Proceedings Paper:

Douthwaite, J.A. orcid.org/0000-0002-7149-0372, Zhao, S. and Mihaylova, L.S. orcid.org/0000-0001-5856-2223 (2018) A comparative study of velocity obstacle approaches for multi-agent systems. In: Proceedings of 2018 UKACC 12th International Conference on Control (CONTROL). Control 2018: The 12th International UKACC Conference on Control, 05-07 Sep 2018, Sheffield, UK. IEEE , pp. 289-294. ISBN 978-1-5386-2864-5

https://doi.org/10.1109/CONTROL.2018.8516848

A Comparative Study of Velocity Obstacle Approaches for Multi-Agent Systems

James A. Douthwaite, Shiyu Zhao and Lyudmila S. Mihaylova

***Abstract*— This paper presents a critical analysis of some of the most promising approaches aimed at geometrically generating reactive avoidance trajectories for multi-agent systems. Several evaluation scenarios are proposed that include both sensor uncertainty and increasing difficulty. An intensive 1000 cycle Monte Carlo analysis is used to assess the performance of the selected algorithms under the presented conditions. The Optimal Reciprocal Collision Avoidance (*ORCA*) method was shown to demonstrate the most scalable computation times and collision likelihood in the presented scenarios. The respective features and limitations of the algorithms are discussed and presented through examples.**

***Index Terms*— Collision avoidance, multi-agent systems, velocity obstacles, VO, RVO, HRVO, OCRA**

I. Introduction

Collision avoidance is a subject that has seen an increasing interest over the last decade with the growth of domestic and commercial robotics. Multi-agent systems are now required to navigate increasingly crowded and dynamic environments, often where inter-agent communication is unreliable. In addition to this, in systems composed of numerous physical agents, agents must be able to generate trajectories in order to avoid other agents and obstacles in the field.

Two principle classifications of reactive avoidance techniques can be drawn from the literature: 1) Cooperative, 2) Non-cooperative. In both cases assumptions are made based on the availability of the obstacle telemetry. Cooperative avoidance algorithms operate on the assumption of a unilateral communication system with explicit communication between each agent and obstacle. Non-cooperative approaches however, rely on obstacle telemetry sensed through an on board tracking system. While typically multi-agent systems rely on communicated information about its neighbours, we examine the scenario where agents are tasked with locally computing their trajectories to achieve a desired way-point. This is synonymous to scenarios where inadequate information is communicated or lost. This reduces the agent trajectory generation to a localised Sense, Detect and Avoidance (SDA) problem [1].

Previous approaches to solving the SDA problem include probabilistic modelling [2], conflict resolution interval and agent trajectory optimisation [3]–[6]. More classical approaches include designed potential fields as seen in [7] and numerous geometry based avoidance techniques [8], [9]. The concept of the *Collision Cone* and the *Velocity Obstacle* is introduced in [8], defining geometric regions as constraints on the agents feasible velocities at time t_{k+1}. Presentation of the collision scenario geometrically, given no prior knowledge or predictions, allows a resolution velocity to be found quickly and with minimal obstacle information.

Iterations of the *Velocity Obstacle* concept include the *Reciprocal Velocity Obstacle* [10], [11], which has been shown to reduce the trajectory oscillation by considering the reactive nature of avoiding agents. Variable acceleration obstacles are addressed under the notion of *Acceleration Velocity Obstacles* in [12]. *Hybrid-Reciprocal Velocity Obstacle* are introduced in an effort to eliminate direction ambiguity in [13] and eliminate the phenomenon known as the *reciprocal dance*. Despite producing smoother trajectories, the *HRVO* is not capable of guaranteeing that trajectories will be smooth. A method proposed to address this is the Optimal Reciprocal Collision Avoidance method, by adopting the concept of half-planes as linear constraints [14]. Similar techniques demonstrate consideration for non-linear obstacle motion, proposed in [15], with the addition of kinematic constraints in the *Kinematic Velocity Obstacles (KVO)* in [16]. Although these approaches have been widely used in multi-agent systems such as pedestrian modelling and small robotic systems, they face challenges in symmetric scenarios where a phenomenon known as *Dead-lock* can occur.

This paper presents an in depth analysis of the most promising models and approaches for multi-agent collision avoidance. These approaches are studied over a range of scenarios with varying levels of difficulty and obstacle numbers. Through an intensive Monte Carlo analysis the pros and cons of these algorithms are demonstrated and discussed. These are evaluated in the light of a minimum separation distance and computation time, and can be applied both to unmanned aerial vehicles and air traffic control. Two dimensional collision avoidance is considered, although the extension to the three dimensional case is natural.

