EUROPEAN FUSION DEVELOPMENT AGREEMENT
JET

EFDA–JET–PR(01)05

B Haist
D Hamilton

A Rational Approach to Remote Handling Equipment Control System Design

A Rational Approach to Remote Handling Equipment Control System Design

B Haist, D Hamilton.

EURATOM/UKAEA Fusion Association, Culham Science Centre,
Abingdon, Oxfordshire, OX14 3DB, UK.

A Rational Approach to Remote Handling Equipment Control System Design

Bernhard Haist, D Hamilton.

Euratom / UKAEA Fusion Association, Culham Science Centre,
Abingdon, Oxfordshire, OX14 3DB,

ABSTRACT

The approach to the design of remote handling control systems for JET (Joint European Torus) has undergone a metamorphosis in recent years. The original concept of specifying functional requirements and tendering external companies without specifying the hardware and software that was permitted, resulted in a myriad of solutions which, although functionally correct, presented JET with a number of unforeseen problems:

- In-house engineers had to become familiar with a wide variety of hardware and software.
- Operators were presented with many different computer interfaces, which had a different look-and-feel, with the robots sometimes behaving in a contradictory manner (e.g. positive rotation being clockwise on one robotic transporter and anti-clockwise on another).
- Developments and fixes to the control systems usually needed long lead times and were sometimes impossible or prohibitively expensive.
- A large selection of hardware spares was needed to provide cover for all the systems.
- Support for obsolete hardware and software was increasingly difficult.

The breakthrough came from the recognition that, although the remote handling equipment was specifically for and unique to JET, at a low level, the controls required were similar to industrial applications. Thus, widely available commercial products could be configured to handle the highly critical motion control, and implemented in a quite uniform control system framework. The high level control software, oriented about the objects and tasks involved, was developed in-house. At this level, the robotic systems have much in common, and a lot of the software modules could be shared or re-used.

1. INTRODUCTION

1.1 The JET Project

The Joint European Torus (JET) is the world's largest experiment on thermo-nuclear fusion, the energy producing process which takes place in the sun. After successful experiments with smaller reactors, the European Community Fusion Programme, co-ordinated by Euratom, decided to build a larger reactor to concentrate the research efforts. Work on the design of JET started in 1973, and in 1978 construction began in Culham, near Oxford, in the UK. Operation started in 1983. Since January 2000 the United Kingdom Atomic Energy Authority (UKAEA) has managed the JET facilities on behalf of the European Fusion Development Agreement (EFDA), with scientists from all over Europe visiting to undertake experiments.

To gain energy from nuclear fusion, nuclei from light elements have to join together to form heavier ones, releasing energy, high-speed neutrons and/or protons. JET is the only machine currently able to study the most interesting fusion reaction, which involves two heavy isotopes of hydrogen: deuterium and tritium.

$$D + T \rightarrow {}^{4}He + n + 17.58\ MeV \qquad (1)$$

This reaction is the least difficult to achieve, because the energy barrier for the joining nuclei is (relatively) low, but still requires heating of the plasma to temperatures over 100 million degrees Celsius [1].

1.2 The Remote Handling Group

The plasma is contained by magnetic fields within a toroidal shaped Ultra High Vacuum vessel. Over time, the released high-energy neutrons render all components and support structures of the reactor radioactive. Furthermore most plasma facing tiles are covered in Beryllium, which, if breathed in as dust, poses a further hazard to anyone working inside the reactor. Therefore, JET always placed great emphasis on its Remote Handling group, to ensure a maximum of tasks can be carried out fully remotely.

During the 1980s Remote Handling engineers at JET designed large robotic transporters, which were to carry out remote handling tasks, either inside or outside the torus The remote maintenance work is performed using a dexterous Master-Slave manipulator transported on the end of a 10m long articulated robot. To gain access to the inside of the torus, two of the eight main horizontal ports are reserved for remote handling, and the Articulated Boom (BOOM) features a total of 16 joints (see fig.1) to allow it to “snake” it’s way through the narrow ports and around the Vessel. A second Boom works in parallel with the first to transfer components and tools to and from storage facilities outside the torus. Other robots are designed for Ex-Vessel work, like the Telescopic Articulated Remote Mast (TARM) with 29 joints, which is suspended from the main 150 ton gantry crane (see fig.1).

