Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Multicomputer Research Desks for Simulation and Development of Control Systems

M.Kh. Dorri * **A.A. Roshchin** *

* *Institute of Control Sciences RAS, Moscow, Russia, 117997, Profsoyuznaya 65,*
e-mail: dorrimax@ipu.rssi.ru

Abstract: This paper is devoted to the detailed description of development processes and features of multicomputer desks based on the development software tool RDS (Research of Dynamic Systems). Powerful capabilities of multicomputer desks developed with use of RDS are illustrated by different examples of control algorithms implementation for technical plants. The software tool RDS developed in Institute of Control Sciences RAS satisfies main design principles for multicomputer desks and facilitates creation of research desks for simulation and development of control systems. *Copyright ©2008 IFAC.*

Keywords: Software development tool, control algorithms, visual system, simulation.

1. INTRODUCTION

Multicomputer desks for simulation and development of control systems are the base for building research desks and training simulators in various technical fields (see Dyakonov and Pen'kov [1996]). This paper gives the detailed description of development processes and features of multicomputer desks based on the development software tool RDS (Research of Dynamic Systems). Powerful capabilities of multicomputer desks developed with use of RDS are illustrated by different examples of control algorithms implementation for technical plants. Use of specialized tools for building multicomputer desks facilitates achieving the following goals:

- Reduction of development terms and costs for operator control panel, training simulators, and research desks.
- Providing the opportunity for building and modification of desks without involvement of highly qualified programmers.

Implementation of the following basic principles is important for development of the research desks:

(1) *Generality*: availability for rather great quantity of homogeneous products and projects.
(2) *Flexibility*: ability of changing module parameters, interfere in computational algorithms in interactive mode.
(3) *Completeness*: sufficient quantity of heterogeneous interchangeable models of blocks and systems are to be included into the desk structure.
(4) *Authenticity*: the units included into the desk structure are to be provided with characteristics indicating the field of their applicability.
(5) *Virtual development environment*: the processes under study should be displayed as clearly as possible at available video displays (in form of diagrams, animation, etc.).
(6) *Temporal scale*: ability of studying processes in various temporal scales.
(7) *Network structure*: ability of connecting several computers into common network for design data interchange.
(8) *Connection to plant*: this feature is conditioned by need of adjustment of algorithmic and software tools for real devices.
(9) *Throughout design*: CAD-CAM technology in development of control systems meaning automation of transition from research algorithms to codes of actual computing devices.

The software system RDS developed in Institute of Control Sciences RAS satisfies main design principles for multicomputer desks and facilitates creation of research desks for simulation and development of control systems.

2. STRUCTURE AND FEATURES OF SOFTWARE SYSTEM RDS

The software system RDS allows simulating systems presented in the form of interconnected blocks, each of them can contain computing program defining its interaction with other system blocks and response to the user actions. Groups of functionally interconnected blocks can be combined into compound blocks (subsystems) that can contain their own programs in addition to the programs contained in the block subsystem. Furthermore, if necessary, the additional computational program can be connected to the whole system. All block programs are loaded from external libraries that can be modified regardless to the main program.

The hierarchical structure of simulated systems applied in RDS allows building computational algorithms by a way being a most convenient one for specific system. Data

