



HAL
open science

New DSP applications on FTS

François Déchelle, Maurizio de Cecco, Enzo Maggi, Norbert Schnell

► **To cite this version:**

François Déchelle, Maurizio de Cecco, Enzo Maggi, Norbert Schnell. New DSP applications on FTS. ICMC: International Computer Music Conference, 1996, Hong Kong, Hong Kong SAR China. pp.1-1. hal-01157141

HAL Id: hal-01157141

<https://hal.science/hal-01157141>

Submitted on 27 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

New DSP applications on FTS

François Déchelle, Maurizio De Cecco, Enzo Maggi, Norbert Schnell

Proceedings of the International Computer Music Conference, Hong Kong (1996)

Copyright © Ircam - Centre Georges-Pompidou 1996

Abstract

This paper describes new real-time applications on FTS. A brief recall of FTS features is given, together with informations on supported platforms. The FTS object modules organization is described. A view of the development environment is given. The FTS vector library is described. The future extensions of the DSP objects programming interface are presented, allowing more flexibility in scheduling. The latest developments in DSP are described.

1. Introduction

FTS is the Ircam programming and runtime environment for signal processing oriented real-time applications. Its architecture has been presented in details at ICMC'95 [[Dechelle and De Cecco,1995](#)]. FTS provides a real-time multi-threaded executive and a message-based object system.

The object system provides a specific programming interface for DSP objects. The execution of DSP operations is supported by the FTL vector computation engine [[Dechelle and De Cecco,1995](#)].

2. Current architectures

2.1 Currently supported platforms

Current platforms running FTS are :

- Ircam Signal Processing Workstation
- Silicon Graphics workstations, with native audio system
- PCI and Power-PC Macintosh and Daystar multi-processors Macintosh, with native audio or add-on audio boards.

2.2 Benchmarks

The following table gives some benchmarks comparing the performances of the different supported platforms. The performance is measured as the maximum sampling rate that can be achieved for a typical patch. All the benchmarks are for one processor. For the ISPW platform, the DSP functions are coded in assembly. For the other platforms, the whole code is C.

| Platform & Hardware | kHz |
|------------------------------|------------------|
| SGI Indy (R4400, 200 MHz) | 310 |
| ISPW (i860, 40 MHz) | 180 |
| Mac 9500 (Power-PC, 132 MHz) | 330 ¹ |

¹: estimated

3. FTS object libraries

In order to better organize the objects set, the objects are now partitioned as FTS modules. The FTS module abstraction is a data structure that is part of the FTS kernel [[Dechelle and De Cecco,1995](#)] and provides mainly a simple way to install the objects classes.

The object code can be linked to the FTS executable statically (i.e. before execution) or dynamically (i.e. during execution) using a dynamic linker such as `a1` on a Silicon Graphics platform. The fact that an object is statically or dynamically loaded requires no changes in the code, thus allowing the user to choose at initialization time which sets of objects will be used and to configure FTS according to its needs.

The module organization is the support of both static and dynamic loading of objects. Through a simple *provide/require* mechanism, the fact that an object class is already installed can be easily tested, and if not installed, this can also trig the dynamic loading of a class on a platform that supports it.

4. FTS as a DSP programming environment

Several clients can be connected to the FTS server simultaneously, using the `ftsd` network daemon [[Dechelle et al.,1995](#)]. This can provides a flexible development environment to the FTS object programmer : together with the FTS editor and control interface [[Maggi,1996](#)], DSP-specific graphical tools can be use to view and edit signals in various representations.

5. The FTS vector library

To achieve complete portability of FTS DSP objects, a library for vector arithmetic and signal processing has been developed for FTS. This library provides basic vector/vector and vector/scalar arithmetic and logical functions and basic signal processing operations , mainly FFT.

The API of the FTS vector library is platform independent. The implementation of each function is platform dependent and can use a platform specific optimized library. A portable C version is provided for each function.

To achieve better performance on pipelined processors, the library provides 2 versions of each function, with and without loop unrolling. The ``unrolled" version can be used only if vector size is a multiple of the unrolling factor. For each function, a FTL-callable [[Dechelle and De Cecco,1995](#)] version is also provided.

6. New DSP programming model

An extension of the DSP scheduling model of FTS is under development, to ease implementation of advanced DSP algorithms, for instance involving FFT. A unified framework will be introduced soon, handling both multi-rate processing, conditional execution and post-optimization. A more flexible and extendible typing system will be added to the DSP computation engine.

7. New DSP objects

7.1 Additive synthesis and processing

The latest synthesis and analysis algorithms developed at Ircam, like the FFT-1 additive synthesis [[Rodet et al.,1993](#)], have been implemented as a set of FTS objects.

Robust pitch tracking and score following, including Miller Puckette's work [[Puckette,1995](#)] on voice score following for Philippe Manoury's *En Echo*, are now available.

7.2 Physical modeling

Work is in progress to implement under FTS Modalys [Eckel *et al.*,1995], a physical modeling synthesis software. For such synthesis algorithms, a sample by sample computation mode will be introduced.

8. Summary

We describe in this paper the current state and the evolutions of FTS, making it a flexible, modular and evolutive environment for DSP programming and real-time control. We also introduced the latest developed DSP applications.

References

[Lindemann *et al.*,1991]

Eric Lindemann, François Déchelle, Michel Starkier, Bennett Smith. *The Architecture of the IRCAM Musical Workstation*. Computer Music Journal, 15(3): pp. 41-50.

[Déchelle *et al.*,1994]

François Déchelle, Maurizio De Cecco, Miller Puckette, David Zicarelli. *The Ircam "Real-Time Platform" : evolution and perspectives*. ICMC 94, Aarhus.

[Déchelle and De Cecco,1995]

François Déchelle, Maurizio De Cecco. *The Ircam Real-Time Platform And Applications*. ICMC 95, Banff.

[Déchelle *et al.*,1995]

François Déchelle, Maurizio De Cecco, Norbert Schnell. *FTS On-line Documentation* : <http://www.ircam.fr/equipes/temps-reel/fts/doc>

[Eckel *et al.*,1995]

Gerhard Eckel, Francisco Iovino, René Caussé. *Sound synthesis by physical modeling with Modalys*. Proceedings of the International Symposium of Music Acoustics 1995, Le Normant, Dourdan 1995.

[Maggi,1996]

Enzo Maggi, François Déchelle. *The Evolutions of the Graphic Editing Environment for the IRCAM Musical Workstation* ICMC 96, Hong-Kong.

[Puckette,1991]

Miller Puckette. *FTS : A Real-Time Monitor for Multiprocessor Music Synthesis*. Computer Music Journal, 15(3): pp. 58-68.

[Puckette,1995]

Miller Puckette. *Score Following Using the Sung Voice*. ICMC 95, Banff.

[Rodet *et al.*,1993]

Adrian Freed, Xavier Rodet, Philippe Depalle. *Synthesis and control of Hundreds of Sinusoidal Partial on a Desktop Computer without Custom Hardware*. ICMC 93, Tokyo.