The structure of the paper is as follows; In Section II we introduce the problem context and the imposed sensor constraints. Section III presents the mainstream *Velocity Obstacle* approaches to collision avoidance and their principle differences. Section IV presents the agent assumptions and conditions used in the context of this paper. In Section V the performance of the presented algorithms is assessed and compared in several given example scenarios. Finally, the results of the proposed experiments are presented and discussed

The authors gratefully acknowledge the support from the UK EPSRC under grant number EP/M506618/1. James A. Douthwaite, Shiyu Zhao and Lyudmila S. Mihaylova are with the Department of Automatic Control and Systems Engineering, The University of Sheffield, UK. `jadouthwaite1, zhao, l.s.mihaylova@sheffield.ac.uk`.

OpenMAS Matlab® simulation environment available at `https://github.com/douthwja01/OpenMAS`

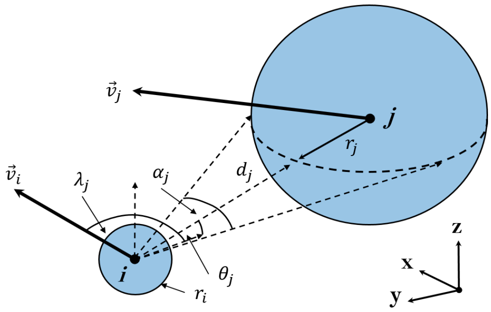

Fig. 1. A description of the adopted sensor model defining the spherical position of obstacle j, at time k, as its position in the azimuth d_j,θ_j and λ_j.

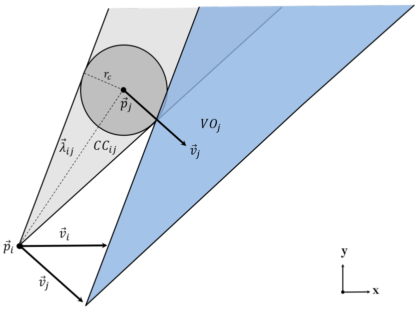

Fig. 2. The Velocity Obstacle VO_j (shaded in dark grey) from the initial CC_{ij}. Here the VO_j defined in the configuration space of i, from the relative position λ_{ij}, configuration radius $r_c = r_i + r_j$ and velocity $\vec{v}_j$.

in Section VI.

II. Problem Description

We begin by considering an interaction between two agents, *i* and *j*, respectively. Both agents are moving through two dimensional (2D) Cartesian space. The agents velocities are denoted by $\vec{v}_i \in \mathbb{R}^{3\times1}$ and $\vec{v}_j \in \mathbb{R}^{3\times1}$, with representative radii r_i and r_j, respectively. Agent *i* considers agent *j* as both a collaborator and an obstacle to be avoided. The position of agent i at time t_{k+1} is defined as $\vec{p}_{i,k+1} = \vec{p}_{i,k} + \Delta t \cdot \vec{v}_{i,k}$ where Δt is the sampling rate. We define a maximum speed constraint v_{max} to limit the velocities available to the avoidance routine; represented simply as $|\vec{v}_i| <= v_{max}$.

A. Sensor Model

We assume that each agent is able to make its own observations of its surrounding using an on board camera and range finder. The resulting measurements represent the spherical position of agent *j* in the form of an elevation, azimuth angle, range and width, denoted by $\theta_j \in [-\pi, \pi]$, $\lambda_j \in [-\pi, \pi]$, $d_j \in [0, d_{max}]$ and $\alpha_j \in [-\pi, \pi]$ respectively. The parameter d_{max} is used here to describe the maximum visual range of agent i. Agent *i* observes agent *j* in its body axes as seen in Figure 1 [3].

For the context of this paper we assume avoidance is to be carried out at constant altitude. Agent i measures the spherical position, $\theta_{j,k}, \lambda_{j,k}, d_{j,k}$, and width $\alpha_{j,k}$ in the body axes of i. The agent then computes its equivalent 2D Cartesian position $\vec{p}_{j,k} = [x_{j,k}, y_{j,k}]^T$ and radius estimate $r_{j,k}$ at time k. The Cartesian velocity of the obstacle is then be calculated from successive position samples $\vec{v}_{i,k} = \frac{1}{\Delta t}(p_{j,k} - p_{j,k-1})$. The relative state of obstacle j, at time t_k, is then defined as $X_{j,k} = [p_{j,k}, v_{j,k}, r_j]^T$. The agents are otherwise assumed capable of retaining a set of states that correspond to all obstacles with in the agent's visual horizon d_{max}.

