

Flocking of Autonomous Unmanned Air Vehicles

Bill Crowther, Lecturer, School of Engineering, University of Manchester, w.j.crowther@man.ac.uk

Xavier Riviere: Research Student, xavier.riviere@caramail.com

Abstract

The use of large numbers of unmanned air vehicles in a given air space presents a challenge for conventional air traffic control methods. Flocking (or schooling, swarming or herding) in nature arises when mobile organisms find benefit in living at high densities. The present paper applies rules of flocking (cohesion, alignment, separation, and migration) to the problem of managing the flight of a number of autonomous unmanned air vehicles. The work is novel in that a non linear 6 degree of freedom aerodynamic model of an existing UAV is used to simulate the flocking flight vehicles. It is found that application of the cohesion and alignment rules is sufficient to generate true flocking behaviour in that the flight vehicle density is increased and the flock members converge on a common heading. Increasing rule strength reduces the time taken to achieve flocking behaviour. However increasing rule strength too far leads to oscillatory or unstable flight paths. It is found that flock behaviour can be adequately described using the time histories of two statistical parameters: the mean radius between flock members and the standard deviation of the flock members' heading angles.

Author Biographies

Bill Crowther has an undergraduate degree in Aeronautical Engineering (1990) and a PhD in High Angle of Attack Aerodynamics from the University of Bath (1994). He was at the Fluid Power department of the University of Bath from 1995 to 1997 working on the application of neural networks to fault diagnosis of hydraulic systems. He has been a lecturer at School of Engineering, University of Manchester since 1997 where his research interests include modelling and control of flight vehicles, flow control, external aerodynamics and neural networks.

Xavier Rivière was a student from 1998 to 2001 at ENSMA, one of the four leading French engineering schools in Aerospace. His research interests include pressure waves generated by trains entering tunnels, shock wave propagation in materials (CAE), modelling of flight vehicles and 3-D animations. He has a MEng in Mechanics and Aeronautics (2001) and is currently working at WS Atkins in Bristol, UK.

Nomenclature

n	number of boids
N	number of rules
p	roll rate (radians/s)
q	pitch rate (radians/s)
r	yaw rate (radians/s)
r	sensor radius (m)
\bar{r}	mean boid radius
s	Laplace operator
T	thrust (N)
u	aircraft velocity in x direction (m/s)
v	aircraft velocity in y direction (m/s)
w	aircraft velocity in z direction (m/s)
w	centroid weight parameter
W	rule weight
xyz	aircraft body axes (moves with aircraft)
XYZ	earth fixed axes
ψ	heading angle (radians)
ϕ	bank angle (radians)
σ_ψ	flock heading standard deviation (radians)
θ	pitch attitude (radians)
η	elevator deflection (radians)
ζ	rudder deflection (radians)
ξ	aileron deflection (radians)

Subscripts

d	(control system) demand
i	boid
j	flock boid
R	rule

1 Introduction

Increasing use of autonomous unmanned air vehicles in a variety of civil and military applications is putting increasing pressure on traditional airspace management capabilities. One solution to this problem is decentralise the management function by delegating control such that individual aircraft manage their own airspace by negotiating with neighbouring aircraft, i.e. free-flight. For example, two aircraft on a potential collision course will negotiate changes to their respective flight plans to remove the risk of collision¹. This negotiation takes place independently of a ground air traffic controller. However, for very high airspace densities, real time negotiation of alternative flight paths becomes impractical and it becomes necessary to impose a set of flight rules that minimises the probability of collisions a priori.

In nature, aggregations of large numbers of mobile organisms are also faced with the problem of

organising themselves efficiently. This selective pressure has led to the evolution of behaviour such as flocking of birds², swarming of insects, herding of land animals and schooling of fish^{3,4}. The reasons why organisms form flocks are varied and include protection from predation, improved food search and improved social cohesion. However, the actual dynamics of the flocking behaviour are essentially constrained by the dynamics of the individual organisms and the flock is relatively limited in the types of behaviour it can exhibit. This gives a flock of a given organism, be it fish or bird, its characteristic look and feel.

A flock may be loosely defined as a clustered group of individuals with a common velocity vector. Note that flocking of aircraft is different to formation flying⁵. In the latter, aircraft are arranged according to predefined relationships that generally remain fixed during the flight. With flocking flight, there are no predefined relationships and the flock members may constantly change their position within the group. The fixed relationships within aircraft formations makes them relatively difficult to manoeuvre whereas the fluid nature of a flock allows relatively rapid changes in flock direction.

The aim of the present work is to demonstrate flocking behaviour of a group of simulated unmanned air vehicles. Of particular interest is the development of meaningful statistical metrics that usefully quantify flocking behaviour and the investigation of the relationship between rule weighting and flocking behaviour. An important aspect of the work is that a full 6-degree of freedom nonlinear aerodynamic model of an actual flying unmanned air vehicle is used as a basis for the simulated flocking behaviour. This is in contrast to previous simulation studies of flocking behaviour^{6,7} that have tended to use particle models or very simplistic aerodynamic models⁸ and as a result have avoided many of the real practical engineering issues involved in flocking of UAVs.

2 Rules of Flocking

The viability of obtaining coherent flocking behaviour from simple rules was first demonstrated by Reynolds⁸. The primary application for this work was in developing realistic motions of groups of 'actors' in computer animation. Conventionally, each 'actor' (known generically as a boid in flocking work) has a scripted path predetermined by the animator. For large numbers of boids, for example a flock of birds, the process was cumbersome and did not produce realistic results. This led to the use of relatively simple flocking rules that would automatically govern the dynamic behaviour.

The flocking rules used in the present work are illustrated schematically in figure 1. The cohesion rule, figure 1a, acts such that the active flock member (outlined triangle located in the centre of the diagram) tries to orient its velocity vector in the direction of the centroid (average spatial position) of the local flock. The degree of locality of the rule is determined by the sensor range of the active flock member, represented diagrammatically by the light coloured circle. Note that for effective cohesion it is necessary to vary the speed of the active flock member as well as its heading such that the speed of boids far from the flock centroid is increased. This is referred to as speed cohesion and allows wayward boids to catch up with the rest of the flock.

