

Velocity Templates for Dense Swarms of Flying Robots

W. Venterie



Velocity Templates for Dense Swarms of Flying Robots

by

W. Venterie

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday July 6, 2017 at 09:30 AM.

Student number: 4012976
Project duration: October 1, 2015 – July 6, 2017
Thesis committee: Prof. dr. ir. Q. Chu, TU Delft, supervisor
dr. G. C. H. E. de Croon, TU Delft
B. D. W. de Croon, TU Delft
dr. J. Guo, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

This thesis is comprised of two parts: a paper with similar title as the thesis submitted to the IEEE Transactions of Robotics special issue on Aerial Swarming, and a literature study into the previous attempts at, and literature on, aerial swarming. The literature study is not a part of the thesis but is included for context on referenced literature and background information.

The object of the thesis is to realise a vision based autonomous quadcopter swarm capable of operating at close proximity and extend on existing literature in order to improve implementation and increase controllability from high level directives.

The ability to control autonomous dense aerial swarm from high level objectives and their ability to resolve conflicts with sub-groups or other swarms with conflicting objectives could in the near future open up the application of aerial swarms for many different applications.

This thesis contributes to the literature on autonomous swarming on unmanned aerial vehicles by providing a demonstrated effective approach. It is unique due to the application of only vision for detection and being effective when scaling and at close proximity.

Experiments indoor have shown the improved performance of the developed approach and a proof of concept experiments have been conducted outdoors. Unfortunately issues with outdoor stable flight of the chosen platform prevented the implementation to be tested at large scale outside.

*W. Vlenterie
Delft, July 2017*

Velocity Templates for Dense Swarms of Flying Robots

Wilco Vlenterie, Guido C.H.E. de Croon, Bart D.W. Remes, Qiping Chu

Abstract—In the near future many tasks could be performed by swarms of flying robots. To successfully implement multiple of these swarms in the same airspace they will have to be decentralised, autonomously cope with high densities and even resolve conflicting objectives of other swarms, while remaining controllable by operators through high-level objectives. This article introduces a novel swarming approach dubbed "Velocity Templates" based on artificial potential fields. These global fields represent the objectives of the swarm, which are balanced with local interaction. Different fields are considered leading to still or sustained motion swarms where conflicting objectives between sub-groups or multiple swarms are gracefully resolved. The approach is implemented on groups of 2 and 4 Parrot Bebop UAVs, using an efficient on-board vision algorithm to locate neighbours and a motion tracking system for guidance. The experiments show promising results for further outdoor tests assessing the scalability of the proposed approach.

Keywords—Aerial Robotics, Distributed Robot Systems, Swarms, Autonomous Agents.

I. INTRODUCTION & GOALS

For decades the complex emergent behaviour and agent-level control of flocks and schools have intrigued biologists and engineers alike. Flocks of sandhill cranes, dunlins and starlings were among the earliest aerial swarms extensively studied [1], [2], [3]. Computer simulations and further studies unravelled much of the mysteries surrounding these flocks and made swarming a topic of interest [4], [5], [6], [7].

For flying robots, which are the focus of this article, swarming has obvious advantages as many robots can explore a given region more quickly or observe multiple targets at the same time [8], [9]. Swarming is also useful for groups of UAVs heading to the same destination to restrict interference with the local airspace.

Some of the more well-known aerial robot swarms are choreographed, where every agent motion is pre-planned, as in the recent impressive show performed by Intel with hundreds of UAVs [10]. However, this approach is complex, offers none of the flexibility required by real-world swarming missions, and is mainly meant for visual appeal.

In order to render large swarms of UAVs practical, they will have to satisfy four requirements. First they will have to operate at close separation in order to limit interference with the local airspace. Second they have to operate in a

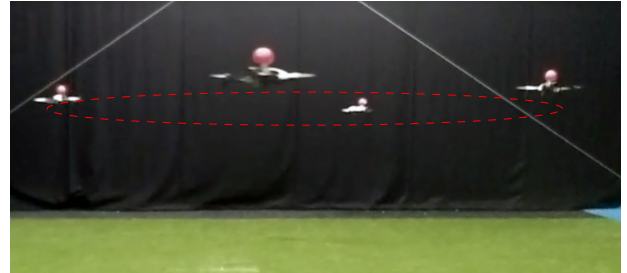


Figure 1: Photograph of 4 flying robots that use an efficient on-board vision algorithm and Velocity Templates developed for Dense Swarms. The photograph depicts a swarm performing circular sustained motion at 1.5m separation. Desired flight path shown in red.

decentralised fashion in order to account for scalability. Third they have to respond to high level centralised objectives to render control practical. Last they have to be able to resolve conflicts with other swarms with conflicting objectives.

To the best of our knowledge, no research has yet been able to satisfy these requirements for aerial swarms. On the one hand there are centralised approaches, which allow for collaboration but rely on a single central computer to compute and distribute commands [11], [12]. This leads to obvious scaling issues and creates a single point of failure, which is clearly undesired.

On the other hand there are decentralised approaches, where each robot interacts locally with its environment often while trying to achieve a global goal, which do not suffer from these scalability issues. A major difficulty for decentralised flying robots is estimating each other's relative positions. This has been tackled in various ways, but often heavily relying on inter-agent communication [13], [14], [15], [16]. A reliance on communication is a risk, as it suffers from bandwidth constraints when scaling up. Finally, many swarm approaches mimic biological swarms, which actually makes it harder for a human designer to create new global behaviours and complicates their control. For example, in a typical application of evolutionary swarm robotics [17], each new behaviour requires a new optimisation, which is not guaranteed to succeed.

The main contribution of this article is the introduction of a novel approach of UAV swarming dubbed "Velocity Templates" satisfying the four above mentioned requirements, and a demonstration of this approach on computationally

The authors are with the Department of Aerospace Engineering, Delft University of Technology, 2629HS Delft, The Netherlands.
E-mail: w.vlenterie@student.tudelft.nl.

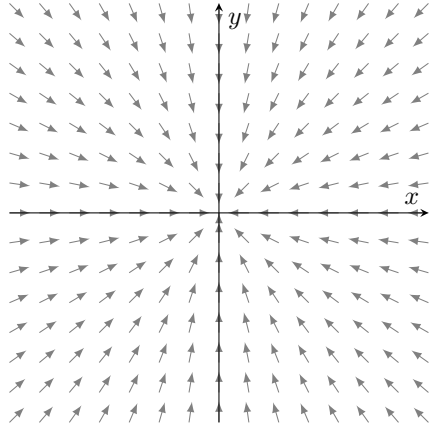


Figure 2: Example of the rendezvous point global field, guiding swarm members at constant velocity towards the predefined centre-point.

restricted consumer grade UAVs. The goal of the approach is to obtain a high-level controllable, scalable, and collision-free swarm at minimal inter-agent separation using on-board sensing and computation only. The swarms will be controlled using Velocity Templates. An example of a velocity template can be seen in Figure 2. The UAVs will be equipped with a Global Navigation Satellite System (GNSS) for outdoor positioning.

The proposed approach has its foundation in artificial potential fields [18] and is based on the scalable shape formation (SSF) approach for pico satellites proposed by Pinciroli et al. [19] to form hexagonal lattices. SSF - as will be shown in this article - generalises poorly to dense swarms of flying robots, leading to excessive oscillations and deadlocks in the case of conflicts. The new approach allows for easy-to-design velocity templates and successfully deals with the aforementioned problems, leading to smooth decentralised control even in dense air space.

First Section II will explain the approach of SSF as proposed in [19]. Then Section III covers the implementation of the algorithm on quadcopter UAVs. The performance of the SSF implementation and the conducted experiments in order to assess it are discussed in Section IV. Subsequently Section V explains the velocity template approach. Experiments with the velocity template implementation are to be discussed and the results compared with that of the SSF experiments in Section VI. Lastly Section VII reflects on the results and discusses further outdoor tests.

II. SCALABLE SHAPE FORMATION FOR SWARMS OF PICO SATELLITES

This section explains the approach of the scalable shape formation (SSF) proposed by Pinciroli et al. for pico satellites [19]. The approach is based on artificial potential

fields [18] and considers three components representative of the following behaviours:

- **Flattening & Gathering** – causes all agents to adhere to a 2-dimensional flat surface on the xy plane and attracts all agents to a common centre, preventing separation and allowing swarm control. As shown in Figure 2.
- **Lattice formation & Collision avoidance** – causes the agents to form a hexagonal lattice, as shown in Figure 3, on the xy plane whilst avoiding collisions.
- **Damping** – causes the agents to converge to a steady state and is intended to prevent undesired oscillations around equilibrium positions and the swarm centre.

The total velocity response is the sum of these behaviours as shown in Equation 1, comprising the flattening and gathering component \vec{g} , the lattice formation and collision avoidance term \vec{l} and the damping component \vec{d} ; onwards referred to as the global, local and dissipative components respectively.

$$\vec{u} = \vec{g} + \vec{l} + \vec{d} \quad (1)$$

The global field is a function of a common centre point, shared between all agents of a particular group, the current location of the agent, and can vary over time as described by Equation 2. The resulting vector field has a direction and magnitude for any combination of agent position P_{xy} and a given centre point C_{xy} . An example of such a global field is shown in Figure 2.

$$\vec{g} = f(C_{xy}, P_{xy}, t) \quad (2)$$

The local component is modelled after inter-molecular forces by means of the Lennard-Jones pair potential [20], having two intuitive parameters: separation range σ and potential-well depth ϵ , as depicted in Figure 4. The local component for any agent is the average of the inter-agent forces for all M agents in range, as defined by Equation 3. The force between an agent i and an in-range neighbour j is defined as the derivative of the Lennard-Jones pair potential, as shown in Equation 4.

It is conjectured that the steady state arrangement is a hexagonal lattice as depicted in Figure 3.

$$\vec{l} = \frac{1}{M} \sum_{j=1}^M \frac{P_j - P_i}{|P_j - P_i|} l_j \quad (3)$$

$$l_j = \frac{dL}{dr} = \frac{12\epsilon}{r} \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] \quad (4)$$

The dissipative component, required to dissipate virtual energy, acts as a viscosity term allowing the swarm to converge to a steady state. The dissipation is described by Equation 5 where ξ is a design parameter, which increases viscosity when raised.

$$\vec{d} = -\xi \cdot \dot{\vec{q}} \quad (5)$$

III. SCALABLE SHAPE FORMATION FOR UAVS

This section discusses the implementation of the SSF approach on UAVs. At first the platform on which it is implemented to conduct experiments is discussed and afterwards the vision pipeline, constructed to obtain on-board decentralised sensing of neighbouring agents, is explained.

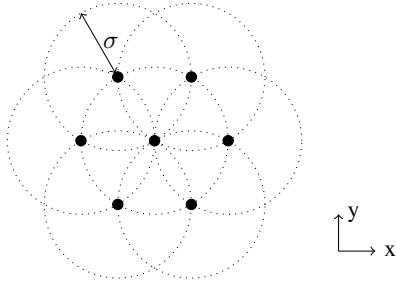


Figure 3: Conjectured steady state arrangement of agents utilising the Lennard-Jones potential - a hexagonal lattice.

A. Platform

The platform used for all experiments is the Parrot Bebop 2 [21], as shown in Figure 5. The bebop 2 is a consumer grade quadcopter featuring a 14MP 180° fisheye camera tilted 30° downwards, dual-core CPU, quad-core GPU, WiFi and GNSS receiver and yet weighs only 500 grams.

Paparazzi UAV [22] runs on-board the UAVs, which is an open-source autopilot with focus on autonomous flight supported on a range of Parrot products [23].

The autopilot inner-loop utilises the cascaded Incremental Nonlinear Dynamic Inversion (INDI) controller designed and implemented in Paparazzi UAV by Smeur et al. [24], which offers superior disturbance rejection when compared to Proportionate Integral Derivative (PID) control. This should allow for highly accurate control even in unfavourable wind conditions, which is essential when flying in close proximity to other UAVs.

Positioning: In order to maximise the potential accuracy of the INDI controller and achieve minimal inter-agent separation as described in Section I the UAVs need accurate position information.

For indoor flights, since GNSS coverage is poor, use was made of an OptiTrack motion capture system, which provides below centimetre level precision.

Outdoors Real-Time Kinematic (RTK) GNSS provides a possibility to decrease the horizontal position accuracy from 2.5 m using the Bebop 2's stock u-Blox M8N [25] to 2.5 cm

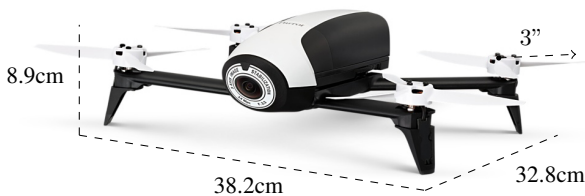


Figure 5: Picture of the Parrot Bebop 2 quad-rotor including dimensions. Source: parrot.com

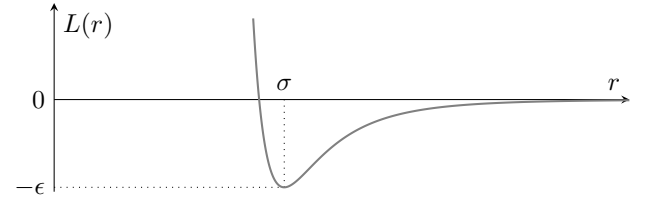


Figure 4: $L(r)$ - The Lennard-Jones potential [20]. With potential-well depth ϵ and separation range σ .

using a RTK base-station and the high precision u-Blox M8P [26]. The stock M8N was replaced with the pin compatible M8P.

Using RTK equipped UAVs enhances the operational weather/wind envelope and drastically reduces the minimum safe inter-agent distance due to the more accurate position control.

Heading: Accurate heading information is essential during autonomous GNSS based flight. For indoor flight the OptiTrack provided accurate heading information. Outdoors however, UAVs tend to suffer from magnetometer drift due to their relatively cheap and inaccurate sensors.

Consistent manual re-calibration using batch methods such as TWOSTEP [27] can remedy this, although this is considered highly undesirable for large swarms of UAVs.

To omit frequent manual re-calibration of all agents an Unscented Kalman Filter (UKF) is used to estimate the full magnetometer calibration on-line during flight. The Kalman filter design is based on the work by Crassidis et al. [28] where the UKF implementation (in C) from the TRICAL library by Ben Dyer is used [29].

B. Vision

In order to implement the SSF approach, a position estimate to the neighbours in range is required. Due to the focus on decentralisation and the complication of adding extra sensors to all UAVs the neighbour detection will be based on computer vision. Other on-board methods for neighbour positioning, such as Radio Frequency based techniques, would also have been possible.

To limit the computational complexity of the vision algorithm, 10 cm diameter coloured styrofoam spheres are placed atop the UAVs. The spheres allow for pose invariant recognition and the algorithm is further simplified by the use of a colour filter for segmentation.

Achieving successful and collision-free flight hinges on an adequate update rate and accuracy of the computer vision algorithm.

Most swarming algorithms in the literature tend to consider agents with omni-directional field of view (FOV) and unlimited

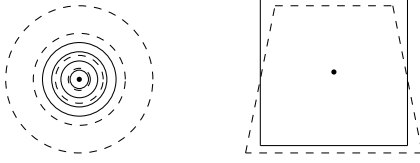


Figure 6: Illustrations of fisheye lens correction (left) and perspective correction (right). With input image (solid) and corrected output image (dashed).

sensing range [30], [31], [32]. Only few applications consider realistic agents with a limited FOV and sensor range [33], [16].

Since the forward positioned camera on the chosen platform only has a 180° diagonal FOV with finite resolution, such realistic limitations should be accounted for and their effects mediated.

Agents are constricted to point their camera towards the direction of the global field. This ensures all agents behind a particular agent are looking in its direction, allowing for backwards motion without collision.

In order to evaluate the relative position of neighbouring agents, their position in the camera frame, given by pixel coordinates and contour area, has to be converted to homogeneous world coordinates. In order to do so both the fisheye lens distortion and 30° downwards tilt have to be corrected for.

Lens correction: Several approaches exist in literature for correcting fisheye distortions on images, the most well-known being the 6^{th} order Brown-Conrady distortion model [34], [35]. A disadvantage to the 6^{th} order polynomial model is the challenge inverting it. This would put computational restriction on the horizon detection and possible GPU implementation.

The invertible nonlinear correction method proposed by Dhane et al. [36] is used for mapping the input radius r to output radius R using invertible Equation 6 where $k = 1.2247445$. The image is mapped as shown on the left hand side of Figure 6 from the solid to dashed circles.

$$R = f * \tan \left(\arcsin \left(\sin \left(\arctan \left(\frac{r}{f} \right) \right) * k \right) \right) \quad (6)$$

Attitude perspective correction: Due to the camera being tilted 30° downwards the image is distorted in the horizontal plane. This distortion is corrected for using model, view and projection (MVP) matrices [37]. Equation 7 illustrates the concept of correcting the camera coordinates to homogeneous coordinates using MVP matrices.

$$\begin{bmatrix} x_h \\ y_h \\ z_h \end{bmatrix} = \begin{bmatrix} projection \\ view \\ model \end{bmatrix} \cdot \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (7)$$

The model matrix positions the centre of the frame in 3D world space, the view matrix maps the 3D world space to a chosen viewport tilted 30° opposite of the physical camera

tilt, and the projection matrix applies a perspective projection causing the part of the frame further away from the viewport to appear smaller than that close-by.

The resulting correction is similar to the right hand side of Figure 6, where the top section of the camera image, which was stretched due to the downward camera tilt, is shrunk back proportionately.

Horizon stabilisation: To reduce computational effort, only a given FOV around the horizon is used for vision processing.

The estimated attitude for each frame is used to map homogeneous output coordinate $(0, 0)$, corresponding to the body x-axis, to a corresponding pixel position on the image (x_h, y_h) .

The horizon position (x_h, y_h) and requested horizon FOV_x and FOV_y, set to 180° and 30° respectively, can then be used to set the image sensor cropping such that only the requested area is read from the sensor.

Contour detection: In order to accurately position neighbours in range at a sufficient update rate an active random colour filter was designed. A sample size N of random pixel positions are tested against the colour filter which if it passes recursively first to the top and then follows the edge of the blob clockwise in search of a closed contour. Figure 7 demonstrates this approach.

The major benefit of this method in comparison to a more straightforward flood-fill is that the clockwise contour approach scales relative to the circumference $\mathcal{O}(r)$, instead of relative to the area $\mathcal{O}(r^2)$, of the blob.

			1	2	3		
36	37	38	40	S_4	4	6	
34	35	39	41	S_3	5	7	9
32	33			S_2		8	10
31	29			S_1		12	11
30	28	26			18	14	13
	27	25	23	19	17	16	15
		24	22	21	20		

Figure 7: Example illustration of contour detection algorithm. Every block represents a pixel. Only grey pixels pass the colour-filter. Search pixels, on the way to the top, are indicated as $S_{1,2,\dots,N}$ and all remaining evaluated pixels are numerically labeled in ascending order. Darker grey squares represent the start and end location.

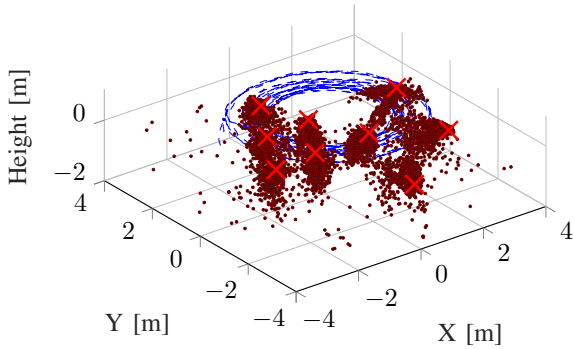


Figure 8: Geolocation results 3D view. Ground truth locations are marked with crosses, circular dots represent on-board estimates and dashed trajectory indicates flight-path.

This results in an essential speedup especially when the blobs are large, i.e. the neighbouring agent is close-range.

Object geolocation: To assess the performance of the complete vision pipeline an experiment is conducted in an indoor controlled area. Nine red styrofoam spheres, identical to those mounted atop of all agents, are positioned across the area and a single UAV is made to fly circles registering its estimates of the area coordinates of the spheres.

The positions of both the UAV and the spheres are measured using a motion capture system so that the estimates can be compared to a ground truth. Figures 8 and 9 show a 3D and top view of the results respectively. The black crosses mark the ground truth locations of the spheres and the grey circular markers indicate onboard position estimates in world coordinates.

The spread in the estimates are correlated with the relative

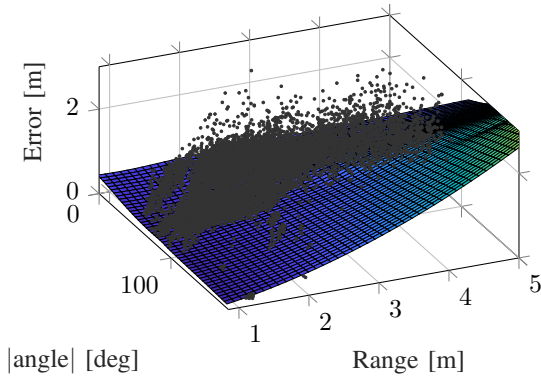


Figure 10: Surface fit of the on-board estimation error versus the absolute relative heading and range of the to be detected object. Surface represents the fit, actual measurements are shown as dots.

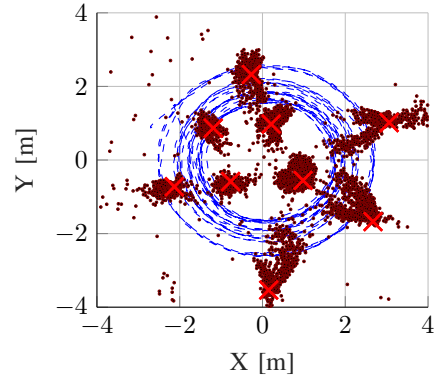


Figure 9: Geolocation results top view. Ground truth locations are marked with crosses, circular dots represent on-board estimates and dashed trajectory indicates flight-path.

angle between the camera principal axis and the spheres due to residual fisheye distortions at the edges of the camera FOV. Figure 10 provides an indication of the correlation between the estimate error and the range and relative angle between the UAV and the spheres. Where in the critical range between 1 m and 2 m the average absolute measurement error from a moving platform was only about 30 cm across the entire FOV domain.

IV. EXPERIMENTS WITH SCALABLE SHAPE FORMATION

In this section SSF experiments are conducted according to the implementation discussed in section III. First the global fields are explained, then the experimental set-up is shown and the results discussed.

A. Global fields

Pinciroli et al. [19] only considered a single global field for a swarm of pico satellites. In our experiments we used different global fields, more representative of the tasks UAVs may encounter.

The global fields used in the experiments can be categorised into three different types, examples of which can be seen in Figure 11:

- Rendezvous point - The simplest field used, consisting of a constant attractor towards the centre of the field. This field can be described by Equation 9 with \vec{q} defined according to Equation 8. Where V is the constant velocity with which agents are attracted.

$$\vec{q} = \begin{bmatrix} C_x - P_x \\ C_y - P_y \end{bmatrix} \quad (8)$$

$$\vec{g} = \frac{V}{|q|} \cdot \vec{q} \quad (9)$$

- Rendezvous bucket - A modification of the rendezvous point field where the magnitude of the global field is scaled down when agents are within a distance R of

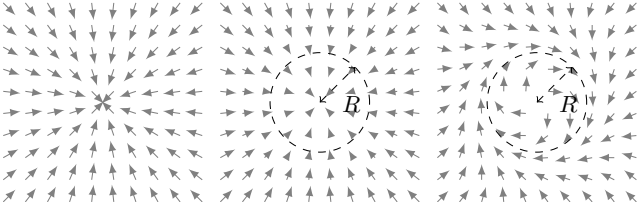


Figure 11: Examples of the three types of global fields used in the experiments. A rendezvous point (left), a rendezvous bucket (middle) and a circular field (right).

the centre of the field. The field can be described as in Equation 9 where V is defined as in Equation 10. Where V_{max} and V_{min} are the maximum and minimum attractive velocities respectively.

$$V = V_{max} \left(1 - \frac{1}{1 + e^{\frac{6}{R} \cdot (|q| - R)}} \right) \in [V_{min}, V_{max}] \quad (10)$$

- Circular field - This field guides all agents in a circular path with radius R around the centre point. This field can be described by Equation 13 where \vec{q} is given by Equation 8, α by Equation 11 and V by Equation 12. The direction of rotation is given by c_d (-1 and +1 for clockwise and counter-clockwise respectively), c_s configures the radial component increase and c_b the increase in magnitude with increasing deviation from the circular path.

$$\alpha = c_d \cdot (90 + c_s \cdot (R - |q|)) \in [0, 180] \quad (11)$$

$$V = c_b (|q| - R)^2 \in [V_{min}, V_{max}] \quad (12)$$

$$\vec{g} = V \cdot \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \cdot \vec{q} \quad (13)$$

B. Experiments with Scalable shape formation on flying robots

Indoor test-flights were conducted in order to assess the performance of the swarming algorithm on small scales of 2 - 4 agents. Larger swarms can, due to the constrained size indoor, only be tested outside.

