Reliable verification of digital implemented filters against frequency specifications

Anastasia Volkova

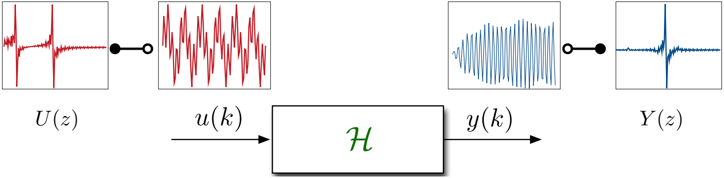
Christoph Lauter, Thibault Hilaire

Sorbonne Universités, UPMC, LIP6

RAIM 2017
Lyon, October 24-26, 2017

Linear Time-Invariant Digital Filters

- Time domain



- Frequency domain

$$H(z) = \frac{\sum_{i=0}^{n} b_i z^{-i}}{\sum_{i=0}^{n} a_i z^{-i}}, \quad z \in \mathbb{C},\ a_i, b_i \in \mathbb{R}$$

Frequency specifications

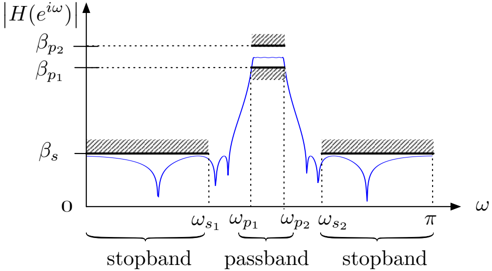
Frequency response ($z = e^{j\omega}$)

$$H\left(e^{j\omega}\right) = \underbrace{\left|H\left(e^{j\omega}\right)\right|}_{\text{magnitude}} e^{\overbrace{\measuredangle H\left(e^{j\omega}\right)}^{\text{phase}}}$$

Frequency specifications

Frequency response ($z = e^{j\omega}$)

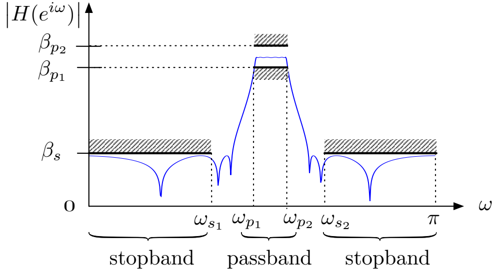
$$H\left(e^{j\omega}\right) = \underbrace{\left|H\left(e^{j\omega}\right)\right|}_{\text{magnitude}} e^{\overbrace{\measuredangle H\left(e^{j\omega}\right)}^{\text{phase}}}$$



Frequency specifications

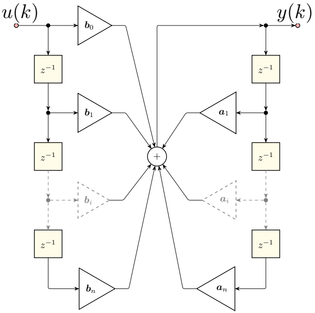
Frequency response ($z = e^{j\omega}$)

$$H\left(e^{j\omega}\right) = \underbrace{\left|H\left(e^{j\omega}\right)\right|}_{\text{magnitude}} e^{\overbrace{\measuredangle H\left(e^{j\omega}\right)}^{\text{phase}}}$$



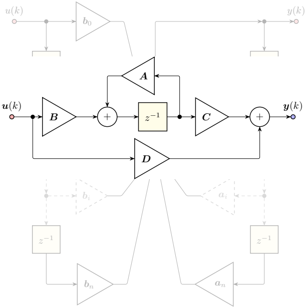
$$\underline{\beta} \leq \left|H(e^{i\omega})\right| \leq \overline{\beta}, \quad \forall \omega \in [\omega_1, \omega_2] \subseteq [0, \pi]$$

Filter evaluation

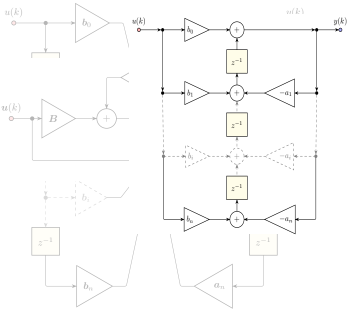


- $y(k) = \sum_{i=0}^{n} b_i u(k-i) - \sum_{i=1}^{n} a_i y(k-i)$

Filter evaluation

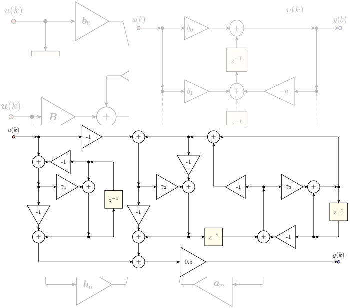


- $y(k) = \sum_{i=0}^{n} b_i u(k-i) - \sum_{i=1}^{n} a_i y(k-i)$
- $\begin{cases} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \\ y(k) & = & \boldsymbol{c}\boldsymbol{x}(k) + du(k) \end{cases}$



- $y(k) = \sum_{i=0}^{n} b_i u(k-i) - \sum_{i=1}^{n} a_i y(k-i)$
- $\begin{cases} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \\ y(k) & = & \boldsymbol{c}\boldsymbol{x}(k) + du(k) \end{cases}$
- ...