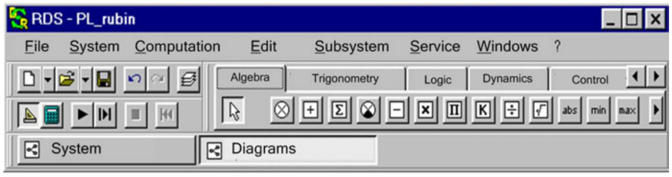

Fig. 1. The main window of software system RDS

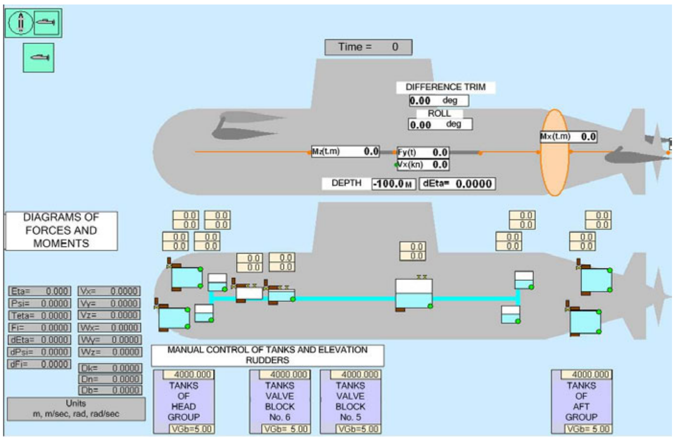

Fig. 2. Common circuit of submersible vehicle

directly related to a system block can be processed by computational program of this block. Data being common for block group can be processed by the subsystem computational program. Finally, data related to the whole system can be processed by the system program.

At that, RDS allows organizing interaction of these programs rather easily. Such approach enables to solve sufficiently large class of simulation problems.

A user sees the simulated system as a block diagram with connections between blocks represented by lines. Block appearance may be changed during simulation that allows informing user on computation results and system condition more clearly. The computation results can also be presented as numbers, diagrams, indicators, etc. Besides that, user can interfere into the system operation directly during simulation process by changing block parameters and observing the results of these changes.

RDS is provided with convenient ways of presentation for system blocks and their variation during dynamic process.

The system RDS is built under the operating system Windows by means of Borland C++ Builder. It has service abilities customary for most users.

The main window of RDS is presented at figure 1. It contains the main menu of the program (on top), control buttons (at the left-hand part), block panel (at the right-hand part), and buttons of open windows (at bottom).

Let us give an example of RDS application for development of the research desk for submersible vehicle (SV) that can provide presentation of abilities of software system RDS. Figure 2 presents the common circuit of SV.

User can simulate SV behaviour under various control actions and algorithms included into the desk. At that,

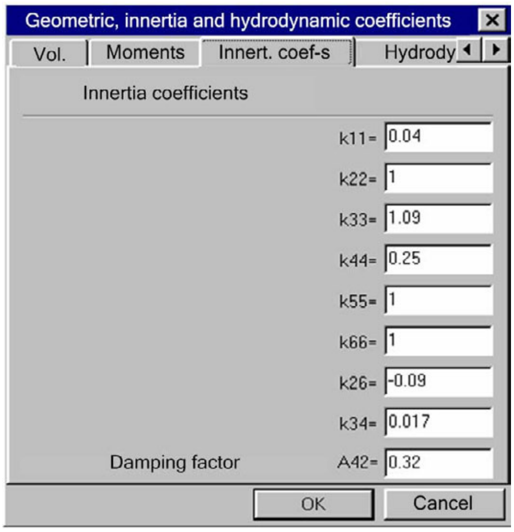

Fig. 3. Parameters setting window

SV condition can be displayed at diagrams and numeric indicators.

2.1 Changing settings of submersible vehicle

If needed, user can change various parameters of submersible vehicle simulated in the research desk, namely, initial values of depth, velocity and course, volume and mass of SV, moments of inertia, and others, by calling the setting window (figure 3).

2.2 Changing model of submersible vehicle

If the user's computer has one of supported compilers Borland C++ 5.5 installed, one can change working algorithm of the submersible vehicle model. RDS supports automatic compilation of system block models in language C/C++. Below, while considering block diagram of neutron capability control system for nuclear reactor, an example of automatically compiled block will be given (figure 8).

In addition to the model of SV, the research desk circuit includes also some quantity of automatically compiled models that can be overlooked and changed.

2.3 Configuration of layers and desk subsystem

Usually just after the desk run, the main circuit window opens that represents graphically SV behaviour and condition of its tanks. Not to crowd the picture, a part of the desk blocks is placed on invisible layers of the main circuit and in the subsystems opening in separate windows.

The subsystem of forces and moments diagrams characterizing SV motion simulation is situated at the left-hand part of the main circuit window (figure 4). By double clicking it, one can open the window containing diagrams of forces and moments affecting SV.

One of the SV steering control systems is presented at figure 5. If needed, this circuit can be modified by user.

