Don't Push Me!
Collision-Avoiding Swarms

T. M. Blackwell and P. Bentley
Department of Computer Science
University College London, Gower St.
London, UK
tim@theHotandtheCool.co.uk, P.Bentley@cs.ucl.ac.uk

Abstract: This paper examines a novel particle swarm algorithm which has been applied to the generation of interactive, improvised music. An important feature of this algorithm is a balance between particle attraction to the centre of mass and repulsive, collision avoiding forces. These forces are not present in the classic particle swarm optimisation algorithms. A number of experiments illuminate the nature of these new forces and it is suggested that the algorithm may have applications to dynamic optimisation problems.

I INTRODUCTION AND BACKGROUND

In a previous paper we have described the application of swarm intelligence to computer music [1]. SWARMUSIC, an artificial improviser, generates music by mapping the positions of swarming particles onto a space of music parameters. The particles are attracted to a target which moves in response to external events. This enables SWARMUSIC to interact with another improviser. The motion of the swarm itself is governed by its own internal dynamics. A novel feature of these dynamics (in a particle swarm context) is the presence of two opposing forces: attraction to the centre of mass and inter-particle repulsion (avoidance).

For the purposes of SWARMUSIC, the particle dynamics have to provide close tracking of a moving target. Previous work by Eberhart has discussed the use of particle swarms to optimise dynamic systems [2]. In his 'classic' particle swarm optimisation (PSO) algorithm, each particle is attracted by a linear spring-like force to a global best position and to an individual best position, where the best positions are determined by evaluation of the particle fitness.

For the models studied here, the concept of fitness is inappropriate. Rather, the particles swarm around a target and the shape of the swarm is translated into melody. The shape of the swarm is important for the coherence of the musical output. The particles must not swarm too closely (or else the melody is too repetitive) yet a definite shape must be maintained if any musical coherence is to emerge [3]. It is for this reason that a dynamics similar to Reynolds's boids simulation [4] was chosen, although the restriction on velocity matching is removed.

The next section describes the details of our particle dynamics and update algorithm. The experiments presented in section 3 form an initial exploration of the algorithm and its many adjustable parameters. The results of these experiments are analysed in section 4. Finally, section 5 draws some conclusions and suggests further work.

II SWARM SYSTEM OVERVIEW

The total acceleration a_i experienced by particle i (= 1,…N) at position x_i is shown in Box 1:

$$A_i = a_{i\ avoid} + a_{i\ centre} + a_{i\ target},$$

$$a_{i\ avoid} = 0, \qquad r_{ij} \geq p$$

$$a_{i\ avoid} = \sum_{j \neq i} \frac{C_{avoid}}{r_{ij}^{2}}, \qquad p_{core} < r_{ij} < p$$

$$= a_{core}, \qquad r_{ij} \leq p_{core}$$

where $r_{ij} = x_i - x_j,$

$$a_{i\ centre} = C_{centre}(x_{centre} - x_i)$$

$$a_{i\ target} = C_{target}(x_{target} - x_i).$$

BOX 1 PARTICLE ACCELERATION

Here, x_{centre} and x_{target} are the centres of mass of the swarm $\{x_i, v_i\}$, i = 1,…N and the target swarm $\{x^T{}_i\}$, i = 1,…M. The particles move in a space of dimension n. The three accelerations are parameterised by the constants $\{C_{avoid}$, p, p_{min}, a_{core}, C_{centre}, $C_{target}\}$.

The two attractive accelerations $a_{i\ centre}$ and $a_{i\ target}$ applied to each particle in the swarm are linear spring forces. These two terms are similar to the accelerations in the PSO algorithms – to make them identical replace x_{centre} and x_{target} by local and best positions. The avoidance acceleration $a_{i\ avoid}$ is zero for separations greater than p – this encourages the attractive accelerations. The particles experience an inverse square repulsion between a core radius p_{core} and a limit of perception p, and a constant 'core' acceleration at separations less than p_{core}. This core acceleration can be made equal to the acceleration at p by setting $a_{core} = C_{avoid} / p_{min}^2$, which ensures piecewise continuity at the core boundary. The particles can be made to experience a constant repulsion for all separations less than p by setting $p_{min} = p$. Note that $p_{min} > 0$ due to the singularity in the inverse square law.

