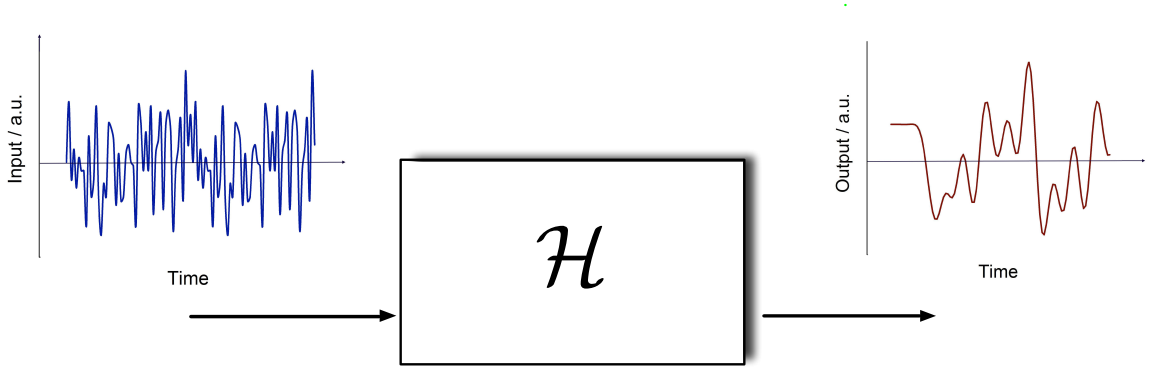
Computing the Worst-Case Peak Gain of Digital Filter in Interval Arithmetic

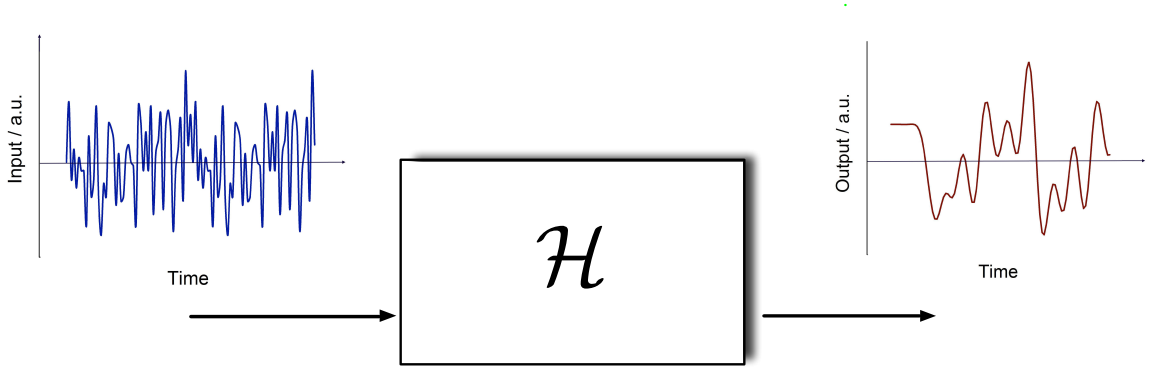
Anastasia Volkova, **Christoph Lauter**, Thibault Hilaire

SCAN 2016
September 28, 2016

Context: digital filters



Context: digital filters

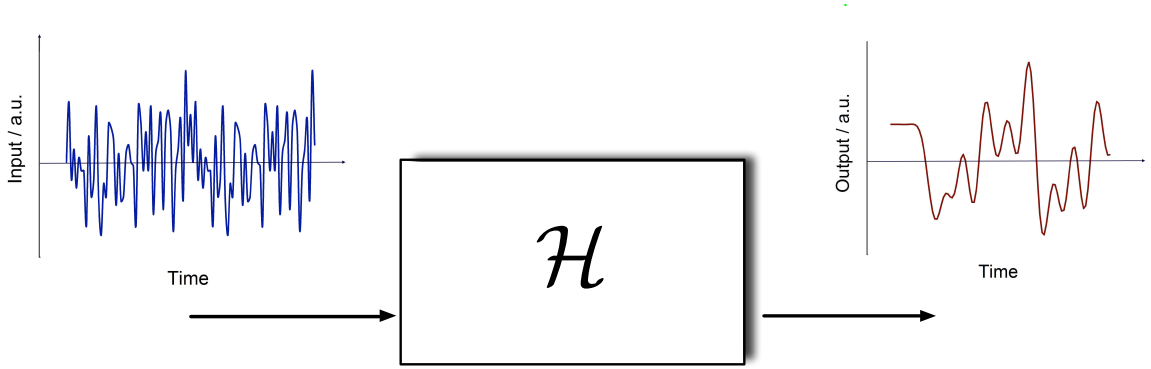


On the one hand

- LTI filter with Infinite Impulse Response
- Its transfer function:

$$H(z) = \frac{\sum_{i=0}^{n} b_i z^{-i}}{1 + \sum_{i=1}^{n} a_i z^{-i}}$$

Context: digital filters



On the one hand

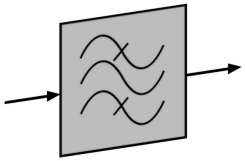
- LTI filter with Infinite Impulse Response
- Its transfer function:

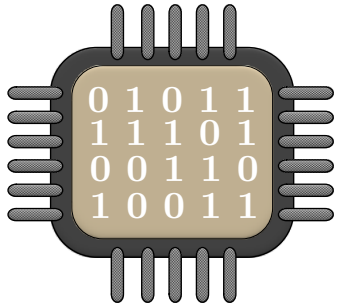
$$H(z) = \frac{\sum_{i=0}^{n} b_i z^{-i}}{1 + \sum_{i=1}^{n} a_i z^{-i}}$$

