Guided bomb release planning based on Monte Carlo in a distributed virtual environment

Z. Wang
flair@126.com
Institute of Automation, Chinese Academy of Sciences
Beijing
China

J. Wang
Department of Computer Technology, School of Telecommunication Engineering
Beijing Vocational College of Electronic Science and Technology
Beijing
China

ABSTRACT

Air-to-ground strike has become one of main forms of modern warfare, and the guided bomb release is an important and key part of the attack. The guided bomb release planning aims to accomplish a precise target hit in enemy's region while guarantee the pilot's safety. We propose a robust Monte Carlo method by taking error perturbations into consideration, comparing to the traditional sequential quadratic programming method under extreme conditions. At the same time using a distributed virtual physics environment we can obtain much more detailed realism relative to the conventional simulator running on a single machine. The experimental results verify that Monte Carlo methods can improve hit probability and weapon efficiency significantly. Furthermore, the 3D visualised environment plays a very important role in training pilots, so this simulator will decrease the cost and time requirements of physical experiment that are not always compatible with strict military task.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press
Paper No. 3877. Manuscript received 11 July 2012, revised version received 17 October 2012, accepted 1 November 2012.

INTRODUCTION

As a vital and kernel part of air-to-ground strike, GBRP (guided bomb release planning) serialises the whole attack sequences to optimise release for maximised hit probability and damage effect as well as pilot's security. Generally speaking, in the process of GBRP a tiny perturbation to the theoretical waypoint resulting from predefined parameters will probably make the entire project unreachable, or unfeasible, even irremediable. In order to increase hit probability we are trying to set aside redundant margin for the theoretical waypoint with a robust effect. And to this end we designed a DVE (distributed virtual environment) considering physics and stochastic processes to emulate uncertainties and perturbations in a GBRP.

In 1980, the US Army started research on the first generation of CAMPS (computer aided mission planning system) based on expensive mainframe and UNIX platforms. In 1983 the MSS (mission support system) was deployed for specified aircraft with data loaded via DTD (data transfer device) or DTC (data transfer cartridge), though pilots were faced with 2D GUI (graphic user interface) and difficult semi-automatic operations[1]. With the development of the third generation of MPS, the WRP (weapon release planning) evolved into a dedicated component with high integration and refinement, usually including functions as follows,

- Manoeuvre planning for target region
- WRP parameterisation
- Projectile trajectory generation and display
- WRP error analysis
- Maximised damage coverage
- WRP waypoint computation

Essentially, WRP should not be treated as a decision making system, but as a set of assisting intermediate computing software. References 2 and 3 provide an overview of the status quo of simulation and computation methods for various guided bombs, and an aircraft route planning algorithm is shown. The release methods and effectiveness evaluation algorithms can be seen in Refs 4 and 5. In addition, there are some other WRP cases shown in Refs 6, 7 and 8. On the whole, these systems have no distinct characteristics relative to the WRP of US Army.

During the corresponding period of WRP, the United States DoD (Department of Defense) developed a protocol for real-time interactive battle DVE called SIMNET[9]. Recently, there has been much interest and research in the area of DVE, and the existing DVEs were mainly applied to military simulations[10,11], social VW (virtual worlds) and MOGs (multiplayer online games). These different applications running in addition to a DVE may impose different requirements. However, there are some fundamental requirements that are common to most DVE-based applications, namely:

- **Consistency**: In order to have meaningful user experiences in a DVE, each client has to perceive the same state of the virtual environment. To do so, the DVE architecture needs to ensure an almost simultaneous distribution of the virtual environment state to the active clients. However, a number of issues such as latency traffic on the Internet or unpredictability of the client demand make consistency in DVEs a very challenging task[12,13].
- **Responsiveness**: User experience is positive when they perceive changes in the virtual environment state, quickly enough e.g., neighbor movements or the creation of a new entity. Thus, clients need to be informed in real-time about any action performed as well as its consequence. Different DVE-based applications have different responsiveness requirements, e.g., 300ms for MOGs and 1s for VWs[14].

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

- **Scalability**: DVE aims at interconnecting millions of clients within a shared environment, i.e., DVEs should ideally scale to an unlimited number of participants. Precisely, a DVE architecture is required to sustain a theoretically unbounded growth in the client demand with a sub-linear impact on the user experience[15].
- **Persistency**: Contents within a DVE may need to exist persistently. For instance, information about a client score as well as user-generated entities should not be lost during the evolution of the virtual environment[16].
- **Security**: In computer science, security is the prevention of risk or danger such as the exposure, destruction or alteration of personal information[17]. Applications running on top of a DVE have created complex virtual economies as well as strong competitions among clients. Therefore, a DVE architecture needs to guarantee security in order to ensure fairness among its clients as well as private information. Nevertheless, cheating has recently emerged as a major security concern in DVEs. DVE clients cheat in several ways, e.g., delaying packet transmission or modifying the client instance[18].

Recently, military simulation research evolved into the HLA (High Level Architecture)[11] IEEE standard. HLA does not specify any particular technology, but simply provides a general architecture for distributed simulation, e.g., training operators of tanks, aircraft and ships, and to use multiple types of vehicles in the same simulation. Thus, HLA can be used for military simulations as well as for MOGs. While in November 2004 World of Warcraft (WoW) became the most famous MOG counting more than 15 million players, i.e., the largest players community in the history of MOGs. The main reason of WoW success is its attention to new-players: WoW makes very easy for its players to learn the game. Similarly to EverQuest, WoW leverages a C/S architecture where the game-play is replicated over different servers. With respect to both load balance and latency traffic, WoW achieves a distinct advantage over HLA or VW.

Guided bomb release should be executed in a distributed and collaborative environment with antagonists, and in order to optimise planning globally we should also consider kinds of uncertainties like threat situation, weapon performance, tactics requirements, weather condition, regional shelter, and target characteristics etc. Because of the execution risk and cost, based on Monte Carlo methods, we design an interactive DVE as a virtual test-bed to improve GBRP robustness. The virtue of our methodology resides in that the residual traces left by using DVE are stable, and the planner can examine them at leisure while the pace of action can be varied at will. Savings in cost and (preparation) time therefore also have another, indirect, effect: this DVE provides the possibility to perform learning sequences in which one takes account of previously unanticipated results. In this way, this DVE allows testing hypotheses that are not constrained by the logical bounds of the premises on which the simulation is based. It thereby enables non-conservative plan, which is important to achieve any distinctive new plan.

