



---

# SIMULATION SYSTEMS OF COMPLEX TECHNICAL SYSTEMS

**Inteasar Yaseen Khudhair, Zainab Qahtan Mohammed, Yasiralimatnee**

Department of Computer Science, Basic Education College,  
University of Diyala, IRAQ

## ABSTRACT

*The article describes the architecture of the training system on the example of a prototype aircraft simulator. The visualization subsystem of the training simulator provides the display of the results of modeling the external environment and the control object using information display devices. It should provide reproduction of the created virtual scene with sufficiently detailed content, allowing operators to successfully perform the assigned tasks. The requirements for the training and training system of complex technical systems, including the requirements for its visualization subsystem, are presented. The developed architecture provides a unification of ways of interfacing with equipment, the use of a training and training system with various information input devices, and also allows you to avoid high connectivity of components. Each device, as a rule, has its own characteristics: specialized software, closed information exchange protocols, connectors of various standards.*

*The developed system of extensions (plug-ins) allows to take into account the features of devices without modifying the main module and other subsystems. A management interface has been created for working with plugins. Plug-ins are independent of each other and can be added or removed without violating the integrity of the system. Depending on the computational load, the data processing of the training system can be implemented on one hardware complex or use a separate one for each subsystem. Any subsystem can be created by independent developers and represented by separate software systems.*

*It is possible to work on designed subsystems on hardware complexes with various processor architectures, ways to store bytes in memory (big-endian or little-endian) and on various operating systems. An algorithm for converting coordinates derived from a mathematical model into a coordinate system of a visualization subsystem is described.*

**Keywords:** TOC, training and training systems, visualization system, three-dimensional modeling, simulator, simulation systems.

**Cite this Article:** Inteasar Yaseen Khudhair, Zainab Qahtan Mohammed, Yasiralimatnee, Simulation Systems of Complex Technical Systems, *International Journal of Civil Engineering and Technology (IJCIET)* 9(13), 2018, pp. 1913–1918.  
<http://www.iaeme.com/ijciyet/issues.asp?JType=IJCIET&VType=9&IType=13>

---

## 1. INTRODUCTION

By a simulate training of complex system (TOC) of an operator of a complex technical system (CTS), we will understand a technical tool for training operators in a single information environment that meets the requirements of training methods, creates the conditions for obtaining knowledge, skills and abilities, implements a model of such systems and provides control over the actions of the learner, as well as for research. In addition to the formation of individual professional skills and abilities, TOC can be used to work out group operations.

Depending on the volume of the TOC problems to be solved, it can be divided into complex, specialized, procedural. Integrated simulators are designed to meet the challenges of practical education and training of operators, fully covering their professional activities, and should ensure the refinement and development of all professional skills. Specialized simulators are designed to train operators to perform certain problem-oriented tasks. Procedural simulators are designed to develop sustainable skills in specific types of activities, working out certain important management procedures for a certain type of equipment performed at specific stages of operation and under certain conditions [1].

This article describes the system on the example of the prototype of the aviation TOC, to which the following requirements apply:

- simulation of procedures in real time (the latency of the simulator is not more than 20 ms, for implementation it is necessary to perform calculations within 15 ms, 5 ms reserve);
- no image breaks during multi-channel visualization;
- elimination of false skills;
- modular architecture of construction, which allows to remove, replace and create modules without deep processing of the system.

## 2. APPLICATION OF MODULAR ARCHITECTURE

The use of a modular architecture allows you to avoid high connectivity of components, which can impose a number of restrictions on the operation of TOC (for example, for each new version of the module from different developers, it is necessary to change the functionality of the entire system for compatibility). A system of extensions (plug-ins) that interact with the main module was developed. Plug-ins are independent of each other and can be added or removed without violating the integrity of the system [2].

