Proceedings of the 33rd Chinese Control Conference
July 28-30, 2014, Nanjing, China

The design and optimization method of near space intelligent target generator

HAN XiaoLong[1,2], CHEN Chen[*1,2]

1 School of Automation, Beijing Institute of Technology, Beijing 100081

2 Key Laboratory of Intelligent Control and complex decision-making, Ministry of Education. Systems, Beijing 100081

E-mail: xiaofan@bit.edu.cn

Abstract: This paper presents the design and optimization method of near space intelligent target generator to simulate the physical characteristics of the near space vehicle. Combined with High Level Architecture distributed simulation technology, a common, repeatable and verified platform for the near space vehicle has been provided. This method used 3D modeling software Creator and 3D visual rendering software Vega, two-dimensional map and three-dimensional vision were constructed to form a simulation environment, which enhanced the authenticity of the simulation. Based on particle swarm optimization, the intelligent path planning study of near space vehicle was conducted in this environment to make up for the inadequate intelligence of traditional target generators.

Key Words: Target Generator, Near space, HLA, Virtual reality, Artificial intelligence;

1 Introduction

Near space usually defined as the space region between 20km and 100km. In such ranges, vehicles can effectively avoid detection of radars, as a result of avoiding the vast majority of ground fire attack, improving the efficiency of the military investigation and the accuracy of the attack, which have immeasurable potentials in the information warfare and electronic warfare in the future[1]. In recent years, the development of near space vehicle promotes the technological innovation, and leads to the needs of the military development, which has attracted worldwide attentions[2].

Currently, the study of the near space vehicle still has several problems, including the very high research funding, the lack of the vehicle simulation characteristics, infeasible repeatable verification and so on. There is no target generator for the near space, which can replace the realistic physical simulation system and provide conditions for the study of near space vehicle. Constructing a suitable target generator and applying it to the simulation platform can effectively solve the above problems.

Target generator can simulate the physical characteristics of the near space vehicle and the complex environment it needs. Combined with HLA, it can provide a platform to verify the characteristics of near space vehicle. However, current target generator still has problems in the authenticity of simulation and the intelligent of the target. Target generator using virtual reality technique can construct a three-dimensional virtual simulation environment, which effectively improves the authenticity of the simulation, so that the participants will be indulged in the environment. Using artificial intelligence technique can add intelligent behaviors to the target, and improve the intelligence of the target.

This paper aims to construct an intelligent target generator specific to near space, which can simulate the characteristics of near space vehicle and the environment of near space. Combined with HLA, it can provide a simulation platform for near space vehicle to replace the original target generator. On this basis, this paper proposes the application of artificial intelligence technique to target generator, and tries to use particle swarm optimization algorithm for the intelligent path planning behavior of the

*This work is supported by National Nature Science Foundation of China under Grant, No. 61203181, Beijing. Outstanding. Ph.D. Program Mentor Grant, No. 20131000704, the foundation for innovative research group, No. 61321002, the National Science Fund for Distinguished Young Scholars, No60925011, the Yangtze River scholars team of Ministry of Education, No IRT1208, Chang jiang Scholars Program.

target. In order to enhance the authenticity of the simulation, the entire simulation process will be combined with virtual reality technique, using the combination of two-dimensional map display methods and three-dimensional visual display methods.

2 Construction of the Virtual Scene

2.1 Creating and Optimizing Models

The construction of target models can be accomplished by modeling software such as Multigen Creator, 3Ds Max, Maya, etc. In this paper, we put the existing 3D models into 3Ds Max, and then the models are transformed into the format FLT, which can be used in Creator. Target models are shown in Fig. 1.

According to the requirements of the project, the entire planet has been taken as the scene, and environmental settings can apply texturing techniques which significantly reduces the number of polygons required in the three-dimensional scene to improve the effectiveness of simulation. Target environment model created by this method is shown in Fig. 2.