2. A RATIONAL APPROACH

2.1 The Problem

The remote handling transporters and their control systems were build by external companies from around Europe over the period from 1984 to the mid 1990’s, and each of the suppliers came up with different solutions for both the hardware and the software technologies used for the control systems. The various control systems were based on computing hardware such as Mikro-Vax and GE Series6 PLCs, running Ladder-Logic, Forth, Assembler, through to C, and possessed primitive man-machine interfaces (MMI) such as 2 line LCD displays and MS-DOS based monochrome graphical interfaces without mouse pointer, most fronting an ascii serial line as the communication interface.

This caused problems not only for the operators, who were presented with many different computer interfaces, which had a different look-and-feel, but also for the support engineers, who had to become familiar with many combinations of hardware and software, and found that support for obsolete hardware and software became increasingly difficult. The overall performance of many of the control system features was limited, but changes and upgrades to the control systems proved often impossible or prohibitively expensive.

2.2 The Solution

When during 1996/1997 a new remote handling transporter, the Tile Carrier Transfer Facility (TCTF) [2] was designed and built, it became necessary to re-think our control strategies, in order to equip the TCTF with a state-of-the-art control system, with a view of migrating a successful solution onto the already existing control systems.

The breakthrough came from the recognition that, although the remote handling equipment was specifically for and unique to JET, at a low level, the controls required were similar to industrial applications. Thus, widely available commercial products could be configured to handle the highly critical motion control, and implemented in a quite uniform control system framework. The high level control software, oriented about the objects and tasks involved, was developed in-house. At this level, the robotic systems have much in common, and 95% of the software modules could be shared or re-used.

This new approach has many advantages:

- Critical motion control software is commercially proven.
- Critical motion control hardware and firmware does not become obsolete, it can be upgraded as necessary.
- New systems and new features can be developed quickly and confidently, and are immediately available on all robots.
- The computer interfaces can be made more uniform and intuitive. Therefore, less emphasis can be placed on training the operators in equipment use, and more on performing the task.
- Dependency on external companies is removed, as systems can be maintained in-house.
- Due to the hierarchical approach component types at any level can be changed, without invalidating the overall design (e.g. amplifiers, sensors, I/O-cards, …)
- Systems are designed to use common components where possible, so that fewer spares need to be kept.

An extensive market survey showed that the Programmable Multi-Axis Controller (PMAC) from Delta Tau Data Systems Inc. was the preferred solution for the low level motion control hardware. The PMAC is a high-performance servo motion controller based on Motorola's DSP56001 CPU, which can serve a wide variety of applications, due to the fact that it can be configured for a particular application by choice of hardware set (through options and accessories), configuration of parameters, and the writing of motion and logic programs [3].

3. THE CONTROL SYSTEM DESIGN

3.1 The control system architecture

At the heart of the control system is a VME-bus rack containing a Motorola 68040 host computer running a real-time operating system (OS-9), a Pentland board for digital and analogue I/O, and up to four daisy-chained PMAC1 8-axis motion controllers (see fig.2).

The host computer communicates with the PMACs via a Dual Port Random Access Memory (DPRAM) interface and is connected to a Local Area Network (LAN). A MMI running on a desktop (or portable) PC communicates with the host software using TCP/IP protocols.

Two RS232 serial lines run from the host to a terminal server, so a PC may become either an OS-9 console or an error log terminal, or optionally one serial line can be used to connect a serial joystick instead of the analogue joystick.

In addition, the PC may run the PMAC executive software which communicates directly with the daisy-chained PMACs over an RS422 serial line.

Multiple graphic workstations and other utility programs may connect to a task running on the host computer which provides them with real-time robot position updates (joint current and target positions).

3.2 The host software architecture

The high level control software on the host (see fig.3) consists of multiple processes which communicate with each other via signals and shared memory modules. The central status-task is programmed as a state-machine, gathering all information from the PMACs and the I/O board, continuously evaluating all status information, determining the current state and sub-modes, and relaying the complete system status to the MMI. The command-task has its own socket connection to the MMI, receiving, checking and executing commands when initiated by the user. Additional tasks perform specialised duties, such as sampling the joystick, driving the error log console, serving the graphic workstations’ socket connections, performing resolved motion calculations, and spawning or killing the entire host application.

3.3 The PMAC configuration

The PMAC motion controller boards run a proprietary operating system, and contain on-board firmware for axis position control and reference trajectory profiling. After initial connection of a joint’s amplifier, position sensor, brake etc. to the dedicated ports at the PMAC (often using specific accessories needed for our non-standard solutions, such as radiation-hard position sensors), the joints are tuned and tested.