A control interface was created for working with external plugins. When the system boots, it searches for plug-ins by extension in the specified directory, after which the plug-in is initialized. Each plug-in works in a separate thread, since the use of one thread for all plug-ins can cause the simulation system to hang, if any plug-in waits for a certain condition. For the loaded plug-in to function, the plug-in function must be implemented, which describes the main stages of the plug-in. For example, in this function, incoming and outgoing ports and IP addresses predefined in the configuration file can be determined, data is prepared for sending to the corresponding subsystem and processing of updated data coming from the subsystem with subsequent recording to the storage,

If initialization was successful, the plugin is added to the list of downloaded. If unsuccessful initialization, it is unloaded from memory. When a request is made to stop the main module, all plugins are unloaded.

### 3. DESCRIPTION OF SUBSYSTEMS THAT ARE PART OF THE TOC

Depending on the computational load, it is possible to implement TOC data processing on one hardware complex or for each subsystem to use a separate one [3]. Any subsystem can be created by independent developers and represented by separate program complexes. It is necessary to take into account that the developed subsystems can work on hardware complexes with different processor architectures, ways of storing bytes in memory (big-endian or little-endian) and on different operating systems. For example, when sending a packet of data to a subsystem that uses a different order of bytes, it is necessary to swap bytes in the plugin after receiving the packet from the subsystem and before sending the data. Also note that for structures that contain nested structures, it is necessary to permute the order of bytes separately for each value. Data exchange between the subsystem and the plug-in can be carried out using Ethernet technology using UDP protocol or by connecting the library to the plug-in.

To specify the location of the subsystems in this prototype TOC, the configuration file is used, which specifies the IP addresses, incoming and outgoing ports, the MAC address of the network card of the hardware complex of the main module. Using a MAC address is possible due to the presence of several network cards on a workstation.

For subsystems, the following approach to working with the network is possible. When a packet is received, the subsystem determines the incoming IP address. After calculating the new parameters, the generated packet will be sent to this IP address on the specified outgoing port. Incoming and outgoing ports can be specified both in the configuration file for a specific subsystem, and as input arguments of the application. If there is no specified IP address or a description of the incoming or outgoing ports, the default values (local host) will be used.

The main module receives parameters from all subsystems, the values of which are stored in a single storage (structure) [4]. Parameter values can be of a different type, access is possible from any plugin. This structure can store numbers, strings, arrays and binary data. Data storage is implemented as a set of key – value pairs. If the data from the structure needs to be extracted and sent to the required subsystem, it is converted using the appropriate functions from the format used in this structure to the format required for the underlying components.

TOC includes subsystems

- Instructor's workplace;
- formation of control actions;
- mathematical modeling of the object, including the database of descriptions of the object of modeling;
- modeling of electronic and mechanical means of displaying the state of an object;
- mathematical modeling of the environment, including physical modeling, modeling of the underlying surface and the database of real environment objects;
- visualization;
- navigation.

The use of such an architecture can also provide a unification of ways of interfacing with equipment and will allow the use of TOC with various input devices. Each hardware, as a rule, has its own characteristics: specialized software, closed information exchange protocols, connectors of various standards [5]. The developed system of plug-ins allows you to take into account the features of the hardware without modifying the main module and other subsystems.

During the initialization of the control actions formation subsystem, all connected devices are detected. The initialization stage consists in obtaining the device indexes in the system, their names, and writing the received data into a common array. The values of the parameters of the steering wheels and control levers are the angles of deviation from the corresponding initial values of the axes.

The subsystem of mathematical modeling of an object also performs the following functions:

- processing of the initial conditions entered by the instructor;
- calculation of the automatic control system (if available);
- calculation of object parameters taking into account control actions, including the determination of the spatial position of the object in a virtual environment and the calculation of parameters for instruments and control levers.

For a number of systems, behavior in emergency situations is not well understood. This may be due to the fact that there is a lack of experimental data, or the acquisition of such data is very expensive or even dangerous.

The instructor's workplace subsystem is a tool for monitoring and changing all parameters of the modeling system [6]. With its help, you can change the operating parameters of all systems that are part of the TOC. Throughout the entire workout, the instructor can set abnormal situations, including failures of various systems and equipment, set environmental parameters, and carry out emergency shutdown of the simulator. The instructor control system provides for the collection, processing, documentation, analysis and issuance of statistical data.

