ANTI-SUBMARINE WARFARE MODELING AND SIMULATION

Deniz Aldogan, Mehmet Haklidir, Levent H. Senyurek , Yasemin Timar , Semuel Franko , Omer Eroglu , A.Faik Guven , H.Murat Genc

Tubitak Informatics and Information Security Research Center

{deniz.aldogan, mehmet.haklidir, levent.senyurek, yasemin.timar, semuel.franko, omer.eroglu, afaik.guven, murat.genc}@tubitak.gov.tr}

ABSTRACT

Submarines, modern naval warfare scenarios, especially in the context of asymmetric war remain the one of the most important platforms. Therefore, improvement of Anti-Submarine Warfare (ASW) capabilities is among the most crucial aims of many countries. In this field, they reduce the cost of training, reduced the potential risks and to provide a variety of tactical situations to military training purposes can be tried for fast simulation applications are more preferred to use.

In this simulation software study, complicated tactical situations can be simulated in various operating conditions and which involves tactical entities that can execute ASW commands autonomously, is explained. Surface, submarine, rotary-wing and fixed-wing platforms modeled in the current simulation system. These target platforms are equipped with various sensors, weapons and acoustic countermeasure systems. Target platforms can realize basic tasks such as moving towards a waypoint or along a path as well as complex tasks such as searching and engagement autonomously both individually and in groups called as *convoys*. Additionally, they can also display reflexive behaviors such as land, entity or enemy/torpedo avoidance. For managing scenario preparation and simulating Computer Generated Forces (CGF), the VR-Forces infrastructure, a commercial application framework, has been customized. The capabilities of the platforms developed to implement the software modules are integrated into the architecture of this infrastructure component CGF simulation engine. Results represent that platforms exhibit realistic behavior even in difficult conditions.

Keywords: Anti-Submarine Warfare, Modeling, Simulation, Virtual Forces

1. INTRODUCTION

In high-fidelity simulations, one of the most critical tasks that can be assigned to a simulation component is the modeling and simulation of different platforms. To provide high-fidelity, both realistic models and realistic controls should be employed for realizing the behavior of computer generated forces (CGF) in simulation. In addition to CGF capabilities, tactical simulations mostly require a scenario preparation application. Distributed simulation frameworks provide collaboration of different types of modules that have their own complicated modeling and algorithm mechanisms. We used a commercial simulation framework for this goal. This framework provides some basic abilities to all system; nevertheless, it is not possible to satisfy all necessities.

In this study, we shortly explain framework we used and modules that we integrated into that framework. Platforms we used have complex equations of motion and speed control, as well as they have various tasks such as move-to, search, engage. While performing main tasks platforms use fuzzy controller, land and other realistic targets avoidance behavior controller. Besides basic types of tasks we have convoy mechanism that we explained below. By the help of convoy mechanism platforms perform more complex tasks and act together in different situations. We designed and implemented our complex modules in different way and integrated them to main framework. Hereby, reusability and flexibility of software is achieved.

The software modules for the motion models, sensors, weapons and the fuzzy controllers belonging to platforms have been implemented in the C++ programming language and have been integrated to the component architecture of VR-Forces CGF application (VR-Forces back-end) as composite objects (VR-Forces Developer's Guide, 2006 ; VR-Forces The Complete Simulation Toolkit, accessed 2011). In addition to this, convoy mechanism and other complex task modules also implemented as separate modules and integrated to main software.

The remainder of this article is organized as follows. We briefly explain our main framework in section 2. In section 3, we explained artificial intelligence and tactical environment simulation. In section 4, we describe our main focus on convoy mechanisms and how to work complicated modules. In section 5, we explained our software design in detail. We illustrate some simulation results in section 6. Finally, section 7 concludes the article.

2. GENERAL SIMULATION ARCHITECTURE

In this study, movement models that take into account the environmental conditions (wave, current, wind, season, day and night difference) and hydrodynamic forces have been developed for surface, submarine and rotary wing platforms each with 6 degrees of freedom. The motion of all platforms is considered in 6 degrees of freedom since six independent coordinates are necessary to calculate state information of a rigid body. We explained in detail our previous work (Haklidir, Aldogan and Tasdelen, 2008; Franko, Koksal and Haklidir, 2009; Haklidir, Guven, Eroglu, Aldogan and Tasdelen, 2009)

Simulation architecture of VR-Forces, which is basically a commercial product being used in architecture, originally developed controllers and modules are integrated into this architecture.VR-Forces mainly have two main modules that are listed as back-end and front-end side. Front-end side provides management of scenario and simulation execution control. On the other hand, back-end side provides modeling and simulation of entities, controlling remote control entities, management of local entities’ plan and all other issues such as task, set.

