Roundoff Noise Analysis of Finite Wordlength Realizations with the Implicit State-Space Framework

T. Hilaire, D. Ménard and O. Sentieys

IRISA, R2D2 Team
Lannion, France

EUSIPCO'07 - September 3-7, 2007, Poznań, Poland

Context

- Implementation of Linear Time Invariant controllers/filters
- Finite Word Length context (fixed-point)

Motivation

- Evaluate the roundoff noise errors in the implementation
- Compare various realizations and find an *optimal* one

Context

- Implementation of Linear Time Invariant controllers/filters
- Finite Word Length context (fixed-point)

Motivation

- Evaluate the roundoff noise errors in the implementation
- Compare various realizations and find an *optimal* one

Outline

Outline

The need of a unifying framework

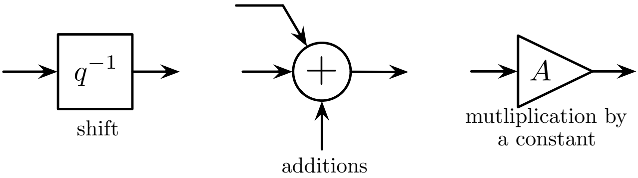
Various implementation forms have to be taken into consideration:

- shift-realizations
- δ-realizations
- observer-state-feedback
- direct form I or II
- cascade or parallel realizations
- etc...

The need of a unifying framework

So, we consider all realizations where the outputs are computed from the inputs with operations like:

- multiplications by a constant
- additions
- shifts (value stored and used at the next step)



The need of a unifying framework

In order to encompass all these implementations, we have proposed a unifying framework to algebraically represent them:

Interests

- macroscopic description of a FWL implementation
- more general than previous realizations (state-space,...)
- more realistic with regard to the parameterization
- directly linked to the in-line computations to be performed

We can describe all possible linear graphs (with additions, multiplications and shift operators) and characterize each computational steps.

The need of a unifying framework

In order to encompass all these implementations, we have proposed a unifying framework to algebraically represent them:

Interests

- macroscopic description of a FWL implementation
- more general than previous realizations (state-space,...)
- more realistic with regard to the parameterization
- directly linked to the in-line computations to be performed

We can describe all possible linear graphs (with additions, multiplications and shift operators) and characterize each computational steps.

The need of a unifying framework

In order to encompass all these implementations, we have proposed a unifying framework to algebraically represent them:

Interests

- macroscopic description of a FWL implementation
- more general than previous realizations (state-space,...)
- more realistic with regard to the parameterization
- directly linked to the in-line computations to be performed

We can describe all possible linear graphs (with additions, multiplications and shift operators) and characterize each computational steps.

The need of a unifying framework

In order to encompass all these implementations, we have proposed a unifying framework to algebraically represent them:

Interests

- macroscopic description of a FWL implementation
- more general than previous realizations (state-space,...)
- more realistic with regard to the parameterization
- directly linked to the in-line computations to be performed

We can describe all possible linear graphs (with additions, multiplications and shift operators) and characterize each computational steps.

The need of a unifying framework

In order to encompass all these implementations, we have proposed a unifying framework to algebraically represent them:

Interests

- macroscopic description of a FWL implementation
- more general than previous realizations (state-space,...)
- more realistic with regard to the parameterization
- directly linked to the in-line computations to be performed

We can describe all possible linear graphs (with additions, multiplications and shift operators) and characterize each computational steps.

The need of a unifying framework

In order to encompass all these implementations, we have proposed a unifying framework to algebraically represent them:

Interests

- macroscopic description of a FWL implementation
- more general than previous realizations (state-space,...)
- more realistic with regard to the parameterization
- directly linked to the in-line computations to be performed

We can describe all possible linear graphs (with additions, multiplications and shift operators) and characterize each computational steps.

