

the
Getting Max out of Max with
^
gen~

ADC 2021

Isabel Kaspriskie



Make sure you're ready

1. Max 8 installed and authorized (must be full Max license, not a trial or Max for Live)
2. Xcode or Visual Studio installed
3. CMake \geq 3.19 installed



Workshop goals

1. Motivation for using gen~
2. Get comfortable with gen~ patching
3. Practice translating block diagrams to gen~ patchers
4. Explore gen~ C++ export API
5. Embed exported code into a JUCE audio effect plugin

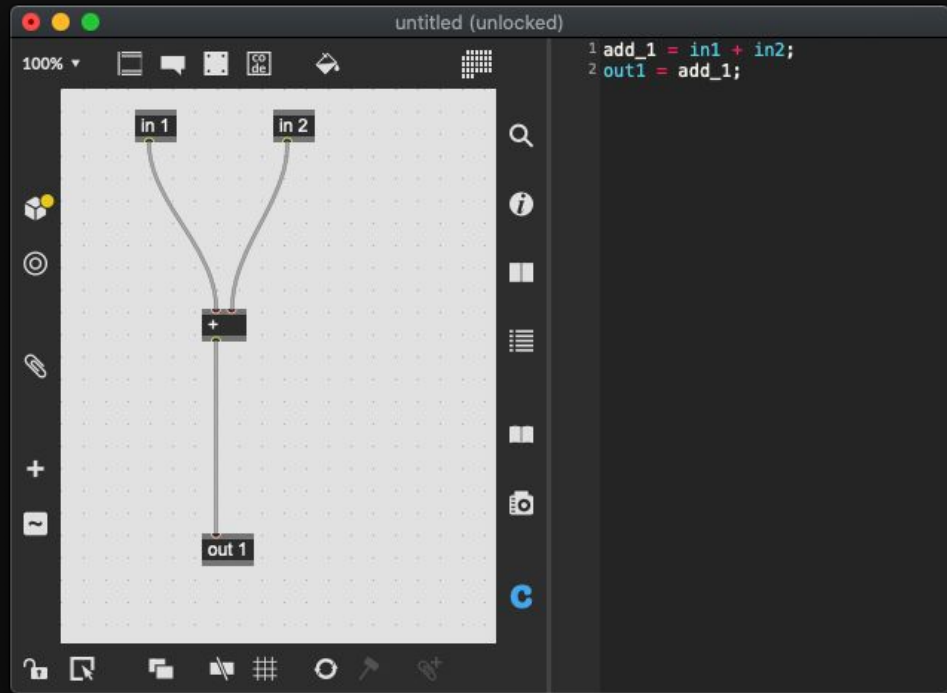


First, a poll.



What is gen~?

- Single-sample audio processing
- Patching environment within Max
- Code generation



https://docs.cycling74.com/max8/vignettes/gen_overview



Where is gen~ useful?

Max programmers

- Flexible feedback loops
- Single-sample processing
- Export and share with C++ programmers

C++ programmers

- Real-time prototyping
- Easier* to translate block diagrams to a visual paradigm
- Codebox: procedural *genexpr* language



Getting comfortable

[github.com/Cycling74/
adc2021-workshop](https://github.com/Cycling74/adc2021-workshop)

- Max interface
- The **gen~** patcher
- Finding help



Practicing with block diagrams

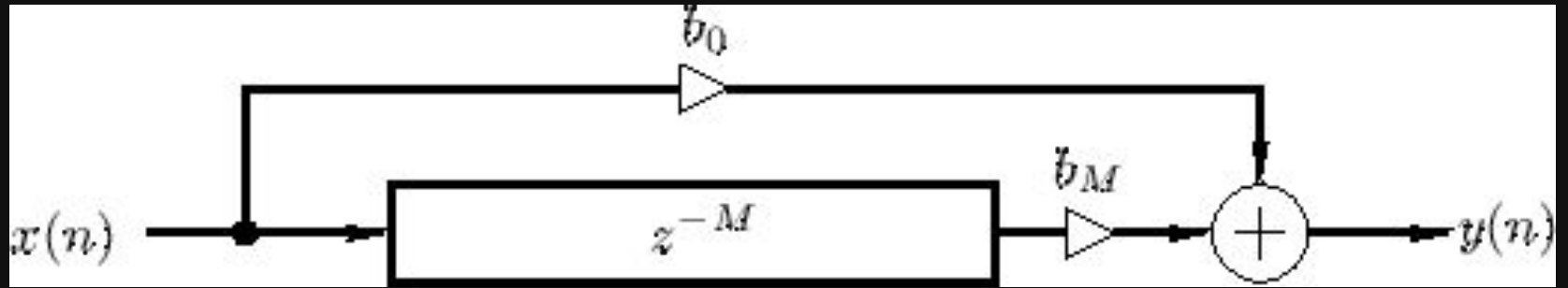
- Comb filters
- Biquad filter
- Flanger
- Reverb