The alignment rule, figure 1b, acts such that the active flock member tries to align its velocity vector with the average velocity vector of the local flock. Once again, the degree of locality of the rule is determined by the sensor range.

Simulations show (see results sections) that reasonable flocking behaviour can be obtained using just cohesion and alignment rules.

Left unchecked, the cohesion rules will tend to lead to flock overcrowding. To balance this, a separation rule is used, figure 1c, where the active flock member tries to translate away from the local flock centroid. Note that for effective flocking behaviour, the sensor range of the cohesion rules will generally be much larger than the separation rule, i.e. cohesion acts at a global level whereas separation works locally.

In practice, the separation rule is not always sufficient to prevent a collision between flock members and it is necessary to implement an evasion rule, figure 1d. This rule is an extremely local version of the separation rule, with the avoidance centroid being equal to the position of the nearest flock member in the range of the sensor. Note that the sensor field for evasion is biased in the forward direction. To achieve effective evasion it is necessary to use a small guidance interval (<0.2 seconds). At a guidance interval of 2 seconds (as used in the present study) evasion is non-effective so was not implemented.

The final rule used in the present study is migration, figure 1e. This rule makes flock members translate towards an arbitrary fixed point in space. It is used to provide the flock with waypoints as a means of translating the flock en masse along a predefined path.

To achieve comprehensive flocking behaviour, the actions of all the rules are weighted and summed to give a net velocity vector demand for the active flock member, figure 1f. The actual behaviour of the flock

depends strongly on the weightings applied to each rule.

Note that the above flocking rules are applicable to any sized flock with 2 or more members.

3 Software implementation

3.1 Boid aerodynamic model

The boid mathematical model used for the present simulation work is based on an existing conventional propeller driven UAV of span 2m and mass 3.5kg. Static aerodynamic derivatives were obtained during full-scale tests in the University of Manchester 7'x9' wind tunnel. Dynamic derivatives were estimated using design methods. The model is implemented in Matlab Simulink.

3.2 Flight control system

The flight control system for the boid model uses a multivariable state feedback approach, figure 2. A controller gain matrix is designed using the Linear Quadratic Regulator (LQR) method, based on a linearised aircraft model. For the present work, the model was linearised about the cruise operating point ($u=18\text{m/s}$, $w=1.2\text{m/s}$, $\theta=0.12$, all other states equal to zero). The controller is tuned by varying the cost functions used in the LQR design process. These cost functions are based on the controlled state errors and actuation levels. Cost functions are adjusted until an acceptable compromise between response, stability and actuation levels is obtained.

Guidance inputs to the control system are speed, heading angle and pitch attitude demand (u_d , ψ_d and θ_d). The controller outputs throttle, elevator, aileron and rudder positions (T , η , ξ and ζ).

3.3 Flocking algorithm

The flocking algorithm works as follows: For a given boid, centroids are calculated using the sensor characteristics associated with each flocking rule. Next, the velocity vector the given boid should follow to enact the rule is calculated for each of the rules. These velocity vectors are then weighted according to the rule strength and summed to give an overall velocity vector demand. Finally, this velocity vector demand is resolved in to a heading angle, pitch attitude and speed demand, which is passed to the control system. The control system then outputs an actuator vector that alters the motion of the aircraft in the appropriate manner.

Details of the mathematical implementation of the flocking rules are presented in Appendix A.

4 Results

4.1 Overview

Simulation results will be presented to show the effect of changing rule weightings. The baseline test case is a flock of 10 boids initially placed at random positions and orientations within a box in space 400m wide, 400m deep and 200m high. Heading, attitude and velocity guidance demand vectors for each boid based on flocking rules are calculated every 2 seconds, i.e. a guidance interval of 2 seconds. Note that simulation time increases linearly with increasing guidance update rate and exponentially with number of boids. A 2-second guidance interval and 10 boids were chosen to give a reasonable compromise between flock modelling fidelity and simulation time. A typical 60s simulation took approximately 3 minutes to compute on a 600MHz PC. Significant increase in speed could be obtained by using compiled code rather than Matlab interpreted code.

4.2 Flock statistics

The two most important attributes of a flock were found to be its density (how close its members are to each other) and the degree of alignment of member velocity vectors. For the present work flock density is represented by the mean radius between flock members, \bar{r} and flock alignment by the standard deviation of the flock member heading angles, σ_{ψ} . Note that for a randomly orientated flock, σ_{ψ} is equal to 1 radian and for a perfectly aligned flock $\sigma_{\psi}=0$.

4.3 Cohesion only

The effect of application of the cohesion rule alone on flock behaviour is shown in figure 3. Figure 3a shows an X-Y plot of boid position for the baseline case of no rules applied. Note that the spacing between each aircraft symbol is 1 second and that the aircraft symbols are drawn a factor 4 larger than the actual aircraft modelled. With no rules applied, boid velocity remains constant, as would be expected.

With the cohesion rule implemented at a strength of 0.5 ($W_{\text{cohesion}}=0.5$), the boids form a circling mass after release, figure 3b. Note that boid speed is not affected by the cohesion rule and remains constant (within the capabilities of the flight control system). Increasing the cohesion rule strength, figures 3c and d, increases the density of the flock. Figure 3d also illustrates that cohesion in the vertical axis occurs less rapidly than in the lateral sense. This is a limitation imposed by stability margins of the flight control system.