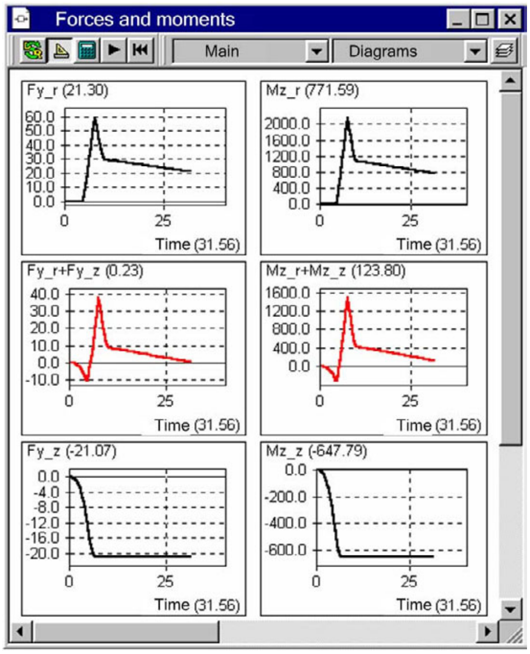

Fig. 4. Diagrams of forces and moments

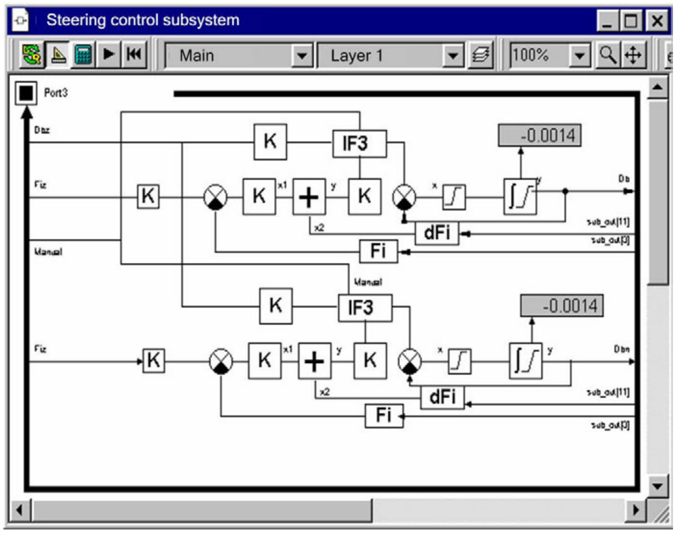

Fig. 5. Steering control subsystem

2.4 Circuit modification and block addition

The block panel serving for quick selection of standard blocks and their entering into the circuit is situated at the right-hand part of the main program window (figure 2). The panel consists of several tabs corresponding to the certain set of blocks ("Algebra", "Dynamics", "Control", etc.).

Each block in RDS has a set of variables that can be inputs, outputs, or internal block parameters. The inputs and outputs of the blocks can be connected one to another by the ties.

The research desk circuit widely uses buses. A bus is a set of independent data transmission channels that can be linked up by ties connecting these channels with blocks or other buses. Each bus channel, just as a block value, has a unique name that is usually indicated at the end of tie coming to the bus or leaving it. The buses allow not crowding the circuit by long ties crossing the circuit window from end to end and entering into subsystems. Irrespective of number of channels, a bus is always displayed as one line.

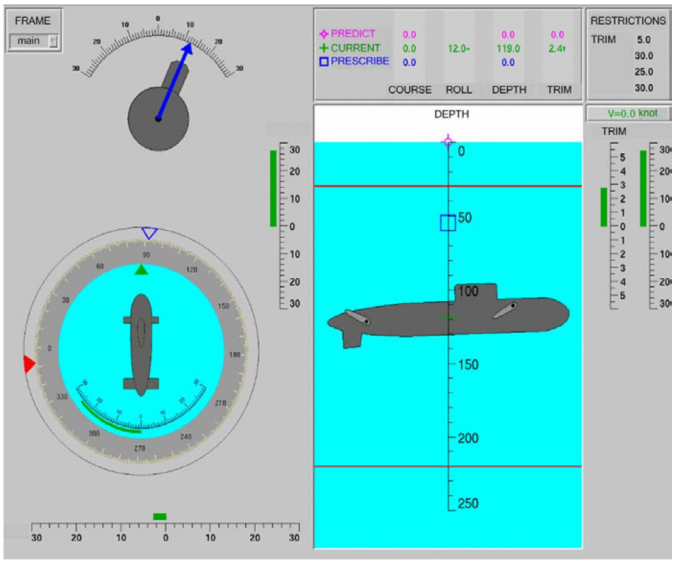

Fig. 6. Video frame with output of SV parameters on run

2.5 Connection of entry fields and indicators

Indicators, diagrams, and entry fields are the blocks, just as the main blocks of the control system. The respective buttons are situated at the block panel. The entry fields are connected to the inputs and parameters of the blocks. The diagrams and indicators are connected to the outputs.