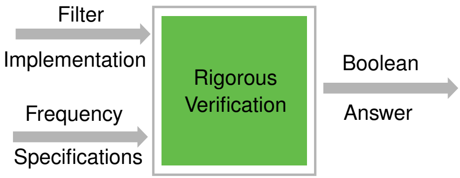
Filter evaluation



- $y(k) = \sum_{i=0}^{n} b_i u(k-i) - \sum_{i=1}^{n} a_i y(k-i)$
- $\left\{ \begin{array}{rcl} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \\ y(k) & = & \boldsymbol{c}\boldsymbol{x}(k) + du(k) \end{array} \right.$
- ...

Typical algorithm : input $u(k)$, state $\boldsymbol{x}(k)$, output $y(k)$

Goal: verify an implemented filter



Goal: verify an implemented filter

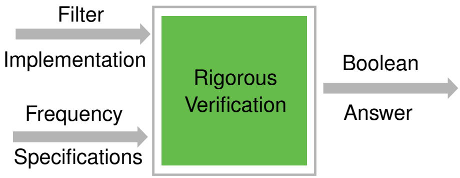
Existing approaches:

- simulations
- approximate magnitude response

Our reliable approach:

- no simulations, only proofs
- rational and interval arithmetic

Goal: verify an implemented filter



Existing approaches:

- simulations
- approximate magnitude response

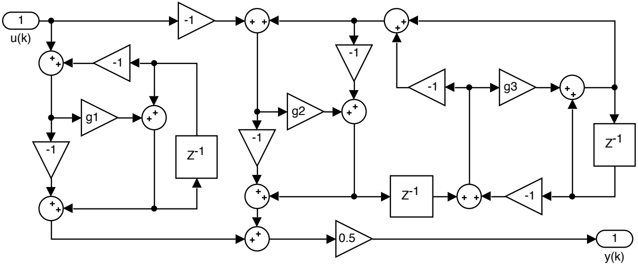
Our reliable approach:

- no simulations, only proofs
- rational and interval arithmetic

We use Computer Arithmetic to make Signal Processing rigorous.

Example

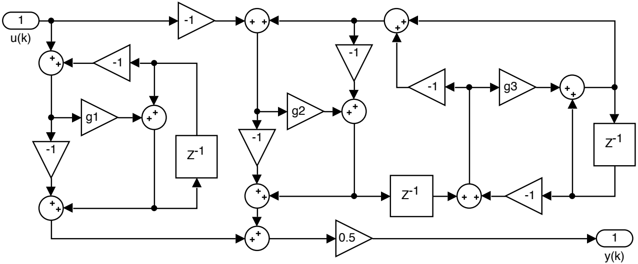
Filter:



$$g_1 = 89 \cdot 2^{-8}$$
$$g_2 = 43 \cdot 2^{-7}$$
$$g_3 = 11 \cdot 2^{-7}$$

Example

Filter:



$g_1 = 89 \cdot 2^{-8}$

$g_2 = 43 \cdot 2^{-7}$

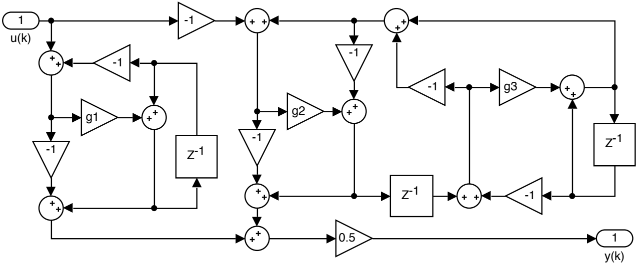
$g_3 = 11 \cdot 2^{-7}$

Specifications:

$$\begin{cases} 1\text{dB} \leq & \left|H(e^{i\omega})\right| \leq & 3\text{dB} & \forall \omega \in [0, \frac{1}{10}\pi] & \text{(passband)} \\ & \left|H(e^{i\omega})\right| \leq & -20\text{dB} & \forall \omega \in [\frac{3}{10}\pi, \pi] & \text{(stopband)} \end{cases}$$

Example

Filter:



$g_1 = 89 \cdot 2^{-8}$

$g_2 = 43 \cdot 2^{-7}$

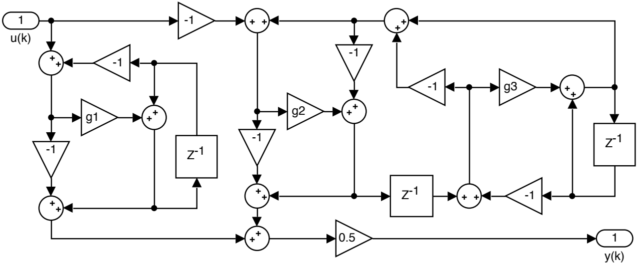
$g_3 = 11 \cdot 2^{-7}$

Specifications:

$$\begin{cases} 10^{\frac{1}{20}} \leq \left|H(e^{i\omega})\right| \leq 10^{\frac{3}{20}} & \forall \omega \in [0, \frac{1}{10}\pi] \quad \text{(passband)} \\ \left|H(e^{i\omega})\right| \leq 10^{-\frac{20}{20}} & \forall \omega \in [\frac{3}{10}\pi, \pi] \quad \text{(stopband)} \end{cases}$$

