MAN-MACHINE INTERFACE FOR REMOTE PROGRAMMING AND CONTROL OF NC MACHINE TOOLS AT THE TASK LEVEL: AN EXAMPLE

Francisco Moreira [a], **Goran Putnik** [b], **Rui Sousa** [c]

Production and Systems Engineering Department - School of Engineering

University of Minho, Campus of Azurém, 4800 Guimarães, Portugal

Tel: ++351 (0)53 510278 Fax: ++351 (0)53 510268

[a] Email: fmoreira@eng.uminho.pt [b] Email: putnikgd@eng.uminho.pt [c] Email: rms@eng.uminho.pt

Abstract

This paper presents an example of Man-Machine Interface for remote programming and control of NC machine tools. The system is based on a computer virtual panel which substitutes the machine control panel and in telematics interfaces which establishes the basis for geographical independence between the physical equipment and human operator.

Keywords : Man-Machine Interface, Internet, Teleoperation, CAM.

1. Introduction

The concept of a competitive enterprise today implies, among many others, a capability of flexible access to the best resources. To provide an “ideal” enterprise system, the domain of the selection/employment of resources needed, the “market” of resources, should be as large as possible. We can say, that the competitive enterprise today, and especially in the future, should be capable to manage all manufacturing functions independently of the distance. This is the main idea in developments of the concepts of Global Manufacturing Systems, Distributed Manufacturing Systems or Virtual Enterprise/Factory. This implies the highest level of communication among enterprise’s resources (to be integrated and being integrated), performed in an organised multimedia and information environment in a real-time, that is, using telematics technologies, which can be seen as a key for the future competitiveness. In this paper we present one of our results in development of industrial services based on telematics technologies, especially telematics interfaces for CAD/CAM/CAPP/PPC, which establish, so called, teleservices/teleworking concept.

2. Conventional and Teleoperated CAM System

A conventional milling machine, as an object of teleoperation, has many design problems. Milling machines are designed and made with the assumption that the human operator locally operates the machine control unit panel, Fig. 1a). DNC integrated with CAD/CAM systems do not change the way machines are operated in terms of hypothetical, or possible, (globally) distribution of system components locations. They improve the way part program for machining are built and the way those programs are sent to the machine, Fig. 1b), but the components’ locations are still limited within the local area, which is demonstrated by the use of Local Area (Communication) Network - LAN.

The hierarchical model of control structure for traditional equipment control, based on the model presented in 1979 by Barbera, Albus and Fitzgerald [BARBERA ET AL. 79], is showed in Fig. 1c) (this model is important to understand which control function are subjected to “distribution”). When operating a machine at distance the operator could execute a sequence of “button touches” in a remote and virtual panel, generating, in real time, the expected actions on the machine that is apart from the operation site. The hierarchical control structure for this new system includes some higher levels of control that are apart from the lower levels control functions. We developed a CAM system based on teleoperation at the task level, given in Fig. 2a). This system corresponds to the hierarchical control structure presented in Fig 1d). The informal representation of the logical structure is shown in Fig. 2b) and the corresponding formal representation, in ESTELLE, in Fig. 3. The perception of the state of the machine is also assumed to be recognised by the senses and knowledge of the operator, thus requiring transmission of useful information from machine control unit and external sensors.

System Specification

The formal specification of the system using the Formal Specification Technique (FDT) ESTELLE (Extended Finite State Machine Language)[1] is given bellow. The complete description would be very large, so only a small part is presented here. ESTELLE was chosen because it is particularly adequate to describe distributed or concurrent systems, and models a system as a hierarchical-structured set of communicating non-deterministic state machines (automata). The automata concept is adequate for the specification of production systems. Even the behaviour of the human operator module in this system could be modelled with FDT's (see examples in [Turner 93]). The architecture of the teleoperation system specified by ESTELLE is in fig. 3. Each box is a module and modules communicate through channels with bi-directional capability.

[1] Although especially developed for the field of telecommunications there is nowadays an increasing interest in the use of FDT's in other areas like robotics, security, finance, production systems, software engineering, etc.

