result, $\beta = 6$ seems ideal[1].

5.3.3 The effect of R_c (and R_c)

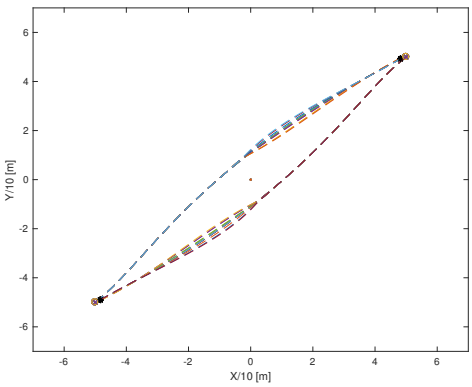
The simulation parameters to test the effect of R_c are as per table 3, however, $d_{c_v} = R_c \in [1, 30]$.

As seen in figure 15, as R_c increases the vehicles pass at a further distance from one another. Intuitively, this makes sense, as the vehicles "know" about each other earlier and are able to respond to the generated repulsive vector field earlier.

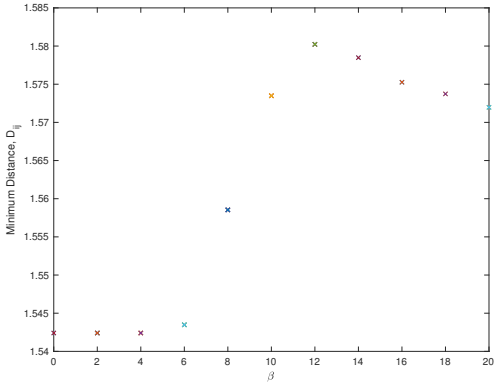
In these tests, a value of $R_c < 7$ had no impact on the minimum passing distance or relative distance travelled for each agent.

When $7 < R_c < 20$, the minimum passing distance increased but the relative distance travelled for each agent remained reasonably constant. This is the case where safety margins are increasing, but overall system cost (assuming the cost is energy used) is not increasing—an ideal situation.

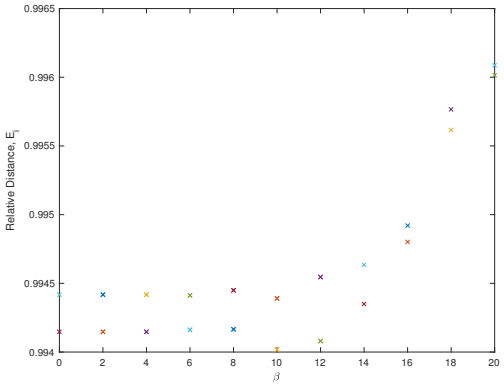
[1] If $\beta > \frac{R_c}{1.5L_v}$, $d_{c_v} > R_c$, which (practically speaking) does not make sense. That said, the impact of R_c on the performance of the system will be investigated in a similar manner.



(a) Paths travelled.

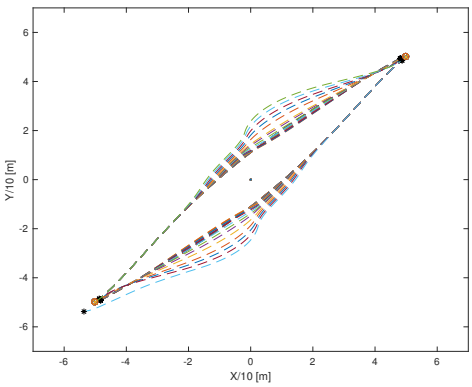


(b) Minimum passing distance, $D_{1,2}$

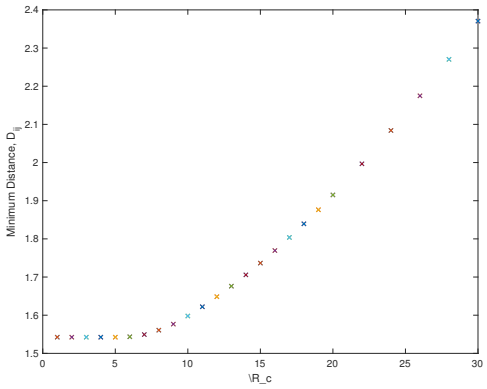


(c) Relative distance travelled, E_i

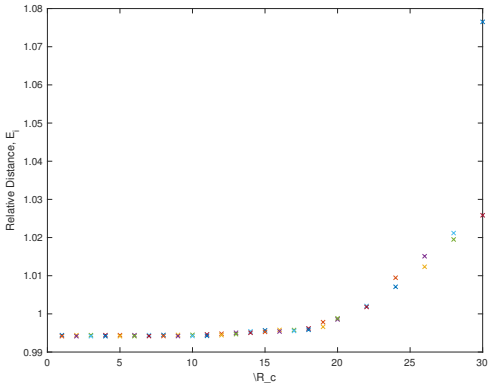
Figure 14: Two vehicles for various $\beta \in [0, 20]$ values. $d_{r_v} = 3$m.



