ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING DEPARTMENT

A Simulation of Armed Unmanned Aerial Vehicles Swarm

By

Cemil Uysal (16050111006)

Levent Gümrükçü (16050111029)

Berkay Kırtıllı (1605011037)

Bilal Emre Karagöz (16050111039)

Supervised by

Tarık Ismet Alkan

February 2021

Abstract

In the study, it was discussed that unmanned aerial vehicles work together by establishing a communication network, based on Complex Systems, Agent-Based Systems, Swarm Technologies, and Multi-Agent-Based systems. A simulation is tried to be created by using the information obtained as a result of the researches. The NetLogo application was used to simulate the collective communication and mission functions of unmanned aerial vehicles. The details of the issues that are the source of the study are as follows:

Agent-based model: Agent-based is the model class of system for being similar to any system behaviors. This topic combines evolutionary programming, complex systems, emergence, computational sociology, multi-agent systems, and game theory. There is the method using for introducing random which name is Monte Carlo Methods. The agent-based model calls individual-based models (IBMs), especially in ecology topic[i]. An agent-based model can be used for biology, ecology, and social science even which are not about computer science[ii]. The main purpose of the agent-based model is to explain common behaviors that provide basic rules of natural systems.

Complex systems: Systems consisting of many components that can interact with each other are called complex systems. Complex systems can also be formed by combining other complex systems. A complex system investigates the behaviors of its components between each other and the interaction of the system with the environment.

Multi-agent systems: Multi-agent systems are a complex system structure that provides more than one system, data transfer, and communication network among themselves. Each module in the system divides the complex system into subtitles and focuses on different problems at the same time to solve the whole system. . With this system, difficult operations such as complex artificial intelligence operations, mathematical equations with many

unknowns, or machine learning operations can be solved more easily and naturally by using different agents together.

Swarm cognitive entity: Swarm of bees has a democratic decision-making system. That is, the swarm of bees make all their decisions jointly by exchanging information. In other words, there are no single decision-making individuals or groups of bees in a swarm. Although each bee is cognitively limited and has very little knowledge, they build their brains very similar to each other. Therefore, the swarm of bees becomes a first-class decision-making group.

Keywords: Simulation of unmanned aerial vehicles, multi-agent systems, agent-based systems, swarm cognitive entity, complex systems, Netlogo.

Table of Contents

Abstract 2

Table of Contents 4

List of Tables 6

List of Figures 7

Introduction 8

Swarm as Cognitive Entity 8

Conceptual Framework for Decision Making 8

The Sensory Transformation in a Swarm 9

The Decision Transformation in a Swarm 9

The Action Transformation in a Swarm 10

Complex Systems 10

Nonlinearity 10

Network 11

Emergence 11

Adaptation 12

Agent-Based Systems 12

History 13

Theory 14

Applications 14

Implementation 15

Verification and Validation 16

Multi-Agent Systems 16

What are Multi-Agent Systems (MAS) 16

Application Field 16

Advantages of Multi-Agent Systems 17

Architecture 17

Methodology / System Design 19

NetLogo 19

Usage and Features of NetLogo 19

Functional Requirements on NetLogo for Project 19

Features of UAV’s 21

Features of Air Defense Systems 22

Interface 23

Code Design 24

Class Diagram 24

Creating Environment Codes in NetLogo 25

Result 29

References 31

List of Tables

Table 1 : Features of UAV’s 21

Table 2 : Features of Air Defense Systems 22

List of Figures

Figure 1 : User Interface 23
Figure 2: Class Diagram 24
Figure 3: Creating hills 25
Figure 4: Creating green areas 25
Figure 5: Creating lake....................25
Figure 6: Creating mountain 26
Figure 7: Creating river....................26
Figure 8: Creating target point....................27
Figure 9: Creating objects....................27
Figure 10 : Creating air defense system....................27
Figure 11 : Creating UAV’s....................28
Figure 12 : Move functions of UAV’s....................28

Introduction

This study focuses on the simulation of unmanned aerial vehicles using the Netlogo application, based on complex systems, agent-based systems, multi-agent-based systems, and swarm intelligence technologies. The main goal of this project is to analyze the future of drone wars which involve some of the most powerful land-based anti-aircraft weapon systems. This can be broken down into the following objectives:

Developing a software tool that could be used for realistic simulations of some selected armed unmanned aerial vehicles swarm.

Using that tool to study the effects of externalities on the dynamics of collective behavior exhibited by UAVs.

Swarm as Cognitive Entity[iii]

Conceptual Framework for Decision Making

When the honeybee swarm learns about the characteristics of more than a dozen sites, they activate the decision-making process to find the best site. According to Leo Sugrue, Greg Corrado, and William Newsome, neuroscientists at Stanford University, there are three different stages of simple perceptual decision making.

First of these stages, a sensory transformation. It is to convert the information acquired by the animal into sensory representation so that it can be further processed. Second of these stages a decision transformation. It gathers in a cluster the movements to be applied for alternative feedback versus sensory representation. Third an action transformation. It transforms this cluster of behavioral action.

Honeybee swarm decision-making looks like animals' decision making. These three stages are shown in both types of decision-making systems. Also, Integrators of information flowing between different options have blocking features for information outside of itself in the stage of sensory transformation. Finally, the decision is made when the accumulation of evidence reaches a sufficient level.

The Sensory Transformation in a Swarm

The honeybee swarm has scout bees. Scout bees fly in all directions and miles scanning areas for settlements. When scout bees think that they have found a possible place, they return to the herd and dance to tell the place they have found. The rotational strength of the dances is directly proportional to the quality of the site. In addition to the characteristics of the site, they find with this dance they also tell where the site is.