Level 1: Comb filters

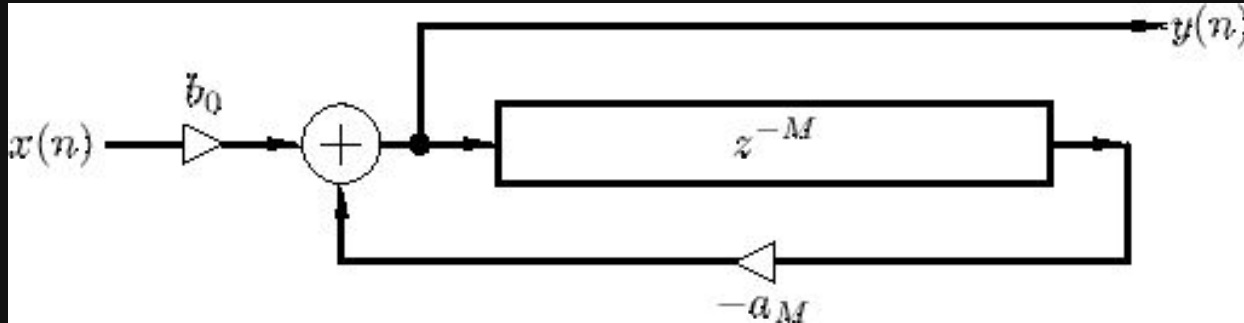


gen.comb-ff~



Feedforward comb filter

gen.comb-fb~

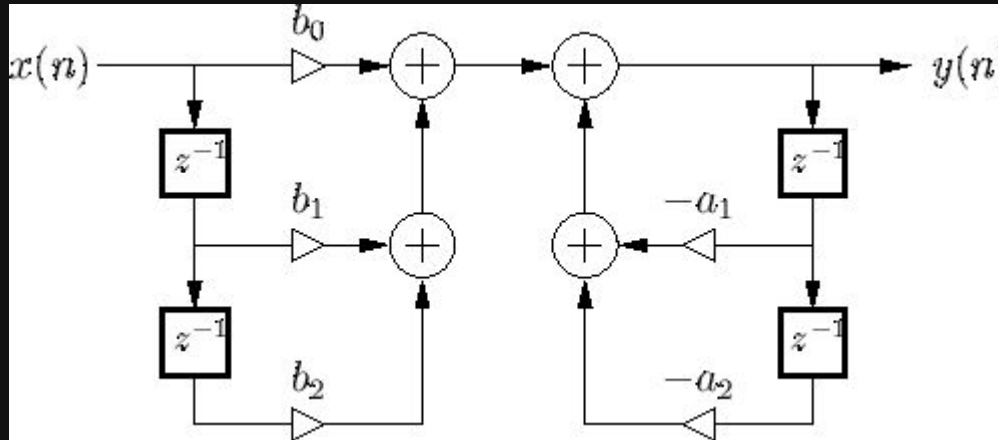


Feedback comb filter

Level 2: Biquad filter

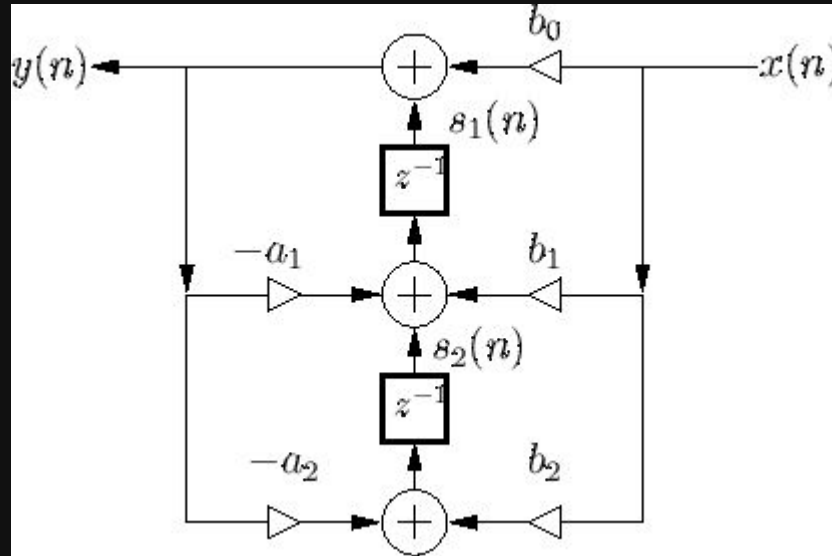


gen.biquad~



Direct Form I

gen.biquad~

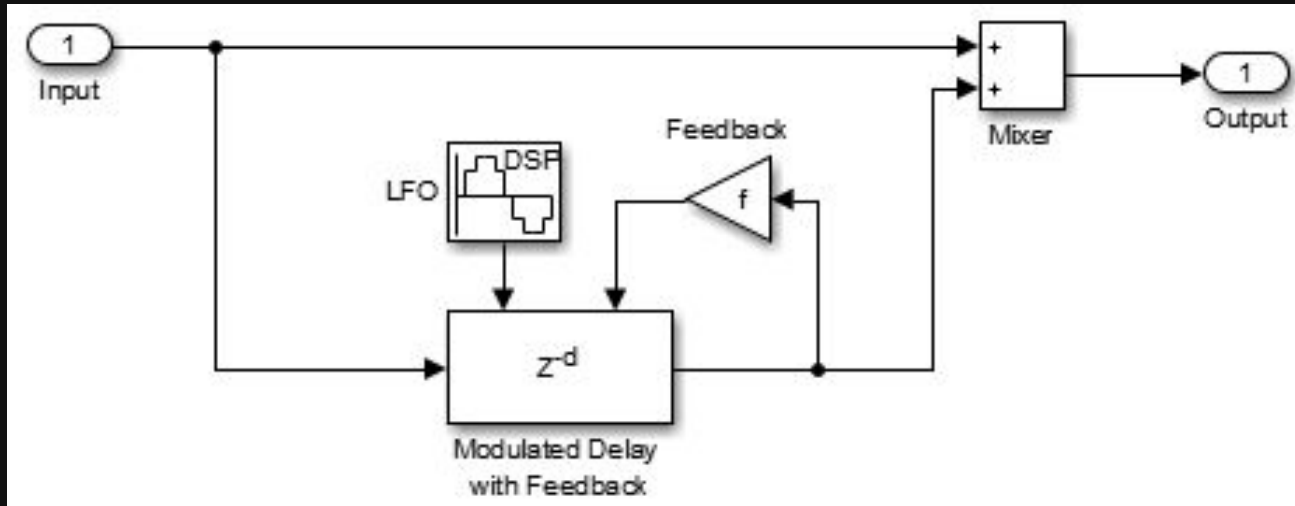


Transposed Direct Form II

Level 3: Flanger



