# SELF-ORGANIZATION IN VOWEL SYSTEMS

## a simulation

compiled by:

ROHIT GUPTA

SIDDHARTH JAIN

# INTRODUCTION

Artificial life, also known as alife or a-life, is the study of life through the use of human-made analogs of living systems. Computer scientist Christopher Langton coined the term in the late 1980s when he held the first "International Conference on the Synthesis and Simulation of Living Systems" (otherwise known as Artificial Life) at the Los Alamos National Laboratory in 1987. His official conference announcement was the earliest description of a field which had previously barely existed.

Since that day, a lot of philosophical, psychological and computer-scientific research work has been directed towards this field. The major topics of these works have been

**Origin of life, self-organization, self-replication**

- Astrobiology, Exobiology
- Artificial and Evolutionary Chemistry
- Autocatalytic Systems, Metabolization
- Synthetic and Supermolecular Chemistry
- Molecular Information Processing, Construction and Nanotechnology

**Development and differentiation**

- Multi-cellular Development
- Gene-regulation Networks
- Natural and Artificial Morphogenesis

**Evolutionary and adaptive dynamics**

- Pattern and mode of evolution, Radiation and Extinction
- Modes of selection (natural, sexual, neutral, kin, etc.)
- Artificial evolutionary ecologies and Life games
- Molecular evolution in models systems
- Immune and Defense systems
- Evolvability and its impact on biological organization
- Cultural evolution, Evolution and Learning
- Evolutionary computation

### Robots and agents

- Bio-inspired robots, Embodied Cognition
- Autonomous and Adaptive Robots and Software agents
- Evolutionary robotics

### Communication, cooperation and collective behavior

- Emergent collective behaviors, Swarm intelligence
- Evolution of communication and cooperation
- Social and Linguistic systems
- Economic systems, Social-Technical systems

### Applications of ALife technologies

- Industrial and Commercial applications
    - Evolvable hardware, Self-repairing hardware and Molecular Computing
    - Genetic Engineering and Nanotechnology
    - Finance and Economics
    - Computer Games
- Medical Applications
- Educational applications

### Simulation and synthesis tools and methodologies

- Formal and Mathematical foundations
- Clarification and Evaluation of ALife methodologies
- Simulation languages, Experimental tools
- Artificial worlds
- Tools for large data sets

In Evolutionary Linguistics, the major aim is to explore and propose theories regarding the evolution and acquisition of language in humans and other animals and attempt to simulate the same. Several papers have been published in the previous years investigating ways in which artificial agents can self-organize languages with natural-language like properties and how meaning can co-evolve with language. The topic studied in this project is a subset of evolutionary linguistics, namely Vowel Systems.

# PREVIOUS WORKS

Glotin, in 1995, published one of the first works related to an exploration into vowel systems found in human languages. He wrote three papers in the following years at the *Institut de la Communication Parlee* which also inspired further research in the field.

Glotin uses a population of agents called *carls,* to demonstrate the organization which his proposed system can achieve. Every agent has a fixed set of vowels which are represented in an acoustic as well as an articulatory space. The game between the two agents proceeds as follows: An agent 'A' picks up a random vowel from its repertoire and then transmits it acoustically to another agent 'B'. B then finds a closest vowel in its repertoire to the one perceived. It then shifts its vowel to the perceived position by means of making articulatory changes. It then produces a new signal and the agent 'A' updates its vowel repertoire accordingly. Also other vowels that are in the repertoire are moved from the vowel being used. A fitness factor is calculated which is inversely proportional to the cost involved in moving. Only those agents are allowed to produce offspring who are fit enough. The initial vowel repertoire of the offspring is inherited from its parent.

There were two principal drawbacks in this work:

- The mapping from acoustic differences to articulatory changes is computationally very demanding and hence Glotin was unable to work on a large population and greater number of vowels.
- A genetic element is added to the population dynamics. The offspring inherits the vowel repertoire from its parents therefore it is not clear whether the organization of vowels is due to the genetic factor or due to optimization or self-organization.

Berrah, in 1998, elaborated on Glotin's work. He used only acoustic representations hence his system was much faster. He concentrated mainly on the repulsion between the vowels prototypes in the system. The interactions between the agents do not play an important role in determining the system. However, according to DeBoer, in this simulation, the role of agent interaction is obscured by the Darwinian selection.

Bart DeBoer, of the University of Brussels, Belgium is the author of another paper on the self-organization in vowel systems, which claims to remove the above-mentioned drawbacks in the previous works. He did not include the Darwinian concepts considering it unimportant in case of organization of vowels. He also considered the energy optimization for the vowel systems. Our simulation is an extension of DeBoer's work. We deviate from his view of allowing no evolutionary concepts (like age and health) and attempt to explore the effects of introducing such effects.

# THE SIMULATION

The Simulation we have made is very similar to DeBoer's simulation.

**Agent Architecture:**

Every agent is equipped with a vowel space which is a list of all the vowels which the agent has learned throughout its life (the number of games it has played). The vowels are described in two ways: using the acoustic and the articulatory parameters. The acoustic parameters are a function of the articulatory parameters. A vowel is described articulatorily by three parameters: p, h and r. p denotes the position of the tongue when the vowel is spoken. h is the height of the tongue and r is the lip rounding. The values of these parameters lie between 0 and 1. For p, 0 means most to the front and 1 means most to the back. In case of h, 0 means lowest and 1 means highest. And, for r, 0 means least rounded and 1 means most rounded.
The parameters assume discrete values with a default increment of 0.1 (this parameter is an input parameter of the simulation and can be changed).

The relation which DeBoer used to map the articulatory parameters to the acoustic parameters has been incorporated in our simulation without any change. This mapping is as follows:

$$
\begin{aligned}
F_1 = {} & ((-392 + 392r)h^2 + (596 - 668r)h + (-146 + 166r))p^2 \\
& + ((348 - 348r)h^2 + (-494 + 606r)h + (141 - 175r))p \\
& + ((340 - 72r)h^2 + (-796 + 108r)h + (708 - 38r))
\end{aligned}
$$

$$
\begin{aligned}
F_2 = {} & ((-1200 + 1208r)h^2 + (1320 - 1328r)h + (118 - 158r))p^2 \\
& + ((1864 - 1488r)h^2 + (-2644 + 1510r)h + (-561 + 221r))p \\
& + ((-670 + 490r)h^2 + (1355 - 697r)h + (1517 - 117r))
\end{aligned}
$$

$$
\begin{aligned}
F_3 = {} & ((604 - 604r)h^2 + (1038 - 1178r)h + (246 + 566r))p^2 \\
& + ((-1150 + 1262r)h^2 + (-1443 + 1313r)h + (-317 - 483r)) \\
& + ((1130 - 836r)h^2(-315 + 44r)h + (2427 - 127r))
\end{aligned}
$$

$$
\begin{aligned}
F_4 = {} & ((-1120 + 16r)h^2 + (1696 - 180r)h + (500 + 522r))p^2 \\
& + ((-140 + 240r)h^2 + (-578 + 214r)h + (-692 - 419r))p \\
& + ((1480 - 602r)h^2 + (-1220 + 289r)h + (3678 - 178r))
\end{aligned}
$$

Here, $F_1$, $F_2$, $F_3$, and $F_4$ are the formant frequencies (acoustic parameters).

An age is associated with each agent which represents the ‘maturity’ of that agent. An agent which has played greater number of games is exposed to the vowel sets of more agents and can be, hence, considered to be more mature.

## The Game:

From the population, two agents are chosen to play a game. One of them is the initiator and the other is the imitator. The initiator chooses a random vowel from its repertoire to speak. If its vowel space is empty then a random vowel is added to its list. The environment adds a random level of noise to the spoken vowel (which lies in a range of maximum allowed noise which is an input parameter ‘a’ with a default value of 0.1). The imitator, then, proceeds to search its list for a vowel which is articulatorily closest to the perceived signal. On finding this vowel it emits the acoustic signal corresponding to it for the initiator to listen. In case of a void vowel space a random vowel is added to the list which is shifted as close as possible to the perceived the signal. The environment, yet again, adds noise to the signal being transmitted. On hearing this signal, the initiator repeats the process of searching for the closest vowel. If this vowel is the same as the vowel which was spoken at the onset of the game, there is success. The feedback about success or failure is (non-verbally) communicated to the imitator. In real life, this could happen by facial expressions (for example). Based on the received feedback, the imitator updates its vowel repertoire according to the same rules followed in DeBoer's paper.