When modeling, simulation models often encounter issues such as the over complexity of the model and time-consuming in loading. These problems are due to the situation that the simulation model is too fine, and there are too many polygons in the model. Of course, if the simulation model is too simple, it would cause the shortage of the simulation authenticity. How to find a balance between these, utilizing few polygons to show a rich and varied performance simulation environment, is a problem faced with visual simulation. This article uses the following methods to resolve this issue. Frist of all, remove the extra polygons. Extra polygons refer to polygons that are not always showing up during the simulation process, which usually include polygons covered by the scene and polygons inside models that do not need to be displayed. Second, delete invisible polygons, including the polygons of the back mountains, the invisible polygons inside buildings, etc. They are safely removed from the model library. Last, using LOD technique provided by Creator. LOD technique can turn model contains more details into a simple one. Using this tool, you can easily achieve the creation of the underlying model[3].

2.2 Rendering the Three-dimensional Scene

After the models are created, we can load the simulation environment and the models by using three-dimensional visual rendering software Vega. Then make full use of Vega to optimize the whole simulation environment.

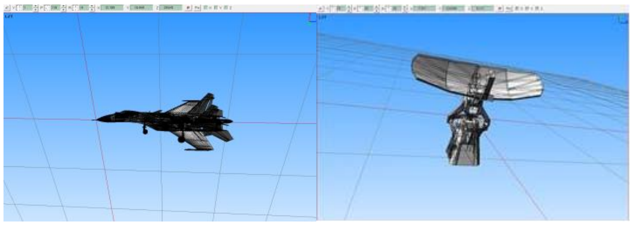


Fig 1: Three-dimensional model

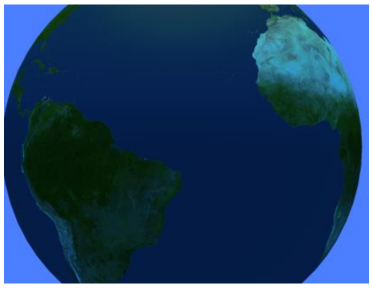


Fig 2: Virtual environment

Using LynX interactive interface provided by Vega can effectively optimize the three-dimensional scene. LynX is an application-defined configuration and a dynamic previewed tool which is convenient and friendly for the user[4]. It can define the attributes and build the relationships between the attributes of all the elements in this 3D scene. You can set parameters in real time and preview the three-dimensional display consequences. Finally, generating ADF file can be utilized to expand the functionality of Vega.

In this paper, LynX has been used for loading the models, setting location information, setting viewer mode, adjusting movement patterns, changing environmental effects, etc. At last, files stored as Adf, which will be combined with MFC to achieve more extensions of Vega, such as window regulator, coordinated transformation, and followed perspective. The flowchart of the development for Vega is shown in Fig. 3.

This article uses Vega to render the whole scene at any time in the simulation process. Vega is used to design and to produce special effects, just like the vehicles' flying smoke effects, which enhances the authenticity. Vega can change viewing mode of the observer, can realize multi-channel observation, adjust the light source, change the time in the simulation, etc. Using Vega, it is allowed for user to view the scene from any angle, and we can view different objects in the same window, making the simulation more flexible.

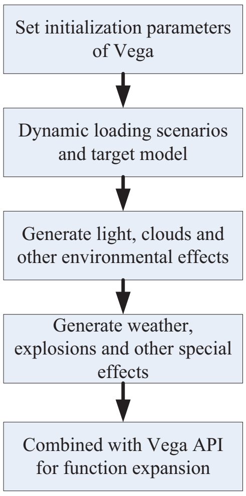


Fig. 3: Flowchart of Vega developing progress

3 The Analysis of the Characteristics of Vehicles

This article focuses on the existing four kinds of near space vehicle, including stratospheric airship, high-altitude UAV, reconnaissance aircraft, and hypersonic aircraft[5]. Different kinds of near space vehicle have different parameters and play different roles in warfare. For instance, high-altitude reconnaissance aircraft can be a platform for the investigation, and large stratospheric airship can be used as air supply platform. To solve these problems, combined with the reality of the situation, the mechanical properties of different vehicles should be considered. Different properties and parameters for each kind of near space vehicle have been designed in this paper, and four kinds of three-dimensional target model have been built by Creator. In this case, we can make the simulation more realistic, and improve the reliability of the simulation.