According to design of VR-Forces, each entity has three types of components: sensors, controllers and actuators. Sensors, allows you to retrieve information about the environment around the object. Controllers, receive information about assigned task and lead object for task. Actuators, organize task information, run motion model regularly and update objects’ information such as speed, location. Commercial toolkit independent controllers are used to simulate quartermasters of the surface platforms. A flexible fuzzy logic that capable of simulating human expert behavior has been implemented. Fuzzy logic controllers, which are in fact heading and speed controllers that utilize fuzzy logic for their calculations, are implemented in conjunction with land avoidance calculations (Senyurek, Koksal, Genc, Aldogan and Haklidir, 2008).

Sensors have very important roles while platforms performing their task and making decisions. Each sensor component have been developed and integrated as a separate software module. Developing every advanced feature to be integrated into the entity behaviors as commercial toolkit independent software modules has been adopted throughout the realization of our system to maintain modularity and reusability. For further investigation, the reader can refer to our previous work (Aldogan, Haklidir, Senyurek, Koksal, Eroglu, Akdemir, Franko, Tasdelen and Akgun, 2009).

3. TACTICAL ENVIRONMENT SIMULATION AND ARTIFICIAL INTELLIGENCE

Utilization of modeling and simulation technologies in military areas is observed more frequently specifically on training and analysis applications. Creation of the tactical environment via Computer Generated Forces (CGF) and construction of war space with sensor and weapon capabilities of the entities in this environment has been seriously dealt with since 1980’s and has come along crucial improvement processes up to now (Pratt, 1996; Kocabas and Oztemel, 1998).

The development of CGF can be analyzed in 5 subsequent phases. First generation CGF realizes scenarios simply without using behavior models. Second generation systems execute simple behavior models. Routes and roads are determined by the user before or during scenario run while interactions can only be on these structures. Third generation systems apply tasks which are composed of previously planned, rule or state based modules. In such systems, there is a hierarchy mechanism between tasks. Furthermore, these tasks can be applied in parallel or sequentially to form other complex tasks and behaviors. Fourth generation systems possess autonomous command control processes over advanced third generation systems. Fifth generation systems have capabilities such as goal selection or learning (Aldogan, Haklidir, Eroglu, Franko, Timar, Guven, Senyurek, Genc, 2013).

The system implemented in this study has abilities of a third generation system. The user is the decision mechanism in command control processes except a few reflexive situations (land avoidance, target avoidance, etc.). The user can decide on issues such as which entities will take place in a specific scenario, which capabilities and parameters the sensors and weapons of the entities will have or on which areas, roads or routes the task will be carried out. These decisions can be made before or during simulation run. Once decisions have been made, tasks are performed autonomously in accordance with the chosen task parameters and behavior models.

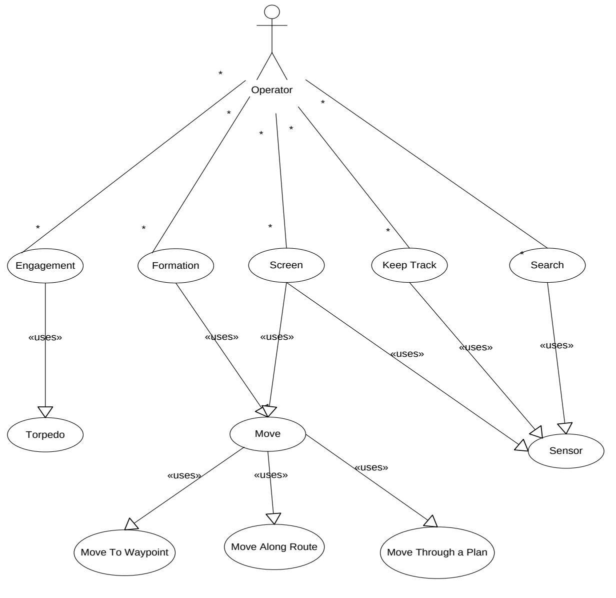

Figure 1 - Task and Behavior Hierarchy

Behaviors have been placed in task frames and these frames have been placed in other task frames. Therefore, main goals have been comprised of a task hierarchy. In this way, it is aimed to facilitate the construction of complex tasks and behaviors. The corresponding hierarchy is given in Figure 1.