2.6 Operator control panels

Figures 6 and 7 present view of operator control panels realized in RDS. At the left-hand part of figure 6, the object course is represented in common way as rotating compass card with coarse and fine reading. In addition, information on the course changing rate is presented. The object attitude with respect to rolling is displayed at the top left-hand side of video frame. Attitude with respect to depth and trim is displayed at the right-hand part of video frame. The object image moves vertically along the depth scale, simultaneously changing slope angle according to trim.

The unified panel at the top right-hand part of video frame serves for representation of fine current and prescribed values of all parameters. This panel is also used for entering the prescribed values. Entering into the line "Prescribed" is provided by means of keyboard. After that, one can select the parameter needed to be modified. Modification of the restrictions for the parameters and elevation rudders is made by the same way.

At figure 7, one can see video frame with SV condition representation. It gives minimum necessary but at the same time sufficiently full information on controls and SV position. To control the motion, operator has an opportunity to evaluate situation by 3-dimensional picture and accurately define the parameters by the scale instruments.

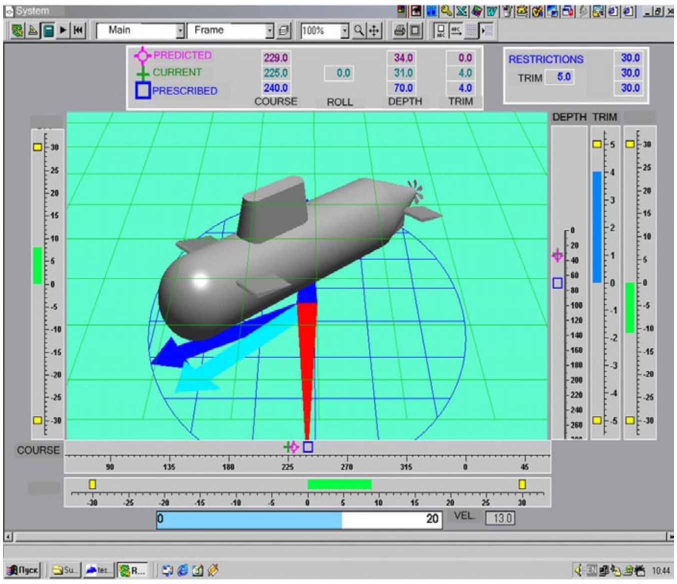

Fig. 7. Spatial representation of SV at operator control panel

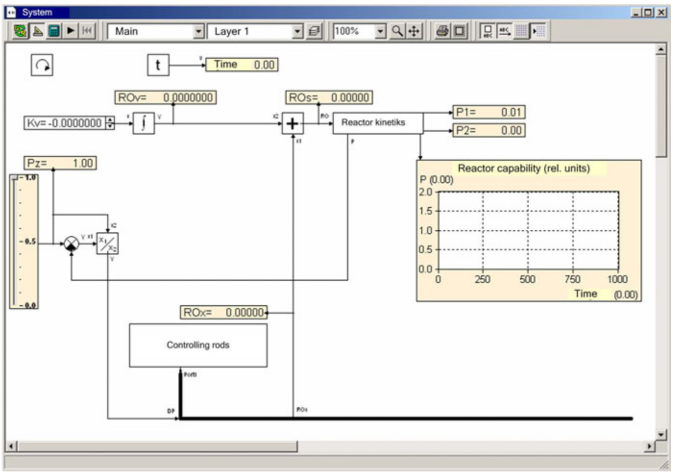

Fig. 8. Block diagram of neutron capability control system for nuclear reactor

Consider another example of the research desk development by means of software system RDS, namely, neutron capability control system for nuclear reactor. Fragments of its subsystems are presented at figures 8-10.