The formal specification is given through channel and module description. The channel description includes the definition of the messages that are allowed to flow in that channel. The channel between Interface1 module and WAN module is defined as:

```
channel Interface1Connection(Interface, Network);
    by Interface:
            MediumInformation(Medium: MIType);
    by Network:
            ReceiveProgram;
            RunProgram;
            Program(file: GProgramType);
```

Messages ReceiveProgram and RunProgram carry no parameters and are only allowed in channel Interface1Connection if originated by the module that plays the role of Network (module WAN in this case). Message MediumInformation carries a parameter Medium of type MIType, which is in fact a record with three fields.

```
type MIType=        {Medium information type}
    record
        Image: ImageType;
        Sound: SoundType;
        Temperature: real;
```

The types ImageType and SoundType are defined in a type declaration as binary files. The type GProgramType in channel description is an ASCII text file containing the "G-program" for machine-tool control unit. Unidirectional channels are obviously allowed and that is the case of the channel inside MachineTool module, between modules MachineActuators and ControlUnit.

```
channel MachineActuatorsServer(Actuators,Control);
    by Control:
            DriversOn;
            DriversOff;
            Xposition(XValue:real);
            Yposition(YValue:real);
            Zposition(ZValue:real);
            Spindle(SValue:real);
            Feed(FValue:real);
```

Specification of a module requires a header definition and a body definition. It is possible to have more than one body to use with the same header. When instantiating the module the appropriated body is chosen. The header definition includes the module's class (process or activity), interaction points, channels involved and, eventually, exported variables.

```
module ControlUnit systemprocess;
    ip
        A: MachineActuatorsServer(Control);
        S: MachineSensorsServer(Control);
        C: MachineToolServer(Control);
end; {Control unit header}
```

Interaction point A is associated with channel MachineActuatorsServer where the module ControlUnit plays de role of Control.

The body specifies the behaviour of the module, based on finite state machine concept. The transition between states depends on the current state and current input. Usually a transition produces some kind of output, for example an output message through one interaction point of the module.

```
body ControlUnitBody for body;

    state RECEIVING, RUNNING, STOP, …
    …
        trans
            …
            when C.RunProgram
                from RECEIVING to RUNNING
                    begin
                    …
                        output A.DriversOn;
                    …
                    end;
            …

    end; {ControlUnitBody}
```

In this formal specification are defined the data types (G program, image, sound, analog data, binary data) exchanged between modules ***Interface3***, ***MachineTool*** and ***Sensors***. All these data types are used in the definition of ***I3L*** channel.

3. Man-Machine Interface

When machines are to be operated at distance a friendly and robust Man-Machine-Interface (MMI) is required. The problems, emerged from the fact of the person in control being apart from the manufacturing equipment, can be minimised this way.

The remote user of the system must have at each instant the notion of being distant, and, at same time, the illusion of having a real machine panel and a real machine in front of him. This way he will be better prepared to deal with new problems he will face. These new problems arise from communications need in both directions. From machine to remote operator the state of machine: visual and acoustic information, and many other binary and analogue data, like temperature, vibration, machine protection ON/OFF, tool ON/OFF, etc. From remote operator to machine, the commands and programs needed to operate the machine and produce the pieces.

As mentioned by Alan Dix [Dix 94] there is no easy answer to the question “what makes a good interface?”. However, Man-Machine Interfaces, like machine control unit panel, have been built for decades. These machine control unit panels, independently of respective builders, have some common characteristics. LED’s indicate the status of the machine, rotating buttons for selection of digital values or machine operation mode and other pushing buttons for emergency machine stop, program start, keyboard and screen for program editing and monitoring, etc. The panel distribution of this buttons, the colour and size has a correspondence to their functionality and actuation way.

In the implementation of the remote interface the similarities to the local Man-Machine-Interface are obvious. Almost all the buttons have the same visual appearance, the same disposition in the **virtual control unit panel**, and are operated in the same way, although they are now operated with a mouse click instead of a finger touch or hand rotation. This virtual panel is much like the original machine panel, Fig. 4. The functionality’s of the virtual buttons are similar to the physical buttons, but *they hide the conversion to commands generated with button touches in a remote PC monitor and the communications required to send those commands from the remote PC to the machine*.

As consequence the Man-Machine-Interface (local operator-milling machine interface) is substituted by a Human-Computer-Interface, HCI, (remote operator-computer interface) and two Machine-Machine Interfaces (local computer to remote computer interface and local computer to milling machine interface), Fig. 2c).