Figures 3e and f illustrate the flock behaviour in terms of time histories of the mean boid radius, \bar{r} , and the heading standard deviation, σ_{ψ} . At low rule strength ($W_{\text{cohesion}} = 0.2$) the boids drift apart

following release (\bar{r} increases), figure 3e. As the rule strength increases ($W_{\text{cohesion}} > 0.5$) the steady state flock density is increased. However, if the rule strength is increased too far ($W_{\text{cohesion}} > 1$), the flock density tends to oscillate. Note that the frequency of oscillation is a function of the flight vehicle and control system dynamics, and the cohesion rule weighting.

Figure 3b shows that the orientation of the flock members is independent of cohesion rule strength (as would be expected).

4.4 Alignment only

The effect of application of the alignment rule alone on flock behaviour is shown in figure 4. The X-Y plots, figures 4a and b, show that the alignment rule drives the boids to fly in the same direction. Increasing the strength of the rule decreases the time taken to achieve a uniform heading angle.

Flock statistics for application of the alignment rule are shown in figures 4c and d. At low alignment rule strength ($W_{\text{alignment}} < 0.3$) the flock direction converges eventually after about 50 seconds, however, from figure 4c, it can be seen that the flock is widely dispersed. Increasing the alignment rule strength ($W_{\text{alignment}} \approx 1$) leads to full flock convergence in about 15 seconds, figures 4b and d. Further increases in rule strength ($W_{\text{alignment}} > 1.2$) leads to oscillatory behaviour in the boid heading angles and time to convergence increases.

4.5 Cohesion and Alignment

True flocking behaviour can be obtained by simultaneous application of cohesion and alignment rules, figure 5. In figure 5c, the alignment weighting has been set to the optimal value of 1 as identified from figure 4d. The different curves show the effect of varying cohesion rule weighting. If W_{cohesion} is set too high, e.g. 1, the flock density *decreases* and the heading angles become oscillatory.

If the cohesion weighting is reduced ($W_{\text{cohesion}} < 0.5$) the desired result of a relatively cohesive flock with aligned velocities is achieved, figures 5c and 5d. However, the boids are still relatively widely spaced along the flight path direction due to the fixed cruise speed constraint. Figure 4b shows an X-Y plot with the speed cohesion rule enabled. Corresponding flock statistical measures are shown in figures 5e and f. Application of the speed cohesion rule enables the minimum mean boid radius to be reduced from approximately 90m to 20m. To put this in context, 20m is approximately equivalent to 1 second of separation at the cruise speed of 18m/s. Note that to achieve higher flock densities it is necessary to reduce the guidance interval (which is currently 2 seconds). Note also that the velocity cohesion rule

does not affect the boid heading standard deviation, figure 5f, as might be expected.

4.6 Cohesion, Alignment and Separation

The effect of the separation rule on the flock behaviour is shown in figure 6. The separation rule is necessary to achieve an even flock density and prevent the need for evasive manoeuvres. However this rule tends to work against the cohesion and alignment rules leading to oscillatory behaviour for separation rule strengths of greater than 2. In practice the separation rule needs careful tuning in concert with the sensor radius and forward bias associated with the separation centroid.

5 Conclusions

- Flocking offers a potentially simple and efficient way of managing the flight paths of a large number of small autonomous UAVs such that the risk of collision and/or the need for evasive manoeuvres is reduced.
- The way in which flocking rules are implemented depends strongly on the nature of the flight control system available on the target flight vehicle.
- Basic flocking behaviour can be obtained as the result of application of just two rules: cohesion and alignment. However the effects of rules are not simply additive.
- The time taken to achieve coherent flocking behaviour is reduced by increasing the cohesion and alignment rule strengths. However increasing the strengths too far leads to oscillatory behaviour and increased flock convergence time.
- Flock behaviour over time can be characterised usefully by the time variation of two statistical parameters: the mean radius between flock members (flock density) and the heading angle standard deviation.

6 References

1. Burdun, I. and Parfentyev, O., 'AI knowledge model for self-organizing conflict prevention/resolution in close free-flight air space', IEEE Aerospace Applications Conference Proceedings, Vol. 2, 1999, pp. 409-428.
2. Potts, W., 'The chorus line hypothesis of manoeuvre coordination in avian flocks', letter in nature, Vol. 309, May 24, 1984, pp. 344-345.
3. Shaw, E., 'Fish in schools', Natural History 84, No. 8, 1975, p. 4046
4. Partridge, B., 'The structure and function of fish schools', Scientific American, June 1982, pp. 114-123.
5. Anderson, M., Robbins, A., 'Formation flight as a cooperative game', AIAA Guidance, Navigation and Control Conference and Exhibit, Boston, MA, August 1998, Technical Papers pt 1 (A98-37001 10-63), pp. 244-251.
6. Czirok, A., Vicsek, M., Vicsek, T., 'Collective motion of organisms in three dimensions', Physica A, Vol. 264, No. 1-2, Feb. 1999, pp. 299-304.
7. Toner, J., Tu, Y., 'Flocks, herds, and schools: A quantitative theory of flocking', Physical Review E, Vol. 58, No. 4, October 1998, pp. 4828-4858.
8. Reynolds, C., 'Flocks, Herds, and Schools: a distributed behavioural model', Computer graphics, 21(4), July 1987, pp. 25-34.

Appendix A

Mathematical implementation of flocking rules

A1 Centroid calculation

For a given boid, i , each flocking rule, R , operates on a centroid specific to that rule. These centroids are calculated using an imaginary sensor mounted on the given boid. The sensor interrogates a spherical region of space according to a Gaussian weighting function, with objects at the edge of the volume having less weight than those at the centre. The range of the sensor is set by specifying a sensor radius r_s for each rule. The sensor focus is located ahead of the boid in the direction of the boid's velocity vector by a distance specified by the sensor's forward bias.

At each guidance interval, a centroid for each rule is required for each of the boids. Firstly, for a given boid and rule, the displacements between the given boid and the flock boids, $j = 1..n$, are calculated, equation A1.

$$\mathbf{e}_{R,ij} = \mathbf{X}_{S_{R,i}} - \mathbf{X}_j \quad \text{A1)}$$

Where $\mathbf{X}_{S_{R,i}}$ is the location of the sensor focus for boid i and rule R , and \mathbf{X}_j is the position of flock boid j .