Figure 8 displays the block diagram of the simplest control system for neutron capability of nuclear reactor. The equations of the reactor kinetics and reactivity forming functions are realized in two blocks presented at the figure. There is reference input unit that allows changing the reactor capability level. The diagram reflects the dynamic process of neutron capability variation.

As an example of automatically compiled block, let us consider the approximate model of nuclear reactor kinetics. It is described by two differential equations

$$0.288\frac{dP_1}{dt} + P_1 = P,$$
$$0.0263\frac{dP_2}{dt} + P_2 = P,$$

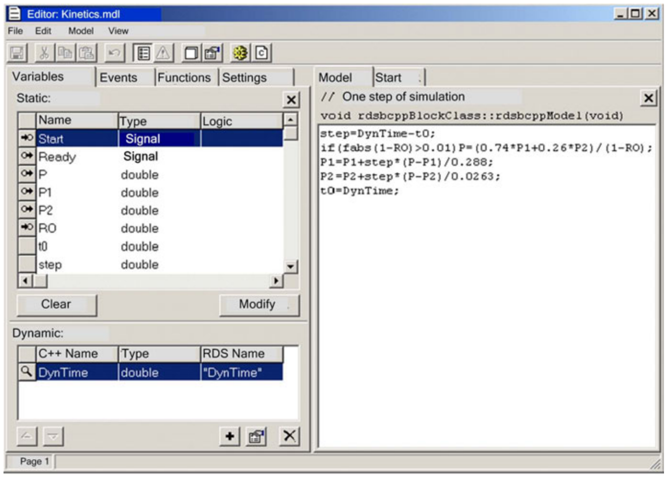

Fig. 9. Difference model of nuclear reactor kinetics

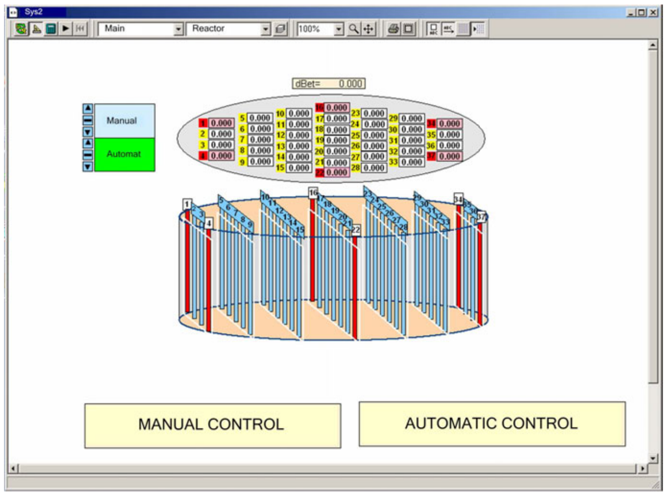

Fig. 10. Operating representation for neutron capability control rods

$$P = \frac{(0.74P_1 + 0.26P_2)}{(1 - RO)} \quad \text{if } |1 - RO| > 0.01,$$

where P is the reactor neutron capability measured in relative units, P_1 and P_2 are the capabilities of delaying neutron groups, RO is the reactivity.

The corresponding difference model for Euler's approximation is presented in figure 9, where `step` is the computation step. The model is written in syntax of language C++ and automatically translated into the needed dll-file. It is very convenient since does not require a researcher to be high-qualified programmer. Only knowledge of subject and computation algorithm is important.

The images of blocks shown at figure 10 allow observing motion of control rods in steady-state and emergency modes of the reactor working. The panel displays the values of reactivity brought by each of the rods, as well as their total value.

At figure 11, the manual control panel is shown. It allows debugging operation of computational blocks as well as carrying out test experiments.

