Simulating Boids Model by Scripting Blockset for Scilab/Xcos

[1] Aditya Raj, Institute fur Informatik, ITIS, Clausthal University of Technology, Clausthal 38678, Germany,
[2]Dr. Unmut Durak, TU Braunschweig,
Braunschweig, Germany

ABSTRACT

With this report, we introduce and investigate about the co-operation of agents specifically with Boids simulation in Scilab/Scicos. We start by defining an agent based system and then we look on a customized perspective of viewing agents and simulating them using Scilab. Here, we go back into the roots of Scilab and develop a customized Agent based modelling block for our purpose. The purpose of this report is to demonstrate the idea of simulating Boids; an Agent based model; in Scilab. We therefore introduce Scicos customized blocks to facilitate drag and drop feature for simulating Boids. These customized droppable boxes add an interesting tool box for mathematics people who do not care much into the inner details of how things at software layer work but rather want to use this feature as a part of the whole work process.

KEYWORDS

Agents, Boids, Scilab, Scicos

INTRODUCTION

In simple terms a model can be defined as an Agent Based Model if it simulates the actions and interactions of both individual or collective entities such as groups or organizations. And further we assess their effect on the system as a whole.

In 1986, Craig Reynolds (Craig, 1995) made a computer model of coordinated animal motion such as bird flocks and fish schools. Boids are birds like object. They can also thought
of as a simple agent based model where Boids play the role of Agents in this model. As defined by Grimm Volker (Grimm, et al.) Agent based models are also called Individual Based Models, and these agents are simpler than fully autonomous agents. Boids can also used to model fish and herding land animals.

Although there are many existing implementations of Boids model in many other programming languages, we have implemented this model in Scicos environment. Scicos is a Graphical User Interface(GUI) within Scilab to model, compile, and simulate dynamical systems as block diagrams. It has predefined commonly used model elements in Palettes of standard blocks. Apart from designing mathematical models and using the existing framework with Scilab, we can Modify existing open source blocks, Program new blocks in C, Fortran (dynamic link), or Scilab, extend current palettes, run simulations in batch mode from Scilab environment, Discover new Scicos capability using additional toolboxes and participate in many other development activities.

Although Scripting the codes directly in Scilab language, would be the first choice of a developer trying to develop/code an agent based model. However, this idea reduces the flexibility for the engineers more from mathematical background; who would like to prefer Simulink/Scicos over scripting everything by hand. A tool box featuring generalized ideas of Agent Based modelling would allow them to have a feel of testing agent based features. This is the main motivation behind demonstrating a customized Boids simulator tools.

Numerous applications areas exist for Boids implementation. It has been used for direct control and stabilization of teams of simple Unmanned Ground Vehicles (UGV) (Saska, et al., 2014). Also, It has been used for Micro Aerial Vehicles (MAV) in swarm robotics (Moere). Further we can use it for visualizing information (Cui, et al., 2009) and for optimization tasks.

We have come across many animated movies demonstrating flocking behavior. For example, in Computer generated bat swarms and armies of penguins marching through the streets of Gotham City in Breaking the Ice (1987), followed by a feature film debut in Tim Burton's film Batman Returns (Lebar Bajec, et al., 2009).

BACKGROUND

Complexity of Boids (Craig, 1995) arises from the interaction of individual agents (Boids in this case). As shown in the figure, a left out bird finds it way to its flock by following three simple rules name Alignment, Separation and Cohesion. These concepts are explained below:

1. Alignment:

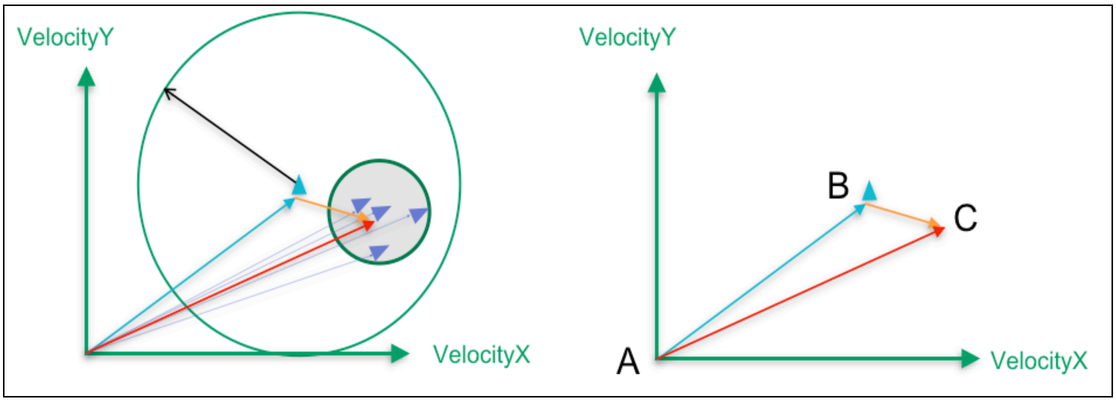


Figure 1 (a). Alignment of Boids. Black arrow shows flocking distance of the Boids, In Blue is the flock of Boids. (b). Computation of alignment vector

The flock of birds are represented (figure 1) here in Blue triangles. In order to join the flock, first the bird performs a search within the flocking radius (say around the visibility distance). Then the left out bird computes the alignment vector in the direction of the average heading of the flock. So, we average the velocity vector of the Boids and calculate the vector pointing towards the target using Vector law of Addition. This is shown in the Figure 1(b). By the law of Addition, we can say that AB+BC=AC.

Finally, we normalize this vector to get a unit vector in the direction of alignment. This gives us the first vector of our Boids model.

2. Cohesion

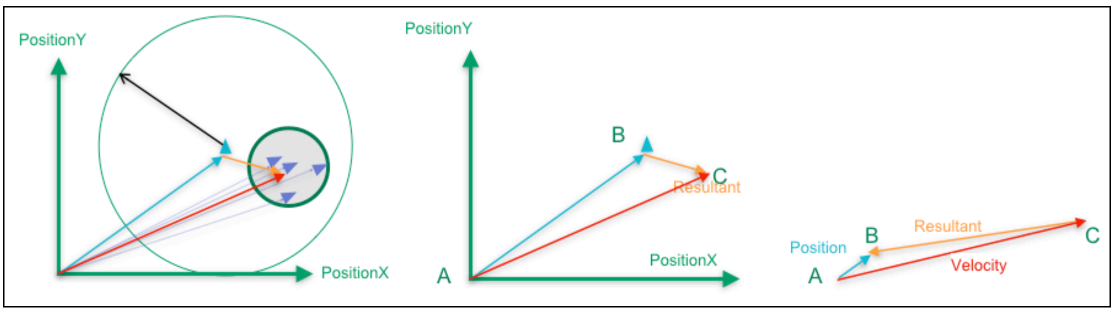


Figure 2 Boids showing Cohesion and Separation

Cohesion of a Boid is defined as steering the Boid to move toward the average position of local flock-mates. The process starts with finding neighbors in the flocking radius. Here, instead of doing computations with Velocity Vector, we take position vectors of Boids and further compute the average position of the Boids. This is because we need to deal with the positioning parameters of the flock-mates. The position vector can tell where the agents are located and then we can calculate the vector pointing from location to target. Next step is to normalize this vector so that we keep the unit vector in the direction of the cohesion. Finally, we need to steer the left out bird towards the flock and calculate the resultant.

3. Separation

A bird after entering the flock also makes sure that it does not collide with other agents(Boids). It first searches for the Boids in approximately one tenth distance of the flocking radius. Then it calculates the separation vector from each individual Boid. We normalize it and weight it by distance. This weighing is necessary to smoothen the motion when the Boid is near to collision. Further, we find the average of these computed differences.