Note that for the alignment rule, the ‘error’ for each flock boid is equal to its velocity vector and equation A1 is replaced with equation A2.

$$\mathbf{e}_{\mathbf{r},ij} = \mathbf{V}_j \quad \text{A2)}$$

Where \mathbf{V}_j is the velocity of flock boid j .

Next, the weight associated with each flock boid’s contribution to the centroid is calculated according to a Gaussian basis function, equation A3, where r_{SR} is the sensor radius for rule R .

$$w_{ij} = \exp\left(-\left(\frac{|\mathbf{e}_{\mathbf{r},ij}|}{r_{SR}}\right)^2\right) \quad \text{A3)}$$

Finally, the centroid is calculated using equation A4.

$$\mathbf{X}_{\mathbf{C}_R i} = \frac{\sum_{j=1}^n w_j \mathbf{e}_{\mathbf{r},ij}}{\sum_{j=1}^n w_{ij}} \quad (j \neq i) \quad \text{A4)}$$

Where $\mathbf{X}_{\mathbf{C}_R i}$ is the centroid position vector for rule R and boid i , and n is the number of boids in the flock.

The migration centroid is pre-specified as part of the navigation function and common to all boids.

A2 Guidance

The guidance algorithm determines heading, pitch attitude and speed demand for each boid based on the pre-specified rule strengths and calculated rule centroids. For each boid and for each rule (apart from alignment), and noting that the i subscript has now been dropped for simplicity, a position error vector is calculated:

$$\mathbf{X}_{\mathbf{E}_R} = \mathbf{X}_{\mathbf{C}_R} - \mathbf{X}_B \quad \text{A5)}$$

For the alignment rule

$$\mathbf{X}_{\mathbf{E}_R} = \mathbf{X}_{\mathbf{C}_R} \quad \text{A6)}$$

For the attractive rules, e.g. cohesion, the direction of the velocity demand has the same sign as the position error and the magnitude decreases as the magnitude of the position error decreases, equation A7. For the repulsive rules, e.g. separation, the direction of the velocity demand has the opposite sign as the position error and the magnitude increases as the magnitude of the position error decreases, equation A8.

$$\mathbf{V}_{\mathbf{D}_R} = \mathbf{W}_R \frac{\mathbf{X}_{\mathbf{E}_R}}{|\mathbf{X}_{\mathbf{E}_R}|} \left(1 - \exp\left(-\left(\frac{|\mathbf{X}_{\mathbf{E}_R}|}{r_R}\right)^2\right) \right) \quad \text{A7)}$$

$$\mathbf{V}_{\mathbf{D}_R} = \mathbf{W}_R \frac{\mathbf{X}_{\mathbf{E}_R}}{|\mathbf{X}_{\mathbf{E}_R}|} \exp\left(-\left(\frac{|\mathbf{X}_{\mathbf{E}_R}|}{r_R}\right)^2\right) \quad \text{A8)}$$

Where \mathbf{W}_R is the weighting (or strength) assigned to rule R . Note that for the alignment rule, a modified version of equation A7 is used.

The overall velocity demand is obtained by summing the velocity demands from all the rules, equation A9.

$$\mathbf{V}_D = \sum_{R=1}^N \mathbf{V}_{\mathbf{D}_R} \quad \text{A9)}$$

Where N is the number of rules.

Finally, the heading and pitch attitude demand angles are calculated by resolving the velocity vector demand into the XY and xZ planes, respectively. The heading angle demand is bounded to be between $+\pi$ and the pitch angle between $+\pi/4$. The speed demand is calculated according to a weighted form of the cohesion velocity vector demand. When a boid is far from the cohesion centre its demanded speed is increased according to the speed cohesion weighting. This gives a maximum increase in speed of 20% of the cruise speed.

