New Forms of Man-Machine Interactions

Mattsson, Sven Erik

1985

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Mattsson, S. E. (1985). *New Forms of Man-Machine Interactions*. (Technical Reports TFRT-7293). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN:LUTFD2/(TFRT-7293)/1-007/(1985)

NEW FORMS OF MAN-MACHINE INTERACTIONS

STATUS REPORT 1985-09-30

STU project 84-5069

STU program: Computer Aided Control Engineering, CACE

Sven Erik Mattsson

Department of Automatic Control
Lund Institute of Technology
Lund, Sweden

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
Report

Date of issue
1985-09-30

Document Number
CODEN:LUTFD2/(TFRT-7293)/1-007/(1985)

Supervisor

Sponsoring organisation
The Swedish Board of Technical Development

Author(s)
Sven Erik Mattsson

Title and subtitle
New Forms of Man-Machine Interactions – Status report 1985-09-30

Abstract

This is a status report for the STU project 84-5069, "New forms of man-machine interactions". The project is a part of the program "Computer Aided Control Engineering, CACE", which is supported by the Swedish Board of Technical Development, STU.

The new workstations with high performance, real-time graphics now appearing on the market open new possibilities to man-machine interaction. In this project these possibilities are investigated by development of a prototype system. The prototype system is chosen to be a simulator for dynamical systems. The idea is to use hierarchical block diagrams to describe the decomposition of the model and the interconnection structure and to use continuous zooming to show internal detail. The simulator will accept models described by ordinary differential equations and algebraic equations. The possibility to enter the model in the form of equations both facilitates and makes the modelling simpler and the documentation becomes better.

Key words
Man-Machine Interaction; Computer Graphics; Simulation; Differential/Algebraic Systems; Hierarchical Blockdiagrams

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
7

Security classification

Recipient's notes

The report may be ordered from the Department of automatic control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

1. INTRODUCTION

This is a status report for the STU project 84-5069, "Nya former av man-maskin-kommunikation" (New forms of man-machine interactions). The project is a part of the program "Datorbaserade hjälpmedel för styrsystem (Computer Aided Control Engineering, CACE)", which is supported by the Swedish Board of Technical Development (Styrelsen för Teknisk Utveckling, STU).

The report is organized as follows. A background and the aims of the project are given in Section 2. The status of the project is described in Section 3. Possible extensions and continuations of the project are outlined in Section 4.

2. BACKGROUND AND AIMS

The man-machine interface is a very important part of a CACE-system. The new workstations with high performance, real-time graphics now appearing on the market open new possibilities to man-machine interaction. See for example the survey paper Marchover and Myers (1984). The purpose of this project is to set up a prototype system so that ideas can be tested and experiences of using graphics for man-machine interaction can be gained.

The prototype system is chosen to be a simulator for dynamical systems inspired by the ideas outlined by Elmqvist (1983). The idea is to use hierarchical block diagrams to describe the decomposition of the model and the interconnection structure. The system should allow continuous zooming to show internal detail. At the top level the user can just see named rectangles connected by lines. The user can move around the cursor with the mouse and by pressing mouse buttons when moving the mouse he can pan and zoom. If he starts to zoom in on a block, the block will open up and show internal details. It will show a combined textual and pictural representation of its interfaces and inside there will either be a new block diagram or when the lowest level is reached a mathematical description of the submodel. The user can also create new windows for viewing of the model. One of these windows are the interaction window in which the panning and zooming are performed. When panning and zooming in one window, rectangles in the other windows outline what part of a window that is shown in the interaction window. This facility helps the user to keep track of where he is. If the user want to move fast he can point on an object in another window and ask for automatic panning and zooming to this point in the interaction window. The user can display the result of a simulation in different windows. New windows are created and the layout are changed by choosing commands from a pop-up menu with the mouse. The mouse is then used to position and stretch the windows in a Macintosh-like fashion. New blocks, interfaces and connections are created in similar ways. Names of the blocks and the interface variables are entered via the keyboard. The mathematical description of the submodels at the lowest levels are entered and modified with a window-oriented editor. There is also a possibility to copy a block and all its subblocks. The model can be saved as a text file which can be read to recreate the model.