Example

Filter:



$$g_1 = 89 \cdot 2^{-8}$$
$$g_2 = 43 \cdot 2^{-7}$$
$$g_3 = 11 \cdot 2^{-7}$$

Specifications:

$$\begin{cases} 10^{\frac{1}{20}} \leq & \left|H(e^{i\omega})\right| \leq & 10^{\frac{3}{20}} & \forall \omega \in [0, \frac{1}{10}\pi] & \text{(passband)} \\ & \left|H(e^{i\omega})\right| \leq & 10^{-\frac{20}{20}} & \forall \omega \in [\frac{3}{10}\pi, \pi] & \text{(stopband)} \end{cases}$$

Transfer Function:

$$H(z) = \frac{\sum_{i=0}^{n} b_i z^{-i}}{\sum_{i=0}^{n} a_i z^{-i}}$$

Transfer function verification

Need to show that $\forall z = e^{j\omega}, \omega \in \Omega \subset [0, \pi]$

$$\underline{\beta} \leq |H(z)| \leq \overline{\beta}$$

Transfer function verification

Need to show that $\forall z = e^{j\omega}, \omega \in \Omega \subset [0, \pi]$

$$\underline{\beta}^2 \leq |H(z)|^2 \leq \overline{\beta}^2$$

Transfer function verification

Need to show that $\forall z = e^{j\omega}, \omega \in \Omega \subset [0, \pi]$

$$\underline{\beta}^2 \leq |H(z)|^2 \leq \overline{\beta}^2$$

We have that

$$|H(z)|^2 = \frac{|b(z)|^2}{|a(z)|^2} = \frac{b(z)\overline{b(\overline{z})}}{a(z)\overline{a(\overline{z})}} = \frac{b(z)b(\frac{1}{z})}{a(z)a(\frac{1}{z})} =: \frac{v(z)}{w(z)},$$

$v(z)$ and $w(z)$ have real coefficients.

Reducing the problem to a real rational function

$$\underline{\beta}^2 \leq \frac{v(z)}{w(z)} \leq \overline{\beta}^2$$

$$z = e^{j\omega}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$

Reducing the problem to a real rational function

$$\underline{\beta}^2 \leq \frac{v(z)}{w(z)} \leq \overline{\beta}^2$$

$$z = e^{j\omega}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$

We don't need to deal with complex variables

Change of variable: $t = \tan \frac{\omega}{2}$

$$z = e^{j\omega} = \cos\omega + j\sin\omega$$

Reducing the problem to a real rational function

$$\underline{\beta}^2 \leq \frac{v(z)}{w(z)} \leq \overline{\beta}^2$$

$$z = e^{j\omega}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$

⚠ We don't need to deal with complex variables

Change of variable: $t = \tan \frac{\omega}{2}$

$$z = e^{j\omega} = \frac{1-t^2}{1+t^2} + j\frac{2t}{1+t^2}$$

Reducing the problem to a real rational function

$$\underline{\beta}^2 \leq \frac{v(z)}{w(z)} \leq \overline{\beta}^2$$

$$\underline{\beta}^2 \leq \frac{v\left(\frac{1-t^2}{1+t^2} + j\frac{2t}{1+t^2}\right)}{w\left(\frac{1-t^2}{1+t^2} + j\frac{2t}{1+t^2}\right)} \leq \overline{\beta}^2$$

$$z = e^{j\omega}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$
$$\downarrow$$
$$t = \tan\frac{\omega}{2}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$

Reducing the problem to a real rational function

$$\underline{\beta}^2 \leq \frac{v(z)}{w(z)} \leq \overline{\beta}^2$$

$$\underline{\beta}^2 \leq \underbrace{\frac{r(t) + j\text{ж}(t)}{s(t) + j\text{щ}(t)}}_{\in\mathbb{R} \text{ due to } |H|^2} \leq \overline{\beta}^2$$

$$z = e^{j\omega}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$
$$\downarrow$$
$$t = \tan \frac{\omega}{2}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$

Polynomials $r, s, \text{ж}, \text{щ} \in \mathbb{R}[x]$

Reducing the problem to a real rational function

$$\underline{\beta}^2 \leq \frac{v(z)}{w(z)} \leq \overline{\beta}^2$$

$$\underline{\beta}^2 \leq \frac{r(t)}{s(t)} \leq \overline{\beta}^2$$

$$z = e^{j\omega}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$
$$\downarrow$$
$$t = \tan \frac{\omega}{2}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$

Now we work only with reals.