(a) Paths travelled.



(b) Minimum passing distance, $D_{1,2}$



(c) Relative distance travelled, E_i

Figure 15: Two vehicles for various $R_c = d_{c_v} \in [1, 30]$ values. $d_{r_v} = 3$m.

A value of $R_c > 20$ increases the relative distance each agent travelled and significantly increases the minimum passing distance (almost quadratically).

Coincidentally, $R_c = 20$m was used used to test the effect of β on the system. Figure 14 shows how the system behaves with this value. As a result, a value of $R_c = 20$ will be used in the remaining trials. This is a realistic choice, as most communication devices are capable of ranges equal to or greater than this value.

A major limiting factor that results from this is the sensing device range. For R_s the same relationships for R_c apply, hence the results from figure 15 indicate $R_s \in [7, 20]$ are sufficient for the system to perform—a realistic range for practical sensor implementation (technology dependent).

One relationship that has not been investigated is $R_c = \eta d_{c_v}$ where $\eta \in \mathbb{R}_{>0}$.

5.4 Performance of a Group of Vehicles

This report has investigated how $\alpha, \beta, d_r, d_c, R_c, R_s$ individually impact system performance for one vehicle and one obstacle (α test), and for two vehicles (β test).

The simulations in this section look at various combinations of the number of agents, n, and the number of obstacles, m, with simulation parameters outlined in table 4. Note, the different (randomized) forward velocity control gains. This simulates vehicles that have different maximum velocities.

As this is purely based on random initial conditions, final destinations, and obstacle locations, it is not expected that all trials are successful. Some interesting cases will be discussed, along with potential explanations of the phenomena in section 5.4.3.

5.4.1 Results

Table 5 summarizes the results, with a "✓" denoting a successful trial (i.e., no crashes, no obstacles hit, etc.), and a "✗" denoting an unsuccessful trial (i.e., crashes). A successful trial where *all* vehicles reach their final destination is indicated with a "☮". A "–" denotes no simulation was performed.

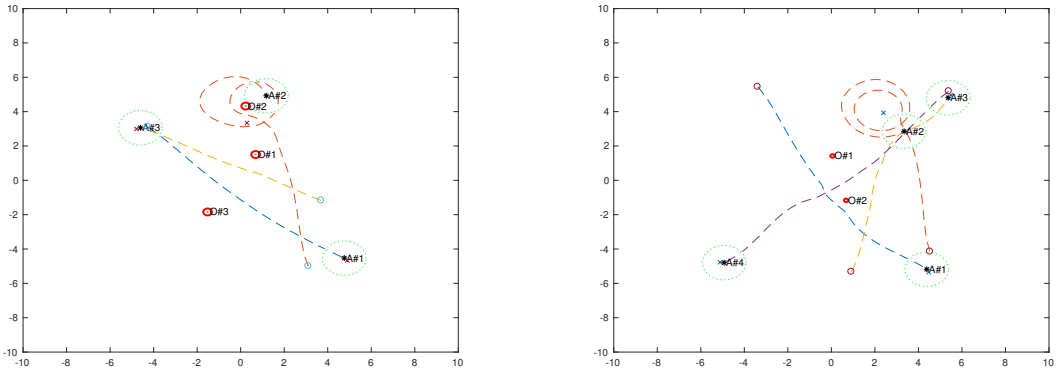
[2]Interesting Case #1. See section 5.4.3.
[3]Interesting Case #2. See section 5.4.3.
[4]Interesting Case #3. See section 5.4.3.

Table 4: Simulation Parameters for multi-agent and obstacle simluations.

Parameter	Description	Simulation Value
T	Simulation Time	30 s
δt	Simulation Step	0.05 s
D	Arena Size	100×100 m^2
δx	Arena Discretization	0.1 m
n	Number of Agents	n
X^0	Initial State	Randomized
X^f	Final Destination	Randomized
L_v	Vehicle Length	2.0 m
R_c	Vehicle Radius of Communication	20 m
R_s	Vehicle Radius of Sensing	10 m
d_{r_v}	Vehicle Radius	$1.5L_v$ m
d_{c_v}	Vehicle Radius of Contribution	$\beta d_{r_v} = 18$ m
k_v	Forward Velocity Control Gain	$[1, 1.5]$
k_ϕ	Steering Rate Control Gain	$[0.05, 0.1]$
m	Number of Obstacles	m
P_o	Obstacle Position	Randomized
d_{r_o}	Obstacle Radius	5 m
d_{c_o}	Obstacle Radius of Contribution	$\alpha d_{r_0} = 25$ m

Table 5: Simulation Results n, m-combinations.

