MADE: an Effective Workbench for the Development of Digital Signal Processing Applications

D. Bulfone, M. Lonza, C. Scafuri
Sincrotrone Trieste, SS 14 - Km 163.5, 34012 Basovizza, Trieste, Italy

Abstract

A powerful workbench called MADE (MAtlab-DSP Environment) has been developed to boost the design, installation and testing of feedback systems at ELETTRA.

Starting from the Matlab commercial package running on UNIX workstations, a set of extensions allows to interact through the Ethernet network with VME based DSP systems installed in the field. The ELETTRA control system Remote Procedure Call routines have also been integrated in Matlab. Such extensions merge input/output, mathematical, graphical and signal processing features within the same software environment. The structure of MADE is described.

1 Introduction

ELETTRA, the Italian third generation synchrotron light facility, is now in its fourth year of routine operation and provides the user community with one of the world's brightest source of VUV and soft X-ray radiation.

To take full advantage of the machine potentiality, feedback systems are being designed and installed to cure multibunch and orbit instabilities respectively. The recent development of the electronic technologies allows us to use DSPs (Digital Signal Processor) in the implementation of these feedback systems, with a dramatic improvement in terms of performance and flexibility. A powerful workbench based on Matlab called MADE (MAtlab-DSP Environment) has been developed to support the design and operation of DSP based systems.

2 Matlab

Matlab [1] is an interactive computing environment suited for scientific and technical applications. In the field of control systems mathematical tools for the design and simulation are very important since they allow to deal with most of the control concerns before applying the feedback to the physical system. The Matlab toolboxes "Control System" and "Signal Processing", together with "Simulink", represent a comprehensive environment particularly suited for this purpose.

An extension of Matlab to interact with the physical system opens new interesting opportunities for a better understanding of the machine and gives an effective tool for quick development of feedback systems.

3 The DSP systems

The DSP systems adopted are based on the VME bus standard (fig. 1). The CPU board is a Pentek model 4284 with one DSP microprocessor (Texas TMS320C40). An additional CPU board with a Motorola 68030 microprocessor acts as a bridge between the DSP and Ethernet. The analog input/output and the DSP boards are connected by a mezzanine bus (Modular Interface eXtension, MIX), while the VME bus is used for the communication between the DSP and the bridge board. The MIX bus is an open, non-proprietary standard adopted by Pentek for fast inter-board communication while leaving the VME bus free for other purposes. The programs running in the DSP are written in "C" language. A complete development environment and a special Ethernet communication protocol called respectively SwiftTools [2] and SwiftNet [3] (by Pentek) allow to compile, download, run and debug the programs in the DSP from UNIX workstations.

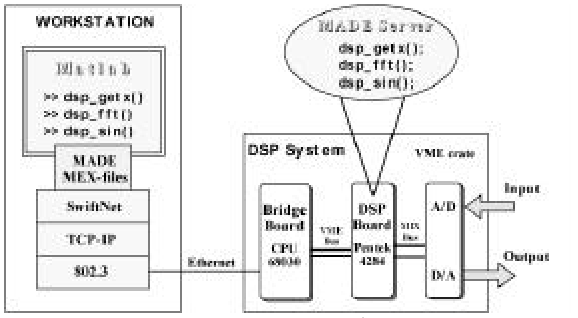

Figure 1: schematic of MADE

4 Made (MAtlab-DSP environment)

A set of new Matlab commands has been developed to remotely operate the DSP system directly from the Matlab workspace. Each command is implemented as a MEX-file, which is the usual way adopted to interact with programs and data external to Matlab. A MEX-file is a C language program which can be executed from the workspace and can exchange data with it. The UNIX workstations running Matlab and the DSP systems are connected by an Ethernet LAN where each DSP is identified by a unique "node name". The link between the MEX-files and the DSP relies on SwiftNet: this communication protocol includes an Application Program Interface (API) library which implements a client-server communication with a Server program running on the DSP. Each MADE MEX-file executed from the workspace command line is associated to a Server routine.

The MADE commands allow to:

- set-up the I/O boards (sampling frequency, anti-aliasing filters, etc.)
- acquire I/O signals
- generate output wave-forms (sinus, square, white noise, arbitrary wave-forms)

• acquire data pre-processed by the DSP (spectra, statistics, frequency responses, etc.)
• set-up digital filters
• set-up a down-sampling of the input data streams
• set-up a slow monitoring of the input signals vs. time by averaging the I/O signals and read the stored arrays of data.