The final step is to find the steer toward the flock. Steering velocity gives the Boid an additional velocity that steer the Boid away from the flock. As shown in figure 2(c), we can see that this Resultant Steer is difference of Velocity vector and Position Vector.

WORKFLOW

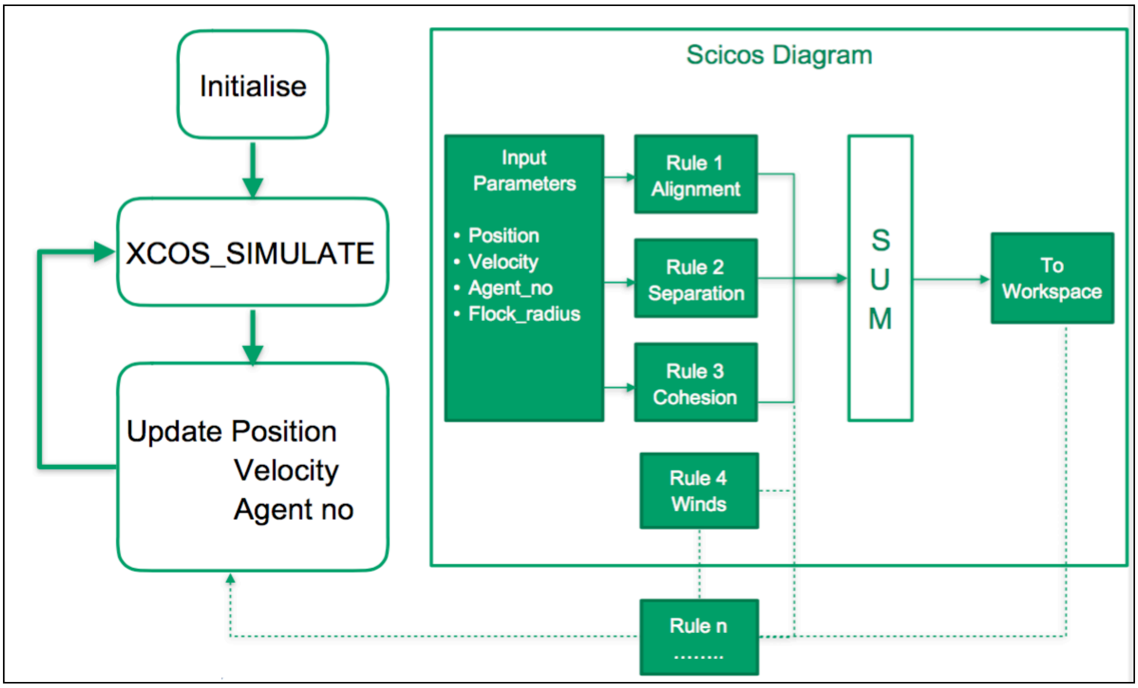
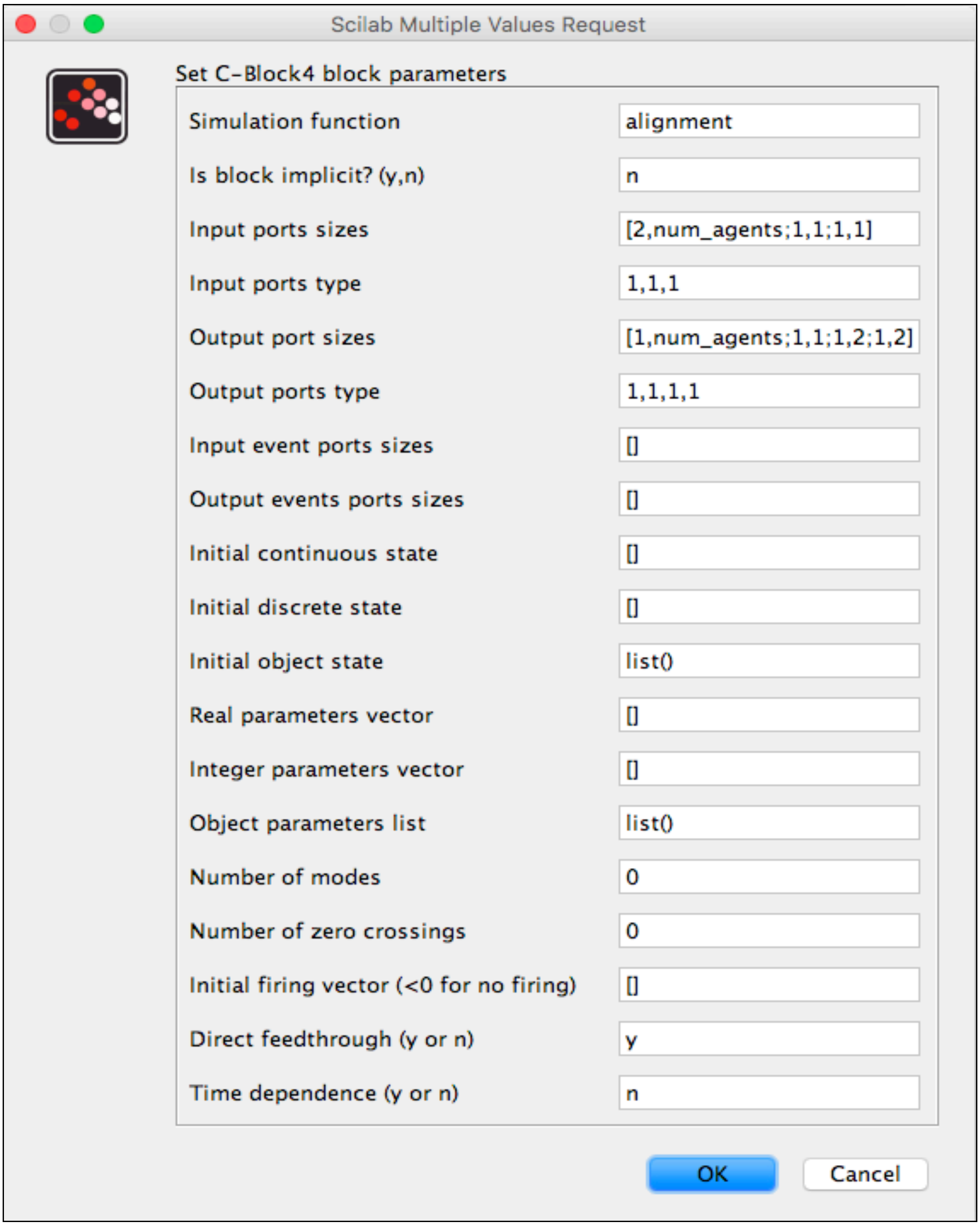