Additional software, in the form of so-called Programmable Logic Control programs (PLCs), has been written to provide functions specific to the control of the remote handling transporters, such as safety checks on disabled joints (un-commanded move), brake control, emergency stop shadowing, automatic swapping of a joint’s set-up parameters when for example a different end-effector is mounted, redundant position sensors, and many more.

3.4 The Man-Machine Interface

A single MMI has been written using C++/MFC which has a compiler switch to become the front end of any JET Remote Handling control system (see fig.4). It connects via two TCP/IP sockets to the command-task and status-task on the host. The status telegram arrives several times per second and drives all the display of the graphical user interface. Each joint and the various global modes have their own display area, from where the user by visualising a target position edit box via mouse click or by dragging and dropping a joint over the joystick mimic intuitively drives (sets of) joints.

Two further windows inside the MDI-application display the error/info log display and the Repeat-File display, from where pre-taught sequences of moves are played back.

A dedicated remote handling control Ethernet network guarantees turnaround times for telegrams and acknowledgement between MMI and host of <20ms at any time, and is achieved by installing multiple network adapter cards in each PC.

3.5 The Graphic Workstation

At the heart of preparing and executing remote handling operations lies the remote handling graphical workstation, running virtual reality software (see fig.5). It displays a 3-dimensional model of the remote handling robotic equipment in its working environment, with all joint positions being updated online, whether the control systems are in motion or simulation mode. Furthermore at the beginning of each point-to-point move a wireframe representation of each transporter visualises the target positions, thus giving an early indication of potential problems which will be encountered during the move.

To ensure a correct match of the real world joint positions with the graphically simulated ones, the main position sensors (resolvers, which are radiation hard and give an absolute position at power-up) are linearised inside the control system using compensation look-up tables, which were originally calibrated offline using a high precision encoder.

Collision detection algorithms are used inside the remote handling graphical workstation software, which currently only alert the user, but are intended to be fed back to the control system for immediate processing.

3.6 Error Handling

The transporters used in the remote handling at JET are generally large and heavy, but their operating conditions are spatially constrained and in close proximity to sensitive and expensive equipment. Safety in error handling is therefore paramount. This is reflected in the large number of error checks and the mostly drastic action taken upon encountering any such error.

Error checks are performed at all levels (PMAC configuration, PMAC PLCs, host, MMI and graphic workstation) and include:

- Tracking error (for enabled joints)
- Un-commanded move (for non-enabled joints)
- Soft-limits and limit switches
- Detection of loss of MMI network connection
- Collision detection inside Virtual Reality software
- Redundant position sensor consistency checks
- Specialised algorithms to prevent folding up of robot-joints to the point of contact
- MMI command integrity checks

For severe errors the action taken is to put the entire system into Safe-state, which comprises of immediate cut of all servo power, engaging of brakes, and opening of the internal emergency stop circuit.

4. CONCLUSIONS

After successfully applying the new remote handling control system to one remote handling transporter with a total development time of at least 2-3 man years, the same approach was then transferred to all other remote handling transporters and applied to each in a matter of months.

The reliability, functionality and usability of all remote handling transporters have vastly improved since the introduction of the new rational approach to control systems [4]. Operators have to be trained on one MMI only, which is intuitive and complies with the normal Windows look and feel. Hardware support engineers similarly have to be trained only on the small set of technologies involved, and spares stocks have been significantly simplified and reduced [5]. Software support is also greatly simplified, as almost all software modules are universally applicable. Because new functionality can now be added swiftly and confidently, the link between operation and development engineers has become a very close one, leading to continuous iterative improvements. The overall benefits of the new rational approach to control systems for the Remote Handling group at JET can not be over-emphasised.

ACKNOWLEDGEMENTS

The rational approach to remote handling control system design was undertaken by and owes much to all members of staff of the Remote Handling group at JET. Outstanding contributions were made by:

Giorgio Benali	who initiated the new approach and led it through the prototyping stages
Mike Irving:	who designed the hardware architecture
Justin Thomas:	who initiated many improvements from the user perspective
Alan Rolfe:	who provided vision and leadership throughout

This work was performed partly under the Framework of the JET Joint Undertaking and partly under the European Fusion Development Agreement (EFDA). The latter part was funded under the JET Operation Contract by EFDA.

