ADEPT: A Framework for Adaptive Digital Audio Effects

Owen Campbell\(^1\), Curtis Roads\(^1\), Andrés Cabrera\(^1\), Matthew Wright\(^3\), and Yon Visell\(^2\)

\(^1\)Media Arts and Technology, University of California Santa Barbara
\(^2\)Media Arts and Technology and Electrical and Computer Engineering, University of California Santa Barbara
\(^3\)Center for Computer Research in Music and Acoustics, Stanford University

Correspondence should be addressed to Owen Campbell (owencampbell@mat.ucsb.edu)

ABSTRACT

The field of music information retrieval (MIR) has enabled research in ‘intelligent’ audio processing. Emerging applications of MIR techniques in music production might provide mixing engineers, musicians and composers control over extant effects processing plugins based on the audio content of the recorded music. In adaptive digital effects (ADAFx) processing, mapping functions are used to modulate algorithm parameters via features that are extracted from the input audio signal or signals. We present an audio software plugin that is designed to facilitate feature-parameter mappings within digital audio workstations. We refer to it as the Adaptive Digital Effects Processing Tool (ADEPT).

1 Introduction

The field of music information retrieval (MIR) has led to the development of numerous signal processing algorithms for extracting semantic information from audio signals. By repurposing some of these audio feature extraction algorithms, emerging applications of MIR techniques in music production might provide mixing engineers, musicians and composers control over extant effects processing and synthesis plugins driven by the content of musical performances.

In adaptive effects processing, mapping functions are used to modulate algorithm parameters via features that are extracted from the input audio signal or signals such that the effects operate in a signal-dependent way. There are many examples of adaptive effects, such as dynamic range processors, gates, and pitch-correctors that have long been mainstays of music production. However, in principle any effect could become an adaptive effect if one or more of its processing parameter settings could be dynamically driven by signal content.

For example, a vocalist could create a control mapping that increases the amount of reverb as he or she sings higher and higher pitches, or the dissonance of electric guitar music could be mapped to the intensity or timbre of a distortion effect. These examples are just a small subset of the possible mappings and novel adaptive effects made possible by a flexible adaptive effects framework. This document describes a proposed system for flexible ADAFx routing using existing music production software tools.

2 Methods

As in [2] and [3], this work proposes an approach to adaptive effects processing in which existing audio software tools can be augmented to facilitate arbitrary
control mappings between the content of input audio signals and the behavior of audio processing and synthesis algorithms. Modern digital audio workstations (DAWs) possess all the capabilities necessary to enable the control mappings required for a generalized framework for adaptive effects processing. An abundance of diverse, high-quality effects processing and synthesis algorithms in the form of audio plugins and support for inter-application communication for remote DAW control are two critical components of the system. DAWs lack only the feature extraction pipeline and mapping capabilities required to complete the system.

The ADEPT adaptive audio processing software implements feature extraction and a mapping algorithm in order to facilitate ADAFx processing (Fig. 1). As a result it can control any processing parameters that are made available as automatable controls within the DAW. Developed using JUCE, a C++ framework which enables cross-platform deployment of audio plugins, it communicates with Ableton Live using Open Sound Control [4] and the LiveOSC MIDI Remote Script. ADEPT works with the DAW and existing plugins, providing immediate benefits due to the large variety and quality of available effects processing and synthesis algorithms, and integration with production and performance practices.

Each instance of the ADEPT plugin computes a stream of audio features using Essentia [5], a C++ audio feature extraction library. Audio features from a given source may be mapped to any plugin parameter in the DAW session, including those belonging to plugins residing on other tracks. At present, the feature-parameter mappings are one-to-many; a single feature stream may feed multiple mappings controlling different parameters. Initial experimentation with the system suggests that there is great potential in this capability, and there are far more possible mappings to investigate than can be explicitly described in this document. For any given effect or synthesis algorithm there are a number of parameters which, when modulated in real time, produce output that sounds markedly different from the same algorithm with static settings. Some of these potential mappings will produce useful or interesting results and others will not. Much of this work has been devoted to discovering the former.

3 Results

To illustrate the capabilities of the ADEPT software, we describe two case examples of adaptive effects using the software. Examples were tested using segments of high-quality multi-track recordings from MedleyDB [6].

Case example 1: One of the simplest but most compelling examples is the mapping from vocal pitch to the decay time of a reverb unit (Fig. 2, Fig. 3a). As the detected pitch [7] increases, so does the reverb’s decay time, resulting in a very dry application of the effect for the lowest notes of the singer’s range and a very wet, sustained reverb for the highest notes. This relationship, though applied in a rather extreme manner in this example, is useful for automatically avoiding muddy low notes with too much reverb while applying a desirable amount of reverb to higher notes.

Fig. 1: Each instance of the plugin computes audio features a stream of audio features which can feed mapping functions for controlling audio plugin behavior.
Fig. 2: Example time series of audio feature and effect parameter data with mapping from vocal pitch to reverb decay time.
Fig. 3: Users can activate mappings between selected audio features and processing parameters (background) and access simple controls and visualization of mapping function behavior (foreground) via the plugin’s GUI.
Case example 2: In another example, the dynamic application of a guitar distortion effect is based on three different audio feature mappings. The primary adaptive processing relationship maps higher detected levels of dissonance [8] to lower settings for the distortion unit’s ‘drive’ parameter (Fig. 3b), thus applying heavy distortion to power chords and single notes and moderate distortion to more complex chords. In the next mapping, the ‘tone’ parameter tracks the detected brightness value (as approximated by spectral centroid [9]). Finally, the detected loudness [10] modulates the ‘dry/wet’ parameter to automatically make the distortion wetter at low levels and dryer at high levels to manage the dynamic range of the effect’s output.

4 Discussion

The adaptive reverb effect in case example 1 above demonstrates that the introduction of even a single adaptive control mapping to a familiar effect can have dramatic results. By combining multiple such mappings, it is easy to turn static effects into highly dynamic ones, as in the adaptive distortion described in case example 2 above. There are many other potential applications, among them several that have been implemented using the ADEPT system, including a cross-adaptive automated panning scheme, an onset-ducking delay effect, and a peak-tracking resonant filter.

With more development time the ADEPT plugin will support logical comparison or arithmetic combination of multiple feature streams for implementing many-to-many mappings, enabling the full taxonomy of adaptive effects outlined in [1]. Future work will also explore the use of additional audio features, more complex mapping paradigms, and extensions to the user interface.

5 Summary

We developed a conceptual framework for flexible adaptive effects routing in the DAW environment and investigated potential use cases for such a system. A software implementation of this framework was created in the form of an audio plugin, where we explored the utility of a number of audio features and an interface for defining control mappings.

References