This paper is organised as follows. Section 1 is an introduction and provides an overview of related works. Section 2 describes this system architecture, and the physics models for the aircraft and guided bomb are presented in Section 3. In Section 4 the whole release process is serialised into tactical sequences, and the optimised solution is obtained. For network communication, parallel to the popular military simulation and multiplayer games, a set of collaborative frameworks are designed in Section 5. An example of implementation and an analysis of simulation results are described in Section 6. Finally, in Section 7 the conclusion is given.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

2.0 A SCALABLE AND EXTENSIBLE SYSTEM ARCHITECTURE

As we can see in Fig. 1, this DVE uses the ECA (entity-component-action) model as a basis for constructing scalable and extensible environments. This model was adapted from contemporary game engine architectures. Entities are unique identities, with no data or typing. They aggregate components, which can be of any type and store arbitrary data. Applications built using this client framework can add their own components so they have the data they need for their own functionality. The code that handles the data exists in preinstalled custom modules or in scripts loaded at runtime as a part of the application data. To get a matching server counterpart where the environment is entirely built with entity components, the server component was added to the client codebase and a new protocol without application-level assumptions. This DVE consists of both client and server executables.

The DVE platform provides basic functionality for all ECAs: persistence, network synchronisation among all the participants via a server framework, a user interface for manipulating components and their attributes, and, eventually, will support security. In addition, this DVE introduces the concept of entity actions, a simple form of remote procedure call.

The reason that this architecture is used is the set of customised GBRP solution requirements, which users might want to implement themselves. In these cases it cannot be assumed that the users implementing the new features have deep knowledge of the insides of the whole GBRP system. Spending a significant amount of time studying the whole of the existing system should not be a requirement for implementing new features. A GBRP use case can be seen in Fig. 2, where each column in the diagram represents a list of identical components, each row can be thought of as representing an object which is simply the sum of its parts, and each entity is comprised of a collection of components. The components themselves can be treated as being independent of the objects they make up.

In addition to having virtual objects extensible, it should be possible to replace or add new functionality to the framework itself based on plug-in, where all major features that are not distinct to a single virtual object are implemented as plug-ins to the system. It should be possible for the user to disable and enable new plug-ins at runtime.

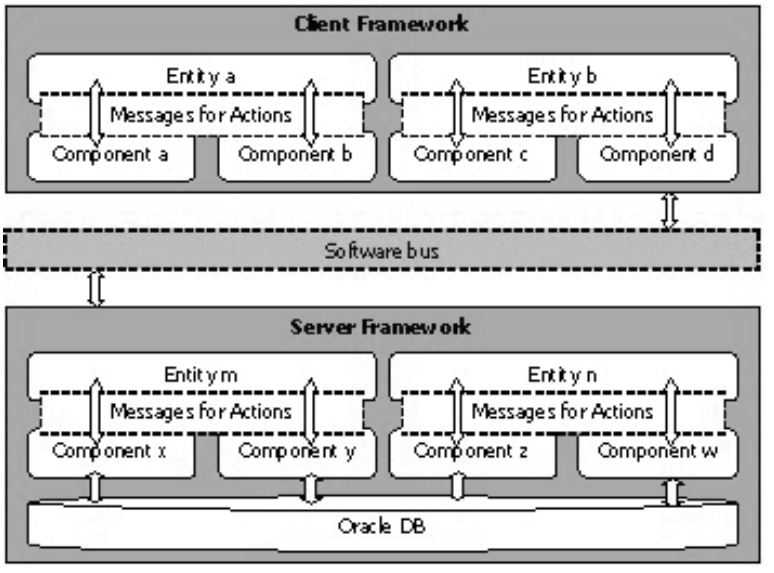


Figure 1. An abstract architecture of this DVE based on ECA model.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

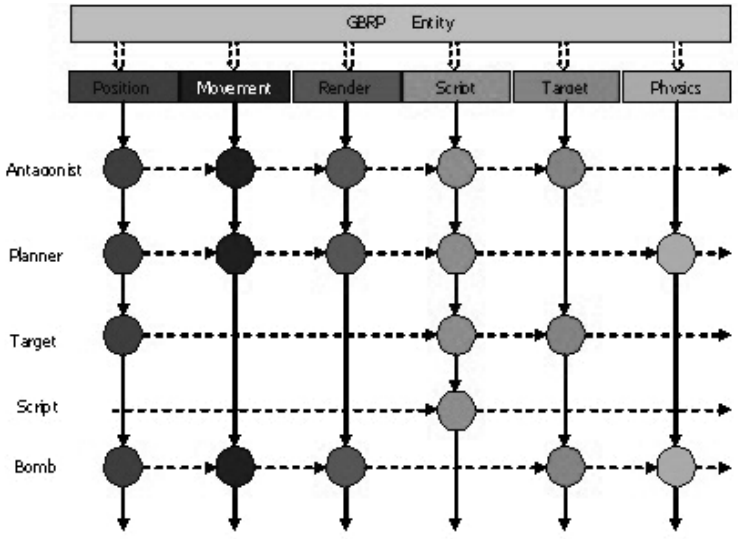


Figure 2. A GBRP entity with virtual object composition using components.

In the present models, physics in GBRP is performed primarily by the server. There are client functions that have some relation to physics, specifically the motion interpolation of the virtual objects. Besides, there are visual effects implemented by some clients that are often colloquially thought of as physics-related. However, the really important observation is that presently only the server is able to initiate a collision, and cause objects to change behavior due to such collisions. While the script or in other words client-side scripting is the execution of sandboxed Python on the client, not the server, avoiding latency and security issues. It is generally the same technique as used in the web by running JavaScript in the web browser.

3.0 PHYSICS MODEL FOR AIRCRAFT AND GUIDED BOMB

3.1 The aircraft model

As shown schematically in Fig. 3, the six degrees of freedom comprise of the three spatial co-ordinates (*x,y,z*) and the three Euler angles, roll, pitch, and yaw (θ, ω, γ). In a real process of air-to-ground strike the equations of aircraft motion with six degree-of-freedom will be solved by the fire control system, the research about GBRP mainly focusing on the aircraft trajectory property, so in GBRP process reducing the motion formulation to a three degree-of-freedom system can accelerate the whole simulation velocity at the same time fulfilling the plan requirements. The simplified model for the aircraft can be represented as follows;

$$\begin{cases} \dot{v}(t) = g(n_x - \mathrm{Sin}\theta) \quad \dot{\theta}(t) = g(n_y \mathrm{Cos}\gamma - \mathrm{Cos}\theta)/v \quad \dot{\omega}(t) = gn_y \mathrm{Sin}\gamma/(v\mathrm{Cos}\theta) \\ \dot{x}(t) = v\mathrm{Cos}\theta\mathrm{Cos}\omega \quad \dot{y}(t) = v\mathrm{Sin}\theta \quad \dot{z}(t) = -v\mathrm{Cos}\theta\mathrm{Sin}\omega \end{cases} \quad \ldots (1)$$

where n_x, n_y are the loads in the tangential and normal directions respectively, v stands for the velocity of the aircraft, and g is the gravity acceleration.