On the other hand

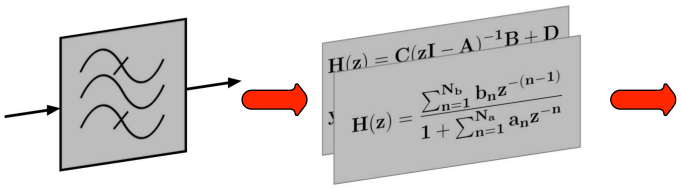
- Hardware or Software target
- Implementation in Fixed-Point Arithmetic

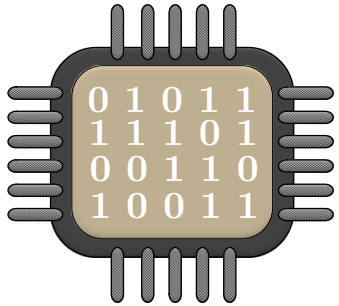
Context: implementation of LTI filters





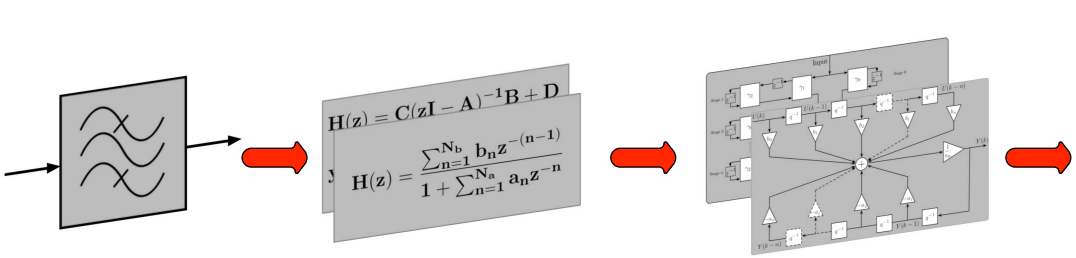
Context: implementation of LTI filters

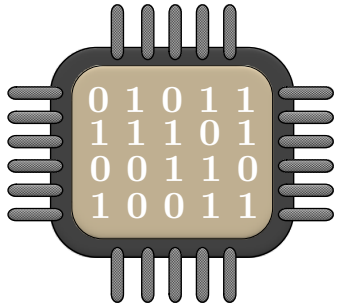




- Transfer function generation

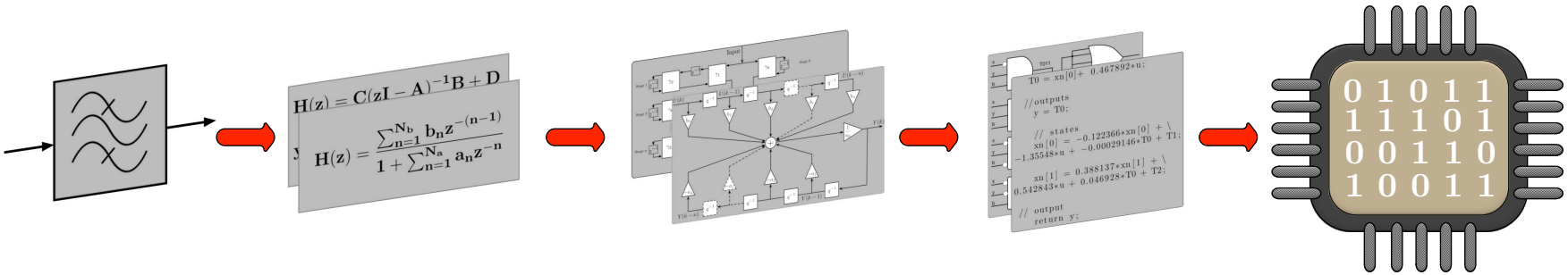
Context: implementation of LTI filters





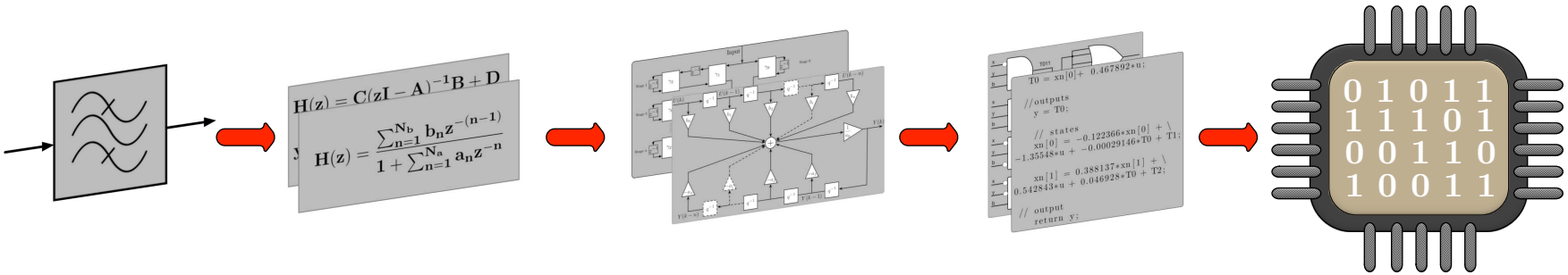
- Transfer function generation
- Algorithm choice: State-space, Direct Form I, Direct Form II, ...