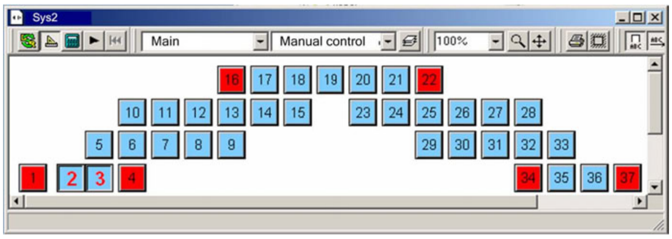

Fig. 11. Manual control of nuclear reactor rods

2.7 Computer networks

At present time, the software system RDS includes tools for interaction between several interconnected computers. Sometimes the need for such interaction is conditioned by insufficient computational capability of one computer, and sometimes — by a great number of operators observing simulated process and specifying control actions from their individual posts. Such a situation often appears during training simulator use.

While developing the software system RDS, we rather quickly came to the need for organization of network interaction between several systems running at different computers.

The system RDS being initially directed to reducing work of well-qualified programmers was added with service functions that allow communicating data of arbitrary format. On this base, several standard block models were developed that can be used without learning service functions themselves and methods of data communication via network.

In order that the systems can exchange data, RDS server has to be run at one of the local network computers. If such opportunity is available, one can run dedicated server at one of the computers. If the number of computers in network is small and non of them can be allotted for dedicated server, one of RDS copies simulating any system can be appointed as server. In such case, it is preferable to use program running at the fastest computer as RDS server since it has to not only simulate the system, but to serve network data transmission.

RDS has several finished models of data transmission blocks that allow user connecting systems via network rather easily. The blocks for receiving/transmitting data of main types - real, logical, signal, matrix and arbitrary - are used most often. In settings of each block, it is necessary to designate IP-address and server port (the default values given in RDS settings can be used), as well as name of data receiving/transmitting channel. If in one system such a block is connected with input tie and fed by any value, this value appears in all other systems at the outputs of blocks with the same channel name.

Besides the blocks mentioned above, one can use the block of group data transmission for data exchange between systems. It allows simultaneous transmission of data of various types that reduces overhead costs of transmission and thereby reduces the network load. For instance, transmission of two real numbers, one logical number, and one matrix requires four channels for transmission by ordinary blocks and only one channel using the group transmission

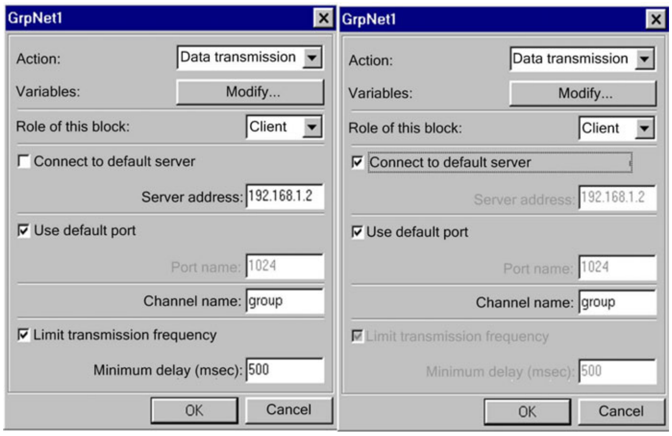

Fig. 12. Settings window for group data transmission block

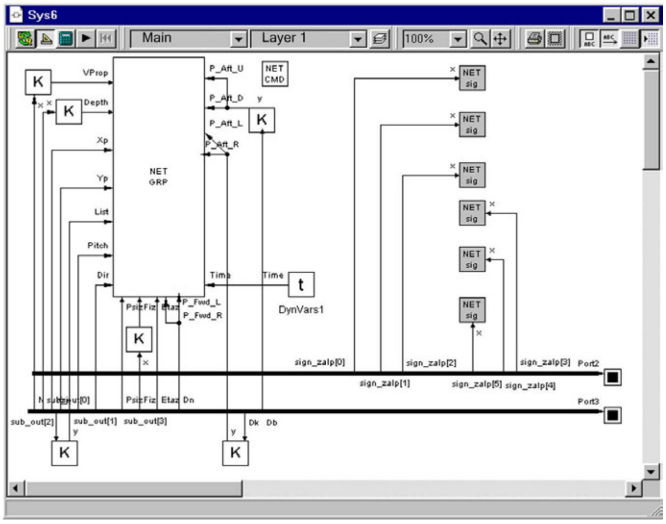

Fig. 13. Data transmission subsystem

block. Figure 12 shows the settings window for group data transmission block in data transmission mode (left) and in receiving mode (right).