The main parameters of the vehicle include speed, acceleration, flight attitude, radar's reflection areas under different attitudes, the minimum turning radius, three-dimensional models of vehicle, etc. Selecting the above parameters determines the credibility of the target generator. The more accurate the parameters are, the more authentic the simulation will be. Suitable parameters allow the target to move in line with the actual object.

Using hypersonic aircraft as an example, the method to design parameters of the vehicle will be explained below.

The masterpiece of the contemporary near space hypersonic vehicle is a model developed by the United States, which is still being studied and improved. This kind of hypersonic vehicle can attack anywhere in the earth in one hour. It has absolute advantages in the following three aspects. Firstly, it is so fast, usually more than six times the speed of sound, that it can reach between 6 to 6.5 Mach. Secondly, it operates in very high altitude, and most of the time it flies in the sub-space. Thirdly, it is very small, which has the same size as the AAM.

The main parameters of this kind of hypersonic vehicle include[6]: Mach Number: 5 ~ 25Ma, Flight Altitude: 20～100km, Geometry: 1～20m, Motor Overload: 2～4G, RCS: 0.1～0.01m^2.

4 Realization of Target Intelligent Behavior

Nowadays, a representative target generator is one with virtual reality. However, this kind of target generator can only simulate simple motions like a single path of uniform linear motion, due to its lack of intelligence, which cannot satisfy with the complexity and variability of the simulation. As a result, with less truthful information of the target, the simulation is less reliable. In order to settle the problem, a particle swarm algorithm is adopted for path planning[7,8].

In order to simulate the behavior of the near space vehicle, the three-dimensional space should be gridded[9]. As a result, the flying and non-flying regional areas are distinguished, and the path can be planned. The gridding method is based on gridding the direction in longitude latitude and altitude reasonably into M part. Since the starting point and finishing point are known, the total path planning points are D=M-1.

Then the fitness function J is designed. In this paper, path planning provides a shortest path considering not being detected by the radars, the real-time environmental information as well as the feature of the near space vehicle. The fitness function is given as followed which is combined with all the factors comprehensively.

$$J = \sum(\omega_1 T + \omega_2 M + \omega_3 H + \omega_4 V + \omega_5 P) \qquad (1)$$

where $\omega_1, \omega_2, \omega_3, \omega_4, \omega_5$ are constant values, representing the importance of each part, which are all 1 here to show the same importance. T represents various threats arising from radar to the near space, that is, the avoiding space for the path. M represents the real-time environmental information, since the near space vehicle generally flies over a certain distance. With different environment information, some of them are even not suitable for flying. H represents the altitude of the vehicle operating. Low flying is generally better since the vehicle can execute the attacking orders or scouting orders. V represents the labels of the path in order to shorten the flying path. P represents the moving features of the near space vehicle, mainly considered the minimum turning radius of the near space vehicle.

Finally, designing and adopting the particle swarm algorithm with varying inertia factors can find the optimal solution after iterations. The iteration formulas are as follows:

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 \cdot rand() \cdot (p_{id} - x_{id}(t)) + c_2 \cdot rand() \cdot (p_{gd} - x_{id}(t)) \quad (2)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (3)$$

$$\omega = 0.8 - 0.4\frac{t}{MaxDT} \quad (4)$$

where ω is the inertia factor, which is a time-varying value. $MaxDT$ is the maximum iterations, t is the iteration time. At the beginning of the iteration, the value of ω is relatively large which provides a large searching space. As ω is decreasing by t, the searching becomes more precise. c_1 and c_2 are learning factors which are normally constant values, providing a self-learning ability and the ability to learn from the best particles. As a consequence, every particle moves to the direction based on the personal best and the global best position.