In time, dozens of scout bees return to the swarm and tell where they found and site characteristics so that they have enough information to decide. Besides, many scouts' bees report on the same site, thus increasing the accuracy of the information by calculating the average of all dancing power due to bee-to-bee differences in dance power.

Each scout bee decides how strong it will dance through an independent assessment. This independence of scouts means that an announcement cannot be imitated. This ensures that the total dance amount is a correct indicator. If the scouts' feedback is found positive by the honeybee swarm, additional scout bees are sent to this site. Over time, the total dance power was exhibited. Thus, information about fewer quality sites will disappear over time. Besides, this reduction in dance levels allows better sites to be discovered over time.

The Decision Transformation in a Swarm

The increase in the number of scout bees in different candidate sites creates a mutual inhibition in the group with a limited number of scout bees. In other words, the higher the number of scout bees on a site, the lower the number of scouts sent to another site.

The second stage is decision transformation. According to the previous stage, many scout bees have different dance powers for a site and the swarm sends additional scout bees to the sites it deems appropriate by averaging these powers. The number of bees in an area in the last few hours integrates the bees' information about the site. The better a site is, the more scout bees will be sent and feedback, the larger the pool of information promoting that site, and the stronger the additional scout bee going to the site.

The Action Transformation in a Swarm

When the number of scout bees in the selected site reached a sufficient level, the scout bees returned to the herd and warned the herd for the journey. While the non-scout bees in the swarm were preparing for flight, the scout bees decided on the signals to be used to direct the swarm and the route to be used. Finally, the scouts who determining the route to be direct the swarm along the way.

Complex Systems

Complex systems science provides new ways to understand the biological, physical, ecological, and social environment. With the development of technology, complex systems are analyzed in more detail and great scientific data is reached. For example, new theoretical questions arise in the fields of physical and human science, leading to new research. As we can see, complex systems do not have to be under a single branch of science. Since complex systems are interdisciplinary, it accelerates the flow of information between disciplines. It reduces the gap between disciplines by establishing a link between natural and social sciences.

Such systems that we can see in different areas have different behaviors. Therefore, complex systems do not have strict general rules. However, it is possible to talk about the 4 features of these systems. These features are nonlinearity, network, emergence, adaptation.

Nonlinearity

The nonlinearity property is not getting the same result with the same input. Inputs increasing at certain rates may not translate into linearly increasing outputs. There may even be no output. While small changes can have dramatic and unexpected effects, major changes will not always lead to major changes in a system's behavior. The butterfly effect is when small changes cause big changes. For example, the corona virus is a virus that originated in Wuhan, China, but it has affected all countries of the world by 2020. Hundreds of thousands of people died and sick. The economies of the country have suffered greatly.

Network

A network means which is a collection of separate objects and the relationships between them. System complexity can increase exponentially as the connection between parts increases. Examining a complex system as a network allows us to benefit from network science. We can simulate global air transportation to better visualize the network in our minds. Aircraft flight routes are links within the network. As the route line increases, the system becomes more complex.

Emergence

Philosopher Aristotle "The whole is more than the sum of its parts". The features and behaviors that are not found in the components and that emerge when the system becomes. This situation may be due to the interaction and relationship between parts. We can reinforce the following word of researcher Peter Dodds "There's no love in the carbon atom, no hurricane in the water molecule, no financial collapse in a dollar bill".

A feature in complex system components can cause a very opposite feature occurrence when the parts come together. To give a concrete example, although hydrogen and oxygen atoms are flammable, the resulting water (H2O) is not a flammable substance. It is even used as an extinguisher in fires.

Complex systems arise because of the behavior of parts rather than standard responses, behaviors linked to a center, or pre-planned behavior. Let us consider a flock of birds. This flock is moving towards a skyscraper. The herd has no previous plans. The herd splits as they approach the skyscraper, cross the building, and reunites at the end of the building. Here the birds do their actions and some flock rules. Some birds do not follow others to avoid making the road longer. According to some, it follows the birds close to it because the distance is equal. Flock behavior occurs because of these instant decisions and interactions. In this example, we can easily see the collective behavior caused by the relationship between parts, which is the goal of complex systems science. It is possible to see the same situation in bees and systems such as the human brain. We can call this situation self-organization. Also, spontaneous order and self-organization can be given as a part of emergence.

Adaptation

Adaptation is a special case for complex systems. Systems that can learn from experienced situations and react accordingly in subsequent situations are called complex adaptive systems. Our immune system can be an example of this issue. It produces antibodies against a virus it has just encountered and can neutralize it the next time they encounter it, without any damage.

Agent-Based Systems

The main purpose of the agent-based model is to explain common behaviors that provide basic rules of natural systems.

ABMs is a subtopic of microscale models[iv] that can revive complicated events and simulate them. This system works by processing from basic structure to complex structure. IBMs are explicable, by using basic rules and moving up to using reproduction, economic benefits, or social status[v]. An agent-based model can have some qualities such as “learning”, “adaptation” and “reproduction”.

The agent-based model has some subtopics which are an environment, decision-making heuristics, learning rules or adaptive processes, interaction topology, and numerous agents. ABMs are generally applied as computer simulations and these simulations are used for testing when individual behaviors change how will affect to general behaviors of the system.

History

In the 1940s people begins to talk about the agent-based model. It could not be improved until the 1990s because it needs lots of computation-intensive procedures.