4. ANTI-SUBMARINE WARFARE (ASW) CAPABILITY

4.1. Preferred Architecture and Basic Components for Implementing ASW Skills

A proper tactical training simulator should provide realistic target behaviour. Therefore ASW capability for marine ships and rotary wing platforms should be integrated into corresponding Artificial Intelligence (AI) models used in the training simulation.

As part of our work, requirements of ASW publications are surveyed for defining proper tactical behavior. Then, a set of basic actions are determined for developing a limited number of corresponding low level controllers which are used as building blocks to construct different complex tactical actions.

As a result, autonomous ASW capability of platform models is implemented considering an hierarchical behavior-based control architecture as in several other approaches in the literature (Michael, Henrik, Paul and John 2010; Krishnamurthy and Khorrami 2011; Aveek, Rafael, Vijay, James, John and Camillo 2002).

It is critical to properly determine the afore-mentioned basic/low level controllers to simplify the construction of complex behaviors. In our work we determined these controllers as a leader following controller. Speed controller for marine ships. For rotary platforms, leader following was unnecessary while an extra altitude controller was needed. These basic/low level controllers are constructed as fuzzy logic controllers for marine ships and as Linear Quadratic Regulator (LQR) controllers for rotary platforms. They all consist of separate components for speed and direction control.

For realistic response, land avoidance and conflict prevention behaviors for marine ships are also integrated into these basic actions. Land avoidance works in parallel with leader following and targeting behavior, as it controls the speed and direction of the models together with these controllers in a weighted manner. On the other hand, conflict prevention takes full control when necessary.

4.2. Constructing Complex Tactical ASW Capabilities

ASW tactics are applied by groups of platforms (marine ships and rotary wing platforms), which are called as *convoys* in these study.

Basic ASW behavior for a convoy is cruising in formation. For implementing such an action, convoy marine ships just apply the afore mentioned (low level) leader following control permanently. Both marine ships and rotary wing platforms might also be given screening duties, which is accomplished through (low level) targeting controllers. This time, for effective screening, several random target points inside a predetermined screen area are assigned to the corresponding models and each point is targeted one after another which is coordinated through high level screen controllers producing a realistic screening action.

Within the scope of ASW, applications of a search mission have similarities to convoy cruising. Searching is mainly cruising in an area with activated sensors, applying some special maneuvers if necessary. Again, each target model is assigned a set of target points which are visited in a specific order determined by high level search controllers. For accomplishing parallel search mission –another ASW search method– on the other hand, platforms apply targeting control for reaching their starting points in the first phase, while in the second phase they cruise parallel to each other exploiting basic formation control.

Similarly, for accomplishing convoy obstacle pass, basic behaviors are serialized in different phases. In the first phase each platform visits the canal points one by one while in the second phase they shift back to formation control.

Note that, applying leader following formation control does not necessarily need a leader platform to be determined. To accomplish most of the ASW tactics, imaginary leaders are created for more stable action, following similar works in the literature.

Attack missions have an additional attack phase in which platforms maneuver for assuring right conditions for weapon firing before firing their torpedoes'. This is also accomplished through basic targeting and altitude controllers.

4.3. Some Additional Information about Rotary Wing Platform Models Applying ASW

Rotary wing platform models, when arrived to a target point, hover at that point at a specified altitude and investigate their neighborhood via dipping sonar – which in our work is modeled as another mechanical element controlled by a separate controller applying the mission specific orders of the corresponding high level mission controller.

Since the flight time of rotary wing platforms are limited, they act in couples backing up each other coordinated by a high level controller for backup which is functioning in parallel with all the high level mission controllers for rotary wing platforms. Like other high level controllers this controller uses basic targeting and altitude controllers as well, for directing the models to proper targets –to a base ship or a mission point– when necessary.

5. SOFTWARE ARCITECTURE

Following sections will introduce the structure of the software components implemented to realize the ASW capabilities presented in the CGF. First section explains how a convoy of a several surface and rotary winged platforms is established. Secondly the high level mission controller and the low-level entity task controllers are described with their relations. Thirdly, rotary wing tasks explained. Lastly, we explained sensor fusion and tactical reflexive avoidance behaviors.

5.1. Convoy Generation, Update and Deletion

A convoy in the CGF is composed of entities of surface platform and/or winged platforms. Firstly the main ship of the convoy which can serve as a Leader is created or selected in the simulation environment and then the escort platforms are added to the convoy. All those entities are in relation to carry out escort tasks (low level) or convoy search and engagement missions (high level).