III. Velocity Obstacle Methods

A. The Velocity Obstacle

The concept of the *Velocity Obstacle (VO)*, based on the geometric assembly of the *Collision Cone (CC)*, was first presented in [8]. Obstacles are observed in the agents local horizontal plane (XY) as their planar cross-section centred at $\vec{p}_j$ as seen in Figure 2. Here, the collision cone for obstacle j is defined as CC_{ij} from the geometric properties of the obstacles relative position $\vec{\lambda}_{ij}$, configuration radius r_c and velocity $\vec{v}_j$.

Velocities that will bring about collision with obstacle j are then be represented in the velocity space by translating CC_{ij} by $\vec{v}_j$ via the Minkowski sum: $VO_{ij} = CC_{ij} \oplus \vec{v}_j$. In the consideration of multiple obstacles, the union of multiple $VO_{1:n}$ is taken. Agent velocities are therefore considered valid if $\vec{v}_{i,k+1} \notin VO_k = \cup_{j=1}^{n} VO_{j,k}$ [8]. Velocities satisfying this constraint describe a collision free trajectory for agent i in the presence of obstacles $VO_{j=1:n}$ for time t_k.

In practice, oscillatory trajectories are often observed in instances where two agents attempt to resolve a conflict with one another using the *VO* method. This often propagates until the point of collision occurs; as the two agents repeatedly resolve velocities $\vec{v}_{i,k+1}$ that imply a new conflict at t_{k+1} [10]. Obstacles that are static, or moving with constant velocity can otherwise be handled using the *VO* approach.

B. The Reciprocal Velocity Obstacle

An iteration of the conventional *VO* method, termed the *Reciprocal Velocity Obstacle (RVO)* [10], attempts to consider the reciprocal motion of the second decision making agent *j* in order to produce smoother avoidance trajectories. The agent generates a *VO* with an apex augmented by the average of the two object velocities $\vec{v}_{i,k+1} \notin CC_{ij} \oplus (\vec{v}_{i,k} + \vec{v}_{j,k})/2$. This concept effectively allows the agent to mediate its correction trajectory $\vec{v}_{i,k+1}$ in accordance with $\vec{v}_j$. At time t_k, the *RVO* contains represents the region of velocities for i that are the average of both the velocity of agent i and the velocity of obstacle j.

The *RVO* is shown to eliminate the VO oscillation mentioned in Section III-A [10], and the resultant resolution trajectories are seen to be smoother. While this is the case, agent i and obstacle j do not explicitly agree on which sides they will approach each other. This can lead to scenarios where agents will mirror the trajectories of their respective obstacles in an attempt to avoid them. The oscillations induced by this behaviour, distinct from those of the *VO*, are often referred to as a *Reciprocal Dance*.

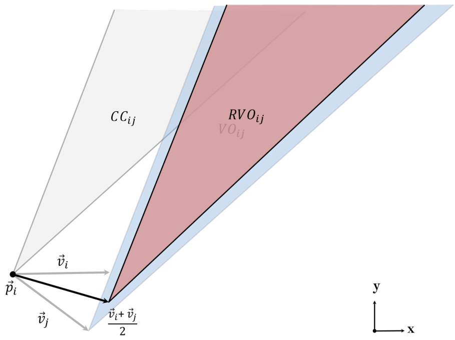

Fig. 3. The construction of the Reciprocal Velocity Obstacle (RVO_j) by averaging the velocities of the agent $\vec{v}_i$ and obstacle $\vec{v}_j$.

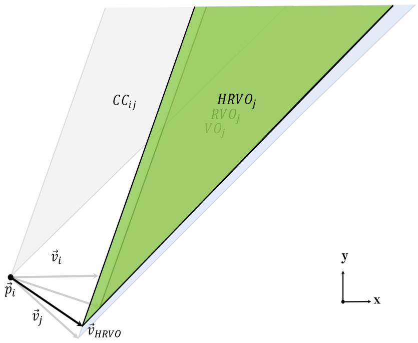

Fig. 4. The relation of the Hybrid Reciprocal Velocity obstacle *HRVO* to the initial VO_j and the RVO_j for a given obstacle B.

C. The Hybrid Reciprocal Velocity Obstacle

An advancement on the *VO* problem has been proposed to negate the causes of reciprocal dance by augmenting the *VO* and *RVO* regions. The *Hybrid Reciprocal Velocity Obstacle (HRVO)*, shown in Figure 4, alters the apex of the *HRVO* in order to example different behaviour depending on the relative motion of the obstacle $\vec{v}_j$.