Early Developments

At the ancient time of agent-based model history, three important names interest in this topic whose are Von Neumann, Stanislaw Ulam, and John Conway. Simula programming language was used for automating computer simulations which are so common in the 1970s. This was the first step for automation.

The 1970s and 1980s (First models)

One of the oldest models for ABMs is the segregation model which belongs to Thomas Schelling[vi]. At the beginning of the 1980s, Robert Axelrod has developed so many agent-based models about political science[vii]. In the 1980s Craig Reynolds and Christopher Langton make so many studies about biological agents-based models. The first term call about “agent” was based on the “artificial adaptive agents in economic theory”[viii] presentation which belongs to John Holland in 1991.

The 1990s

In the 1990s Joshua M. Epstein and Robert Axtell were having studies about social science (seasonal migrations, pollution, sexual reproduction, combat, and transmission of disease)[ix]. Other studies are belonging to Kathleen Carley and Nigel Gilbert in the 90s.

The 2000s

In the 2000s Ron Son found a method which is about human cognition which is called cognitive social simulations[x]. Bill McKelvey, Suzanne Lohmann, and Dario Nardi contributed to the decision-making topic.

Theory

ABMs are adhering to rules set by programmers. Agents can act like clever or fool. It depends on situation to situation.

When people try to analyze the balance of a system ABMs can be used for this process and it is the most addictive side of ABM's benefit.

Framework

The last research of simulations shows that agent-based and complex networks must be combined[xi] under four sup topics which are complex network modeling, exploratory agent-based modeling, descriptive agent-based modeling, and validated agent-based modeling.

Applications

Biology

ABMs used for so many simulation areas such as biowarfare, population Dynamics[xii], stochastic gene expression[xiii], plant-animal interactions, vegetation ecology, analysis of the spread of epidemics, landscape diversity, ancient civilizations, evolution of ethnocentric behavior, forced to migration, language choice dynamics, inflammation, and human immune system[xiv].

Epidemiology

These days' ABMs are used for SARS, Cov-2 simulations[xv]. This usage is criticized by the science community because these simulations are based on basic phenomena and unrealistic phenomena[xvi].

Business, Technology and Network Theory

After the 1990s ABMS beginner to usage. Example usage areas are marketing, organizational behaviors, team working, supply chain, logistic, modeling of consumer behaviors, social network effects, and analyze traffic congestion[xvii].

Economics and Social Science

After the economic crisis, to analyze economic ABMs importance is increased in 2010, journal of The Economist publishes research which is about ABMs can use instead of DSGE models. This journal recommends to readers that ABMs are the better source to represent the economic complexity. Based on ABMs research, it shows that network morphology and stock market index have connected each other[xviii].

Organizational ABMs

ABMs have a subbranch which is ADS (agent-directed simulations) and this brand separates two topics which are “Systems for agents” and “Agent for systems”[xix].

Self-Driving Cars

There is a technology development company about autonomous driving in America which name is Waymo. Waymo develops a simulation environment which name is Car craft[xx]. It uses agent-based models for simulation. This simulation includes car drivers, pedestrians, and their behaviors.

Implementation

Beginning of the research of ABMs, most of them designed connected with von-Neumann computer architecture. Because of this relationship, its limited scalability and this problem can cause hinder model validation[xxi]. But with the improvement of technology GPUs can use as parallel architecture. In this way, this problem can exceed.

Integration with other modeling forms

ABMs are generally used with another modeling type because it is not software. For example, geography information systems relate to ABMs and they have benefited from this[xxii].

Verification and Validation

For simulations that include ABMs, it important that verification. Validation progress consists of sensitive analysis, face validation, calibration, and statical validation[xxiii].

Multi-Agent Systems

Multi-agent systems emerge with the combination of more than one system to create a smart system. Connect these different systems as a collective structure which are acts as a whole body.

In recent years, developing artificial intelligence technologies and the resulting need to solve complex systems emerged as a result. Distributed artificial intelligence algorithms solve the emerging problems with 3 basic methods. These are parallel artificial intelligence algorithms, distributed problem-solving algorithms, and Multiagent systems. [xxiv]

What are Multi-Agent Systems (MAS)

These are systems that are developed by more than one agent on their own, each specially developed for solving problems in different areas, formed by coming together and have communication networks among themselves.

Application Field

Multiagent systems are used in many different areas today, these areas are mostly requiring high processing capacity and heavy operations. Some of these areas are.

In the science of Biology, this system is used for making examinations and research about the 3-dimensional structure of proteins. [xxv]

This system is also frequently used in computer network systems. The main examples are cloud computing, social networks, and security. [xxvi] [xxvii]

MAS is used in robot technology, in the coordination of the parts used. [xxviii]

Also, there are several application fields for Multi-Agent Systems like modeling complex systems, the city builds and environments, and smart grids. [xxix]

Advantages of Multi-Agent Systems

Since multiagent systems are not managed over a single unit, an error that occurs at one point does not cause bad results such as the collapse of the entire system. While an error at one point in centralized systems may render the whole system dysfunctional, it is very unlikely that the same situation will occur in multiagent systems.

In multiagent systems, data from subcomponents can be globally coordinated from a single channel.

A MAS enhances overall system performance, specifically along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, and reuse. [xxx]

Multi-agent systems are faster than single-agent systems because in the single-agent systems whole processes are made in one unit but on the MAS whole processes divide into sub-agents and these sub-agents make operations simultaneously. This makes the MAS much faster than the others. [xxxi]

Architecture

Multiagent systems generally have two different structures, these are homogeneous structures and heterogeneous structures. These different structures express how agents are constructed and how they differ from each other. The structure preference of the system may vary depending on the problem to be solved. Depending on the problem, each structure may have advantages and disadvantages.