Figure 3 Boids Model In Scilab/Scicos

The workflow of the Scicos model is explained in the Figure (3). We can see the workflow by first learning how to write a customized function, in this case Alignment block, and then we can look at the whole model which uses these customized blocks.

We initialize the Scicos model in Scilab editor. And then these parameters are defined as constant blocks from Pallet Browser in Scicos editor. Further we pull out 'CBLOCK' [7] from user defined blocks from Pallet Browser and set the apt. parameters to make it an Alignment block. The parameters defined in this case is shown below:


Scilab Multiple Values Request

Set C-Block4 block parameters

Simulation function	alignment
Is block implicit? (y,n)	n
Input ports sizes	[2,num_agents;1,1;1,1]
Input ports type	1,1,1
Output port sizes	[1,num_agents;1,1;1,2;1,2]
Output ports type	1,1,1,1
Input event ports sizes	[]
Output events ports sizes	[]
Initial continuous state	[]
Initial discrete state	[]
Initial object state	list()
Real parameters vector	[]
Integer parameters vector	[]
Object parameters list	list()
Number of modes	0
Number of zero crossings	0
Initial firing vector (<0 for no firing)	[]
Direct feedthrough (y or n)	y
Time dependence (y or n)	n

OK Cancel

The important parameters here are explained below:

1. Simulation Function: Name of the simulation function (In this case alignment).

2. Input Port Sizes: We can specify the number of input ports we want to have in our CBLOCK. In our case we need 3 Input ports, Velocity vector of size[2, number of agents], Agent Number of size[1,1] and Flocking radius of size[1,1]

3. Input Port types: Should we 1 because we are dealing with real numbers here.

4. Output Port Sizes: I have defined here 4 Output ports. In reality we only need one output port but I have filled 3 ports with other values so that we can use several other features of the alignment block.

    - Port 1: Not in Use
    - Port 2: Number of agents found within the flocking radius.
    - Port 3: Sum of the positions of the agents found in the radius.
    - Port 4: Normalized vector in this direction of alignment

5. Output Port types: Should we 1 because we only can get real values here.

After setting these parameters, we are ready to grab these parameters (using the function GetRealInPortPtrs(block, port_number))inside CBLOCK and further code it to set the values at the out ports using the function GetRealOutPortPtrs(block, port_number). These methods return a pointer to the port values. So basically we are doing here pointer operations to set/unset the values.

Other methods that help determining the number of columns and rows of an input port are

- GetInPortRows(block, port_number);
- GetInPortCols(block, port_number);

These again return pointers.

The final step is to sum up these vectors using MATSUM block and save the value in a workspace. This is shown in the figure below:

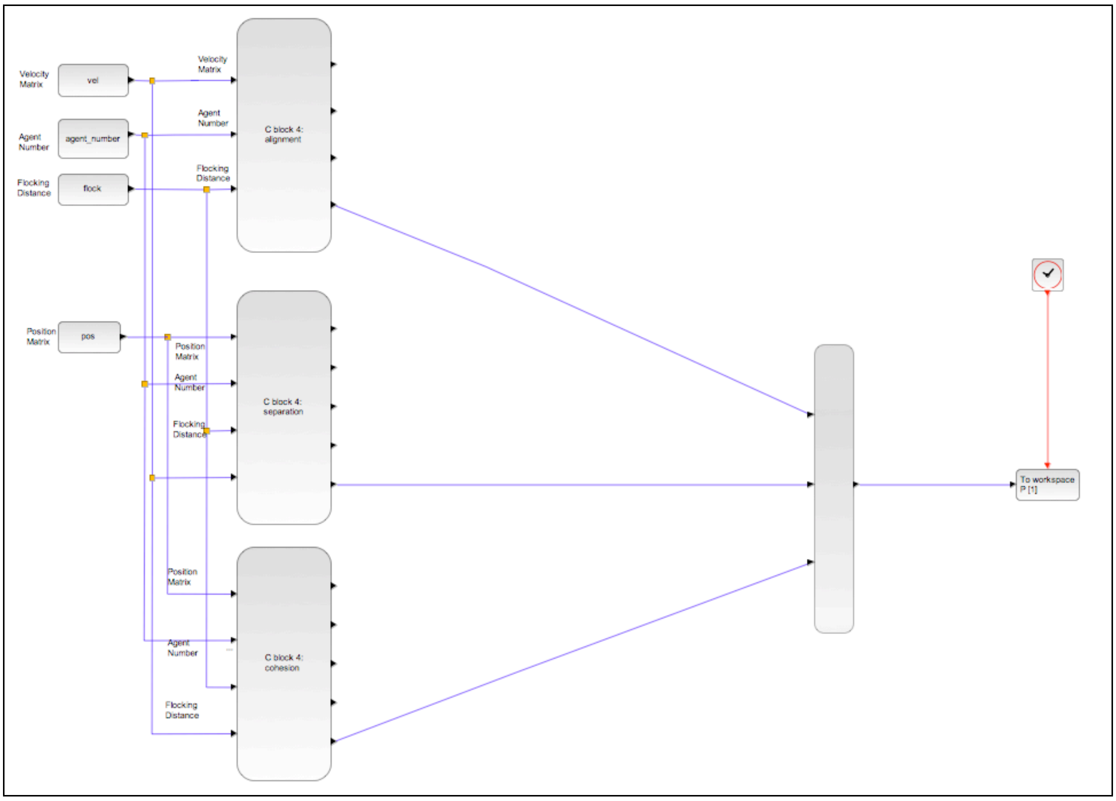
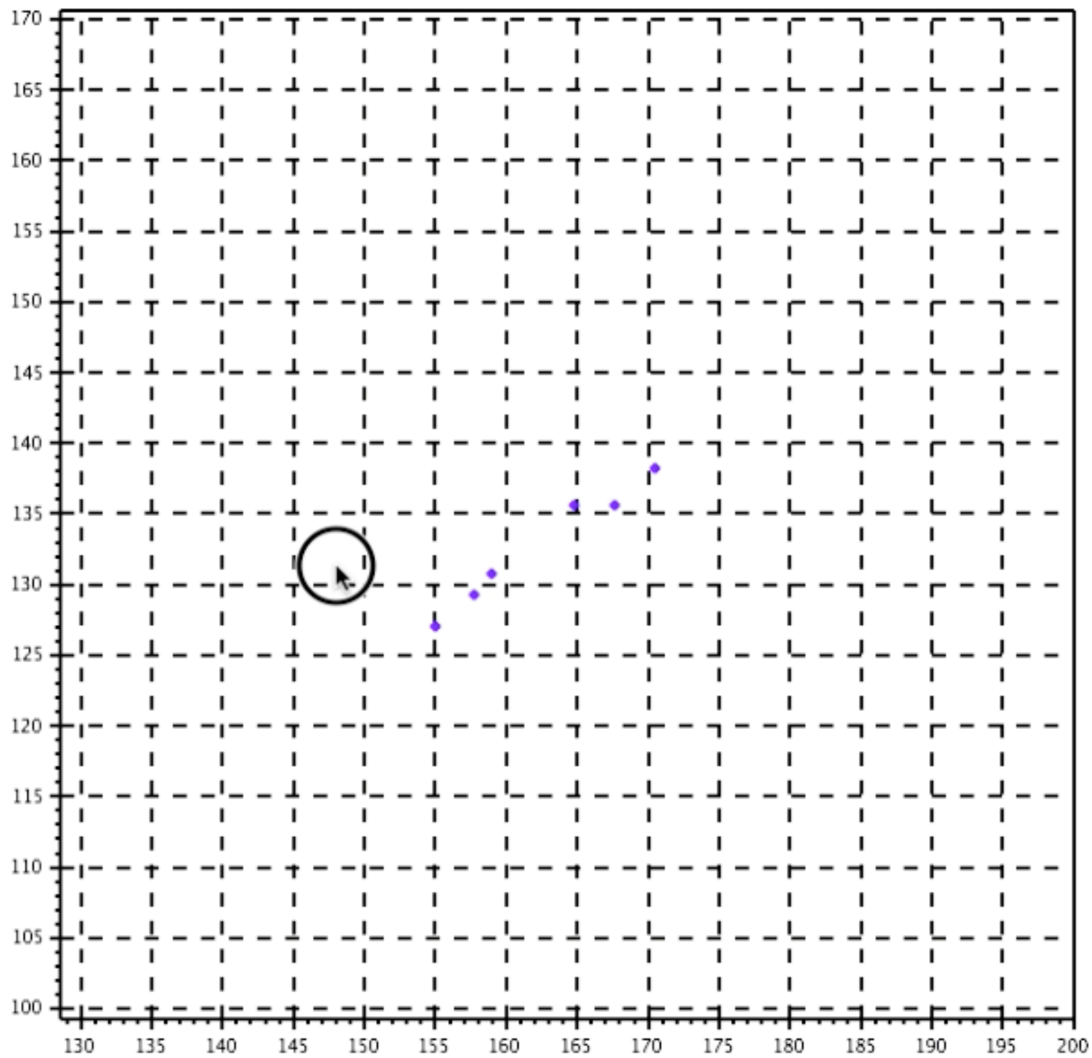