The centreline of the VO_j and RVO_j are collinear in nature, therefore if the obstacle is moving right, the agent should resolve a trajectory $\vec{v}_{i,k+1}$ to pass the obstacle on the left and vice versa. Failure to do so brings about the phenomena of the *Reciprocal Dance*. Although the method is shown to improve the generation of smooth avoidance trajectories, it cannot guarantee it theoretically [13]. In the example given in Figure 4, directional bias is established by adjusting the apex of the $HRVO_j$ to be the intersection of the leading edge of RVO_j the trailing edge of VO_j (i.e. $HRVO_{ij} = CC_{ij} \oplus \vec{v}_{HRVO}$. The resulting constraint set imposed upon agent i at time t_k is then written $\vec{v}_{i,k+1} \notin HRVO_k = \cup_{j=1}^{n} HRVO_{i,k}$ [13].

Typically the *RVO* and *HRVO* are only necessary in the computation of inter-agent avoidance trajectories. The global *VO* set for agent i can instead be written as the union of the reciprocal variants (*RVO* or *HRVO*) for surrounding agents A_j and the *VO* for obstacles O_j: $\vec{v}_{i,k} \notin HRVO_k = \bigcup_{A_j=1}^{n} \mathrm{HRVO}_{A_j} \cup \bigcup_{O_j=1}^{n} \mathrm{VO}_{O_j}$.

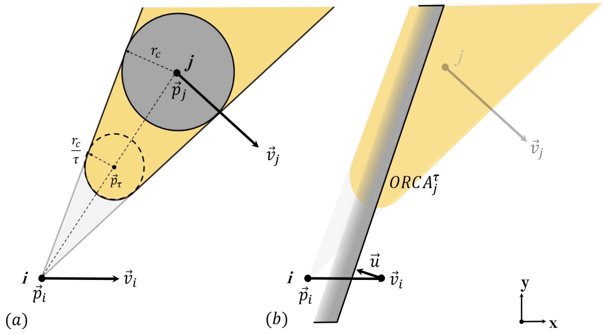

Fig. 5. a) The geometric description of the truncated *VO* for obstacle *j*, defined by the truncation parameter τ, relative position ($\vec{p}_j - \vec{p}_i$) and configuration radius $r_c = r_i + r_j$. b) The assembled ORCA obstacle and velocity correction $\vec{u}$ as a result of obstacle *j*.

D. Optimal Reciprocal Collision Avoidance

The *RVO* concept has be extended more recently in a method termed *Optimal Reciprocal Collision Avoidance (ORCA)*. The ORCA approach is described well in [17], demonstrating how the ORCA velocity obstacle is formulated for a given reciprocally collision avoiding agent pair *i* and *j*. The resultant trajectory is not only smooth but, for small time steps, can be seen as continuous in the velocity space. The truncation parameter, τ, represents the time window for which a collision free trajectory should be guaranteed, i.e the agent can move at its new velocity for τ seconds.

If we assume that $\vec{v}_i$ and $\vec{v}_j$ are those that will bring about a collision in the future, then we define $\vec{u}$ as the vector to the point closest to the boundary of VO_j: $\vec{u} = (\arg\min_{\vec{v}\in\delta\mathrm{VO}^\tau} ||\vec{v}-(\vec{v}_i-\vec{v}_j)||)-(\vec{v}_i-\vec{v}_j)$(see Figure 5). Here $||\vec{v}||$ denotes the euclidean norm of $\vec{v}$. Using the "outward" facing normal $\vec{n}$ of the boundary at the point $(\vec{v}_i - \vec{v}_j) + \vec{u}$ and the assumption that the responsibility that the avoidance is shared equally, the formulation for the $ORCA_j$ constraint can be written as $ORCA_k^\tau = \vec{v}|\vec{v} - (\vec{v}_i + \frac{1}{2}\vec{u}).\vec{n} \geq 0$. The geometric representation of $\vec{v}$ is given in Figure 5(b). Here it is represented as a *half-plane* with normal $\vec{n}$, with the initial point at $\vec{p} = \vec{v}_i + \frac{1}{2}\vec{u}$ [17].

The *ORCA* lines themselves allow the scenario to be described using only linear constraints. In addition, representation of the *RVO* as half-planes allows for simplification of the constraint set by eliminating those already covered by other *ORCA* lines, whilst guaranteeing continuously smooth agent trajectories.