The update parameters, UP, are constituted from the acceleration constants, a clamping velocity v_{max} and a target cube length x_{max}: UP = $\{C_{avoid}$, p_{core}, p, a_{core}, C_{centre}, C_{target}, v_{max}, $x_{max}\}$. The update algorithm is shown in Box 2.

There are two important features of this algorithm.

1. Particle positions are updated in turn, but the swarm centre is calculated from the positions of the old swarm. This is

similar to PSO algorithms. However, the avoidance acceleration is computed from the updated positions. This ensures that the update algorithm seeks to reduce particle avoidance at each *individual* particle update.

2. The updated velocity v_i is clamped to the range [0, v_{max}] *after* position update $x_i = x_i + v_i$. Velocity clamping prevents uncontrolled growth of v_i, but allows for changes in position $\Delta x_i > v_{max}$. Since all targets lie in cube T, particles that stray much beyond T will be pulled back in by the linear attraction a_{target} and this serves to prevent uncontrolled growth in x_i.

Choose dimensions {n, N, M}
Initialise swarm $\{x_i, v_i\}$
Place target swarm $\{x^T_i\}$ in cube $T = [0, x_{max}]^n$
Initialise UP
Loop
  if (*interact*) **capture** events
  update $\{x^T_i\}$
  Find x_{centre} , x_{target}
  for each particle
    $v_i = v_i + a_i$
    $x_i = x_i + v_i$
    if ($|v_i| > v_{max}$) $v_i = (v_{max} / |v_i|) v_i$
  endfor
  if (*play*) **interpret** swarm
until stopping criterion is met

BOX 2 UPDATE ALGORITHM

External interaction is included in this model through the possibility of adjusting the swarm in response to streamed audio events and is invoked by setting *interact* to true. The **capture** algorithm parses an input audio stream and adds targets, up to a maximum number. Then, as new events are parsed, the targets are re-positioned according to a target update algorithm. This means that the dynamics of the target swarm are determined by the interaction of the external musician. The **interpret** algorithm is responsible for mapping the particle positions to a space of MIDI parameters which are then sent to a MIDI synthesizer. The details of capture and interpretation, which are unnecessary for our purposes here, are explained further in [1] and [2].

III EXPERIMENTS

In these experiments (except for experiment 4), *interact* and *play* were both false, so that targets were not determined by outside events, and the musical output was ignored. A swarm of five particles was released in n = 3 dimensions from fixed starting positions and velocities, with a single target. The acceleration constants were determined as fractions of certain limit values, except for p_{core} which was set arbitrarily to 1.0. The target dimension x_{max} was set to 128.0. These choices for p_{core} and x_{max} were originally made for interpretative reasons [3]. The clamping velocity v_{max} was also chosen as a fraction of x_{max}. The limits to the acceleration constants were determined by the requirement that position updates should be on a scale commensurate with x_{max}. The relationships are set out in tables 1 and 2.

TABLE 1 LIMITS TO ACCELERATION CONSTANTS

$C_{avoid\ lim}$	p_{lim}	$a_{core\ lim}$	$C_{attr\ lim}$	$v_{max\ lim}$
$2x_{max}\ p_{core}^2$	$n^{1/2}x_{max}$	$2x_{max}$	$2\ n^{-1/2}$	x_{max}

TABLE 2 STANDARD VALUES FOR UPDATE PARAMETERS

C_{target}, C_{centre}	C_{avoid}	p_{core}	P	a_{core}	v_{max}	x_{max}
$0.5C_{attrlim}$	$0.5C_{avoidlim}$	1	$0.5p_{lim}$	$0.25a_{corelim}$	$0.25v_{maxlim}$	128

TABLE 3 STANDARD CONFIGURATION
OF FIVE PARTICLE SWARM

Particle	x_0	x_1	x_2
1	64.0	64.0	76.8
2	66.324936	71.15542	74.355415
3	54.151413	71.15542	67.955414
4	54.151413	56.84458	60.044582
5	66.324936	56.84458	53.644585
	v_1	v_2	v_3
1	0.0	0.0	8.000002
2	1.4530085	4.4721365	6.4721346
3	-6.155367	4.4721365	2.4721336
4	-6.155367	-4.4721365	-2.4721336
5	1.4530085	-4.4721365	-6.4721346

The starting positions and velocities were arbitrarily chosen to be on the surface of spheres of radius x_{max} / 10 centred on the centre of the target cube, and of radius $v_{max}/4$ centred on the origin, respectively. Table 3 lists these initial values.