Now based on Equation (1), we can deduce different dynamics models for the tactics movements in GBRP. For example, the uniform straight motion should follow formulations

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

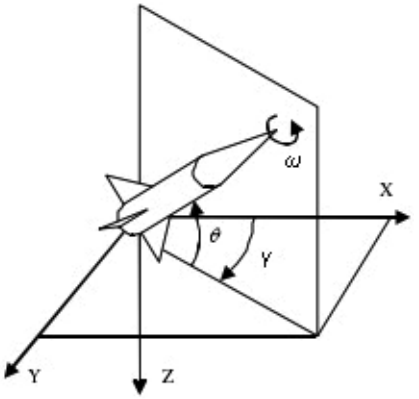


Figure 3. Rigid body dynamics Schematic for the aircraft and bomb in GBRP.

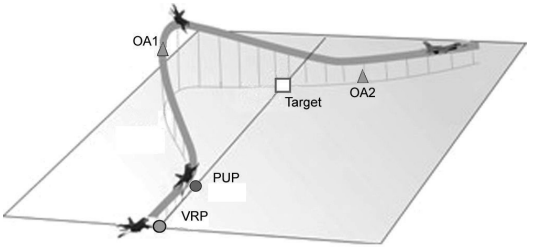


Figure 4. POPUP release serialisation.

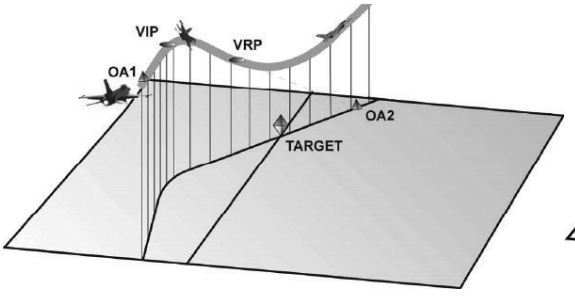


Figure 5. TOSS release serialisation.

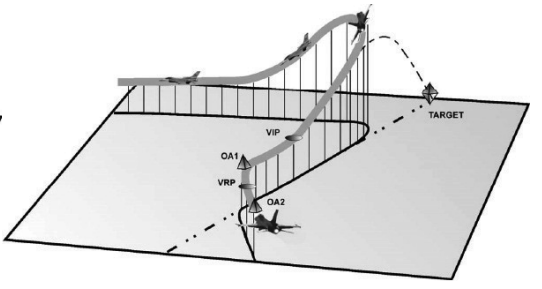


Figure 6. HADB release serialisation.

$$\begin{cases} P = D \\ L\mathrm{Cos}\gamma = mg \\ mv^2 = RL\mathrm{Sin}\gamma \end{cases} \quad \ldots (2)$$

where P is the course short and D is the range of fire, L is the force applied to the aircraft in the normal direction and R is the radius of straight motion, so considering the characteristics of uniform motion the straight aircraft should experience zero n_x and a normal load

$$n_y = \sqrt{v^4/(R^2g^2)+1} \quad \ldots (3)$$

at the same time we should have;

$$\gamma = \mathrm{arc\,Cos}1/n_y \quad \ldots (4)$$

As for the diving movement, in the approach section we will have;

$$\Delta H = \frac{v^2}{g}\ln\frac{n_y - \mathrm{Cos}\theta}{n_y - 1} \quad \ldots (5)$$

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

and the distance variations in the vertical and horizontal direction should be;

$$\Delta H = \frac{v^2}{g} \ln \frac{n_y - \mathrm{Cos}\theta}{n_y - 1} \quad \ldots (6)$$

$$\Delta L = \frac{v^2}{g} \left(\frac{n_y}{\sqrt{1-n^2}} \mathrm{Arth} \frac{\sqrt{1-n^2}\mathrm{Sin}\theta}{n\mathrm{Cos}\theta - 1} - \theta \right) \quad \ldots (7)$$

However, according to the energy conservation law in the recovery section the aircraft reacts according to the following equation;

$$\Delta H = \frac{v^2}{2g} \left[\left(\frac{n_y - \mathrm{Cos}\theta}{n_y - 1} \right)^2 - 1 \right] \quad \ldots (8)$$

In the straight dive section the aircraft should keep (θ, ω) constant and we can obtain n_x and n_y from Equation (1).

3.2 The guided bomb model

Theoretically, the description of the released guided bomb is the same as that of the aircraft, so rigid body dynamics can be used to formulate the bomb's equations as follows;

$$\begin{cases} \dot{v}(t) = -Q/m - g\mathrm{Sin}\theta \\ \dot{\theta}(t) = g(n_y - \mathrm{Cos}\theta)/v \\ \dot{\psi}(t) = -gn_z/(v\mathrm{Cos}\theta) \end{cases} \quad \ldots (9)$$

where Q is the air drag force, ψ is the projectile deflation angle and n_z is the load in lateral direction.

3.3 Variable separation model

A change of loads will occur in the release when the bomb separates from the aircraft, and then the thrust-weight ratio shifts with it, which impacts on the aircraft. Given the total weight of the aircraft containing the weight of the bomb M_b and the aircraft M_a before release $M = M_a + M_b$, lifting force L and propelling force F, we can obtain $n_x = (F - D)/\mathrm{M}$ and $n_y = (L - M)/M$ while after release $n_x = (F - D)/M_a$ and $n_y = (L - M_a)/M_a$.

4.0 RELEASE ANALYSIS AND OPTIMISATION

4.1 The common tactics for the aircraft

In a GBRP there exist three common tactics for the aircraft, that is, POPUP as in Fig. 4, TOSS as in Fig. 5 and HADB (high altitude dive bombing) as in Fig. 6, where the former two tactics opt for low-flying approach and zoom-bombing while the latter uses high-flying in order to find target. For an instance of POPUP, the aircraft takes a low-fly approach at VRP, pulls up from PUP to OA1 at maximum height to attack target with aiming point OA2, and then escapes. The detailed waypoints in a POPUP attack can be seen in Fig. 7 and the corresponding notation of these points can be seen in Table 1.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

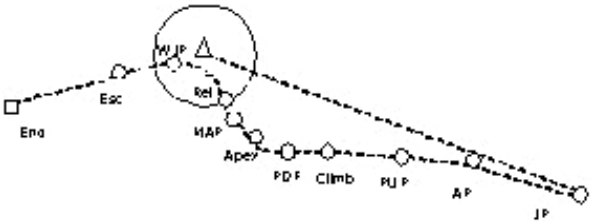


Figure 7. Scheme of waypoints in a POPUP.

Table 1
Waypoint notation

Waypoint	Denotation
IP	Initial point
AP	Action point
PUP	Pull up point
Climb	Maximum angle of climb
PDP	Pull down point
Apex	The highest point in the aircraft route
MAP	Manoeuvre action point
Rel	Release point
WIP	Weapon impact point
Esc	Escape point

4.2 The release optimisation based on SQP (sequential quadratic programming)

The SER (space extent for release) is dependant on many uncertain parameters like weapon performance, weather condition, wind direction etc., so first of all in practice the planner qualify the weapon and the parameters decoupled with spatial dimension besides v and γ as in Equation (1). Under this circumstance a SER should be solved from an optimisation criteria, that all predefined parameters will work. Now we adopt the maximised release distance from the target as an optimum. Because of the bomb's off-axis, the SER will depend on X and Z in the ground plane co-ordinate system and the problem evolves into a multi-object optimisation.