E. Trajectory Selection

How the optimal resolution velocity is determined from the constraint sets defined in Sections III-A-III-D, is also subject to strategy [8]. In the literature this is typically determined by considering the minimum deviation from a desired trajectory $\vec{v}_i^{pref}$ subject to the union of the VO_k set. In such cases the optimal velocity can then be expressed as $\vec{v}_i^* = \arg\min_{\vec{v}\notin VO}(||\vec{v} - \vec{v}_i^{pref}||)$. In this paper, the optimal resolution velocity is determined similarly, using the *Clear*

Path method [17], subject to the global constraint set of a given algorithm.

IV. Agent Kinematics & Control

A. Way-point Navigation

In this case study, way-points are used to both ensure contradictory trajectories and to indicate task completion. At all times, the position of agent i's way-point $\vec{p}_{wp,i}$ is assumed observable in its surroundings. The preferred velocity is that in the direction of $\vec{p}_{wp,i}$, expressed as $\vec{v}_i^{pref} = \frac{\vec{p}_{wp,i}-\vec{p}_i}{||\vec{p}_{wp,i}-\vec{p}_i||} \cdot v_{pref}$ where v_{pref} is the preferred speed.

B. Neighbour Consideration

For the purposes of this paper it is assumed that the agents have an infinite visual horizon. This allows the agents to observe the trajectories of the complete agent set, and their target way-points. To limit the number of constraints (and therefore complexity of solution) a local neighbourhood is adopted using the maximum separation d_{max} stated in Table I.

V. Experimental Results

A. A Problem of Symmetry

In collision scenarios involving greater than two agents, there exists a problem of symmetry. This occurs when the obstacle configuration is perfectly symmetrical about the agents velocity vector $\vec{v}_i$. Similar to the *Dead-lock* scenario in [13]; no feasible solution can be found either because of this symmetry, or because the velocity space is saturated with *VO*. Despite this scenario being unlikely in real world applications, the agent is incapable of resolving an avoidance heading without violating or relaxing a given constraint.

In such scenarios a higher level strategy must be applied to intelligently preserve a collision-free trajectory by manipulating the constraint (or *VO*) set or designing a new desired velocity ($\vec{v}_{pref}$). As part of the Monte Carlo analysis, the initial positions of the agents are perturbed by a small noise signal $0.5m$. This process also aids in the prevention of the dead-lock by ensuring that the scenario is asymmetrical.

B. Experimental Conditions

In this section we demonstrate the conflict resolution methods outlined in Section III. The agent population is initialised with the parameters defined in Table I. The noise parameters are applied to better represent sensor-derived measurement of the obstacle trajectory. Agents are designated a target way-point at the antipodal position of a concentric circle with a radius of $20m$. The agents are tasked with crossing the circle to reach their way-point positions $\vec{p}_{wp,i}$ whilst avoiding collision. In Figure 6 the agent initialise at their origins (circles) and move through the collision centre to reach their respective way-points (triangles).

Events such as collisions or way-point incidence are said to occur when the following condition is violated $||\vec{p}_i - \vec{p}_{wp,i}|| < (r_i + r_{wp,i}) - K_{tol}$, where the parameter K_{tol} is a condition tolerance that aims to eliminate ambiguity between collisions and narrow-misses caused by the nature of discrete simulation.

Parameter	Value
Maximum speed	2 m/s
Agent critical radius	0.5 m
Neighbour horizon	15 m
Camera standard deviation	5.208×10^{-5} rad
Range-finder standard deviation	0.5 m
Airspeed standard deviation	0.5 m/s
Position standard deviation	0.5 m
Agent orbital radius	10 m
Way-point orbital radius	20 m
Object position standard deviation	0.5 m
Cycles	1000
Sampling rate	0.25 s
Event tolerance (Way-points, Collisions etc..)	1×10^{-3} m

TABLE I

The unilateral agent parameters, including assumptions on sensor uncertainty.

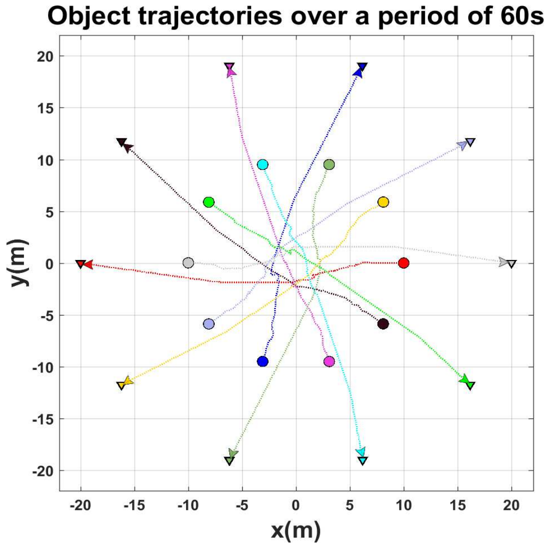

Fig. 6. A depiction of ten agents using the *VO* based reactive avoidance in a concentric collision scenario. The oscillations due to obstacle compensative motion can be clearly observed as the agent progress towards the collision centre.