In the experiments that follow, standard values and configurations were used throughout, unless specified to the contrary. The graphs show a plot of a single coordinate (interpreted as MIDI pitch value) for each particle as a function of time in update units (one unit = one iteration). Each dot in the graphs corresponds to a single particle and the continuous wavy line and continuous straight lines are plots of the pitch coordinate of the swarm centre and target centre respectively.

Experiment 1. A single target was set in the middle of the target cube, at (60, 60, 60). The first 300 updates are shown in Figure 1.

Experiment 2. In the previous experiment, the initial swarm centre is close to the target. Another important test of the particle swarm as a *search* algorithm is to change the target and observe the swarm's reaction. In the next three tests, the target jumps from $x_2 = 60$ to $x_2 = 30$ after 100 updates. The experiments differ in the setting of v_{max} (Figures 2-4).

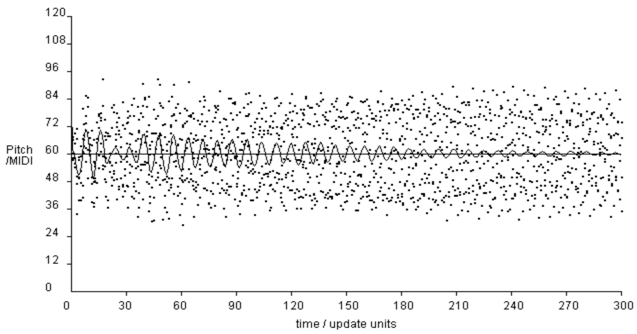

FIGURE 1 FIXED TARGET

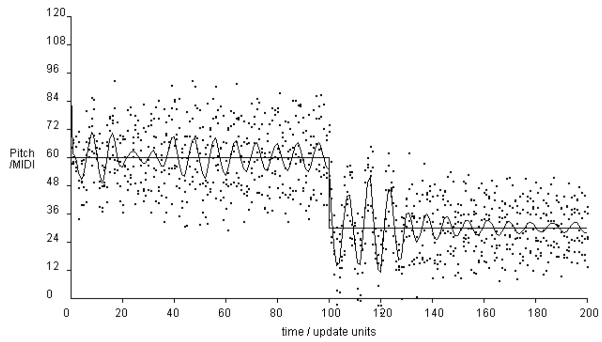

FIGURE 2 TARGET JUMP WITH $V_{MAX} = 0.25\ V_{MAX\ LIM}$

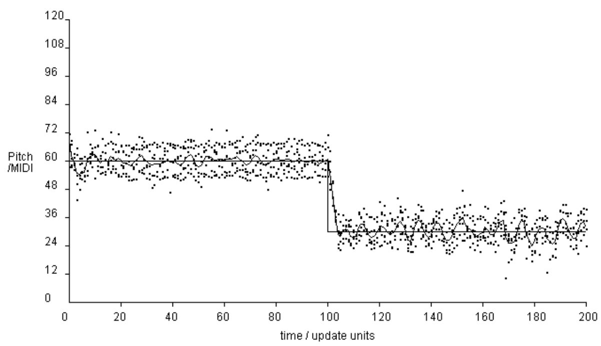

FIGURE 3 TARGET JUMP WITH $V_{MAX} = 0.05\ V_{MAX\ LIM}$

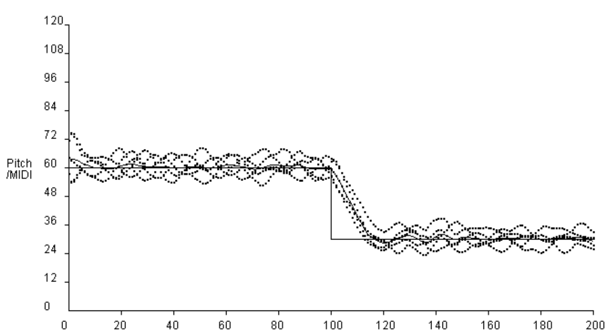

FIGURE 4 TARGET JUMP WITH $V_{MAX} = 0.01\ V_{MAX\ LIM}$

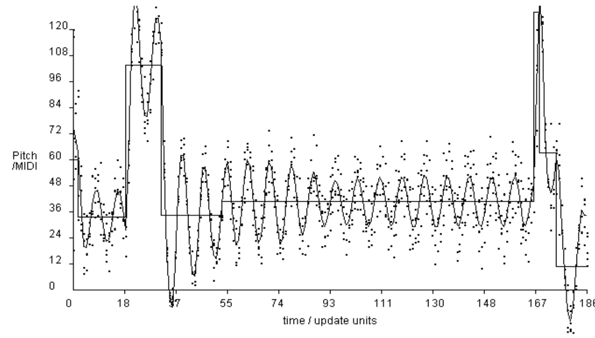

FIGURE 5 RANDOM TARGET JUMPS

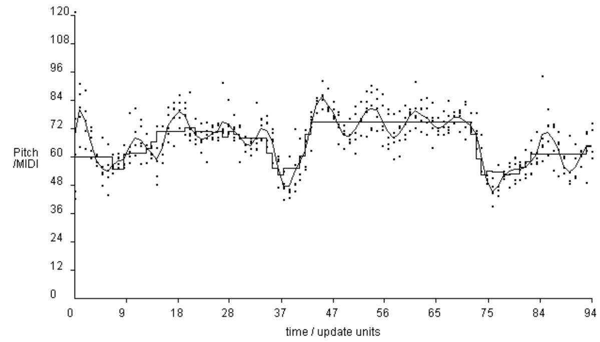

FIGURE 6 CAPTURING LIVE EVENTS

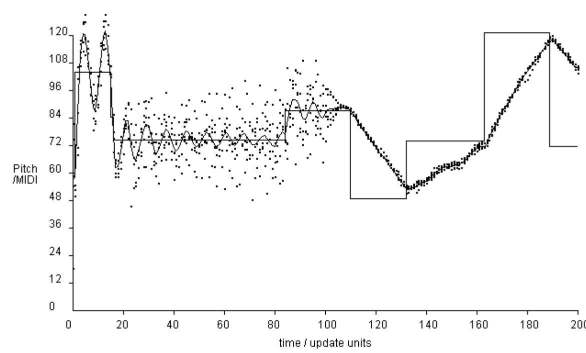

FIGURE 7 PARTICLE 'CLUMPING' WHEN V_{MAX} IS CHANGED TO 0.01 $V_{MAX\ LIM}$ AT 100 UPDATES