Homogenous Architecture

All agents used in homogeneous architecture have the same properties as each other. Examples for this agent may be the same brand and model. The most important point in this architecture is that it is simple to implement and build. It allows more operations to be performed at the same time by assigning the operations to be performed to sub-components. xxxii

Heterogeneous Architecture

In heterogeneous architecture, agents can have different characteristics and brands. Even different agent groups can be created to create a heterogeneous architecture. Each agent group can have different numbers and characteristics of agents. With all these groups coming together, the system becomes capable of solving different problems. Heterogeneous architecture can have a more flexible structure compared to homogeneous architecture. Since the ability of each agent group is different from each other, each agent group can fulfill the task of its subcomponent.

Methodology / System Design

NetLogo

Usage and Features of NetLogo

NetLogo Agent-based is an IDE where we can model and develop simulations. The language used in the NetLogo environment is an agent-based programming language inspired by an interpreted language called a logo. It is supported by different operating systems (Windows, Mac, Linux, etc.). It also allows working with the command line. You can either use it from the internet or download it to your computer for free and start using it. This environment allows us to see the changes that occur in the system over time, and to see how changes at the beginning of the system affect the result. NetLogo offers a model library. It is possible to make observations about that subject using these ready-made models. Besides being able to model easily, it is an environment that is advanced enough to create powerful models for researchers. In this project, we could use any logic system to simulate agent-based modeling. The reason we chose NetLogo in this project is that it is a good starting tool for those who will do agent-based modeling for the first time.

Functional Requirements on NetLogo for Project

There should be 5 different agent types in this system. These are unmanned aerial vehicles, air defense systems, convoy trying to reach to target, enemy soldier and civilians. Civilians can be added or removed with user's choice.

Environmental conditions must contain all environmental factors to reflect reality. These factors are mountain, lake, river.

These natural elements are an obstacle and agents cannot settle on them. Convoy can use bridges over the river.

Unmanned aerial vehicles must have flight time. Changes in flight time should be made according to the amount of load on it.

When they approach the end of their flight time, they should return to the base. It is necessary to calculate that they have enough time (fuel) to return to the military base.

UAV must do their flight in different modes, circular and random, following the convoy in different modes. This way, they better control the environment and better protect the convoy.

To be able to detect enemy factors during flight, they must have a radar system with a certain radar range.

UAV must have the ability to fire so that they can destroy enemy factors.

To shoot, it must have fire, unlike the radar range.

Drones need to communicate with each other. In this way, they can be more effective against the enemy.

Air defense systems must have two different range units as radar range and fire range. So, they will be able to imitate the truth.

Enemy soldiers should be able to move on the map as opposed to air defense systems.

It should be able to shoot convoys and drones. There must be a range for fire.

Convoy should try to reach the target point the shortest and most reliable way.

When a car which is in convoy is damaged by an enemy soldier, they should be able to continue their way. If it is judged that the safe road is broken, they should follow a new route.

Features of UAV's

Table 1 : Features of UAV's

	Altitude(feet)	Flight Time	Max Speed	Load capacity	Number of Loads	Fire Range	Communication Range
TB2 Armed UAV	18000-27000	27 hours	222 kmh	150 kg	4 pieces	8-14 km (MAM-L)	300 km (LOS)
Anka-Aksungur	40000	24 hours	180 kmh	750 kg	8-24 pieces	250 km (Som-A)	Unlimited (SATCOM)
Anka S	30000	24-32 hours	217 kmh	200 kg	4 pieces	8-14 km (MAM-L)	Unlimited (SATCOM)
Karayel UAV	22500	20 hours	148 kmh	70 kg	2 pieces	8-14 km (MAM-L)	150 km (YVT)
Akıncı	40000	24 hours	361 kmh	1350 kg	10 pieces	250 km (Som-A)	Unlimited (SATCOM)

As seen above, the properties of 5 different UAVs are given. Our reason for choosing these drones as Turkish drones is to find out in which directions, we need to develop them by applying simulations on our national technologies.

We will not use all the features that appear in the table. Because our simulation environment is not suitable for using all. For example, we will not be able to use the altitude feature because we are currently working in a two-dimensional environment. The properties we will use are flight time, number of loads, fire range, communication range. The firing range may vary depending on the type of missile the drone carries. We can see the missile types compatible with the drone through the interface. As seen in the table, the communication range of some unmanned aerial vehicles seems to be unlimited. Since the drones in our project work as autonomous flocks, they will only be in communication with each other. Therefore, communication ranges will be different from those here. We make other features using real data as much as possible.

We chose TB2, Akıncı and Aksungur among them. We think there are three different types of drones. We think this tree drones will be an effective team.

Domestic users the users of TB2, which is the Turkish Armed Forces, EGM and Gendarmerie General Command, are Ukraine, Azerbaijan, Qatar, and Libya. Since the mass production of the raider has just started, it is only used by the Turkish armed forces in our country. While these two unmanned aerial vehicles are produced by the BAYKAR company, our last drone, Anka-Aksungur, is produced by TUSAŞ. And mass production has just begun.

Features of Air Defense Systems

Table 2 : Features of Air Defense Systems

	Altitude of Target (feet)	Distance of Target
Pantsir S1	50000	20 km
9K33 Osa	49000	14,8 km
Tor – M2KM	32000	32 km
Hisar-A	~40000	15 km

We will only use the fire range from the data above. In addition, we will add radar range. We choose Pantsir S1, Tor-M2KM and Hisar-A among these systems. The reason we chose them is that we choose air defense systems with different features.