No. Agents, n \ No. Obstacles, m	2	4	6	8	10
2	✇	✓	✓	✗	✇
4	✓	✓	✗	✗	✓
6	✗[2]	✇	✗	✓[3]	✗
8	✗	✗	✗	–	–
10	✓[4]	✗	✗	–	✗



(a) $n = 3$, $m = 3$.

(b) $n = 4$, $m = 2$.

Figure 16: Agent 2 circling its final destination.

5.4.2 Observations

After many simulations, it is observed that:

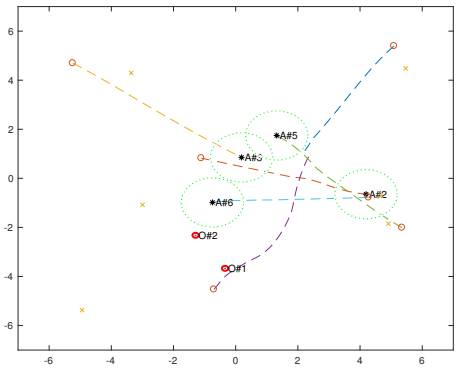
- The radius of contribution for an obstacle, d_{c_o}, has to be 20 m or greater (and then 4 or 5 times d_r as an object increases in size.
- If a vehicle passes close to an object while another vehicle is near it, it affects the *a priori* map, i.e, the agent "forgets" the obstacle is there. This sometimes leads to agents not reacting early enough to avoid the object. This is a function of the radius of communication, as it over-influences the system. From β testing, $R_c = 10$ was also sufficient, so this value was utilized in later simulations, although it did not prevent the phenomenon for all trials. A safeguard could be built into the code to account for this.
- With more obstacles, a control gain of $k_\omega = 0.1$ is too high. A value of 0.05, as tested earlier, still yields results, and is used for higher numbers of vehicles or obstacles. This is essentially placing less weight on the "original" path so the agent becomes more responsive to changes in the vector field.
- Agents approach their final destination and end up circling the coordinate. This is perhaps caused by the final destination being located within the radius of contribution of an object or the control gain k_v being too high. Examples are shown in figure 16. One could implement a "close" control scheme to account for this.

5.4.3 Interesting Cases

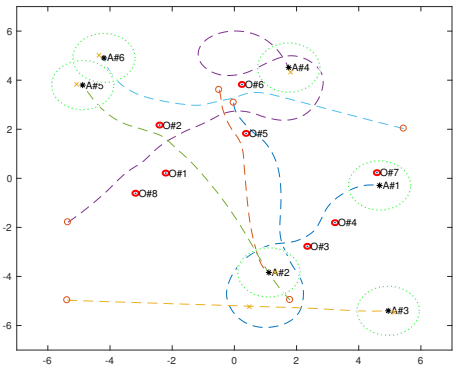
There are four interesting cases outlined below, with results given in figure 17:

1. With $n = 6$, and $m = 2$.

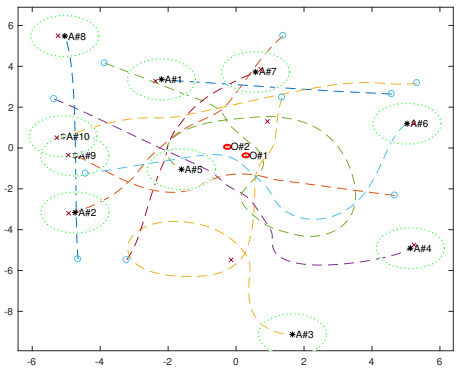
   The vehicles avoid the obstacles but are unable to avoid each other. Agents 1 and 4 collide. This is unusual as vehicles are generally good at avoiding each other. Perhaps the "wake" left by agent 5 impacted agent 1's and/or 4's ability to respond to disturbances.



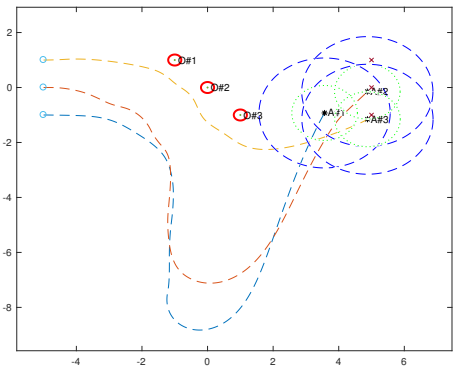
(a) Interesting Case #1. $n = 6$, $m = 2$.