The process and the parameter of the algorithm are as follows:

Step 1. The searching space for the path planning is set by 20km along longitude as x axis, 20km along latitude as y axis and 20-100km in altitude as z axis. Each path has two points in y and z axis. The particle swarm is random initialized with N swarms. N is set generally from 10 to 40. In this paper, it is selected as 40 due to the huge searching space. The learning factors c_1 and c_2 are 1.4. The maximum iteration $MaxDT$ is 1000. The starting point is (0, 20, 20) and the finishing point is (20, 0, 100).

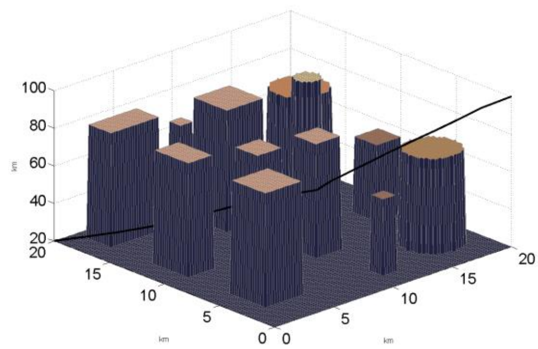
Step 2. In order to judge the quality of the particles, a fitness function J for the path planning is introduced.

Step 3. Comparing with the fitness of the particle, Pbest represents the best position of one particle and gbest represents the best position of the whole particles.

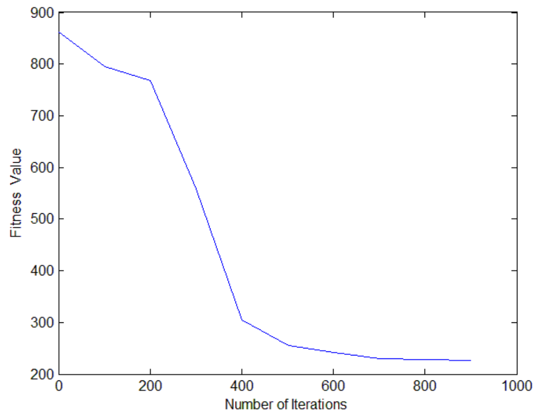
Step 4. Adjusting the inertia factor based on the velocity and the position of the particles according to the iteration formula.

Step 5. Repeating Step 2 to Step 4, the iteration is finished when the best fitness is reached or the maximum iteration is met.

The simulation result is in Fig. 4 programmed under MATLAB to verify the path planning of the target generator. The starting point is (0, 20, 20), the finishing point is (20, 0, 100). The rectangle represents the sensing range of the radar, which is set according to a certain kind of radar. The cylinder represents the area which is not fitted for flying, and generally refers to the area where equipments are not able to work affected by the atmospheric density, wind, and solar radiation. The black line refers to the route after path planning.



a) Path planning result



b) Fitness function of path planning

Fig 4: Matlab simulation results

After 50 independent simulations, the maximum optimal fitness Pbest is 226.2370, the minimum is 225.3127, with less error which is in line with the simulation need. As a result, this method is acceptable, which adds intelligence to the near space vehicle by embedding the algorithm in the target generator. In this paper, the programming environment is under Visual C++ platform, and the target generator is built on it and verified in Section 5.

5 Realization of the System based on HLA

The target generator built in this article is part of a simulation system which is aimed to detect near space vehicles. This simulation system use HLA to achieve integration target generator with other members, then the system can simulate whether the sensor network can detect near space vehicles or not.

The construction of distributed simulation systems with HLA usually requires developers to consider the simulation requirements comprehensively. Then developers should design each federate based on the parameter they need[10]. As a simulation system for detecting near space vehicles, there are five federates in this system, including target federate, environment federate, information processing federate, radar federate and management federate. Target federate in this system is the target generator, aiming to create targets and show the entire simulation environment. Radar federate is designed to detect the targets and track how they move. Management federate is a monitor while the system is running. In this system, each federate can communicate through RTI(Run Time Infrastructure), and make use of Oracle9.2 to record data inside this system.