4. Implementation

A real-time prototype of remote programming and control was implemented on a FANUC 0-M machining centre. This machine is located in the Mechanical Laboratory at University of Minho and the operation room is located in the Laboratory for Automatic Production Systems located in another building of the same complex. Mitsuishi [MITSUISHI et al. 92] describes a similar installation: between a machining centre in Tokyo and a operation room located in the same building, and the same machine controlled from George Washington University. Considering the hierarchical model of the control structure for NC machine tools, the "distribution" of control function is implemented between the level of machine tasks - machine programming (CONTROL LEVEL 4), and the level of interpolation and auxiliary functions (CONTROL LEVEL 3). Of course, for a machine tool case, considering functional requirements of precision and axis' speeds, it is not possible to consider any "distribution" at that level and below, but for the higher levels, the "distribution" of functions, namely system control functions, is welcomed (which is under development).

The development tool used in the construction of the programs was LabVIEW 4.1 for Windows 95. This program development tool uses a graphical programming language – the "G" Language. LabVIEW has the basic graphical elements for building virtual equipment panels. These basic elements are LED's, pressing buttons, rotating buttons, displays, graphs and so on. This elements are combined in a way to form the application Front Panel. All the programming tasks are done in the Block Diagram, which is a pictorial solution to a programming problem. LabVIEW programs are called Virtual Instruments (VI) because their appearance and operation can imitate actual instruments.

The front panel of the program that implements the remote operator-computer interface is shown in the Fig. 4. This program is called the Client VI(**RemoteController**). In the machine site there is another PC that implements the local machine-computer interface. This program is called the Server VI (**Interface1**), and since human does not operate it, it has no front panel. An RS232C communication link connects the machine to the local computer. The Server VI and the Client VI implement, as well, the computer-computer interface, which is working on a Wide Area Network (WAN), and was implemented under TCP/IP protocol, usually known as Internet protocol.

When the milling machine is turned ON and the server application is running in the local PC, Client applications (apart from the machine) can ask for working sessions. To do that a Client establishes a connection to the Server and begins working. The Client application can operate the milling machine in two modes: the manual mode and semi-automatic mode. In manual mode the remote operator makes some "clicks" in the buttons of the virtual machine panel and those "clicks" are converted in commands that will be sent to the local PC, via Internet, and from local PC to milling machine via RS232C. In semi-automatic mode, the remote operator selects a program from a library of programs, located in the remote PC, confirms that the program to be sent is really that one, by watching the code in the virtual machine panel, and "clicks" a button to send and execute the program in the machine.

5. Conclusions

Teleoperation can enable new production paradigms to take place in physical factories. Advanced concepts of production systems, based in global distribution of manufacturing equipment, can establish a new era in the way people do their work in industrial environments. The structure to implement this is based on interfaces, required from the fact that person in control is apart from the equipment location: telematics interfaces and man-machine interfaces are required. A PC based man-machine interface with similarities to the traditional control panel of machines takes advantage of the developed familiarity with buttons types, disposition and functionality, and, at the same time, hides the complexity (in terms of network functions and commands generated through button touches) that his behind the virtual control panel.

Regarding our future work, besides the further developments and optimisation of the system presented, we think two technical issues could be important to mention:

1) The relation between the system we developed and open architectures for NC controllers.

2) New standards for machine interfaces could be considered to enable easy construction of this kind of systems.

The project is being developed at the Centre for Production Systems Engineering (CESP-Centro de Engenharia de Sistemas de Produção) at University of Minho, Portugal, within the Virtual Factories/Distributed Production Systems project (VF/DPS).

References

[BARBERA et al. 79] BARBERA, A.J., ALBUS, J.S., FITZGERALD, M.L., "Hierarchical Control of Robots using Microcomputers ", 9th Industrial Symposium on Industrial Robots, 1979.

[BLATTER and DANNENBERG 92] BLATTER, Meera M., DANNENBERG, Roger B., "Multimedia Interface Design", ACM Press, 1992.

[DIX 94] DIX, Alan, "The Human Interface", Assembly Automation, Vol. 14 no. 3, 1994, pp. 9-13, MCM University Press, 1994.

[MITSUISHI et al. 92] MITSUISHI, Mamoru, NAGAO, Takaaki, HATAMURA, Yotaro, KRAMER, Bruce and WARISAWA, Shin´ichi, "A Manufacturing System for the Global Age", Human Aspects in Computer Integrated Manufacturing, pp. 841-852, Elsevier Science, 1992.

[NI 96a] NI, "LabVIEW User Manual", National Instruments Corporation, 1996.

[NI 96b] NI, "LabVIEW Communications VI Reference Manual", National Instruments Corporation, 1996.