Four experiments were conducted in order to assess the performance of the SSF approach in four different scenario's:

- 1) Steady state behaviour of a four agent group.
- 2) Sustained motion of a four agent group.
- 3) Residual oscillations between one static and one dynamic agent.
- 4) Conflict resolution between two agents with conflicting global fields.

In order to evaluate the steady state performance of a group of four agents a bucket global field was used. The radius of the bucket was configured as in Equation 14 to be able to contain all $N = 4$ agents at their separation distance $\sigma = 1.5$ m, according to the maximum density for circle packing in a flat plane $\frac{\pi}{\sqrt{12}}$ [38].

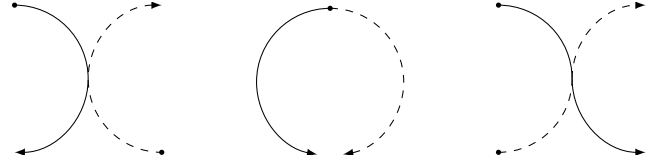


Figure 12: Examples of conflicting objectives for two swarms. The two conflicting objectives are indicated as solid and dashed respectively. Dots represent possible starting locations.

$$R = \sqrt{N \cdot \frac{\sigma^2}{4} \cdot \frac{\sqrt{12}}{\pi}} \quad (14)$$

Sustained motion, due to the constrained area available indoors, was mimicked using a circular global fields and a four agent swarm. The radius was chosen at $R = 1.5$ m such that the agents can still dodge each other without flying out of the geofenced area of 8 by 8m. The four agents at a separation of $\sigma = 1.5$ m all are able to fit on the circular path.

A rendezvous point global field was used to evaluate residual oscillations between a static and dynamic agent under the presence of a non-zero global field. A single stationary agent was positioned in the centre of the field and a second agent was instructed to follow the field while maintaining a separation distance of $\sigma = 1.5$ m.

For the experiment assessing the conflict resolution performance, two agents with conflicting goals are made to interact with each other. Examples of what are considered conflicting goals are illustrated in Figure 12.

Such conflicting goals and multiple interacting swarms were not considered for pico satellites by Pincirolini et al. However, since UAVs operate in an entirely different spatial domain and airspace density such situations are more likely to occur and thus should be considered. For this experiment two counter-rotating circular fields, with $R = 1.5$ m and a single agent per field, were used.

All experiments are conducted for a duration of $T = 10$ minutes, at a separation of $\sigma = 1.5$ m, using a potential well depth $\epsilon = 0.06$ and viscosity coefficient $\xi = 0.03$.

C. Results Scalable Shape Formation experiments

This section discusses the performance and experimental results of the SSF implementation on UAVs. Although initial test-flights were collision-free, limitations can be anticipated when scaling up the size of the swarm, the complexity of the objectives or both.

Experiment 1 - Group steady state behaviour: The results of the first experiment, assessing the steady state behaviour of a four agent swarm can be seen in Figure 13, which shows the flight-paths of all agents. From the flight-paths the

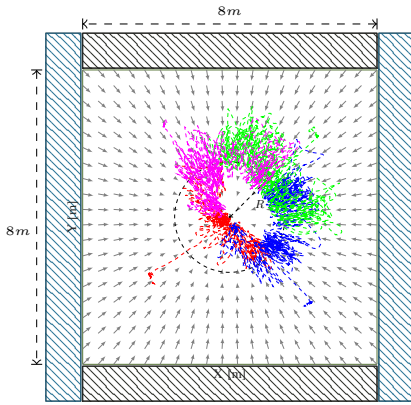


Figure 13: Experimental set-up to test steady state behaviour of a four agent swarm. Flight paths of agents are shown in black, global field shown in light grey.

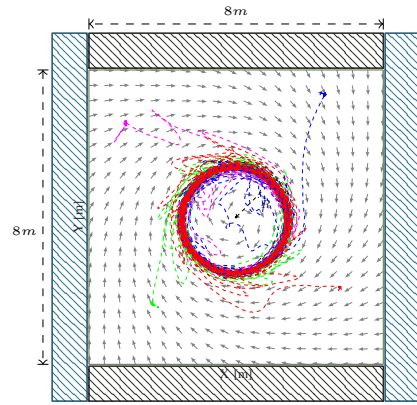


Figure 14: Experimental set-up to test sustained motion of a four agent swarm. Flight paths of agents are shown in black, global field shown in light grey.

oscillations can easily be seen. Additionally the total average x-axis displacement was 143.0m of which 69.7m backwards which is also indicative of the extend of the longitudinal oscillations. The average absolute longitudinal velocity was 0.27 m/s, which ideally should have been close to 0.

These oscillations tend to propagate outwards from the swarm centre. Effectively constraining the maximum swarm size, which is detrimental to the research objective. When scaling to larger swarms, self-induced oscillations tend to become undamped. Pinciroli et al. recognised these residual oscillations and proposed increasing the viscosity coefficient over time, eventually locking every agent in its position.

This is however not longer a viable solution when the global field does not lead to an eventual standstill but sustains agent motion, e.g. in the case of the second experiment testing sustained agent motion.

Experiment 2 - Group sustained motion: Sustained swarm motion is hindered by the oscillations also identified in the first experiment. Agents that get too close to their neighbours, $r < \sigma$, are repelled backwards and severely disrupt the collective motion of the swarm as can be seen from the flight-paths in Figure 14.

In Figure 14 the oscillations can be best seen from the saw-tooth like circumference of the flight-paths. Where an outwards dodging agent, forming a saw-tooth, excites the agent behind it to divert in a similar fashion. This leads to oscillations travelling through the swarm in the opposite direction of movement, similar to highway traffic jams.

The mean and standard deviation of the body x-axis velocity are 0.36 ± 0.26 m/s which would ideally be large and small respectively indicating fast and constant sustained motion. The average total distance travelled backwards along the body x-axis was 13.32m which should ideally be 0.

A steady state where all agents are separated equally over the circumference of the circle is eventually reached, albeit after a period where both the collective swarm motion and oscillations are very chaotic. Additionally this steady

state is easily disturbed, leading to new periods of oscillations.

Experiment 3 - Oscillations: As can be clearly seen in Figure 15, depicting the flight path of the dynamic agent during the third experiment, the oscillations are very poorly damped and no constant separation distance is achieved. The lateral motion of the UAV is due to sensing and actuation noise.

The mean separation was 1.65 m and the standard deviations 0.399 m, which is representative of the intensity of the oscillations.

Residual oscillations will occur long after a steady state has been achieved. Moreover these oscillations significantly reduce the minimum experienced separation distance, potentially leading to collisions.

Experiment 4 - Conflicts: An additional disadvantage is the slow resolution of conflicting objectives when considering multiple global fields spread over multiple swarms of agents.

Figure 16 shows the flight paths of all agents during the experiment testing conflicts. Each conflicts is eventually resolved due to sensing and actuation noise but takes a long time and the motion is chaotic and oscillatory in nature.

The oscillations in the flight-paths in Figure 16 can best be seen from the saw-tooth like exterior circumference, caused by the most outward neighbour dodging further outwards.

The average effective distance travelled in body x-axis was 57.9m forwards, whereas the average backwards distance travelled was 63.7m. Clearly indicative of the high degree of oscillations due to which the effective distance covered is low compared to the distance covered avoiding the conflicting neighbour.

When scaling to larger conflicting swarms the resolution will either take much longer or never occurs.

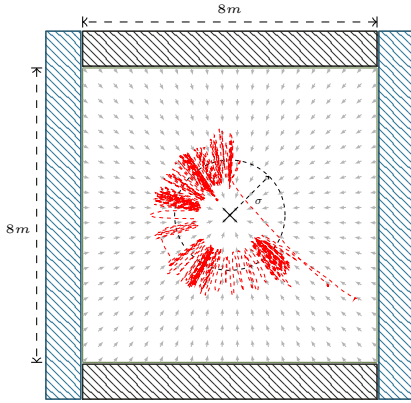


Figure 15: Experimental set-up to test residual oscillations. A static agent is positioned in the centre, flight path of dynamic agent indicated in black. Global field shown in light grey.

V. VELOCITY TEMPLATES FOR DENSE SWARMS

In order to alleviate the problems uncovered in Section IV two extensions to the SSF approach are proposed. These extensions preserve the feasibility and scalability of the swarming algorithm when changing from satellite swarms to dense swarms of flying robots.

These extensions allowed us to consider more complicated fields, interactions between multiple swarms and fields that vary over time.

To emphasise this distinction between the simple linear point global field used by Pinciroli et al. and the more sophisticated fields considered in this paper the global fields will be referred to as "Velocity Templates".

A. Oscillations

Oscillations seem to be caused by the lack of an equilibrium state at the separation range in the presence of a non-zero global field $|\vec{g}| > 0$ since $|\vec{u}| \neq 0$ when $r_i = \sigma$.

The proposed solution involves scaling down the magnitude of the global component $|\vec{g}|$ when it is directed towards one or multiple neighbours such that the magnitude of the global field is unaltered when $r_i \gg \sigma$ and tends to zero when $r_i = \sigma$. This ensures there always exists an equilibrium point at $r_i = \sigma$.

On detailed examination, whenever agents are in a perfect lattice pattern, as depicted in Figure 3, the local component is zero: $|\vec{l}| = dL(\sigma)/dr = 0$, as is evident from Figure 4. However, they are still excited by the global field component \vec{g} , and hence there is no equilibrium at $r_i = \sigma$ when $|\vec{g}| \neq 0$ since the total response is non-zero $|\vec{u}| = |\vec{g}| + |\vec{d}| \neq 0$.

When there is no global field ($|\vec{g}| = 0$) the total combined force \vec{u} changes sign at $r_i = \sigma$ where the slope of the Lennard-Jones potential is minimal since $dL(\sigma)/dr = 0$.

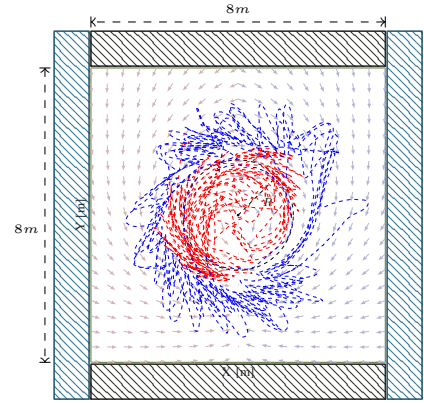


Figure 16: Experimental set-up to test conflict resolution between conflicting global fields. Halves of two circular fields are shown, one clockwise (light blue) the other counter-clockwise (light red). Flight paths of agents guided by the fields are indicated in blue and red respectively.

However when there is a global field ($|\vec{g}| > 0$) this decreases the radius at which the total combined force \vec{u} changes sign to the distance where $|\vec{l}| = |\vec{g}|$. Due to the exponential nature of the Lennard-Jones potential the repulsive force is rapidly increasing for $r_i < \sigma$, resulting in an equilibrium point with a much steeper slope.

This is the case especially when \vec{g} is directed towards a neighbour i located at the separation range ($r_i = \sigma$) and leads to undamped self-excitation of the swarm leading to rapid outwards expansion of the swarm due to the disturbance being propagated outwards.

To mediate this phenomenon let us first introduce the lattice-ratio κ , defined as in Equation 15, where r_{cam} is the camera range and σ the separation range as defined in Equation 4. In order to form stable lattices the lattice-ratio should be smaller than 2, otherwise agents are drawn to neighbours at $r_i = 2\sigma$ at the expense of neighbours at $r_i = \sigma$, and larger than 1 in order to be able to detect neighbours at $r_i = \sigma$.

$$\kappa = \frac{r_{cam}}{\sigma} \in [1, 2] \quad (15)$$

The "relative range" \bar{r}_i , which can be considered a neighbour weighing factor, can then be defined as in Equation 16. The numerator represents the distance agent i has intruded into the camera range, and the denominator the difference between the camera range and the separation range. The relative range stretches from 0 for agents at the camera range, $r_i = r_{cam}$, to 1 for agents at the separation range or closer, $r_i \leq \sigma$.

$$\bar{r}_i = \frac{\kappa \cdot \sigma - \max(r_i, \sigma)}{(\kappa - 1) \cdot \sigma} \quad (16)$$

The total weighed neighbour vector \vec{W}_N can then be expressed as in Equation 17, where $p_{x,y}$ is the agent position and $n_{i,x,y}$ is the estimated position of neighbour i . The contribution of

each neighbour i is summed over all in-range neighbours N . The magnitude of the weighed neighbour vector is restricted to the domain $|\vec{W}_N| \in [0, 1]$. The vector reaches unity for a single neighbour at range $r_i = \sigma$ or with any larger number of neighbours at ranges $r_i > \sigma$.

$$\vec{W}_N = \sum_{i=1}^N \vec{r}_i^2 \left[\begin{array}{c} (p_x - n_{i_x})/r_i \\ (p_y - n_{i_y})/r_i \end{array} \right] \quad (17)$$

The weighed neighbour vector contains both directional information about the mean squared direction of the neighbours $\angle \vec{W}_N \in [-\pi, \pi]$ and scalar information about the relative proximity of those neighbours $|\vec{W}_N| \in [0, 1]$.

Using Equations 15-17 a scalar coefficient for the global field component \vec{g} can now be defined as in Equation 18. The coefficient is constrained to the domain $g_{coef} \in [0, 1]$. The desired agent velocity \vec{u} from Equation 1 can now be rewritten as Equation 19. Figure 17 illustrates the vectors $\vec{W}_N \cdot |g|$ and $\text{proj}_{\vec{g}}(\vec{W}_N \cdot |g|)$ expressed in this section as a result of two nearby neighbours N_1 and N_2 . For the sake of clarification the neighbours are situated such that $|\vec{W}_N| = 1$.

$$g_{coef} = \frac{g - \text{proj}_{\vec{g}}(\vec{W}_N \cdot |g|)}{|g|} \quad (18)$$

$$\vec{u} = g_{coef} \cdot \vec{g} + \vec{l} + \vec{d} \quad (19)$$

As a result of Equation 19 the global field component \vec{g} is scaled from 1 to 0 dependant on the nearby neighbours and their spatial configuration. Therefore all agents in Figure 3, due to their global field scalar coefficient $g_{coef} = 0$ since their separation $r_{1, \dots, N} = \sigma$, are in equilibrium: $|\vec{u}| = |g_{coef} \cdot \vec{g}| = |\vec{l}| = |\vec{d}| = 0$.

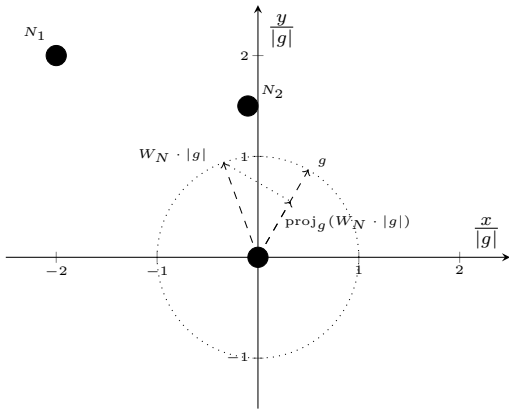


Figure 17: Example illustration of derivation of $\vec{W}_N \cdot |g|$ and $\text{proj}_{\vec{g}}(\vec{W}_N \cdot |g|)$ used in calculating g_{coef} . UAVs are indicated as black dots and vector quantities as dashed arrows. N_1 and N_2 are chosen such that $|\vec{W}_N| = 1$.

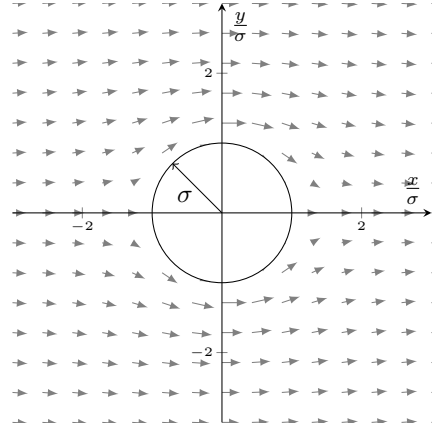


Figure 18: Illustration of uniform flow in x -direction around a circular cylinder with radius σ , represented by a flow doublet.

B. Conflict resolution

The SSF approach does not account for multiple swarms with conflicting objectives, some examples of which can be seen in Figure 12.

In the original setup two counter rotating swarms around the same centre point would lead to almost indefinite oscillations. Although the chaotic motion during the oscillations could cause the blockage to clear eventually, the performance was not sufficient.

Now, with the addition of the oscillation remedying coefficient g_{coef} , introduced in Section V.V-A, the oscillations are severely reduced. This in turn prevents chaotic conflict resolution resulting in both swarms arriving at a standstill at the point of first encounter and forming a combined lattice. Although less chaotic both global objectives are not achieved and the performance is thus still undesirable.

In order to enhance resolution of conflicting objectives the concept of a flow doublet is borrowed from hydrodynamics [39]. The proposed inclusion of flow doublets should guide agents around each other, greatly enhancing the conflict resolution performance.

A flow doublet is a source/sink combination where the distance between the source and sink tend towards 0. This source and sink combination models incompressible uniform potential flow around a circular cylinder.

The equations for the X and Y components, u and v , of a uniform flow \vec{g} around circular cylinder i with radius σ are given in terms of radial coordinates r_i and θ_i in Equations 20 and 21.

$$u_i = |\vec{g}| \cdot \sigma^2 \cdot \frac{\sin(\theta_i)^2 - \cos(\theta_i)^2}{r_i^2} \quad (20)$$

$$v_i = 2 \cdot |\vec{g}| \cdot \sigma^2 \cdot \frac{\sin(\theta_i) \cdot \cos(\theta_i)}{r_i^2} \quad (21)$$

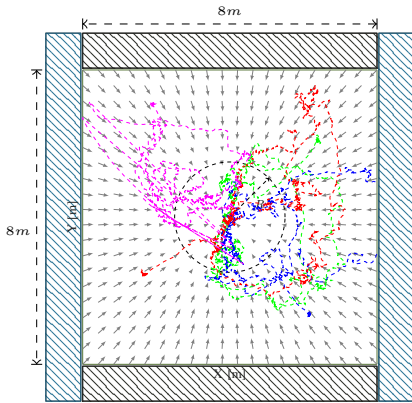


Figure 19: Experimental set-up to test steady state behaviour of a four agent swarm. Flight paths of agents are shown in black, global field shown in light grey.

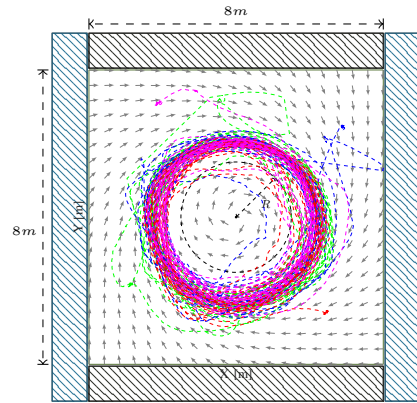


Figure 20: Experimental set-up to test sustained motion of a four agent swarm. Flight paths of agents are shown in black, global field shown in light grey.

The flow-field is illustrated in Figure 18, where the flow travels around the cylinder to continue undisturbed on its initial path afterwards.

In order to incorporate this behaviour into the swarming dynamics, such that agents are inclined to travel around each other with a separation of σ instead of colliding head-on and coming to a standstill, Equation 2 can be rewritten as Equation 22. Where a flow doublet according to Equations 20 and 21 is added to the global field contribution \vec{g} for every neighbour i of the in-range neighbours N .

$$\vec{g} = f(C_{xy}, P_{xy}, t) + \sum_{i=1}^N \begin{bmatrix} u_i \\ v_i \end{bmatrix} \quad (22)$$

Important to note is that the contribution of the flow doublets is added to the global contribution before calculating the oscillation prevention coefficient g_{coef} . Therefore Equations 18 and 19 in Section V.V-A consider the contribution of the global field including flow doublets.

Effectively Equation 19 can be rewritten to the form of Equation 23 to show all contributions to the desired agent velocity \vec{u} .

$$\vec{u} = g_{coef} \left(f(C_{xy}, P_{xy}, t) + \sum_{i=1}^N \begin{bmatrix} u_i \\ v_i \end{bmatrix} \right) + \vec{l} + \vec{d} \quad (23)$$

VI. EXPERIMENTS WITH VELOCITY TEMPLATES FOR DENSE SWARMS OF FLYING ROBOTS

To test the improvements the Velocity Template method makes to the SSF approach, the indoor experiments conducted with the SSF are also conducted for the Velocity Template method.

Additionally outdoor experiments are conducted to assess the scalability of the approach using up to 20 agents.

A. Indoor experiments

This section will discuss the experimental results for the Velocity Templates method and will compare the results with those from the SSF experiments. Section IV explains each experiment in more detail.

Experiment 1 - Group steady state behaviour: Figure 19 shows the flight-path of all four agents during the group steady state experiment using Velocity Templates. Most notable when comparing this figure to Figure 13 is the reduction of the oscillating motion clearly observable in the flight-paths of the SSF experiment.

The difference can also be noticed in total body x-axis displacement which was on average 37.2m of which 17.0m backwards compared to 143.1m and 69.7m backwards for the SSF experiment.

The agents also seem to move around more across the area when compared to the SSF experiment, caused by the lateral motion introduced by the addition of flow doublets. This makes agents rotate around the circumference of the formed lattice, allowing them to find lower energy positions.

The ideal steady state would show limited oscillating movement whilst the deviation of the configuration of agents from a perfect lattice is minimal.

A clear comparison can be made by comparing the body x-axis velocities of the agents, which provide an indication of the degree of longitudinal oscillations. Figure 21 shows the body x-axis velocities of all agents for both the SSF (blue) and Velocity Templates (red) experiments.

Evident is that the degree of longitudinal oscillations is much lower than for the SSF, a absolute average longitudinal velocity of 0.10 m/s compared to 0.27 m/s for SSF.

Experiment 2 - Group sustained motion: Figure 20 shows the flight-paths of all four agents during the group sustained motion experiment for the Velocity Template method. Most

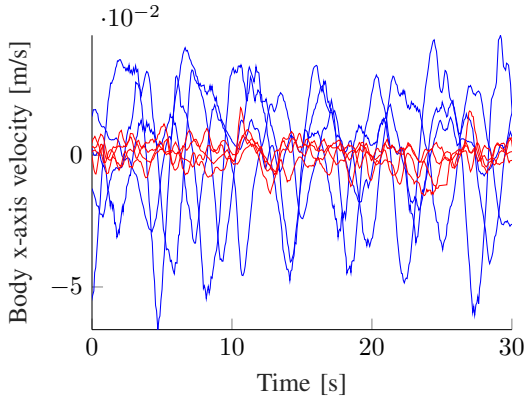


Figure 21: Body x-axis velocity for all agents during group steady state experiment for both Scalable Shape Formation (blue) and Velocity Templates (red). 30 Second time span used for visual clarity.