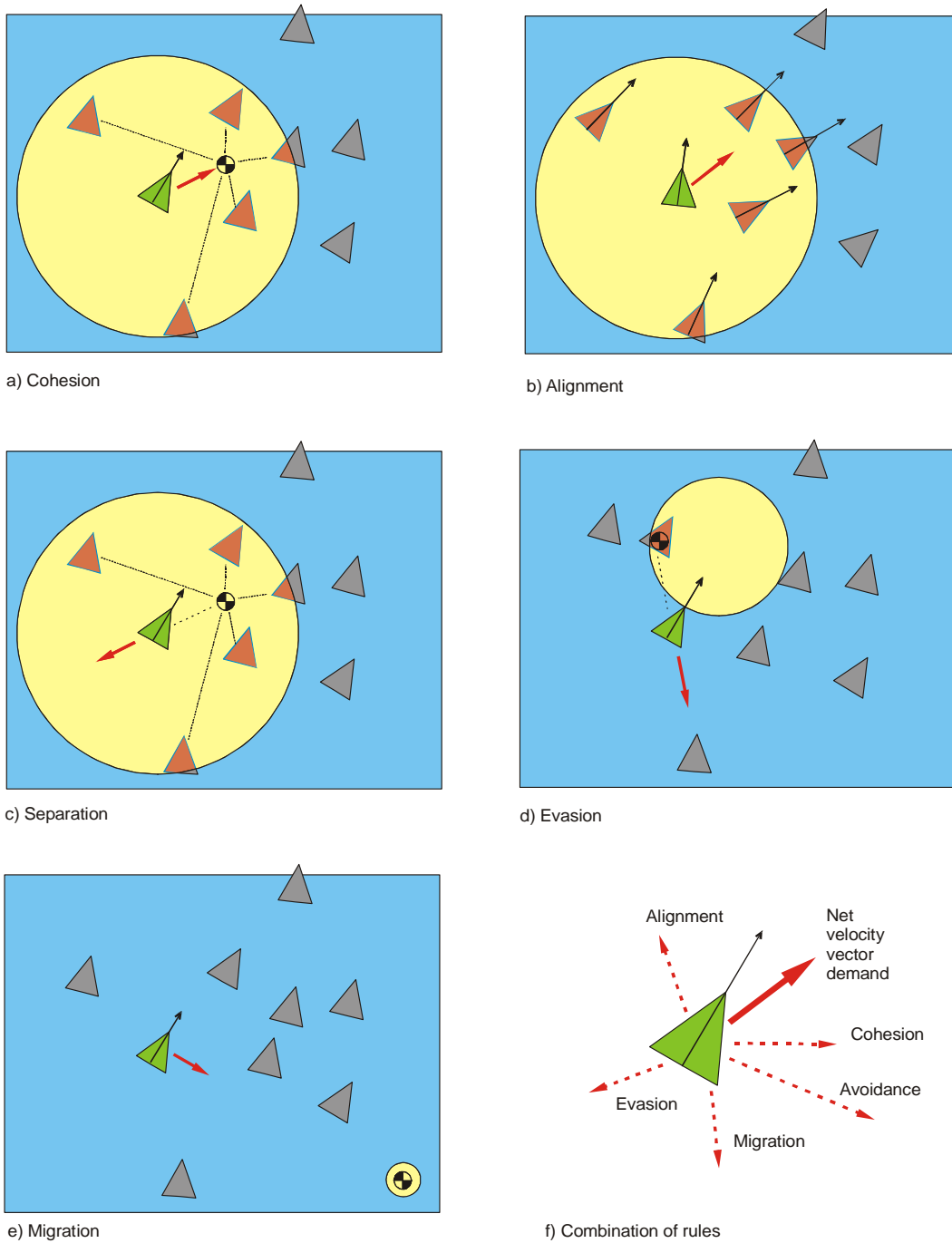


Figure 1 Rules of flocking

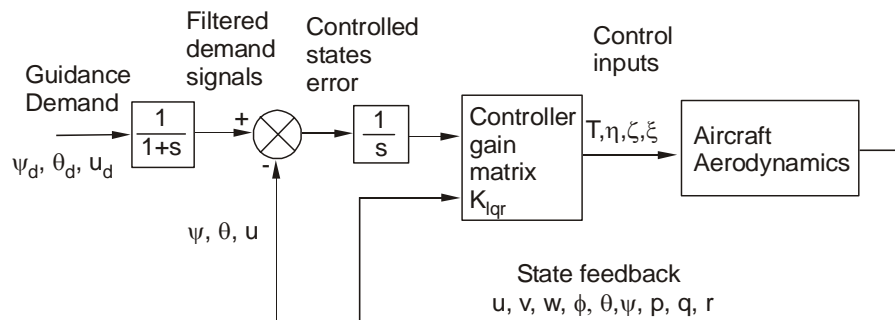
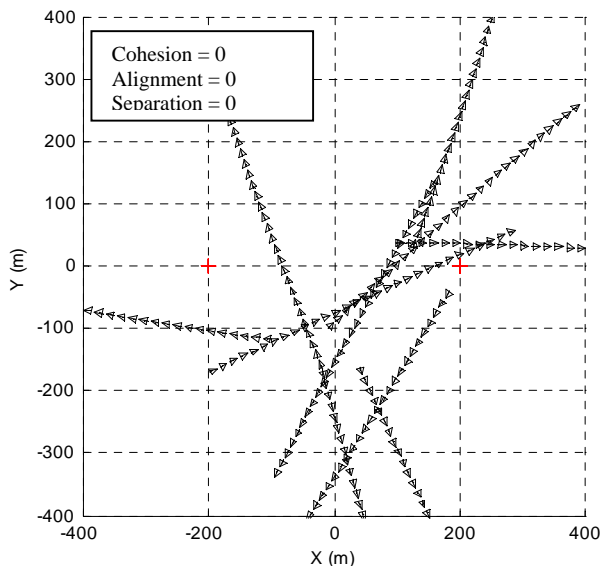
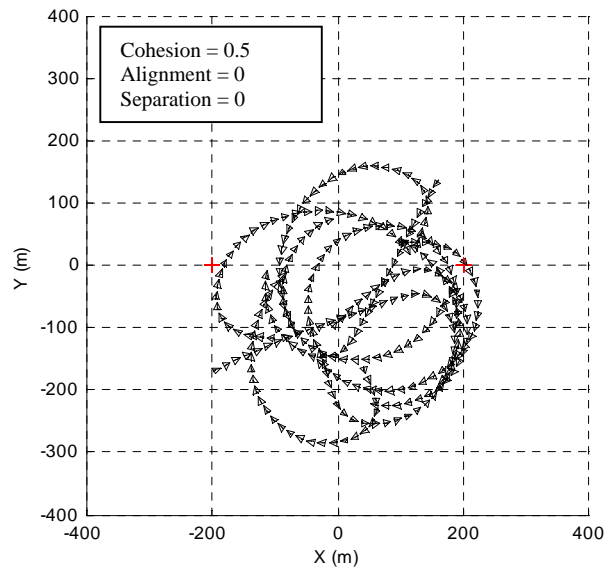