[SHERIDAN 92] SHERIDAN, T.B., "Telerobotics, Automation and Human Supervisory Control", MIT Press, 1992.

[TURNER 93] TURNER, KENNETH J., (ed.), "Using Formal Description Techniques - An Introduction to ESTELLE, LOTOS and SDL", John Wiley & Sons.

a)

b)

c)

d)

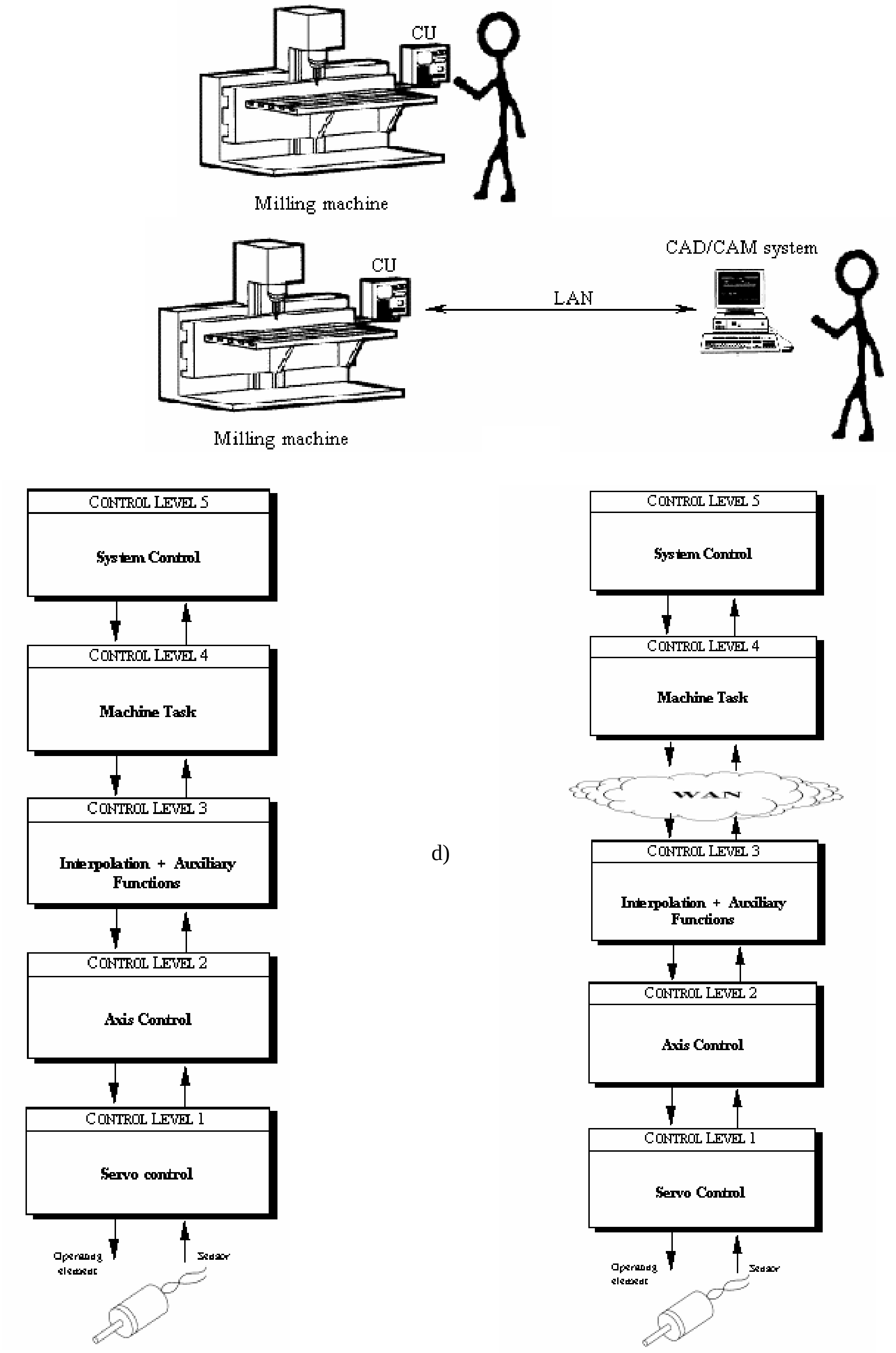
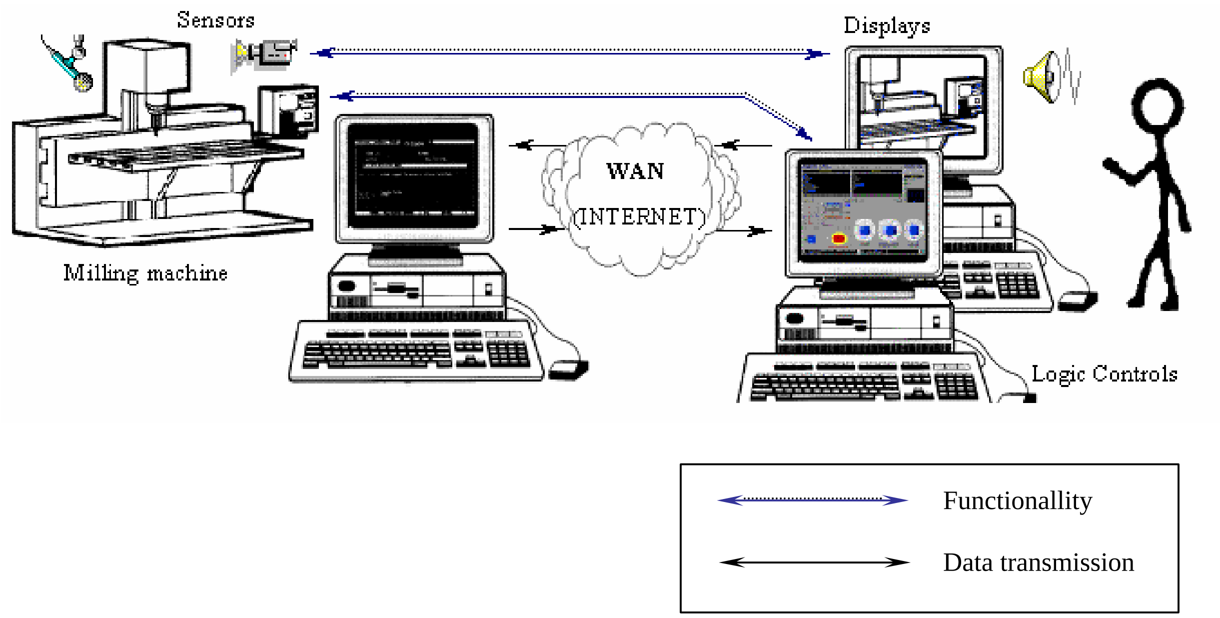
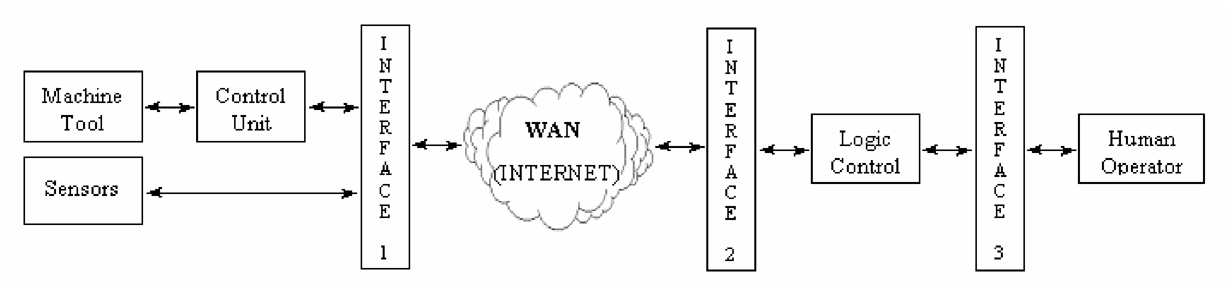


