Towards a Modular System for Real-Time Multiple-Input/Multiple-Output Audio Processing

Karim Helwani*, Till Rettberg*, Herbert Buchner◇, and Sascha Spors*

*Quality and Usability Lab, Telekom Innovation Laboratories, ◇Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany

{karim.helwani@telekom.de, till.rettberg@campus.tu-berlin.de}

Abstract

In this paper, we present a modular multi-threaded software framework for real-time adaptive MIMO signal processing. It is planned to release it as an open source project for other developers to collaborate on.

Introduction

Multichannel audio acquisition and reproduction techniques offer spatial selectivity and diversity as additional degrees of freedom over single-channel systems. In terms of enhancing sound realism in telecommunication systems there is a growing interest in array-based audio signal processing. A major challenge to fully exploit the potential of array processing in practical applications lies in the development of Multiple-Input/Multiple-Output (MIMO) systems. Adaptive filters are a key component in this context. In a typical full-duplex communication system many algorithms can be integrated as depicted in Fig. 1. The system $\mathbf{G}$ can be described using the scattering parameter representation as suggested in [1, 2]

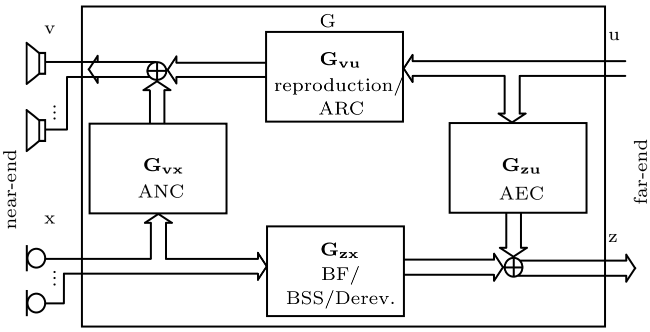

Figure 1: Example configuration of MIMO blocks in a full-duplex communication system

$$\begin{pmatrix}\mathbf{v}\\ \mathbf{z}\end{pmatrix} = \begin{pmatrix}\mathbf{G_{vu}} & \mathbf{G_{vx}}\\ \mathbf{G_{zu}} & \mathbf{G_{zx}}\end{pmatrix} * \begin{pmatrix}\mathbf{u}\\ \mathbf{x}\end{pmatrix},$$

where the submatrices $\mathbf{G}_{..}$ contain the impulse responses of the MIMO-subsytems and $*$ denotes convolution. Ideally, the subsystem $\mathbf{G_{vu}}$ (active room compensation unit) inverts the influence of the room impulse responses if the signal processing for sound reproduction should be independent of the far-end signal, $\mathbf{u}$. Local noise is canceled at the ears of the users if a noise compensation signal is fed to the loudspeakers by $\mathbf{G_{vx}}$ such that the resulting signal at the reference position annihilate the noise signal. Moreover, $\mathbf{G_{zx}}$ could be a special realization of some desired multichannel signal enhancement, e.g., beamforming, blind source separation, or dereverberation.

In order to compensate the feedback between the loudspeaker signals $\mathbf{v}$ and the microphone signals $\mathbf{x}$, a replica of the corresponding echo signals has to be estimated using $\mathbf{G_{zu}}$ and subtracted from the captured signal.

The challenges that a developer faces while implementing such a system are the typically high computational complexity and ill-posedness and of the underlying adaptive filtering problem. It has been shown that treating adaptive filtering problems for such broadband multichannel systems in spatio-temporal transform domains is a powerful approach [3], that aims at decoupling the MIMO system both in the temporal and in the spatial dimensions and enhancing the conditioning of the problem.

The associated computational complexity however remains high, especially for massive multichannel applications. Real-time constraints imposed by telecommunication application demands efficient utilization of processing resources. The availability of multicore and multiprocessor architectures call for a multi-threaded signal processing framework. The graphical patching paradigm of *Pd* [1] or the commercial *Max/MSP* [2] can be considered ill-suited for massive-multichannel systems. Furthermore, a slim stand-alone application may be desirable. An early full-duplex framework was presented in [2]. A recent development is presented in [4], although more targeted towards playback-oriented renderer model. For adaptive (transform-domain) MIMO processing, the available solutions are not optimal.

Design Considerations

We aim at easing the deployment of MIMO algorithms on a symmetric multiprocessor architectures, shortening the way from a prototype in numerical simulation software to real-time-capable environment. Noticing that a MIMO system is intuitively represented by a block-based structure as depicted in Fig. 1, a similar high-level modularity is desired. Large number of channels should be easily manageable for the application programmer. Real-time efficiency is a must, parallel processing should be as transparent as possible.

[1] `http://puredata.info/`

[2] `http://cycling74.com/`

Engine Architecture

The framework is implemented in C++, currently for UNIX-like operation systems, aiming at an open-source release. An application within the framework is built from independent MIMO signal processing `Blocks`, forming nodes in a processing graph. An associated `Engine` object supervises data-flow, thread coordination and communication with the driving back-end. Several abstract `Block`-prototypes with different virtual processing functions provide a high-level abstraction of thread synchronization.