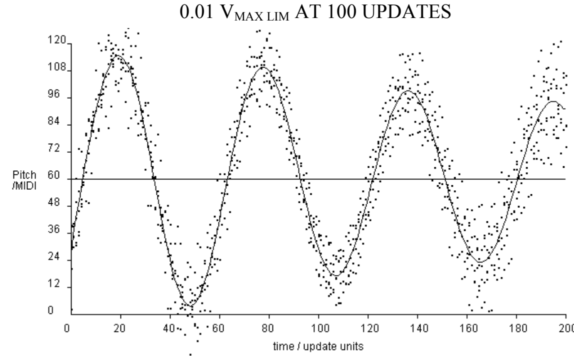

FIGURE 8 WEAK ATTRACTION TO TARGET

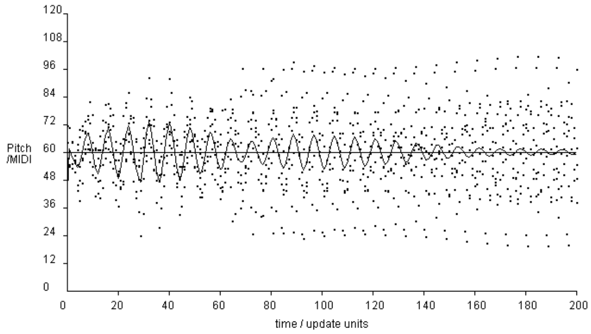

FIGURE 9 WEAK ATTRACTION TO SWARM CENTRE

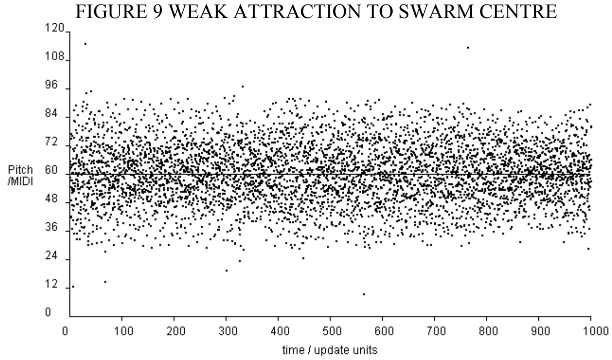

FIGURE 10 P = 0.02 PLIM - SWARM AND TARGET

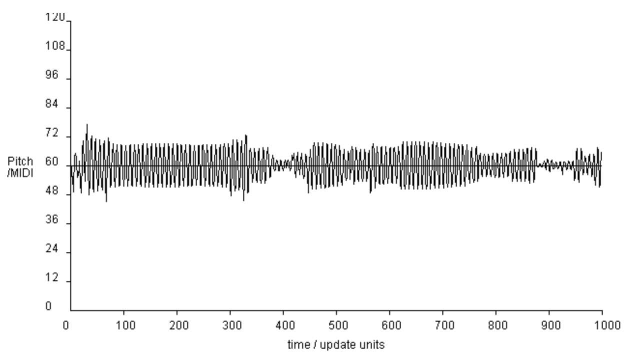

FIGURE 11 P = 0.02 PLIM – SWARM CENTRE AND TARGET

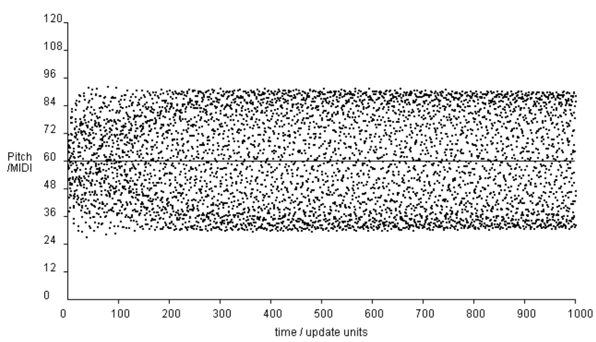

FIGURE 12 P = PLIM – SWARM AND TARGET

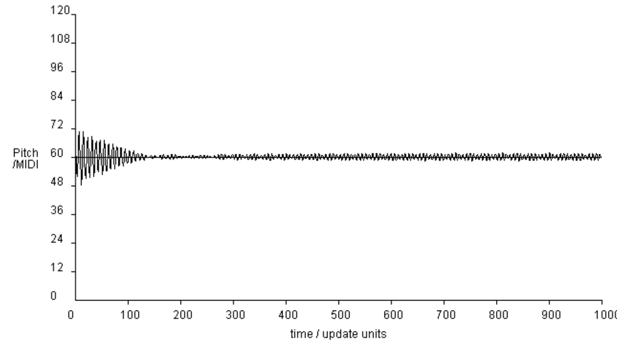

FIGURE 13 P = PLIM – SWARM CENTRE AND TARGET

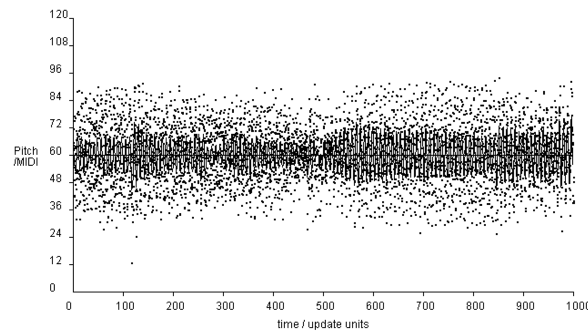

FIGURE 14 P = 0.02 PLIM - ALTERNATIVE ALGORITHM

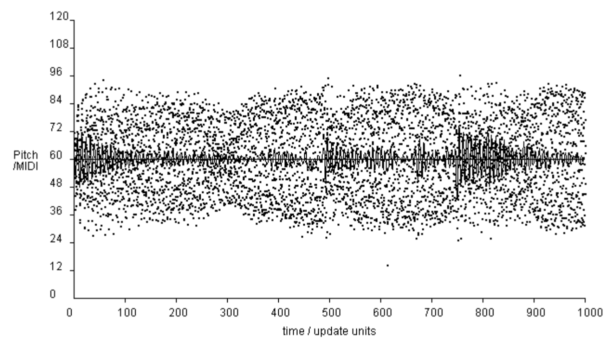

FIGURE 15 P = PLIM – ALTERNATIVE ALGORITHM

Experiment 3. In this experiment, the target changes to a new position when the swarm centre is within 10 units. Figure 5 shows the results of 186 iterations.