To solve the problem at first the range of the optimum is found with a global search algorithm, and then the height of the boundary of SER is stratified and an envelope is set up. In order to derive unifrom Pareto points on the boundary, a total distance D from Rel to the target, and D_x/D_y from Rel to X/Y of the target, are defined as evaluation indicators. The fitness function will be formulated as follows;

$$f = k_1 D_x + k_2 D_y + k_3 D \quad \ldots (10)$$

where k_1, k_2, k_3 are the weights of the parameters respectively, associated with the upper bound Max(f) and the lower bound Min(f) of the SQP with accelerating convergence towards the optimum.

As shown in Fig. 7 from AP to Rel, the aircraft will carry out a sequence of acts including climb, turn and level flight. The height H_{search} at AP should meet the search condition for the target,

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

correlating with the distance between the aircraft and the target, since the higher at AP and the easier to be attacked by the antagonist. During the climb we have;

$$\theta = \arcSin(H_{rel} - H_{ap} - H_{level}) / vT_c \quad \ldots (11)$$

where H_{rel}, H_{ap}, H_{level} are the heights at Rel, AP and level flight respectively, and T_c is a time interval of climb. If v is kept constant, the thrust should be;

$$n_x = \Sin\theta \quad \ldots (12)$$

For stability, the following equation should be met;

$$\gamma_a = \arcCos \frac{1}{n_y} \quad \ldots (13)$$

With reverse integration to Equation (1) the aircraft trajectory can be obtained.
Similarly, from IP to AP the aircraft will carry out accelerate, fly straight and level, and climb.

4.3 The release optimisation based on Monte Carlo

Currently, the performance of the aircraft at the first assault is the most significant factor in determining the effectiveness of the whole air-to-ground strike[19]. Hitting the target is of the greatest importance. However, conventional programming methods cannot fulfill this requirement because of various uncertainties and perturbations to the ideal waypoints in practical situations. Finding a method for measuring waypoint perturbation will be the key to improve the hit probability and robustness, and even the whole strike efficiency.

With probability theory, given most parameters and perturbations obeying the normal distribution N, we have the relation between the variable X-$N(\mu, \sigma^2)$ and the standard normal variable X-$N(0, 1)$ as follows;

$$X = \mu + X'\sigma \quad \ldots (14)$$

where μ and σ are the mean value and mean square deviation, respectively. If the parameter obeys the normal distribution, we have;

$$A_j^{(i)} = M(A_j) + \alpha_j^{(i)} \Delta A_{j\max} \quad \ldots (15)$$

where j is the index of the parameters and i represents the sample order, $M(A_j)$ is the mathematical expectation about A_j and ΔA_{jmax} occupies only a half distribution interval 3σ of A_j, and α_j is a normal random obeying $N(0,1/3)$ approximately.

From the perspective of stochastic procedure, we propose a robust and reliable release planning based on Monte Carlo algorithm described as follows,

Step 1. Determine all of the parameters correlating with the planning results and their distribution law e.g. v, ω, γ, H_{level}.

Step 2. Serialise release sequences and construct the corresponding models, as well as the ideal waypoints.

Step 3. Generate sample values for stochastic perturbation variables based on the distribution law and probability model, calculating deviation ranges by referring to the ideal waypoints.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

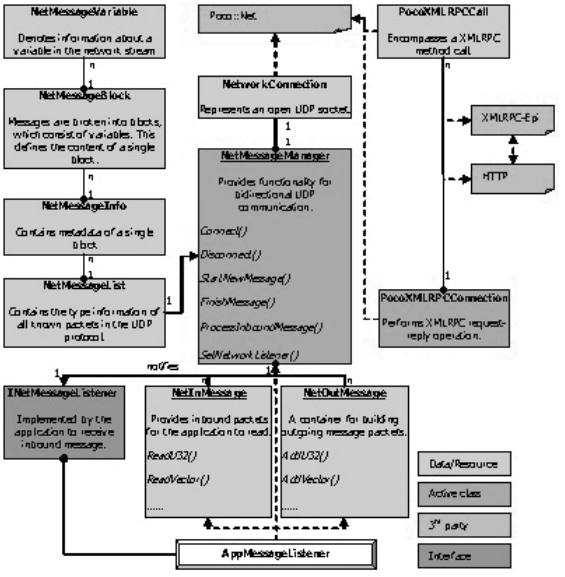


Figure 8. Network communication design.

Step 4. Build the sample values in the constructed models and to the result distribution, adjusting parameters to calculate deviation ranges again.

Step 5. Count and analyse the results, deciding if the adjusted deviation ranges are feasible. If no, Step 4; If yes, Step 6.

Step 6. End.

Applying the algorithm to a release, we should also make a detailed reverse planning analysis from AP to Rel and from IP to AP etc as mentioned in Section 5.2, the analysis procedure and simulation results are shown in Section 7 with comparisons.

5.0 NETWORK COMMUNICATION FOR DVE

Usually, operational patterns for modern warfare work in cooperation with various distributed sectors and disciplines to handle the joint campaign of integrative aerospace. The traditional release procedure of single aircraft to single target no longer meets the advanced air-to-ground attack needs. The simulation based on a single machine should be replaced with the distributed interactive simulation, that is, all of the simulation actors such as planners, pilots, antagonists located on different client machines should login the same shared virtual environment grid server to undertake respective tasks.

5.1 Network interface

The design of low level network interfaces for communication is based on the idea that the application-layer UDP protocol is specified by data, instead of hard-coding. This kind of core protocol allows easy changes to the packet data format and addition of new custom packet types later on, and it also functions as a form of documentation of the protocol. The communication works using two different methods. XML-RPC over HTTP is used to perform the initial connection handshake and login, which is followed by a stream of UDP messaging, until the

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

client disconnects. This low-level connection library encompasses both of these and provides the client with a suitable application-level interface.

The library is based around two active classes, NetMessageManager and PocoXMLRPCConnection, as seen in Fig. 8, which provides the application with the capabilities to connect to the server. The classes drawn in gray contain data or represent resources and do not perform any global state changes or communications of their own. The orange packages represent dependencies to external open source libraries.

NetMessageManager handles a single bidirectional UDP communication channel between the client and the server. It implements the necessary low-level UDP protocol features, like trusted transmission, reliability, resending of packets, pinging etc.. It also provides a way for the application to receive and send properly crafted message packets, as defined by the common protocol data file. Receiving packets is implemented through the INetMessageListener interface. The manager notifies this listener for all new packets. The NetInMessage structure is the object that contains the message, and to examine the received packet the application uses the methods of NetInMessage to query both the type and content of the message. In order to send out messages, the application uses the NetOutMessage structure. The methods of NetOutMessage allow the application to add in all the data to be send, and finally the application calls FinishMessage() to send out the message. To perform the initial XML-RPC login, the application uses the PocoXMLRPCConnection class. It handles the crafting of the request and serialises it over the network. When the reply is received, PocoXMLRPCConnection interprets it and passes it back to the application in a readable form.