MainShip entity state repository includes the list of the escorts which are in the convoy. The escort entity state repository also includes the name of the mainship entity. This relationship is heavily used in the escort task controllers which require the mainship state as an input. Main ship state is also a reference for related algorithms especially leader following control.

In order to establish a convoy through the GUI and HLA1516, user input is transformed into interactions and objects that is processed by the CGF. The main interactions related with convoy generation and update are the selection of the main ship, adding escorts with a formation or screen mission, assigning search or engagement tasks to the convoy. Those interactions are first processed in the ASWCallbackHolder and ConvoyMapCallbackHolder which forms the necessary structure for entity tasks and controllers.

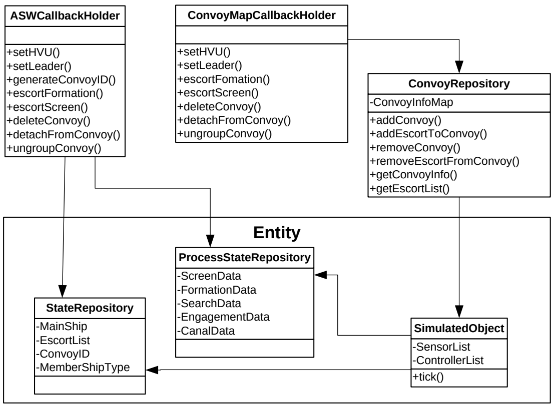

Figure 2 - Convoy Generation Information Flow

As you can see in the Figure 2, convoy generation interactions include selecting a mainship (setLeader, setHVU), adding escorts (escortFormation, escortScreen). User can also (1) delete an existing convoy (deleteConvoy) which deletes all of the entities from the simulated environment or (2) ungroup the convoy which makes all of the entities to end their convoy task or (3) detach an escort (detachFromConvoy) from the convoy. Interaction includes the information of which entity to be a leader/HVU, escort with formation and screen mission. According to this information ASWCallbackHolder updates the convoy identifier, and membership type of the related simulated entity. ConvoyMapCallbackHolder keeps all the relations of all of the convoys existing in the simulated environment in the Convoy Repository during runtime. Convoy sensor fusion or similar convoy mission/task managers which require all the entities of an existing convoy can query from ConvoyRepository.

5.2. Convoy Mission Control Architecture

ASW subsystem of the CGF enables the user to assign a group task to all or some of the escorts in a convoy. User interactions about a mission to search for a hostile subsurface entity in a specified region or a mission to engage to a hostile subsurface entity are first processed in the ASWCalbackHolder. The mission information (search/engagement region, selected escorts, etc.) is passed to the corresponding convoy task manager derived from BaseManager. You can see the various convoy task manager in Figure 3. BaseManager can access to ConvoyRepository to enable the task manager to query all of the convoys in the environment. Each convoy task manager assign the specific task to each escort to accomplish the convoy mission. This relationship is visualized in Figure 4. For example for a convoy to accomplish an engagement mission , ConvoyEngagementManager assign an attacker task to one escort and engagement task to the other escorts in the convoy which are selected to be a part of the mission.

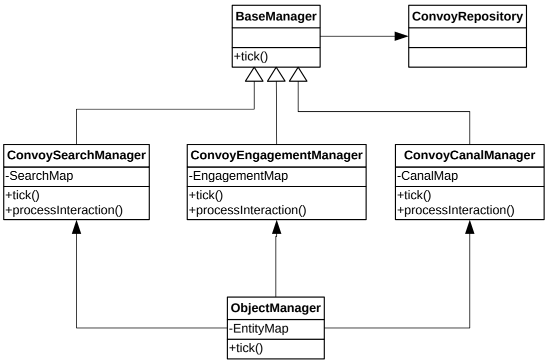

Figure 3 - Convoy Mission Architecture

ObjectManager, one of the most important classes of the CGF, executes each sensor, component and actuator of each entity in each simulation step. Convoy task managers are also executed by ObjectManager in each simulation step to handle convoy missions. The “tick()” function is overloaded in each task manager since it is executed in every simulation step.