#### **4. REQUIREMENTS FOR THE TOC IMAGING SYSTEM**

Consider the requirements that apply to the visualization system TOC. Information received by visual observation is perceived differently depending on the distance of the observed objects. There is a division into three zones: up to 1 meter, from 1 to 30 meters and over 30 meters from the observer. In a zone of more than 30 meters, when binocular vision weakens (both eyes get the same image), occlusion and dimensions become important for adequate perception. If the screen is placed at a distance of about 3 meters from the learner, and the object according to the training scenario is removed from the observer less than 3 meters, misalignment may occur when using projection display systems.

In the simulator, the transmission of visual signals about movement is based on the fact that the human brain hardly distinguishes changes in situations (using only the visual channel) when the observer is stationary, and the surrounding objects are in motion and when the observer moves, and the objects are stationary. Therefore, a moving environment is generated, creating the illusion of movement of an observer (a mobile platform enhances the perception of movement) [7].

Detailing the playback environment is directly related to the performance of the simulation system. To increase the efficiency of using the resources of the simulator, it is necessary to determine the most important for achieving the goals of training a group of objects and, therefore, to reproduce them in detail.

To recreate the environment using real images of the underlying surface. The formation of the underlying surface can be carried out in the system for creating and editing three-dimensional graphics or using procedural generation [8]. When recreating the runway, you must display the touch points of the landing gear of the aircraft during landing. After creating a three-dimensional scene, it is saved into an intermediate format, for example COLLADA, which is later converted by the graphics engine converter into the internal format of three-dimensional scenes.

## The method of converting the coordinates of the mathematical model of TOC to the coordinate system of the visualization subsystem

Through the plugin, the visualization subsystem transmits the identification values of the packet, longitude, latitude and height (WGS 84), angles (in degrees) of roll, yaw, and pitch. These values are used to change the position and tilt angles of the virtual camera in the three-dimensional scene. Since the visualization subsystem has its own coordinate system, it is necessary to convert the obtained values into the coordinate system of the visualization subsystem. The calculation of the distance between two points on the surface of the Earth takes place according to the formula.

$$d=R\sqrt{\Delta\lambda^2 \cos(\lambda_1) \cos(\lambda_2) \Delta\varphi^2}$$

$$Dl = l2 - l1, Dj = j2 - j1, \quad (1)$$

where  $l2 - l1$  are the latitudes of two points;  $j2 - j1$  is the longitude of two points;  $R$  is the radius of the Earth;  $Dl, Dj$  - the distance between the points of latitude and longitude.

Next, you need to calculate the value of the azimuth, which is the angle between the plane of the meridian of the observation point and the vertical plane passing through this point and the observed object [9]. The azimuth value is necessary to calculate the final position of the aircraft:

$$\Psi = \tan^{-1} \left( \left| \frac{\cos(\lambda_2) \sin(\Delta\varphi) \cos(\lambda_1)}{\sin(\lambda_2) - \sin(\lambda_1) \cos(d)} \right| \right). \quad (2)$$

The absolute values of the numerator and denominator of the formula (2) used to calculate the relative azimuth are limited to the range  $\pm \pi / 2$ . Therefore, if the value of the denominator of formula (2) is less than zero, then  $y = p - y$ , if the value of the numerator of formula (2) is less than zero, then  $y = -y$ . After that, the total value of the direction of the magnetic field within the runway in radians is calculated:

$$q = RQDM + \text{MagVar}, \quad (3)$$

where  $RQDM$  is the direction of the magnetic field;  $\text{MagVar}$  - magnetic variation.

Magnetic variation represents various changes in the geomagnetic field in time, due to the existence of both magnetic and internal sources, both internal and external to the surface of the Earth [6].

Next, you need a numerical expression of the direction of the magnetic field to convert from degrees relative to the magnetic meridian to degrees relative to the true meridian. After that, the result is reduced to a range of  $\pm 180$  degrees according to the following algorithm.

If  $q > p$ , the value will take the following form:  $q = q - 2p$ .