The Russian-made missile defense system Pantsir-S1 was acquired by countries such as Algeria, Iraq, Syria, United Arab Emirates, Oman, Jordan and Brazil. One of the superior features of Pantsir S1 is that they are used at extremely high and low temperatures. For example, these systems perform their military duties at +50 degrees in Syria and -50 degrees in the Arctic.

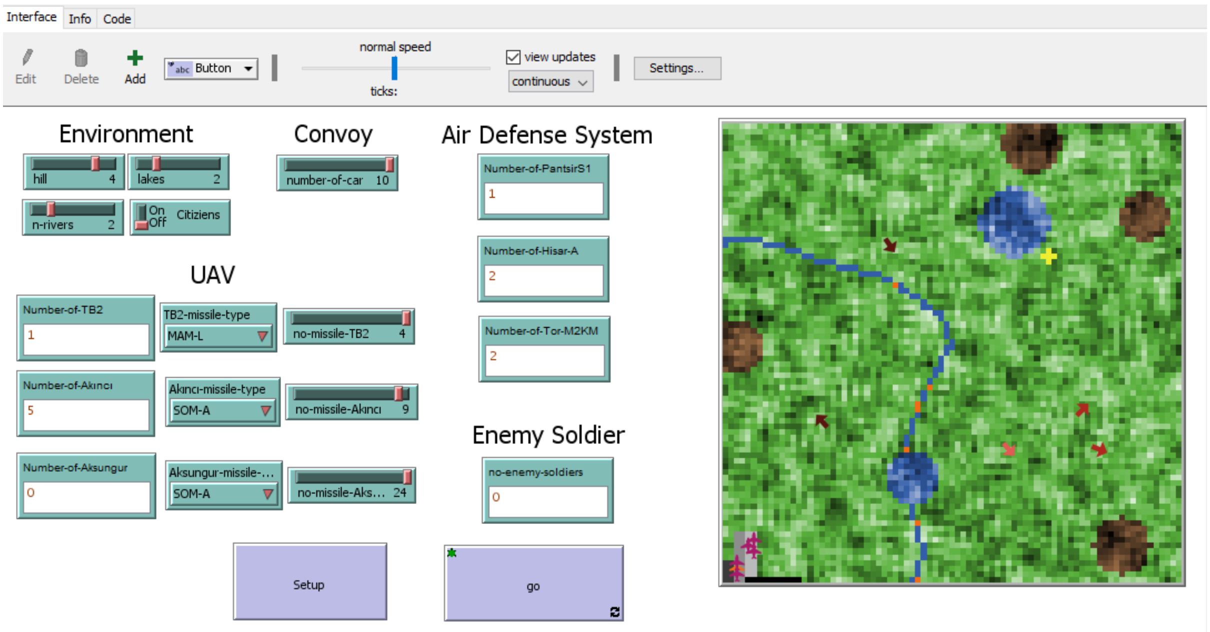
The TOR-M2KM can block medium and long-range cruise missiles, ship missiles up to 5 meters above the sea surface, winged and guided bombs, unmanned aerial vehicles, aircraft and helicopters. TOR-M2KM, which can operate in all conditions, can be integrated into any platform with carrying capacity. The TOR-M2KM, which can be set up in 10 minutes with a 25 thousand kilos crane, can detect approximately 140 targets and hit the four most dangerous targets at the same time.

Hisar-A (Low Altitude Air Defense Missile System) consists of four vertical Hisar-A lancers and missiles mounted on the FNSS ACV-30 vehicle chassis. An air surveillance radar and an electro-optic / infrared (EO / IR) system have been installed in the vehicle Hisar-A system to be an independent system.

We chose two defense systems from Russia because the Russians are very successful in the air defense system. Our other air defense system we chose is the Hisar-A system produced by Aselsan.

Interface

Figure 1 : User Interface



As seen in the screenshot above, we can change information about the environment, drones, air defense systems and enemy soldiers via the interface. The things that we cannot change in the interface are fixed in the code section. For example, drones have a fixed flight time and decrease in proportion to the number of loads they take. This data is data that cannot be changed from the interface, so it is determined in the background, not in the interface.