Any of the DSP systems connected to the Ethernet can be addressed from the same Matlab session by simply changing a workspace string variable called "TARGET" that contains the node name.

The MADE Server program has been written for the Pentek 4284 DSP board and three different Pentek I/O boards (models 6102, 4243 and 4252). The hardware dependent part of the program, represented by a few C functions, can be easily modified to fit any other Pentek CPU or I/O board.

MADE will be soon adapted to become also a linkable library for the development of C language application programs.

5 Integration of the general control system with matlab

The ELETTRA Control System is a distributed computer system based on a layered architecture [4]. Programs running at the presentation layer interact with the lower level by means of Remote Procedure Call (RPC) besed libraries.

The integration of the ELETTRA Control System with Matlab is based on the existing RPC libraries for general accelerator control and on standard UNIX services (fig. 2). The specific control libraries which are made available in the Matlab environment are the "equipment access library" and the "beam orbit library".

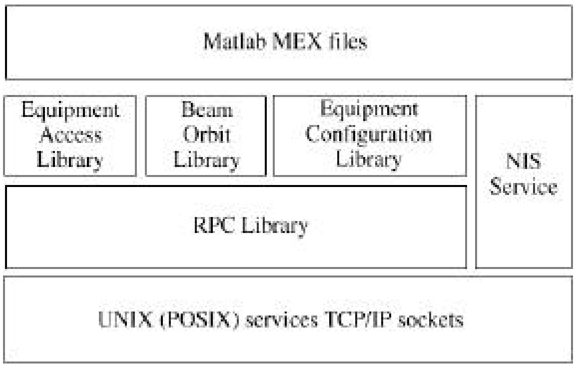

Figure 2: software layers for the integration of the ELETTRA Control System in Matlab

The equipment access library allows to read/set values from/to any machine equipment. Each equipment is identified by a name which is a simple character string. The library calls automatically perform all the network connection steps. First of all a Network Information System database (NIS - formerly yellow pages) is queried to find the network address of the so called "configuration server". This provides the network address of the ELETTRA Control System Local Process Computer handling the named equipment. The connection to the field is finally established using this address and the remote procedure is called. Although this scheme seems rather involved, it has good response times and proved to be extremely robust and easy to use.

The beam orbit library allows to read the beam position at the location of 96 electron Beam Position Monitors. The technique used for the integration of this library is identical to the previous case.

6 A case study: the local orbit feedback

To fully exploit the high-brightness photon beams generated by ELETTRA, a very high level of stability is requested for the accumulated electron beam. A digital local orbit feedback has been developed to stabilize the electron orbit at each photon source point.

A couple of dedicated photon Beam Position Monitors (phBPM) and four steering magnets act respectively as sensors and actuators. The phBPM signals are sampled by A/D converters and processed by a DSP system which implements the control algorithm. The D/A converters re-transform the resulting output samples in analog signals which drive the power supplies of the steering magnets [5].

The design and implementation of the local orbit feedback system have been effectively supported through all their phases by MADE (fig. 3).

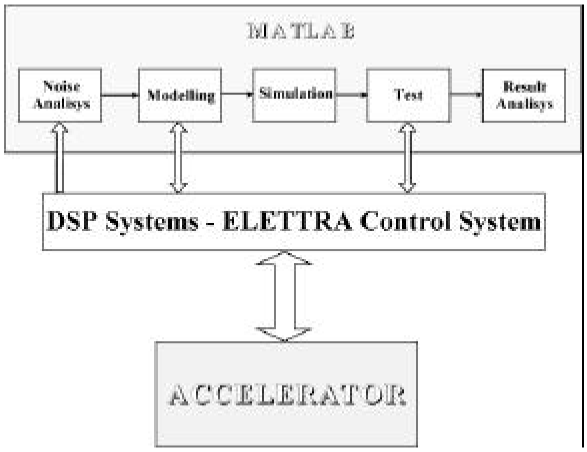

Figure 3: the phases of the Local Orbit Feedback project

First of all a spectral analysis of the beam noise has been done. Then the dynamics the overall system made of power supply, magnet, vacuum chamber, beam and phBPM has been characterized: the frequency response has been measured on the real machine by stimulating the system with sinusoidal signals at different frequencies. A polynomial model in *z*, which best fits the frequency response, has been calculated and several closed-loop simulations have been carried out using different control algorithms. This phase is particularly important because it allows to optimize the control parameters and try new control techniques before applying them to the real machine. The analysis of the results achieved with the real machine during the test phase showed good agreement with the simulations.