Experiment 4. In this run, SWARMUSIC was allowed to capture audio from a singer. The target swarm had five targets. The straight line in Figure 6 is the pitch component of the target centre.

Experiment 5. In this experiment, which starts with default update parameters and has random target jumps, C_{avoid} is set to zero and v_{max} is changed to 0.01 $v_{max\ lim}$ at 100 updates. The results are plotted in Figure 7.

Experiment 6 In order to experiment with a weak attraction to the target, the swarm was released with the target force constant, C_{target}, set to $0.01C_{target\ lim}$. Figure 8 shows the results with a fixed target and 200 updates.

Experiment 7 In this experiment, the swarm was released with only a small attraction to the swarm centre, $C_{centre} = 0.01\ C_{centre\ lim}$. Once more, the target is fixed and the results of the first 200 updates are shown in Figure 9.

Experiment 8. The radius of perception, p, is the update parameter which specifies the range over which each particle feels the repulsive inter-particle force. A particle will feel a repulsive force from all other particles lying within a sphere of radius p centred on itself. In the next two tests, the swarm starts in standard configuration, the target is fixed, and the force constant C_{avoid} is set to $C_{avoid\ lim}$. Firstly, a small value of p was chosen, $p = 0.02\ p_{lim} = 4.434$. Particles will not be repelled from each other unless they are within 4.434 units of each other. Figure 10 shows a plot of particle coordinate and target. For clarity, the swarm centre is plotted in Figure 11. The tests were then repeated with p set to p_{lim} ensuring that particles are aware of each other throughout the whole cube that represents music space. Figures 12 and 13 show the results.

Experiment 9. A feature of the update algorithm is that velocities are clamped after position update. The influence of this can be demonstrated by repeating Experiment 8 with an alternative algorithm (Box 3) more similar to the classic PSO approach where the position updates occur after velocity clamping.

for each particle
$v_i = v_i + a_I$
if $(|v_i| > v_{max})\ v_i = (v_{max} / |v_i|)\ v_i$
$x_i = x_i + v_i$

endfor

BOX 3 ALTERNATIVE ALGORITHM

The results are shown on Figures 14 and 15.

IV ANALYSIS

The experiment with a fixed target and standard update parameters (Figure 1) illustrates many of the important features of the update algorithm: the motion of the swarm centre is oscillatory and with diminishing amplitude. In contrast, the individual particles are moving within a strip of approximate width 26 on either side. At any time, the particles are evenly distributed around the target. The period, T, for simple harmonic motion is given by $T = 2\pi (m / C_{target})^{1/2}$. This gives a period of 8.3 for a spring system with constant 0.5 $C_{attr\ lim}$, in agreement with the period of the swarm centre, as shown in Figure 2 (37 oscillations in 300 units, $T \approx 8.1$. The motion of the swarm centre is very dominant when the target pull is weak (Figure 8) – the swarm as a whole moves to either side of the target with a period of 59.5, again agreeing with the theoretical period T = 59.3.

The kinetic energy of each particle is limited by v_{max}. This in turn sets a limit to the amplitude of vibration, which is evident in the strips seen in Figures 2, 3 and 4. The strip narrows as v_{max} is lowered. If the swarm centre is some way from the target, as seen in Figure 4, the swarm flies in formation when the target jumps, rather like a flock. This is because the target attraction dominates the both the other forces and the randomising effect of the small velocity. However, when the target is reached, swarming motion can begin again, but with small amplitude. However, the time taken to attain the target is longer when v_{max} is lowered. This illustrates an important feature of swarm algorithms. Each particle can be made to track the target very closely, but the swarm is then slow to respond to change. The particle 'clumping' of Figure 7 is a good illustration of this principle.

If a wide strip of movement is allowed, the response of the swarm to a changing target is very rapid, see Figures 5 and 6. Figure 6 is interesting since it is the only experiment to show the response of the swarm to an external source. In this case, the target centre changes often, but not by far, and this change is tracked well by the swarm. (Experiments 3 and 4 are also very interesting from a musical perspective [1].)