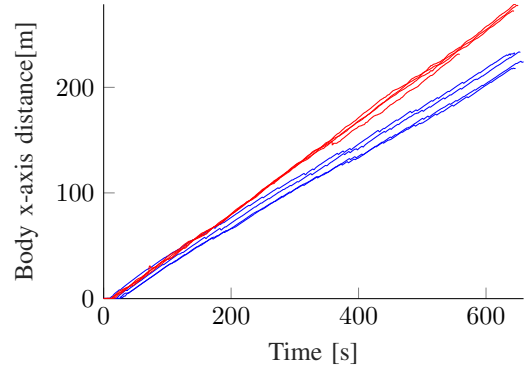


Figure 22: Body x-axis distance travelled for all agents during group sustained motion experiment for both Scalable Shape Formation (blue) and Velocity Templates (red).

notable when compared to Figure 14, the flight-path of the same experiment for the SSF approach, is the increase in effective template circle radius R . Additionally the sawtooth like circumference of the flight-paths in Figure 14 are not present in Figure 20.

The increase in circle radius is due to the outwards lateral velocity introduced by the added flow doublets, which enlarges the effective circle radius instead of causing the traffic jam like longitudinal oscillations observed during the SSF experiment.

In order to objectively compare between the SSF and Velocity Template experiments the mean and standard deviation of the velocity in body x-axis direction as well as the integrated body x-axis distance are measured.

The mean and standard deviation of the body x-axis velocity, expressed as $\mu \pm \sigma$, would ideally be large and small respectively indicating fast and constant sustained motion.

The SSF experiment velocity 0.36 ± 0.26 m/s is lower than that of the Velocity Template experiment 0.43 ± 0.26 m/s with almost identical standard deviation.

The implications of the mean velocity difference is best visualised by Figure 22, showing the integrated body x-axis velocity or longitudinal distance travelled for both the SSF (blue) and Velocity Template (red) experiments. The higher mean velocity results in nearly 40m more being travelled by the Velocity Template swarm on average.

Another measure of continuity is the total distance travelled backwards in the body x-axis, which is on average 13.32m for the SSF experiment and only 2.30m on average for the Velocity Template experiment.

Experiment 3 - Oscillations: The flight-path of the dynamic agent and the location of the static agent for the oscillation experiment using the Velocity Template method can be seen in Figure 23. When comparing the flight-path to that of the

same experiment using the SSF approach, shown in Figure 15, a very noticeable reduction in oscillations can be observed.

Additionally more lateral movement can be seen, this is caused by the addition of the flow doublets. The lateral motion causes the agent to search rotationally for the lowest energy position it can obtain.

A comparison between the achieved separation distance for both experiments, and the ideal separation distance of $\sigma = 1.5$ m, is shown in Figure 25.

Most notable is the relatively large reduction in amplitude of the separation oscillations between the SSF (blue) and the Velocity Template method (red).

Whereas the mean separation for the SSF is similar to that of the Velocity Template experiment, 1.65 m and 1.63 m respectively, for the timespan shown in Figure 25, the standard deviations for the SSF experiment is nearly four times as large, 0.399m versus 0.105 m respectively.

The decrease in oscillations, illustrated by the decrease in separation standard deviation, besides allowing the size of the swarm to be increased also allows for the separation between agents to be decreased.

The separation distance could be decreased due to the fact that the minimum expected separation with a 99% confidence interval, given by $\mu - 2 \cdot \sigma$, which increases from 0.85 m for the SSF approach to 1.42 m for the Velocity Template method.

Experiment 4 - Conflicts: Figure 24 shows the flight-paths for both agents during the conflict resolution experiment. When comparing the flight-paths to those of the SSF experiment, shown in Figure 16, most notably is the transition of the saw-tooth like circumference, seen in the SSF experiment, into a much smoother circumference for the Velocity Template experiment.

To be able to objectively compare both experiments the total travelled distance in body x-axis is measured as well as the

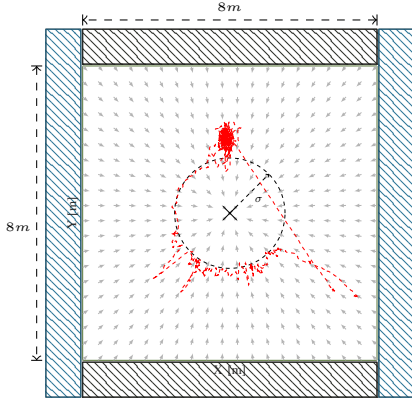


Figure 23: Experimental set-up to test residual oscillations. A static agent is positioned in the centre, flight path of dynamic agent indicated in black. Global field shown in light grey.

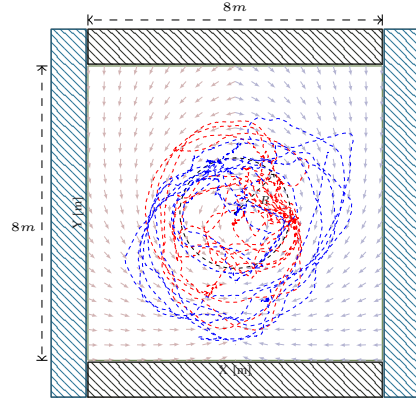


Figure 24: Experimental set-up to test conflict resolution between conflicting global fields. Halves of two circular fields are shown, one clockwise (light blue) the other counter-clockwise (light red). Flight paths of agents guided by the fields are indicated in blue and red respectively.

total distance travelled backwards in body x-axis. In the ideal scenario no distance would be travelled backwards, and the total average distance would be maximal.

The total average distance travelled in body x-axis increases from 57.9m to 87.6m when changing from the SSF to Velocity Template approach respectively. Additionally the total distance travelled backwards sees a major decrease from 63.7m for the SSF to only 7.5m for the Velocity Template experiment.

Figure 26 shows the flight-path for a typical conflict resolution using the SSF (left) and Velocity Template method (right).

Very noticeable is the saw-tooth like flight-path, typical of the longitudinal oscillations from the SSF approach, when

compared to the much smoother flight-path of the Velocity Template resolution, where both agent travel around each other.

B. Outdoor experiments

After successful indoor testing, showing the reduction in oscillations and enhanced conflict resolution performance, the scalability of the approach has yet to be demonstrated. In order to do so outdoor experiments using up to 15 agents are being prepared.

Besides demonstrating the scalability of the swarm, the test will also assess outdoor performance and the robustness of the vision pipeline in a real-world environment.

The final article will include the results of the outdoor experiments.

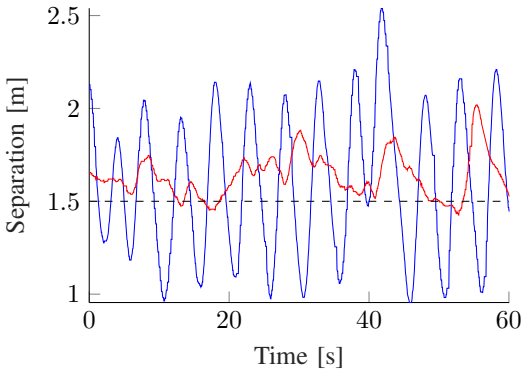


Figure 25: Separation distance between static and dynamic agent during oscillations experiment for both Scalable Shape Formation (blue) and Velocity Templates (red). 60 Second timespan chosen for visual clarity.

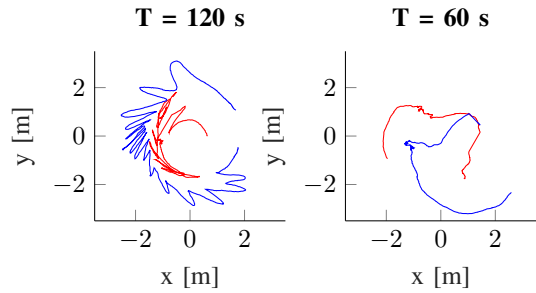


Figure 26: Flight-path of typical conflict resolution for both clockwise agent (blue) and counter-clockwise agent (red) during conflict resolution experiment, shown for both Scalable Shape Formation (left) and Velocity Templates (right). The timespans are chosen such that only a single resolution is visible.

VII. CONCLUSION

A novel approach to decentralised swarming for swarms of flying robots dubbed "Velocity Templates" is proposed. The new approach extends the scalable shape formation (SSF) for pico satellites approach proposed by Pincioli et al. [19]. The extensions are introduced in order to reduce oscillations and enhance conflict resolution performance.

Indoor flight tests were conducted in order to demonstrate the effectiveness of the approach. A comparison is made between results from SSF and Velocity Template experiments, resulting in much improved oscillatory and conflict resolution behaviour and showing an increased adherence to the desired velocity template.

We are currently working towards outdoor experiments with the velocity template approach. We will include the corresponding results - featuring a larger number of drones - in the next version of the article.

ACKNOWLEDGEMENT

The authors would like to thank the Paparazzi UAV community at large for all their contributions. Especially Ewoud Smeur and Freek van Tienen for their work on support of the Parrot products. As well as Erik van der Horst for his help with flight tests and demonstrations.

REFERENCES

- [1] R. S. Miller and W. J. D. Stephen, "Spatial relationships in flock of sandhill cranes (*Grus canadensis*)," *Ecology*, vol. 47, no. 2, pp. 323–327, 1966.
- [2] P. F. Major and L. M. Dill, "The three-dimensional structure of airborne bird flocks," *Behavioral Ecology and Sociobiology*, vol. 4, no. 2, pp. 111–122, 1978.
- [3] B. L. Partridge, "The Structure and Function of Fish Schools," *Scientific American*, vol. 246, no. 6, pp. 114–123, 1982.
- [4] G. Nicolis and I. Prigogine, *Self Organisation in Nonequilibrium Systems*. New York: Wiley, 1977.
- [5] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.
- [6] H. D. Parunak, "Go to the Ant: Engineering Principles from Natural Agent Systems," *Annals of Operations Research*, vol. 75, pp. 69–101, 1997.
- [7] J. Toner and Y. Tu, "Flocks, herds, and schools: A quantitative theory of flocking," *Physical Review E*, vol. 58, no. 4, p. 4, 1998.
- [8] E. Bahceci, O. Soysal, and E. Sahin, "A Review : Pattern Formation and Adaptation in Multi-Robot Systems," Tech. Rep. October, 2003.
- [9] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control Engineering Practice*, vol. 19, no. 9, pp. 1023–1036, sep 2011.
- [10] Guinness World Records, "Intel stuns during CES keynote with record for most drones airborne simultaneously," 2016.
- [11] M. Turpin, N. Michael, and V. Kumar, "Decentralized formation control with variable shapes for aerial robots," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, may 2012, pp. 23–30.
- [12] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.
- [13] S. Hauert, S. Leven, and M. Varga, "Reynolds flocking in reality with fixed-wing robots: communication range vs. maximum turning rate," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE*, 2011, pp. 5015–5020.
- [14] A. Bürkle, F. Segor, and M. Kollmann, "Towards Autonomous Micro UAV Swarms," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 339–353, oct 2011.
- [15] S. A. P. Quintero, G. E. Collins, and J. P. Hespanha, "Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach," *American Control Conference (ACC)*, pp. 2025–2031, 2013.
- [16] G. Vasarhelyi, C. Viragh, G. Somorjai, N. Tarcai, T. Szorenyi, T. Nepusz, and T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2014, pp. 3866–3873.
- [17] D. Izzo, L. F. Simões, and G. C. H. E. de Croon, "An evolutionary robotics approach for the distributed control of satellite formations," *Evolutionary Intelligence*, vol. 7, no. 2, pp. 107–118, jul 2014.
- [18] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *Robotics and Automation, IEEE Transactions*, vol. 8, no. 5, pp. 501–518, 1992.
- [19] C. Pincioli, M. Birattari, E. Tuci, M. Dorigo, M. d. R. Zapatero, T. Vinko, and D. Izzo, "Self-Organizing and Scalable Shape Formation for a Swarm of Pico Satellites," in *NASA/ESA Conference on Adaptive Hardware and Systems*. IEEE, jun 2008, pp. 57–61.
- [20] C. Kittel, *Introduction to solid state physics*. New York: Wiley, 1986.
- [21] Parrot, "Parrot Bebop Drone. Lightweight yet robust quadcopter - Specifications."
- [22] Paparazzi, "Paparazzi UAV."
- [23] B. Remes, D. Hensen, F. V. Tienen, C. D. Wagter, E. V. D. Horst, and G. D. Croon, "Paparazzi : how to make a swarm of Parrot AR Drones fly autonomously based on GPS ." in *IMAV 2013: Proceedings of the International Micro Air Vehicle Conference and Flight Competition, Toulouse, France*, no. September, sep 2013, pp. 17–20.
- [24] E. J. J. Smeur, G. C. H. E. de Croon, and Q. Chu, "Cascaded Incremental Nonlinear Dynamic Inversion Control for MAV Disturbance Rejection," jan 2017.
- [25] U-Blox, "NEO-M8 u-blox M8 concurrent GNSS modules: Data Sheet," Tech. Rep., 2015.
- [26] U-blox, "NEO-M8P u-blox M8 high precision GNSS modules: Data Sheet," Tech. Rep., 2016.
- [27] R. Alonso and M. Shuster, "TWOSTEP: A fast robust algorithm for attitude-independent magnetometer-bias determination," *The Journal of the Astronautical Sciences*, vol. 50, pp. 433–451, 2002.
- [28] J. L. Crassidis, K.-I. Lai, R. R. Harman, F. Dynamics, and A. Branch, "Real-time attitude-independent three-axis magnetometer calibration."
- [29] Ben Dyer, "TRICAL easy magnetometer calibration," 2013.
- [30] V. Gazi, "Swarm aggregations using artificial potentials and sliding mode control," *Robotics, IEEE Transactions*, vol. 21, no. 6, pp. 1208–1214, 2005.
- [31] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, mar 2006.
- [32] D. Izzo, "Autonomous and Distributed Motion Planning for Satellite Swarm," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 2, 2007.
- [33] C. Virág, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, and T. Vicsek, "Flocking algorithm for autonomous flying robots," *Bioinspiration & biomimetics*, vol. 9, no. 2, p. 025012, jun 2014.
- [34] A. Conrady, "Decentred Lens Systems," *Monthly notices of the royal astronomical society*, 1919.
- [35] D. Brown, "Decentering Distortion of Lenses," *Photometric Engineering*, vol. 32, no. 3, pp. 444–462, 1966.

- [36] P. Dhane, K. Kutty, and S. Bangadkar, "A Generic Non-linear Method for Fisheye Correction," *International Journal of Computer Applications*, vol. 51, no. 10, pp. 58–65, 2012.
- [37] R. S. Wright, N. Haemel, G. Sellers, and B. Lipchak, *OpenGL super-bible : comprehensive tutorial and reference*, 5th ed. Addison-Wesley, 2011.
- [38] H.-C. Chang and L.-C. Wang, "A Simple Proof of Thue's Theorem on Circle Packing," *arXiv*, p. 4, sep 2010.
- [39] L. M. Milne-Thomson, *Theoretical Hydrodynamics*. Courier Corporation, 1968.



Dr. Qiping Chu Dr. Qiping Chu obtained his Ph.D degree from TU Delft, The Netherlands in 1987. He was appointed as an associate professor at Chinese Academy of Sciences in 1987 and re-joined TU Delft since October 1991. Dr. Chu is presently an associate professor and the cluster leader of Aerospace Guidance, Navigation and Control at the Section of Control and Simulation, Faculty of Aerospace Engineering, TU Delft. The research field of Dr. Chu is in nonlinear adaptive control with applications. He has (co)authored more than 200 journal, conference

and book chapter publications in the related field.



Wilco Venterie Is currently pursuing his M.Sc. in Aerospace Engineering at the Technical University of Delft on the topic of Control and Simulation of swarming robotics.



Guido de Croon received his M.Sc. and Ph.D. in the field of Artificial Intelligence at Maastricht University, the Netherlands. His research interest lies with computationally efficient algorithms for robot autonomy, with a particular focus on bio-inspired algorithms for vision-based control, learning, and swarm robotics. From 2011-2012 he worked as a Research Fellow in Artificial Intelligence at the European Space Agency. Since 2013, he is an assistant-professor at the Micro Air Vehicle lab of Delft University of Technology, the Netherlands.



Bart Remes Received his M.Sc. in Aerospace Engineering at the Technical University of Delft on the topic of Aerospace for Sustainable Engineering and Technology. He then founded the Micro Air Vehicles Laboratory in 2003 (<http://mavlab.tudelft.nl/>). Currently, he is Cofounder of DAR (<http://darfly.com/>) and project manager at the Micro Air Vehicle lab and was involved in the design and construction of various Micro Air Vehicles, such as the DelFly II and the DelFly Micro(<http://www.delfly.nl/>). Both these Micro Air Vehicles (MAVs) have won multiple

awards including a mentioning as the Smallest Camera Airplane, DelFly Micro in the Guinness Book of records, 2009. Besides flapping wing MAVs, he has initiated and worked on hybrid air vehicles(<http://www.atmosuav.com/>)and the design of the smallest open source autopilot in the world, the Lisa S, a 2x2 cm device,based on the Paparazzi open source software. Other projects where Bart played a significant role are<http://www.dutchuas-tudelft.nl/> and <http://www.roboswift.nl/>.

Literature study

A scalable approach to decentralized MAV swarming

W. Vlenterie
4012976

Delft University of Technology

Contents

1	Introduction	1
2	Purpose of swarms?	3
3	Approaches at swarming	5
4	Swarm taxonomy	7
5	Project requirements	9
6	Swarming control theory	11
6.1	Reynolds flocking model	11
6.2	Stigmergy & artificial pheromones	12
6.3	Swarming using artificial potential fields	13
6.3.1	Formation control	14
6.3.2	Conclusion	14
6.4	Equilibrium shaping	14
6.4.1	Formation control	16
6.4.2	Conclusion	17
6.5	Evolutionary Robotics shaping	18
6.5.1	Formation acquisition task	18
6.5.2	Agent control architecture	18
6.5.3	The artificial neural networks	19
6.5.4	Evolution of the neural networks	19
6.5.5	Conclusion	20
6.6	Alpha-lattice: collective potential flocking	21
6.6.1	Topology: Graphs & Proximity nets	21
6.6.2	Geometry: alpha-Lattices	22
6.6.3	Adjacency elements & collective potential functions	22
6.6.4	Flocking algorithm	23
6.6.5	Conclusion	23
6.7	Scalable shape formation	24
6.7.1	Gathering	24
6.7.2	Lattice formation	25
6.7.3	Energy dissipation	26
6.7.4	Flocking algorithm	26
6.7.5	Conclusion	26
6.8	Bio-inspired self-propelled flocking	27
6.8.1	Realistic model of a flying robot	27
6.8.2	The self-propelled flocking model	28
6.8.3	Collective target tracking	29
6.8.4	Conclusion	30
6.9	Algorithm comparison	30
7	Neighbour sensing techniques	33
7.1	Image recognition	33
7.1.1	Classifier cascade	33
7.1.2	Light / Infrared markers	34
7.1.3	Fiducial markers	34
7.1.4	Conclusion	35

7.2	Radio frequency communication	36
7.2.1	Bluetooth	36
7.2.2	ZigBee	36
7.2.3	WiFi	36
7.2.4	Conclusion	37
7.3	Sensing comparison	38
8	The platform	39
8.1	Flight controller	39
8.1.1	Paparazzi open-source autopilot	39
8.1.2	Inner control loop	40
8.2	Sensors & position determination	40
8.2.1	RTK GNSS	40
9	Project roadmap	41
9.1	Literature study	41
9.2	Platform selection	41
9.3	Algorithm selection	41
9.4	Parameter selection	41
9.5	Platform validation	41
9.6	Algorithm validation	42
9.7	Algorithm porting	42
9.8	Small scale test	42
9.9	Full scale test	42
	Bibliography	43

1

Introduction

Anyone who has ever gazed at a flock of passing starlings must have wondered what and why each and every individual bird contributes to the flock. For decades the complex emergent behaviour and deceptively simple agent-level control of flocks and schools have intrigued biologists and engineers alike[1–3].

Using modern techniques and equipment, much of the mysteries surrounding flocks, herds and schools have been unravelled[4–7]. However, whilst swarms of micro aerial vehicles (MAVs) provide possibilities no single agents can offer, they are not yet part of our technological landscape.

In the past three decades many studies have focused on simulating swarming agents with particular control and studying their emergent behaviour and dynamics [5, 8–10]. However, only few of these studies have considered decentralised swarms and the feasibility of realistically implementing these techniques using currently available technology [11].

Whilst centralised swarming behaviour has been successfully implemented on small scales, distributed swarming remains a challenging area[12–14]. In order to achieve a practically scalable and environment independent swarm it should possess distributed autonomous behaviour.

The objective of this literature research report is to determine a set of feasible technological requirements for a set of micro aerial vehicles (MAVs) and appropriate control dynamics. These should enable the MAVs to swarm autonomously in a decentralised fashion amongst homogeneous agents. The requirements and control dynamics will be based upon the available literature in this area.

The report will start with the benefits and possible applications of MAV swarms in chapter 2. Then chapter 3 focusses on historic attempts at swarming and multi-agent collaboration.

Chapter 6 gives a review of the definitions of swarming and the project's requirements, after which swarming algorithms and control laws of both aerial and space vehicles, proposed in existing scientific literature, are described and compared.

Chapter 7 will examine existing implementations, found in literature, of autonomous MAVs with the ability to detect other, homogeneous, agents. The ability to sense neighbours, and to accurately measure the distance, is of critical importance for the MAVs since the swarming strategies rely heavily on this ability and accurate measurements.

Chapter 8 will showcase the characteristics and on-board technology of the platform and existing literature available for the chosen demonstration platform, the Parrot Bebop drone.

The thesis which will be based on this literature research has as objective to develop an approach to achieve large (ideally ~200 agents, 20 for demonstration purposes) stable real-life decentralised swarm. The report will conclude with a roadmap on how to achieve these goals in chapter 9.

2

Purpose of swarms?

Swarming is useful for approaches where the cost and/or complexity of a single agent possessing all required functions is outweighed by that of multiple co-operating agents able to achieve the same functions or when the operations of a single agent are so operation critical a redundant multi-agent approach is more successful.

According to Parunak, who carried out studies to evaluate the use of swarming for military applications, swarming is appropriate for a problem that has four main characteristics[15]:

- **Diverse** – It can integrate diverse functions, handle information of diverse kinds concerning diverse entities originating from diverse sources.
- **Distributed** – Due to local interactions agent need only communicate with neighbours. The swarm itself can form a communication network. The information considered has a strong geographical component and is most relevant to the closest agents.
- **Decentralised** – Centralised control can causes choke points that impede system operations. Additionally, time delay due to remote control can be unacceptable in rapidly changing scenarios and a reduction in manpower cost of unmanned missions is necessary.
- **Dynamic** – The self-organising nature of swarms allow them to respond to an uncertain and rapidly changing environment even using imperfect knowledge.

Additionally, the application of the swarm has to be practical. This emphasises that the behaviour of the swarm is justifiable. Some forms of self-organisation, such as oscillations and riots might be interesting for research but undesirable in the current application.

Several applications for the deployment of MAV swarms have been discussed in literature, often with the distinction between civil, military and space purposes. Some examples are given of each of these categories[16–18]:

- **Civil applications** – Enhancing agricultural practices, police surveillance, pollution control, environment monitoring, fighting fires, inspecting dams; pipelines or electric lines, video surveillance, motion picture film work, cross border and harbour patrol, light cargo transportation, natural disaster inspection, search and rescue.
- **Military applications** – military operations in urban terrain, building exploration, protect military camps; convoys; industrial premises and/or other safety critical infrastructures, perimeter monitoring, mine detection, search and capture.
- **Space applications** – Coordinated observation, planet exploration, on-orbit self-assembly of solar panels; large antennas or large reflectors in space.

3

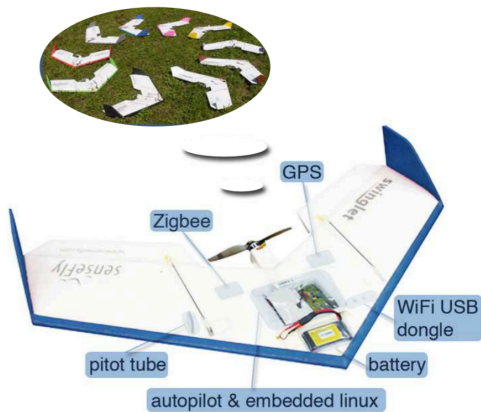
Approaches at swarming

This chapter will list the approaches taken up to date to achieve swarming/cooperative flight. The approaches will be characterised according to several criteria: type of vehicle, number of agents, degree of decentralisation, environment, (technologic) dependencies and the uniqueness of solution.