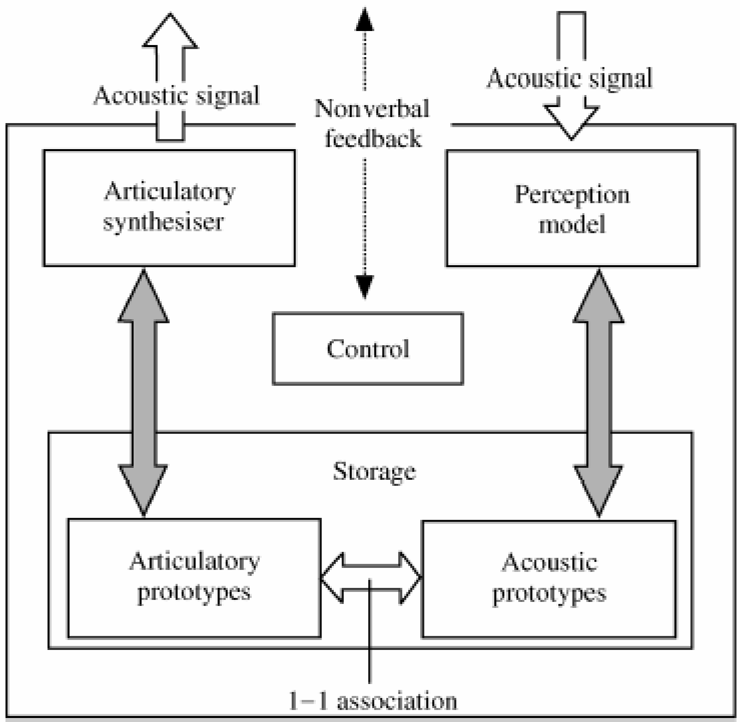



*Figure 1: Agent Architecture*

As shown in Figure 1, the Articulatory synthesizer uses the above equations to generate the acoustic signal for a particular vowel from its repertoire. The perceptive model captures the acoustic signal as heard from another agent and finds the vowel in its list with the closest acoustic prototypes.

*Selection of Imitator and Initiator:* Unlike DeBoer's simulation, where the initiator and imitator are selected randomly, we assign a probability distribution over the population to do the selection. A number 'Prob' is associated with every agent which signifies its probability to be chosen as the initiator. Implicatively, the probability for an agent to be chosen as the imitator is (1-Prob). The value of Prob is calculated as follows:

Prob = (Age of the Agent) / (Sum of ages of all agents in the population)

Thus, an agent which is 'older' is preferentially selected as the initiator and a 'young' agent is preferentially selected as the imitator. This is similar to the real world, in that a mature person teaches language to the younger ones by talking to them.

*Adding Noise*: If the noise parameter is 'a'. Then a random number 'x' is generated which lies between –a and a. Every formant of the acoustic signal is, then, altered thus: $F_i = F_i * (1 + x)$

## **Input Parameters to the simulation**:

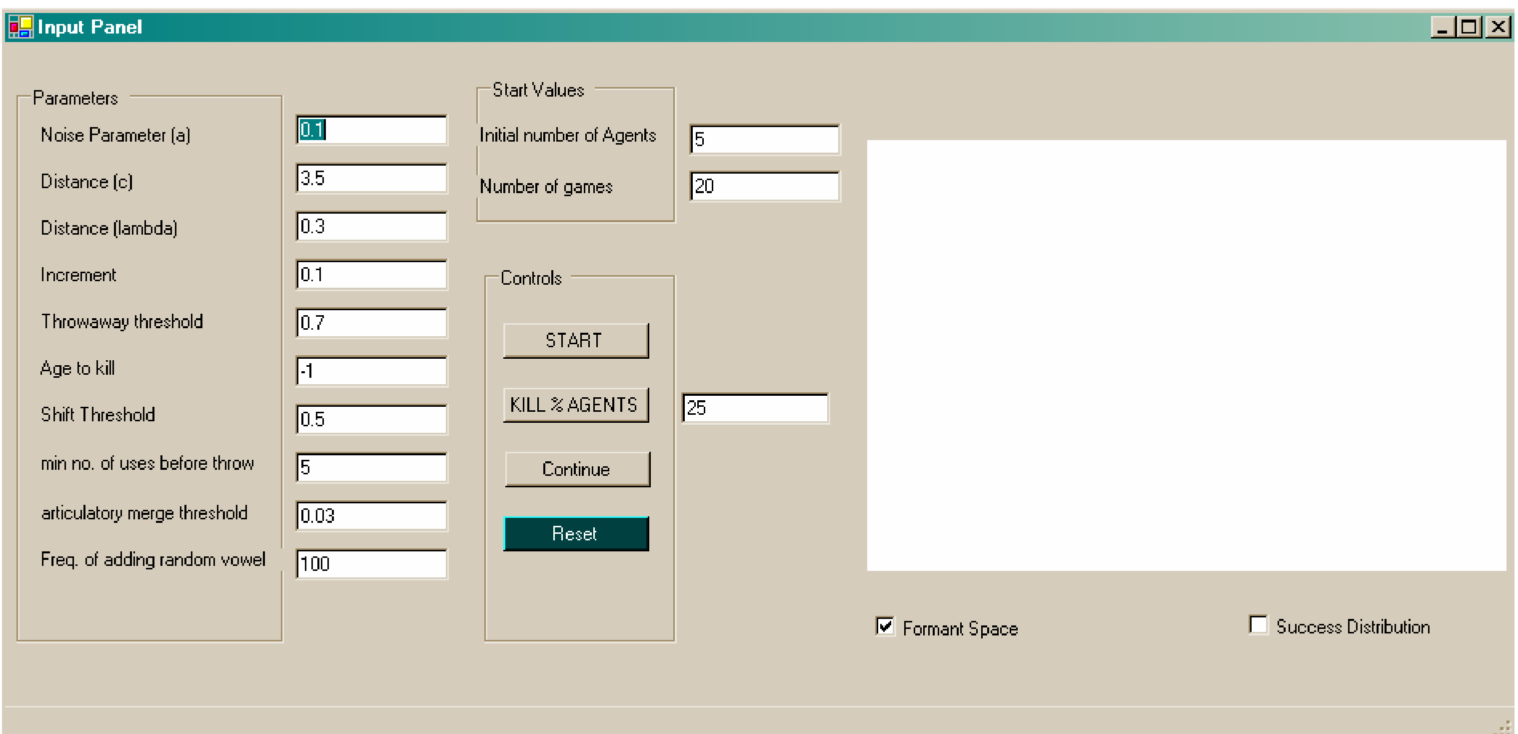



*Figure 2: GUI of the Simulator. Input Parameters and their default values.*

Figure 2 shows a snap-shot of the GUI of the simulation. There are 10 input parameters listed on the left side.

- Noise Parameter (a): This is maximum fraction of noise that is allowed in the formants produced.
- Distance (c): The calculation of the effective second formant frequency is based on a critical distance, c (measured in Barks).
- Distance (λ): λ determines the relative weight of the second formant (F2) with respect to the first frequency (F1). It is used to calculate the 'distance' between two acoustic signals, which plays an important role in the simulation. The distance is calculated as follows: $D = \sqrt{(F_1^a - F_1^b)^2 + \lambda(F_2^{a'} - F_2^{b'})^2}$
- Increment: This parameter defines the size of the discreet steps for the values of p, h and r.
- Throwaway Threshold: After every game is over, there is an update in the vowel space of both the participating agents. One of these updates is to remove the vowels with low success ratio (bad vowels). Success Ratio is defined for each vowel as the ratio of the number of times that vowel is used successfully to the number of times it is used in total. Throwaway Threshold is the minimum ratio allowed for a vowel, below which it is removed.
- Min. uses before throw: Before a vowel is removed as 'bad' it must be used at least the number of times defined in this parameter.
- Age to kill: This defines the maximum age for which an agent is alive in the system. On crossing this age, the agent is killed due to 'old age' and a new agent is added to the population. Hence, the birth rate is equal to the death rate.