REFERENCES

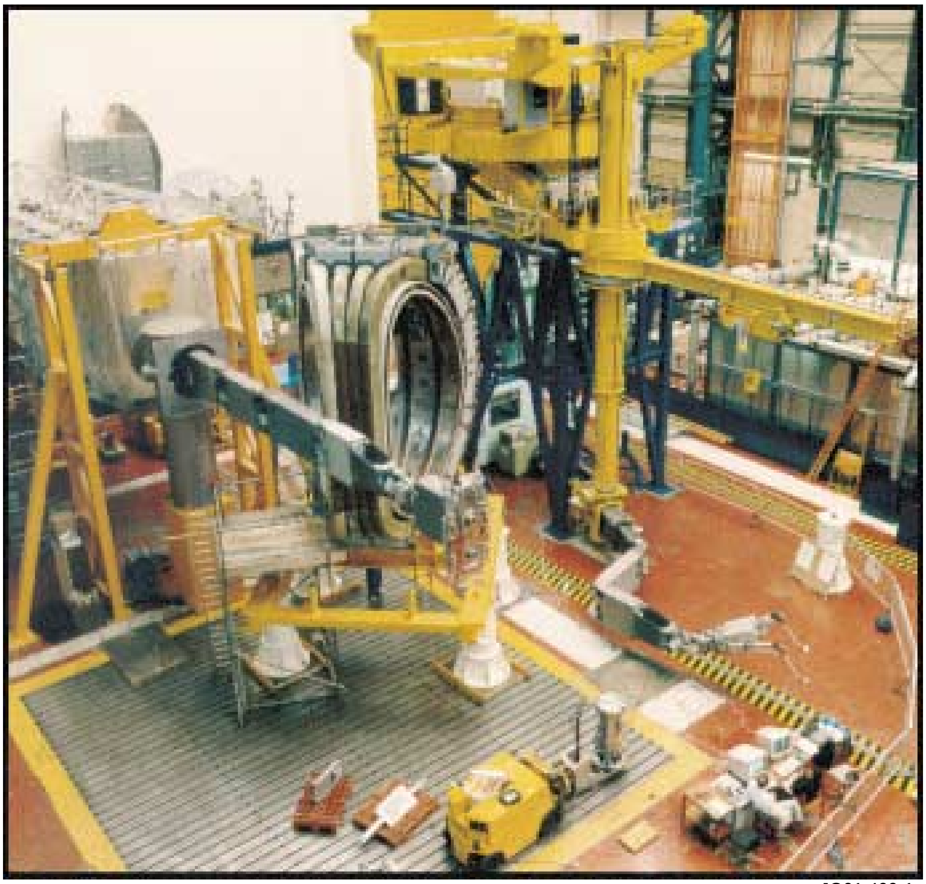
[1] “Grundlagen der Kernfusion”, Kernfusion – Berichte aus der Forschung, Isabella Milch, Folge 1, Max-Planck-Institut für Plasmaphysik, Garching, 1995

[2] Lawrence Jones, et al, “Design and Development of a new transporter facility for JET”, 19th Symposium on Fusion Technology, Lisbon, 1996

[3] Delta Tau Data Systems, Inc., “PMAC User’s Manual”, Chatsworth, 1998

[4] Rodney Cussack, et al, “Operational Experience from the JET Remote Handling Tile Exchange”, 20th Symposium on Fusion Technology, Marseille, 1998

[5] Leandro Galbiati, et al, “Experience from Remote Handling Support during the JET Remote Tile Exchange”, 20th Symposium on Fusion Technology, Marseille, 1998



JG01.409-1c

Fig.1: The BOOM, TARM and ROLLT (Remotely Operated Low Level Transporter)