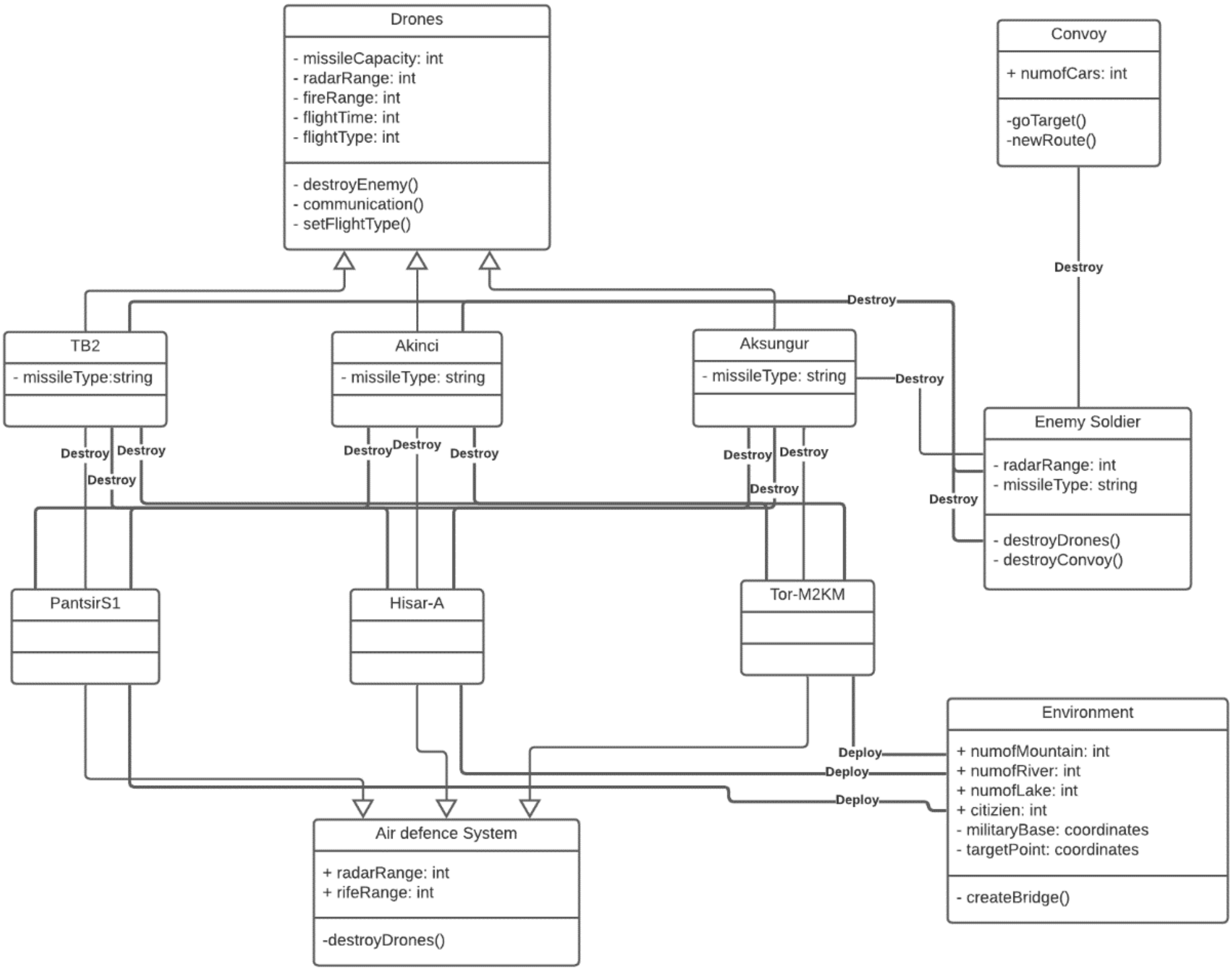
Our military base, which appears on the lower left on the map, is the point where our drones will take off and land and our convoy will act. It is always located at the bottom left.

Our target point is randomly determined somewhere on the upper right of the map. Other environmental factors are created randomly on the map and at certain distances from each other according to the numbers entered. In addition, air defense systems and enemy soldiers are randomly positioned outside of obstacles. To cross the rivers, we assign random bridges to the rivers. You can see these bridges as orange squares on the map.

Code Design

Class Diagram

Figure 2: Class Diagram



Creating Environment Codes in NetLogo

Creating Hills

Figure 3: Creating hills

```
; create mountains
ask n-of hill patches with [obstacle? = false and military-base? = false and enemies-area? = false]
[
      set pcolor brown
]

ask patches with [pcolor = brown] [
  ask patches in-radius ( 4 + random-float 2.5)
  [
    set obstacle? true
    set pcolor scale-color brown elevation 2000 13000
  ]
]
```

Creating green areas used on the map

Figure 4: Creating green areas

```
to setup-patches
  ask patches[
    set obstacle? false
    set hangar? false ; where uavs will land
    set base? false   ; where uavs take off
    set target? false ;target of our convoi
    set military-base? false ; hangar,base, and target
    set enemies-area? false ; enemies
    set elevation (random 10000)
  ]

  diffuse elevation 1
  ask patches[
    set pcolor scale-color green elevation 1000 9000
  ]
```

Creation of Lake

Figure 5: Creating lake

```
; create lakes
ask n-of lakes patches [
   ifelse random-float 2 <= 0.5 [
     ask n-of 1 patches with [ pcolor = blue and military-base? = false and enemies-area? = false] [
       set pcolor magenta
     ]
   ]
   [
    ask n-of 1 patches with [ obstacle? = false and military-base? = false and enemies-area? = false ] [
      set pcolor magenta
    ]
   ]
]
```

Creation of Mountain

Figure 6: Creating mountain

```
; create mountains
ask n-of hill patches with [obstacle? = false and military-base? = false and enemies-area? = false]
[
      set pcolor brown
]

ask patches with [pcolor = brown] [
  ask patches in-radius ( 4 + random-float 2.5)
  [
    set obstacle? true
    set pcolor scale-color brown elevation 2000 13000
  ]
]
```

Creation of river

Figure 7: Creating river

```
; create river
if n-rivers > 0[
    create-envconstructors n-rivers  [
      ; We place it river randomly on the map
      ifelse random-float 1 <= 0.5 [
        set xcor 0
        set ycor random max-pycor
        set heading 180
      ]
      [
        set ycor 0
        set xcor random max-pxcor
        set heading 0
      ]
      ; growing river
      repeat (2 * max-pxcor ) + (2 * max-pycor) [
        ; change directory
        rt random 30 - 15
        ; forward
        fd 1
        ; create a bridge
        ask patch-here [
         if military-base? = false and enemies-area? = false[
          ifelse random-float 1.5 <= 0.1 [
            set obstacle? false
            set pcolor orange
          ]
          [
            set pcolor blue
            set obstacle? true
          ]
        ]
      ]
      ]
      die
    ]
```

