State-space implementations of digital IIR filters

Lab 7, SDP

Objective

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

Theoretical notions

Exercises

1. Consider the IIR system with the system function

$$H(z) = \frac{1 + 2z^{-1} + 3z^{-2} + 2z^{-3}}{1 + 0.9z^{-1} + 0.8z^{-2} + 0.5z^{-3}}$$

   a. Write the equations and draw the type I and type II state-space implementations of this system
   b. Compute the first 5 values of the step response, considering the initial conditions $v[0] = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

2. Consider the system with the following state-space equations:

$$v[n+1] = \begin{bmatrix} 0 & 1 \\ -0.81 & 1 \end{bmatrix} v[n] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} x[n]$$

$$y[n] = \begin{bmatrix} -1.81 & 1 \end{bmatrix} + x[n]$$

a. Find the system function of this system
b. Compute the first 5 values of the step response, considering the initial conditions $v[0] = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
c. Draw the the type I and type II state-space implementations of this system, as well as the direct form II implementation

3. In the Matlab environment, use the `fdatool` tool to design a stopband filter of order 4, elliptic type, with stop band between 1kHz and 3kHz, at a sampling frequency of 44.1kHz. Export the coefficients in the Matlab workspace as the vectors `a` and `b`.
4. In the Matlab environment, implement a function `filter_spst(b, a, x)` which filters a signal `x` with the filter defined by the coefficients `b` and `a`. Implementation shall follow the type I state-space equations.
5. Test the function written above with the coefficients designed at step 3, by filtering a sample audio signal.
6. Modify the function to perform temporal filtering of a video sequence, only for a filter of order 3. Test the function on the video sequence `veh_small.mp4`.

   To read frames from a video sequence in Matlab, you can use the following snippet:

```
v = VideoReader('videofile.mp4');

% Read all the frames from the video, one frame at a time.

while hasFrame(v)
 frame = readFrame(v);

 % Do the processing here

end
```

Final questions

1. TBD