The prototype system will be a simulator for dynamical systems, which can be described by sets of ordinary differential equations and algebraic equations. The simulator will allow the mathematical description of submodels at the lowest level to be in the form of equations. This will facilitate the use of the simulator since it corresponds to the original formulations of the models as a basic set of mass- and energy-balances and other equations. There are also other important advantages with the equation form compared to the assignment form. The documentation becomes better in this way since the reader will recognize the equations. It will also be safer since it is easier to check that the model is input correctly and there is no risk of introducing errors during manual transformation to assignment statements. Furthermore, the equation form is the only reasonable if one want to build model libraries, because different environment of the submodel impose different causality relations so the transformation to assignment form of a submodel is dependent on its environment. Allowing equations we can have a more sophisticated connection mechanism. The simulator will allow two types of interface variables: across variables which are equal in the cuts (examples are voltages, pressures and temperatures) and through variables which have a direction and are summed to zero in the cut (examples are currents, flows, thrust and torque). Allowing equations make the simulator more complex than when just assignment statements are allowed. There are two major approaches to solve this. An implicit ODE-solver can be used straightforward to solve the problem or the equations can be manipulated symbolically to solve for the derivatives and auxiliary variables before integration. We intend to take an intermediate way and make some symbolic formula manipulation before solving the equation with an implicit solver. For example nonlinearities can make it impossible to transform the equations to assignment statements by symbolic manipulation. On the other hand it is advisable to make some kind of analysis and symbolic formula manipulation before using the implicit ODE-solver. A number of simple operations can be done which may reduce the order of the problem drastically. There are efficient procedures for choosing a unique variable for each equation. These procedures give indications if the system of equations are structurally over- or underdetermined. They can also detect algebraic relations between variables that also appears differentiated in the system of equations. It is of special interest to eliminate such variables, since systems of equations containing that type of algebraic relations may be be very difficult to solve numerically. Then there are also efficient procedures for partioning the equations into minimal subsets requiring simultaneous solution and for sorting the subsets into correct computational order. These procedures may drastically decrease the order of system of equation, since auxiliary variables are often defined by equations actually being on assignment form and the ODE-solver needs not solve for these variables. The connection of across variables leads to simple equations of type A = B, which are easy to eliminate. It is also simple to eliminate the equations resulting from connections of through variables since these equations are linear.

We have got the differential/algebraic system solver DASSL written in Fortran from Linda Petzold, Applied Mathematics Division 8331, Sandia National Laboratories, Livermore, California. DASSL has the reputation of being one of the best and most robust solver for differential/algebraic systems.

Besides that a simulator is an important part of a CACE-system, it can probably be achieved with reasonable effort, since software in Pascal from the Dymola package (Elmqvist, 1978) and the LICS-project (Elmqvist, 1985) can be used.

A workstation IRIS 2400 from Silicon Graphics has been purchased. The IRIS 2400 is a high performance engineering workstation designed for interactive color graphics and computing applications (Clark and Davis, 1983). The operating system is based on UNIX System V with C, FORTRAN 77 and Pascal compilers. The processor is a 10-MHz MC68010 with a floating point accelerator. The workstation is equipped with 1.5 MB CPU memory, 24 bit-planes image memory, a 72 MB Winchester disk and a 1/4" cartridge tape drive (60 MB). The display is a 19" RGB monitor with a resolution of 1024x768 pixels and a refresh rate of 60