The signal flow is organized by `Portgroups`, which intend to bundle multiple channels of the same processing semantics, allowing a convenient construction of massive multichannel systems. The signal paths are typed with respect to frequency- or time-domain data, assuming the latter will be real-valued in most audio applications. A callback- and block-based back-end is assumed as a driver for the `Engine`, the current alpha version supports the JACK audio framework [5]. An abstract `Controller` object handles asynchronous I/O in a separate thread to keep the real-time processing threads undisturbed.

The implementation relies heavily on object-oriented design patterns [6], to allow extensibility within familiar programming idioms. Where real-time constraints prohibit run-time polymorphism, C++ templates are employed.

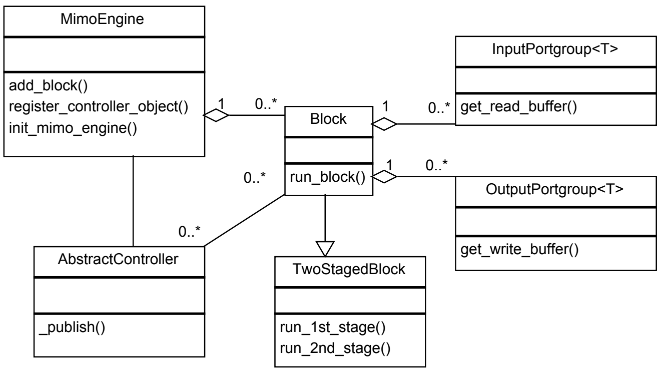

Figure 2: Simplified excerpt of the class diagram, depicting the core classes from an application programmer's view. The inherited *TwoStageBlock* is an extended *Block* prototype providing two synchronized processing stages for inputs and outputs.

Parallelization Strategy

The individual signal paths (or single `Ports` of the `Portgroups`) at a particular processing `Block` form a job set. The `Engine` object manages a fixed size thread pool, transparently dispatching available jobs. Thread intercommunication within the individual processing blocks is defined by abstract block prototypes, offering different synchronized processing stages for the `Blocks`' inputs and outputs. The default case handles the outputs in parallel, corresponding to the decomposition of a MIMO system into multiple MISO (multiple input/single output) systems. User definition of new `Block` prototypes for arbitrarily sophisticated thread cooperation is possible.

Application Scenarios

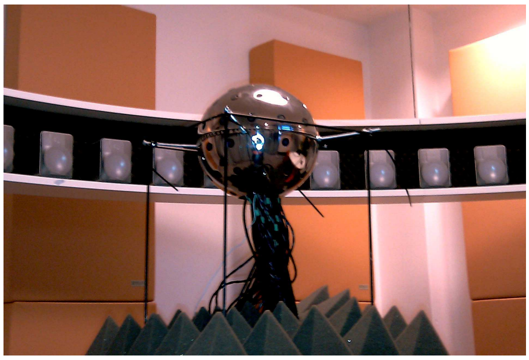

Figure 3: 56-channel loudspeaker array and 64-channel microphone array for full-duplex communication interface.

Our real-time setup of the full-duplex multichannel communication system is depicted in Fig. 3. It consists of up to 56 wide-band loudspeakers plus sub-woofer, and a 64-channel rigid-sphere microphone array. Two 8-cores processor PC platforms, one for the reproduction side and one for the recording side, provide the necessary processing power. By now the framework's alpha version has been employed for real-time beamforming with the microphone array, as presented in [7].

Conclusion

This paper outlined the design and realization of a modular, real-time, full duplex multiple-input/multiple-output signal processing framework, consisting of a multi-threaded engine. Architecture and implementation of the framework are discussed, with respect to signal processing and software design considerations.

References

[1] H. Buchner, W. Herbordt, and W. Kellermann, An Efficient Combination of Multi-Channel Acoustic Echo Cancellation With a Beamforming Microphone Array, Conf. Rec. 1st IEEE Intl. Workshop on Hands-Free Speech Communication (HSC), Kyoto, Japan, pp. 55-58, April 2001

[2] W. Herbordt, H. Buchner, W. Kellermann, R. Rabenstein, S. Spors, and H. Teutsch. Full-Duplex Multichannel Communication: Real-Time Implementations in a General Framework. Proc. IEEE Int. Conf. on Multimedia and Expo (ICME), Baltimore, MD, USA, pp. 49-52, July 2003

[3] H. Buchner, and S. Spors. A General Derivation of Wave-Domain Adaptive Filtering and Application to Acoustic Echo Cancellation. Proc. Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, Oct. 2008

[4] M. Geier, T. Hohn, and S. Spors. An Open Source C++ Framework for Multithreaded Realtime Multichannel Audio Applications. Linux Audio Conference, May 2012

[5] Jack Audio, URL: `http://jackaudio.org/`

[6] E. Gamma, R. Helm, R. Johnson and J. Vlissides, Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994

[7] T. Rettberg, K. Helwani, S. Spors and H. Buchner, Practical Aspects of the Calibration of Spherical Microphone Arrays, German Annual Conference on Acoustics (DAGA), March 2012