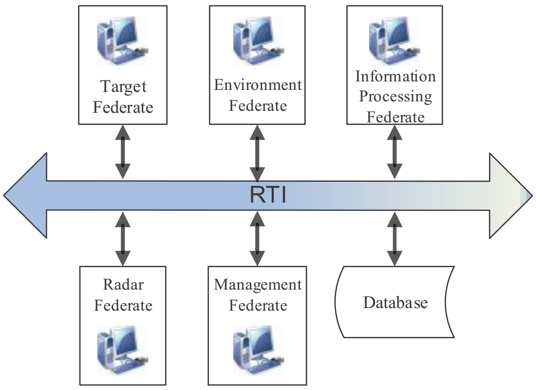


Fig. 5: Structure of the simulation system

For the entire simulation system, as target federate, the target generator needs to exchange data with other federates. There are two types of data exchange with HLA. Data should be classified into object class or interaction class. Data of object class exchange at any time during the simulation, while data of interaction class can be sent or received if other federate need it. Target generator sends its attributes to other federates as object class, including quantity, type, position, velocity, acceleration and so on. Other federates receive these data and execute their own mission, and radar federate uses these data to detect the targets and track them. While target generator also needs to receive interactive information of other federals, such as starting point terminal point, radar's position, path information and so on.

Target federate can load map information then transform the coordinate into the map to the actual longitude and latitude. There are two modes for the occurrence of target. One of them is set by yourself, you can select the starting point and the terminal point by clicking points on the map, and you can design the path for each target, the type of target and other parameters. The other is set by reading the database. All the simulation parameters have been written in the database before simulation, which are ready for use.

This target generator is different from the traditional target generator. Three-dimensional scene and path planning algorithms have been embedded, so that this kind of target generator makes the simulation more accurate.

To achieve the above function, the design of this target generator includes the simulation application module, the RTI Interface module, the RTI Ambassador module, the Federate Ambassador module, the Simulation Object Model, etc.

The design of target federate contains the following steps:

Step 1. Design parameters rationally and selected as the interactive class or object class though OMDT.

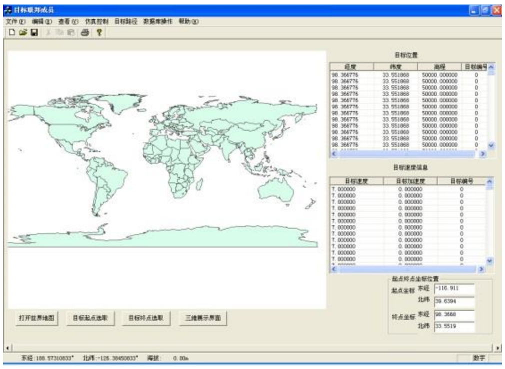
Step 2. Design Vega thread. Use the ADF file generated by Vega to expand its functions. Combined ADF file with VC++, and make full use of the API of Vega to render the environment.

Step 3. Design HLA thread. The development of HLA thread includes creating federate, joining federation, registration, publish, subscribe, etc. This step can make the target federate connect with other federates.

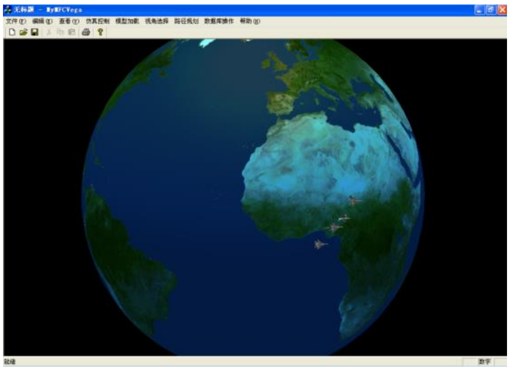
Step 4. Add the characteristic of the near space vehicle into HLA thread and Vega thread. Set position of targets, velocity of each target, RCS and other parameters the target federate needs.