5.2 The GBRP viewer platform

The client is a viewer designed to complement the GBRP server-side platform. As a platform for enabling a large number of joint campaign applications, the same way a web browser enables a large number of web-based applications, it must offer core functionalities, and then open extensibility of that core through scripting and modules. At the same time, as a networked platform, the viewer cannot rely on any one protocol, or allow one protocol's assumptions to penetrate the design too deeply.

The core functionalities of the GBRP viewer are streaming and rendering interactive 3D scenes. To extend the core functionalities for a number of uses such as telecommunication, planning, or bombing, we rely on the framework to provide a deeply integrated module system. Modules must not assume anything about other modules, and rely only on the services provided by the framework. While a 3D scene may be assumed to be fundamental to a viewer, we wish to avoid depending on one specific rendering library, so this internal data model should not be tied to the renderer.

To decouple such dependencies, we define a module very loosely: it is any object which understands the local messaging and event system, and can generate replies in coordination with other modules. One may consider such modules as agents interacting in self-interest, over a shared space with a shared channel. To minimise the overhead of messaging or events, and maintain the real-time nature of the simulation, we seek to limit the number and size of messages or events. Modules keep their own data, expose a limited subset of their state to other modules, and send only terse updates to explicitly subscribed modules. We also note that the viewer possess a distinct data flow: from network or user inputs, to logical layer for decision making, to rendering for display; that will in practice limit the combinatorial nature of such messaging.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

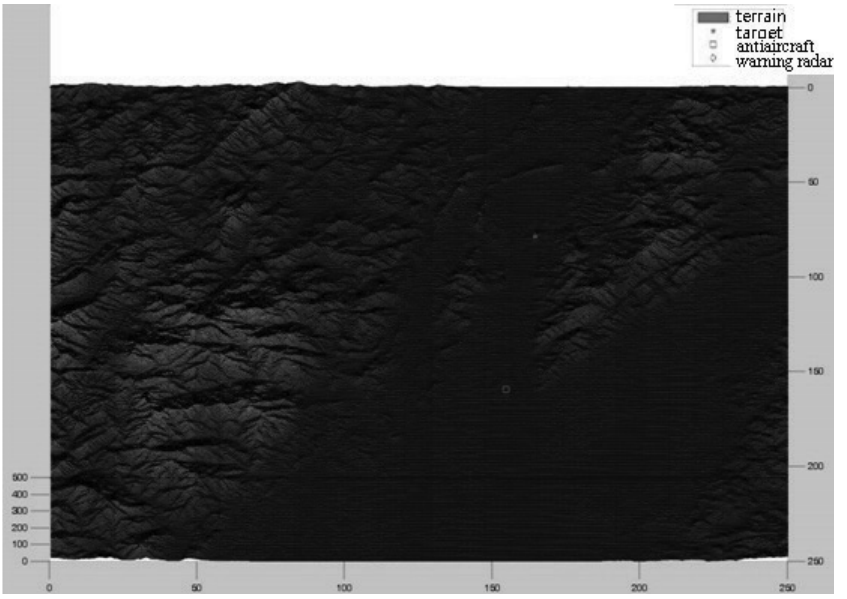


Figure 9. The schematic of the battlefield terrain and threats.

Table 2
Threats location

Threat position	X(km) coverage(km)	Y(km)	Z(km)	Threat type	
The ground-to-air missile position	164·5	74	70	Destroy	8
The antiaircraft position	159·5	154·5	41·25	Destroy	40
The warning radar position	52·5	135	589·5	Destroy	200

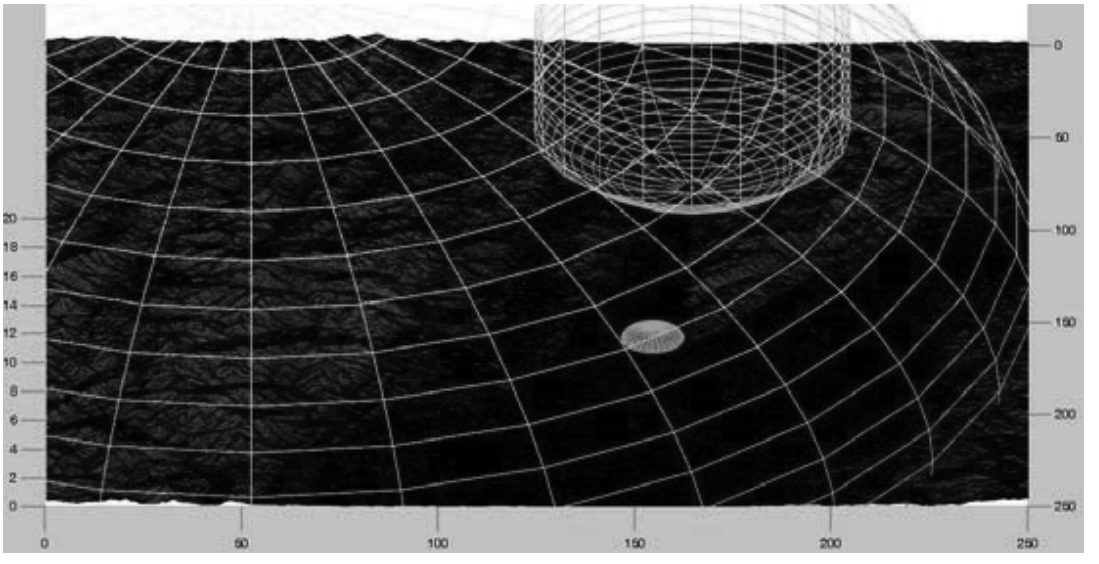


Figure 10. Visualising the threat coverage.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

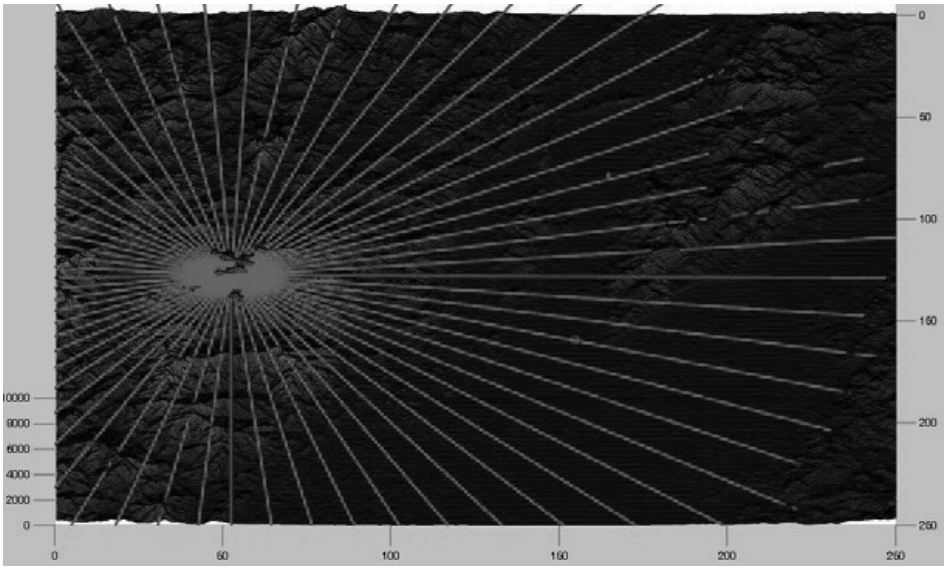


Figure 11.The inter-visibility condition of the warning radar coverage to the aircraft with a distance of 200km and a low altitude of 300m.