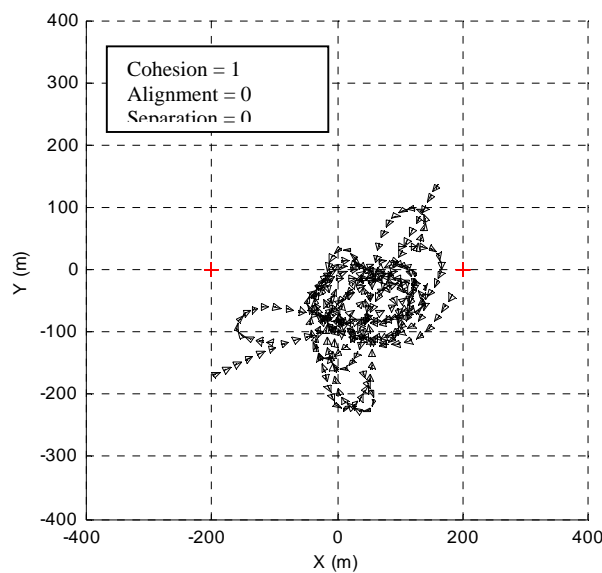
Figure 2 Boid flight control system



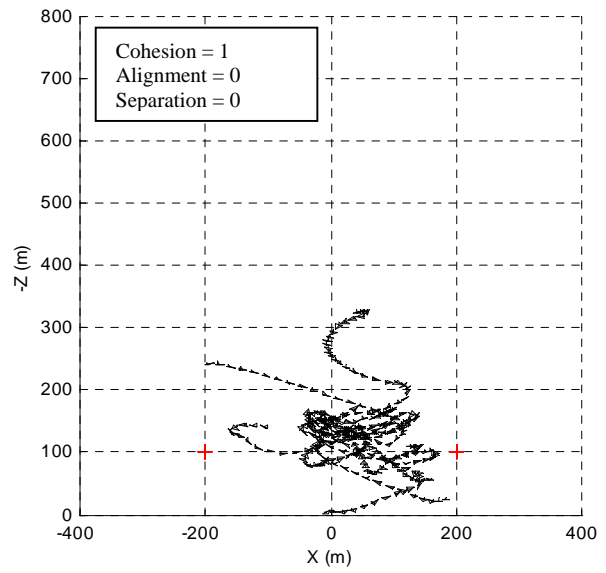
a) Boid X-Y plots for no rules applied



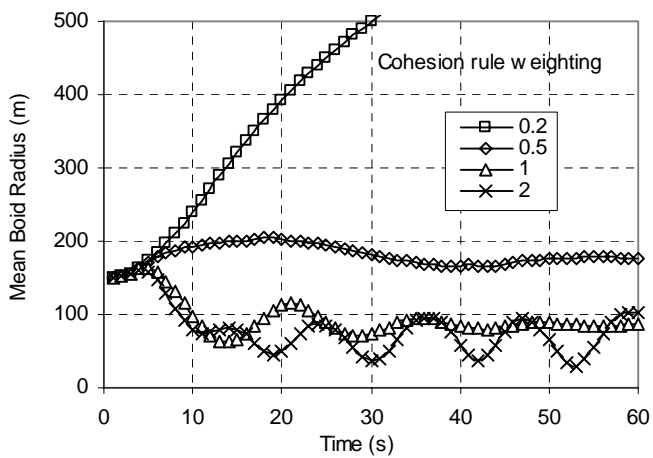
b) Boid x-y plots for cohesion rule weight = 0.5



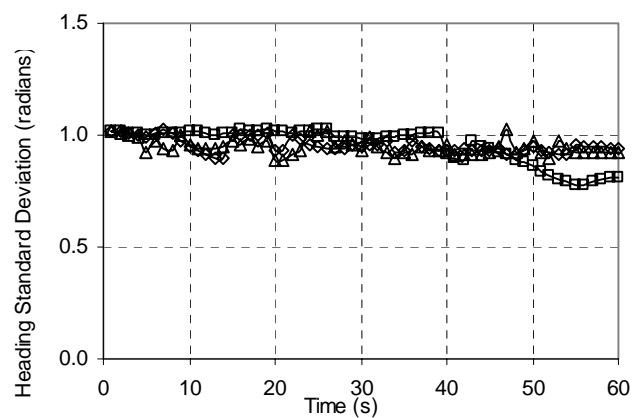
c) Boid X-Y plots for cohesion rule weight = 1



d) Boid X-Z plots for cohesion rule weight = 1

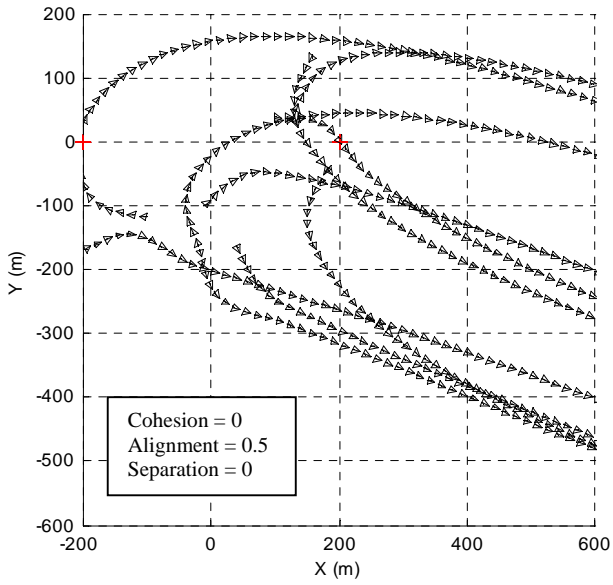


e) Flock density time histories for varying cohesion rule strength

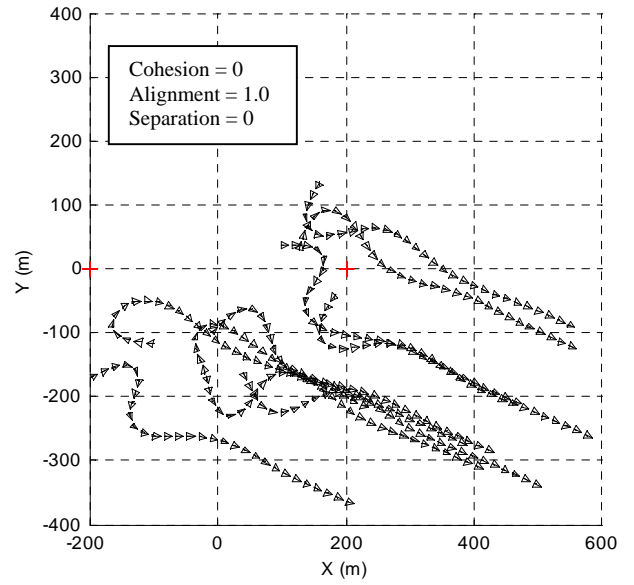


f) Flock heading angle standard deviation time histories for varying cohesion rule strength