Hz non-interlaced. The mouse has three buttons. There are also an Ethernet-interface with XNS protocol. The graphics is based on vector graphics using real valued coordinates. Graphical operations like 2-D and 3-D transformations (object rotations, translations, scaling, and perspective and orthographic projections), clipping, generation of cubic curves, filling of polygons, shading and removal of hidden surfaces are performed by 12 proprietary custom VLSI circuits called the Geometry Engine. The computational power of these 12 Geometry Engines corresponds to more than 6 Mflops (32-bit floating point). This means that graphic data is transformed, clipped and displayed at up to 65 000 coordinates/s. Up to 150 000 bit-mapped characters can be displayed per second. Polygons are filled at up to 44 million pixels per second with texturing. Lines are drawn at a rate of 3 million pixels per second with texturing. Parametric curves and surfaces are generated at less than 10 microseconds per point. The program interface to the graphics is the IRIS Graphics Library. It is callable in C, FORTRAN 77 and Pascal. There are routines for definition and manipulation of objects in (local) world coordinate systems and projection of these objects onto the screen.

3. STATUS

The workstation IRIS2400 was ordered per telex in October 1984 just a few days after this project was granted by STU and the workstation was ready for shipping in March 1985. However, export licence was not received until the end of May and we got the workstation on the 9th of July 1985. There were some minor errors in the interface between Pascal and the Graphics Library. We got a corrected version from Silicon Graphics Inc., but could not install it since the 1/4" cartridge tape drive was defective. Silicon graphics made a warranty exchange and a new drive arrived on the 6th of September 1985. The new drive works well.

Sven Erik Mattsson started the project in February 1985. Since the workstation was not available at that time, he started with the model and language part of the simulator and used the VAX11/780 computer at the department. The major part of the code for handling the equations is written and is now ported to the IRIS2400 workstation for testing and debugging. The design of the graphical part has just begun. Dag Brück participates in this work on part time (30%). Hilding

Elmqvist participates as a consultant. We hope to have a runnable prototype at the end of March 1986.

4. EXTENSIONS AND CONTINUATIONS

At the meeting of the steering committee on the 6th of September 1985 it was discussed how this technique of using graphics should be evaluated. It was suggested that a workshop should be arranged when the prototype is running. This is a good idea since different users will hopefully give us ideas how to improve the system, when they are asking questions like "I would like to ... and how do I do that?".

The first goal of this project is to get a simulator running on the IRIS workstation. When this goal is reached, there is a number of possible extensions.

Consider first the model and language part. In the first version of the prototype each submodel has its own model description implying that if two subsystems have the same model the model descriptions is duplicated. This means that maintenance becomes more complicated. This problem can be avoided by introducing model type concept which can be used to generate several models from one description. It may also be of interest to introduce more symbolic formula manipulation. A possibility to also deal with discrete time system is desirable.

The possibilities to present the results must be flexible. The user should be able to not only choose the scale of the axes but also for example how it should be marked. Another possibility is to simulate instrumentation and present the results in windows as if you were viewing them on real instrumentation.

An exciting idea is to present the results as some kind of animation of the process simulated. The IRIS workstation is for example capable of animating in real time a robot in shaded 3-D graphics so that the user can move around in this 3-D world and study how the robot performs its work cycles. The mouse can be used as an input device to control the movement of the user in this animated world. To make it simple for an user to design this kind of animation, an interface (language) must be designed so the users easily can enter descriptions of their desired 2-D or 3-D worlds and how its behaviour is related to the simulation model.

REFENCES

Clark, J.H. and Davis, T. (1983): Work station unites real-time graphics with Unix, Ethernet. Electronics, Oct. 20, 1983, pp. 113-119.

Elmqvist, H. (1978): A Structured Model Language for Large Continuous Systems. Ph.D-thesis. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden. Report CODEN: LUTFD2/(TFRT-1015)/1-226/(1978)

Elmqvist, H. (1983): Combining Graphical Descriptions and Equations for Dynamical Modelling. Fourth IASTED International Symposium on Modelling and Simulation, June 21 - 24, 1983, Lugano, Switzerland.

Elmqvist, H. (1985): LICS - Language for Implementation of Control Systems. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden. To appear.

Marchover, C. and Myers, W. (1984): Interactive Computer Graphics. Computer, Vol. 17, No. 10, Oct 1984, pp. 145-161.