(b) Interesting Case #2. $n = 6$, $m = 8$.



(c) Interesting Case #3. $n = 10$, $m = 2$.



(d) Interesting Case #4. $n = 3$, $m = 3$.

Figure 17: Interesting Case plots.

2. With $n = 6$, and $m = 8$

   All vehicles avoid the obstacles and reach their destination, except agent 1, who begins circling back but is then influenced by other obstacles which perturb it away from its circling path. It would be interesting to see where agent 1 finished if the simulation continued.

3. With $n = 10$, and $m = 2$

   All vehicles avoid the obstacles and reach their destination, except agents 3 and 5. Agent 5 is disturbed off track earlier in the simulation when avoiding all the other agents around the origin. Agent 3 is experiencing circling phenomenon around its final destination.

4. With $n = 3$, $m = 2$ and $R_c = 20$m (a slightly more contrived example).

   This phenomenon (described as "dragging") is when two (or more) vehicles influence their paths away from what is optimal. It usually takes one vehicle some time to recover and escape the influence of the other vehicle. The other vehicle also recovers and heads towards its final destination. This is more likely to occur as a vehicles radius of communication increases.

6 Conclusions and Future Work

One conclusion is that a successful result is dependent on the system's initial conditions and parameters. Through the modelling and simulation process, no explicit relationships or rules have been identified. Instead, the process yielded loose rules and trends, thus rendering this combination method of guidance, control and navigation using vector-fields unreliable for practical implementation (for now).

However, it has been shown that a vehicle's radius of communication does indeed impact system performance. Two major examples are vehicles can travel further than they have to (due to the dragging phenomenon, or a large value of R_c) or vehicles do not respond at all and collide (due to a smaller value of R_c). Hence, the choice of R_c is critical.

Also, it has been shown that obstacles need to contribute their repulsive vector field a minimum distance (20 metres in these simulations) to ensure vehicles are able to respond in a timely manner.

The work above is a building block to add greater complexity and explore other factors that impact the performance of the system. The model was kept as simple as possible, with no safeguards or general protocols built in to prevent collisions (other than those described regarding repulsive vector fields). For example, there are linear and angular velocity coordination protocols outlined in [3] that were not explored in this research.

The system's performance could be enhanced by implementing additional protocols. For example, ignoring a vehicle within the radius of communication if it is moving in the same direction at a faster speed, or in an opposite direction but sufficiently "far" away. Another protocol would be switching control gains depending on how close an agent is to its final destination.

Ideally, this would increase system reliability (i.e., no collisions) to a level where performance metrics can be assessed while modifying simulation parameters to determine which variables (if any) have a significant impact on the overall performance of the system.

This can be looked at in future work, as well as:

- implementing and analysing more detailed cost functions of different agents;
- differentiating agent properties, other than speed;
- modelling communication and sensing profiles realistically;
- refining the control Law;
- implementing control law on 4 degree of freedom vehicles;
- including state estimation from navigation sensors;
- including vehicle dynamics; and
- having obstacle properties and different features, i.e., can a wall be represented as a collection of point obstacles.

References

[1] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Applied Mathematics Series, Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[2] D. Panagou, H. G. Tanner, and K. J. Kyriakopoulos, "Control of nonholonomic systems using reference vector fields," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 2831–2836, Dec 2011.

[3] D. Panagou, "A distributed feedback motion planning protocol for multiple unicycle agents of different classes," *IEEE Transactions on Automatic Control*, vol. 62, pp. 1178–1193, March 2017.

[4] W. Dong, "Tracking control of multiple-wheeled mobile robots with limited information of a desired trajectory," *IEEE Transactions on Robotics*, vol. 28, pp. 262–268, Feb 2012.

[5] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, pp. 169–182, Feb 2005.

[6] L. N. Tan, "Distributed optimal integrated tracking control for separate kinematic and dynamic uncertain non-holonomic mobile mechanical multi-agent systems," *IET Control Theory Applications*, vol. 11, no. 18, pp. 3249–3260, 2017.

[7] J. Shercliff, *Vector Fields: Vector Analysis Developed Through Its Application to Engineering and Physics*. Cambridge University Press, 1977.

[8] B. Sethuraman, *Rings, Fields, and Vector Spaces*. Undergraduate Texts in Mathematics, New York-Heidelberg-Berlin: Springer Verlag, 1997.