  If this parameter is kept at -1, then the age factor is not taken into account (there is, hence, no preferential selection for the imitator and initiator). The option of 'turning off' the age factor allows us to notice the exact effects of age on the simulation.
- Shift Threshold: If the feedback received by the imitator is a failure, the success ratio of the vowel used is compared with this threshold. If found smaller, then it is just a 'bad'

vowel and is hence shifted closer to the vowel perceived for the first time.

- Articulatory Merge Threshold: During the updates, the Euclidean distance between each pair of vowels is calculated. If found below this threshold, then these two vowels are merged. Their use counts and success counts are added.
- Frequency of adding a random vowel: This parameter gives the number of games after which a random vowel is introduced in the system. This is done to keep the vowel system evolving and to introduce novelty.

The rest of the definitions and algorithms are identical to those used in DeBoer's paper.

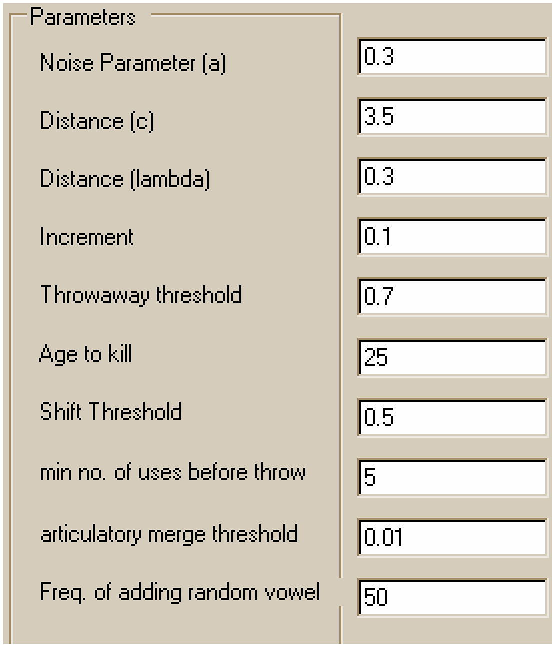
# EXPERIMENTS AND INFERENCES

## Experiment 1:- Plotting of formant space for varying number of games

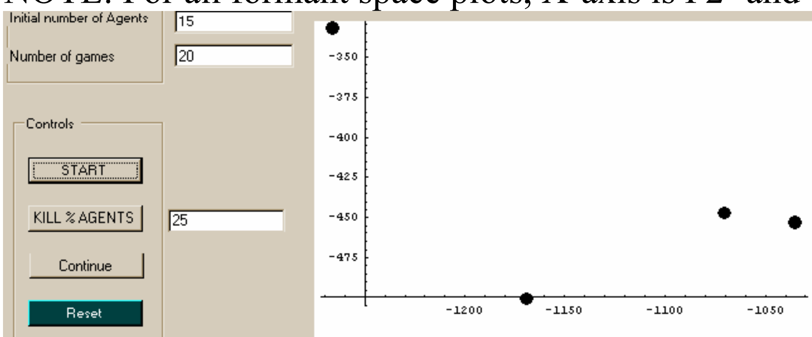
Our first experiment was to observe the formant space for a varying number of games with the 'Age to Kill' factor set to 25. We observed that the definiteness of the vowel space increased with the number of games as did the clustering of vowels.

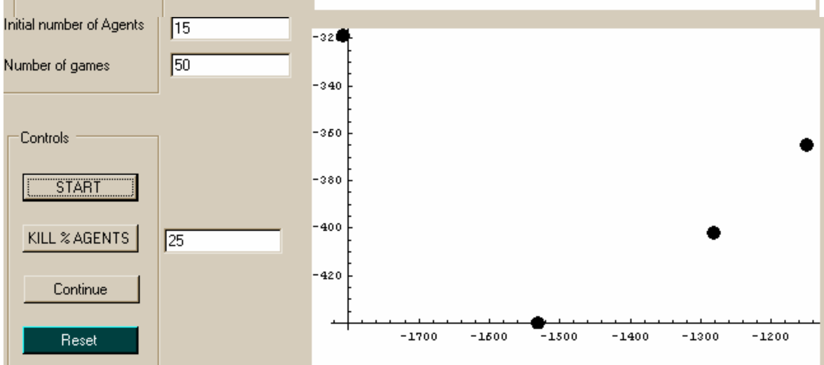
In DeBoer's experiments, the age factor is not considered. However, the results we obtained are similar to those in DeBoer's paper. Hence, one might infer that the effect of age is really not that significant. That is, the entire process of preferential selection and killing off of agents has no significant effect on the self-organization if the vowel system provided that the other parameters remain same and the population remains constant.

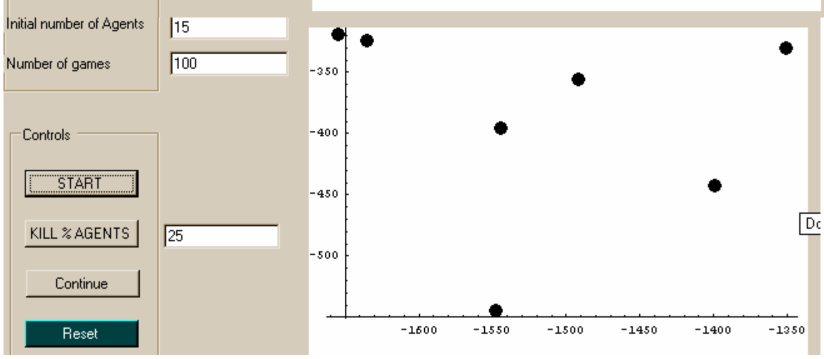
The plots of this experiment are presented below:

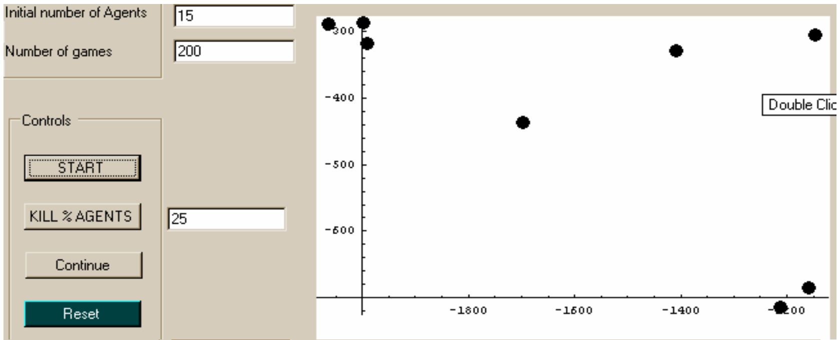


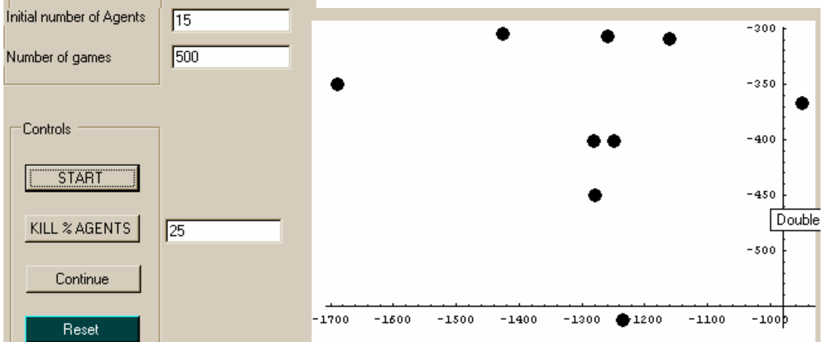

NOTE: For all formant space plots, X-axis is F2` and Y-axis is F1.