Reducing the problem to a real rational function

$$\underline{\beta}^2 \leq \frac{v(z)}{w(z)} \leq \overline{\beta}^2$$

$$\underline{\beta}^2 \leq \frac{r(t)}{s(t)} \leq \overline{\beta}^2$$

$z = e^{j\omega}$
$\forall \omega \in \Omega \subseteq [0, \pi]$
$\downarrow$
$t = \tan \frac{\omega}{2}$
$\forall \omega \in \Omega \subseteq [0, \pi]$

⚠ Mapping $t = \tan \frac{\omega}{2}$ maps ω onto the whole $\mathbb{R}$

Change of variable: $\xi = \frac{t+2-\sqrt{t^2+4}}{2t}$

Reducing the problem to a real rational function

$$\underline{\beta}^2 \leq \frac{v(z)}{w(z)} \leq \overline{\beta}^2$$

$$\underline{\beta}^2 \leq \frac{r(t)}{s(t)} \leq \overline{\beta}^2$$

$$\underline{\beta}^2 \leq \frac{r\left(\frac{1-2\xi}{\xi(1-\xi)}\right)}{s\left(\frac{1-2\xi}{\xi(1-\xi)}\right)} \leq \overline{\beta}^2$$

$$z = e^{j\omega}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$
$$\downarrow$$
$$t = \tan \frac{\omega}{2}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$
$$\downarrow$$
$$\xi = \frac{t+2-\sqrt{t^2+4}}{2t}$$
$$\forall \xi \in \Xi \subseteq [0, 1]$$

Reducing the problem to a real rational function

$$\underline{\beta}^2 \leq \frac{v(z)}{w(z)} \leq \overline{\beta}^2$$

$$\underline{\beta}^2 \leq \frac{r(t)}{s(t)} \leq \overline{\beta}^2$$

$$\underline{\beta}^2 \leq \frac{p(\xi)}{q(\xi)} \leq \overline{\beta}^2$$

$$z = e^{j\omega}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$
$$\downarrow$$
$$t = \tan \frac{\omega}{2}$$
$$\forall \omega \in \Omega \subseteq [0, \pi]$$
$$\downarrow$$
$$\xi = \frac{t+2-\sqrt{t^2+4}}{2t}$$
$$\forall \xi \in \Xi \subseteq [0, 1]$$

We compute the PGCD(p, q) with a rigorous heuristic of Char et al.

Reducing the verification problem to showing the non-negativity of a polynomial

Need to show $\forall \xi \in \Xi \subseteq [0, 1]$ that

$$\underline{\beta}^2 \quad \leq \qquad \frac{p(\xi)}{q(\xi)} \qquad \leq \quad \overline{\beta}^2$$

Reducing the verification problem to showing the non-negativity of a polynomial

Need to show $\forall \xi \in \Xi \subseteq [0, 1]$ that

$$\underline{\beta}^2 - \frac{\underline{\beta}^2 + \overline{\beta}^2}{2} \quad \leq \quad \frac{p(\xi)}{q(\xi)} - \frac{\underline{\beta}^2 + \overline{\beta}^2}{2} \quad \leq \quad \overline{\beta}^2 - \frac{\underline{\beta}^2 + \overline{\beta}^2}{2}$$

Reducing the verification problem to showing the non-negativity of a polynomial

Need to show $\forall \xi \in \Xi \subseteq [0, 1]$ that

$$-\frac{\overline{\beta}^2 - \underline{\beta}^2}{2} \leq \frac{p(\xi) - \left(\underline{\beta}^2 + \overline{\beta}^2\right) q(\xi)}{2q(\xi)} \leq \frac{\overline{\beta}^2 - \underline{\beta}^2}{2}$$

Reducing the verification problem to showing the non-negativity of a polynomial

Need to show $\forall \xi \in \Xi \subseteq [0, 1]$ that

$$-1 \leq \frac{2}{\overline{\beta}^2 - \underline{\beta}^2} \left(\frac{p(\xi) - \left(\underline{\beta}^2 + \overline{\beta}^2 \right) q(\xi)}{2q(\xi)} \right) \leq 1$$

Reducing the verification problem to showing the non-negativity of a polynomial

Need to show $\forall \xi \in \Xi \subseteq [0,1]$ that

$$-1 \leq \frac{g(\xi)}{h(\xi)} \leq 1$$

Reducing the verification problem to showing the non-negativity of a polynomial

Need to show $\forall \xi \in \Xi \subseteq [0,1]$ that

$$\frac{g^2(\xi)}{h^2(\xi)} \leq 1$$

Reducing the verification problem to showing the non-negativity of a polynomial

It suffices to show $\forall \xi \in \Xi \subseteq [0, 1]$ that

$$h^2(\xi) - g^2(\xi) \geq 0$$

Reducing the verification problem to showing the non-negativity of a polynomial

It suffices to show $\forall \xi \in \Xi \subseteq [0,1]$ that

$$f(\xi) \geq 0$$

All these transformations are performed exactly with rational arithmetic.