Step 5. Path planning algorithm should be added into this target federate. The PSO path planning algorithm design in section 4 is transferred into VC++ platform.

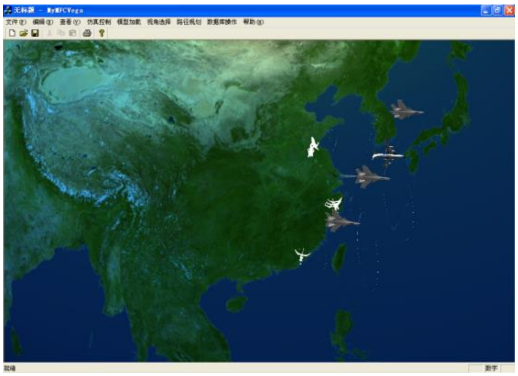
Through the above steps, a new kind of target generator has been made, which is more intelligent and more authentic. Fig. 6 shows how the target generator works during the simulation process.



a) Interface of Map information



b) Three-dimensional distant vision of simulation process



c) Three-dimensional close vision of simulation process

Fig. 6: The simulation process

This kind of target generator can provide repeatable simulation platform for the study of near space vehicle, such as testing and verifying the radar's detection results under different RCS, testing the impact of different environments on the near space vehicle, and finding out how the radar network tracking the target, etc.

6 Conclusion

In light of the development of the near space vehicle, this paper puts forward a design and optimization method of near space intelligent target generator, and constructs the near space intelligent target generator based on HLA, which can replace the original physical target generator. Therefore this design decreases the cost of large-scale simulation system and avoids artificial operation risk.

In this study, three-dimensional modeling software Creator and three-dimension visual rendering software Vega are used to create a 3D simulation environment, and adopt the intelligent path planning behavior into the near space vehicle with the particle swarm optimization algorithm. Combined with the HLA distributed simulation technology, this method integrates target generator into the simulation environment and constructs the target federal members. Finally, the simulation shows that the target generator can effectively overcome the drawbacks of the traditional target generator in the aspects of simulation authenticity and target intelligent, providing a simulation platform for testing the characteristics of near space vehicle with a promising application prospect.

References

[1] Stephens H. Near-space[J]. *Air Force Magazine*, 2005, 88(7): 31.

[2] W. Huang, S. Luo, and Z. Wang. "Key techniques and prospect of near-space hypersonic vehicle." *Journal of Astronautics* 5 (2010): 003.

[3] G. Hong, H. Li, J. Mu. The Simplification of the 3D Model Based on Creator[J]. *Computer Simulation*, 2004, 21(1): 57-58,115.

[4] P. Yang, X Qiu, S Huang, et al. Development of Virtual Reality Application with Vega[J]. *Computer Simulation*, 2005, 5: 165-168.

[5] Y. Li, L. Zhi, and H. Shen. "Analysis on Development and Application of Near Space Vehicle [J]." *Journal of the Academy of Equipment Command & Technology* 2 (2008): 017.

[6] J. Dai, J. Cheng, and R. Guo. "Research on Near-space Hypersonic Weapon Defense System and the Key Technology." *Journal of the Academy of Equipment Command & Technology* 21.3 (2010): 58-61.

[7] Y. Qin, D. Sun, N. Li, et al. Path planning for mobile robot using the particle swarm optimization with mutation operator[C]. *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on. IEEE*, 2004, 4: 2473-2478.

[8] Foo J, Knutzon J, Oliver J, et al. Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization[C].*11thAIAA/ISSMOMultidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia. 2006.

[9] G. Tan, and G Liu. "Global optimal path planning for mobile robots based on particle swarm optimization." *Application Research of Computers* 24.11 (2007): 210-212.

[10] Dahmann, Judith S. "High level architecture for simulation. "*Distributed Interactive Simulation and Real Time Applications, 1997. First International Workshop on. IEEE*, 1997.