The chapter will progress chronologically through the attempts made in the past in order to sketch the progress and achievements made in swarming flight. Figure 3.1 shows some examples of swarming approaches.

- **2001 Welsby et al.** – The first project exploring multi-agent systems in three dimensions was conducted using 3 small controllable helium balloons. On-board relative infrared (IR) positioning allowed for decentralised control in an indoor environment[19].
- **2011 Hauert et al.** – The first approach to swarming using a larger group of 10 agents was conducted using fixed wing drones. The drones incorporated Reynolds flocking methodology and used a global navigation satellite system (GNSS) in order to swarm decentralised outdoor[20].
- **2011 Bürkle et al.** – The first extendible swarming platform and framework adapted from commercially available quadcopters. Using 5 quadcopters equipped with a GNSS and a centralised integrating ground control station outdoor swarming was achieved outdoor[18].
- **2011 Hoffmann et al.** – Another testbed developed in order to experiment with quadrotor swarming. The approach includes the complex vehicle dynamics of the quadrotors. Experiments included only 3 agents and a centralised ground station, extended with either GNSS or an overhead camera when flying outdoor and indoor respectively[21].
- **2012 Kushleyev et al.** – An indoor swarming approach using 20 micro-quadrotors, a VICON camera system and a centralised ground station. The positioning precision is very accurate yet heavily reliant on the external sensors[13].
- **2012 Turpin et al.** – Using an indoor VICON system, a centralised ground station and using micro-quadrotors formation flight is achieved. The swarm of only 4 agents is able to change formations, such as a straight line or a square, rapidly[22].
- **2012 Stirling et al.** – The first approach to use only on-board relative sensing on quadrotors. Using two agents fixed to a ferromagnetic ceiling, a third agent is able to position itself relatively to the others. The approach is truly decentralised. However, using only a single flying agent[23].
- **2013 Quintero et al.** – Using a flock of 3 fixed-wing UAVs, a GNSS and a centralised ground control station this approach demonstrates distributed sensing. The resulting sensor network is less sensitive to sensor errors or faults[24].
- **2014 Vasarhelyi et al.** – Using 10 quadrotors and a GNSS this decentralised approach relies only on on-board sensing and satellite signals. The swarm is optimised to cope with large inner and outer errors such as noise, delays and communication range[25].

- **2014 Rubenstein et al.** – Although not a three dimensional swarm, this project is included due to its large number of agents. Using a swarm of 1024 2D agents arbitrary shapes can be constructed by following the swarm contour[26].



(a) Source: Hauert et al.[20]



(b) Source: Hoffmann et al.[21]



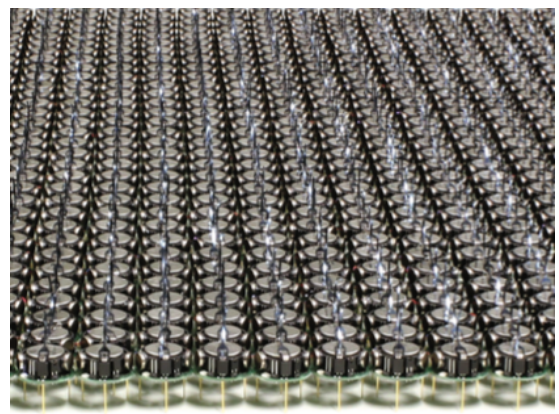
(c) Source: Kushleyev et al.[27]



(d) Source: Turpin et al.[22]



(e) Source: Vásárhelyi et al.[25]



(f) Source: Rubenstein et al.[26]

Figure 3.1: Illustrations and images of approaches to swarming.

4

Swarm taxonomy

The emergence of complex characteristics in large groups of agents could arguably only be a trick of our own perception. This discussion is not particular to swarming MAVs but a philosophical debate on the existence of emergence spanning at least the past 150 years[28]. A good, but general, definition of emergence is given in definition 1 [29].

A second term, often encountered when analysing global behaviours of distributed systems, is self-organisation. The term self-organisation was originally found in the context of chemistry to describe global effects arising from interactions on the microscopic or particle level[4]. Bonabeau et al. later extended the concept to social insects and artificial systems and defined it as in definition 2[30].

Swarming-intelligence, another important definition, was originally used only in the context of cellular robotics[31], but was later extended by Bonabeau et al. to definition 3 [30].

Definition 1 Emergence

Properties of a complex physical system are emergent just in case they are neither (i) properties had by any parts of the system taken in isolation nor (ii) resultant of a mere summation of properties of parts of the system.

Definition 2 Self-organization

A set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components. It relies on four basic ingredients: (i) positive feedback, (ii) negative feedback, (iii) amplification of fluctuation and (iv) reliance on multiple interactions.

Definition 3 Swarming Intelligence

Any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies.

These definitions, though helpful in understanding the governing concepts, do not define any particular swarm characteristic. In order to classify how a swarm can be structured Dudek et al. proposed a more exhaustive taxonomy for swarm robots and their characteristics[32].

The taxonomy proposed in 1993 organises the variety of possible swarm designs along a set of taxonomic axes[32]. The purpose of the taxonomy, as described by Dudek et al., is to “clarify the strengths, constraints and tradeoffs of various designs, and also to highlight various design alternatives”. Table 4.1 lists and describes these taxonomic axes. Each of these axes have key sample points used in the characterisation of swarms. The rest of this chapter focusses on explaining these key sample points. In chapter 5 the taxonomy is used to express the project requirements.

Collective size:

- ALONE – 1 robot. The swarmless swarm.
- PAIR – 2 robots. The simplest group.
- LIM-GROUP – Multiple robots. The number n is small relative to the size of the task/environment.
- INF-GROUP – $n \gg 1$ robots. There is effectively an infinite number of robots.

Table 4.1: Summary of the taxonomic axes. Source: Dudek et al. 1996[33]

Axis	Description
Collective size	The number of autonomous agents in the collective.
Communication range	The maximum distance between two elements of the collective such that communication is still possible.
Communication topology	Of the robots within the communication range, those which can be communicated with.
Communication bandwidth	How much information elements of the collective can transmit to each other.
Collective reconfigurability	The rate at which the organisation of the collective can be modified.
Processing ability	The computational model used by the elements of the collective.
Collective composition	Are the elements of the collective homogeneous or heterogeneous.

Communication range:

- COM-NONE – Robots can not communicate with other robots directly.
- COM-NEAR – Robots can only communicate with other robots which are sufficiently nearby.
- COM-INF – Robots can communicate with any other robots.

Communication topology:

- TOP-BROAD – Broadcast. Every robot can only communicate with all of the other robots.
- TOP-ADD – Address. Every robot can communicate with any other robot by name or address.
- TOP-TREE – Robots are linked in a tree and may only communicate through this hierarchy.
- TOP-GRAPH – Robots are linked in a general graph.

Communication bandwidth:

- BAND-HIGH – Communication is (essentially) free.
- BAND-MOTION – Communication costs similar to the cost of moving the robot.
- BAND-LOW – Communication costs much more than the cost of moving the robot.
- BAND-ZERO – No communication. Robots are unable to sense each other.

Collective reconfigurability:

- ARR-STATIC – The topology is fixed.
- ARR-COM – Coordinated. Re-arrangement with members that communicate.
- ARR-DYN – Dynamic. The relationship of members of the swarm can change arbitrarily.

processing ability:

- PROC-SUM – Non-linear summation unit.
- PROC-FSA – Finite state automaton.
- PROC-PDA – Push-down automaton.
- PROC-TME – Turing machine equivalent.

Collective composition:

- HOMOGENEOUS – Made up of units all with the same characteristics.
- HETROGENEOUS – Robots differentiated by programming or behaviour.

5

Project requirements

Chapter 4 proposed a common taxonomy to describe various aspects of swarms. This section will focus on how to apply this taxonomy to the current project and what requirements can be constructed.

The ultimate purpose of this project is to facilitate a scalable approach to decentralised and autonomous swarming. Where scalable refers to ideally hundreds of drones, as demonstrated by Intel in order to break the Guinness world record for most drones airborne simultaneously, broken early in 2016[34].

In order to truly decentralise the swarm two requirements will have to be imposed on the project.

The first requirement is to minimise agent-to-agent communication. This requirement aids in the scalability of the project, since it removes the need for ever increasing inter-agent communication bandwidth, leads to the swarm obtaining a decentralised character and eliminates the need for a complex central command. The first requirement stimulates a vision based method for neighbour detection since digital communication between agents is penalized.

The second requirement is to minimise the three-dimensional volume occupied by the swarm. In other words: the inter-agent distances should be minimised, ideally to 1 metre, whilst retaining collective stability. This objective is added in order to challenge the control strategy and commercially available technology and assesses the currently feasible minimum inter-agent distance. Minimising the inter-agent distance is very important when considering: indoor flight, swarm visibility, footprint or signal coverage.

In order to omit inter-agent communication, but allow for accurate collective position tracking, use will be made of a real time kinematic (RTK) differential global navigation satellite system (DGNSS). The RTK DGNSS, further explained in section 8.2.1, transmits from a ground station to all in range agents and enhances their onboard GPS signal. The addition of RTK DGNSS technology facilitates both the first and second requirements.

When considering the swarm taxonomy, discussed in chapter 4, the desired swarm of agents should be classified as:

- **INF-GROUP** – Due to the requirement for scalability effectively an infinite number of agents should be able to join.
- **COM-NEAR** – The range of communication, which includes visual cues used for neighbour sensing, is limited yet nonzero.
- **TOP-TREE** – The communication hierarchy tree allows the agents to communicate with a ground-station, to allow for tracking, yet discourages inter-agent communication.
- **BAND-LOW** – Communication between agents is discouraged, thus the bandwidth is low and ideally zero.
- **ARR-DYN** – The swarm has to configure itself autonomously thus there exists no fixed topology.

- **PROC-TME** – The on-board processing of our platform is turing machine equivalent.
- **HOMOGENEOUS** – The composition of the swarm will be consisting of only identical agents.

6

Swarming control theory

The dynamic patterns and apparent complexity of a flock of birds is truly a beautiful sight to behold. Whether concerning flocks, herds, schools or colonies, the overall construct seems agile, fluid and synchronised whilst each individual agent exhibits only simple behaviours [1, 3, 35]. The characteristics of these groups, as discussed in chapter 4, emerge only as a collective behaviour.

This chapter focusses on different techniques researchers have proposed as methods to achieve swarming. Each strategy is explained in detail and concluded with an overview of its strengths and weaknesses. At the end of this chapter the different algorithms are compared and the most feasible set is selected for further study.

6.1. Reynolds flocking model

Swarming behaviour in animals has been studied throughout the second half of the twentieth century [1, 3]. However, research in the field did not gain true attention until it was used by Reynolds for animation purposes [5]. Since manually scripting bird trajectories for animations did not look natural, Reynolds decided to govern his animated boids autonomously¹.

In order to make realistic computer-animations of bird flocks, Reynolds determined three simple behaviours every boid should adhere in order to achieve robust realistic flocking [5]. These behaviours are defined in this section in order of decreasing precedence and lay the groundwork for flocking dynamics.

- **Collision avoidance** – Avoid collisions with neighbouring boids by negatively summing their relative position vectors. This behaviour is illustrated in figure 6.1a.
- **Velocity matching** – Match velocity, heading and speed, to neighbouring boids. Can be interpreted as a predictive version of collision avoidance and is illustrated in figure 6.1b.
- **Flock centering** – Move toward the locally perceived centre of neighbouring boids. This causes boids on the boundary of the flock to steer towards the centre and is illustrated in figure 6.1c.

Each of these behaviours return a desired acceleration consisting of a normalised 3D vector and a unit interval strength which indicates the emphasis of the acceleration requests. The strength factor is taken inversely proportional to the exponential of the distance to the relevant neighbouring boid. It is then up to the navigation module to collect all requests and determine the appropriate response, after which the pilot module instructs the flight model to fly the selected course.

Due to each boid only considering the states of nearby boids, the flock obtains a decentralised characteristic.

In order to combine behaviours, Reynolds proposes either a weighted average or prioritised acceleration allocation. The latter considers that acceleration is a limited resource for the boid and distributes

¹Reynolds refers to simulated bird-like objects generically as 'boids' even when they represent other sorts of creatures

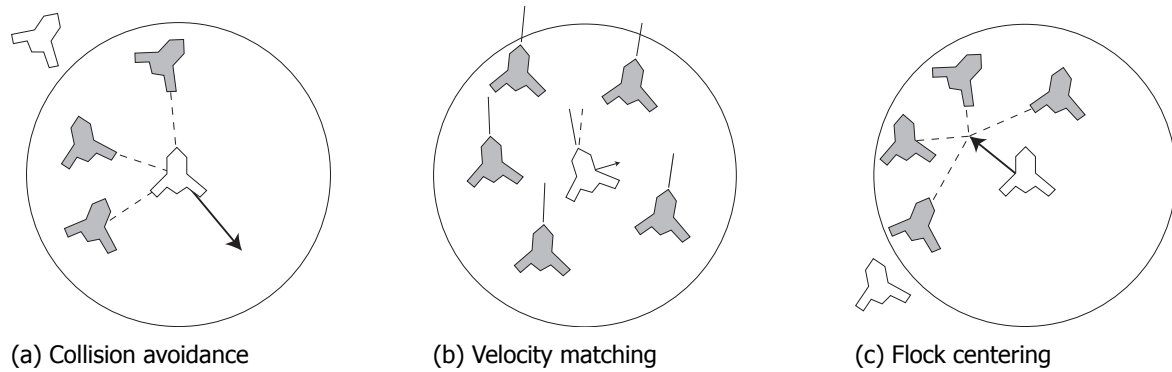


Figure 6.1: Basic behaviours governing Reynolds flocking boids. Source: Park, 2003[10]

available capacity in order of priority. This approach is used to counter the indecision which can be encountered when using a weighted average approach, due to the potential of behaviours averaging each other out.

Reynolds made two extensions to his flocking model in order to gain direct control over the flock. The first extension considers a migratory urge. All boids have a globally preferred direction or position to which it will incrementally turn.

The second extension is environmental collision avoidance, either in the form of a repulsive force field or a steer-to-avoid model. The latter, which detects obstacles in the direction of flight and steers towards a body length beyond the boundary of the object, is a better resemblance of a bird guided by vision.

Although Reynolds included these extensions mainly to control his animations and make them more interactive with the environment, they are potentially desired behaviours for a flock of MAVs.

6.2. Stigmergy & artificial pheromones

The term stigmergy was first introduced by French zoologist Grassé to explain the behaviour of termites[36]. It describes a mechanism in which individual work (*Greek: ergon*) is triggered through the use of signs (*Greek: stigmata*) from the environment. For instance with termites, as a result of an action, pheromones can be released into the environment which triggers a, possibly different, action in neighbouring termites which sense the pheromone.

There are generally two types of stigmergy. Sign-based stigmergy involves the use of non-contributing markers in the environment which influence the behaviour of sensing agents. This is a type of stigmergy used for instance to find the shortest path when termites are gathering food. The other type is sematectonic stigmergy, this involves agents reacting to a physical change in the environment. This can be found for instance in termite nest-building where the agents reacts to the structure of the nest being built.

The stigmergy approach to multi-agent systems can also be applied digitally. Those digital pheromones can be considered a variation of artificial potential functions[37]. Researchers have successfully applied this approach, using digital pheromones, for finding information in networked environments[38].

Pheromones deployed by animals in the real world show three time-variant processes[39]:

- **Aggregation** – Deposits of multiple pheromones from multiple agents aggregate, fusing the information.
- **Evaporation** – Pheromone strength decreases over time, allowing for adaptations of the environment and the resolving of inconsistencies.
- **Diffusion** – Pheromones spread in time allowing access for nearby agents.

In the past two decades researchers have started studying the use of digital pheromones for autonomous coordination of multi agent systems[39, 40]. However, the current pain point is that in order to have a physical implementation, these processes have to be maintained in the same de-centralised and autonomous fashion as the swarm itself, imposing heavy requirements on the communication bandwidth and communication topology of the swarm.

6.3. Swarming using artificial potential fields

A variation on digital pheromones, described in section 6.2, is the use of artificial potential fields. Artificial potential field are often used to conduct research in or simulations of swarms[17, 37, 41, 42]. This method uses fixed or mobile attraction and/or repulsion functions in order to create a potential field along which individual agents can navigate. This section will focus on research done by Gazi and Passino into the application of artificial potential functions for stable swarm aggregation[43–45].

In his work on artificial potential functions, Gazi considers the motion dynamics of each of the swarm members as in equation 6.1[45].

$$\dot{x}_i = \sum_{\substack{j=1 \\ i \neq j}}^N g(x_i - x_j) \quad (6.1)$$

In equation 6.1 x_i represents the position vector of individual agent i and $g(\cdot)$ is the attractive/repulsive function governing the interaction between agents. The function $g(\cdot)$ can be considered as a function consisting of an attractive $g_a(\cdot)$ and a repulsive part $g_r(\cdot)$ as shown in equation 6.2 where $\|y\| = \sqrt{y^T y}$ is the Euclidian norm.

$$g(y) = -y [g_a(\|y\|) - g_r(\|y\|)] \quad (6.2)$$

The functions chosen to represent these attractive and repulsive parts are given in equation 6.3 and 6.4, resulting in the combined attractive/repulsive function in equation 6.5 where a , b and c are positive constants satisfying $b > a$ [43].

$$g_a(\|y\|) = a \quad (6.3)$$

$$g_r(\|y\|) = b e^{-\frac{\|y\|^2}{c}} \quad (6.4)$$

$$g(y) = -y \left[a - b e^{-\frac{\|y\|^2}{c}} \right] \quad (6.5)$$

When combining equation 6.1 and 6.5 the motion dynamics of agent i can be rewritten as shown in equation 6.6.

$$\dot{x}_i = \sum_{\substack{j=1 \\ i \neq j}}^N (x_j - x_i) \left[a - b e^{-\frac{\|x_i - x_j\|^2}{c}} \right] \quad (6.6)$$

The motion dynamics can also be expressed in terms of the potential field, as shown in equation 6.7 where $J(x)$ is the artificial potential function describing the attractive and repulsive relationship between all agents and $x^T = [x_1^T, \dots, x_N^T]$ is a vector containing all individual agents positions.

$$\dot{x}_i = -\nabla_{x_i} J(x) \quad (6.7)$$

Using the the attractive/repulsive function given in equation 6.5 the artificial potential function can be rewritten as in equation 6.8.

$$\begin{aligned}
J(x) &= \sum_{i=1}^{N-1} \sum_{j=i+1}^N [J_a(\|x_i - x_j\|) - J_r(\|x_i - x_j\|)] = \sum_{i=1}^{N-1} \sum_{j=i+1}^N J_{ar}(\|x_i - x_j\|) \\
&= \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left[\frac{a}{2} \|x_i - x_j\|^2 + \frac{bc}{2} e^{-\frac{\|x_i - x_j\|^2}{c}} \right] \tag{6.8}
\end{aligned}$$

Gazi showed that using the artificial potential function of the type mentioned in equation 6.8 a set of desirable results can be obtained summarised below.

- The swarm centre $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$ is stationary for all time.
- For any initial position $x(0)$, as $t \rightarrow \infty$, $\dot{x}(t) \rightarrow 0$ and the swarm converges to a constant configuration.
- For $g(\cdot)$ as in equation 6.5 with linear attraction and bounded repulsion as t progresses all agents will converge to a hyperball: $\|x_i - \bar{x}\| \leq \epsilon$ where ϵ is dependant on the parameters of the attractive/repulsive function in equation 6.5.
- The convergence to the hyperball occurs in finite time limited by:

$$\bar{t} = \max_{i=1, \dots, N} \left\{ -\frac{1}{2a} \ln \left(\frac{\epsilon^2}{\|x_i(0) - \bar{x}(0)\|^2} \right) \right\} \tag{6.9}$$

6.3.1. Formation control

An extension to the artificial potential field algorithm, discussed in section 6.3, is proposed by Gazi in order to solve the multi-agent formation problem[45]. In this extension the control dynamics, described by both equations 6.6 and 6.7, consider pair dependant potentials.

Different pairs of agents can have different sets of attractive/repulsive and potential functions, pairwise denoted as $g_{i,j}(\cdot)$ and $J_{ar_{i,j}}(\cdot)$. A formation constraint is introduced such that a unique minimum of $J_{ar_{i,j}}(\|x_i - x_j\|)$ occurs at $d_{i,j}$ and $\nabla J_{ar_{i,j}}(\|x_i - x_j\|) = g_{i,j}(x_i - x_j) = 0$ when $\|x_i - x_j\| = d_{i,j}$.

6.3.2. Conclusion

The potential field method considered by Gazi is a straightforward approach to swarming. It includes only attractive and repulsive components per agent and is not too computationally demanding or complex. Additional desirable results shown by Gazi, summarised in section 6.3, cause the swarm to converge to a hyperball around a stationary centre within finite time.

However, a disadvantage is that the movement of an individual agent is dependant on the respective distances to all other agents, every agent is assumed to be able to detect all other agents.

An additional shortcoming, also identified by Gazi, is that the dynamics in equation 6.1 are not correspondent to that of realistic agents[45].

Another difficulty is the direct relation between the initial swarm position $x(0)$ and the final swarm position $x(\infty)$ is not easily found, which makes it significantly harder to estimate the behaviour of the swarm.

Although the configuration of the swarm can be controlled using the formation control strategy, discussed in section 6.3.1, the formation constraints have to be designed in such a way a unique minimum occurs. The formation constraints thus fix specific agents to specific positions within the formation, reducing the level of autonomy.

6.4. Equilibrium shaping

In this section another distributed flocking algorithm, proposed by Izzo and Petazzi, is presented which was originally designed for the motion planning of satellite swarms[46, 47]. This algorithm extends

earlier work done by Gazi and Passino, discussed in section 6.3, on preassigned formation flocking using artificial potential fields[43, 45]. This extension by Izzo and Petazzi eliminates the requirement to preassign positions in the formation for each agent, and thus solves the target assignment problem autonomously.

The algorithm uses a dynamical system where the equilibria are permutations of all agents at all the pre-defined target formation points. Using this system the agents are guided based on velocity potentials.

The velocity potential field contains three distinct contributions: gathering, docking and avoidance. A brief explanation and mathematical description of each of the three components, henceforth described as behaviours, are given in this section.