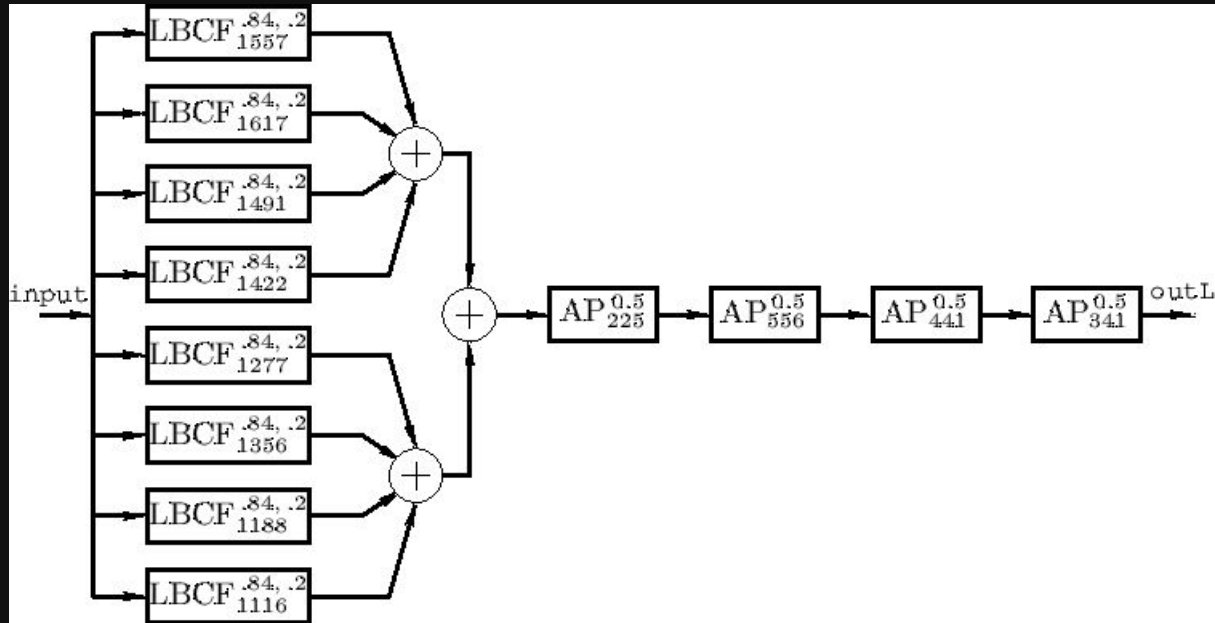
gen.flanger~



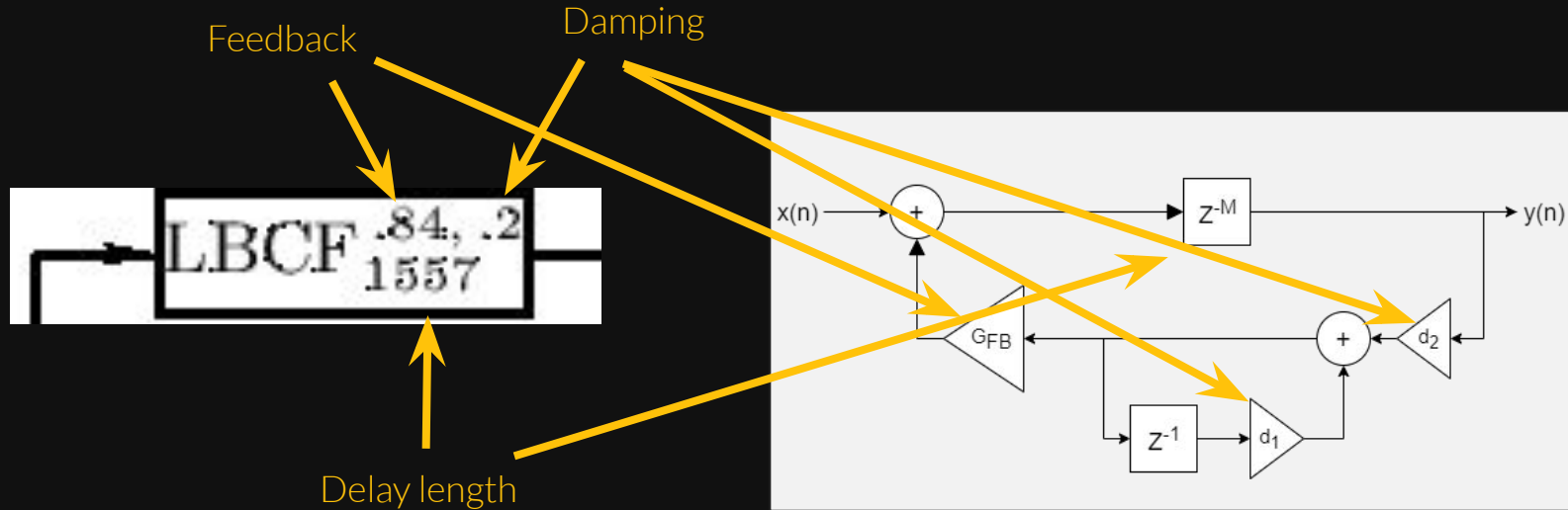
Level 4: Reverb



gen.freeverb~

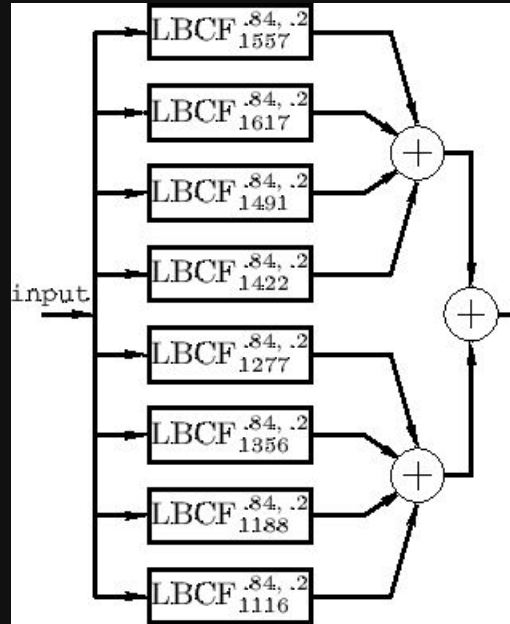


Part 1: LP Feedback Comb Filter

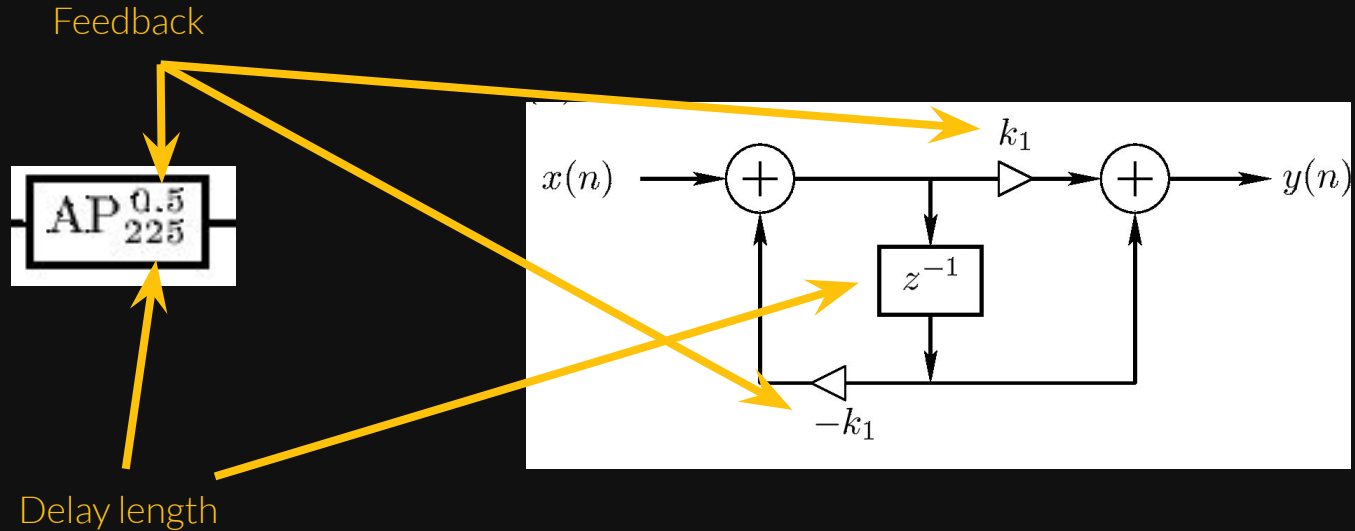


Modified diagram from
<https://www.cankosar.com/digitales-tiefpass-kammfilter/>

Part 2: Parallel LBCFs



Part 3: Allpass filter



Part 4: Making it stereo

*“Processing for the right channel is obtained by adding an integer to each of the twelve delay-line lengths. This integer is called **stereospread**, and its default value is 23.”*

<https://ccrma.stanford.edu/~jos/pasp/Freeverb.html>



gen~ C++ Export

- How to export
- How to include in a project
- A few tips



Export demo



API Overview

- Include headers with `#include "gen_export.h"`
- Include all sources in the compilation step
- Methods you should interact with are in `gen_export.h` and are in the `gen_export:: namespace`

Note: This slide assumes exported code is named **gen_export**. It can be any name you would like — simply replace any references with the name you choose.



gen~ → JUCE VST demo