One of the main requirements of this kind of feedback is to keep "local" the corrections in order to move the orbit only at the photon source point [6]. To evaluate the "leakage", i.e. the global orbit perturbation, the integration with the ELETTRA Control System has been used.

All the operations needed to perform a measurement sequence have been coded in M-files (Matlab language scripts and functions) and the results saved in variables or files, or immediately displayed and analysed. Other M-files have been written to support the operation of the local feedback, like the empirical response matrix calculation, preparation of the controller before closing the loop, check of the open/closed loop spectra, etc..

The following is an example of M-file which prepares the DSP system before closing the loop and evaluates the feedback efficiency by comparison of the open/closed loop spectra. The statement in *italic* is directed to the ELETTRA Control System, while the statements in **bold** are MADE commands.

```
TARGET='lfb22'            %DSP system node name
cutoff=100;               %low-pass filter cutoff
                          %frequency
Kp=3;Ki=0.01;Kd=10        %PID parameters
b_matrix=meas_bmatrix()
                          %calls an M-file function
                          %that calculates the bump
                          %matrix
lfb_start                 %starts the DSP system
current = mequip('DCCT_S4_CURR_RR')
                          %reads the machinecurrent
                          %from the ELETTRA Control
                          %System
lfb_current(current)      %lets the DSP know the
                          %current
lfb_write([0 0 0 0])      %sets DACs to zero
lfb_setpnt([0 0 0 0])
                          %sets setpoints to zero
pause(2)                  %waits two seconds
lfb_setpnt(lfb_read(10000)')
                          %sets the setpoints to
                          %the current position
[b, a] = butter(1, cutoff/4000)
                          %calculates a Butterworth
                          %filter coeficients
lfb_lowpass(-b, a)        %sets the filter
                          %coeficients
lfb_pid(Kp, Ki, Kd)       %sets the PID parameters
lfb_bumpv(b_matrix)       %sets the bump-matrix
subplot(2,1,1);           %selects the upper part
                          %of the plot
lfbm_spec(3, 0, 400)      %plot the spectrum from 0
                          %to 400Hz
title('Loop OPEN')        %writes the plot title
lfb_closepid(10000)       %closes the loop
pause(5)                  %waits five seconds
subplot(2,1,2);           %selects the lower part
                          %of the plot
lfbm_spec(3, 0, 400)      %plot the spectrum from 0
                          %to 400Hz
title('Loop CLOSED')      %writes the plot title
```

7 Conclusions

An extension of Matlab called MADE has been developed to support the design and implementation of DSP based feedback systems. The hardware framework is based on VME DSP systems connected to the UNIX control room workstations by Ethernet.

The integration of the ELETTRA Control System with Matlab completes the MADE I/O capabilities.

The experience gained with a real application, the digital local orbit feedback system, demonstrates that the overall development time can be greatly reduced using a comprehensive workbench where I/O features are merged with mathematical and graphical capabilities.

Future projects of digital feedback systems will be carried out using the same workbench.

References

[1] 'Matlab User's Guide', The MathWorks, Inc. 24 Prime Park Way, Natick, Mass. 01760-1500, 1994.

[2] 'SwiftTools DSP Development Software User's Guide', Pentek, Inc. 55 Walnut Street, Norwood, NJ 07648, 1995.

[3] 'SwiftNet Host Software User's Guide', Pentek, Inc. 55 Walnut Street, Norwood, NJ 07648, 1995.

[4] D. Bulfone: 'Status and Prospects of the ELETTRA Control System', Nucl. Instr. and Meth. A 352, p. 63, 1994.

[5] C. J. Bocchetta et al.: 'The Design of the ELETTRA Fast Local Feedback System', Proc. PAC97, Vancouver, BC, Canada, 1997.

[6] C. J. Bocchetta et al.: 'First Operational Results with the ELETTRA Fast Local Feedback System', Proc. PAC97, Vancouver, BC, Canada, 1997.