

Implementing digital IIR filters in the lattice form

Lab 6, SDP

Objective

The students should become familiar with *lattice*-type realization structure used for implementing IIR filters.

Theoretical notions

The *lattice-ladder* implementation form of an IIR filter of order 3 is below:

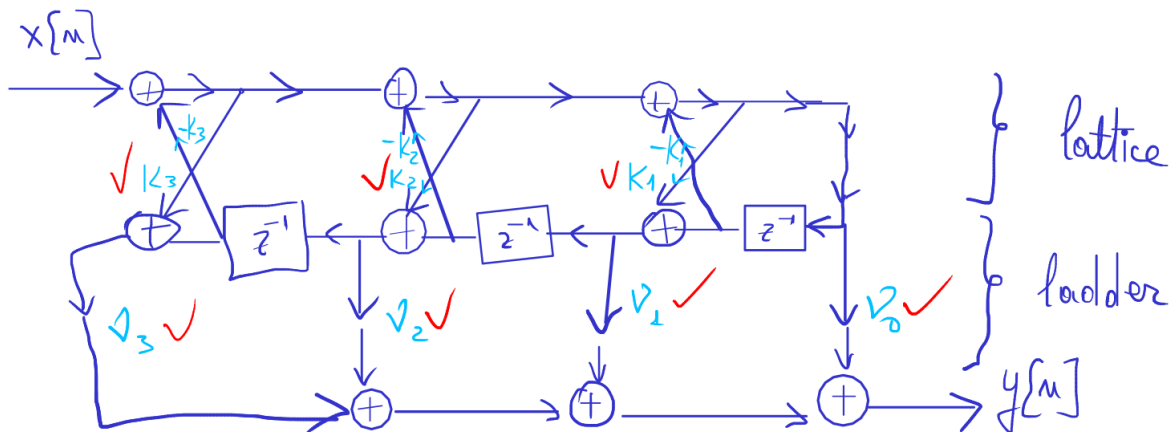


Figure 1: Lattice-ladder form, IIR order 3

Equations:

$$H(z) = \frac{C(z)}{A(z)}$$

The reflection coefficients K_i are found just like for FIR systems (previous lab.):

$$\begin{aligned} A_0(z) &= B_0(z) = 1 \\ A_m(z) &= A_{m-1}(z) + K_m \cdot z^{-1} \cdot B_{m-1}(z) \\ A_{m-1}(z) &= \frac{A_m(z) - K_m \cdot B_m(z)}{1 - K_m^2} \\ B_m(z) &= z^{-m} B_m(z^{-1}) = \text{like } A_m(z), \text{ with coefficients reversed} \end{aligned}$$

Additionally, for the ν_i coefficients we use a similar equation:

$$C_{m-1}(z) = C_m(z) - \nu_m \cdot B_m(z)$$

Exercises

1. Consider the causal IIR system with poles and zeros, with the system function:

$$H(z) = \frac{1 + 2z^{-1} + 3z^{-2} + 2z^{-3}}{1 + \frac{2}{5}z^{-1} + \frac{7}{20}z^{-2} + \frac{1}{2}z^{-3}}$$

Find and draw the equivalent *lattice* structure for the IIR filter.

2. Consider the causal IIR system, with no zeros, with the following system function:

$$H(z) = \frac{1}{1 + \frac{2}{5}z^{-1} + \frac{7}{20}z^{-2} + \frac{1}{2}z^{-3}}$$

Find and draw the equivalent *lattice* structure for the IIR filter.

3. In the Matlab environment, use the `fdatool` tool to design one of the following filters:
 - a. A low-pass IIR filter of order 4, elliptic type, with cutoff frequency of 6kHz at a sampling frequency of 44.1kHz;
 - b. A high-pass IIR filter of order 4, elliptic type, with cutoff frequency of 2.5kHz at a sampling frequency of 44.1kHz;
 - c. A band-pass IIR filter of order 4, elliptic type, with passband between 0.5kHz and 5.5kHz at a sampling frequency of 44.1kHz.
4. Export the coefficients to Matlab and convert them to the *lattice* form using `tf2latc()`
5. In Simulink, implement the above filters in *lattice* form. Apply at the input an audio signal and show and listen to the output signal, as well as the to original. How does the filtered signal sound like, compared to the original?
6. In Simulink, apply a video file to the input (select a video file in the **From Multimedia File** block). Replace the audio output blocks with a **Video Viewer** block. How does the filtered signal look like?