Context: implementation of LTI filters



- Transfer function generation
- Algorithm choice: State-space, Direct Form I, Direct Form II, ...
- Software or Hardware implementation

Context: implementation of LTI filters



- Transfer function generation
  - ! Coefficient quantization
- Algorithm choice: State-space, Direct Form I, Direct Form II, ...
  - ! Large variety of structures with no common quality criteria
- Software or Hardware implementation
  - ! Constraints: power consumption, area, error, speed, etc.
  - ! Computational errors due to finite-precision implementation

Filter-to-code generator

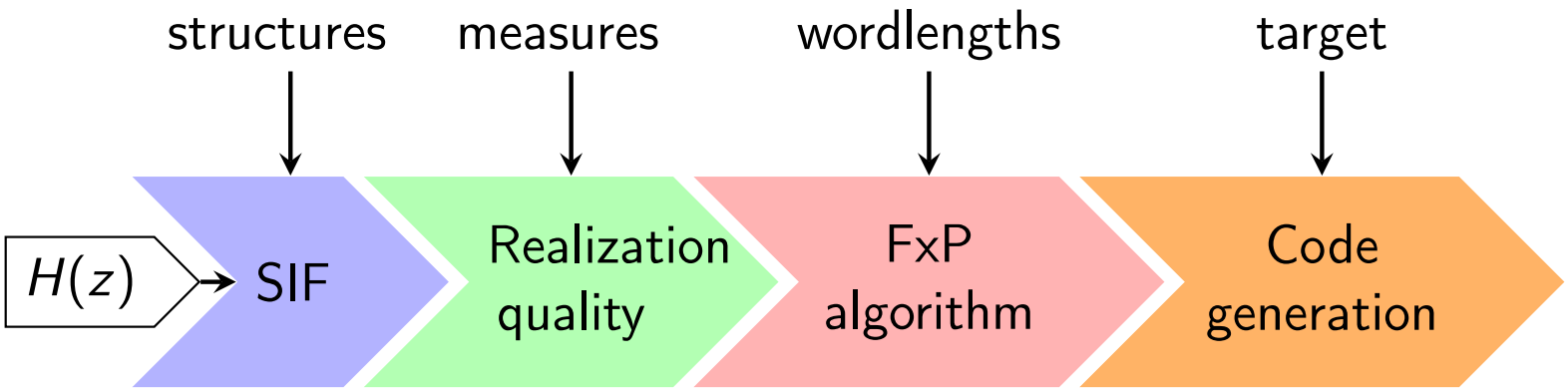


Figure: Automatic filter generator flow.

Filter-to-code generator

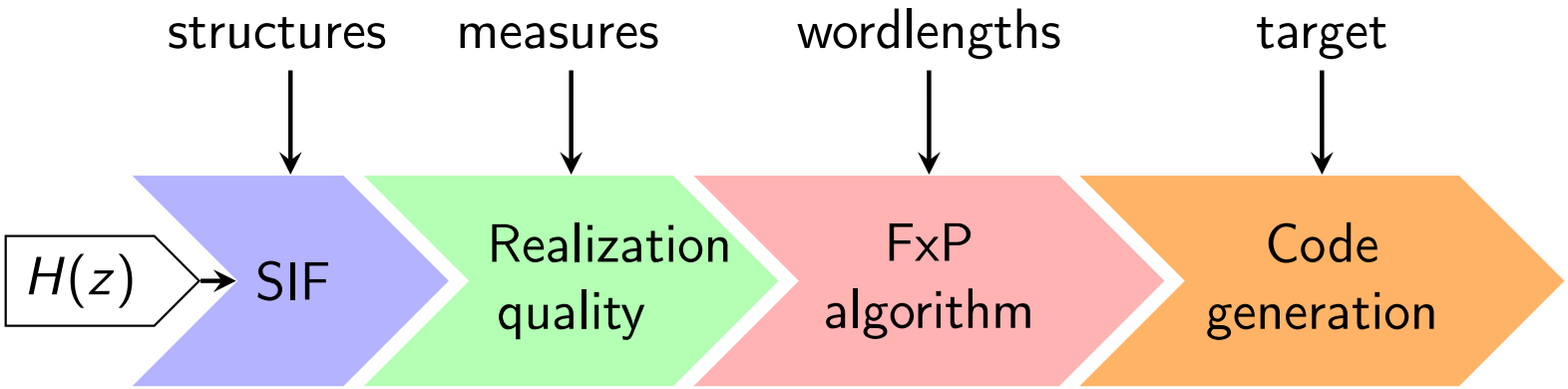


Figure: Automatic filter generator flow.

Stage 1: analytical filter realization representation

Stage 2: filter quality measures

Stage 3: reliable fixed-point algorithm (rigorous approach, computational errors taken into account)

Stage 4: Fixed-Point Code Generator

Filter-to-code generator

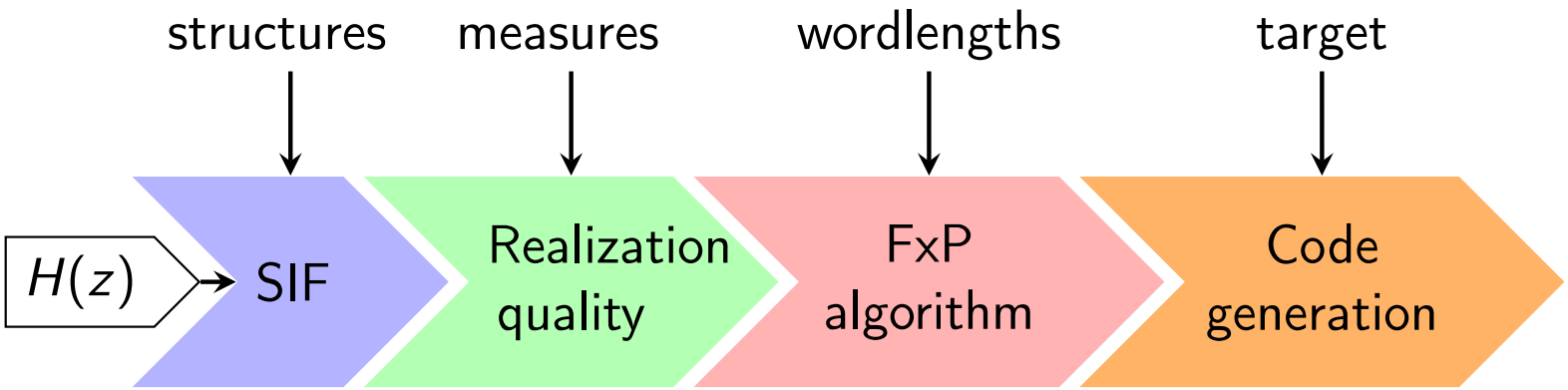


Figure: Automatic filter generator flow.

