DSP Station, a HyperCard environment for DSP sound processing algorithms.

Hugues VINET, Olivier KOECHLIN, Didier BRISSON
INA-GRM
116, av du President Kennedy
75766 PARIS cedex 16
FRANCE
Tel : (33-1) 42 30 21 82
Fax : (33-1) 42 30 49 88

Abstract: DSP Station, a HyperCard environment for the design and the use of DSP6001 sound processing algorithms is demonstrated. Working with the Sound Tools™ hardware, it allows the implementation of real-time and deferred time sound transformations between various sources: analog and digital signals, sampled sound files. Real-time control of the processing algorithms is done with graphical objects such as faders, displays, control panels, fully configurable by the user. A set of high performance real-time transformations, designed for a musical work on sound morphologies, are also introduced.

1-Introduction

DSP Station was originally designed at the GRM as an internal development tool for real-time sound processing algorithms running on the Motorola DSP6001™. The Apple Macintosh™ and the HyperCard™ 2 software were chosen as a good environment for quick prototyping of easy-to-use graphical interfaces; the Digidesign Sound Tools™ hardware seemed interesting as a low-cost system providing a complete set of interfaces to sound sources of professional quality. HyperDSP™, developed by Adrian Freed, was a first software of this kind, offering interesting features, but it could not easily be adapted to meet our specifications.

The DSP Station software is based on 3 components:
- a real-time kernel running on the DSP6001 and a complete library of optimized DSP macros for standard sound input/output, host communication and basic signal processing,
- a graphical interface based on HyperCard, for the load of DSP algorithms, access to various sound sources and real-time control and display of DSP parameters,
- a set of applications, in the form of general purpose and more specific music oriented high performance signal processing algorithms, associated with specific graphical control interfaces.

The demonstration essentially focuses on the 2 last components.

ICMC 589
2. Functional description

DSP Station is a software running on the Apple Macintosh™ (MAC IIcx,ci, or fi), associated with the DIGIDESIGN Sound Tools™ hardware. The only specific hardware requirement is a 32K word memory extension in the X and Y banks of the DSP. Implemented as a Hypercard™ 2 stack, it works like an application with documents in the form of other Hypercard stacks open in other windows.

The sound sources available in input and output for processing are those available in the Sound Tools hardware, i.e. stereo analog signals (ADIN and internal DACs of the Sound Accelerator), AES/EBU and S/PDIF digital signals (DAT I/O), and mono and stereo sound files (SD2I and MIDN sound files formats). The input file can be read between 2 points once or within a loop. A speed variation playing algorithm and a control panel are available for the input file. The sound output of the system can be recorded in real-time in mono or stereo sound files. The recording is non-modal, i.e. DSP parameters can be varied while the recording is active. Simultaneous playback and recording of two sound files is available in real-time (mono files). The disk management routines and the associated DSP I/O algorithms can also operate in a non real-time, file t. file, sample or block oriented, processing mode.

The software includes a control panel for the remote control of a RDAT player/recorder, and the display of the current SMPTE or A-time time-code. The user can create time-code markers for automated loops and jumps to particular instants. For the moment, this control is only available for the POSTEX D20; future developments could provide a compatibility with other RDAT machines.

A modular DSP code architecture is provided, containing the basic initialisation and interface functions for real time stereo to stereo or file to file block oriented processing. This structure is loaded at the start of the application so that the DSP programmer does not need to include it in his code and can concentrate on the processing algorithms.

The DSP code loader works with .lod ASCII files obtained in output of the standard MOTOROLA development tools. For more permanent DSP codes, this information can also be stored inside of the document in fields or in resources. The user can also load and save contiguous areas of DSP memory named “tables” in specific binary files. The DSP code loader also includes a feature that the automates the load of such coefficient table files.

The software also includes a DSP debugger that performs reads and writes of DSP memories and special bits, launches host commands, starts and stops the DSP, etc...

The real time controls available are 1 and 2 dimensional graphical objects named “raders” for the control and “displays” for the display of DSP parameters. The software includes an interface builder that creates these objects and associates them with a list of attributes describing their action: space and address of the DSP parameters being controlled, variation extremaities and law, unit for numerical values. These objects are constructed from Hypercard standard objects and can be very easily created, moved, copied, duplicated, deleted and restored. Their action is described in their script and can completely be configured by the user during the execution of the program.
DSP variables can be declared in the DSP source code as simple variables or as faders or displays, associated with a list of attributes. This information, present in the DSP code .jad files, is loaded with the code so that the assignment of faders and displays to DSP variables is possible through the only knowledge of the symbolic names defined in the source code.

3 - Signal processing applications

A number of optimized real-time sound processing algorithms, associated with specific graphical interfaces, have been implemented in this environment. They can be divided into three categories:

- General purpose signal processing algorithms: sharp lowpass, highpass and bandpass elliptic filters, a linear phase 8 band equaliser using multirate sampling techniques, phase locked window harmonisation and time stretching, of better quality than most existing commercial products.

- Music oriented sound transformations: additive synthesis, space simulation and Doppler effect, time "freezing" (loops of short duration), ring modulation, resonant filters, sound "shaking" (in French: brassage), random playing of small scraps of sounds.

- Restoration of old recordings: noise subtraction based on transformations in the frequency domain (up to 1024 points in real time).

4 - Conclusion

DSP Station provides a complete environment for the development of real-time and deferred sound processing algorithms for the DSP 56003 and quick design of associated graphical control interfaces. It has applications in research, development, education and all the fields where a high sound quality and a modularity of the graphical interface are required.

The existing signal processing applications provide the composers a set of original and high quality, ready- and easy-to-use sound transformations running on a low-cost hardware.

DSP Station is currently in development: the next developments may include the control of DSP parameters with MIDI, the implementation of a multiprocessor DSP interface, the display of the signal samples or a sonogram of the input file, and the implementation of new algorithms for musical and restoration applications. It may also be adapted for new DSP boards.

Acknowledgments

The development of DSP Station was possible thanks to the GERDSM, a research and development laboratory of the French Navy that uses DSP Station for the validation of signal processing algorithms designed for underwater detection and simulation. We particularly thank Jean-Yves Bernier for his cooperation.