FLOcking Walkers (FLOW): Quick Start Guide

Leonidas Eleftheriou and Ath. Kehagias
leonidas.eleftheriou@gmail.com and kehagiat@gmail.com

September 19, 2016

1 Introduction

[*These instructions are for Windows users with Matlab 2009b or higher. Earlier Matlab versions should also work but have not been tested. On other platforms (Mac, Linux) the Matlab part should also work but have not been tested. The visualizer with DarkBasic will work only in Windows running DirectX 9.0c (October 2006) or later.*]

Unzip FLOW2D.zip in a folder of your choice, for example in C:\Matlab\FLOW2D. Start MATLAB, go to the folder C:\Matlab\FLOW2D\Simulator and type `NLW2D`. The GUI on figure 1 will appear. You are ready to go.

1.1 First Example

On the GUI that appeared replace the `Flock` rule as `u=Flock(10,100,2*pi,n)`, the `Flock` weight as 1 the `Global Rule Weight` as 0.3.

The expected results are shown in figures 2-4

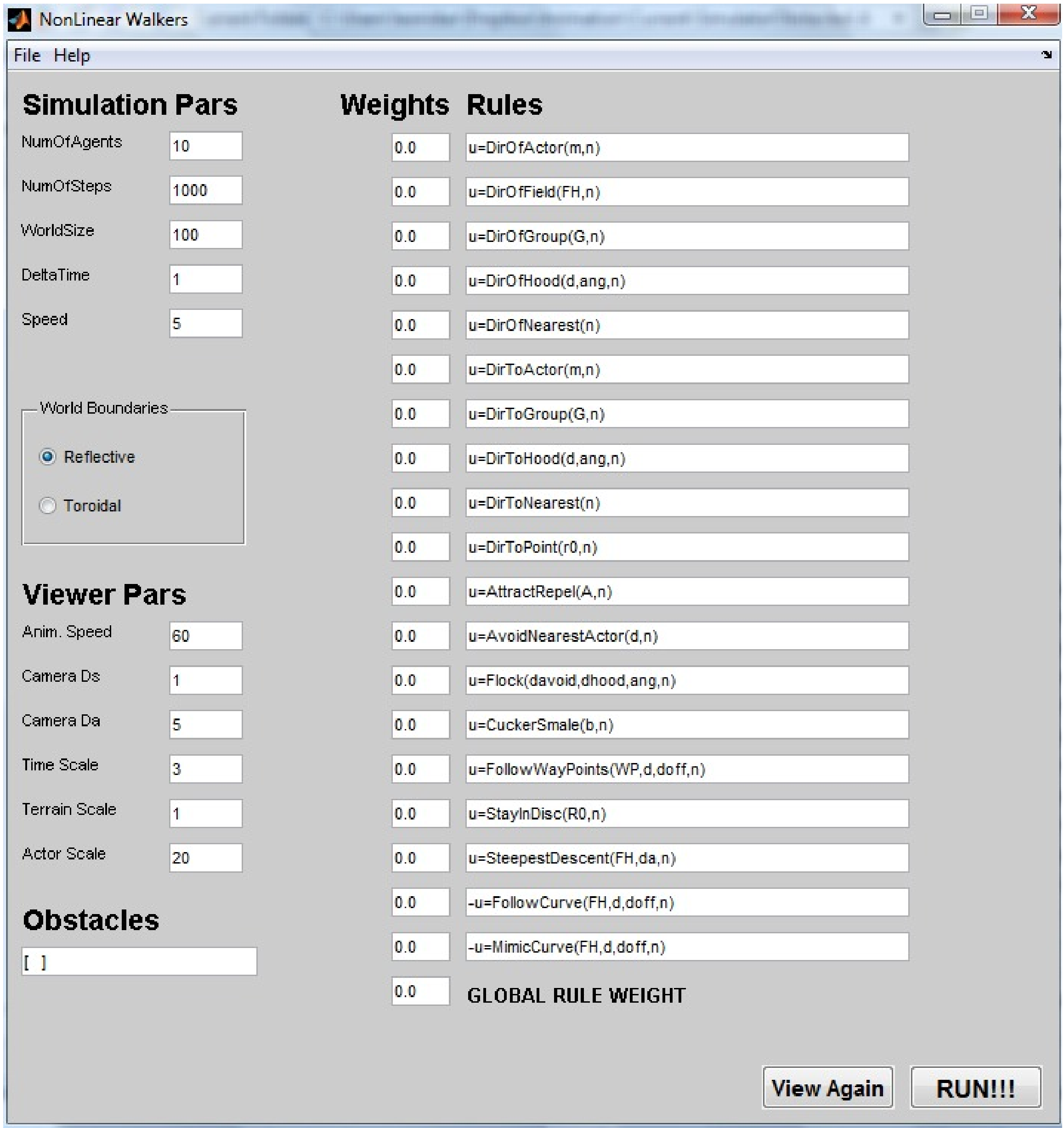

Figure 1: the MATLAB GUI

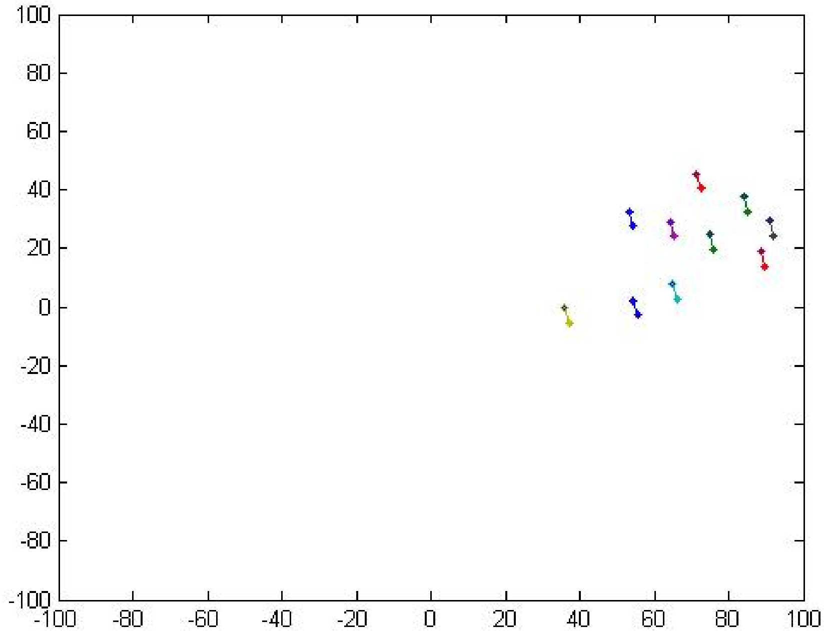