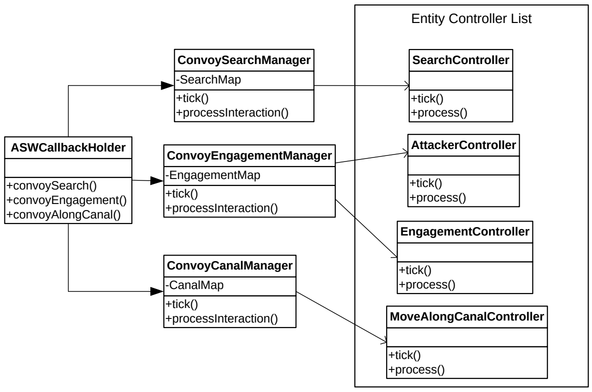

Figure 4 - Convoy Task Controller using Entity Task Controller

Escort task controllers are implemented using the existing task controller architecture to utilize the avoidance and dynamic models already implemented in the CGF. The relationship and the properties of the controllers are visualized in Figure 5. BaseController uses the CollisionAvoidance for calculating new routes to avoid colliding with the land or the other platforms in the environment. BaseController also executes AuxilaryController which calculates the basic state parameters of the platform model. Being inherited from BaseController, entity task controllers are driving the entity's behavior in a convoy mission in which several numbers of escorts are involved.

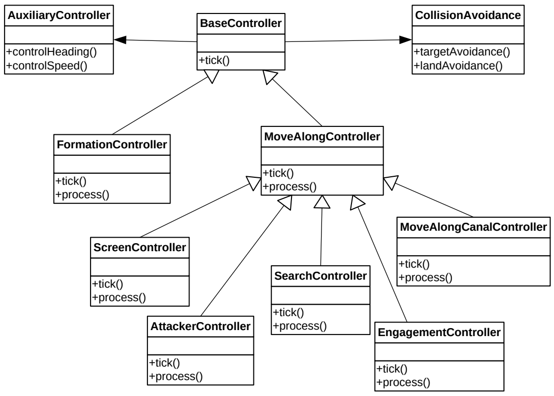

Figure 5 - The relationship and the properties of the controllers

5.3. ASW RotaryWing Tasks

The relationship and the in simulation environment combat ships have inventory helicopters. By using graphical interfaces, user can give screen, search or engagement tasks to inventory helicopters. After the task completion helicopters go back to their bases or user can stop task and command helicopter to return its base manually.

When user commands a task to rotary wing entity, graphical user interface sends the interaction to WingedCallbackHolder. This class collects winged entities' callbacks. User can command screen search task to helicopter. That interaction has the corner points' data of the search area. User can command a general search task which includes more general searching movements. After determination of the target platform user can command an engagement to target task. Engagement interaction includes approximate location, approximate bearing and possible route of target. Also torpedo attack points are passed.

As seen in Figure 6, callback holder's functions pass data to ASWManager class. ASWManager, which is inherited from BaseManager, evaluates the interaction data and processes controllers. Rotary wing platforms naturally exist in their ship base or land bases. When a search or engagement task is received, processTask runs and gives command to related controller (ScreenController, SearchController, EngagementController, ReturnToBaseController). If skipTask command is received, scheduleSkipTask method registers current timestamp. After 5 minutes of this timestamp helicopter's task will be stopped. Because of the operational time limitation, most of the helicopter tasks are paired tasks. When the flight time of a helicopter decreases to critical values its pair is commanded to continue the task.

ObjectManager, which controls all objects inside the simulation, runs its tick in every simulation step. Its tick also runs ASWManager's and other managers' ticks. ASWManager's tick checks and updates flight times of the helicopters. When needed it creates pair by using createPair, evaluates current task and pass current task parameters to pair helicopter. It also checks for the position of the new helicopter. If helicopter reaches pair's position, returnToBaseController runs, current helicopter returns to base for fuelling its fuel, torpedoes etc. Also user can stop helicopter's search missions in the middle of the task or delete inventory helicopter. In this case task is deleted and helicopter will be returned to its ship or land base.

Although callback holder and manager classes pass the commands, in most of the simulation time controller classes run. For instance if the task is screen, processTask of the ASWManager runs process method of the ScreenController. It assigns search points and dipping sonar depths to helicopter. After this initialization controller's tick runs each step. In each tick RotaryActuator is run. Actuator makes calculations and updates data in its state repository. State repository has heliData, taskData and motionData. HeliData includes helicopters name, id, pair number, current flight time, total flight time. Task data includes detailed task info and skip task's timestamp. Motion data includes linear position values, angular positions, linear and angular velocities of helicopter. Local network interface classes use these data and publish it to graphical user interface for operator information. Other controllers work similarly but their tick method implementations differ regarding to their algorithms.