If  $q < -p$ , the value will take the following form:  $q = q + 2p$ .

The calculation of the final position of the aircraft inside the virtual scene will be made using the following formulas:

$$x = -(-d \sin(y) \cos(2p - q) + d \cos(y) \sin(2p - q));$$

$$z = -(-d \sin(y) \cos(2p - q) - d \cos(y) \sin(2p - q));$$

$$y = \text{altitude}, \quad (4)$$

where  $x, y, z$  are the transformed coordinates of the model; altitude - the height of the model above sea level.

The values of roll (roll), yaw (yaw), pitch (pitch) will be converted from radians to degrees as follows:

$$rx = \frac{180^\circ}{\lambda} \text{pitch};$$

$$r_y = -\frac{180^\circ}{\lambda} \text{yaw}; \text{ (five)}$$

$$r_z = -\frac{180^\circ}{\lambda} \text{roll},$$

Where  $r_x$ ,  $r_y$ ,  $r_z$  are the model deviation values in degrees from the OX, OY, OZ planes, respectively.

The time intervals between the receipt of new data may impose restrictions on the smoothness of movement of the virtual camera [10]. This is due to the fact that the mathematical model operates at a constant frequency, and the visualization system can change the frame refresh rate throughout the entire working cycle. To maintain smooth movement, it is necessary to use intermediate coordinate values, which will be calculated using interpolation.

## 5. CONCLUSION

To recreate the sensations of the real world in the process of training sessions, it is necessary to recreate the signals that can be received by the operator during the execution of tasks. In order to increase the efficiency of using the resources of the simulator, it is necessary to determine which groups of objects are most important for achieving the objectives of the training and, therefore, to reproduce these groups in detail.

The developed system of extensions (plug-ins) allowed, depending on the computational load, to implement data processing on one hardware complex or to use a separate complex for each subsystem. Plug-ins are independent of each other and can be added or removed without violating the integrity of the system. The TOC subsystems described in the article can work on hardware complexes with different processor architectures, ways to store bytes in memory (big-endian or little-endian) and on different operating systems.

An algorithm for converting geographic coordinates derived from a mathematical model into the coordinate system of the visualization subsystem is presented. At the next stage of development of the TOC prototype, it is planned to develop and implement a system for modeling atmospheric phenomena and a system for procedural generation of the earth's surface.

## REFERENCES

- [1] R.Choquet, D.J.Cole A hybrid symbolic-numerical method for determining model structure, Elsevier B.V Volume 236, Issue 2, April 2012, Pages 117-125.
- [2] O. Grottko A. Ntoubas S. Ullrich W. Liao E. Fried A. Prescher T. M. Deserno T. Kuhlen R. Rossaint, Virtual reality-based simulator for training in regional anaesthesia, British Journal of Anaesthesia, Volume 103, Issue 4, 1 October 2009, Pages 594–600.
- [3] V.N. Gudivada, V.V. Raghavan, Content based image retrieval systems, IEEE : Computer, Volume: 28, Issue: 9, Sep 1995.
- [4] Allerton D. Principles of Flight Simulation, Chichester, UK, Wiley Publ., 2009, 471 p.
- [5] Kunc M., Morecroft JDW, Brailsford S. J. of Simulation. 2018, pp. 87–89.
- [6] Ljumović M. C++ Multithreading Cookbook. Packt Publ. Ltd, 2014, pp. 422.
- [7] Ian D, Richard H, Mike R. Companion Encyclopedia of Geography, published in the Taylor & Francis e-Library, 2004.s
- [8] Georgi K, Valentin S. Software Architecture for Implementation of Complex Simulation Systems. Cybernetics and Information Technologies. Volume 8, No 4 Sofia. 2008.
- [9] R.L. Carovillano, J.M. Forbes (Eds.) Solar-Terrestrial Physics. Series: Astrophysics and Space Science Library, Vol. 104. 1982.
- [10] Olsson O., Assarsson U. Tiled Shading. J. Graph. GPU Game Tools. 2011, vol. 15, no. 4, pp. 235-251.