Figure 2: Dynamic trajectories of "*Boids*-like" simulation

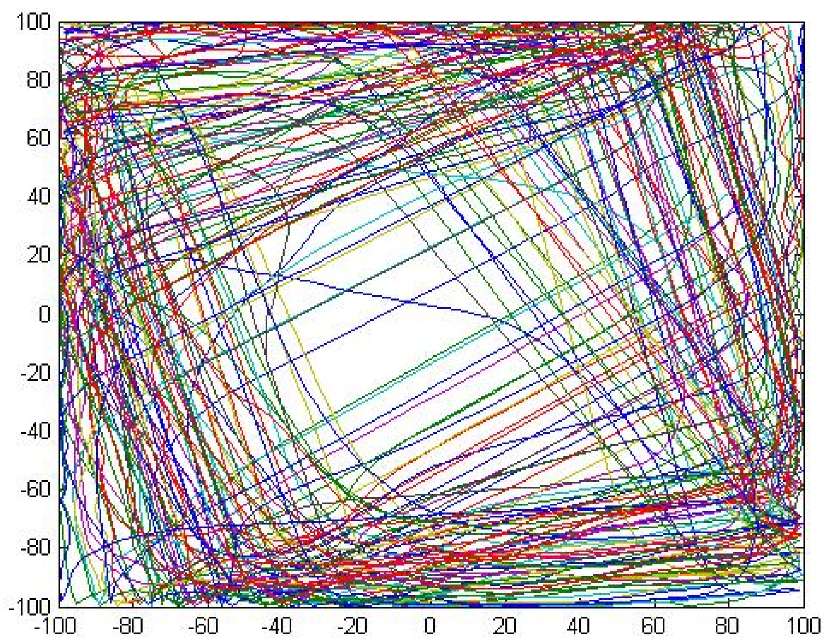

Figure 3: Static trajectories of "*Boids*-like" simulation

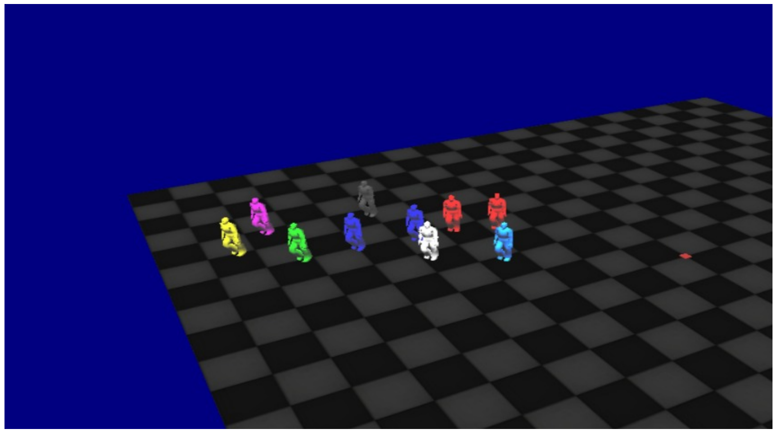

Figure 4: DarkBasic Visualization of "*Boids*-like" simulation

This experiment is intended to simulate 10 walkers that follow flocking behavior, with the neiborhood to be considered in a 100-unit radius and 2π angle and the minimum distance among the walkers in a 10-unit radius. The walkers also have a flexibility ability of 30%.

1.2 Second Example

Lets suppose that we want to make a few changes to the above experiment. We want only 10 walkers to keep moving as a flock, trying to stay in a disc of radius equal to 80 units. The world will also include two obstacles, centered at [50,50] and [-50 -50], both having a radius of 30 units. This means that the obstacles field should have the value [50 50 30;-50 -50 30]. All we have to do is to go to the already open GUI and apply the changes. To do so, replace the `obstacles` field as `[50 50 30;-50 -50 30]`, the `StayInDisc` rule as `u=StayInDisc(80,n)` and the `StayInDisc` weight as 0.5.

The result of the above procedure is shown on figures 5-7.

2 Extended Guidelines

In the GUI in general, you have to do the following steps:

1. Define the parameters of the experiment (simulation parameters). These are the number of agents, the number of the time steps of the experiment, the size of the world, the DeltaTime (dt) parameter, the speed of the agents and the kind of the world -reflective, toroidal-.

2. Define the parameters of the visualisation of the experiment (visualization parameters). These are the animation speed, the linear mobility of the camera, the angular mobility of the camera, the time scale, the terrain scale and the actor scale.

3. Define any desired obstacles. The obstacles should be an Nx3 matrix, where N stands for the number of the obstacles, the first and second column of the matrix stand for the centers of the obstacles and the third column for the radius of the obstacles.