- **Gathering** – This behaviour concerns global attractors for each of the N points in the formation so that a velocity field is generated around each point. The analytical expression for the i -th agent can be written as in equation 6.10 where the function ψ_G is a non-linear dependency from the target distance, and ξ_j and \mathbf{x}_i are the formation point and agent location respectively.

$$\mathbf{v}_i^{Gather} = \sum_j^N c_j \psi_G (\|\xi_j - \mathbf{x}_i\|) (\xi_j - \mathbf{x}_i) \quad (6.10)$$

- **Docking** – The docking behaviour concerns local attractors towards each of the N formation points. The desired velocity component due to docking is only non-negligible in the vicinity of the formation point. Equation 6.11 describes the analytical expression for the docking behaviour, where the terms are comparable to equation 6.10 except ψ_D is a local function where the k_D parameter determines the radius of influence of the formation point.

$$\mathbf{v}_i^{Dock} = \sum_j^N d_j \psi_D (\|\xi_j - \mathbf{x}_i\|, k_D) (\xi_j - \mathbf{x}_i) \quad (6.11)$$

- **Avoidance** – The avoidance behaviour establishes a repulsive velocity potential between different agents in proximity of each other. The velocity component can be analytically described as in equation 6.12, where the terms are comparable to equation 6.10 except ψ_A is a local function and k_A determines its radius of influence comparable to a danger of collision radius. In order to maintain symmetry between agents the parameter b does not depend on a particular agent.

$$\mathbf{v}_i^{Avoid} = \sum_j^N b \psi_A (\|\mathbf{x}_i - \mathbf{x}_j\|, k_A) (\mathbf{x}_i - \mathbf{x}_j) \quad (6.12)$$

According to the analytical definitions of the three guiding behaviours, the desired velocity \mathbf{v}_d for agent i can be described as in equation 6.13.

$$\mathbf{v}_{d_i} = \mathbf{v}_i^{Avoid} + \mathbf{v}_i^{Dock} + \mathbf{v}_i^{Gather} \quad (6.13)$$

The system can now be summarised as in equation 6.14, where $\mathbf{v}_d = [\mathbf{v}_{d_1}, \dots, \mathbf{v}_{d_N}]$, $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ and $\lambda = [c_j, d_j, b]$. The last vector in equation 6.14 contains the weighing parameters which have to be chosen such that all defined formation points are equilibrium points.

$$\dot{\mathbf{x}} = \mathbf{v}_d = \mathbf{f}(\mathbf{x}, \lambda) \quad (6.14)$$

In order to ensure all the defined formation points are equilibrium points, the relation in equation 6.15 has to be satisfied where $\mathbf{x}_e = [\xi_1, \dots, \xi_N]$. Note that equation 6.15 is, by design, not dependant of agent permutations since λ is not dependant on any agent position.

$$\mathbf{f}(\mathbf{x}_e, \lambda) = 0 \quad (6.15)$$

Izzo and Petazzi refer to equation 6.15 as the Equilibrium shaping formula since it determines the values of λ that shape the equilibria of the dynamical system described by equation 6.14[46]. When expanding equation 6.14, using equation 6.12-6.14, a set of N vectorial equations related to each target position ξ_i is obtained as in equation 6.16.

$$\sum_{\substack{j=1 \\ i \neq j}}^N \{ [c_j \psi_G(\|\xi_j - \xi_i\|) + d_j \psi_D(\|\xi_j - \xi_i\|, k_D) - b \psi_A(\|\xi_j - \xi_i\|, k_A)] (\xi_j - \xi_i) \} = 0 \quad (6.16)$$

When considering b as a parameter, equation 6.16 can be rewritten in the form of equation 6.17 where matrix \mathbf{A} and vector \mathbf{g} are dependant on the potential fields ψ_G, ψ_D, ψ_A and the chosen formation points $[\xi_1, \dots, \xi_N]$.

$$\mathbf{A}[c_1, \dots, c_N, d_1, \dots, d_N]^T = \mathbf{g} \quad (6.17)$$

6.4.1. Formation control

Equation 6.17 has, for each formation and set of chosen parameters k_D and k_A , a set linear set of equations in the $2N$ unknowns c_j, d_j . This set of $3N$ equations in $2N$ unknowns has no solution in the general case, so we must consider a linear dependency in order to reduce the number of independent vectorial equations. In order to do this we must create a punctual symmetry group of the target points, denoted G .

Whenever there exists a punctual symmetry in G that maps point ξ_i to point ξ_j the corresponding vectorial equations are linearly dependant when we set $c_i = c_j$ and $d_i = d_j$. Using this symmetry the set of independent vectorial equations reduce to either an identity, a single or a double scalar equation depending on whether more than one symmetry axis, a single symmetry axis or a symmetry plane passes through the considered point respectively[46]. If no symmetry axis or plane passes through the point it reduces to a set of three scalar equations. Figures 6.2a and 6.2b show visualisations of Bravais lattices, which have high degrees of symmetry and to which these simplifications are especially applicable.

In the work of Izzo and Petazzi, the example of a hexagonal Bravais lattice is considered, illustrated in figure 6.2a[46]. Two symmetry groups can be identified: the vertices and the hexagonal centres.

A symmetry plane passes through the vertices, resulting in two independent equations, and a symmetry axis passes through the hexagonal centres, resulting in a single independent equation. The equilibrium shaping equation, given in equation 6.15, thus reduces to a set of only three independent equations when $c_i = c_j$ and $d_i = d_j$ is set for all the point belonging to the same symmetry group.

When the number of equations is smaller than the number of unknown many solutions to the equilibrium equations exist. Every agent can have a "subjective" view of the equilibrium conditions and there is no need to synchronise the chosen solution since all options are equally valid.

When considering the case of regular solids, such as an icosahedron, where a symmetry axis passes through every point and all points belong to the same symmetry group, the equilibrium shaping equation can be reduced to a single independent equations with two unknowns.

Using regular solids allow the equilibrium shaping formula, given in equation 6.16, to be rewritten to equation 6.18 which describes the relation between weights c and d for any chosen parameter b . The parameters ℓ and R in equation 6.18 are the number of edges of the icosahedron and the radius of the circle in which it can be inscribed respectively, h and ℓ' are defined as in equations 6.19 and 6.20.

$$c = \left[5(R-h)e^{-\frac{\ell^2}{k_A}} + 5(R+h)e^{-\frac{\ell'^2}{k_A}} + 2Re^{-\frac{2R^2}{k_A}} \right] b - \left[5(R-h)e^{-\frac{\ell^2}{k_D}} + 5(R+h)e^{-\frac{\ell'^2}{k_D}} + 2Re^{-\frac{2R^2}{k_D}} \right] d \quad (6.18)$$

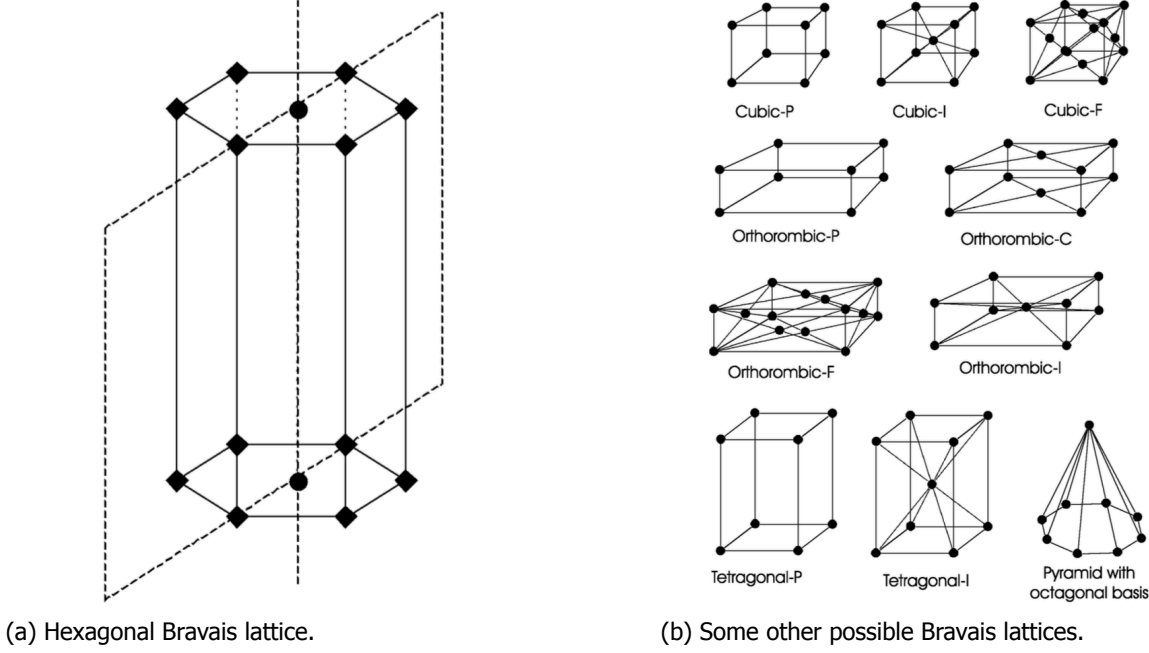


Figure 6.2: Visualizations of Bravais lattices. Source: Izzo and Petazzi[46]

$$h = \sqrt{R - \left(\frac{l}{2 \sin(0.1\pi)}\right)^2} \quad (6.19)$$

$$\ell' = \sqrt{(R + h)^2 + \left(\frac{l}{2 \sin(0.1\pi)}\right)^2} \quad (6.20)$$

Any choice of parameters b , c and d , according to equation 6.18, leads to the dynamical system defined in equation 6.21. In this example the function chosen for ψ is the one proposed by Gazi[45] and Gazi and Passino[43], also discussed in section 6.3 and which can be found in equation 6.5.

In order to decrease the computational load other choices for $\psi(\cdot)$ can also be used, for instance a simple sine function as used by Large et al.[48].

$$\dot{\mathbf{x}}_i = \sum_{j=1}^N \left[-be^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{k_A}} \right] (\mathbf{x}_i - \mathbf{x}_j) + \sum_{j=1}^N \left[-c - de^{-\frac{\|\xi_i - \xi_j\|^2}{k_D}} \right] (\xi_i - \xi_j) \quad (6.21)$$

6.4.2. Conclusion

The equilibrium shaping approach considered by Izzo and Petazzi is, as mentioned earlier in this section, an extension to the artificial potential field control method discussed in section 6.3. Whereas the formation control strategy extension proposed by Gazi required every agent to have a pre-assigned position in the formation, the work done by Izzo and Petazzi eliminates this requirement. Using three global behaviours the agents are autonomously guided to the final formation.

Since they too use a behaviour based approach additional behaviours could relatively easily be added to the control strategy of the agents. Also the ability to alter the used potential fields to match the computational ability of the platform is a clear advantage to this control strategy.

A disadvantage, also described by Izzo and Petazzi, is that there exists a chance that an agent gets stuck in an undesired equilibrium position. Although the final formation is an equilibrium point for the system there is no guarantee that it is globally stable. However, there exist strategies to minimise the

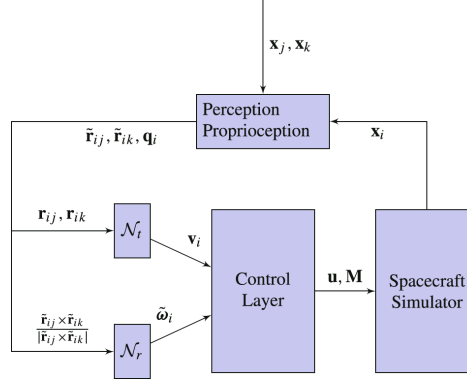


Figure 6.3: Overall control architecture for each agent. Source: Izzo, Simões and de Croon[49]

effect that this local minima problem has on the final configuration.

Since the chosen platform, discussed in chapter 8, does not have a 360° field of view it will pose a challenge to measure the inter-agent distances in every direction. When the algorithm is considered in a two dimensional environment these problem could be reduced and only a semi-sphere field of view can be used.

6.5. Evolutionary Robotics shaping

The control strategy proposed by Izzo and Petazzi, discussed in section 6.4, was able to autonomously solve the target assignment problem encountered in multi-agent shape formation. However, the possible configurations were limited to symmetrical shapes.

In an extension to his earlier work Izzo, in cooperation with Simões and de Croon, developed an approach using evolutionary robotics (ER) techniques[49]. The ER approach allows for more flexibility with respect to task allocation, and enables the formation of asymmetrical triangles.

Although the technique by Izzo, Simões and de Croon was developed for the MIT SPHERES robotic platform aboard the international space station, it can be extended straightforwardly to formation flight of MAVs[50].

6.5.1. Formation acquisition task

Consider a virtual structure, defined by frame \mathcal{F}^* located in point O^* consisting of N target vectors $\vec{\xi}_i = \mathcal{F}^{*T} \xi_i$, defining the desired target geometry of a swarm of N agents. Additionally N sets of possible target quaternions \mathcal{Q}_i define the allowed final orientations for each target position ξ_i .

The formation acquisition task can then be defined for a group of N agents in randomised initial positions as constructing the before mentioned virtual structure.

6.5.2. Agent control architecture

Since the work by Izzo, Simões and de Croon considers satellites, where preventing a “lost in space” situation is critical, they assume the i -th agent is able to detect all other agents in the formation and estimate their respective relative position $\vec{x}_{ij} = \vec{x}_j - \vec{x}_i$.

A second assumption, proven to be valid by Izzo and Petazzi, is that agents do not sense velocity components. Although velocity information is useful in action planning, it is not necessary for the formation acquisition task.

Based on the information perceived by the i -th agent the velocities v_i and ω_i , leading to the desired formation behaviour, are determined using two different artificial neural networks \mathcal{N}_t and \mathcal{N}_r .

Subsequently, the desired velocities are translated by a control layer into control inputs u_i and M_i . A schematic view of the controller architecture is given in figure 6.3.

6.5.3. The artificial neural networks

Two different artificial neural networks are used for each agent. The first network, \mathcal{N}_t , controls the translational kinematics by translating the perceived relative positions of all other agents into desired velocities v_{d_x} , v_{d_y} and v_{d_z} . The second network, \mathcal{N}_r , controls the rotational kinematics by translating a final target orientation into desired angular velocity ω_d in body coordinates.

When considering only three agents the desired translational and rotational velocities can be summarised as $\mathbf{v}_d = \mathcal{N}_t(\mathbf{r}_1, \mathbf{r}_2)$ and $\boldsymbol{\omega}_d = \mathcal{N}_r(\mathbf{n})$.

Using denotations N_i, N_o and N_h for the number of input, output and hidden neurons respectively and \mathbf{I} , \mathbf{h} , \mathbf{O} for the values of the input, hidden and output neurons; equations 6.22 and 6.23 detail the workings of the selected networks. Where w_{kj} and w'_{kj} indicate the network weights, b_k and b'_k the network biases and $\sigma(x) = 1/(1 + e^{-x})$ the sigmoid activation function. The output domain is transformed linearly from the domain $[0, 1]$ to $[O_m, O_M]$ using $O_k = O_m + \hat{O}_k(O_M - O_m)$.

$$h_k = \sigma \left(b_k + \sum_{j=1}^{N_i} I_j w_{kj} \right) \quad k = 1, \dots, N_h \quad (6.22)$$

$$\hat{O}_k = \sigma \left(b'_k + \sum_{j=1}^{N_h} h_j w'_{kj} \right) \quad k = 1, \dots, N_o \quad (6.23)$$

The translational kinematics network \mathcal{N}_t

The neural network \mathcal{N}_t , in charge of desired velocity \mathbf{v}_d , is designed using three requirements:

1. Avoid disorientation effects by relating to a common absolute frame.
2. Be invariant to agent permutations.
3. Allow for micrometre precision in the final formation maintenance.

In order to prevent disorientation, the absolute cartesian components of the relative position vectors \mathbf{r}_1 and \mathbf{r}_2 are used. The sum of \mathbf{r}_1 and \mathbf{r}_2 is input to the network, which is equivalent to inputting the vectors separately and forcing the output's invariance over satellite permutations. In order to achieve micrometer precision an additional, redundant, input is added: $d_1 + d_2 = |\mathbf{r}_1| + |\mathbf{r}_2|$. An output domain of $[O_m = -0.3, O_M = 0.3]$ is considered, relating to the minimum and maximum output velocities.

The resulting artificial neural network has $N_i = 4$, $N_h = 10$, $N_o = 3$ corresponding to $(N_i + N_o)N_h = 70$ network weights and $N_h + N_o = 13$ network biases. A graphic representation of the network \mathcal{N}_t is given in figure 6.4a.

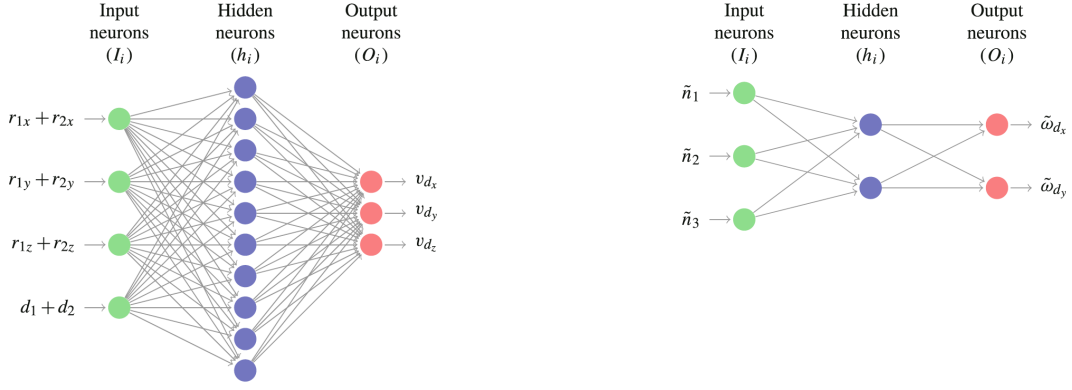
The rotational kinematics network \mathcal{N}_r

The neural network \mathcal{N}_r , in charge of desired rotational velocity $\boldsymbol{\omega}_d$, is asked to keep the agents' z-axis aligned perpendicular to the formation plane. Thus, the input to the network is the target unit vector $\hat{\mathbf{n}} = \mathbf{C} \frac{\mathbf{r}_{ij} \times \mathbf{r}_{ik}}{|\mathbf{r}_{ij} \times \mathbf{r}_{ik}|}$, where $\mathbf{C} = \mathcal{F}_i \mathcal{F}^T$ is the relation between the absolute and the agent body frame. An output domain of $[O_m = -0.3, O_M = 0.3]$ is considered, relating to the minimum and maximum output angular velocities.

The resulting artificial neural network has $N_i = 3$, $N_h = 2$, $N_o = 2$ corresponding to $(N_i + N_o)N_h = 10$ network weights and $N_h + N_o = 4$ network biases. A graphic representation of the network \mathcal{N}_r is given in figure 6.4b.

6.5.4. Evolution of the neural networks

To design the network weights and biases of the networks \mathcal{N}_t and \mathcal{N}_r , defined in section 6.5.3, the ER technique is used. Analogous to biological evolution the artificial neural networks are encoded in chromosomes $\xi = [\mathbf{w}, \mathbf{b}]$ containing their respective weights and biases. Simulations evaluating such neural network provide a measure of quality through a fitness function which drives an evolutionary



(a) Artificial neural network defining the translational kinematics.

(b) Artificial neural network defining the rotational kinematics.

Figure 6.4: The artificial neural networks \mathcal{N}_t (a) and \mathcal{N}_r (b). Source: Izzo, Simões and de Croon[49]

algorithm, interbreeding the best chromosomes in order to optimise the neural network performance. In the case of Izzo, Simões and de Croon a Particle Swarm Optimiser was selected.

Fitness evaluation for \mathcal{N}_t

In order to compare the performance of the different \mathcal{N}_t networks a fitness function is used. First the agent dynamics are simulated integrating equations 6.24 - 6.26, where desired velocities are assumed to be actuated perfectly and randomised initial positions are used. Subsequently, at time $t = T$, the performance of the network is evaluated based on the deviation from the desired formation and residual velocity, as described by equation 6.27. Where l_1, l_2, l_3 are sorted values of the three inter-agent distances and L_1, L_2, L_3 the sorted values of the targeted triangle sides.

$$\dot{\mathbf{x}}_1 = \mathcal{N}_t(\mathbf{x}_2 - \mathbf{x}_1, \mathbf{x}_3 - \mathbf{x}_1) \quad (6.24)$$

$$\dot{\mathbf{x}}_2 = \mathcal{N}_t(\mathbf{x}_3 - \mathbf{x}_2, \mathbf{x}_1 - \mathbf{x}_2) \quad (6.25)$$

$$\dot{\mathbf{x}}_3 = \mathcal{N}_t(\mathbf{x}_1 - \mathbf{x}_3, \mathbf{x}_2 - \mathbf{x}_3) \quad (6.26)$$

$$f_t = (L_1^2 - l_1^2)^2 + (L_2^2 - l_2^2)^2 + (L_3^2 - l_3^2)^2 + |v_1^2| + |v_2^2| + |v_3^2| \quad (6.27)$$

Fitness evaluation for \mathcal{N}_r

Similarly to the evaluation of the translational network, the different \mathcal{N}_r networks are evaluated using a fitness function. The agent dynamics are simulated integrating equation 6.28, where \mathbf{Q} defines the quaternion kinematics similar to the work by Hughes and using randomly generated unit quaternions according to the procedure described by Shoemaker[51, 52], and subsequently evaluated at time $t = T$ using the fitness function given in equation 6.29.

$$\dot{\mathbf{q}} = \mathbf{Q}\tilde{\omega}_d = \mathbf{Q} \begin{bmatrix} \mathcal{N}_r(\mathbf{Cn}) \\ 0 \end{bmatrix} \quad (6.28)$$

$$f_r = \text{acos}(\hat{b}_3 \cdot \hat{n}) \quad (6.29)$$

6.5.5. Conclusion

The ER approach discussed in this section is an extension of the equilibrium shaping method by Izzo and Petazzi discussed in section 6.4. Whereas the equilibrium shaping method can only autonomously solve the target assignment problem for symmetrical shapes, the ER approach proposed by Izzo, Simões and de Croon solves this problem even for asymmetrical shapes.

Using the ER approach they have achieved automated asymmetrical formation flight. Using evolution, artificial neural networks were obtained capable of the position and attitude control of the agent.

Even though micrometre and microradian precision were achieved in low “reality gap” simulations, the simulations were only conducted using three agents. As identified by Izzo, Simões and de Croon themselves, their ER approach has to be redefined in order to extend it to larger groups of agents and further investigation into achievable formations is required.

Additionally, the efficiency of the variable step numerical integrator, which allowed fast evolution using large populations, might deteriorate when considering larger groups of agents.

Computational requirements are also expected to rise drastically when using larger swarms. This could cause the neural networks to run slower than required aboard the MAVs. Also, since the neural networks require extensive training in order to solve the target assignment problem, the flexibility of the swarm could be compromised.

The ER approach, although promising, seems to be restricted mainly by computation power and the uncertainty when considering larger groups of agents. Whilst the approach itself is quite straightforward, the knowledge developed in the neural networks is black-box and therefore extensions to the method might prove difficult.

6.6. Alpha-lattice: collective potential flocking

In this section a theoretical framework by Olfati-Saber for a distributed flocking algorithms is explained [53]. First the underlying graph theory is explained and defined and later a flocking algorithm is presented which forms swarms into α -lattices.

Olfati-Saber proposes three different algorithms which incorporate Reynolds’ three behaviours covered in section 6.1. The proposed algorithms achieve flocking excluding and including a migratory urge and flocking including both a migratory urge and environmental collision avoidance[5].

Since the first algorithm, without a migratory urge, leads to regular separation of the flock, which is not desired, this algorithm will not be considered. Additionally, including a migratory urge enables the control of the flock.

The proposed form of collision avoidance, although excellent in simulation results, will be hard and computationally costly to implement and will thus be omitted. Therefore sections thus focusses on the flocking algorithm including only a migratory urge.

6.6.1. Topology: Graphs & Proximity nets

Consider a graph $G = (\mathcal{V}, \mathcal{E})$ consisting of vertices $\mathcal{V} = \{1, 2, \dots, n\}$ and edges $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, j \neq i\}$. Now $|\mathcal{V}|$ and $|\mathcal{E}|$ are respectively called the *order* and *size* of the graph.

The *adjacency matrix* $A = [a_{ij}]$ of the graph is a matrix with elements satisfying $a_{ij} \neq 0 \Leftrightarrow (i, j) \in \mathcal{E}$.

Now the set of *neighbours* of each node i can be defined as in equation 6.30.

$$N_i = \{j \in \mathcal{V} : a_{ij} \neq 0\} = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\} \quad (6.30)$$

Let $q_i \in \mathbb{R}^m$ denote the position of node i for all $i \in \mathcal{V}$. The *configuration* of all nodes of the graph is then defined by $q = \text{col}(q_1, \dots, q_n) \in Q = \mathbb{R}^{mn}$. The *framework* is a pair (G, q) that consists of a graph and the configuration of its nodes.

Now consider a group of agents with equations of motion as in equation 6.31.