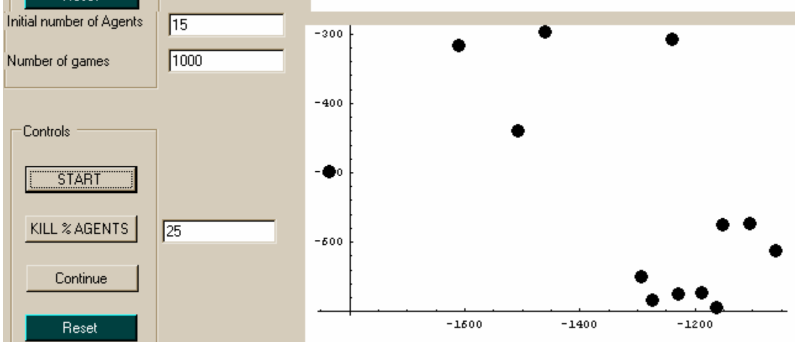


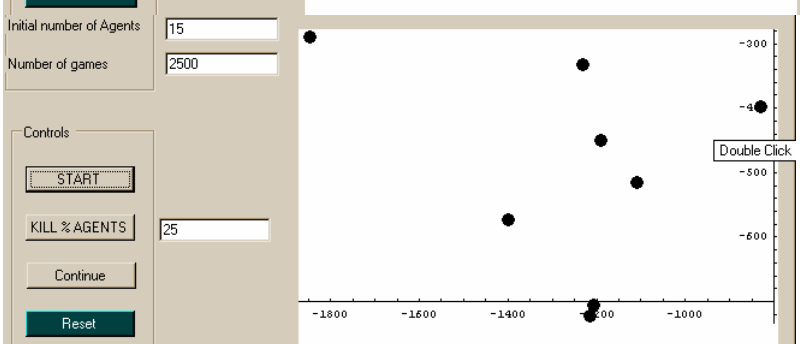



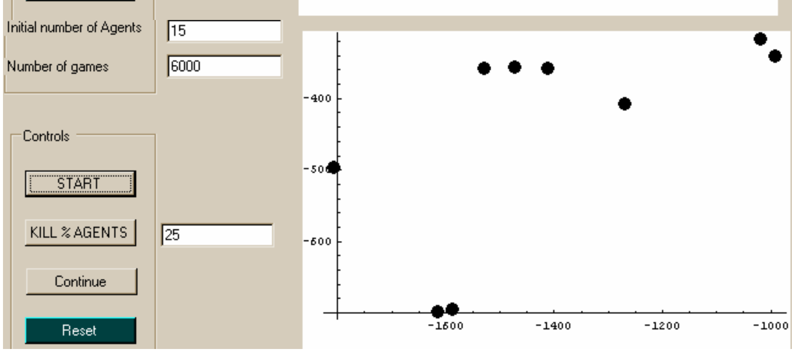



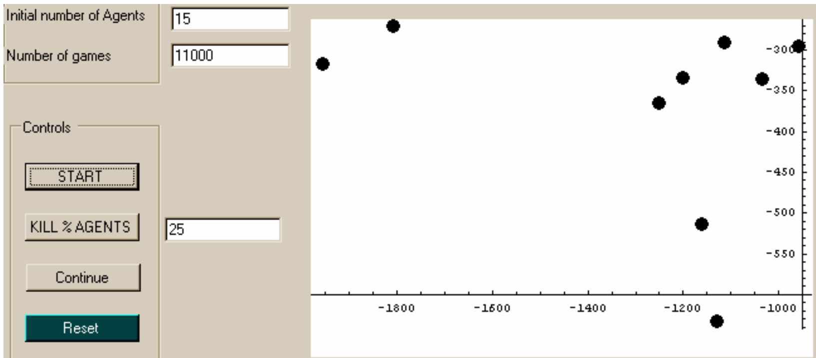

Initial number of Agents
15
Number of games
200
Controls
START
KILL % AGENTS
25
Continue
Reset
-400
-500
-600
-1800
-1600
-1400
Double Clic

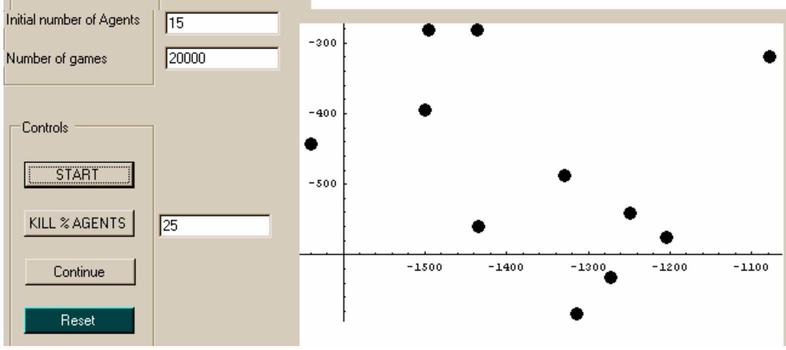

Initial number of Agents
15
Number of games
500
Controls
START
KILL % AGENTS
25
Continue
Reset
-300
-350
-400
-450
-500
-1700
-1600
-1500
-1400
-1300
-1100
Double


Initial number of Agents
15
Number of games
1000
Controls
START
KILL % AGENTS
25
Continue
Reset
-300
-400
-600
-1600
-1400
-1200


Initial number of Agents
15
Number of games
2500
Controls
START
KILL % AGENTS
25
Continue
Reset
-300
-500
-600
-1800
-1600
-1400
-1000
Double Click

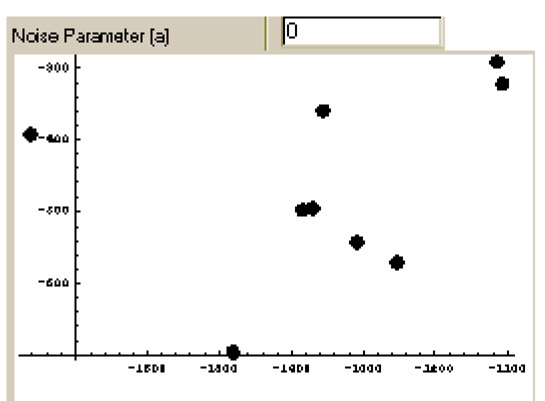

Initial number of Agents
15
Number of games
6000
Controls
START
KILL % AGENTS
25
Continue
Reset
-400
-600
-1600
-1400
-1200
-1000



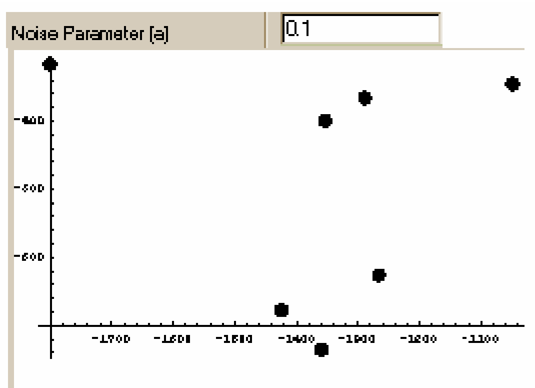




## Experiment 2:- Plotting of formant space for varying noise

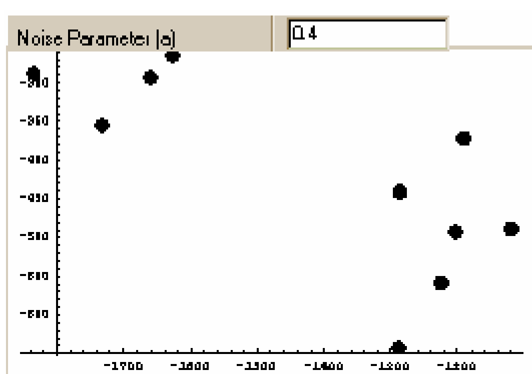
In the next experiment we observed the effects of gradually increasing the noise parameter ‘a’ on the formant space.
As the noise increases, the number of vowels decreases, and clustering increases. This can be explained by the effect of noise in the game. If the noise is high then two vowels which were distinct for smaller noise will now be merged with each other. Hence, the number of clusters will also reduce.
The plots have been tabulated below:



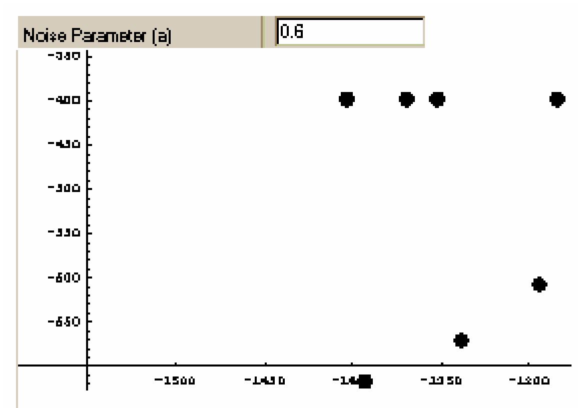

Number of clusters is approximately 5-6



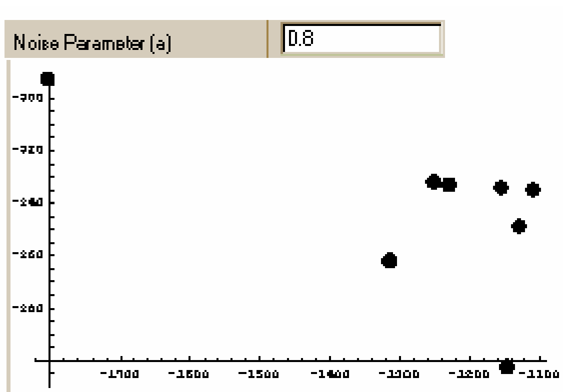

Number of clusters is approximately 4-5



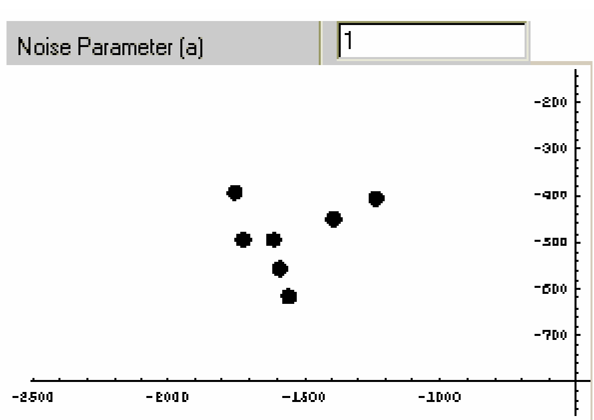

Number of clusters is approximately 3-4




Number of clusters is approximately 3




Number of clusters is approximately 2

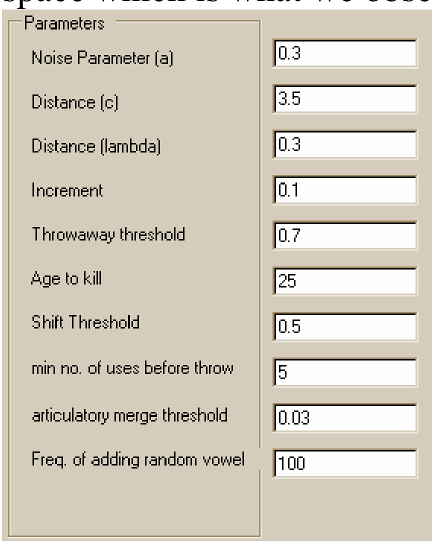



Number of clusters is approximately 1

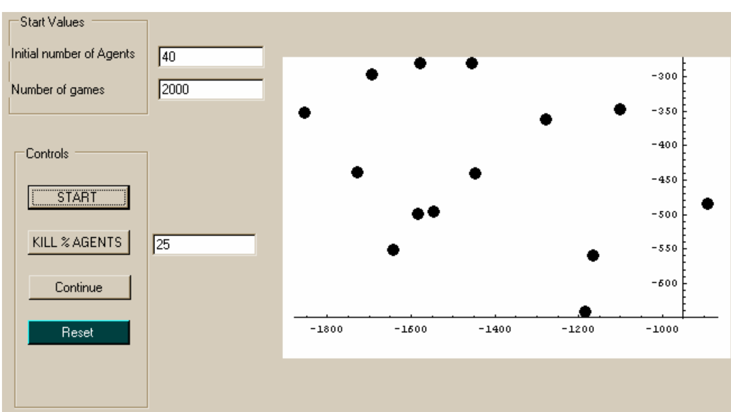
## Experiment 3:- Killing of agents: Language is a complex system

In this experiment we have tested the theory that language is a complex system. According to this theory, behavior is an indirect, non-hierarchical consequence of interactions among the agents. In other words, language is a global phenomenon and the resulting organization of the system is not explicitly intended by the agent interaction. Hence if a significant change is made to the population then it should be reciprocated with a significant change in the vowel space.

When agents are killed we expect a major change in the shape of the formant space which is what we observe in the following experiment.

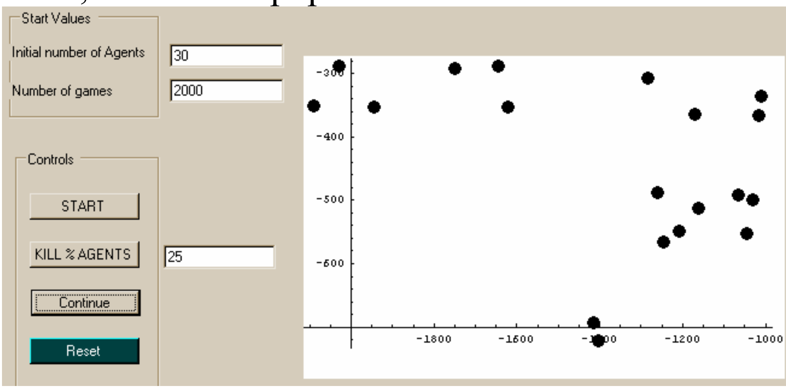


These are the input parameters.



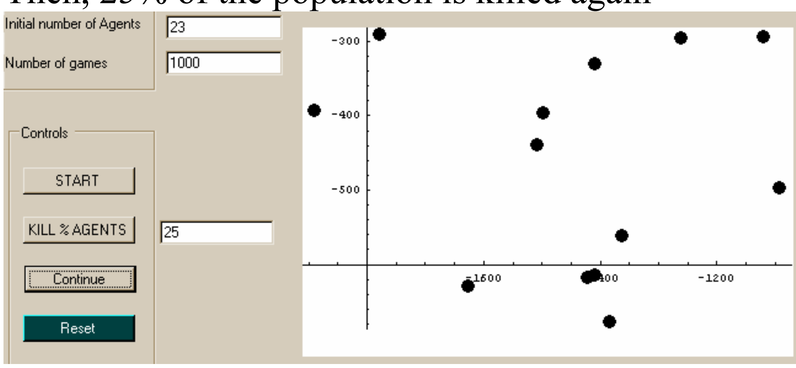
For the first time the simulation is run, for 2000 games and 40 agents.

Then, 25% of the population is killed.



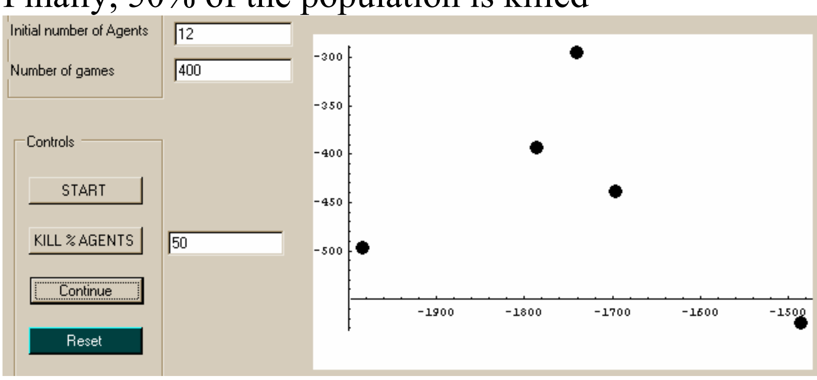

and after running the simulation for 2000 games, the result is as shown above. Note the marked change in the distribution of vowels.