6.0 EXPERIMENT

6.1 Mission assignment and DVE configuration

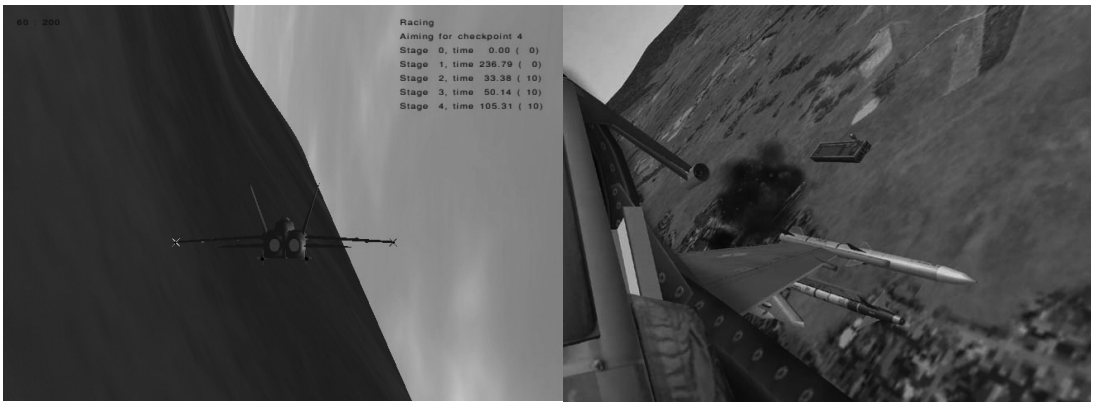
The simulated GBRP condition as shown in Fig. 9, in a geographical environment of the battlefield with size of 250km × 250km, there exists a warning radar position at the highest altitude denoted by a diamond, an antiaircraft position at the valley gate denoted by a rectangle, and a highly mobile ground-to-air missile position in the valley denoted by a star. All associated data can be seen in Table 2, corresponding visualisation schematic shown in Fig. 10, and by a terrain masking analysis for the warning radar coverage according to Fig. 11 we can conclude that there appears a blind zone of the radar in the positive eastward direction if the aircraft enters the coverage with an altitude of 300m.

The mission is defined as an attack on the ground-to-air missile position. Since the missile position exists as a threat at the same time, the guided bomb release has to be accomplished out of the reach of missile range. Taking the mobility of the missile position into account, the GBRP should be optimised with the shortest aircraft route out of radar coverage to avoid the antagonist moving and intercepting.

In the mission the bomb is guided with the proportional law possesses parameters and properties including the interval of release velocity of [150,300]ms^{-1}, the interval of release height of [500,5,000]m, mass of 560kg and the lift coefficient C_L of 0·5 (Fig. 12). The cruising velocity of the aircraft is 200ms^{-1}, whose radar coverage is [+45°,–45°] and the maximum of overload is 7G. It takes 18s for the airborne radar to search and detect a target, so the aircraft will be tracked and locked by the antagonist if the exposure time is over 35s. Taken together, it will be better for the aircraft to approach the target from the east than from the valley gate, and the height of Rel should be 5,000m with level flight velocity of 300ms^{-1}.

As seen in Table 2 and Fig. 12, in the DVE we will configure four types of client-side viewers in different locations including an attacker and three types of antagonists for every threat consisting

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press



(a) (b)

Figure 12. The aircraft with guided bomb executing the mission in the DVE

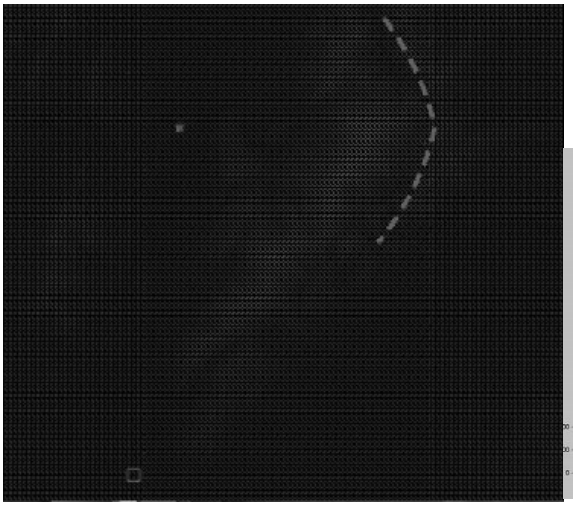


Figure 13. The guided bomb release boundary.

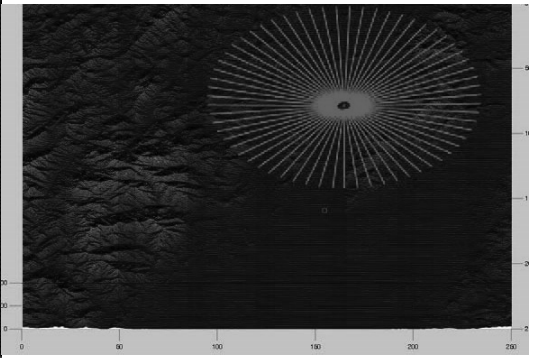


Figure 14. The inter-visibility condition of the airborne radar.

of multiple actors or players. The GBRP server is where client login with ID or legacy login protocol then authenticates and figures out where viewer is trying to log in. The network status for experiment has been summarised in Table 3.