The need of a unifying framework

In order to encompass all these implementations, we have proposed a unifying framework to algebraically represent them:

Interests

- macroscopic description of a FWL implementation
- more general than previous realizations (state-space,...)
- more realistic with regard to the parameterization
- directly linked to the in-line computations to be performed

We can describe all possible linear graphs (with additions, multiplications and shift operators) and characterize each computational steps.

Implicit State-Space Framework

All the possible graphs are described by

1. $J.T_{k+1} = M.X_k + N.U_k$
2. $X_{k+1} = K.T_{k+1} + P.X_k + Q.U_k$
3. $Y_k = L.T_{k+1} + R.X_k + S.U_k$

Intermediate variables computation

Implicit State-Space Framework

$$\begin{pmatrix} J & 0 & 0 \\ -K & I & 0 \\ -L & 0 & I \end{pmatrix} \begin{pmatrix} T_{k+1} \\ X_{k+1} \\ Y_k \end{pmatrix} = \begin{pmatrix} 0 & M & N \\ 0 & P & Q \\ 0 & R & S \end{pmatrix} \begin{pmatrix} T_k \\ X_k \\ U_k \end{pmatrix}$$

Implicit State-Space Framework

All the possible graphs are described by

1. $J.T_{k+1} = M.X_k + N.U_k$
2. $X_{k+1} = K.T_{k+1} + P.X_k + Q.U_k$
3. $Y_k = L.T_{k+1} + R.X_k + S.U_k$

State-vector computation

Implicit State-Space Framework

$$\begin{pmatrix} J & 0 & 0 \\ -K & I & 0 \\ -L & 0 & I \end{pmatrix} \begin{pmatrix} T_{k+1} \\ X_{k+1} \\ Y_k \end{pmatrix} = \begin{pmatrix} 0 & M & N \\ 0 & P & Q \\ 0 & R & S \end{pmatrix} \begin{pmatrix} T_k \\ X_k \\ U_k \end{pmatrix}$$

Implicit State-Space Framework

All the possible graphs are described by

1. $J.T_{k+1} = M.X_k + N.U_k$
2. $X_{k+1} = K.T_{k+1} + P.X_k + Q.U_k$
3. $Y_k = L.T_{k+1} + R.X_k + S.U_k$

Output computation

Implicit State-Space Framework

$$\begin{pmatrix} J & 0 & 0 \\ -K & I & 0 \\ -L & 0 & I \end{pmatrix} \begin{pmatrix} T_{k+1} \\ X_{k+1} \\ Y_k \end{pmatrix} = \begin{pmatrix} 0 & M & N \\ 0 & P & Q \\ 0 & R & S \end{pmatrix} \begin{pmatrix} T_k \\ X_k \\ U_k \end{pmatrix}$$

Implicit State-Space Framework

All the possible graphs are described by

1. $J.T_{k+1} = M.X_k + N.U_k$
2. $X_{k+1} = K.T_{k+1} + P.X_k + Q.U_k$
3. $Y_k = L.T_{k+1} + R.X_k + S.U_k$

Implicit State-Space Framework

$$\begin{pmatrix} J & 0 & 0 \\ -K & I & 0 \\ -L & 0 & I \end{pmatrix} \begin{pmatrix} T_{k+1} \\ X_{k+1} \\ Y_k \end{pmatrix} = \begin{pmatrix} 0 & M & N \\ 0 & P & Q \\ 0 & R & S \end{pmatrix} \begin{pmatrix} T_k \\ X_k \\ U_k \end{pmatrix}$$

Intermediate variables

The intermediate variables introduced allow to

- make explicit all the computations done
- show the order of the computations
- express a larger parameterization

Example

A realization with the δ-operator is described by :

$$\begin{cases} \delta X_k &= A_\delta X_k + B_\delta U_k \\ Y_k &= C_\delta X_k + D_\delta U_k \end{cases} \qquad \delta \triangleq \frac{q-1}{\Delta}$$