Then, 25% of the population is killed again



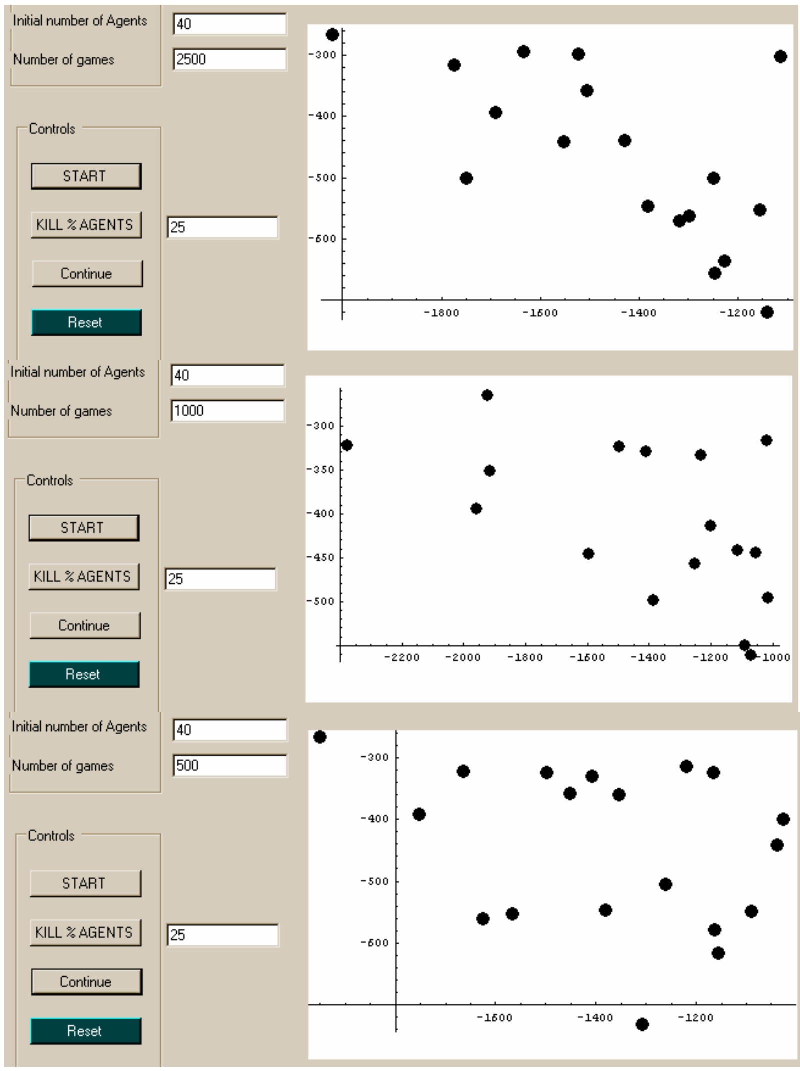

and 1000 more games are played to give the above plot which seems to bear little resemblance to the vowel space before this.

Finally, 50% of the population is killed




and 400 further games are played to get the above graph. The face of the vowel space stands completely changed.

Now, when a similar procedure is applied but the population is not reduced each time, we expect the vowel space to maintain the same distribution of vowels.

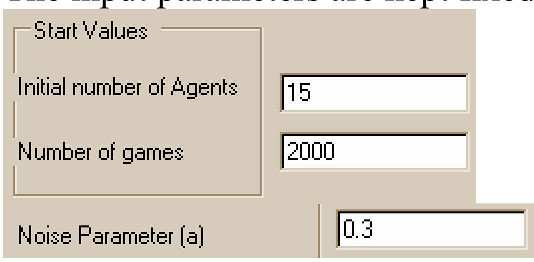


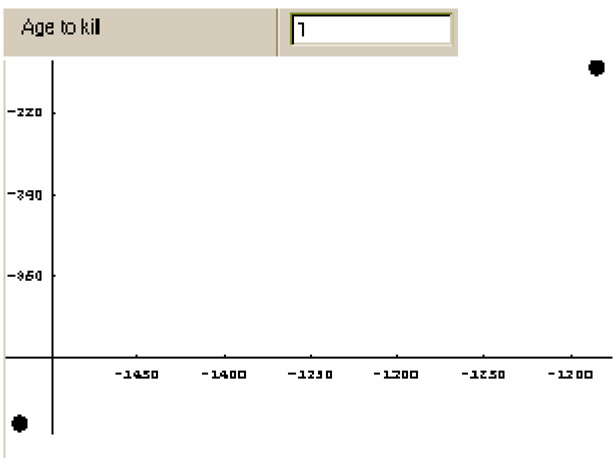

Clustering increases around the pre-formed vowels and there’s no significant change vowel distribution. The shape is almost same throughout.

## Experiment 4:- Plotting the formant space for varying 'Age to Kill'

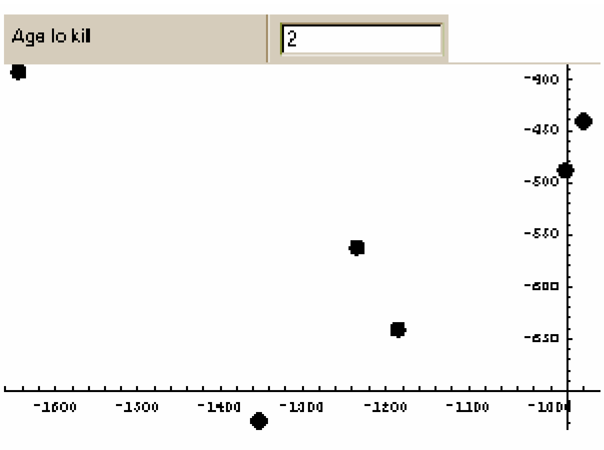
In this experiment we try to vary the 'Age to Kill' factor and see its effect on the formant space.

The input parameters are kept fixed at the following values:

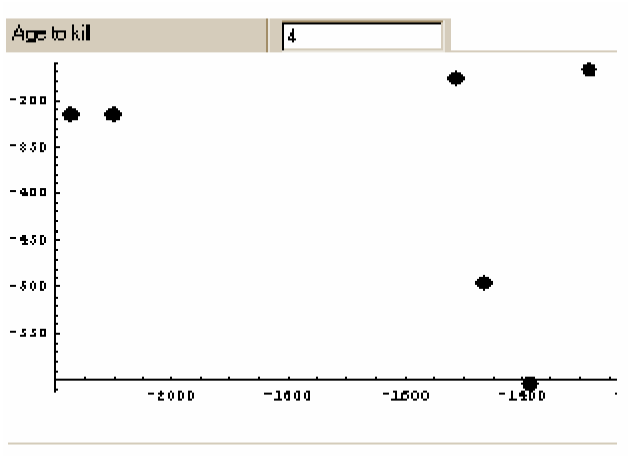


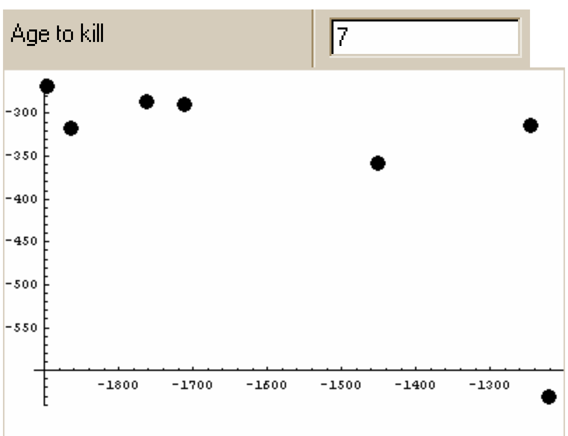



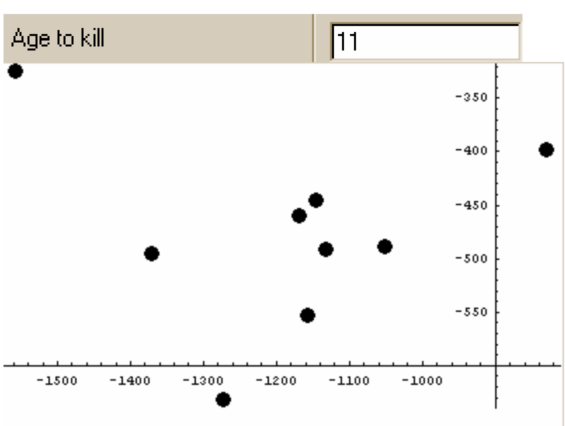

When 'Age to Kill' is kept at 1, agents will be killed off after playing only 1 game. Consequently, the vowel system does not develop.