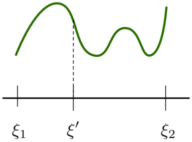
Checking non-negativity of a real polynomial

To verify $f(\xi) \geq 0, \; \forall \xi \in \Xi = [\xi_1, \xi_2] \subseteq [0, 1]$ we check if

Checking non-negativity of a real polynomial

To verify $f(\xi) \geq 0$, $\forall \xi \in \Xi = [\xi_1, \xi_2] \subseteq [0, 1]$ we check if

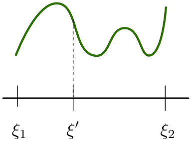
(i) $f(\xi)$ has no zeros
$f(\xi') > 0$ for some $\xi' \in [\xi_1, \xi_2]$



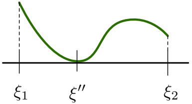
Checking non-negativity of a real polynomial

To verify $f(\xi) \geq 0$, $\forall \xi \in \Xi = [\xi_1, \xi_2] \subseteq [0, 1]$ we check if

(i) $f(\xi)$ has no zeros
$f(\xi') > 0$ for some $\xi' \in [\xi_1, \xi_2]$



(ii) $f(\xi)$ has one zero
$f(\xi_1) > 0$ and $f(\xi_2) > 0$



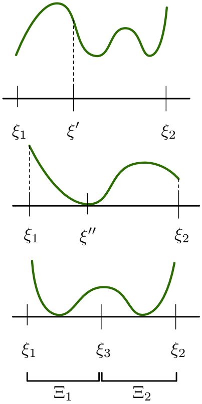
Checking non-negativity of a real polynomial

To verify $f(\xi) \geq 0$, $\forall \xi \in \Xi = [\xi_1, \xi_2] \subseteq [0, 1]$ we check if

(i) $f(\xi)$ has no zeros
$f(\xi') > 0$ for some $\xi' \in [\xi_1, \xi_2]$

(ii) $f(\xi)$ has one zero
$f(\xi_1) > 0$ and $f(\xi_2) > 0$

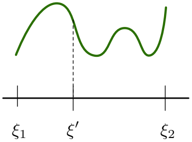
(iii) interval Ξ can be split into subintervals
s.t. (i) or (ii) are satisfied for every subinterval



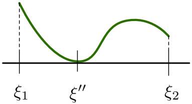
Checking non-negativity of a real polynomial

To verify $f(\xi) \geq 0,\ \forall \xi \in \Xi = [\xi_1, \xi_2] \subseteq [0, 1]$ we check if

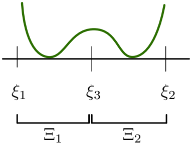
(i) $f(\xi)$ has no zeros
$f(\xi') > 0$ for some $\xi' \in [\xi_1, \xi_2]$



(ii) $f(\xi)$ has one zero
$f(\xi_1) > 0$ and $f(\xi_2) > 0$



(iii) interval Ξ can be split into subintervals
s.t. (i) or (ii) are satisfied for every subinterval



We use Sollya tool for the implementation

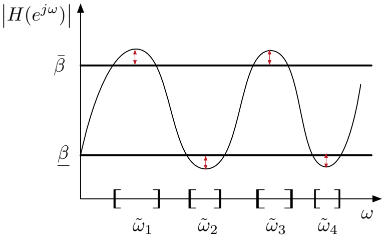
- Number of zeros: Sturm's theorem
- Evaluations: interval multiple precision arithmetic

Wrapping-Up

Does this transfer function verify the frequency specifications?

Yes

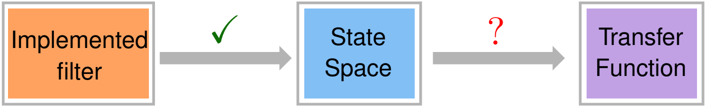
No



Verifying arbitrary filter implementation

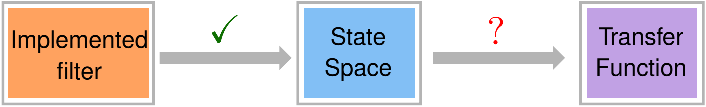
Verifying arbitrary filter implementation

Verifying arbitrary filter implementation



State-Space system:

$$\mathcal{S}\left\{\begin{array}{rcl} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \\ y(k) & = & \boldsymbol{c}\boldsymbol{x}(k) + du(k) \end{array}\right.$$



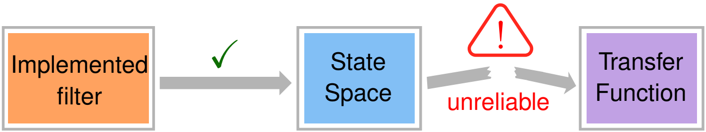
State-Space system:

$$\mathcal{S}\begin{cases} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \\ y(k) &= \boldsymbol{c}\boldsymbol{x}(k) + du(k) \end{cases}$$