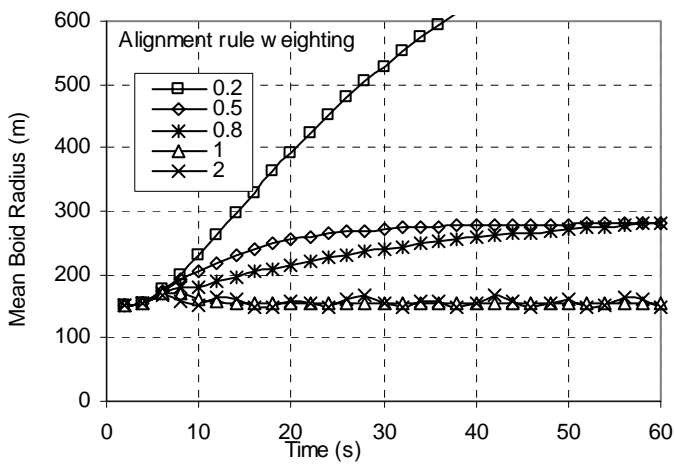
Figure 3 Effect of cohesion rule strength on flock behaviour



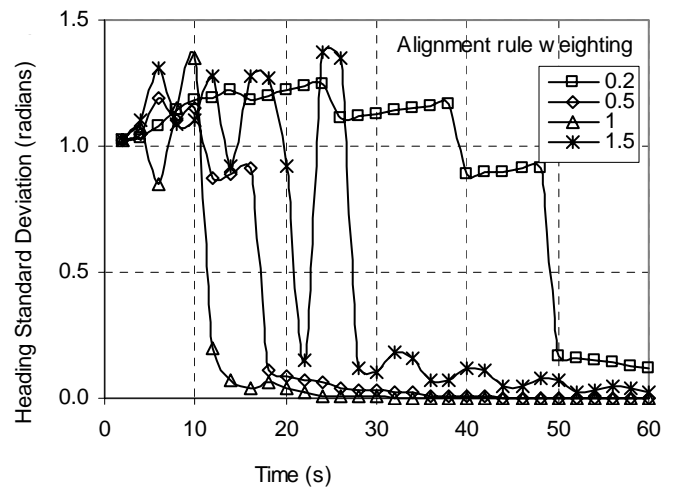
a) Boid X-Y plots for alignment weight = 0.5



b) Boid X-Y plots for alignment weight = 1.0

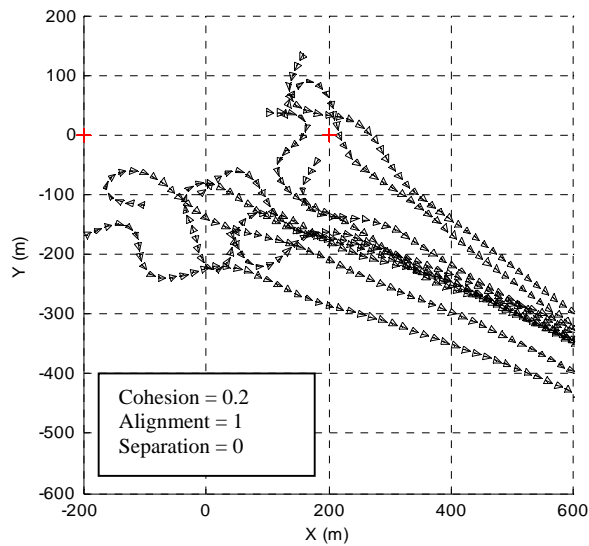


c) Flock density time histories for varying alignment rule strength

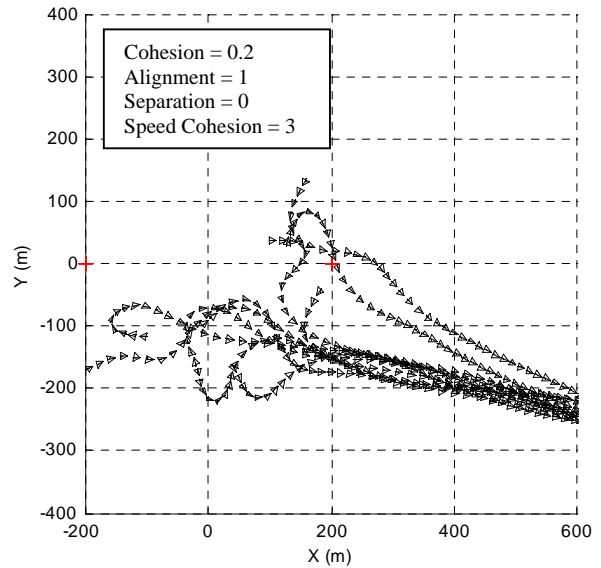


d) Flock heading angle standard deviation time histories for varying alignment rule strength

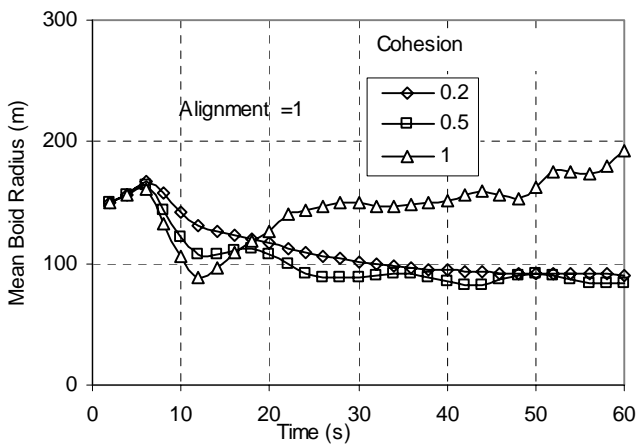
Figure 4 Effect of Alignment rule strength on flock behaviour