Creation of Target Point

Figure 8: Creating target point

```
;create target point
ask n-of 1 patches with [obstacle? = false and enemies-area? = false and pxcor < 30  and pxcor >= 15  and pycor < 30 and pycor >= 15 ] [
  ask patches in-radius 1
  [
     set pcolor yellow
     ask patches in-radius(6) [ set military-base? true]
  ]
]
```

Creating objects with using breed structure

Figure 9: Creating objects

```
patches-own [elevation obstacle? hangar? base? target? military-base? enemies-area?]
breed [envconstructors envconstructor]

breed [TB2s TB2]
breed [Akıncıs Akıncı]
breed [Aksungurs Aksungur]

breed [PantsirS1s PantsirS1]
breed [Hisar-As Hisar-A]
breed [Tor-M2KMs Tor-M2KM]

TB2s-own[ fuel location map-analysis]
Akıncıs-own[ fuel location map-analysis]
Aksungurs-own[ fuel location map-analysis]

PantsirS1s-own [ number-of-missile radar shooting-range ]
Hisar-As-own [ number-of-missile radar shooting-range ]
Tor-M2KMs-own [ number-of-missile radar shooting-range ]
```

Creation of air defense system

Figure 10 : Creating air defense system

```
;creating air-defence systems
to create-PantsirS1

  create-PantsirS1s Number-of-PantsirS1

  ask pantsirS1s [
    let x -28 + random 60
    let y -28 + random 60
    setxy x y
    set color red + 1
    set sıze 3
    set pcolor blue - 4
    set shape "arrow"
  ]
end
```

Creation of UAV's

Figure 11 : Creating UAV's

```
; Creating UAVs
to create-akıncı
  create-akıncıs Number-of-Akıncı
  ask akıncıs [
    set heading 0
    set color magenta
    set size 3
    move-to one-of patches with [base? = true]
    set fuel 100
    set shape "airplane"
  ]
end
```

Move function of UAV's

Figure 12 : Move functions of UAV's

```
to move-akıncı
  ask akıncıs[
    ;Movement part will be improved.
    right random 50
    left random 30
    forward 1
  ]
end
```

Result

The developing unmanned aerial vehicles and autonomous systems, the importance of drones in our lives cannot be denied. Unmanned aerial vehicles, which have achieved great success in our country, are of great importance in wars. We need these vehicles to keep our soldiers at less risk and protect our borders at lower costs. Again, in line with the developing technologies, these aircraft can now function as a team without the guidance of a person.

Autonomous and non-autonomous unmanned aerial vehicles technologies go through trial and error and testing processes. While developing these technologies, trial and error ways and testing processes take quite a long time and are costly. Simulation enables these stages to be done at a lower cost and in a shorter time. Using simulations, we can test it as if we have produced that product and make necessary changes according to the results. We can perform our experiments by creating the situations that our product may encounter in a computer environment. Almost every company that already produces such aircraft uses simulations.

The aim of this project is to protect our convoy, which is trying to reach the target point in the shortest and safest way, from enemy soldiers and air defense systems through fully autonomous unmanned aerial vehicles. The drones will communicate among themselves and distribute duties where necessary and work like a fully autonomous swarm.

The gains in using this simulation are, of course, time and cost savings. If we need to see these gains more concrete, it is possible to understand better with a few examples. We can see how effective the change in communication distance between each other is in locating and destroying the air defense system. According to the type of ammunition they carry, it can be found that there is a change in the number of ammunitions. We can find it more advantageous if we combine which drones and create a flock. Which flight modes are more effective in protecting the convoy can be identified? UAV can fly in different flight modes by sharing tasks between them. For example, some fly without ammunition to detect enemies and while doing this, they may not be caught by enemy radars because they do not carry ammunition. Later, the air defense system detected by sending coordinates to the striking drones ensures their destruction.

As given in the examples, it provides us speed and financial savings in many areas. Although it is beyond the scope of this project, if some changes are made on the project, it can be determined what kind of consequences the data about the structure of the aircraft have caused.

References

[i] *Grimm, Volker; Railsback, Steven F. (2005)*

ISBN 978-0-691-09666-7.

[ii] *Niazi, Muaz; Hussain,Amir(2011)*

https://web.archive.org/web/20131012005027/http://cecosm.yolasite.com/resources/Accepted_Scientometrics_ABM_Website.pdf

[iii] Honeybee Democracy. (2021). Retrieved 3 February 2021, from https://books.google.com.tr/books?hl=tr&lr=&id=txMkdG9G5acC&oi=fnd&pg=PP1&dq=honey+bee+democracy&ots=rFp_jC-3tP&sig=6OSI0WtQ9cM6RcXyuSuFSN8HqlU&redir_esc=y#v=onepage&q=honey%20bee%20democracy&f=false

[iv] *Gustafsson, Leif; Sternad, Mikael (2010)*

https://doi.org/10.1016%2Fj.mbs.2010.02.003

[v] *October 6, 2003*

https://web.archive.org/web/20110720041914/http://policy.rutgers.edu/andrews/projects/abm/abmarticle.htm

[vi] *Schelling, Thomas C. (1971)*

[vii] *Axelrod Robert (1997)*

ISBN 978-0-691-01567-5.