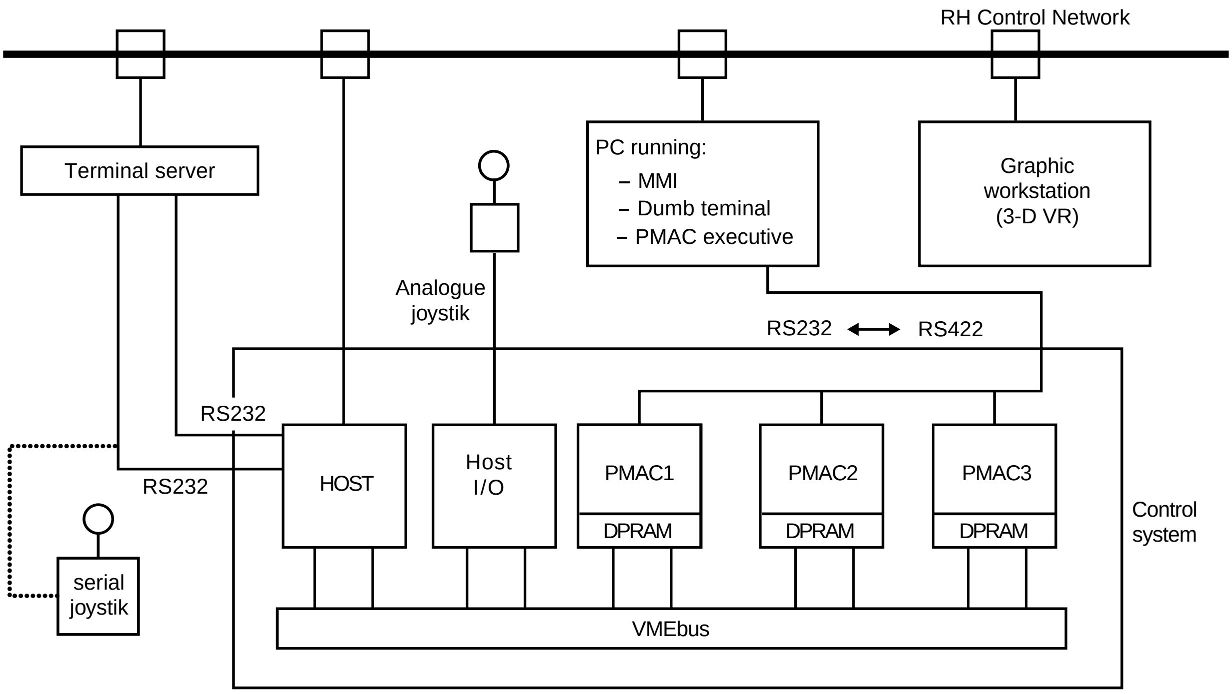


Fig.2: The computer system architecture

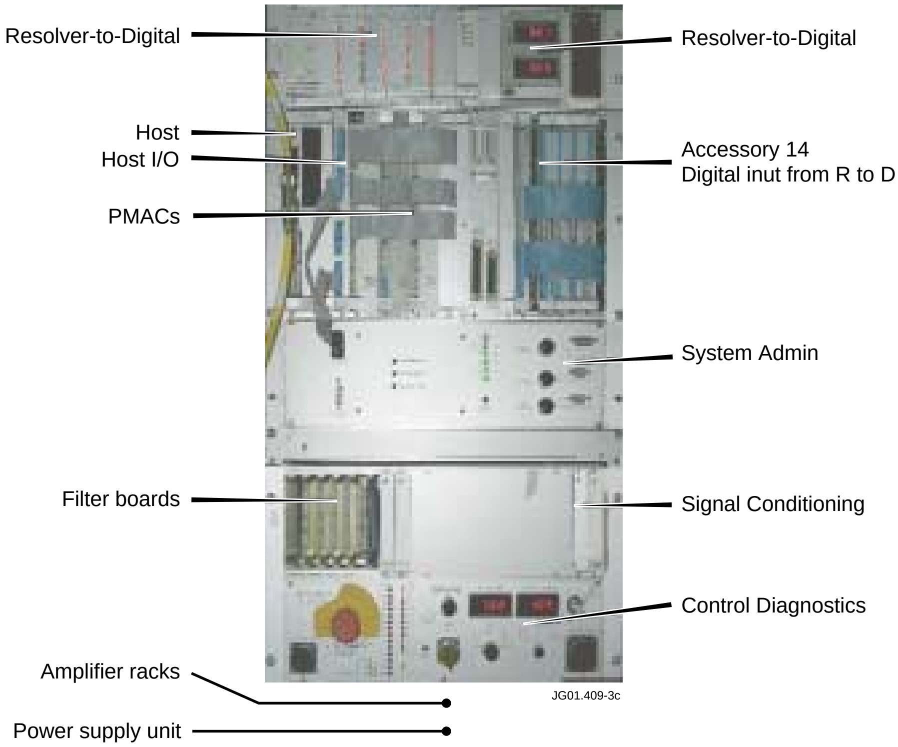


Fig.3: The BOOM control system cubicle

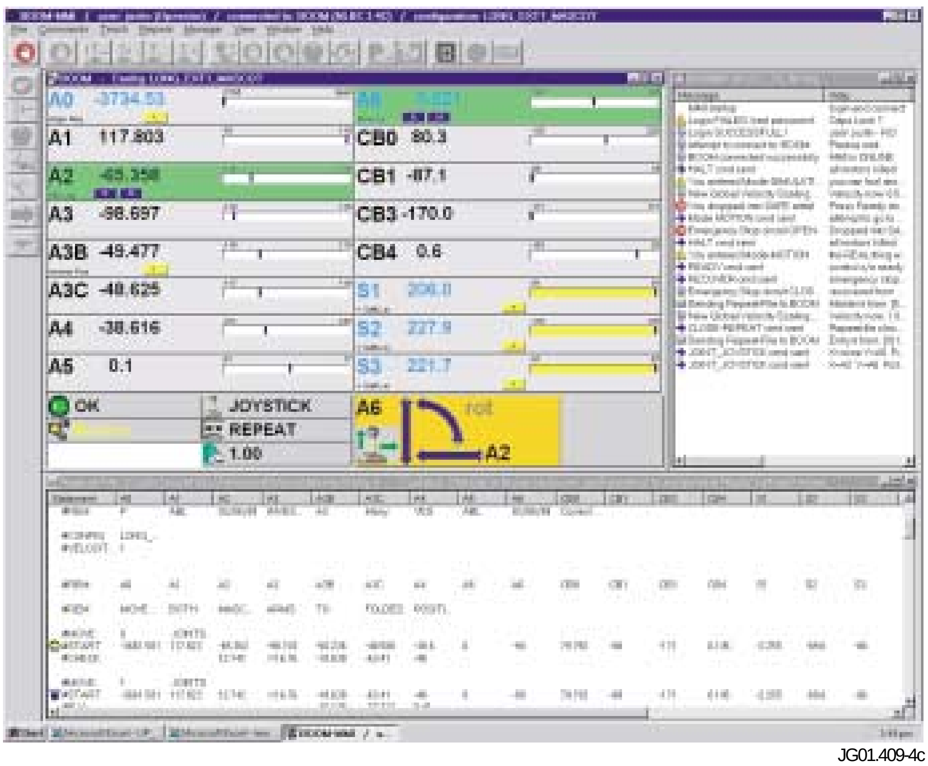