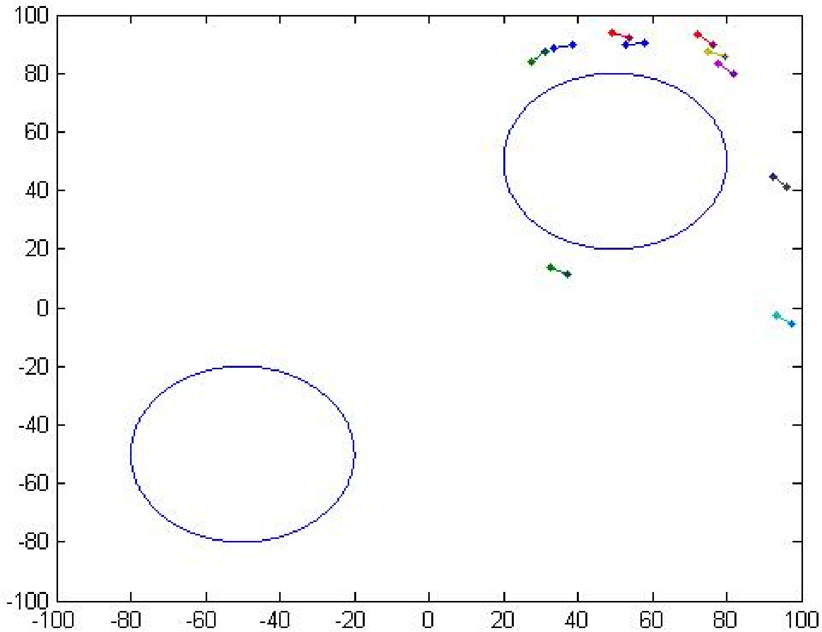

Figure 5: Dynamic trajectories of "*Boids*-like" simulation with obstacles

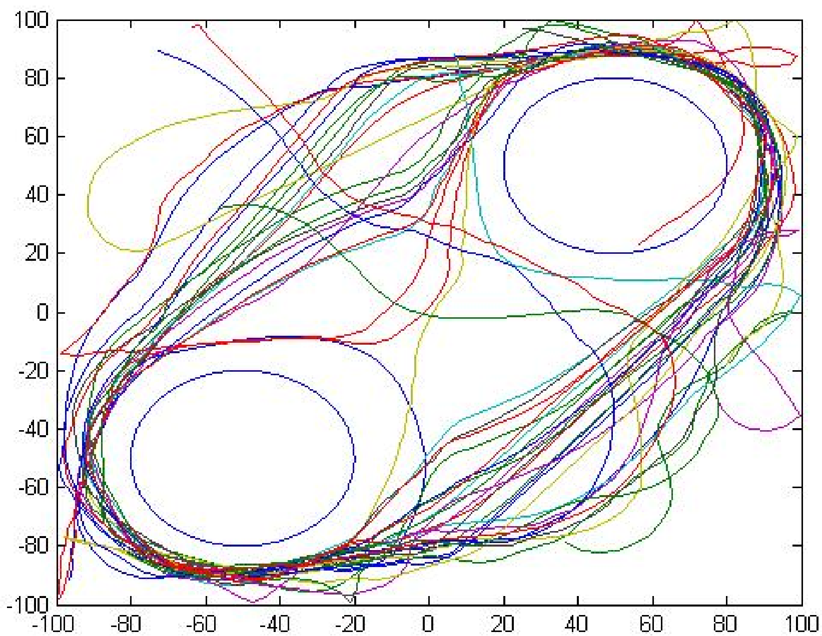

Figure 6: Static trajectories of "*Boids*-like" simulation with obstacles

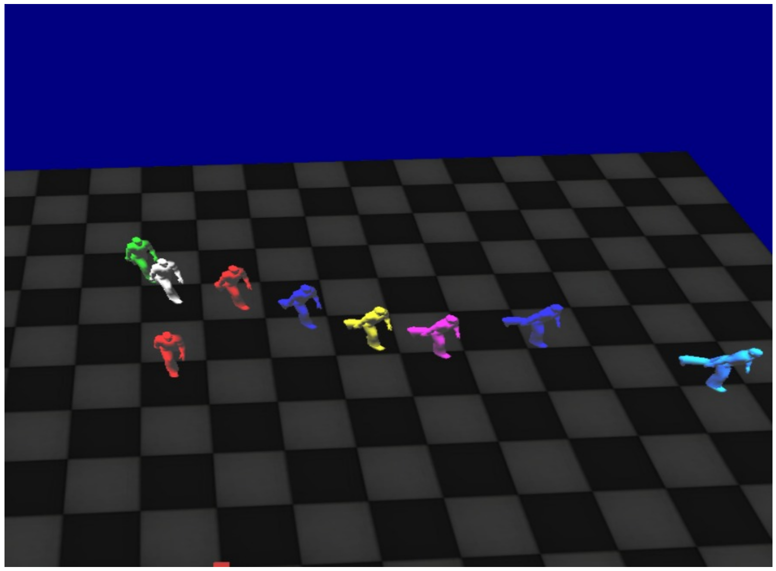

Figure 7: DarkBasic Visualization of "*Boids*-like" simulation with obstacles and motion inside a disc

4. Define the weights and the input parameters of the rules to be used. IMPORTANT: Do NOT change any variable with the name `n`.

5. Define the global rule weight, namely the flexibility factor of the walkers.

6. Hit the `RUN` button.

Optional: Define the name of the desired experiment file in the `File->Save` option at the menubar.

Here are some details about the above parameters:

Simulation parameters

1. Size of the world: The value of this field is to be saved as a variable named `Bnds`. The world's bounds will be $\pm Bnds$ along every dimension of the world.

2. DeltaTime parameter: This parameter when multiplied by the speed of the walkers produces the size of each step of the walkers, from one time step to the next.

3. The kind of the world: Reflective world means that the walkers reflect on a wall of the world when they try to overcome this wall. Toroidal world means that the walkers disappear from the side of the world that they initially were, and appear on the opposite side.

Visualization parameters

1. Animation speed: This parameter defines the speed of the animation at the DarkBasic visualization, namely how fast the walkers will move.

2. Linear mobility of the camera: The visualization consists of plenty of camera modes. The camera will move forward and backward - on the user's demand - with the speed defined by the linear mobility variable.