Stage 1: analytical filter realization representation

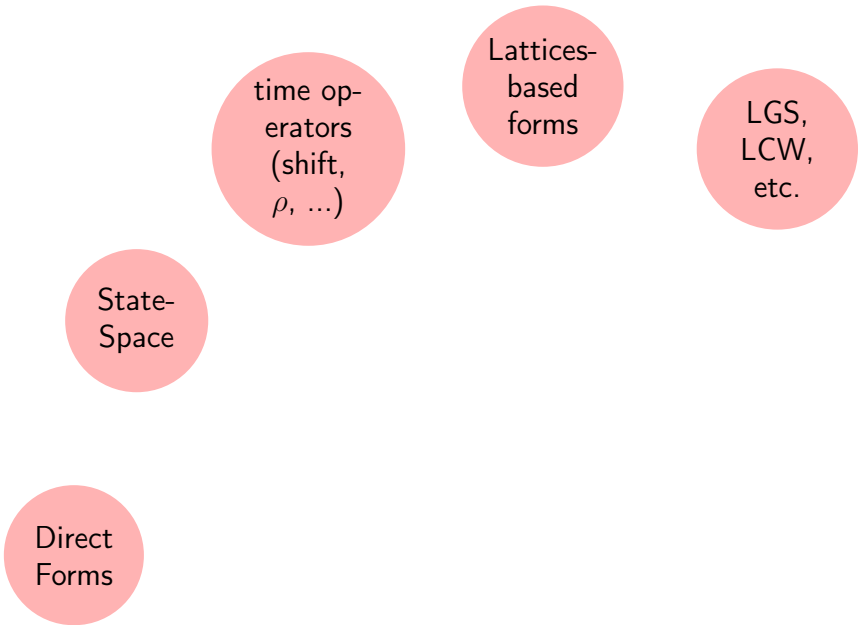
Stage 2: filter quality measures

Stage 3: reliable fixed-point algorithm (rigorous approach, computational errors taken into account)

Stage 4: Fixed-Point Code Generator

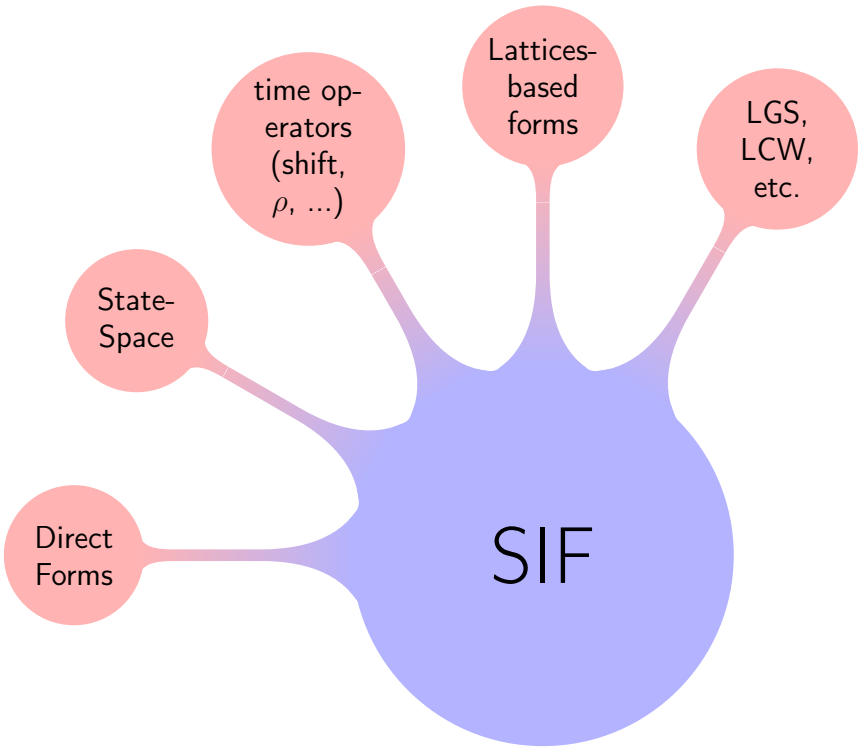
SIF and State-Space

A linear signal processing or control algorithm can be implemented under various structures (algorithms).