The agent and scenario parameters are otherwise explicitly stated in Table I.

The selected algorithms presented in Section III were placed in scenarios with an increasing numbers of agents. We examine the ten agent scenario to discuss the principle difference in algorithm behaviour. Figure 6 demonstrates the trajectories generated by the *VO* algorithm. When compared to the *RVO* in Figure 7 the trajectory adjustments can be seen to be abrupt, with greater oscillation throughout, until all conflicts are resolved.

The compensation for obstacle movement is clearly seen in Figure 7 as the trajectories are shown more gradual. This indicative of the adjustment of the *RVO* in response to the movement of the obstacles; leading to fewer instances of harsh correction. Oscillation in the form of *Reciprocal Dance* can still be observed however as the direction of pass is resolved.

In comparing the *RVO* trajectories to that of the *HRVO*

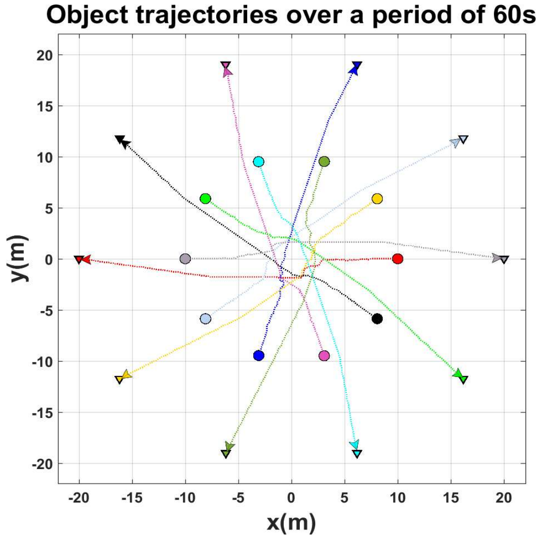

Fig. 7. A depiction of the ten agent concentric scenario and applying the *RVO* based avoidance method. Abrupt trajectory changes can be seen observed, with distinct oscillations as new agents j enter the visual horizon of agent i.

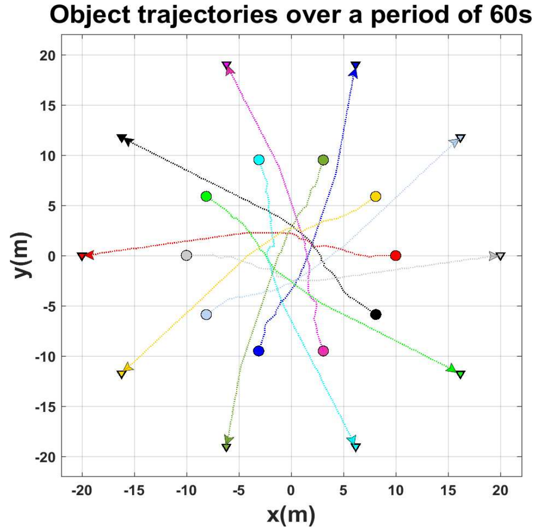

Fig. 8. The ten agent concentric scenario repeated with the *HRVO* obstacle generation method applied. Oscillations can be observed as the procedure begins, however shown to be near linear as the direction of pass is resolved.

in Figure 8; their is a clear reduction in the oscillation as the agents initially determine their direction of pass. The *HRVO* directional bias can also be observed from the agent trajectories, indicated by the emergent spiral behaviour around the conflict centre.

The representation of the *VO* as *ORCA* constraints is shown to produce trajectories similar to that the of *HRVO* in Figure 9. The linearity of the of the constraints however is shown to create smooth trajectories throughout the conflict scenario, resulting in a smaller overall course deviations.

The selected algorithms were exampled in scenarios with 2, 5, 10 and 20 agents and their performance measured over 1000 Monte Carlo independent iterations. In addition to this, two sensor conditions were observed; A) *Ideal Sensing*; the agents are given perfect knowledge of the surrounding obstacles B) *Representative Sensing*; the agents adopt the sensor properties defined in Table I.