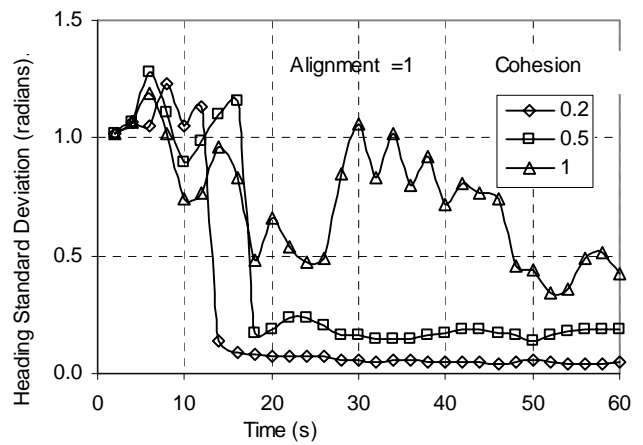
a) Cohesion = 0.2, Alignment = 1



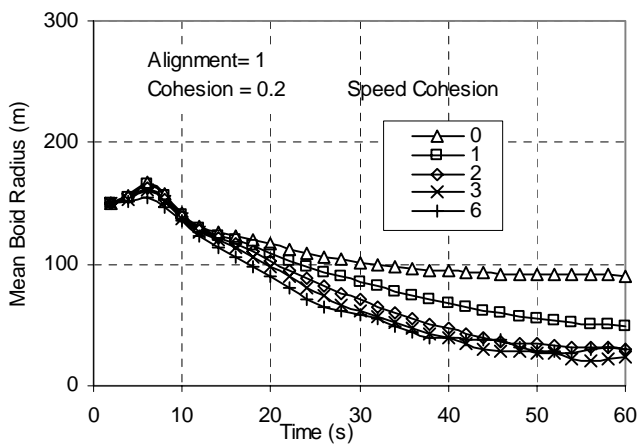
b) Cohesion = 0.2, Alignment = 1, Speed Cohesion = 3



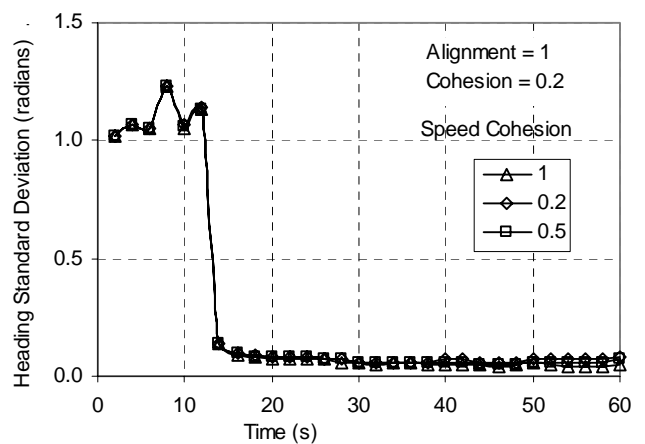
c) Effect of combined cohesion and alignment on flock density



d) Effect of combined cohesion and alignment on flock heading standard deviation

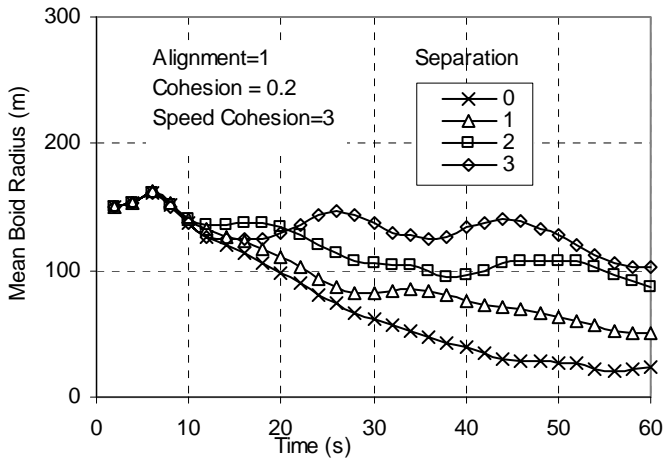


e) Use of Speed Cohesion rule to achieve high flock density

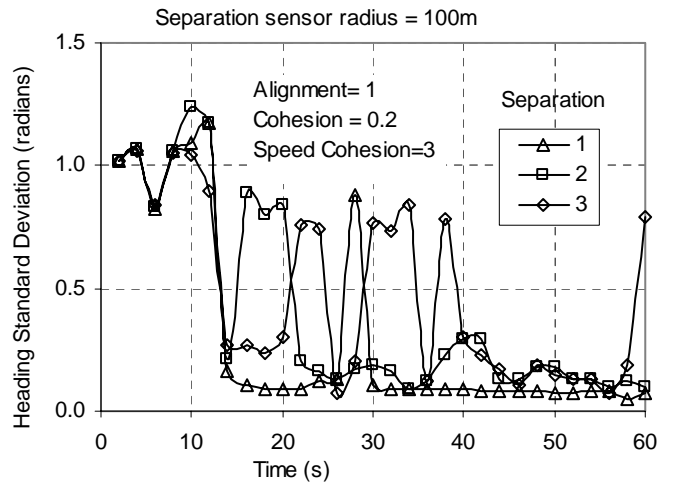


f) Demonstration that the Speed Cohesion rule does not affect flock heading angle standard deviation

Figure 5 Effect of cohesion and alignment rules on flock behaviour



a) Mean boid radius



b) Heading angle standard deviation

Figure 6 Use of the separation rule to reduce flock density