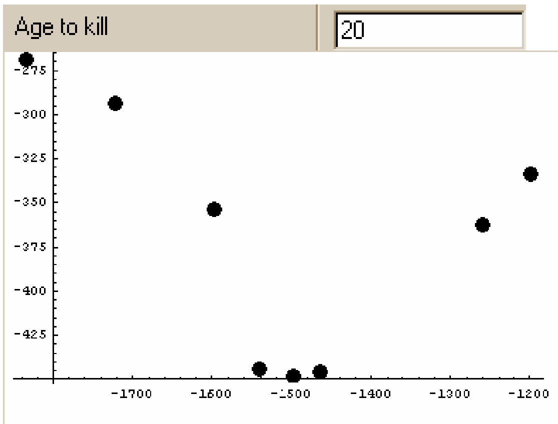


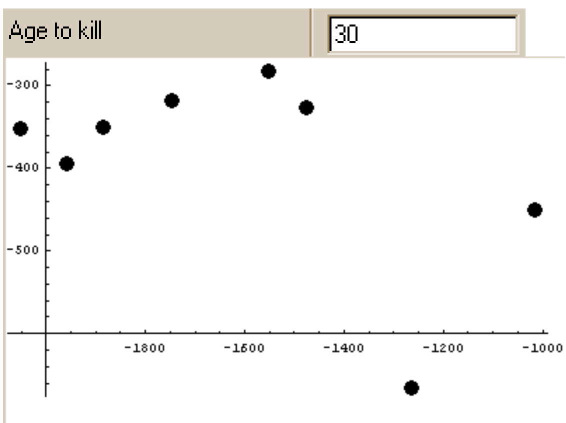
Now as soon as the factor is increased to 2 we get a developed vowel systems which remains almost same thereafter.

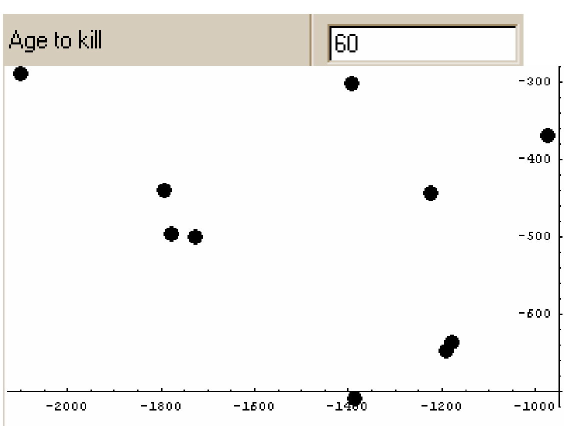


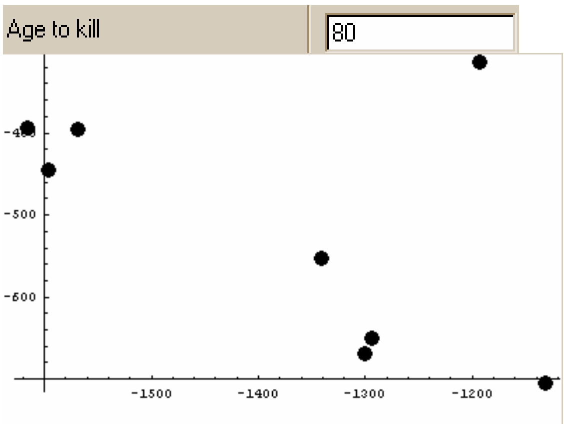




Age to kill
20
-275
-300
-325
-350
-375
-400
-425
-1700
-1600
-1500
-1400
-1300
-1200


Age to kill
30
-300
-400
-500
-1800
-1600
-1400
-1200
-1000

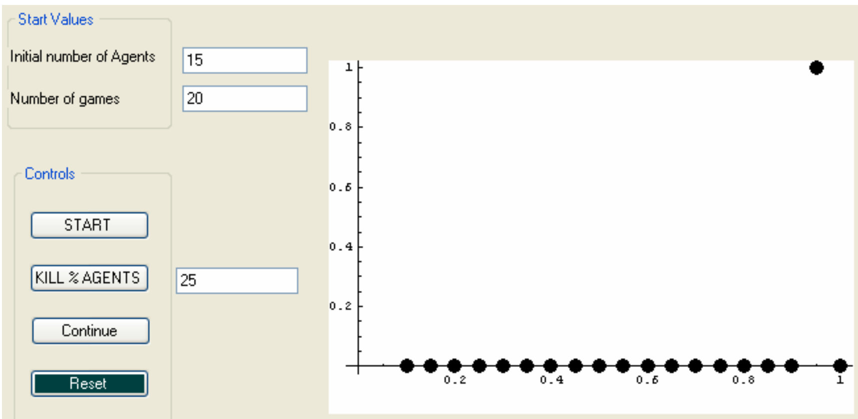

Age to kill
60
-300
-400
-500
-600
-2000
-1800
-1600
-1400
-1200
-1000

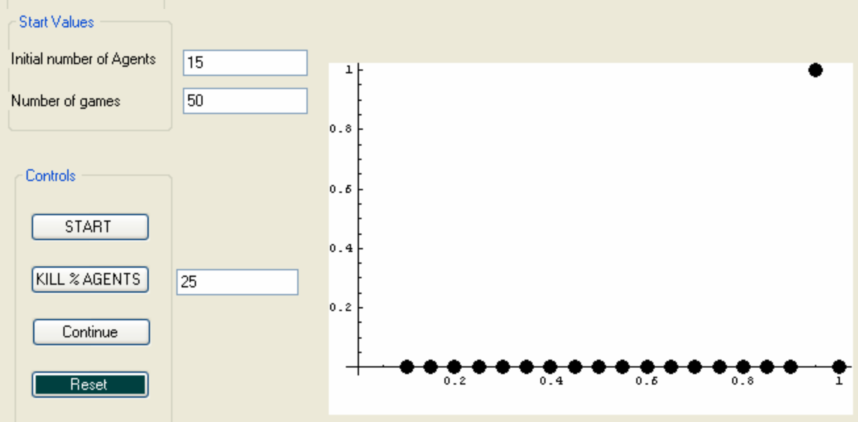


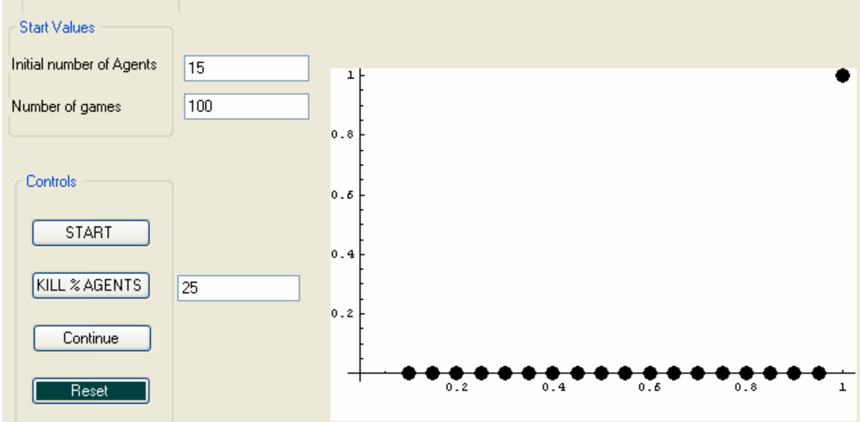

Now, we observe that as the factor has increased, the organization of the vowel pace remains more or less the same, but it has become a bit more defined as the clustering of vowels have become more prominent. It can be explained by the fact that the percentage of mature agent in the system increases as they are allowed to stay in the system for a longer time and hence, they have a stronger influence in the system thereby making the vowel space more organized and defined. These agents are preferentially selected to be initiators and hence, they drive the system towards a well-defined vowel space.