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i \end{cases} \quad \text{where } q_i, p_i, u_i \in \mathbb{R}^m \text{ and } i \in \mathcal{V} \quad (6.31)$$

Let $r > 0$ denote the *interaction range* between two agents. In this model a sphere with radius r determines the set of *spatial neighbours* of agent i denoted by equation 6.32.

$$N_i = \{j \in \mathcal{V} : \|q_j - q_i\| < r\} \quad (6.32)$$

Given an interaction range $r > 0$, a *proximity net* $G(q) = (\mathcal{V}, \mathcal{E}(q))$ can be defined by \mathcal{V} and the set of edges as denoted in equation 6.33. The framework $(G(q), q)$ is called a *proximity structure*.

$$\mathcal{E}(q) = \{(i, j) \in \mathcal{V} \times \mathcal{V} : \|q_j - q_i\| < r, i \neq j\} \quad (6.33)$$

If the interaction range r of all agents is identical the proximity net $G(q)$ becomes an undirected graph. The proximity net is generically a digraph under the assumption that every agent uses a conic neighbourhood to determine its neighbours, as suggested by Reynolds[5]. The graphs considered by Olfati-Saber are all bidirectional[53].

6.6.2. Geometry: alpha-Lattices

The structure used to model the desired configuration of the geometry of the flock is a lattice-type structure. Ideally every agents would be equally distanced from all of its neighbours on a proximity net. This can be described as the solutions of equation 6.34.

$$\|q_j - q_i\| = d \quad \forall j \in N_i(q) \quad (6.34)$$

The solutions q of the conditions in equation 6.34 are the desired formations of agents in a flock. Since this geometric object appears throughout his paper, Olfati-Saber defines it as a lattice-type object as in definition 4[53].

Definition 4 α -Lattice

A configuration q satisfying the set of constraints in equation 6.34. We refer to d and $k = r/d$ as the scale and ratio of the lattice, respectively.

Equation 6.35 describes conformations q' that are very close to an α -lattice and are referred to as *quasi α -lattices*.

$$-\delta \leq \|q_j - q_i\| - d \leq \delta \quad \forall j \in N_i(q) \quad (6.35)$$

6.6.3. Adjacency elements & collective potential functions

Since $\|z\|$ is not differentiable at $z = 0$, we need to define a new norm. The nonnegative map called a σ -norm is defined as in equation 6.36, this new map $\|z\|_\sigma$ is differentiable everywhere. With a fixed parameter $\epsilon > 0$ and gradient $\sigma_\epsilon(z) = \nabla\|z\|_\sigma$ as given in equation 6.37. We can then use this property to construct smooth collective potential functions for groups of nodes.

$$\|z\|_\sigma = \frac{1}{\epsilon} \left[\sqrt{1 + \epsilon\|z\|^2} - 1 \right] \quad (6.36)$$

$$\sigma_\epsilon(z) = \frac{z}{\sqrt{1 + \epsilon\|z\|^2}} = \frac{z}{1 + \epsilon\|z\|_\sigma} \quad (6.37)$$

In order to construct smooth potential functions with finite cut-offs and smooth adjacency matrices a bump function is used. A *bump function* is a scalar function $\rho_h(z)$ that varies smoothly between 0 and 1. Olfati-Saber proposes the bump function given in equation 6.38, though there are other possibilities[53, 54].

$$\rho_h(z) = \begin{cases} 1 & z \in [0, h] \\ \frac{1}{2} \left[1 + \cos\left(\pi \frac{z-h}{1-h}\right) \right] & z \in [h, 1] \\ 0 & \text{otherwise} \end{cases} \quad \text{where } h \in (0, 1) \quad (6.38)$$

The bump function is a C^1 -smooth function with the property that $\rho'_h(z) = 0$ over the interval $[1, \infty)$ and $|\rho'_h(z)|$ is uniformly bounded in z . Equation 6.39 defines the *spatial adjacency matrix* $A(q)$ via its elements using the chosen bump function, where $r_\alpha = \|r\|_\sigma$ and $a_{ii}(q) = 0$ for all i and q .

$$a_{ij}(q) = \rho_h\left(\frac{\|q_j - q_i\|_\sigma}{r_\alpha}\right) \in [0, 1] \quad i \neq j \quad (6.39)$$

Next we integrate an action function $\phi_\alpha(z)$, defined in equation 6.40, that vanishes for all $z \geq r_\alpha$ with $r_\alpha = \|r\|_\sigma$ and $d_\alpha = \|d\|_\sigma$.

$$\phi_\alpha(z) = \rho_h \left(\frac{z}{r_\alpha} \right) \phi(z - d_\alpha) \quad (6.40)$$

$$\phi(z) = {}^{1/2}[(a+b)\sigma_1(z+c) + (a-b)] \quad 0 < a \leq b, c = |a-b|/\sqrt{4ab}$$

$$\sigma_1(z) = \frac{z}{\sqrt{1+z^2}}$$

Then the pairwise attractive/repulsive potential $\psi_\alpha(z)$ can be defined as in equation 6.41. To construct a *smooth collective potential function* as denoted in equation 6.42.

$$\psi_\alpha(z) = \int_{d_\alpha}^z \phi_\alpha(s) ds \quad (6.41)$$

$$V(q) = {}^{1/2} \sum_i \sum_{j \neq i} \psi_\alpha(\|q_j - q_i\|_\sigma) \quad (6.42)$$

6.6.4. Flocking algorithm

Now that all the definitions and functions related to the α -lattices are derived, the flocking algorithm can be constructed. Physical agents are referred to as α -agents whilst virtual agents, which model the collective objective, are referred to as γ -agents. The primary objective of all α -agents is to construct an α -lattice with its neighbouring agents.

The control input of an agent is composed of three terms as depicted in equation 6.43. Where $f_i^g = -\nabla_{q_i} V(q)$ is a gradient-based term, f_i^d is a *velocity consensus term* which acts as a damper and f_i^Y is a navigational feedback term corresponding to the group objective.

$$u_i = f_i^g + f_i^d + f_i^Y \quad (6.43)$$

$$u_i = \underbrace{\sum_{j \in N_i} \phi_\alpha(\|q_j - q_i\|_\sigma) \mathbf{n}_{ij}}_{\text{gradient-based term}} + \underbrace{\sum_{j \in N_i} a_{ij}(q)(p_j - p_i)}_{\text{consensus term}} - \underbrace{c_1(q_i - q_r) - c_2(p_i - p_r)}_{\text{navigational feedback}} \quad c_1, c_2 > 0$$

In equation 6.43 $\mathbf{n}_{ij} = \sigma_\epsilon(q_j - q_i)$ is a vector along the line connecting q_i to q_j and the pair (q_r, p_r) is the state of the γ -agent.

The state of the γ -agent can be seen as a dynamic/static rendezvous point.

Olfati-Saber demonstrates that his flocking algorithm embodies all three rules of Reynolds, explained in section 6.1, in a single equation[5, 53]. He shows the f_i^g term embodies both collision avoidance and flock centering and the f_i^d term embodies velocity matching. The virtual γ -agent embodies the migratory urge only proposed by Reynolds[5].

6.6.5. Conclusion

Flocking using alpha-lattices is, of the algorithms considered, the control strategy which most resembles natural swarming as it can be observed in bird flocks. The three separate components to the flocking algorithm resemble the three basic building behaviours identified by Reynolds.

Although there is no direct control over the formation and shape of the swarm, its objectives can be controlled using the navigational feedback term. Additional terms can relatively easily be introduced to the flocking algorithm in order to model additional behaviour such as for instance environmental collision avoidance. Additionally, the control model can be analysed analytically using graph theory, which is yet another advantage to the strategy.

Olfati-Saber analyses the algorithm in both three and two-dimensional environments where all agents are able to detect each other regardless of their direction. In a three dimensional environment the

chosen platform only has a semi-spheric neighbourhood to detect other agents which results in not all agents being detected and consequently could lead to undesired results.

When using a two-dimensional environment the requirements imposed on the platform can be met. Agents can be able to see in every direction when using a top-mounted camera due to the semi-spheric field of view.

The approach proposed by Olfati-Saber thus provides a feasible two-dimensional model which can easily be extended to the third dimension.

The control strategy proposed is build on a solid theoretical basis but is computationally quite complex. This could cause the algorithm to update too slow when running on the MAVs.

Concluding, this algorithm might be too computationally complex. Although a two-dimensional model might prove feasible, limited control over the shape of the flock and the computational requirements for the platform reduce its practicality.

6.7. Scalable shape formation

The swarming approaches discussed in sections 6.2 - 6.6 have not been specifically been designed nor tested to work with swarms consisting of a large number of agents. An approach that is specifically designed to be scalable up to a large number of autonomous agents is the algorithm designed by Pinciroli et al.[17]. The control strategy proposed autonomously forms a 2-dimensional hexagonal lattice in space around a predefined meeting point. Pinciroli et al. have conducted simulations using the method including up to 500 agents.

The control method proposed by Pinciroli et al. resembles the artificial potential approach, discussed in section 6.3[37]. However, in previous approaches potential fields were composed of both an attractive and repulsive part.

In this approach however, the artificial potential field is a combination of a local and global contribution and a dissipative term. This definition allows for defining the local lattice structure and global external shape separately.

Three behaviours can be identified in this control approach:

- **Flattening & Gathering** – This behaviour causes all agents to adhere to a 2-dimensional flat surface on the xy plane and attracts all agents to the centre of the swarm, preventing separation.
- **Lattice formation & Collision avoidance** – This behaviour causes the agents to form a lattice on the xy plane whilst avoiding collisions.
- **Damping** – This behaviour, comparable to viscosity, causes the agents to converge to a steady state and prevents undesired oscillations around equilibrium positions and the swarm centre.

The control strategy \mathbf{u}_i can now be expressed according to the before mentioned behaviours as in equation 6.44.

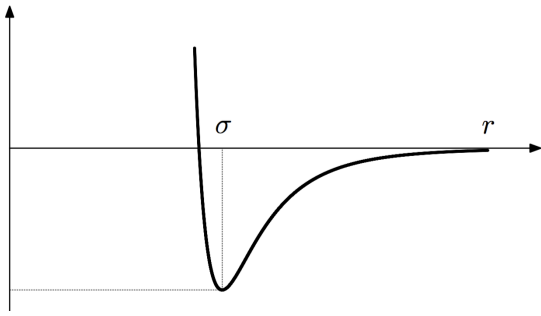
In equation 6.44 \mathbf{g}_i is the force corresponding to the flattening & gathering behaviour, attracting the agents towards the desired centre of the swarm; \mathbf{l}_i is the force corresponding to the lattice formation & collision avoidance, creating local clusters with neighbouring agents and \mathbf{d} is the a damping force, used to stabilise the swarm and ensure convergence.

$$\mathbf{u}_i = \mathbf{g}_i + \mathbf{l}_i + \mathbf{d}_i \quad (6.44)$$

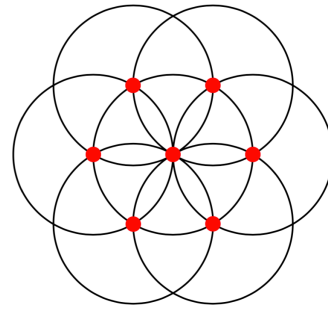
In order to realistically implement the proposed control method, the magnitude of \mathbf{u}_i , in equation 6.44, is limited to a maximum u_{MAX} . Similarly a change the direction of \mathbf{u}_i between two successive control actions is considered to be bounded by $\Delta\theta_{MAX}$.

6.7.1. Gathering

When we define point \mathbf{p} as the desired centre of the swarm, we can define vector $\mathbf{q}_i = [q_{x_i}, q_{y_i}, q_{z_i}]^T$ as the relative position between the agent position and \mathbf{p} and the normalised position vector as $\hat{\mathbf{q}}_i = [\hat{q}_{x_i}, \hat{q}_{y_i}, \hat{q}_{z_i}]^T = \mathbf{q}_i / \|\mathbf{q}_i\|$.



(a) The Lennard-Jones potential L . At σ the potential presents a minimum whose value is ϵ .



(b) The points of minimum energy, when using the Lennard-Jones potential to model the interactions between agents, are conjectured to form a hexagonal lattice.

Figure 6.5: Illustrations of the Lennard-Jones potential and the conjectured hexagonal lattice formed by its point of minimum energy. Source: Pinciroli et al. [17]

Using these notations we can define the flattening & gathering component \mathbf{g}_i as in equation 6.45. Where η_{xy} is a parameter accounting for the attraction of all agents towards \mathbf{p} on the xy plane, and η_z attracts all agents towards the xy plane parallel to the z axis.

$$\mathbf{g}_i = \begin{bmatrix} -\eta_{xy} \|\mathbf{q}_i\|^2 \hat{q}_{x_i} \\ -\eta_{xy} \|\mathbf{q}_i\|^2 \hat{q}_{y_i} \\ -\eta_z q_{z_i} \end{bmatrix} \quad (6.45)$$

The outer shape of the swarm is controlled by the \mathbf{g}_i component. For the function in equation 6.45 the component is a paraboloid in the subspace xy , with circular cross-sections parallel to the xy plane. The outer shape will therefore be circular.

Substituting the x and y components of \mathbf{g}_i with other types of functions other external shapes can be obtained.

6.7.2. Lattice formation

The component \mathbf{l}_i , to autonomously create the desired lattices, draws inspiration from a model of molecular interaction: the Lennard-Jones pair potential, given in equation 6.46 and depicted in figure 6.5a[55].

$$L(r) = \epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - 2 \left(\frac{\sigma}{r} \right)^6 \right] \quad (6.46)$$

When using the Lennard-Jones potential to model the interaction between agents the derivative of this potential, with respect to the distance r , gives the force acting between agents. When $r < \sigma$ the agents experience a repulsive force, thus introducing collision avoidance behaviour, when $r > \sigma$ the force between agents is attractive and for either $r = \sigma$ or $r \rightarrow \infty$ the force is null, as can be derived from figure 6.5a. The Lennard-Jones potential thus causes two agents to achieve and maintain a distance σ from each other.

When considering more than two agents the Lennard-Jones potential is the sum of the pair-potentials of all possible pairs. It is conjectured that the stable arrangement on a flat plant is a hexagon, as can be seen in figure 6.5b.

In order to incorporate the communication range of an agent only its M closest neighbours are concerned.

The magnitude of the artificial force acting between an agent i and its j -th neighbour is obtained by differentiating equation 6.46 with respect to their distance r resulting in equation 6.47.

$$l_{ij} = -\frac{dL}{dr} = \frac{12\epsilon}{\|\mathbf{q}_j - \mathbf{q}_i\|} \left[\left(\frac{\sigma}{\|\mathbf{q}_j - \mathbf{q}_i\|} \right)^{12} - \left(\frac{\sigma}{\|\mathbf{q}_j - \mathbf{q}_i\|} \right)^6 \right] \quad (6.47)$$

In order to obtain the total force vector acting between agents i and its j -th neighbour, equation 6.47 has to be multiplied by their normative position difference as shown in equation 6.48.

$$\mathbf{l}_{ij} = l_{ij} \begin{bmatrix} \hat{q}_{x_j} - \hat{q}_{x_i} \\ \hat{q}_{y_j} - \hat{q}_{y_i} \\ 0 \end{bmatrix} \quad (6.48)$$

The total contribution of all forces acting on agent i due to its M closest neighbours is defined as the average of all contributions, as described by equation 6.49.

Without averaging \mathbf{l}_i would be dependent on M ; which in turn would, since \mathbf{g}_i and \mathbf{l}_i are summed, also make the choice of parameters η_{xy} and η_z dependant on M . Averaging removes this undesired dependency.

$$\mathbf{l}_i = \frac{1}{M} \sum_{j=1}^M \mathbf{l}_{ij} \quad (6.49)$$

6.7.3. Energy dissipation

A third term is added to the control strategy \mathbf{u}_i in order to dissipate the energy contained in the conservative fields that define the forces \mathbf{g}_i and \mathbf{l}_i . Without the dissipative term convergence would be impossible.

The expression for dissipative term \mathbf{d}_i is analogous to the physical property of viscosity, and is shown in equation 6.50. Where ξ is a design parameter usually < 0.2 .

$$\mathbf{d}_i = -\xi \dot{\mathbf{q}}_i \quad (6.50)$$

After the conversion of the swarm has finished, residual oscillations around the equilibrium points are present. In order to dampen the oscillations the virtual viscosity can be increased, by increasing parameter ξ until the residual speed of the agents is not sufficient to overcome the viscosity. The resulting stabilisation will trap the agents in the desired equilibrium positions.

Pincirolì et al. suggest on either triggering the final stabilisation after a certain time threshold T , which would be another design parameter; or to trigger the stabilisation more elegantly with a distributed consensus algorithm, which however would impose higher limitations on the inter-agent or agent-groundstation communication bandwidth[56].

6.7.4. Flocking algorithm

All terms in equation 6.44, the control strategy for \mathbf{u}_i , have now been defined and can be combined as shown in equation 6.51. Where the bounds u_{MAX} and $\Delta\theta_{MAX}$, discussed at the beginning of this section, are not yet included.

$$\mathbf{u}_i = \underbrace{\begin{bmatrix} -\eta_{xy} \|\mathbf{q}_i\|^2 \hat{q}_{x_i} \\ -\eta_{xy} \|\mathbf{q}_i\|^2 \hat{q}_{y_i} \\ -\eta_z q_{z_i} \end{bmatrix}}_{\mathbf{g}_i} + \underbrace{\frac{1}{M} \sum_{j=1}^M \frac{12\epsilon}{\|\mathbf{q}_j - \mathbf{q}_i\|} \left[\left(\frac{\sigma}{\|\mathbf{q}_j - \mathbf{q}_i\|} \right)^{12} - \left(\frac{\sigma}{\|\mathbf{q}_j - \mathbf{q}_i\|} \right)^6 \right]}_{\mathbf{l}_i} \underbrace{\begin{bmatrix} \hat{q}_{x_j} - \hat{q}_{x_i} \\ \hat{q}_{y_j} - \hat{q}_{y_i} \\ 0 \end{bmatrix}}_{\mathbf{d}_i} - \xi \dot{\mathbf{q}}_i \quad (6.51)$$

6.7.5. Conclusion

Since scalability is an important requirement of this project a definite advantage of the control strategy proposed by Pincirolì et al. is that they specifically designed their algorithms to be scalable to large numbers of contributing agents. In their work successful simulation results using up to 500 agents are shown.

Similar to the approaches proposed by Gazi & Passino and Izzo & Petazzi this algorithm is based on artificial potential fields. However, in contrast to the other potential field strategies Pincirolì et al. include a local, global and dissipative term.

These terms allow the outer shape of the formation to be controlled using different functions for $g(\cdot)$, shown in equation 6.45.

Also, since the model includes the visibility range of an agent by only including its M closest neighbours, the computational load of the agent and the types of lattices formed and can be considered as design parameters.

The control strategy considered uses a two-dimensional environment which significantly simplifies the requirements for field of view and reduces the overall complexity of the formations. Whilst a three-dimensional formation might eventually be desirable, it will prove easier to start with a two-dimensional approach.

A disadvantage, also encountered by Pinciroli et al., are the residual oscillations around the equilibrium points. In order to reduce these oscillations a stabilisation procedure is proposed by increasing the virtual viscosity.

However, this introduces an additional problem: when to trigger the stabilisation. The easiest approach is to trigger after a pre-defined time, although this would reduce the autonomy of the swarm.

Pinciroli et al. conclude that local minima problem only seldom occur and the overall lattice formation is very accurate in simulations.

6.8. Bio-inspired self-propelled flocking

This section describes a control algorithm developed by Virágh et al. and Vásárhelyi et al. inspired by bird flocks and realistic robotic features[25, 57]. The algorithm is designed to be implementable on autonomously flying MAVs, and has been tested successfully using up to 10 agents.

6.8.1. Realistic model of a flying robot

First a model is presented based on general robotic features present in many realistic robotic systems. In the type of system considered the motion of the robots is controlled by a low-level control algorithm such as a PID controller. The low-level control algorithm typically has the desired velocity vector as input. The time-dependance of the desired velocity of the i -th agent is a function of the positions \mathbf{x}_i and velocities \mathbf{v}_i of the other agents, as described by equation 6.52. Where N is the number of agents and $f_i(\cdot)$ contains the controlling dynamics.

$$\mathbf{v}_i^d = f_i \left(\{\mathbf{x}_j(t)\}_{j=1}^N, \{\mathbf{v}_j(t)\}_{j=1}^N \right) \quad (6.52)$$

In the ideal case the desired velocity vector $\mathbf{v}_i^d(t)$ is realised instantly. However, a robotic system possesses some typical deficiencies which should included in the model.

- **Inertia** – An agent cannot instantly change its attitude or velocity. An assumption is made that in an optimal setup the system reaches the desired velocity with exponential convergence with characteristic time τ_{CTRL} . Additionally, the magnitude of acceleration is limited to a_{max} .
- **Inner noise** – The inaccuracy of the onboard sensors that provide position and velocity information has to be taken into account. A stochastic function $\eta_i^s(t)$, characterised by standard deviation σ_s , is used to model the sensor uncertainty.
- **Sensor refresh rate** – The reaction time and agility of an agent is fundamentally defined by the refresh rate of the sensory inputs. A limited constant sensor refresh rate with frequency t_s^{-1} is included in the model.
- **Communication locality** – Agents have a finite communication range r_c . Thus function $f_i(\cdot)$ is dependant on x_j only if $|x_j - x_i| < r_c$.
- **Time delay** – Due to data processing and transmission delays a time delay is introduced. A constant time delay of t_{del} is assumed.
- **General noise** – In order to model an unpredictable environment, for instance due to wind, an outer noise term $\eta_i(t)$ with standard deviation σ is included in the acceleration of the agent.

The considered model is thus defined by the set: $\left\{ \tau_{CTRL}, a_{max}, r_c, t_{del}, t_s, \left\{ \eta_j(t), \eta_j^s(t) \right\}_{j=1}^N \right\}$ and the model for the acceleration $\mathbf{a}_i(t)$ of agent i is given by equation 6.53 where $\mathbf{v}_i^d(t)$ is defined by equation 6.54. Where $\mathbf{x}_i^s(t)$ and $\mathbf{v}_i^s(t)$ are measures of the integrated noise for random variable $\eta_i^s(t)$ and $\{...\}_{j \neq i}$ denotes a set with iterator $j \neq i$. The function $f_i(\cdot)$ is dependant on the actual velocity and position of agent i and the delayed velocity and position of the other agents within radius r_c and updates with t_s^{-1} frequency.

$$\mathbf{a}_i(t) = \eta_i(t) + \frac{\mathbf{v}_i^d(t) - \mathbf{v}_i(t) - \mathbf{v}_i^s(t)}{|\mathbf{v}_i^d(t) - \mathbf{v}_i(t) - \mathbf{v}_i^s(t)|} \cdot \min \left\{ \frac{\mathbf{v}_i^d(t) - \mathbf{v}_i(t) - \mathbf{v}_i^s(t)}{\tau_{CTRL}}, a_{max} \right\} \quad (6.53)$$

$$\mathbf{v}_i^d(t) = f_i \left(\left\{ \mathbf{x}_j(t - t_{del}) + \mathbf{x}_j^s(t - t_{del}) \right\}_{j \neq i}, \mathbf{x}_i(t) + \mathbf{x}_i^s(t), \left\{ \mathbf{v}_j(t - t_{del}) + \mathbf{v}_j^s(t - t_{del}) \right\}_{j \neq i}, \mathbf{v}_i(t) + \mathbf{v}_i^s(t) \right) \quad (6.54)$$

Equation 6.54 can be simplified under the assumptions that $f_i(\cdot)$ is only dependant on relative coordinates of the agents and it can be expressed as a sum of local pairwise interactions $f_{ij}(\cdot)$. Additionally, the communication range is defined as function $\theta(x)$ which equals 0 if $x < 0$ and 1 otherwise and \tilde{x} and \tilde{v} are defined as the measured values including the modelled inner noise term so: $\tilde{x}_i = x_i + x_i^s$ and $\tilde{v}_i = v_i + v_i^s$. Under said assumptions equation 6.54 can be rewritten to equation 6.55.

$$\mathbf{v}_i^d(t) = \sum_{j=1}^N f_{ij}(\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i, \tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j) \theta(r_c - |\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j|) \quad (6.55)$$

6.8.2. The self-propelled flocking model

Using the agent model discussed in section 6.8.1 a minimal algorithm is proposed capable of driving collective robotic systems towards a stable, collision-less, self-organised correlated flocking state. The algorithm is based on animal swarms similar to Reynolds work discussed in section 6.1.