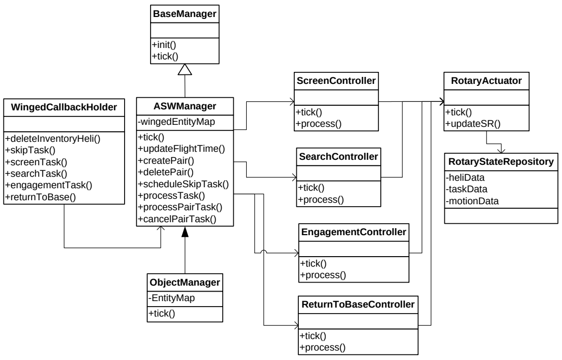

Figure 6 - Inventory Rotary Wings and Paired Tasks

5.4. Sensor Fusion and Tactical Reflexive Avoidance Behaviors

In this study, each entity of a scenario owns several sensors through which the entity may acquire detections of an enemy entity or its torpedoes in water. With the aid of a specific derived VR-Forces controller, namely the avoidance controller, a surface platform can halt performing its current task and make certain maneuvers in order to avoid from such a detected threat.

For entities that have been assigned in the same group, a sensor fusion manager module obtains all the sensor detections of these entities and inputs them to a sensor fusion algorithm in order to calculate an approximate location for the enemy entity or its torpedoes. After that, an avoidance manager module checks whether such a location has been detected for each entity group. For entity groups with a valid enemy detection, avoidance controllers of each entity in the group are evoked with necessary parameters.

Sensor fusion result for each entity group is also published in the simulation since it can be benefitted from while assigning certain search or engagement tasks. Once a corresponding location approximation can no longer be evaluated due to loss of detections in the sensors, a special point, namely the datum point, is displayed on the tactical screen for a predetermined amount of time.

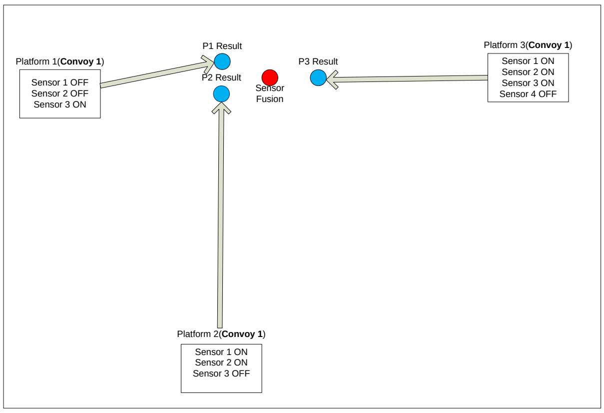

Figure 7 - Genereal Sensor Fusion Presentation

In Figure 7, there three types of platforms belong to same convoy (**Convoy 1**). Each platform has different types of platform and some of them ON and some of them are OFF. According their own algorithm they have their own sensor detection results. We implemented a algorithm that have input all platforms' sensor detection results (for this example, platform 1 result, platform 2 result and platform 3 result) and output *sensor fusion point*.

6. SIMULATION RESULTS

In previous section we explained in detail different types of modules we used. In this section, we illustrate some sample result about developed modules.

In Figure 8, we show an example those 2 platforms given search task. Originally those platforms have leader. Leader gives them a search task in geometrical region. According to their assigned region platforms first reach that region and then follow a pattern (as shown in figure, like 8). In the middle of area there is a region that forbidden for platforms. Each platform knows that rule and when it comes to border that area, otomatically escape from that area. But same time it knows its own original tasks (searching for this example) and finds new path to reach its search area. For that reason, Platform 2 follows sharp path, but Platform 1 follows smooth path. Because, Platform 1 has no overlap with with forbidden zone but Platform 2 has.

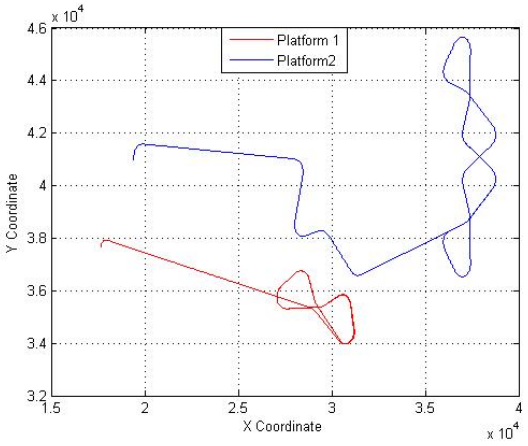

Figure 8 - Search algorithm behaviour for 2 different platforms

In Figure 9, there are 2 different platform that given engagement task. In normal situation platforms have an engagement mission pattern. They first move to related path and then they follow a pattern as shown in figure. As we mentioned before they are convoy members and they have communication with leader. Leader have ability to give orders them any time in simulation. In this example we see that Platform 1 leaves from its original path, attack target (shown as diamond) and then come back to his original path again.