Corresponding Transfer Function:

$$H(z) = \boldsymbol{c}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} + d$$

Verifying arbitrary filter implementation



State-Space system:

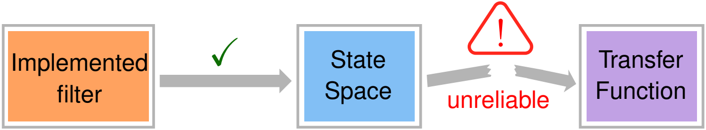
$$\mathcal{S}\begin{cases} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \\ y(k) &= \boldsymbol{c}\boldsymbol{x}(k) + du(k) \end{cases}$$

Corresponding Transfer Function:

$$H(z) = \boldsymbol{c}(z\boldsymbol{I} - \boldsymbol{V}\boldsymbol{E}\boldsymbol{V}^{-1})^{-1}\boldsymbol{b} + d$$

can be approximated using the eigendecomposition of $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{E}\boldsymbol{V}^{-1}$

Verifying arbitrary filter implementation



State-Space system:

$$\mathcal{S}\begin{cases} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \\ y(k) &= \boldsymbol{c}\boldsymbol{x}(k) + du(k) \end{cases}$$

Corresponding Transfer Function:

$$H(z) = \boldsymbol{c}(z\boldsymbol{I} - \boldsymbol{V}\boldsymbol{E}\boldsymbol{V}^{-1})^{-1}\boldsymbol{b} + d$$

can be approximated using the eigendecomposition of $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{E}\boldsymbol{V}^{-1}$

Need to:

- Compute an approximation $\widehat{H}(z)$ with arbitrary precision (`mpmath`)
- Exhibit a reliable bound on the approximation error $\left|\left(H - \widehat{H}\right)(e^{j\omega})\right|$

Bounding the approximation error

Computing the transfer function $H(z)$ of the state-space system $\mathcal{S}$:

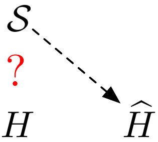
$$\mathcal{S}$$

$$\textcolor{red}{?}$$

$$H$$

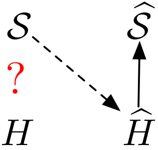
Bounding the approximation error

Computing the transfer function $H(z)$ of the state-space system $\mathcal{S}$:



Bounding the approximation error

Computing the transfer function $H(z)$ of the state-space system $\mathcal{S}$:



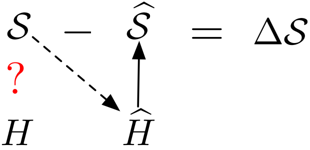
Transformation from $\widehat{H}$ to $\widehat{\mathcal{S}}$ is exact:

$$\widehat{\boldsymbol{A}} = \begin{pmatrix} -\widehat{a}_1 & 1 & & \\ \vdots & & \ddots & \\ \vdots & & & 1 \\ -\widehat{a}_n & 0 & \dots & 0 \end{pmatrix} \quad \widehat{\boldsymbol{b}} = \begin{pmatrix} \widehat{b}_1 - \widehat{a}_1\widehat{b}_0 \\ \vdots \\ \vdots \\ \widehat{b}_n - \widehat{a}_n\widehat{b}_0 \end{pmatrix}$$

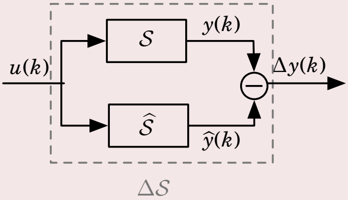
$$\widehat{\boldsymbol{c}} = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} \qquad \widehat{d} = \widehat{b}_0$$

Bounding the approximation error

Computing the transfer function $H(z)$ of the state-space system $\mathcal{S}$:



Difference of filters is defined as:



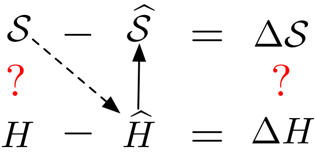
Bounding the approximation error

Computing the transfer function $H(z)$ of the state-space system $\mathcal{S}$:

$$
\begin{array}{ccccc}
\mathcal{S} & - & \widehat{\mathcal{S}} & = & \Delta\mathcal{S} \\
? & & \uparrow & & ? \\
H & - & \widehat{H} & = & \Delta H
\end{array}
$$

Bounding the approximation error

Computing the transfer function $H(z)$ of the state-space system $\mathcal{S}$:



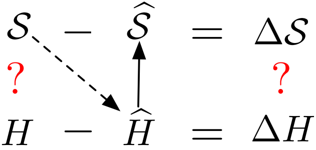
Relation between $\Delta\mathcal{S}$ and ΔH:

$$\left| \left(H - \widehat{H} \right) \left(e^{j\omega} \right) \right| \leq \langle\langle \Delta\mathcal{S} \rangle\rangle \, , \quad \forall \omega \in [0, 2\pi]$$

where $\langle\langle \Delta\mathcal{S} \rangle\rangle$ is the Worst-Case Peak Gain of the system $\Delta\mathcal{S}$.