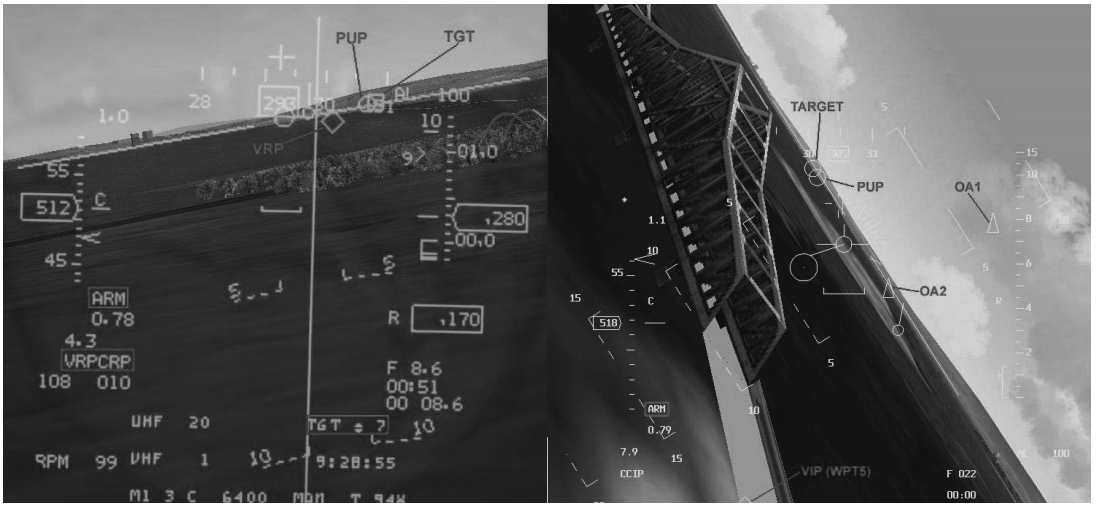
6.2 Execute GBRP based on SQP

With the computation method from Section 5.2 we can obtain the guided bomb release boundary denoted by the red dash line in Fig. 13. For simplicity we select the point of (79, 220.31, 5.0) in the positive eastward direction of the boundary as Rel. Considering the inter-visibility condition and performance of the airborne radar as seen in Fig. 14, we define AP at the distance of 70km from the target and WIP at the height of 2,000m. Similarly, we select the points of (79, 224.35, 2.0) and (79, 226.98, 0.3) in the positive eastward direction of the attack and action boundary as MAP and IP, respectively. Alternatively, all the waypoints will be displayed on the screen of the aircraft cockpit for the client-side attack player, as shown in Fig. 15. And consequently, as shown in Fig. 16 by the physics model in Section 4 the whole flying trajectory can be generated from the waypoints.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

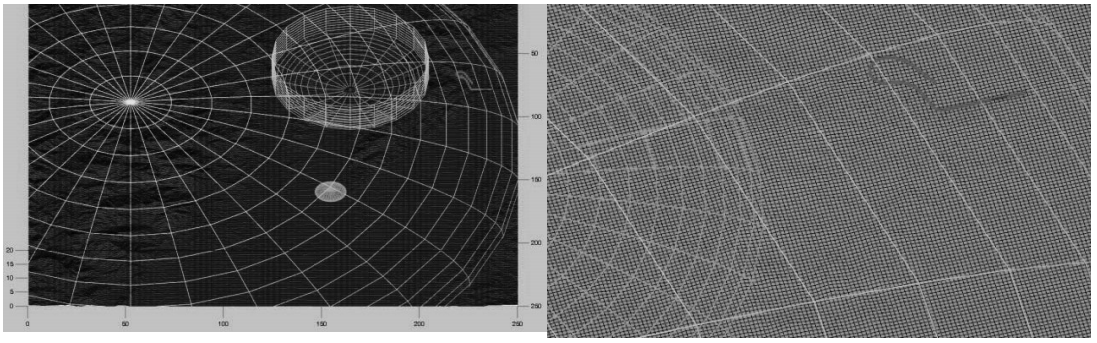
Table 3
The network status for experiment

	Server to Client			Client to Server	
Number of players	Bytes/s	Packets/s	Bytes/s/client	Bytes/s	Packets/s
1	5·7K	50	5·7K	3K	50
2	8·5K	50	4·25K	3K	50
3	11·6K	50	3·87K	3K	50
19	37K	60~90	1·95K	4·8K	20~100



(a) (b)

Figure 15. The waypoints displayed on the aircraft screen of the cockpit in the DVE.



(a) (b)

Figure 16. The aircraft trajectory during executing GBRP based on SQP.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

Table 4
Parameters error distribution for Rel

Parameters	Distribution type	Distribution law
Height	Normal	$N(5000, 5^2)$
Velocity	Normal	$N(300, 3^2)$
Deflation angle	Normal	$N(0, 0{\cdot}01^2)$
Drift angle	Normal	$N(0, 0{\cdot}01^2)$
Vertical position	Normal	$N(55893,10^2)$
Horizontal distance	Normal	$N(0, 10^2)$

Table 5
Parameters error distribution for MAP

Parameters	Distribution type	Distribution law
Velocity	Normal	$N(300, 3^2)$
Deflation angle	Normal	$N(0.52, 0{\cdot}01^2)$
Drift angle	Normal	$N(0, 0{\cdot}01^2)$
Vertical position	Normal	$N(3868,10^2)$
Horizontal distance	Normal	$N(0, 10^2)$

Table 6
Parameters error distribution for IP

Parameters	Distribution type	Distribution law
Height	Normal	$N(1000, 10^2)$
Velocity	Normal	$N(300, 3^2)$
Deflation angle	Normal	$N(0, 0{\cdot}01^2)$
Drift angle	Normal	$N(0, 0{\cdot}01^2)$

6.3 Execute GBRP based on Monte Carlo

Based on Monte Carlo, all the initial parameters should be perturbed according to the probability distribution to simulate uncertainties. In GBRP the parameter errors can be defined as Table 4, Table 5 and Table 6. Then resorting to the computation method from Section 5.3 those waypoints can be obtained with some redundancy to deal with perturbations. In that case, the ratio of success about the transition between waypoints and release should be over 90%. The aircraft trajectory is generated and displayed in pink in Fig. 17.

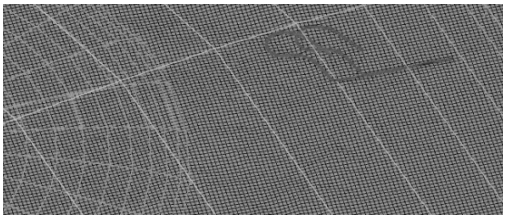


Figure 17. The aircraft trajectory in pink color generated by Monte Carlo.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

6.4 Analyse simulation results

As seen in Fig. 17, the pink trajectory based on Monte Carlo is closer to the target than the red one based on SQP, and the corresponding slope between IP and Rel is smaller. With a quantitative analysis we can define the ratio of success as follows,

$$P = P_{rel} P_{all} P_{ap} \quad \ldots (16)$$

where P is the total ratio of success, P_{rel} is the ration of release success, P_{all} is the ratio of success about transition from MAP to Rel and P_{ap} is the ratio of success about transition from IP to MAP.

The comparison results can be seen in Table 7. It turned out that the total ratio of success and the release efficiency by Monte Carlo has been improved much more than by the traditional SQP, at the same time the exposure time under the warning radar coverage does not increase noticeably, although it approaches close to threats in order to destroy them and the risk level is higher.

6.5 Discussions

Some other optimisation techniques for trajectory planning including energy-optimal, simplex, conjugate gradient, differential dynamics etc as well as their hybrid methods, as is always converted to a complex non-linear programming problem, generally obtain only the numerical solution. Without the help of some AI algorithms, these methods cannot deal with the problem of redundancy such as uncertain factors or perturbations[20-23].