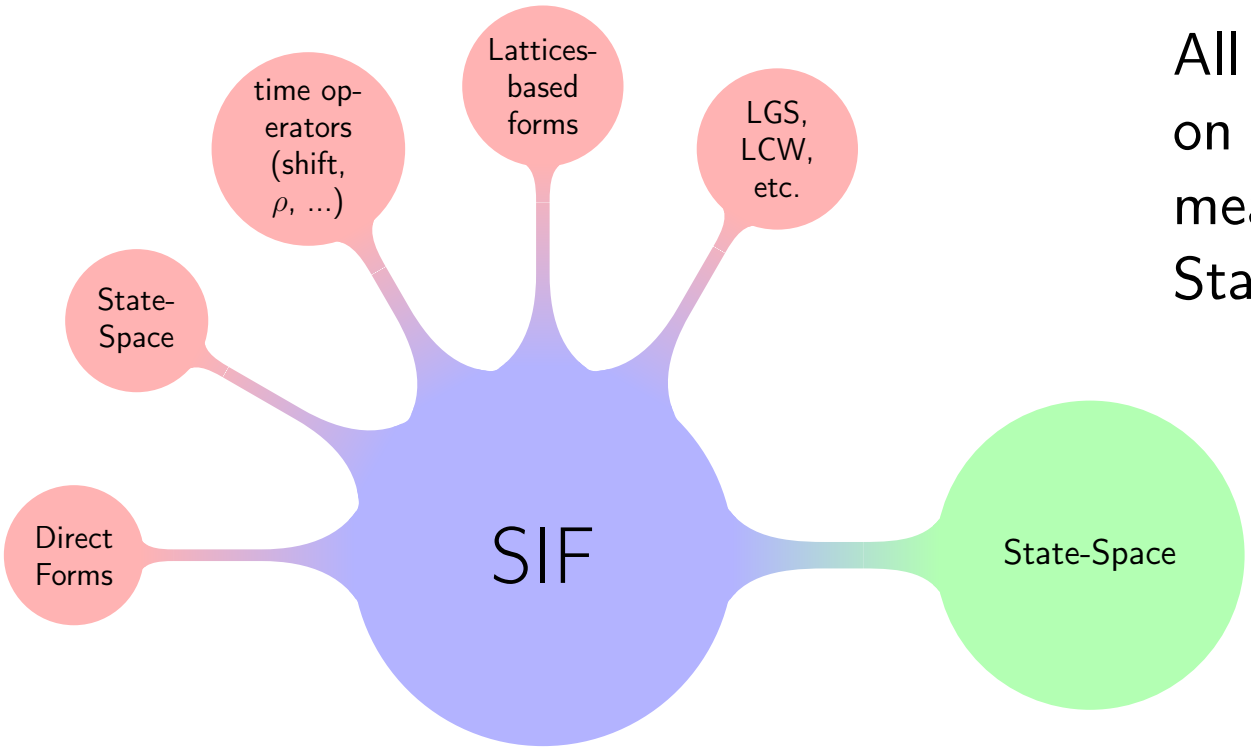
SIF and State-Space

A linear signal processing or control algorithm can be implemented under various structures (algorithms).
They can be all encompassed in a *matrix representation* called SIF (Specialized Implicit Framework).



SIF and State-Space

A linear signal processing or control algorithm can be implemented under various structures (algorithms).
They can be all encompassed in a *matrix representation* called SIF (Specialized Implicit Framework).



All our measures and analysis on SIF can be transformed on measures on different State-Spaces.

LTI filters

Let $\mathcal{H} := (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ be a LTI filter in state-space representation:

$$\mathcal{H} \begin{cases} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) \end{cases}$$

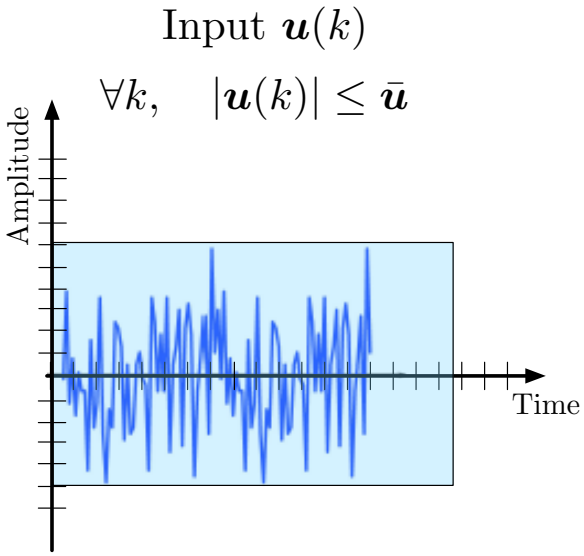
The filter $\mathcal{H}$ is considered Bounded Input Bounded Output stable if

$$\rho(\boldsymbol{A}) < 1.$$

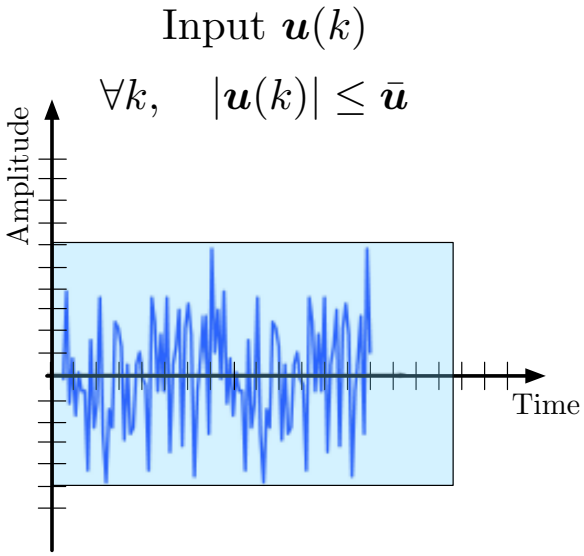
Reliable implementation:

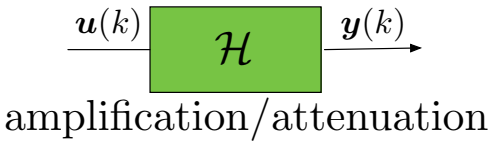
- determine the output interval
- take into account the computational error propagation and determine the Fixed-Point implementation parameters