Bounding the approximation error

Computing the transfer function $H(z)$ of the state-space system $\mathcal{S}$:



Relation between $\Delta\mathcal{S}$ and ΔH:

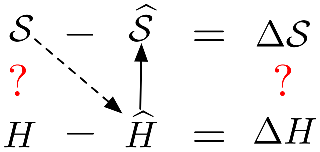
$$\left|\left(H - \widehat{H}\right)\left(e^{j\omega}\right)\right| \leq \langle\langle \Delta\mathcal{S} \rangle\rangle\,, \quad \forall\omega \in [0, 2\pi]$$

where $\langle\langle \Delta\mathcal{S} \rangle\rangle$ is the Worst-Case Peak Gain of the system $\Delta\mathcal{S}$.

We can evaluate $\langle\langle \Delta\mathcal{S} \rangle\rangle$ with *a priori* error ε [ARITH2015].

Bounding the approximation error

Computing the transfer function $H(z)$ of the state-space system $\mathcal{S}$:



Relation between $\Delta\mathcal{S}$ and ΔH:

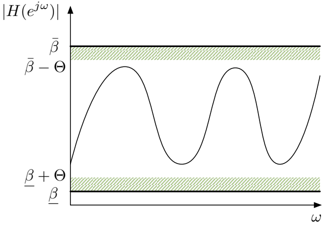
$$\left|\left(H - \widehat{H}\right)\left(e^{j\omega}\right)\right| \leq \langle\langle \Delta\mathcal{S} \rangle\rangle, \quad \forall\omega \in [0, 2\pi]$$

where $\langle\langle \Delta\mathcal{S} \rangle\rangle$ is the Worst-Case Peak Gain of the system $\Delta\mathcal{S}$.

We can evaluate $\langle\langle \Delta\mathcal{S} \rangle\rangle$ with *a priori* error ε [ARITH2015].

We obtain a multiple precision approximation $\widehat{H}$ on the transfer function with a reliable error bound.

Verifying a LTI filter implementation



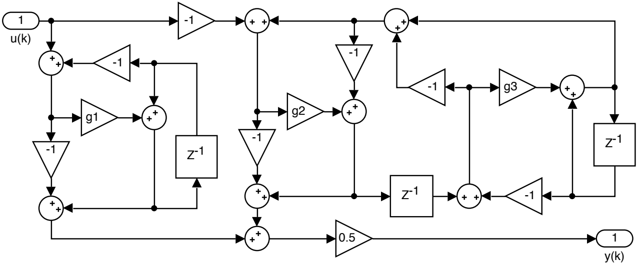
Taking approximation error into account

Narrow the bounds by $\Theta = \langle\langle \Delta \mathcal{S} \rangle\rangle + \varepsilon$ and verify the approximation $\widehat{H}(z)$ against updated specifications:

$$\underline{\beta} + \Theta \leq \left| \widehat{H}(e^{i\omega}) \right| \leq \overline{\beta} - \Theta, \quad \forall \omega \in \Omega$$

Numerical results: example 1

Filter implementation:



$g_1 = 89 \cdot 2^{-8}$

$g_2 = 43 \cdot 2^{-7}$

$g_3 = 11 \cdot 2^{-7}$

Specifications:

$$\begin{cases} 10^{\frac{1}{20}} \leq & \left|H(e^{i\omega})\right| \leq & 10^{\frac{3}{20}} & \forall \omega \in [0, \frac{1}{10}\pi] & \text{(passband)} \\ & \left|H(e^{i\omega})\right| \leq & 10^{-\frac{20}{20}} & \forall \omega \in [\frac{3}{10}\pi, \pi] & \text{(stopband)} \end{cases}$$

Verification result: implemented filter *passed* the verification against frequency specifications

Verification time: 1.9 s

Numerical results: example 2

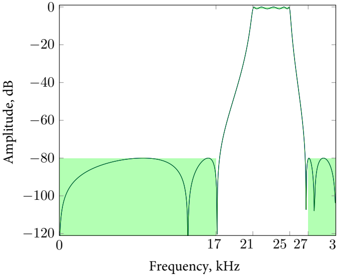
Filter implementation: 14$^{\text{th}}$ order bandpass filter
Specifications:

$$\begin{cases} & \left|H(e^{i\omega})\right| \leq & -80\text{dB} & \forall\omega \in [0, 17\text{kHz}] & \text{(stopband)} \\ 0dB \leq & \left|H(e^{i\omega})\right| \leq & 1 - 10^{-4}\text{dB} & \forall\omega \in [21\text{kHz}, 25\text{kHz}] & \text{(passband)} \\ & \left|H(e^{i\omega})\right| \leq & -80\text{dB} & \forall\omega \in [27\text{kHz}, 30\text{kHz}] & \text{(stopband)} \end{cases}$$