Notes:

- Set the following parameters for the Simulink model, to enable a discrete simulation with fixed (auto) step:
 - Type: *Fixed-step*
 - Solver: *discrete (no continuous states)*

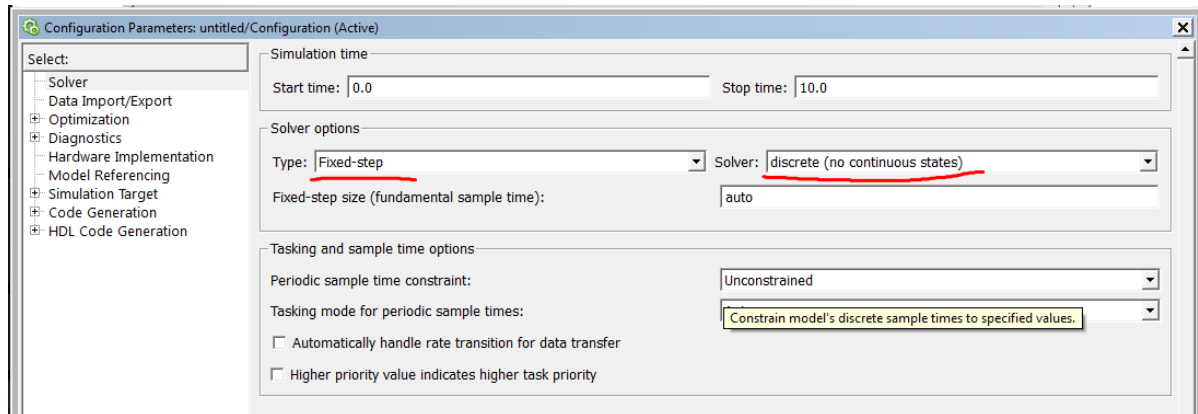
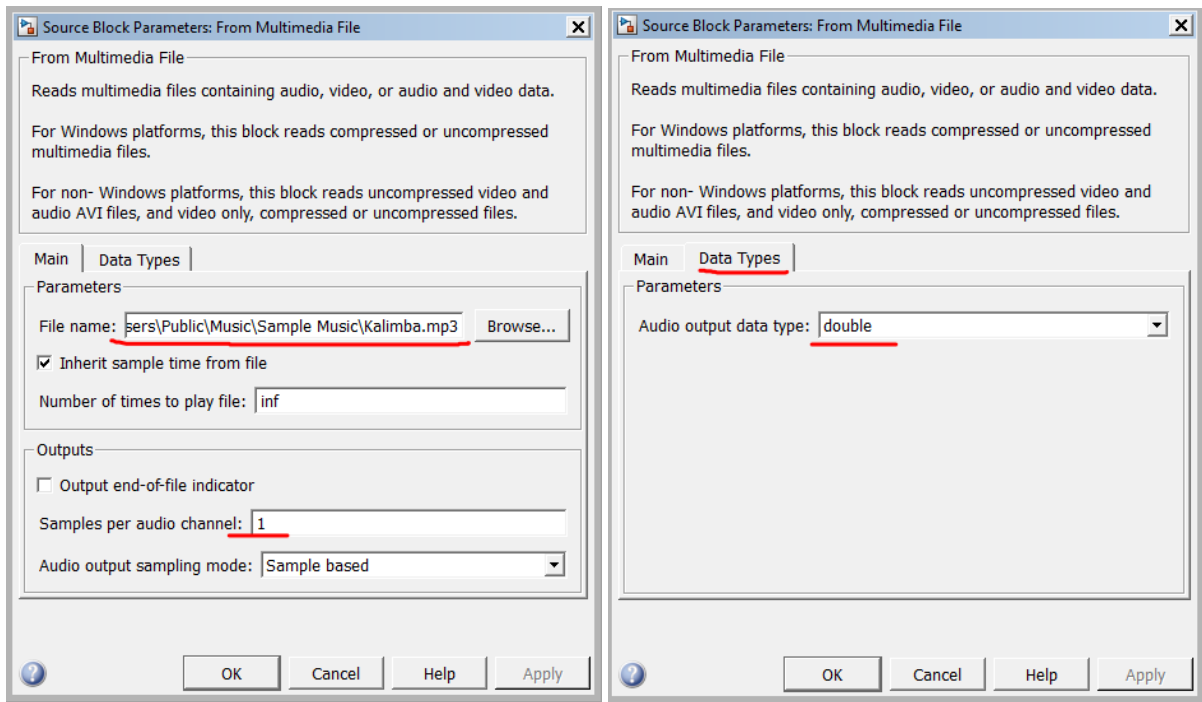


Figure 2: Model settings for discrete models

- You will need the blocks *Unit Delay*, *Sum* and *Gain*
- At the input put a *From Multimedia File* block, and at the output put a *Buffer* followed by *Audio Device Writer* block
- At the output, before the *To Audio Device* block, put a *Manual Switch* block in order to be able to switch easily between the original signal and the filtered one
- For the *From Multimedia File* block, select an audio file (de ex. Kalimba.mp3 from My Documents) and update the following settings
 - choose *Sample-based*
 - *Samples per audio channel* = 1
 - “DataTypes/Audio output data type” = *double*



Final questions

1. TBD