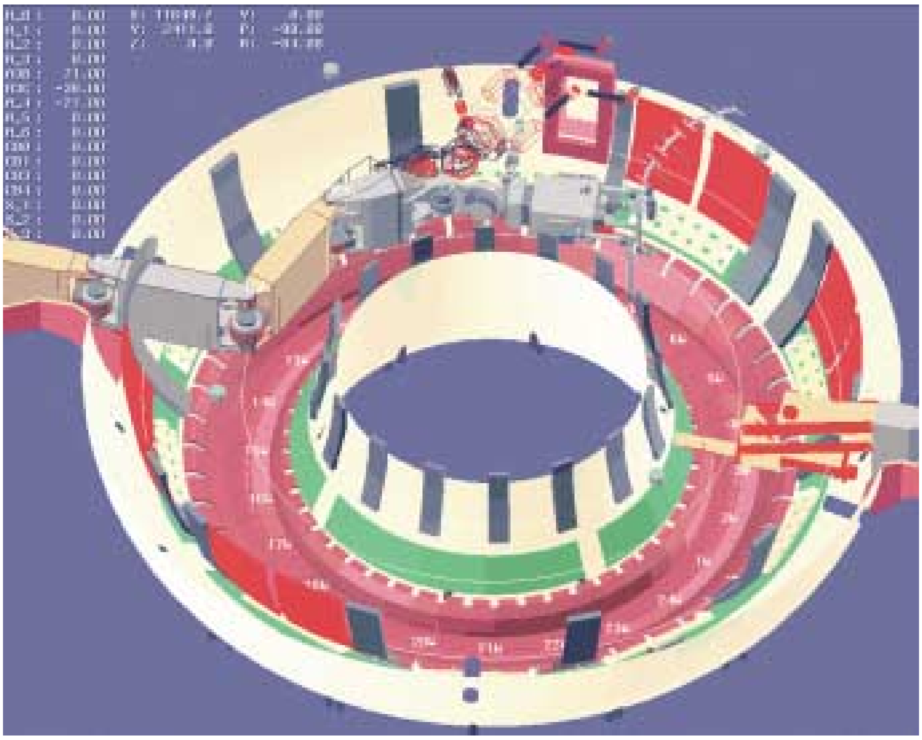


Fig.4: The remote handling MMI for all control systems



JG01.409–5c

Fig.5: The remote handling graphic workstation, running KISMET