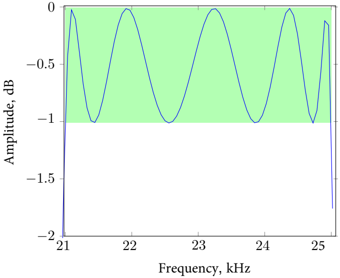
Verification result: implemented filter *does not* pass the verification against frequency constraints
Verification time: 53 s

Numerical results: example 2

Filter implementation: 14^{th} order bandpass filter
Verification result: implemented filter *does not* pass the verification against frequency constraints
Verification time: 53 s
Frequency response:

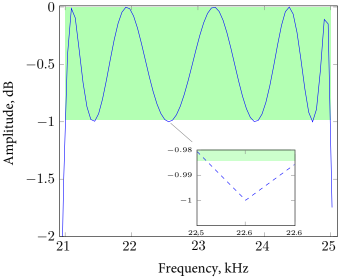


Filter implementation: 14^{th} order bandpass filter
Verification result: implemented filter *does not* pass the verification against frequency constraints
Verification time: 53 s
Frequency response:



Numerical results: example 2

Filter implementation: 14^{th} order bandpass filter
Verification result: implemented filter *does not* pass the verification against frequency constraints
Verification time: 53 s
Frequency response:



Numerical results: example 3

Verification of the 9^{th} order FIR filter from Silviu's presentation:

- coefficients quantized to 7 bits
- error on the transfer function is roughly 0.047
- passband $[0, \frac{1}{3}\pi]$, stopband $[0.5\pi, \pi]$

Numerical results: example 3

Verification of the 9^{th} order FIR filter from Silviu's presentation:

- coefficients quantized to 7 bits
- error on the transfer function is roughly 0.047
- passband $[0, \frac{1}{3}\pi]$, stopband $[0.5\pi, \pi]$

Result:

```
Overall check okay:  true
Computing this result took 7209ms
```

Conclusion and Perspectives

Conclusion:

- Reliable *a posteriori* verification of any implemented linear filter
- Multiple precision approximation of any filter's transfer function
- Approximation errors of the transfer function are fully accounted for
- Algorithm implemented using a combination of rational and interval arithmetic in Sollya
- Use-cases: verification and comparison of implementations, verification on design-stage, verification of design methods

Perspectives:

- Improve algorithm timings
- Prove our implementation with Coq
- Exploit information on the problematic frequencies for more robust design and implementation

Thank you!
Questions?

Transfer Function of a State-Space

Transfer function of a single-input single-output state-space $\mathcal{S}$:

$$H(z) = \boldsymbol{c}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} + d$$

Using the eigendecomposition $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{E}\boldsymbol{V}^{-1}$:

$$H(z) = \frac{P(z)}{Q(z)} + d$$

$$P(z) = \sum_{i=1}^{n} (\boldsymbol{c}\boldsymbol{V})_i (\boldsymbol{V}^{-1}\boldsymbol{b})_i \prod_{j \neq i} (z - \lambda_j)$$

$$Q(z) = \prod_{j=1}^{n} (z - \lambda_j)$$

We compute an approximation $\widehat{H}(z)$ in Multiple Precision arithmetic.

Numerical results

Input: four realizations of the same filter
Problem: verify realizations after coefficient quantization to 32/16/8 bits
Results:

	wordlength	32	16	8
DFIIt	margin	✓	unstable	unstable
	time	12.49s	-	-
ρ DFIIt	margin	✓	✓	4.68e-3 dB
	time	13.12s	4.19s	104.01s
State-Space Balanced	margin	6.16e-10 dB	✓	6.71e-1 dB
	time	12.27s	18.18s	92.05s
Lattice Wave	margin	3.80e-10 dB	✓	1.73e-2 dB
	time	920.88s	4.58s	200.83s

Numerical results:

Input: four simple frequency specifications
Problem: Verify and compare transfer function design methods.
Results: comparison of SciPy in Python and Matlab

		Butterworth	Chebyshev	Elliptic
		margin (dB)	margin (dB)	margin (dB)
lowpass	Matlab	1.29e-17	7.93e-17	✓
	SciPy	2.14e-15	4.48e-2	4.48e-2
highpass	Matlab	2.77e-16	6.94e-17	4.48e-2
	SciPy	3.02e-15	2.29e-16	4.48e-2
bandpass	Matlab	3.04e-17	✓	✓
	SciPy	✓	4.48e-2	4.48e-2
bandstop	Matlab	4.59e-16	3.09e-15	✓
	SciPy	✓	6.36e-15	7.02e-6

Sturm's sequence is a sequence of polynomials $p_0(x), \ldots, p_m(x)$:

$$
\begin{aligned}
p_0(x) &= p(x) \\
p_1(x) &= p'(x) \\
p_2(x) &= -rem(p_0, p_1) = p_1(x)q_0(x) - p_0(x), \\
p_3(x) &= -rem(p_1, p_2) = p_2(x)q_1(x) - p_1(x), \\
&\ldots \\
0 &= -rem(p_{m-1}, p_m)
\end{aligned}
$$