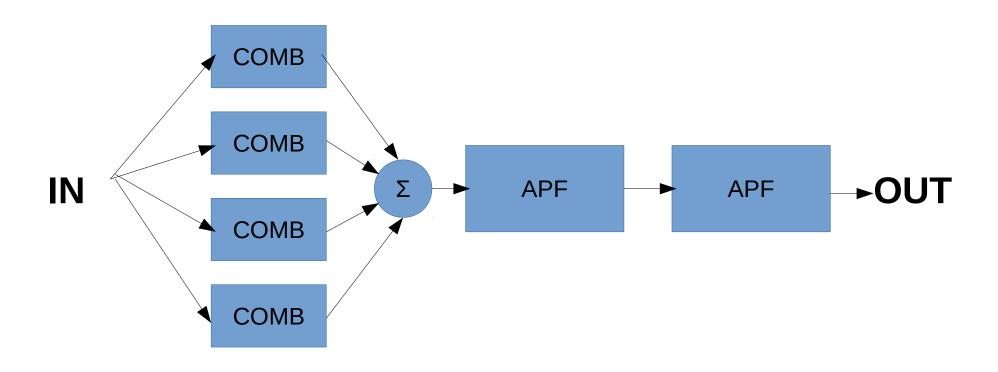
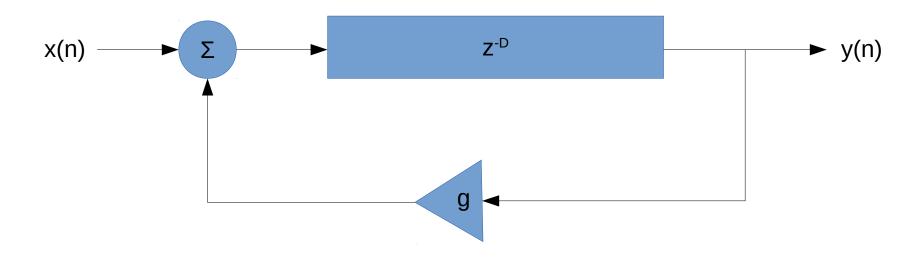
Schroeder's Reverberator

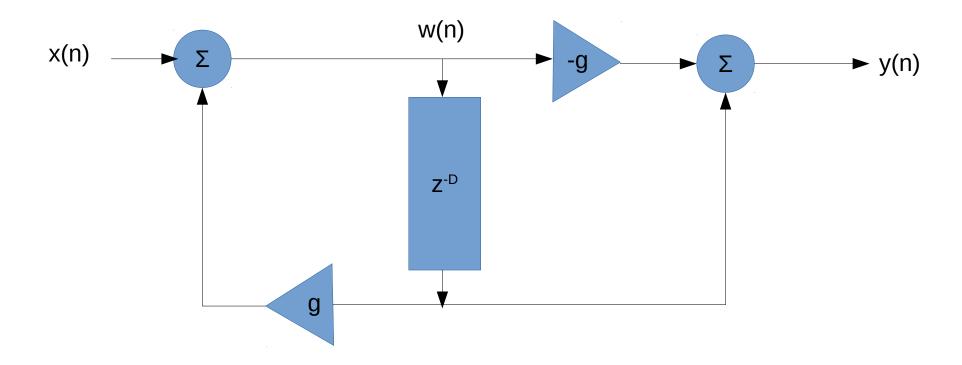


Comb Filter



$$y(n) = x(n-1) + g * y(n-1)$$

All Pass Filter



$$w(n) = x(n) + g * w(n - D)$$

 $y(n) = -g * w(n) + w(n - D)$

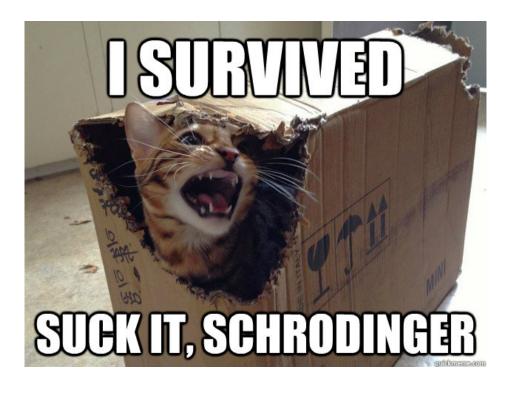
Class Diagram

jackmodule.h main.cpp Code by Marc Groenewegen schrodingersReverb.h filter.h Circular Buffer that process(inBuf, outBuf) keeps track of the Reverb based on played samples. Algorithm of filters Schroeder's Seperate read and write Reverberator with heads allpass filters and comb filters. combFilter.h allPassFilter.h Sample values in type: float See Page 2. See Page 3. Calculation in double Or in int (ask Pieter)

Schrödinger's Reverb

Functionaliteit:

- Raspberry Pi
- Realtime input/output
- Reverb effect based on Schroeder's reverberator.
- Use filter.h base class from synthesizer assignment.



Extra:

- tweaking the parameters
- changing the algorithm
- add potmeters to change the parameters.