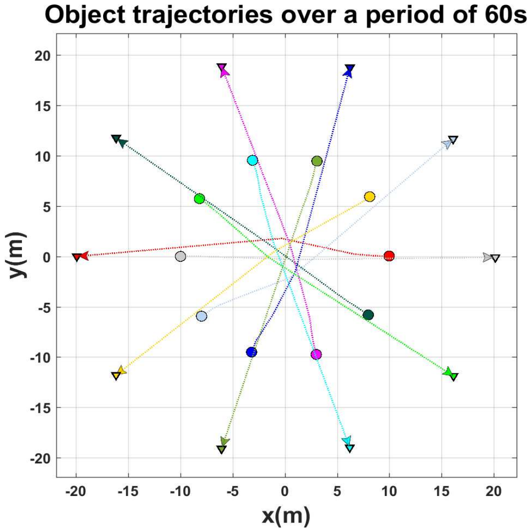

Fig. 9. The ten agent concentric scenario repeated under the *ORCA* obstacle generation method. The resultant trajectories appear as smoother, more gradual adjustments than the previous methods.

Algorithm Condition	Mean Collisions	Mean Minimum Separation (m)	Mean Computation Time (ms)
Condition A			
VO	9.203	0.581	2.000
RVO	3.140	0.831	2.100
HRVO	0.053	0.996	2.400
ORCA	0.038	1.000	0.460
Condition B			
VO	7.749	0.624	2.000
RVO	9.380	0.577	2.100
HRVO	2.878	0.836	2.600
ORCA	6.881	0.757	0.463

TABLE II

ALGORITHM PERFORMANCE OF IN THE SAME 10 AGENT SCENARIO. CONDITION A) SENSING CAPABILITIES ARE ASSUMED IDEAL, CONDITION B) ASSUMING REPRESENTATIVE SENSING. EACH VALUE REPRESENTS THE MEAN ACROSS 1000 INDEPENDENT MONTE CARLO ITERATIONS.

The mean behaviour of the presented approaches are shown in Tables II, where a clear difference can be seen between the *Ideal* and *Representative* sensing conditions during the 10 agent example scenario. Under the assumptions of ideal obstacle telemetry, the compensative nature of the *RVO* is shown to reduce the mean number of collisions to 31.40%. This is a significant reduction from the 92.03% achieved in same scenario using the original *VO* method. The innate directional bias in the formation of the *HRVO* is shown to further reduce the number of collisions to 0.53%. The lowest mean number of collisions was however found using the *ORCA* method; averaging 0.38% over 1000 iterations.

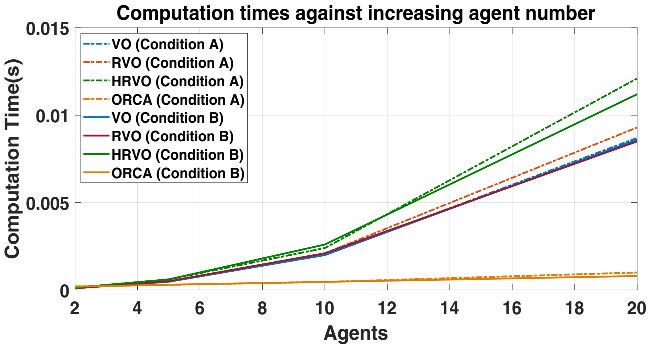

Fig. 10. The mean algorithm computation times in both condition A) Ideal obstacle knowledge is assumed B) Obstacle telemetry data is subject to interference. Their effect on computation time is observed with an increasing number of obstacles.

Observing the behaviour of the algorithms in the presence of sensor uncertainty demonstrated a 5.08% mean increase in computation time. This can be seen more clearly in Figure 10. The *RVO* is shown to be sensitive to obstacle trajectory uncertainty; with a factor of 3 increase in mean collision likelihood across the 1000 iterations. This may be due the aggravation of the reciprocal corrections (*Reciprocal Dance*) by the uncertainty in obstacle trajectory. Similar behaviour can also be observed for the *ORCA* algorithm, as the sensor uncertainty is shown to significantly increase the likelihood of collision under this regime also.

The *ORCA* algorithm was also shown to have achieved a mean minimum separation closest to the desired $1m$. This suggests the *ORCA* algorithm was more consistent in its ability to maintain the intended boundary condition. Although, in considering noisy telemetry this resulted in a mean collision likelihood 40.03% higher than the *HRVO* approach.