Fig. 1 a) CNC system with local control
b) DNC integrated with CAD/CAM system with local area communication network (LAN)
c) Hierarchical control structure of a milling machine
d) Hierarchical control structure of a remote operated milling machine



a)



b)

Fig. 2 a) Remote programming and control of a milling machine
b) Logical struture of interfaces for teleoperation of NC machine tools (unformal representation)

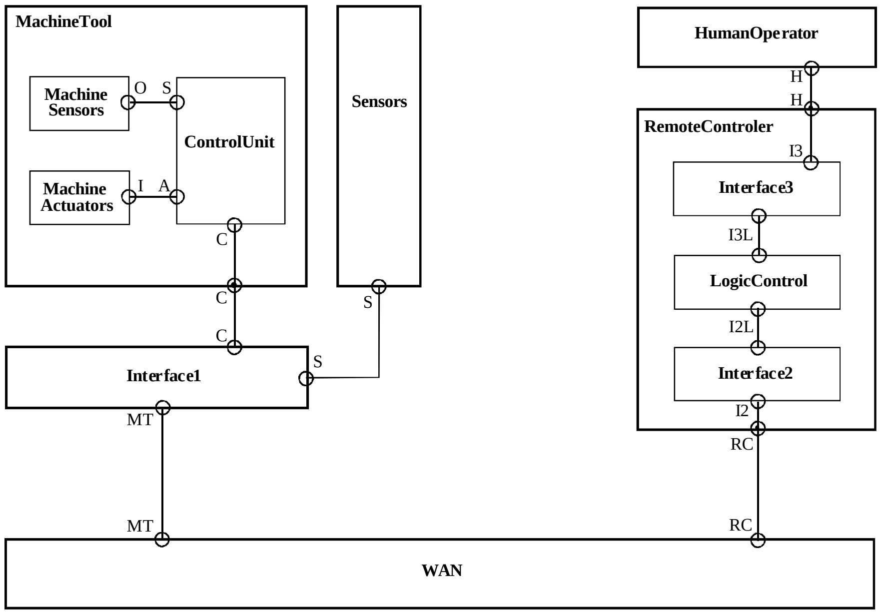


Fig. 3 ESTELLE formal representation of CAM systems based on teleoperation at the task level.

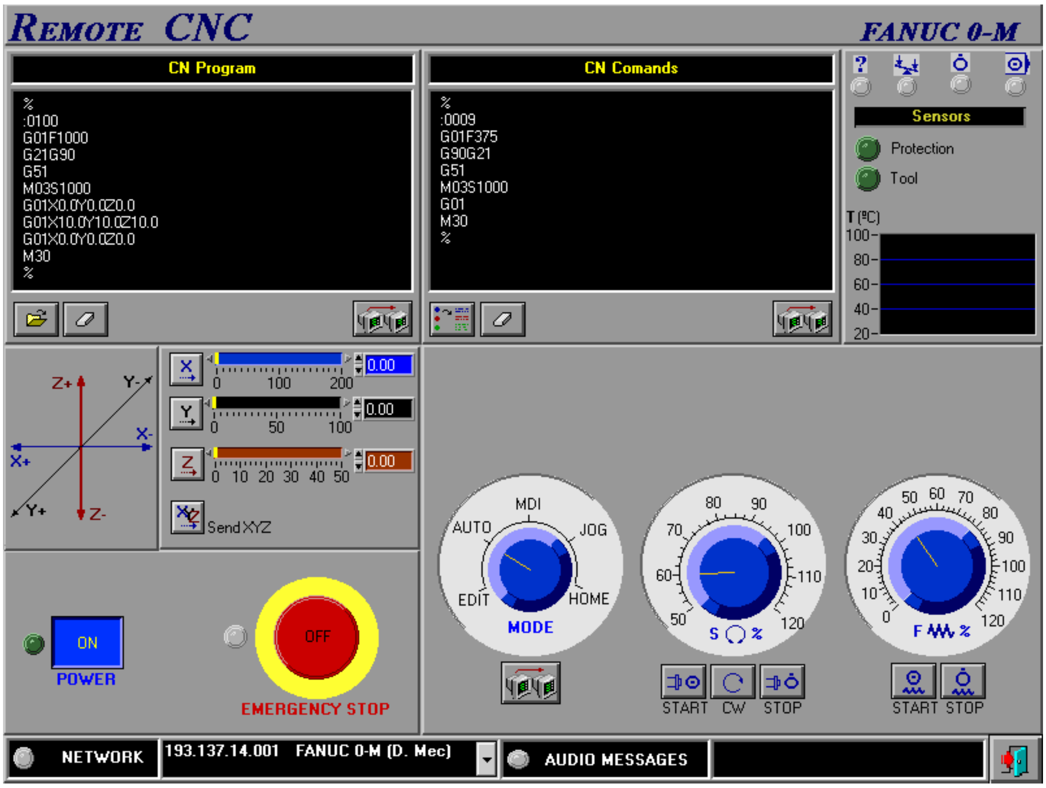


Fig. 4 Virtual panel of the FANUC 0-M at the remote site.