Convoys have common information that shared among members. In this example, convoy has no sensor detection in the beginning of simulation. After some time someone in convoy detected a target and shared in information pool (sensor fusion). Now, all convoy know where target is detected. Leader might give most suitable members to attack. So it attacks the target (Platform 1 in figure).

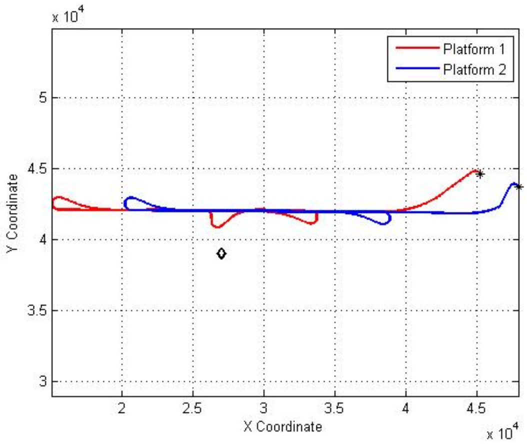

Figure 9 – Engage algorithm behavior for 2 different platforms

Initially platforms have 0 speed. Until reaching the related path, they increase their speed to max and then decarease the pattern speed. Except turning bahaviours they follow the pattern speed. But, as shown in Figure 10. Platform 1 has increased its speed to max again because of attacker phase. After attacker phase it sets its own pattern speed.

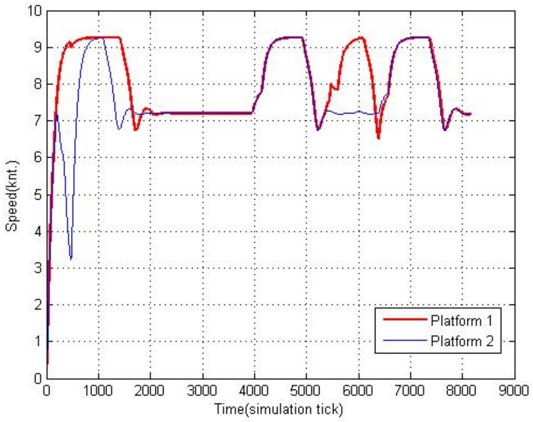

Figure 10 - Speed change during simulation

7. CONCLUSION

In this paper, we explained our training based simulation architecture and Anti-Submarine Warfare concept. We explained in detail our software design about ASW. We illustrate some simulation results about our work.

The whole design is integrated into the component architecture of VRForces, which provides a framework for developing Computer Generated Forces (CGF) applications. Also, commercial toolkit independent simulation components that specialize on algorithmic behaviors are integrated into the commercial toolkit based CGF and GUI applications via developing original control architecture.

There are lots of task that platforms have ability to perform. While performing their original task, they perform some reflexive behaviors such as land avoidance, collision avoidance and step aside maneuver. Our results represent that all single and convoy task successfully achieved as high fidelity.

REFERENCES

VR-Forces Developer's Guide, MAK Technologies, 2006. Chapter 2, The VRForces Simulation Engine, Chapter 5,Creating New Components, Chapter 10, The VR-Forces Remote Control API.

VR-Forces The Complete Simulation Toolkit. MAK Technologies. Available from http://www.mak.com/products/vrforces.php [accessed 11 April 2011]

Aldogan, D., Haklidir, M., Senyurek, L., Koksal, S., Eroglu, O., Akdemir, C., Franko, S., Tasdelen, I., Akgun, S., 2009. *A General Approach To High Fidelity Modeling, Simulation and Control Of Tactical Entities And Implementation In a Commercial Compter Generated Forces Toolkit.* EMSS 2009

Aldogan, D., Haklidir, M., Eroglu, O., Franko, S., Timar. Y., Guven, A. F., Senyurek, L., Genc, H.M., 2013. *Denizaltı Savunma Harbi Modelleme ve Simulasyonu*. USMOS 2013

Haklidir, M., Aldogan, D., Tasdelen, I., 2008. *High Fidelity Modeling and Simulation of Surface Platforms in a Computer Generated Forces Toolkit*, SUMMERSIM 2008, Edinburgh, Scotland.