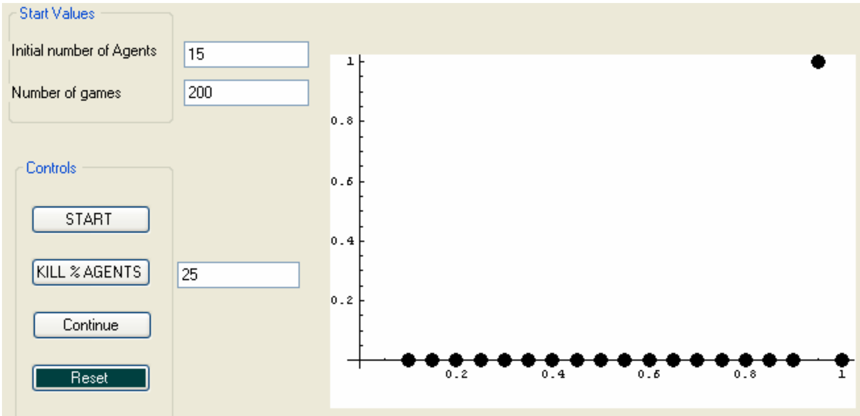
## **Experiment 5:- Success Ratio for varying number of games**

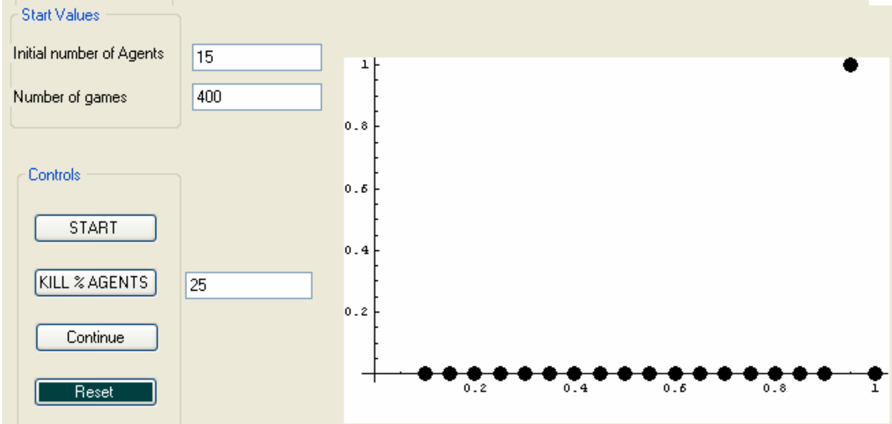
The simulation also plots the success ratio versus the number of agents. So we tried plotting some graphs of success ratio for varying number of games with the ‘Age to Kill’ factor included.
The plots are as following:

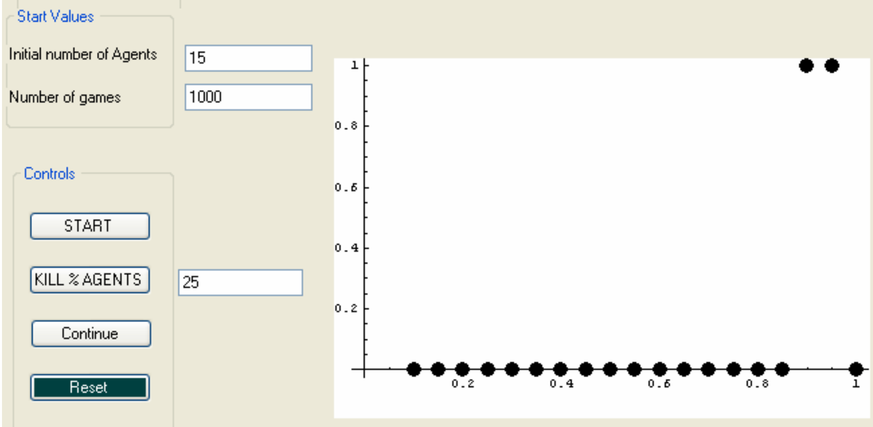


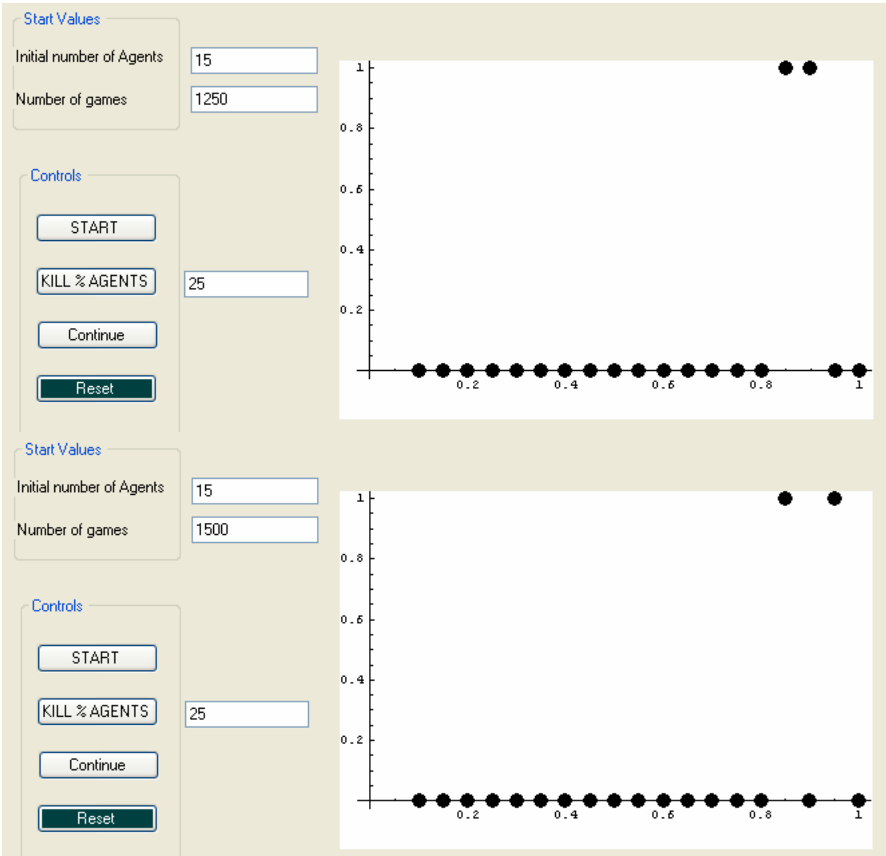




Start Values
Initial number of Agents
15
Number of games
200
Controls
START
KILL % AGENTS
25
Continue
Reset


Start Values
Initial number of Agents
15
Number of games
400
Controls
START
KILL % AGENTS
25
Continue
Reset


Start Values
Initial number of Agents
15
Number of games
1000
Controls
START
KILL % AGENTS
25
Continue
Reset




We observed that the number of agents with high success ratio is high while the others is zero.

NOTE: Initially, we did some simulations in which once an agent was killed off due 'old age' a new agent was not added in its place. But, this did not give us concrete results. Moreover, we could not infer anything physical from the plots. Hence we kept the population constant.

# CONCLUSIONS

The results of the experiments show that it is possible to generate realistic vowel systems in a distributed population of agents that try to imitate each other under constraints. No innate features that determine the form of the vowel systems were needed, nor does it appear to be necessary for the agents to inspect each others internal state. Also, the factor of age does not seem to have any significant effect as far as the self-organization of the system is concerned (for physically reasonable values of parameters like 'Age to Kill', self-organization always takes place and in a more-or-less similar way).

The experiments have also shown that the generated vowel systems are not static. They are constantly changing as a result of the invention of new phonemes, the shifting of existing phonemes due to noisy production, and the deletion and merging of phonemes. This is a phenomenon that is also present in natural language, albeit in less extreme way than in the presented system. The agents in our system are probably not conservative enough. However, the observed changes seem to indicate that sound change in human language can be explained by the mechanisms that have been proposed in this simulation.

Luc Steels' theory of language as a complex system is supported and justified by the results of our simulation.

*Future Works:* Another modification of the system would be to investigate more complex sounds. Investigating only vowels is easy, but also quite unrealistic if one wants to learn things about human language. One possible extension would be to investigate consonant-vowel syllables.

# REFERENCES

1. Self-organization in vowel systems (Bart de Boer, 2000)
2. Language as a complex adaptive system (Luc Steels, 2000)
3. www.csl.sony.fr