3. Angular mobility of the camera: Likewise, the angular momentum of the camera is defined by the angular mobility variable.

4. Time Scale: The walkers will move from a position to the next but not immediately, but through intermediate positions. This variable defines the number of those intermediate positions. This option exists so that the visualization can be smoother.

The expected output of the software consists of

1. The dynamic trajectories diagram, which means the positions and directions of the walkers at every time step.

2. The static trajectories diagram, which means the orbits of the walkers on their whole.

3. The visualization of the result, using the DarkBasic programming language.

Please note that in case you want to reset a rule, namely its parameters and its weight, all you have to do is to right clic on the rule text-box and press the reset button on the bottom right corner of the popup window. On this window you can find a short explanation of the rule and its arguments.

3 The 3D version

The 3-dimensional version of the software looks a lot like the above described one, except that it is not accompanied by the visualization with the DarkBasic, and that it does not support the obstacle avoidance option.

Unzip FLOW3D.zip in a folder of your choice, for example in C:\Matlab\FLOW3D. Start MATLAB, go to the folder C:\Matlab\FLOW3D\Simulator and type `NLW3D`. The GUI on figure 8 will appear. You are ready to go.

The user only specifies the simulation parameters, the rule weights and parameters, and the global rule weight.

3.1 Third Example

On the GUI that appeared replace the `Flock` rule as `u=Flock(10,100,n)`, the `Flock` weight as 1 the `Global Rule Weight` as 0.3.