The desired velocity \mathbf{v}_i^d is decomposed into a sum of interaction terms and the self-propelling preferred velocity v_{flock} .

Self-propelling velocity

The agents are defined as self-propelling particles with preferred velocity v_{flock} , this behaviour is defined by equation 6.56.

$$\mathbf{v}_i^{SPP} = v_{flock} \frac{\mathbf{v}_i}{|\mathbf{v}_i|} \quad (6.56)$$

Short-range repulsion

In order to avoid collisions with neighbouring agents a local linear repulsion between agents is defined by equation 6.57. Where $\mathbf{d}_{ij} = \mathbf{x}_j - \mathbf{x}_i$, D is the repulsive strength and r_0 is the interaction range.

Virágh et al. consider a linear instead of higher-order repulsion due to the inner-noise affecting the measured range. Since fluctuations are considered to be possible in the same range as r_0 linear repulsion is superior because it does not cause sudden changes or singularities in the output. However, when the position accuracy increases higher-order functions like the Lennard-Jones potential, described in equation 6.46 in section 6.7, could be used.

$$\mathbf{v}_{ij}^{rep} = \frac{D(|\mathbf{d}_{ij}| - r_0)}{|\mathbf{d}_{ij}|} \mathbf{d}_{ij} \theta(r_0 - |\mathbf{d}_{ij}|) \quad (6.57)$$

Velocity alignment of neighbours

The velocity alignment term should satisfy three assumptions. First it should relax the velocity difference between neighbouring agents. Secondly it should only act locally. And lastly it should have an upper threshold, even when inter-agent distances approach zero. Virágh et al. chose a viscous friction-like interaction term as defined in equation 6.58. Where C_{frict} is the strength of the alignment and r_{min} defines a threshold to avoid division by close-to-zero distances.

$$\mathbf{v}_{ij}^{frict} = C_{frict} \frac{\mathbf{v}_j - \mathbf{v}_i}{\max\{r_{min}, |\mathbf{d}_{ij}|\}^2} \quad (6.58)$$

Flocking model

The terms contributing to the desired agent velocity \mathbf{v}_i^d can now be added in order to obtain the self-propelled flocking model given by equation 6.59.

$$\mathbf{v}_i^d = \mathbf{v}_i^{SPP} + \sum_{j \neq i}^N (\mathbf{v}_{ij}^{rep} + \mathbf{v}_{ij}^{frict}) \theta(r_c - |\mathbf{d}_{ij}|) \quad (6.59)$$

Virág et al. include an additional term bounding the flight area using repulsive virtual walls. Such a global positional constraint is an important contribution to the integrity of the flock. However, in this section we will consider the collective target tracking extension proposed by Virág et al. discussed in the next section.

6.8.3. Collective target tracking

An extension to the flocking model described by equation 6.59 is a collective target tracking algorithm using an a priori defined target point. Using both algorithms the flock is able to smoothly transition between flocking, far away from the target, and collective hovering, close to the target. During the transitional phase between flocking and hovering the preferred velocity has to smoothly approach zero and the coherence and robustness of the flock should be maintained without any jamming or oscillations.

Suppose the centre of mass of the flock is a virtual agent with preferred velocity v_0 , then every agent is tasked to approach this virtual agent, in order to join the flock, and move parallel to it, in order to reach the target.

However, since interactions between the agents are assumed local, determining the centre of mass of the flock is impossible for an agent. Therefore the agent calculates a local centre of mass (CoM), based on the information available within its communication range.

Attraction toward the target point can then be defined as in equation 6.60. Where v_0 is the magnitude of the preferred velocity, \mathbf{x}^{trg} is the target position, \mathbf{x}_i^{CoM} the perceived centre of mass as seen by the i -th agent, r_{trg} the target area radius, r_{CoM} the perceived flock radius and $s(\cdot)$ a sigmoid function as defined in equation 6.61.

$$\mathbf{v}_i^{trg} = v_0 \left[s(|\mathbf{x}_i^{CoM} - \mathbf{x}_i|, r_{CoM}, d) \frac{\mathbf{x}_i^{CoM} - \mathbf{x}_i}{|\mathbf{x}_i^{CoM} - \mathbf{x}_i|} + s(|\mathbf{x}^{trg} - \mathbf{x}_i^{CoM}|, r_{trg}, d) \frac{\mathbf{x}^{trg} - \mathbf{x}_i^{CoM}}{|\mathbf{x}^{trg} - \mathbf{x}_i^{CoM}|} \right] \quad (6.60)$$

$$s(x, R, d) = \begin{cases} 0 & x \in [0, R] \\ \sin\left(\frac{\pi}{d}(x - R) - \frac{\pi}{2}\right) & x \in [R, R + d] \\ 1 & x > R + d \end{cases} \quad (6.61)$$

The magnitude of the target tracking term is saturated at v_0 as defined in equation 6.62. The flocking algorithm including collective target tracking is then given by equation 6.63.

$$\hat{\mathbf{v}}_i^{trg} = \frac{\mathbf{v}_i^{trg}}{|\mathbf{v}_i^{trg}|} \min\{v_0, |\mathbf{v}_i^{trg}|\} \quad (6.62)$$

$$\mathbf{v}_i^d = \hat{\mathbf{v}}_i^{trg} + \sum_{j \neq i}^N (\mathbf{v}_{ij}^{rep} + \mathbf{v}_{ij}^{frict}) \theta(r_c - |\mathbf{d}_{ij}|) \quad (6.63)$$

6.8.4. Conclusion

In their papers Virágh, Vásárhelyi et al. present a realistic flocking and simulation framework for decentralised control of autonomous MAVs. In addition to simulating flocks of agents real-life experiments were conducted to check the validity using 9 MAVs. The experiment was conducted outdoor in order to test the stability under realistic environmental conditions.

A definite advantage to the approach discussed in this section is the fact that it takes into account several realistic, yet not robot specific, features such as delays and noise, refresh rates and inertial effects. The algorithm is inspired by natural bird flocks and successfully mimics the robustness to disturbances found in said natural swarms.

The considered approach is computationally not too complex, since it computes only based on agents in range and does not rely on numerically complex functions. The simulation framework can cope with realistic disturbances and is thus able to realistically simulate outdoor conditions.

Although the considered quadcopters used the XBee protocol in order to relay their position information, the approach could be implementing using vision feedback.

The control method could be applied in both two and three dimensional environments, allowing for a step-based increase in complexity of the swarm.

6.9. Algorithm comparison

In this section the algorithms considered in the current chapter will be compared and the most promising algorithms will be selected for further study, as described in chapter 9.

From the control strategies considered in this chapter the scalable shape formation proposed by Pinciroli et al., discussed in section 6.7, could be implemented most straightforwardly using a large group of agents. The theory assumes a two-dimensional space in which agents have a limited field of view and the shape of both the lattice structure and the external shape can be controlled.

This in turn allows for a limited form of formation control. The shaping capabilities are somewhat limited however, when they are compared to the equilibrium shaping approach proposed by Izzo and Petazzi, covered in section 6.4 or the extension by Izzo, Simões and de Croon. These extensions to the control strategy devised by Gazi solve the target assignment problem autonomously for symmetrical and both symmetrical and asymmetrical shapes respectively. Although, due to computational complexity and the required evolutionary training in the ER approach cause the approach of Izzo and Petazzi to be preferred over that by Gazi or Izzo, Simões and de Croon.

In three dimensions the approach by Izzo and Petazzi allows for the shaping of complicated Bravais lattices and simplifications to two-dimensional space still grant a higher degree of formation control where shapes are made possible.

Several control strategies, excluding the work by Olfati-Saber, Pinciroly et al. and Vásárhelyi et al., assume unrestricted field of view, which could lead to possible problems when scaling the algorithm, for instance due to computational constraints or inaccuracies measuring the inter-agent distances.

Although the equilibrium shaping and ER approach allow for the highest degree of formation control, the only algorithms which are designed with large groups of agents in mind are those by Olfati-Saber and Pinciroli et al., discussed in sections 6.6 and 6.7 respectively. When these algorithm are individually compared, the approach suggested by Olfati-Saber is computationally more complex and does not allow for the same degree of shaping.

The realistic features, such as delays, noise, refresh rates and inertial effect, included in the framework developed by Vásárhelyi et al. can not be found in any other method discussed. They allow for the evaluation of the swarms sensitivity to several realistic parameters such as wind or communication delay or error.

The decision by Vásárhelyi et al. to implement such realistic features in their simulation framework is a definite tribute to their approach. The simulation framework developed in the course of this project

could be based upon their realistic design in order to test robustness with respect to wind gusts for instance.

Lastly, there are several control strategies, by Izzo & Petazzi, Pinciroli et al., Vásárhelyi et al. and Izzo, Simões and de Croon, which allow for straightforward adaptation to a two-dimensional implementation. A two-dimensional swarm is more feasible at this point in time than a three-dimensional swarm, since it reduces the computational and vision requirements for the platform. When a two-dimensional set-up proves successful an extension to the third dimension could be made.

Concluding, the algorithms proposed by Izzo & Petazzi, Pinciroli et al. and Vásárhelyi et al., discussed in sections 6.4, 6.7 and 6.8 respectively, are selected for further study. They all show great promise in a two-dimensional environment, which could later be extended to three dimensions.

The work by Pinciroli et al. shows most promise when considering scalability to a large number of agents, the work by Izzo & Petazzi especially excels at detailed shape formation and the work by Vásárhelyi et al. is the only to consider wind gusts and time delays.

These control strategies will be tested for feasibility and compared further in Matlab simulations in the next stage of the project.

7

Neighbour sensing techniques

The successful implementation of any swarming technique is heavily dependant on the ability of every agent to detect its neighbours. A slight variation might not be undesirable, since in natural swarms the robustness of the swarm appears to result from the variance in accuracy[30]. However, an inability to sufficiently accurately detect its closest neighbours could result in collisions between agents or flock segmentation and separation.

When considering real world implementations of multi-agent systems several approaches to neighbour sensing techniques have been tried, as briefly discussed in chapter 3. This chapter will focus on vision and radio frequency based sensing .

7.1. Image recognition

Although image recognition has been around for over two decades, fast and robust image recognition algorithms for different applications are still an active topic of research[58, 59].

Ideally every agent in a multi-agents system would be able to detect location, range and pose of neighbouring agents. Considering the advanced, but nevertheless limited, processing power of the current generation of MAVs, detecting relative position and an estimate for the range would possibly suffice in order to achieve robust swarm dynamics. However, the scarcity of an MAV's computational power and processing time determine the restraints when using image recognition.

Generally, image recognition is a computationally intensive task, especially in real-time. Additionally, in order to attain accurate estimates of distances intensive training, using large sets of sample images, is often required[60]. Therefore this chapter will focus on relatively computationally less intensive forms of image recognition.

7.1.1. Classifier cascade

A paper on a multi-agent ground-based robotic swarm proposes an extremely rapid approach using trained classifier cascade for real-time pose and position tracking[61]. Using a cascade of simple classifiers, only detecting objects or non-objects, the algorithm is trained to detect only other agents and rejects all other frames.

In order to estimate the position and pose a multi-coloured agent is used as depicted in figure 7.1b. An illustration of this approach is given in figure 7.1a.

Whilst Katalenic et al. achieve impressive accuracies (>90%) and timing (15fps) on their mobile platform, a classifier cascade approach might not suffice for this project.

First, the considered distances are smaller than would be encountered in outdoor MAV flight. Increasing the distance will significantly reduce the accuracy.

Secondly, the on-board camera uses a low resolution sensor. When substituting the low-resolution sensor with a 14MP sensor, as is included on our platform, the computational demand will drastically

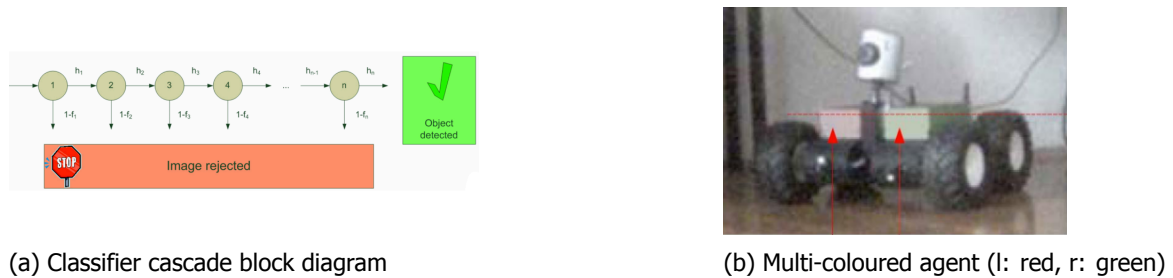


Figure 7.1: Algorithm and agent used for cascade detection. source: Katalenic, Draganjac, Mutka et al.[61]

increase and the frame-rate will drop.

This will cause the classifier cascade approach to either not be accurate or fast enough.

7.1.2. Light / Infrared markers

The use of LED lighting is also an efficient way to enhance the image recognition effectiveness[62]. Since LED lights emit brightly coloured or infrared (IR) light they are relatively easy to recognise due to the lack of other similarly bright objects in the field of view, reducing the chance of a false-positive. Additionally, since the objects stand out more compared to the environment less interference is to be expected. Examples of this approach are shown in figures 7.2a and 7.2b.

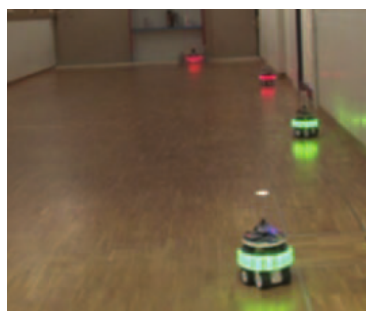
Distance measurement using only LED markers might prove complicated in a dynamic multi-MAV setup. In a two-dimensional scenario only the sides of other MAVs can be identified, hence a recognisable structure of LEDs should be visible from all viewing angles which in turn would require a large number of LEDs per MAV.

However, light markers can also be combined with other image recognition techniques to accentuate or enhance certain features in the image. Also in low-light conditions LED lighting can prove to be very beneficiary.

7.1.3. Fiducial markers

Fiducial markers are another computationally effective method to recognise objects in images[11, 63–66]. Since there is no need for an internal model, and the markers are constructed such that lighting and camera sensitivity play a minimal role the object can be recognised faster and relatively computationally efficient.

In recent years fiducial markers have been successfully used for small scale MAV formation flight[11]. Nägeli et al. propose a distributed control algorithm for environment-independent formation flight of MAVs, relying only on embedded sensing and agent-to-agent communication. Using on-board monocular cameras in order to estimate the relative distances, and UART bridges to transmit and receive inter-agent communication. They reach consensus on an inertial reference frame and estimate the formation state. Stable three-agent formation flight is achieved independent from any external sensing

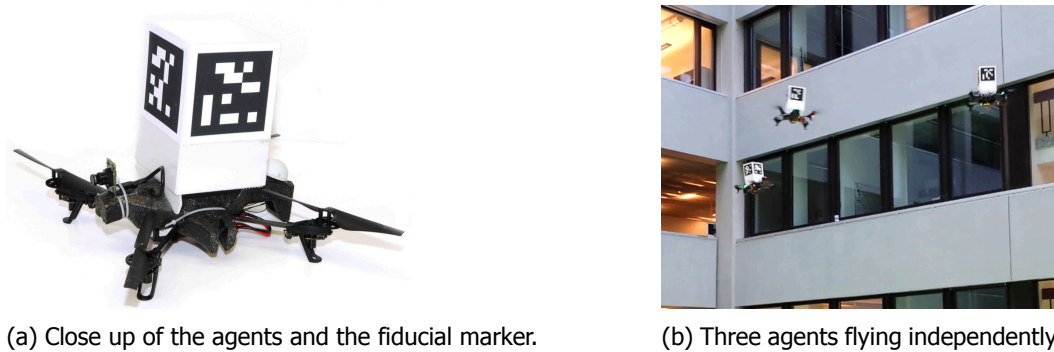


(a) Multiple ground-based robots.



(b) Single ceiling-suspended robot.

Figure 7.2: Position determination using LED lighting. Source: Dorigo, Floreano, Gambardella et al.[62]



(a) Close up of the agents and the fiducial marker.

(b) Three agents flying independently in formation.

Figure 7.3: Overview of the agent and result of the distributed formation flight. Source: Nægeli et al.[11]

structure such as e.g. GPS.

Figures 7.3a and 7.3b show the agents used by Nægeli et al. and a picture of their three agent formation in an outdoor environment respectively. Standard AR markers have been used in combination with the ARToolkitPlus library to attain the relative distance and orientation measurements of neighbouring agents[67].

However, Nægeli et al. do admit that whilst their proposed solution is theoretically seamlessly scalable to larger formations, practical limitations on the maximum number of members in the formation are present due to the linear scale in communication bandwidth. A problem that could possibly be overcome by replacing the consensus-agreed inertial frame by an accurate DGNSS reference frame and using self-assembling distributed formation control which will eliminate the need for inter-agent communication.

7.1.4. Conclusion

Although the field of image recognition for mobile applications is fast developing only few applications can be seen as a proven technology for MAVs[11, 61, 62]. The project requires fast robust distance tracking of multiple neighbours at once from a MAV so a lightweight and proven concept would be beneficial.

The most ideal tracking solution is presented in section 7.1.1 since it does not require any additional structure to be added to the agent. However, the algorithm is computationally complex and requires extensive training from sample images which can be a memory-costly and time-consuming task and the achievable accuracy is debatable.

The main selling point for tracking using illuminated or light emitting markers is excellent tracking quality in low-light conditions and enhanced performance over all lighting conditions. The ability to detect each other in the dark is a definite advantage of this approach. However, the light source consumes some of both the power and weight budget of each agent. Light or lighted markers could serve as a solution when low-light conditions are expected but otherwise should not be necessarily required.

The use of fiducial markers is the only concept considered which has been proven to work robustly on MAVs. Another advantage of the technique is that it requires minimal training and set-up and the supporting structure can be made from lightweight materials. Additionally fiducial markers can not only assess the distance between agents but can also unambiguously identify one another.

The use of fiducial markers will be considered for this project. The combination of a proven concept, easy implementation and set-up, low computational requirements and a low weight structure leads to fiducial markers being the ideal choice.

Fiducial markers could be made to work in a three dimensional environment where all sides of the agent should be marked, although they function much more efficiently in a planar two dimensional environment.

7.2. Radio frequency communication

Radio frequency communication can be used in order to detect nearby neighbours. Either the signal strength can be used as a distance measure or positional data can be transmitted. However using only the signal strength does not provide any directional data. Three types of short-range wireless networks are considered in this section: Bluetooth, ZigBee and WiFi.

7.2.1. Bluetooth

Bluetooth, also known as the IEEE 802.15.1 standard is a wireless radio system developed for short-range and low-cost communication. The personal operating space (POS), which defines the general communication range, is approximate 10 metres. There are two main connection topologies defined in the bluetooth standard: the piconet and the scatternet.

The piconet is a wireless personal area network (WPAN) formed by a Bluetooth device serving as master and one or more devices serving as slaves. A frequency hopping channel defined by the address of the master defines each piconet.

Slaves can only use point-to-point communication with the master whilst the master can transmit both point-to-point and point-to-multipoint.

Every piconet can exist of a maximum number of 7 slaves and a single master.

A scatternet is a collection of overlapping operational piconets. Two piconets can be connected in order to form a scatternet. Every bluetooth device may participate in several piconets at the same time, operating as a slave several times but being limited to a single master role.

Bluetooth uses the 2.4GHz band, which is unlicensed in most countries. It uses frequency hopping (FHSS) using 79 channels and 1MHz bandwidth, where channel collision is avoided using adaptive frequency hopping.

The typical maximum data rate of a bluetooth connection is about 0.72 Mbit/s with a maximum data payload of 339 bytes[68].

7.2.2. ZigBee

ZigBee over IEEE 802.15.4 is a standard for low-rate WPAN (LR-WPAN) developed to support simple low-powered devices operating in a POS of 10 metres, however in some applications this can be extended to 100m.

A ZigBee network can consist of two types of devices: a full-function device (FFD) and a reduced-function device (RFD). An FFD can communicate with RFDs and other FFDs, whilst a RFD can only communicate with one FFD at a time.

ZigBee uses the direct sequence spread spectrum (DSSS) with 16 channels and 2 MHz bandwidth. In order to avoid channel collision the protocol uses dynamic frequency selection and transmission power control.

The typical maximum data rate of a ZigBee connection is approximately 0.25 Mbit/s with a maximum data payload of 102 bytes[68].

A ZigBee star network can facilitate up to 65000 devices, whilst even more complex network-structures can be built such as cluster tree or mesh networks.

7.2.3. WiFi

Wireless fidelity (Wi-Fi) includes IEEE 802.11a/b/g/n/ac standards for wireless local area networks (WLAN). The basic cell of a 802.11 LAN is called a basic service set (BSS), which is a set of mobile and/or fixed stations. When a station moves out of its BSS it can no longer communicate directly to other members. A structured Wi-Fi BSS can cope with a maximum of 2007 devices[68].

Based on the BSS there are two additional types of service sets: the independent basic service set and the extended service set. Figure 7.4 portrays the different service sets.

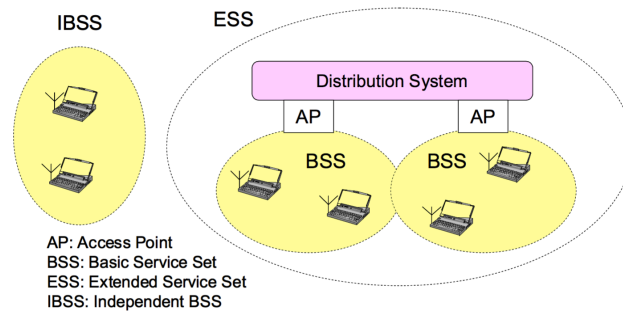


Figure 7.4: IBSS and ESS configurations of Wi-Fi networks. Source: Lee et al.[68]

Table 7.1: Comparison of the Bluetooth, ZigBee and Wi-Fi protocols. Source: Lee et al.[68]

	Bluetooth	ZigBee	Wi-Fi
IEEE spec.	802.15.1	802.15.4	802.11 a/b/g/n/ac
Frequency band	2.4 GHz	868/915 MHz; 2.4GHz	2.4 GHz; 5 GHz
Max signal rate	1Mb/s	250 Kb/s	11/54/150/866.7 Mb/s
Nominal range	10m	10-100m	100m
Nominal TX power	0-10 dBm	(-25)-0 dBm	15-20 dBm
Number of RF channels	79	1/10;16	14 (2.4 GHz)
Channel bandwidth	1 MHz	0.3/0.6 MHz; 2 MHz	22 MHz
Coexistence mechanism	Ad. freq. hopping	Dyn. freq. selection	Dyn. freq. selection
Basic cell	Piconet	Star	BSS
Extension of the basic cell	Scatternet	Cluster tree, Mesh	ESS
Max number of cell nodes	8	> 65000	2007

The IBSS allows stations to communicate directly without using an access point (AP), and is often referred to as an ad-hoc network.

The ESS consists of multiple BSS network connected through a distribution system. The distribution system allows for the creation of ESS networks of arbitrary size and complexity.

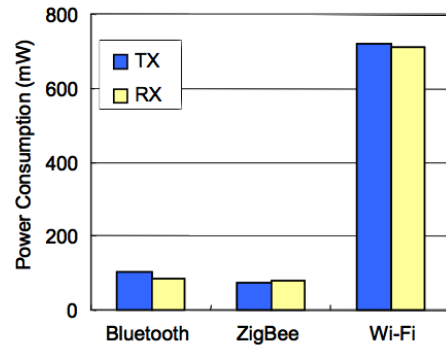
Wi-Fi uses DSSS (802.11), complementary code keying (CCK, 802.11b), or OFDM modulation with up to 14 RF channels and 22 MHz bandwidth. Like the ZigBee protocol it uses dynamic frequency selection and transmission power control in order to avoid channel collision.