Figure 4 Scilab Scicos Boids Diagram

For Simulation purpose we assign random matrices to Position(2 by Number of agents), Velocity(2 by Number of agents), Agent Number(Current agent number for which we want to calculate flocking parameters), Flocking radius(Search radius of an agent). This initialization of the constants is done inside a Scilab editor. The workflow in Scilab editor is shown in figure (3). We initialize it, import the diagram, call XCOS_SIMULATE to process the simulation in batches and further update the values received from workspace. This goes on in a loop where we increment the agent number in each iteration and final we receive the updated positions of all the Boids. After having all the values updated, we plot the positions of Boids all at once in a 2D plot.

RESULTS

Simulation Results were conforming to the predicted Boids behavior.



DISCUSSION

This paper provides an interesting way to model different behaviors of agents using Scicos blocks. The most important part of this demonstration was to design a working model for simulating agent based behavior using customized Scilab blocks. And further use them in such a way that they can generate a working solution with minimal coding effort for end users. The only thing user has to do is initialize the parameters and rest the Scicos customized block takes care of.

CONCLUSION AND IMPLICATIONS

The Boids simulation will serve as a great tutorial for building up new tools for Scilab toolbox. However, there can be many improvements suggested in this design Firstly we can replace for loop called to calculate position of each Boid with Streams. These streams can calculate several positions of agents at once thus reducing the time complexity. Further we can use existing frameworks and thus ensuring reusability. This also will fasten up the code. So for further designers and researchers, this tutorial can be a great starting point for following the footsteps but the coding should be adapted to the newest versions of Scilab with new features.

Do not simply repeat results or discussion, but provide some overall comments on the findings and their applicability in other settings or applications. The discussion of implications should tell the reader what the importance of the work is for others including researchers, building designers, owners and operators, or occupants.

ACKNOWLEDGEMENTS

I thank Prof. Dr. Hartmann and Dr. Unmut Durak for assisting me in completion of this project.

REFERENCES

Grimm Volker and Railsback Steven F. (2005) Individual-based Modeling and Ecology. [Book]. - [s.l.] : Princeton University Press. - p. 485. - ISBN 978-0-691-09666-7..

Craig Reynolds [Online] // Boids . - 6 29, 1995. - http://www.red3d.com/cwr/boids/.

Min Hongkyu and Wang Zhidong Design and analysis of Group Escape Behavior for distributed autonomous mobile robots. [Conference] // IEEE International Conference on Robotics and Automation (ICRA).. - 2011.

Saska Martin, Jan Vakula and Libor Preucil Swarms of micro aerial vehicles stabilized under a visual relative localization. [Conference] // IEEE International Conference on Robotics and Automation (ICRA).. - 2014.

Moere A V (2004) Time-Varying Data Visualization Using Information Flocking Boids [Journal]. - [s.l.] : Proceedings of the IEEE Symposium on Information Visualization. - pp. 97–104. doi:10.1109/INFVIS.2004.65..

Cui Zhihua and Shi Zhongzhi "Boid particle swarm optimisation" [Journal] // International Journal of Innovative Computing and Applications. - 2009. - 77–85. doi:10.1504/IJICA.2009.031778..

Enterprises Scilab Scilab: Open source software for numerical computation [Online]. - http://www.scilab.org/resources/documentation.

Lebar Bajec Iztok and Heppner Frank H. "Organized flight in birds" [Journal]. - 2009. - Animal Behaviour. pp. 777–789. doi:10.1016/j.anbehav.2009.07.007..