The expected results are shown in figures 9 and 10

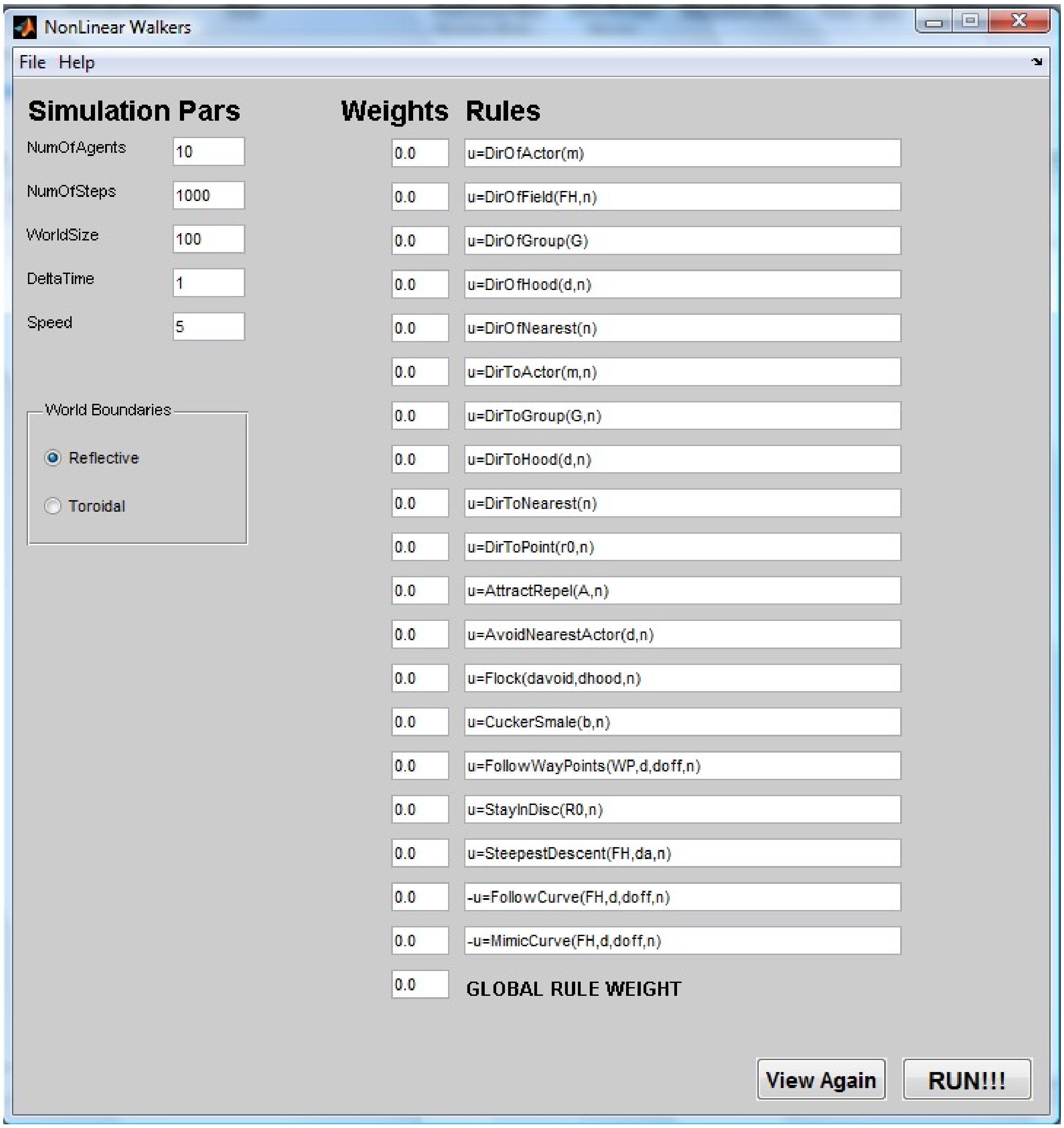

Figure 8: The GUI for the 3D version

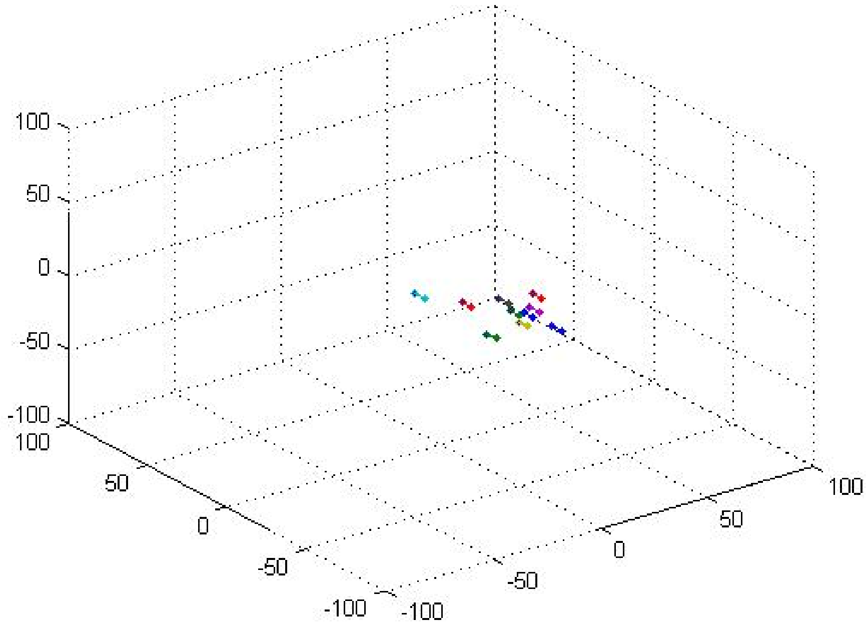

Figure 9: Dynamic trajectories of 3D "*Boids*-like" simulation

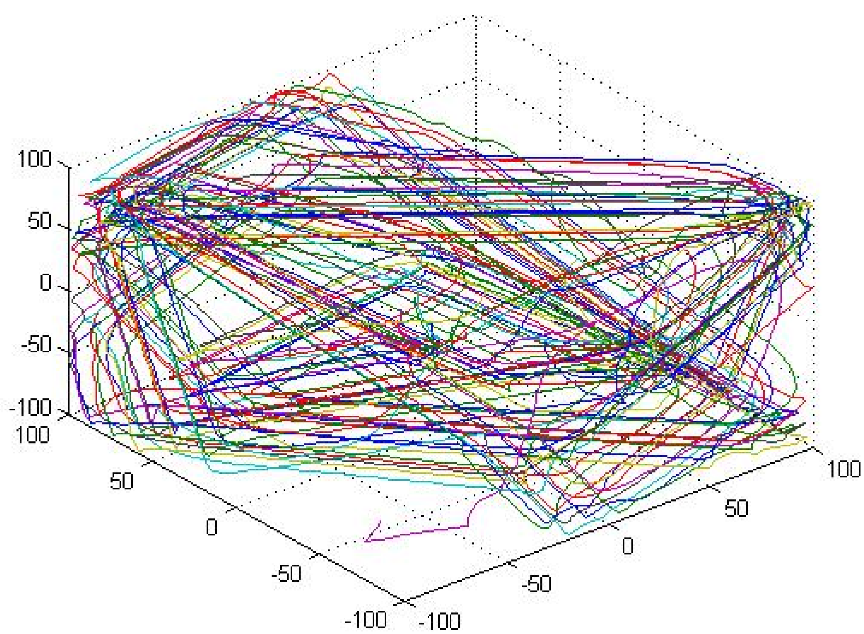

Figure 10: Static trajectories of 3D "*Boids*-like" simulation

This experiment is intended to simulate 10 walkers that follow flocking behavior, or the so called "*Boids*-like" behavior, with the neiborhood to be considered in a 100-unit radius and the minimum distance among the walkers in a 10-unit radius. The walkers also have a flexibility ability of 30%.