Basic brick: the Worst-Case Peak Gain theorem



Basic brick: the Worst-Case Peak Gain theorem



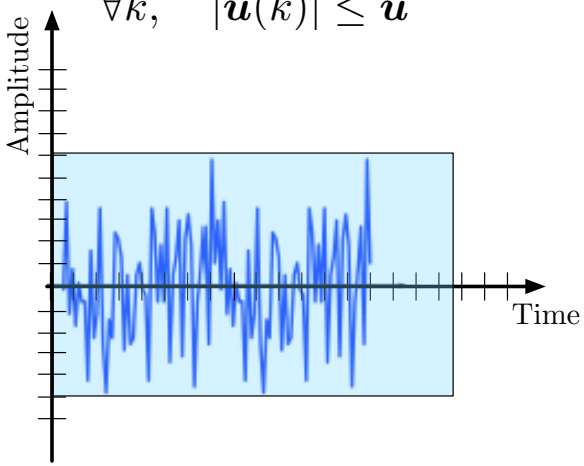


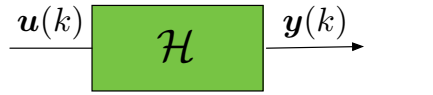
amplification/attenuation

Basic brick: the Worst-Case Peak Gain theorem

Input $\boldsymbol{u}(k)$

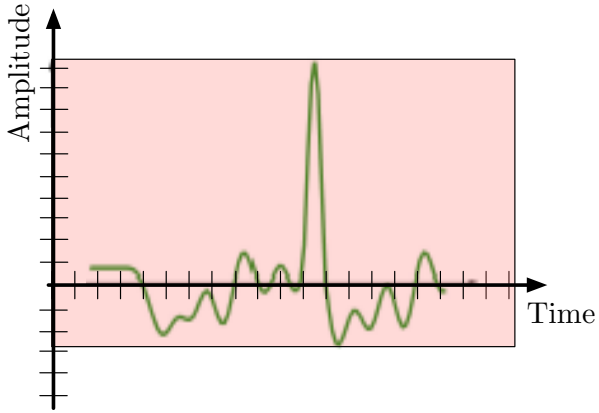
$\forall k, \quad |\boldsymbol{u}(k)| \leq \bar{\boldsymbol{u}}$





amplification/attenuation

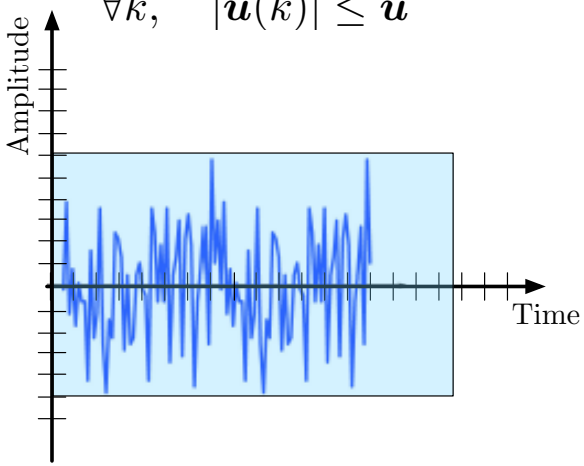
Output $\boldsymbol{y}(k)$

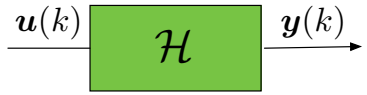


Basic brick: the Worst-Case Peak Gain theorem

Input $\boldsymbol{u}(k)$

$\forall k, \quad |\boldsymbol{u}(k)| \leq \bar{\boldsymbol{u}}$

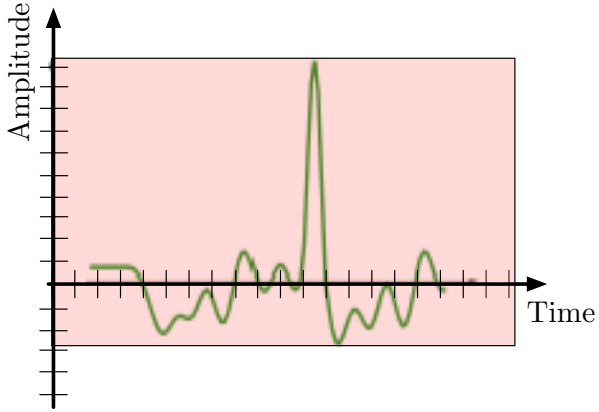




amplification/attenuation

Output $\boldsymbol{y}(k)$

$\forall k, \quad |\boldsymbol{y}(k)| \leq \langle\langle \mathcal{H} \rangle\rangle \, \bar{\boldsymbol{u}}$



Worst-Case Peak Gain

$$\langle\langle \mathcal{H} \rangle\rangle = |\mathbf{D}| + \sum_{k=0}^{\infty} |\mathbf{C}\mathbf{A}^k\mathbf{B}|$$

Computing the Worst-Case Peak Gain (WCPG)

When $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$ are exact:

$\longrightarrow$ we compute the Worst-Case Peak Gain with arbitrary precision[1].

[1] A.V. et al., "Reliable Evaluation of the Worst-Case Peak Gain Matrix in Multiple Precision", ARITH22, 2015

Computing the Worst-Case Peak Gain (WCPG)

When $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$ are exact:

$\longrightarrow$ we compute the Worst-Case Peak Gain with arbitrary precision[1].

Cases when $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$ are not exact:

- coefficients are results of finite-precision computations
  (e.g. quantization, SIF $\leftrightarrow$ State-Space transformation etc.)

[1] A.V. et al., "Reliable Evaluation of the Worst-Case Peak Gain Matrix in Multiple Precision", ARITH22, 2015

Computing the Worst-Case Peak Gain (WCPG)

When $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$ are exact:

$\longrightarrow$ we compute the Worst-Case Peak Gain with arbitrary precision[1].

Cases when $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$ are not exact:

- coefficients are results of finite-precision computations
  (e.g. quantization, SIF $\leftrightarrow$ State-Space transformation etc.)

To take these properties into account we use Interval Arithmetic.
$\longrightarrow$ Need to compute the WCPG in interval arithmetic.

[1] A.V. et al., "Reliable Evaluation of the Worst-Case Peak Gain Matrix in Multiple Precision", ARITH22, 2015

Computing the Worst-Case Peak Gain (WCPG)

When $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$ are exact:

$\longrightarrow$ we compute the Worst-Case Peak Gain with arbitrary precision[1].