Studying Figure 10, we observe an exponential relationship between the size of the agent population and the mean algorithm computation time for the *VO*, *RVO* and *HRVO* methods. The *ORCA* approach is however shown to benefit greatly from the linear representation of the constraint set; computation time is shown to scale linearly with increasing agent number. The relationship between the performance reduction rate $r_{ORCA} = 3.4 \times 10^{-5} s/n$ is shown to be distinctly lower than the other presented approaches. The *ORCA* algorithm therefore has a clear advantage when considering scalability for larger multi-agent systems, abeit more susceptible to uncertainty than the *HRVO* method. *All analyses were completed using an Intel Core i7-6600HQ quadcore (@2.8GHz) CPU. Code for the presented algorithms and scenarios are also available on Github [18].*

VI. Conclusions

In this paper several well established approaches to non-cooperative collision avoidance are presented for use in multi-agent systems. Uncertainty in obstacle trajectory is shown to increase the mean computation time of all the proposed approaches by without compensative measures. The *HRVO* and *ORCA* methods are shown to be more effective in both negotiating dense environments without collision, and handing obstacle trajectory uncertainty. The *ORCA* method is also shown to generate both smoother resolution trajectories and scalable mean computation times.

The presented algorithms have shown that reactive collision avoidance can be sufficient to mitigate multiple collisions in a communication denied environment. Further work into inherent avoidance will examine such algorithms in the presence static and dynamic obstacles in more sophisticated coordinated tasks.

References

[1] A. Zhahir, A. Razali, and M. Ajir, "Current Development of UAV Sense and Avoid System," in *Proc. of the IOP Conference on Materials Science and Engineering*, vol. 152, Universiti Putra Malaysia, 2016.

[2] S. Ramasamy and R. Sabatini, "A Unified Approach to Cooperative and Non-Cooperative Sense-and-Avoid," in *Proc. of the International Conference on Unmanned Aircraft Systems, ICUAS 2015*. IEEE, 2015, pp. 765–773.

[3] J. Douthwaite, A. D. Freitas, and L. Mihaylova, "An interval approach to multiple unmanned aerial vehicle collision avoidance," in *Proc. of the 11th Symposium Sensor Data Fusion: Trends, Solutions, and Applications, SDF 2017*. IEEE, September 2017.

[4] J. Marin, R. Radtke, D. Innis, D. Barr, and A. Schultz, "Using a genetic algorithm to develop rules to guide unmanned aerial vehicles," in *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, vol. 1, 1999, pp. 1055–1060.

[5] A. Richards and J. How, "Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming," in *Proc. of the American Control Conference*, vol. 3, no. 2, 2002, pp. 1936–1941.

[6] H. Chen, V. Jilkov, and X. Li, "On threshold optimization for aircraft conflict detection," in *Proc. of the 18th International Conference on Information Fusion, Fusion 2018*. IEEE, 2015, pp. 1198–1204.

[7] M. Suzuki and K. Uchiyama, "Three-Dimensional Formation Flying Using Bifurcating Potential Fields," in *Proc. of the AIAA Guidance, Navigation and Control*, vol. 22, 2009, pp. 1753–1759.

[8] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[9] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized Velocity Obstacles," in *Proc. of the International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 5573–5578.

[10] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation," in *Proc. of the International Conference on Robotics and Automation*. IEEE, 2008, pp. 1928–1935.

[11] V. G. Santos, M. F. Campos, and L. Chaimowicz, "On segregative behaviors using flocking and velocity obstacles," in *Distributed Autonomous Robotic Systems*. Springer, 2014, pp. 121–133.

[12] J. van den Berg, J. Snape, S. Guy, and D. Manocha, "Reciprocal Collision Avoidance with Acceleration-Velocity Obstacles," in *Proc. of the International Conference on Robotics and Automation*, 2011, pp. 3475–3482.

[13] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "The Hybrid Reciprocal Velocity Obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, 2011.

[14] M. Kim and J. Oh, "Study on Optimal Velocity Selection using Velocity Obstacle (OVVO) in Dynamic and Crowded Environment," *Autonomous Robots*, vol. 40, no. 8, pp. 1459–1470, 2016.

[15] Z. Shiller, F. Large, S. Sekhavat, and C. Laugier, "Motion Planning in Dynamic Environments: Obstacles Moving Along Arbitrary Trajectories," in *Proc. of the International Conference on Robotics and Automation*, vol. 4, Grenoble, France, 2001, pp. 3716–3721.

[16] J. Wilkerson, J. Bobinchak, M. Culp, J. Clark, T. Halpin-Cha, K. Estabridis, and G. Hewer, "Two-Dimensional Distributed Velocity Collision Avoidance," Naval Air Warfare Center Weapons Division, Tech. Rep. February, 2014.

[17] J. Van Den Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer Tracts in Advanced Robotics, 2011, vol. 70, pp. 3–19.

[18] J. Douthwaite, L. Mihaylova, and S. Zhao, "Matlab® open source multi-agent simulator (openmas)," https://github.com/douthwja01/OpenMAS, 2018.