It can operate, dependant on the used standard, at a maximum data rate of 11, 54, 150 or up to 866.7 Mbit/s using the b, a/g, n and ac standard respectively with a maximum data payload of 2312 bytes[68].

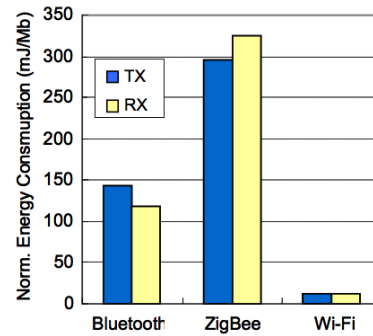
7.2.4. Conclusion

Three popular wireless standards have been examined in this section: Bluetooth, ZigBee and Wi-Fi. A comparison of the different standards is given in table 7.1. A comparison of the power consumption and normalised power consumption per Mb of data are given in figures 7.5a and 7.5b respectively.

Although a wide range of wireless standards is available, only small scale experiments, using up to 10 agents, have been conducted using wireless transmission of flight data[25]. The complexity of operating a network with hundreds of mobile agents on any standard is therefore a large uncertainty. Although wireless communication could provide a graceful method for transmitting position and velocity data, signal interference or error could affect the entire swarm potentially rendering them uncontrollable.



(a) Power consumption comparison



(b) Normalised power consumption comparison

Figure 7.5: Power consumption comparison of three wireless protocols: Bluetooth, ZigBee and Wi-Fi. Source: Lee et al.[68]

7.3. Sensing comparison

In order to assess whether vision or RF techniques should be implemented to detect neighbouring agents a comparison has to be made between these methods. Sections 7.1.4 and 7.2.4 briefly compare the considered methods.

Complexity is the major disadvantage and uncertainty to the RF approach. Although a swarm of a 100 agents has been demonstrated[34] it is hard to draw a definite conclusion on the feasibility of an inter-agent communication network of that size. Complications could arise when simultaneously using hundreds of transmitters/receivers, or achieving the imposed required bandwidth.

The image recognition challenge, although requiring more computational power, leads to less complexity when scaling up to hundreds of agents. The addition of agents will less likely cause the performance of other agents to drop when compared to the RF approach.

Variations of both methods have been successfully realised in swarming MAVs. Therefore it is above all a matter of practicality when selecting. All relevant hardware to minimally integrate a vision based techniques is available, whilst from the RF techniques only on-board Wi-Fi is available out-of-the-box, as mentioned in chapter 8. Therefore a vision based approach leaves more options to be explored in the next stage of the project. Although a inter-agent Wi-Fi based approach could be considered without modifying the platform, possibly as an extension or backup to the vision based approach.

8

The platform

The platform that will be used for the swarming demonstration is the Parrot Bebop 2 drone, depicted in figure 8.1. This drone is lightweight and has relatively high computational power. The out-of-box specifications can be found in table 8.1.

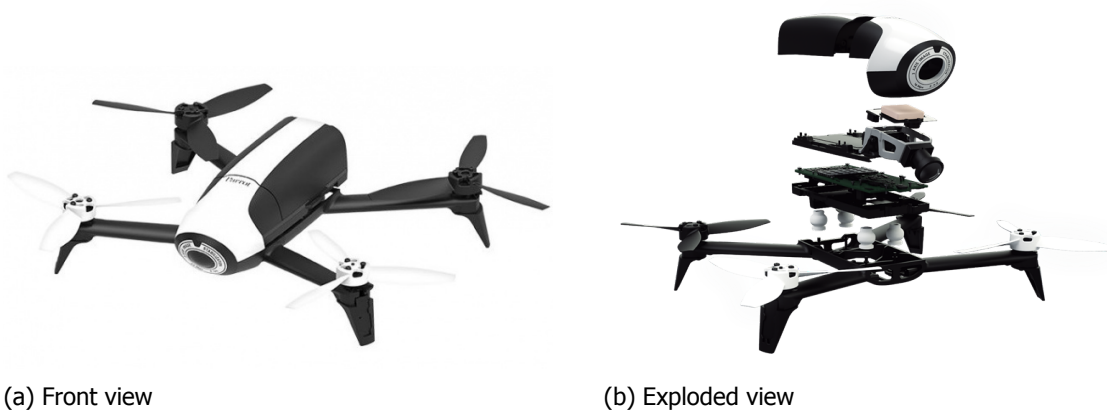


Figure 8.1: Parrot bebop 2 drone, source: parrot.com

8.1. Flight controller

Recent studies have shown that the micro UAVs produced by Parrot offer an excellent relatively low-cost platform for researchers to conduct experiments[69–71].

The swarming algorithm aimed to be designed in this paper will use the cascaded incremental nonlinear dynamic inversion (INDI) controller as proposed by E. Smeur at the TU Delft [72]. The autopilot used for the project: “Paparazzi the open-source autopilot” is explained in more detail in section 8.1.1 and the cascaded INDI flight controller is discussed in section 8.1.2.

8.1.1. Paparazzi open-source autopilot

Paparazzi open-source autopilot is a software collection created to give researchers and hobbyists the opportunity to construct and program their own MAV[73]. Other similar projects include: ArduPilot and OpenPilot[74, 75].

Researchers have successfully implemented the Paparazzi open-source autopilot on the Parrot AR 2 drone[71] and current work focusses on implementing the use of paparazzi on the Parrot Bebop 1 and 2 drones.

Table 8.1: Out of box specification of the Parrot Bebop drone

Parameter	Specification
Weight:	500g (including battery)
Dimensions:	38.2 x 32.8 cm
Battery:	LiPo 2700mAh removable battery (approx. 25 minutes of flight time)
Frame:	Glass fiber reinforced (15%) ABS structure
Motor:	4 brushless Outrunner motors
Propellor:	three-blade 6-inch auto-block polycarbonate propellers
Camera:	180° fisheye lens and 14 Mega pixel sensor (full HD 30fps)
Processor:	Parrot P7 dual-core CPU cortex 9, Quad core GPU, 8GB flash memory
Sensors:	3-axis magnetometer, 3-axes gyroscope, 3-axes accelerometer Optical flow sensor, Ultrasound sensor (up to 8m), pressure sensor
Geolocation:	GNSS (GPS + GLONASS)
Connectivity:	Wi-Fi MIMO dual-band, 2 sets of dipole (2.4 & 5 GHz) antennas (up to 21 dBm)

8.1.2. Inner control loop

MAVs are limited in their close proximity operation by their gust sensitivity. Incremental nonlinear dynamic inversion is an on-board sensor based control technique to control nonlinear systems subject to disturbances. It outperforms other commonly used controllers such as Proportional Integral Derivative (PID) control when rejecting wind gusts[72]. Due to the intended close proximity flight of the agents in an outside environment susceptible to gusts the swarming controller will rely on INDI for low level control.

8.2. Sensors & position determination

The Parrot Bebop 2 is equipped with a standard consumer grade GNSS sensor. Even though the low-level flight controller, discussed in section 8.1.2, is robust to GNSS inaccuracies [72], in order to facilitate accurate inter-agent-cooperation a more accurate GNSS technique is implemented. Section 8.2.1 explains the use of real-time kinematic GPS, a local augmentation system to GPS.

8.2.1. RTK GNSS

Most commercially available MAVs come equipped with standard low-cost GNSS receivers with a absolute position accuracy of 2-15m. This accuracy can be drastically increased by using a differential GNSS receiver. Differential GNSS uses a ground station with a known position which broadcasts disturbance and error information measured from the satellite signal to nearby receivers in order to enhance their accuracy.

Real Time Kinematics (RTK) is a specialised form of differential GNSS which also observes phase measurements in order to estimate the measurement error of the GNSS receiver[76]. RTK is capable of reducing GNSS accuracies from the meter level down to the centimeter or, using postprocessing, down to the millimetre level[77].

9

Project roadmap

This chapter will focus on the steps and decisions required to be made in order to achieve the goal of this project of swarming flight. The steps are presented in chronological order.

9.1. Literature study

The Literature study is the first building block of the project, in order to examine what research has been conducted and what developments can prove interesting a thorough review of the existing scientific field is done.

The literature study reviews the available feasible solutions found in research and concludes with proposed feasible control algorithms and vision based neighbour sensing strategies.

9.2. Platform selection

When the Literature study is finished a drone set-up has to be designed. This configuration of hardware on the platform, mentioned in section 8, is dependant both on the chosen control algorithm and its requirements and the neighbour sensing strategy and its respective requirements.

The design of the platform, after having been verified feasible, will be locked in this stage of the project in order to prevent conflicts arising later in the project due to changing platform specifications.

9.3. Algorithm selection

In this stage of the project a selection of the control algorithms proposed in the Literature study will be tested in order to assess their feasibility.

A Matlab simulation model for the swarming MAVs will be designed in which different control algorithms can be tested. Depending on the results and comparison between algorithms the most feasible algorithm will be selected for implementation.

9.4. Parameter selection

When a certain control algorithm is chosen, the Matlab simulation model will be used to assess the most optimal configuration of the parameters. In this stage parameters such as the inter-agent distance, visible range of each agent and other algorithm specific constants have to be selected and their effects on the stability of the swarm will have to be assessed.

9.5. Platform validation

When an optimal set of parameters is found for the theoretical model it has to be validated that the chosen platform can perform up to the required standard. For instance the neighbour sensing strategy has to be validated, otherwise the algorithm parameters discussed in the previous section will have to be re-examined and set to more realistically feasible levels.

Also the hardware and additional components of the platform will be tested such as the RTK DGNS and the camera. When the selected set of parameters prove to be feasible for the platform and all components function properly the project will continue to the next stage.

9.6. Algorithm validation

Once the most optimal and feasible set of variables for the chosen control algorithm is proposed, the design of the control laws will be frozen and the implementation of the algorithm verified and validated. When all the aspects of the control laws have been tested against theory the design for both the control laws and the platform will be locked.

9.7. Algorithm porting

In order to implement the chosen design laws on the platform the control and sensing strategies have to be programmed in a language native to the platform. A combination of code in C/C++ and Paparazzi will have to be written and tested in order to successfully port the project from it's high-level code in Matlab to a code which can be executed by the platform.

9.8. Small scale test

When the autopilot for the platform has been written a series of small scale tests will be conducted. Starting with a single-agent test and incrementally moving towards a swarm of approximately 6 agents. During this stage unforeseen changes to the control strategy can be first implemented and tested in the Matlab simulation framework and if proven successful be implemented in the control of the agents.

9.9. Full scale test

When all small scale test have proven successful a large scale test using 20 agents will be conducted. The completion of the full scale test will prove the project successful in realising scalable swarming flight.

Bibliography

- [1] R. S. Miller and W. J. D. Stephen, *Spatial relationships in flock of sandhill cranes (*Grus canadensis*)*, *Ecology* **47**, 323 (1966).
- [2] P. F. Major and L. M. Dill, *The three-dimensional structure of airborne bird flocks*, *Behavioral Ecology and Sociobiology* **4**, 111 (1978).
- [3] B. L. Partridge, *The Structure and Function of Fish Schools*, *Scientific American* **246**, 114 (1982).
- [4] G. Nicolis and I. Prigogine, *Self Organisation in Nonequilibrium Systems* (Wiley, New York, 1977).
- [5] C. W. Reynolds, *Flocks, herds and schools: A distributed behavioral model*, *ACM SIGGRAPH Computer Graphics* **21**, 25 (1987).
- [6] H. D. Parunak, *Go to the Ant: Engineering Principles from Natural Agent Systems*, *Annals of Operations Research* **75**, 69 (1997).
- [7] J. Toner and Y. Tu, *Flocks, herds, and schools: A quantitative theory of flocking*, *Physical Review E* **58**, 4 (1998).
- [8] M. W. Trahan, J. S. Wagner, K. M. Stantz, P. C. Gray, and R. D. Robinett, *Swarms of UAVs and Fighter Aircraft*, in *In Proceedings Second International Conference Nonlinear Problems in Aviation and Aerospace*, Vol. 2 (1998) pp. 745–752.
- [9] J. Macgill and S. Openshaw, *The use of Flocks to drive a Geographic Analysis Machine*, in *International Conference on GeoComputation* (1998).
- [10] C.-S. Park, M.-J. Tahk, and H. Bang, *Multiple Aerial Vehicle Formation Using Swarm Intelligence*, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, August (2003) pp. 11–14.
- [11] T. Nageli and C. Conte, *Environment-independent formation flight for micro aerial vehicles*, in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference* (2014) pp. 1141–1146.
- [12] B. Kate, J. Waterman, K. Dantu, and M. Welsh, *Simbeeotic: A simulator and testbed for micro-aerial vehicle swarm experiments*, in *Proceedings of the 11th international conference on Information Processing in Sensor Networks* (2012) pp. 49–60.
- [13] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, *Towards a swarm of agile micro quadrotors*, *Autonomous Robots* **35**, 287 (2013).
- [14] R. Moore and K. Dantu, *Autonomous MAV guidance with a lightweight omnidirectional vision sensor*, in *Robotics and Automation (ICRA), 2014 IEEE International Conference* (2014) pp. 3856–3861.
- [15] H. V. D. Parunak and P. D., *Making Swarming Happen*, in *Conference on Swarming and C4ISR* (2003).
- [16] D. Izzo and L. Pettazzi, *Self-assembly of large structures in space using intersatellite Coulomb forces*, *57th International Astronautical Congress, Valencia, Spain, Paper IAC-06-C3.4/D3.4.07* **4**, D3 (2006).
- [17] C. Pinciroli, M. Birattari, E. Tuci, M. Dorigo, M. d. R. Zaptero, T. Vinko, and D. Izzo, *Self-Organizing and Scalable Shape Formation for a Swarm of Pico Satellites*, in *NASA/ESA Conference on Adaptive Hardware and Systems* (IEEE, 2008) pp. 57–61.

- [18] A. Bürkle, F. Segor, and M. Kollmann, *Towards Autonomous Micro UAV Swarms*, *Journal of Intelligent & Robotic Systems* **61**, 339 (2011).
- [19] J. Welsby and C. Melhuish, *Autonomous minimalist following in three dimensions: a study with small-scale dirigibles*, *Computer Science Tech Report UMCS-01-4-1* (2001).
- [20] S. Hauert, S. Leven, and M. Varga, *Reynolds flocking in reality with fixed-wing robots: communication range vs. maximum turning rate*, in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE (2011) pp. 5015–5020.
- [21] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, *Precision flight control for a multi-vehicle quadrotor helicopter testbed*, *Control Engineering Practice* **19**, 1023 (2011).
- [22] M. Turpin, N. Michael, and V. Kumar, *Decentralized formation control with variable shapes for aerial robots*, in *2012 IEEE International Conference on Robotics and Automation* (IEEE, 2012) pp. 23–30.
- [23] T. Stirling and J. Roberts, *Indoor navigation with a swarm of flying robots*, in *IEEE International Conference on Robotics and Automation (ICRA)* (2012) pp. 4641–4647.
- [24] S. A. P. Quintero, G. E. Collins, and J. P. Hespanha, *Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach*, *American Control Conference (ACC), 2013* (2013).
- [25] G. Vasarhelyi, C. Viragh, G. Somorjai, N. Tarcai, T. Szorenyi, T. Nepusz, and T. Vicsek, *Indoor flocking and formation flight with autonomous aerial robots*, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2014) pp. 3866–3873.
- [26] M. Rubenstein, A. Cornejo, and R. Nagpal, *Programmable self-assembly in a thousand-robot swarm*, *Science* **345**, 795 (2014).
- [27] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, *Towards a swarm of agile micro quadrotors*, *Autonomous Robots* **35**, 287 (2013).
- [28] J. S. Mill, *A system of logic ratiocinative and inductive: Being a connected view of the principles of evidence and the methods of scientific investigation* (Harper, New York, 1884).
- [29] C. Eliasmith, *Dictionary of Philosophy of Mind*, (2007).
- [30] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems* (Oxford university press, 1999).
- [31] G. Beni and J. Wang, *Swarm Intelligence in Cellular Robotic Systems*, *Robots and Biological Systems: Towards a New Bionics? NATO ASI Series* **102**, 703 (1993).
- [32] G. Dudek, M. Jenkin, E. Miliotis, and D. Wilkes, *A taxonomy for swarm robots*, in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, Vol. 1 (IEEE, 1993) pp. 441–447.
- [33] G. Dudek, M. M. R. M. Jenkin, E. Miliotis, and D. Wilkes, *A taxonomy for multi-agent robotics*, *Autonomous Robots* **3**, 375 (1996).
- [34] Guinness World Records, *Intel stuns during CES keynote with record for most drones airborne simultaneously*, (2016).
- [35] E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, S. Aron, and S. Camazine, *Self-organization in social insects*, *Trends in Ecology and Evolution* **12**, 188 (1997).
- [36] P. P. Grassé, *La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs*, *Insectes Sociaux* **6**, 41 (1959).
- [37] E. Rimon and D. E. Koditschek, *Exact robot navigation using artificial potential functions*, *Robotics and Automation, IEEE Transactions* **8**, 501 (1992).

- [38] C. Holden, *On the scent of a data trail*, *Science* **278**, 5342 (1996).
- [39] H. Parunak, L. Purcell, F. Six, N. Station, and M. O'Connell, *Digital pheromones for autonomous coordination of swarming uav's*, *Ann Arbor* **1001**, 48105 (2002).
- [40] S. Brueckner and D. H. D. Burkhard, *Return from the Ant: Synthetic Ecosystems for Manufacturing Control*, *Ph.D. thesis* (2000).
- [41] K. Han, J. Lee, and Y. Kim, *Unmanned aerial vehicle swarm control using potential functions and sliding mode control*, in *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 222 (2008) pp. 721–730.
- [42] E. de Vries and K. Subbarao, *Cooperative control of swarms of unmanned aerial vehicles*, *International Journal of Robust and Nonlinear Control* **18**, 115 (2008).
- [43] V. Gazi and K. Passino, *A class of attraction/repulsion functions for stable swarm aggregations*, *International Journal of Control* **77**, 1567 (2002).
- [44] B. V. Gazi, *Stability analysis of swarms*, in *IEEE Transactions on Automatic Control*, Vol. 48 (2003) pp. 692–697.
- [45] V. Gazi, *Swarm aggregations using artificial potentials and sliding mode control*, *Robotics, IEEE Transactions* **21**, 1208 (2005).
- [46] D. Izzo and L. Pettazzi, *Equilibrium shaping: DISTRIBUTED motion planning for satellite swarm*, *European Space Agency, (Special Publication) ESA SP* **30**, 727 (2005).
- [47] D. Izzo, *Autonomous and Distributed Motion Planning for Satellite Swarm*, *Journal of Guidance, Control, and Dynamics* **30** (2007).
- [48] E. Large, H. Christensen, and R. Bajcsy, *Dynamic robot planning: Cooperation through competition*, in *IEEE International Conference on Robotics and Automation* (1997) pp. 2306–2305.
- [49] D. Izzo, L. F. Simões, and G. C. H. E. de Croon, *An evolutionary robotics approach for the distributed control of satellite formations*, *Evolutionary Intelligence* **7**, 107 (2014).
- [50] D. Miller, A. Saenz-Otero, J. Wertz, A. Chen, G. Berkowski, C. Brodel, S. Carlson, D. Carpenter, S. Chen, S. Cheng, D. Feller, S. Jackson, B. Pitts, F. Perez, and J. Szuminski, *Spheres: A testbed for long duration satellite formation flying in micro-gravity conditions*, *Advances in the Astronautical Sciences* **105**, 167 (2000).
- [51] P. Hughes, *Spacecraft attitude dynamics* (Courier Corporation, 2012).
- [52] K. Shoemake, *Uniform random rotations*, *Graphics Gems III Academic Press Professional* , 124 (1992).
- [53] R. Olfati-Saber, *Flocking for multi-agent dynamic systems: Algorithms and theory*, *IEEE Transactions on Automatic Control* **51**, 401 (2006).
- [54] R. Saber and R. Murray, *Flocking with obstacle avoidance: cooperation with limited communication in mobile networks*, in *IEEE International Conference on Decision and Control*, Vol. 42 (IEEE, 2003) pp. 2022–2028.
- [55] C. Kittel, *Introduction to solid state physics* (New York: Wiley, 1986).
- [56] W. Ren and R. Beard, *Distributed consensus in multi-vehicle cooperative control* (Springer-Verlag, London, 2008).
- [57] C. Virágh, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, and T. Vicsek, *Flocking algorithm for autonomous flying robots*. *Bioinspiration & biomimetics* **9**, 025012 (2014).
- [58] R. Schalkoff, *Digital image processing and computer vision* (New York: Wiley, 1989).

- [59] F. Bonin-Font, A. Ortiz, and G. Oliver, *Visual Navigation for Mobile Robots: A Survey*, *Journal of Intelligent and Robotic Systems* **53**, 263 (2008).
- [60] P. Moreels and P. Perona, *Evaluation of Feature Detectors and Descriptors based on 3D Objects*, *International Journal of Computer Vision* **73**, 263 (2007).
- [61] A. Katalenic, I. Draganjac, A. Mutka, and S. Bogdan, *Fast visual tracking and localization in multi-agent systems*, in *4th IEEE Conference on Industrial Electronics and Applications* (IEEE, 2009) pp. 1864–1870.
- [62] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugniere, G. Di Caro, F. Ducatelle, E. Ferrante, A. Forster, J. M. Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, N. Mathews, M. Montes de Oca, R. O’Grady, C. Pinciroli, G. Pini, P. Retornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stutzle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard, *Swarmanoid: A Novel Concept for the Study of Heterogeneous Robotic Swarms*, *IEEE Robotics & Automation Magazine* **20**, 60 (2013).
- [63] M. Fiala, *Vision guided control of multiple robots*, in *First Canadian Conference on Computer and Robot Vision, 2004. Proceedings.* (IEEE, 2004) pp. 241–246.
- [64] S. Y. Chiem and E. Cervera, *Vision-based robot formations with Bezier trajectories*, *Intelligent autonomous systems* **8**, 191 (2004).
- [65] D. Claus and A. Fitzgibbon, *Reliable Automatic Calibration of a Marker-Based Position Tracking System*, in *7th IEEE Workshops on Applications of Computer Vision (WACV/MOTION’05)*, Vol. 1 (IEEE, 2005) pp. 300–305.
- [66] O. Orqueda, *Vision-based control of multi-agent systems*, *Ph.D. thesis*, Oklahoma State University (2006).
- [67] D. Wagner and D. Schmalstieg, *Artoolkitplus for pose tracking on mobile devices*, *Computer Vision Winter Workshop - Graz University of Technology* (2007).
- [68] J.-S. Lee, Y.-W. Su, and C.-C. Shen, *A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi*, *33rd Annual Conference of the IEEE Industrial Electronics Society (IECON 2007)*, 46 (2007).
- [69] A. Visser, N. Dijkshoorn, M. V. D. Veen, and R. Jurriaans, *Closing the gap between simulation and reality in the sensor and motion models of an autonomous AR. Drone*, in *Proceedings of the International Micro Air Vehicle Conference and Flight Competition (IMAV11)* (2011) pp. 40–47.
- [70] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit, *The Navigation and Control technology inside the AR . Drone micro UAV*, *Proceedings of the 18th IFAC World Congress, 2011* **18**, 1477 (2011).
- [71] B. Remes, D. Hensen, F. V. Tienen, C. D. Wagter, E. V. D. Horst, and G. D. Croon, *Paparazzi : how to make a swarm of Parrot AR Drones fly autonomously based on GPS . in IMAV 2013: Proceedings of the International Micro Air Vehicle Conference and Flight Competition, Toulouse, France, September (2013)* pp. 17–20.
- [72] E. J. J. Smeur, G. C. H. E. de Croon, and Q. Chu, *Cascaded Incremental Nonlinear Dynamic Inversion Control for MAV Disturbance Rejection*, (2017), [arXiv:1701.07254](https://arxiv.org/abs/1701.07254) .
- [73] Paparazzi, *Paparazzii UAV*, .
- [74] copter.ardupilot.com, .
- [75] openpilot.org, .
- [76] S. Gopi, *Global positioning System: Principles and applications* (Tata McGraw-Hill Education, 2005).
- [77] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications* (Artech house, 2005).