The redundancy problem in GBRP should be suitable for Monte Carlo method, because the error appears subject to certain possibility distribution and its boundary is very difficult to be obtained to counterbalance the influence of parameters random perturbation. And it is unnecessary to model GBRP accurately for an aircraft pilot never to execute a predefined GBRP precisely. By Monte Carlo method the GBRP can obtain a reasonable redundancy to improve the success rate and efficacy of release. In fact, provided abundant experimental samples the solution from Monte Carlo method often has a higher accuracy than that from some simplified mathematical models.

Additionally, based on an open DVE framework a joint campaign can simulate much more complex uncertainties by participant nodes than based on only one single machine. Considering the time-space consistency in a wide networked DVE[24-26] this simulation environment has been constructed in a private local network ensuring communication open, smooth and safe. So every node can log into or out dynamically and randomly from the whole system.

Table 7
Comparison results

Method	Waypoint	Co-ordinate (km)	Deflation angle (rad)	Velocity (ms^{-1})	Ratio of success	The total ratio of success
	Rel	(79,220.31,5.0)	0	300	46·3%	
SQP	MAP	(79,224.35,2.0)	1	300	47·1%	10·12%
	IP	(79,226.98,0.3)	0	200	46·4%	
Monte Carlo	Rel	(79,217.11,5.0)	0	300	90·5%	
	MAP	(79,220.75,2.0)	0·93	300	90·2%	73·95%
	IP	(79,223.41,0.3)	0	200	91·1%	
Target		(79,164.51,0.31)				

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

7.0 CONCLUSION

The procedure of planning the guided bomb release is confronted with many constraints and uncertainties, and the methods are in dire need of increasing the level of aircraft safety. The essential basic models and tactics for the whole release process have been built, and a reverse GBRP analysis has been carried out. Compared to the traditional SQP method, the proposed Monte Carlo method exhibits better robustness, efficiency and accuracy by virtue of introducing redundant error perturbations into the initial planning parameters. In order to simulate reality and reliability we design a DVE architecture for GBRP, where multiple players will act differently as to participants in a joint campaign.

With the development of modern weapons, considering their redundancy ranges and error perturbations, uncertainty generalisation will become very difficult, so further research should focus on including more detailed models and tactics, as well as more precise redundant data for Monte Carlo method. Additionally, experiments for conventional arms like rocket bombs will confirm and improve the method and software further.

ACKNOWLEDGEMENTS

We should acknowledge the reviewers of the paper for a series of improvement and the editor's detailed revision advice. At the same time the authors are grateful to National Natural Science Foundation of China (No.61103153/F020503) and 863 plan (No. SS2013AA010903) in carrying out this research for financial support.

REFERENCES

1. Command USAF Materiel. Capability requirements specification for the joint mission planning system (JMPS) auto router (AR) common capability (CC), 2004.
2. WEST, W.J. Developmental testing of a laser-guided bomb simulation, US Air Force T&E Days, 2008.
3. LYACH, Jr. D. Terminally guided weapon delivery system, Institution, FC, USA, 1986.
4. MURPHY, W.J. and YOUNG, W.L. Integrated flight/weapon control design and evaluation, Proceedings of the IEEE NAECON, 1981, (2), pp 536-543.
5. NOWAKOVSKI, V.M. Autonomous precision weapon delivery using synthetic array radar, USA, 1991.
6. BUSHER, R.G. and HILLS, W. *Weapon Delivery System*, Lear Siegler, USA,1976.
7. W, BM, D, WR, V, TJ. Kalman filter design and performance for an operational F-4 loran inertial weapon delivery system, Proceeding of the National Aerospace and Electronics Conference, 1976.
8. SIMEONE, A.R. and BEAN, C.M. *Weapon Guidance System*, LVS Corporation,1986.
9. SINGHAL, S. and ZYDA, M. *Networked Virtual Environments*, ACM Press, 1999.
10. CANTERBURY, M. An automated approach to distributed interactive simulation (DIS) protocol entity development, September 1995.
11. Institute of Electrical and Electronic Engineers. 1516-2000, IEEE standard for modeling and simulation high level architecture-framework and rules, IEEE, New York, USA, September 2000.
12. ANTONELLO, R., FERNANDES, S., MOREIRA, J., CUNHA, P., KAMIENSKI, C. and SADOK, D. Traffic analysis and synthetic models of second life, *Multimedia Systems*, 2009, **15**, (1), pp 33-47.
13. JAMES, K. and MARK, C. Traffic analysis of avatars in second life, NOSDAV, Braunschweig, Germany, May 2008.
14. CLAYPOOL, M. and CLAYPOOL, K. Latency and player actions in online games, *ACM Communication*, 2006, **49**, (11), pp 40–45.
15. HU, S.Y., CHEN, J.F. and CHEN, T.H. A scalable peer-to-peer network for virtual environments, *IEEE Network*, 2006, **20**, (4), pp 22–31.
16. BARABASI, A.L. and ALBERT, R. Emergence of scaling in random networks, *Science*, 1999, **286**, (5439), pp 509–512.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press

17. *Oxford. Dictionary of Computing*, Oxford University Press, 1996.
18. YAN, J. and RANDELL, B. A Systematic Classification of Cheating in Online Games, NETGAMES, Hawthorne, USA, October 2005.
19. YOUNG, J-S. Profile emulations of target waypoints for mission planning systems of tactical flights, Control Conference (ASCC), 2011, pp 453-458.
20. CHELOUAH, R. and SIARRY, P. Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions, *European J Operational Research*, 2003, **148**, (2), pp 335-348.
21. MCLAIN, T.W. and BEARD, R.W. Trajectory planning for coordinated rendezvous of unmanned air vehicles, *AIAA*, 2000, pp 4369-4372.
22. JUDD, K.B. and MCLAIN, T.W. Spline-Based Path Planning for Unmanned Air Vehicles, Proceedings of Guidance Navigation and Control Conference, 2001, pp 1182-1190.
23. KIM, Y., GULAN, D-W. and POSTLETHWAITE, I. Real-time path planning with limited information for autonomous unmanned air vehicles, *Automatica*, 2008, **44**, (3), pp 696-712.
24. AGGARWAL, S., BANAVAR, H. and KHANDELWAL, A. Accuracy in dead-recking based distributed multi-player games, Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games, 2004, pp 161-165.
25. SCHEUERMANN, B., KIESS, W. and ROOS, M. On the time synchronization of distributed log files in networks with local broadcast media, *IEEE/ACM Transactions on Networking*, 2009, **17**, (2), pp 431-444.
26. ZHOU, S., CAI, W. and TURNER, S.J. Critical causal order of events in distributed virtual environments, *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2007, **3**, (3), p 15.

https://doi.org/10.1017/S0001924000008228 Published online by Cambridge University Press