[viii] *Holland, J.H.; Miller, J.H. (1991)*

https://web.archive.org/web/20051027152415/http://zia.hss.cmu.edu/miller/papers/aaa.pdf

[ix] *Axtell, Robert (October 11, 1996)*

https://archive.org/details/growingartificia00epst/page/224

[x] *Sun, Ron, ed. (March 2006)*

https://www.cambridge.org/0521839645

[xi] *Niazi, M.A. and Hussain, A (2012)*

https://www.springer.com/biomed/neuroscience/book/978-94-007-3851-5

[xii] *Bauch, Chris (March 10, 2008).*

https://doi.org/10.1016%2Fj.ecolmodel.2007.10.002

[xiii] *Thomas, Philipp (December 2019)*

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6345792

[xiv]

- Fedriani JM, T Wiegand, D Ayllón, F Palomares, A Suárez-Esteban and V. Grimm. 2018

Journal of Applied Ecology 55: 600–611.

- Ch'ng, E. (2009)

http://complexity.io/Publications/NII-alifeVeg-eCHNG.pdf

- Situngkir, Hokky (2004)

https://arxiv.org/abs/nlin/0403035

- Czinkóczky, A. (June 7, 2016)

https://doi.org/10.5194%2Fisprs-annals-iii-8-145-2016

- Tarik; Stauffer, Dietrich (2009)

https://en.wikipedia.org/wiki/ArXiv_(identifier) , https://arxiv.org/abs/0905.2672

- Edwards, Scott (June 9, 2009)

https://en.wikipedia.org/wiki/VDM_Verlag

- Schulze, Christian (2009)

https://ui.adsabs.harvard.edu/abs/2009NYASA1167..221H

- Hunt, C. Anthony (2007)

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1839892

- Celada, Franco (2015)

http://www.crcpress.com/product/isbn/9781466597488

[xv] *Adam, David (April 2, 2020)*

https://www.nature.com/articles/d41586-020-01003-6

[xvi] *. Sridhar, Devi; Majumder, Maimuna S. (April 21, 2020)*

https://www.bmj.com/content/369/bmj.m1567

- Giardini, Francesca; Gilbert, Nigel (2020)

http://jasss.soc.surrey.ac.uk/23/2/10.html

[xvii] *Rand, William; Rust, Roland T. (2011)*

---

https://doi.org/10.1016%2Fj.ijresmar.2011.04.002

- Hughes, H. P. N.; Clegg, C. W.; Robinson, M. A.; Crowder, R. M. (2012)

https://doi.org/10.1111%2Fj.2044-8325.2012.02053.x

- Crowder, R. M.; Robinson, M. A.; Hughes, H. P. N.; Sim, Y. W. (2012)

https://doi.org/10.1109%2FTSMCA.2012.2199304

- https://en.wikipedia.org/wiki/United_States_Department_of_Transportation

[xviii] *Stefan, F., & Atman, A. (2015)*

Statistical Mechanics and Its Applications, (419), 630-641

[xix] *http://www.eng.auburn.edu/~yilmaz/ADS.html*

[xx] *Madrigal, Story by Alexis C August 14,2020*

https://www.theatlantic.com/technology/archive/2017/08/inside-waymos-secret-testing-and-simulation-facilities/537648/

- Connors, J.; Graham, S.; Mailloux, L. (2018)

[xxi] *Gulyás, László; Szemes, Gábor; Kampis, George; de Back, Walter (2009)*

https://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1649189

[xxii] *. Brown, Daniel G.; Riolo, Rick; Robinson, Derek T.; North, Michael; Rand, William (2005)*

https://doi.org/10.1007%2Fs10109-005-0148-5

[xxiii] *Sargent, R. G. (2000)*

https://en.wikipedia.org/wiki/CiteSeerX_(identifier)

- Galán, José Manuel; Izquierdo, Luis; Izquierdo, Segismundo S.; Santos, José Ignacio; del Olmo, Ricardo; López-Paredes, Adolfo; Edmonds, Bruce (2009)

http://jasss.soc.surrey.ac.uk/12/1/1.html

- Klügl, F. (2008)

https://doi.org/10.1145%2F1363686.1363696

[xxiv] M. Wooldridge, An introduction to multiagent systems. John Wiley & Sons, 2009.

[xxv] https://www.sciencedirect.com/topics/neuroscience/multi-agent-system

[xxvi] J. Bajo, F. De la Prieta, J. M. Corchado, and S. Rodríguez, “A lowlevel resource allocation in an agent-based cloud computing platform,” Applied Soft Computing, vol. 48, pp. 716–728, 2016.

[xxvii] A. Singh, D. Juneja, and M. Malhotra, “Autonomous agent based load balancing algorithm in cloud computing,” Procedia Computer Science, vol. 45, pp. 832–841, 2015

[xxviii] G. Dudek, M. R. Jenkin, E. Milios, and D. Wilkes, “A taxonomy for multi-agent robotics,” Autonomous Robots, vol. 3, no. 4, pp. 375–397, 1996.

[xxix] https://www.researchgate.net/publication/324847369_Multi-Agent_Systems_A_survey

[xxx] https://www.cs.cmu.edu/~softagents/multi.html

[xxxi] Balaji PG, Srinivasan D. An introduction to multi-agent systems. In Innovations in multi-agent systems and applications-1. Berlin, Heidelberg: Springer; 2010. pp. 1-27

[xxxii] Stone P, Veloso M. Multiagent systems: A survey from a machine learning perspective. Autonomous Robots. 2000;8(3):345-383