Cases when $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$ are not exact:

- coefficients are results of finite-precision computations (e.g. quantization, SIF $\leftrightarrow$ State-Space transformation etc.)

To take these properties into account we use Interval Arithmetic.
$\longrightarrow$ Need to compute the WCPG in interval arithmetic.

Notation: interval matrix $\boldsymbol{M}^{\mathcal{I}}$ is centered at $mid(\boldsymbol{M}^{\mathcal{I}})$ and has radius $rad(\boldsymbol{M}^{\mathcal{I}})$.

[1] A.V. et al., "Reliable Evaluation of the Worst-Case Peak Gain Matrix in Multiple Precision", ARITH22, 2015

Interval WCPG computation

Problem: compute the interval Worst-Case Peak Gain matrix

$$\left\langle\left\langle \mathcal{H}^{\mathcal{I}} \right\rangle\right\rangle = \left|\boldsymbol{D}^{\mathcal{I}}\right| + \sum_{k=0}^{\infty} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}} \right|.$$

Interval WCPG computation

Problem: compute the interval Worst-Case Peak Gain matrix

$$\left\langle\left\langle \mathcal{H}^{\mathcal{I}} \right\rangle\right\rangle = \left|\boldsymbol{D}^{\mathcal{I}}\right| + \sum_{k=0}^{\infty} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}} \right|.$$

Approach:

- Cannot sum infinitely $\Rightarrow$ need to truncate the sum
- Evaluate the truncated sum $\left\langle\left\langle \mathcal{H}_N^{\mathcal{I}} \right\rangle\right\rangle$ using multiple precision interval arithmetic

Interval WCPG computation

Problem: compute the interval Worst-Case Peak Gain matrix

$$\left\langle\left\langle \mathcal{H}^{\mathcal{I}} \right\rangle\right\rangle = \left|\boldsymbol{D}^{\mathcal{I}}\right| + \sum_{k=0}^{\infty} \left|\boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}}\right|.$$

Approach:

- Cannot sum infinitely $\Rightarrow$ need to truncate the sum
- Evaluate the truncated sum $\left\langle\left\langle \mathcal{H}_N^{\mathcal{I}} \right\rangle\right\rangle$ using multiple precision interval arithmetic

Ensure:

- enclosure property: $\forall \left\langle\left\langle \mathcal{H} \right\rangle\right\rangle \in \left\langle\left\langle \mathcal{H}^{\mathcal{I}} \right\rangle\right\rangle \Longrightarrow \left\langle\left\langle \mathcal{H} \right\rangle\right\rangle \in \left\langle\left\langle \mathcal{H}_N^{\mathcal{I}} \right\rangle\right\rangle$
- if coefficients' radii $\to 0$ and the precision $\to \infty$, then $\left\langle\left\langle \mathcal{H}_N^{\mathcal{I}} \right\rangle\right\rangle$ is a ε-neighbourhood of the exact WCPG matrix for arbitrary $\varepsilon > 0$

Truncation

$$\sum_{k=0}^{\infty} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}^k} \boldsymbol{B}^{\mathcal{I}} \right| \quad \longrightarrow \quad \sum_{k=0}^{N} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}^k} \boldsymbol{B}^{\mathcal{I}} \right|$$

Truncation

$$\left| mid \left(\sum_{k=0}^{\infty} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}} \right| \quad - \quad \sum_{k=0}^{N} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}} \right| \right) \right| \leq \varepsilon_1$$

Compute an approximate lower bound on truncation order N such that the truncation error of the center matrix is smaller than ε_1.

Lower bound on truncation order N

$$N \geq \left\lceil \frac{\log \frac{\varepsilon_1}{\inf \boldsymbol{M}^{\mathcal{I}}}}{\log \rho(\boldsymbol{A}^{\mathcal{I}})} \right\rceil, \quad \text{with } \boldsymbol{M}^{\mathcal{I}} := \sum_{i=1}^{n} \frac{\left| \boldsymbol{R}_i^{\mathcal{I}} \right|}{1 - \left| \lambda_i^{\mathcal{I}} \right|} \frac{\left| \lambda_i^{\mathcal{I}} \right|}{\rho(\boldsymbol{A}^{\mathcal{I}})}$$

where

$(\boldsymbol{R}_i^{\mathcal{I}})_{kl} := (\boldsymbol{C}^{\mathcal{I}} \boldsymbol{V}^{\mathcal{I}})_{ki} (\boldsymbol{V}^{\mathcal{I}^{-1}} \boldsymbol{B}^{\mathcal{I}})_{il}$ – i^{th}residue matrix

$\lambda^{\mathcal{I}}, \boldsymbol{V}^{\mathcal{I}}$ – enclosures for the eigenvalues and eigenvectors of matrix $\boldsymbol{A}^{\mathcal{I}}$

Computing the eigensystem of interval matrix

Eigenvalues of interval matrix

Compute enclosures $\boldsymbol{\lambda}^{\mathcal{I}}$ such that $\forall \boldsymbol{A} \in \boldsymbol{A}^{\mathcal{I}}, \boldsymbol{\lambda}(\boldsymbol{A}) \in \boldsymbol{\lambda}^{\mathcal{I}}$

Approach

Following the works of Xu and Rachid (1996) and Rohn(1998), use the Generalized Gershgorin's Circles theorem.

Computing the eigensystem of interval matrix

Eigenvalues of interval matrix

Compute enclosures $\boldsymbol{\lambda}^{\mathcal{I}}$ such that $\forall \boldsymbol{A} \in \boldsymbol{A}^{\mathcal{I}}, \boldsymbol{\lambda}(\boldsymbol{A}) \in \boldsymbol{\lambda}^{\mathcal{I}}$

Approach

Following the works of Xu and Rachid (1996) and Rohn(1998), use the Generalized Gershgorin's Circles theorem.