Franko, S., Koksal, S., Haklidir, M., 2009. *Modeling, Simulation And Control Of Rotary Wing Platforms In A Computer Generated Forces Toolkit*, The Industrial Simulation Conference ISC 2009, Loughborough, United Kingdom, 1-3 June 2009

Haklidir, M., Guven, A. F., Eroglu, O., Aldogan, D., Tasdelen, I., 2009. *High Fidelity Modeling and Simulation of Submarine in a Commercial Computer Generated Forces Toolkit*, The Society for Modeling & Simulation International [SCS] 2009 Summer Computer Simulation Conference (SCSC'09), Istanbul, Turkey, 13-16 July 2009.

Senyurek, L., Koksal, S., Genc, H.M., Aldogan, D., Haklidir, M., 2008. *Implementation Of Fuzzy Control For Surface Platforms In A Computer Generated Forces Toolkit*, International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA08), Vienna, Austria, 10-12 December.

Pratt, D.R. (1996). *Next Generation Computer Generated Forces*. Proceedings of the Sixth Computer Generated Forces and Behavioral Representation. s. 3-8.

Koçabaş, Ş., Öztemel, E.,1998, *Harp Oyunlarında Yapay Zeka Uygulamaları* TSK-MODSİM, Modelleme ve Simülasyon Konferansı, 1-3 Nisan 1998. Kara Harp Okulu, Ankara.

Michael R. B., Henrik, S., Paul M. N., John J. L., 2010. *Nested Autonomy for Unmanned Marine Vehicles with MOOS-IvP*. Journal of Field Robotics 27(6), 834–875 (2010) Published 2010 Wiley Periodicals, Inc.

Krishnamurthy, P., Khorrami, F., 2011. *A Hierarchical Control and Obstacle Avoidance System for Unmanned Sea Surface Vehicles*. 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, Aralık 12-15, 2011

Aveek K. D., Rafael, F., Vijay K., James P. O., John, S., Camillo J. T., 2002. *A Vision-Based Formation Control Framework*. IEEE Transactions on Robotic and Automation, Vol. 18, No. 5, Ekim 2002

AUTHORS BIOGRAPHY

Deniz ALDOGAN is a senior researcher in The Scientific and Technological Research Council of Turkey, BILGEM (TUBITAK BILGEM). She is also a PhD student in Computer Engineering Department of Istanbul Technical University. She received her M.S. and B.S. degree in Computer Engineering from Middle East Technical University. Her main areas of research interest are modeling, simulation and artificial intelligence. She has published several papers in these areas.

Mehmet HAKLIDIR is a chief researcher in The Scientific and Technological Research Council of Turkey, BILGEM (TUBITAK BILGEM). He is also a PhD student in Control Engineering Department of Istanbul Technical University. He received his M.S. degree in Mechatronics Engineering from Istanbul Technical University and B.S. degree in Mechanical Engineering from Istanbul University. His main areas of research interest are modeling, simulation and control of dynamic systems, robotics, artificial intelligence. He has published several papers and proceedings in these areas.

Levent SENYUREK (1975) is a systems & control engineer, who has experience in RF systems and training simulations. He has worked in/directed RF system projects which aim to develop wearable monitoring systems. He involved in simulation projects to design control and AI models for simulated platforms. His research interests are wearable monitoring systems, fuzzy logic control and artificial intelligence. He has several national and international publications and 1 patent on these topics.

Yasemin Timar has been practicing software engineering, presently titled as a chief researcher in The Scientific and Technological Research Council of Turkey, BILGEM (TUBITAK BILGEM) since 2002. She received her B.S. degree from Computer Engineering Department of Middle East Technical University (METU) and M.S. degree from Computer Engineering department of Bogazici University. She is also a Ph.D. student in the same department. Her research focuses on modeling and simulation technologies, computer vision, artificial intelligence and cognitive science. She has several proceedings and papers published in these fields.

Semuel FRANKO is a senior researcher in The Scientific and Technological Research Council of Turkey, BILGEM (TUBITAK BILGEM). He is also a PhD student in Control Engineering Department of Istanbul Technical University. He received his M.S. degree in System Dynamics and Control Engineering from Istanbul Technical University and B.S. degree in Mechanical Engineering from Istanbul University. His main areas of research interest are modeling, simulation, artificial intelligence and data mining.

Omer EROGLU (1983) is a senior researcher and software developer expert in The Scientific and Technological Research Council of Turkey, BILGEM (TUBITAK BILGEM). He is also a PhD student in Computer Science Engineering in Istanbul Technical University. He received his M.S. degree in Data Distribution Management in High Level Architecture and B.S. degree in Computer Science Engineering from Yildiz Technical University. His main areas of research interest are modeling & simulation, distributed systems, artificial intelligence, robotic and software development.