Dual-computer research desk for simulation of SV motion under various controls and disturbances has been developed using the blocks described above. First computer simulates transient processes and displays SV parameters as mnemonic diagrams and plots. Second computer being connected with first one via network displays the operator control panel and 3-dimensional representation of SV spatial motion. Figures 13 and 14 show the subsystem of first computer transmitting data to second one by means of the blocks described above and its connection to buses of the main system.

As a result of addition of service abilities described above, the software system RDS get sufficiently universal and user-friendly tool that allows data exchanging between systems simulated on different computers.

3. CONCLUSION

The software system RDS (Research of Dynamic Systems) developed at Institute of Control Sciences RAS progressively becomes more and more convenient tool for development of research desks and training simulators (see

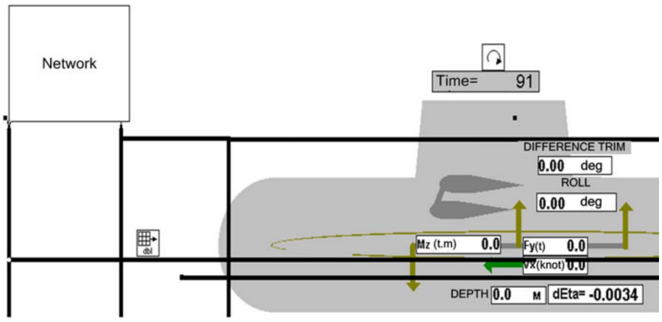

Fig. 14. Connection of data transmission subsystem

Dorri and Roshchin [2003], Dorri et al. [2006]). It assists in development of research desks that allow evaluating quality of proposed algorithms, clearly presenting processes in normal and emergency conditions, analyzing computation results, and modifying both control laws and representation form of occurring events.

Along with creation of concrete control systems, the authors for a long time have been engaged into development of software for simulation and research of control algorithms. One of the first projects was published in Jamshidi and Herget [1992].

Unfortunately, we can not give a reference to description of the system presented in this paper because it is a new product that has not been shaped as a ready commercial product.

The widely known software products — MATLAB, LabView, and various SCADA systems — can be considered as prototypes of the system considered in this paper. Having developed RDS, we emphasized 23 criteria that could be used for comparison of similar systems. But such consideration would be subjective, and it is not a topic of this paper. Undoubtedly, MATLAB excels all known systems of this type in saturation with first-class software tools solving various problems of control systems analysis and synthesis. In comparison with RDS, LabView and SCADA systems have gain in means of communication with real objects. They also have other advantages. However, if all criteria are considered, then, in our humble opinion, RDS is in Pareto set, that is, by some criteria, RDS is inferior to similar software tools, but, by other criteria, it excels them.

On the whole, RDS could be characterized by the following advantages:

- Ability of graphic representation of processes using animation.
- Creation of automatically compiled units in syntax of high-level languages.
- Availability of visible and invisible layers while creating and studying control systems.
- Convenient means for subsystem interaction by buses, as well as some other features that could be evaluated only by results of working with the development software complex RDS.

"A proof of the pudding is in the eating," as one English proverb says.

REFERENCES

V.P. Dyakonov, and A.A. Pen'kov. Modern mathematical systems. *PC WEEK/RE*, 29 October, 1996. (*in Russian*)

M.Kh. Dorri, and A.A. Roshchin. Instrumental software system for research of virtual dynamic systems. *Proc. of XXX All-Russian conference on motion control.* ICS RAS, Moscow, 2003. (*in Russian*)

M.Kh. Dorri, V.M. Korchanov, G.E. Ostretsov, and A.A. Roshchin. Development principles for full-scale research desks and ways of their implementation in software system RDS. *Proc. of XXXIII All-Russian conference on motion control.* ICS RAS, Moscow, 2006. (*in Russian*)

M. Jamshidi, and C.J. Herget. Recent Advances in Computer-Aided Control Systems Engineering. *Studies in Automation and Control.* Amsterdam – London – New York – Tokyo, 1992. V.9. P. 417–429.

Certificate of official registration of computer program. No. 2004611323, Research of Dynamic Systems (RDS), 2004. (*in Russian*)