Eigenvectors of interval matrix

Given the enclosures on eigenvalues $\boldsymbol{\lambda}^{\mathcal{I}}$, compute enclosures $\boldsymbol{V}^{\mathcal{I}}$ such that $\forall \boldsymbol{\lambda} \in \boldsymbol{\lambda}^{\mathcal{I}}, \forall \boldsymbol{A} \in \boldsymbol{A}^{\mathcal{I}}$ if $\boldsymbol{A}\boldsymbol{\lambda} = \boldsymbol{A}\boldsymbol{V}$, then $\boldsymbol{V} \in \boldsymbol{V}^{\mathcal{I}}$.

Approach

Use Rump's theory of Verified Inclusions.

Evaluating the truncated sum

Once the sum is truncated, we need to compute $\sum_{k=0}^{N} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}} \right|$.

- Take into account the truncation error by adding it to the radii of the computed interval WCPG $\left\langle \left\langle \widehat{\mathcal{H}}_N^{\mathcal{I}} \right\rangle \right\rangle$

Evaluating the truncated sum

Once the sum is truncated, we need to compute $\sum_{k=0}^{N} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}} \right|$.

- Take into account the truncation error by adding it to the radii of the computed interval WCPG $\left\langle \left\langle \widehat{\mathcal{H}}_N^{\mathcal{I}} \right\rangle \right\rangle$
- Naive powering a dense matrix $\boldsymbol{A}^{\mathcal{I}k}$ to large k yeilds wide intervals

Evaluating the truncated sum

Once the sum is truncated, we need to compute $\sum_{k=0}^{N} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}} \right|$.

- Take into account the truncation error by adding it to the radii of the computed interval WCPG $\left\langle \left\langle \widehat{\mathcal{H}}_N^{\mathcal{I}} \right\rangle \right\rangle$
- Naive powering a dense matrix $\boldsymbol{A}^{\mathcal{I}k}$ to large k yeilds wide intervals
  $\Longrightarrow$ diagonalize the interval matrix using eigendecomposition $\boldsymbol{\Lambda}^{\mathcal{I}}, \boldsymbol{V}^{\mathcal{I}}$

Evaluating the truncated sum

Once the sum is truncated, we need to compute $\sum_{k=0}^{N} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}} \right|$.

- Take into account the truncation error by adding it to the radii of the computed interval WCPG $\left\langle \left\langle \widehat{\mathcal{H}}_N^{\mathcal{I}} \right\rangle \right\rangle$
- Naive powering a dense matrix $\boldsymbol{A}^{\mathcal{I}k}$ to large k yeilds wide intervals

  $\Longrightarrow$ diagonalize the interval matrix using eigendecomposition $\boldsymbol{\Lambda}^{\mathcal{I}}, \boldsymbol{V}^{\mathcal{I}}$

  $\Longrightarrow$ now we compute $\sum_{k=0}^{N} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{V}^{\mathcal{I}} \boldsymbol{\Lambda}^{\mathcal{I}k} \boldsymbol{V}^{\mathcal{I}-1} \boldsymbol{B}^{\mathcal{I}} \right|$

Evaluating the truncated sum

Once the sum is truncated, we need to compute $\sum_{k=0}^{N} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{A}^{\mathcal{I}k} \boldsymbol{B}^{\mathcal{I}} \right|$.

- Take into account the truncation error by adding it to the radii of the computed interval WCPG $\left\langle \left\langle \widehat{\mathcal{H}}_N^{\mathcal{I}} \right\rangle \right\rangle$
- Naive powering a dense matrix $\boldsymbol{A}^{\mathcal{I}k}$ to large k yeilds wide intervals
  $\Longrightarrow$ diagonalize the interval matrix using eigendecomposition $\boldsymbol{\Lambda}^{\mathcal{I}}, \boldsymbol{V}^{\mathcal{I}}$
  $\Longrightarrow$ now we compute $\sum_{k=0}^{N} \left| \boldsymbol{C}^{\mathcal{I}} \boldsymbol{V}^{\mathcal{I}} \boldsymbol{\Lambda}^{\mathcal{I}k} \boldsymbol{V}^{\mathcal{I}-1} \boldsymbol{B}^{\mathcal{I}} \right|$
- Adjust precision for each interval matrix multiplication, addition and absolute value computation s.t. zero coefficients radii yeild

$$\left| rad \left(\langle\langle \mathcal{H} \rangle\rangle - \left\langle \left\langle \widehat{\mathcal{H}}_N^{\mathcal{I}} \right\rangle \right\rangle \right) \right| < \varepsilon$$

Numerical Example

Random stable filter with 1 input, 1 output, 10 states.
Interval coefficients obtained via quantization to 16 bits (round up).
$\varepsilon = 2^{-64}$

- Spectral radius: $[0.983967 \pm 4.25e-14]$
- Truncation order: $N = 4847$

Approach	*mid*	*rad*
WCPG original system	91.535729	2^{-64}
WCPG quantized system	91.535743	2^{-64}
Naive iWCPG	91.535730	$4.750624 \times 10^{1183}$
iWCPG quantized system	91.535729	1.568769
iWCPG zero radii	91.535729	5.568769×10^{-22}

✓ Inclusion property ensured

✓ Zero radii give ε-neighbourhood of the exact WCPG

Conclusion and Perspectives

Conclusion

- Applied traditional techniques for the eigendecomposition of an interval matrix combined with multiple precision interval arithmetic.
- Ensured the enclosure property
- Ensure that with tightening the coefficients' intervals the computed result converges to the ε-neighbourhood of the exact one

Perspectives

- Integrate our approach into the automatic filter generator to take into account the quantization of coefficients.
- Adapt the filter quality measures for the interval case (require interval discrete Lyapunov equations solver)

Thank you!
Questions?