Another interesting feature of this algorithm is that although each particle may be oscillating with large amplitude, the swarm centre itself remains fixed at the target. Figures 12 and 13 illustrate this for a large number of iterations.

A small pull to the swarm centre means that the repulsive inter-particle force is more noticeable, and we might expect the particles to stray further from the target. This is indeed evident in Figure 9, where the particle strip is of width 44 (compared to 26 in Figure 2).

The results of the experiment with a small radius of perception, Figures 10 and 11, show a marked difference from the other figures. Particles are more often to be found near the target. In contrast, Figure 1, for example, shows particles evenly distributed within a corridor. Also, the plot of the swarm centre reveals more erratic motion, with less damping. The swarm is locked to the target but the particle motion is more chaotic.

In contrast, with p set to p_{lim}, (Figures 12 and 13) we see that the motion or the swarm is more ordered and with a very well defined particle strip. The swarm centre is damped down to a small amplitude after 120 updates. In contrast to Figure 10, Figure 12 shows a greater density of points plotted at the edges of the corridor. This is to be expected if the particles are undergoing regular oscillation about the target, since at the extremes of the motion they will be moving more slowly than at the centre.

The results suggest that long-range particle repulsion has a stabilising effect on particle motion. When the repulsion is only experienced at short-range, the particle motion is no longer restricted to a well defined strip, and particles spend more time close to the target. The motion of the swarm centre is not consistently damped, but fluctuates with large amplitude. The ability of the swarm centre to lock into a target is therefore seen to be enhanced by inter-particle repulsion.

Finally, the effects of velocity clamping before position update (Figures 14 and 15) leads to larger fluctuations in the motion of the swarm centre, an effect that is most pronounced for small p. It seems that the swarm centre is more focussed on the target if velocity clamping occurs *after* position update.

V CONCLUSIONS

The research described here shows that the most stable and ordered swarms are produced using dynamics where particles experience *collision avoiding forces* over a large range, and when velocity clamping occurs *after* position update. Compared to standard PSO approaches [2], this algorithm produces a swarm that moves in an extended region around the target. This region can be enlarged by reducing the attraction to the swarm centre. Low attractions to the target produce swarm oscillations to either side. The swarm with standard values of the update parameters responds quickly to a changing environment. When applied to the problem of music improvisation, all of these features lead to interesting and musical interpretations [1].

The quick response to target change is due to the large extent of the swarm about the target. Rather like a genetic algorithm with a diverse population, outlying particles can quickly respond if the target jumps into their immediate vicinity. On the other hand, the near coincidence of the swarm centre with the target suggests that such a swarm can track a moving target *accurately*. From this analysis, this work suggests that collision-avoiding swarms may have a useful application as a dynamic search tool for continuously changing problems. In this case, the target is replaced by the global and best positions found by the swarm, where 'best' is determined with respect to the global minimum of some function.

Conventional particle swarms have only been shown to be good at dynamic search if the global minimum jumps a small distance, i.e., low 'severity' [2]. Current investigations of collision-avoiding swarms for dynamic searches of high severity are generating impressive results [5].

References

[1] Blackwell T.M. and Bentley P.J. "*Improvised Music with Swarms*", submitted to CEC-2002

[2] Eberhart R.C. and Shi Y. "*Tracking and Optimising Systems with Particle Swarms*" Proceedings of the Congress on Evolutionary Computing, 2001

[3] Blackwell T. M. (2001) "*Making Music with Swarms*," MSc thesis, University College London.

[4] Reynolds C.W. "Flocks, herds and schools: a distributed behavioural model," Computer Graphics, Vol 21, pp 25-34, 1987

[5] Blackwell T.M. and Bentley P.J. "*Dynamic search with Collision-avoiding Swarms*", submitted to GECCO-2002

[6] Kennedy J. and Eberhart, R.C. "Particle Swarm Optimisation," Proceedings of the IEEE International Conference on Neural Networks, IV, pp1942-1948, 1995

[7] Kennedy J. and Eberhart R.C. "Swarm Intelligence," Morgan Freeman, 2001

[8] Bonabeau E., Dorigo M. and Theraulax, G. "*Swarm Intelligence: From Natural to Artificial Systems"* Oxford University Press, 1999