It is computed with

$$\begin{cases} T &= A_\delta X_k + B_\delta U_k \\ X_{k+1} &= X_k + \Delta T \\ Y_k &= C_\delta X_k + D_\delta U_k \end{cases}$$

and it corresponds to the following implicit state-space :

$$\begin{pmatrix} I & 0 & 0 \\ -\Delta I & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} T_{k+1} \\ X_{k+1} \\ Y_k \end{pmatrix} = \begin{pmatrix} 0 & A_\delta & B_\delta \\ 0 & I & 0 \\ 0 & C_\delta & D_\delta \end{pmatrix} \begin{pmatrix} T_k \\ X_k \\ U_k \end{pmatrix}$$

Example

A realization with the δ-operator is described by :

$$\begin{cases} \delta X_k &= A_\delta X_k + B_\delta U_k \\ Y_k &= C_\delta X_k + D_\delta U_k \end{cases} \qquad \delta \triangleq \frac{q-1}{\Delta}$$

It is computed with

$$\begin{cases} T &= A_\delta X_k + B_\delta U_k \\ X_{k+1} &= X_k + \Delta T \\ Y_k &= C_\delta X_k + D_\delta U_k \end{cases}$$

and it corresponds to the following implicit state-space :

$$\begin{pmatrix} I & 0 & 0 \\ -\Delta I & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} T_{k+1} \\ X_{k+1} \\ Y_k \end{pmatrix} = \begin{pmatrix} 0 & A_\delta & B_\delta \\ 0 & I & 0 \\ 0 & C_\delta & D_\delta \end{pmatrix} \begin{pmatrix} T_k \\ X_k \\ U_k \end{pmatrix}$$

Example

A realization with the δ-operator is described by :

$$\begin{cases} \delta X_k &= A_\delta X_k + B_\delta U_k \\ Y_k &= C_\delta X_k + D_\delta U_k \end{cases} \qquad \delta \triangleq \frac{q-1}{\Delta}$$

It is computed with

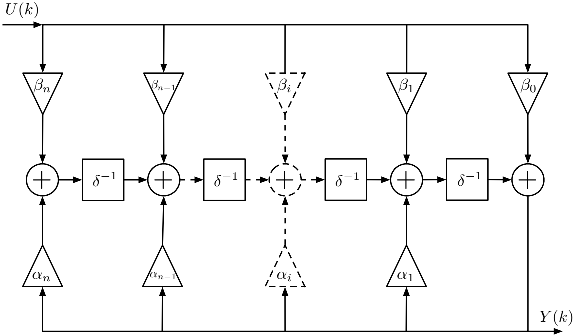
$$\begin{cases} T &= A_\delta X_k + B_\delta U_k \\ X_{k+1} &= X_k + \Delta T \\ Y_k &= C_\delta X_k + D_\delta U_k \end{cases}$$

and it corresponds to the following implicit state-space :

$$\begin{pmatrix} I & 0 & 0 \\ -\Delta I & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} T_{k+1} \\ X_{k+1} \\ Y_k \end{pmatrix} = \begin{pmatrix} 0 & A_\delta & B_\delta \\ 0 & I & 0 \\ 0 & C_\delta & D_\delta \end{pmatrix} \begin{pmatrix} T_k \\ X_k \\ U_k \end{pmatrix}$$

Example

One can find the Direct Form II transposed with δ-operator



with

$$A_\delta = \begin{pmatrix} -\alpha_n & 1 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -\alpha_1 & 0 & \dots & 0 & 1 \\ -\alpha_0 & 0 & \dots & \dots & 0 \end{pmatrix} \quad B_\delta = \begin{pmatrix} \beta_n \\ \vdots \\ \vdots \\ \beta_1 \\ \beta_0 \end{pmatrix} \quad C_\delta = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$$

Outline

Preleminaries

Let's consider a MIMO transfer function G defined by

$$G : z \rightarrow C(zI - A)^{-1}B + D$$

and a noise $U(k)$ with moments

$$\mu_U \triangleq E\left\{U(k)\right\}, \quad \Psi_U \triangleq E\left\{U(k)U^\top(k)\right\}, \quad \sigma_U^2 \triangleq E\left\{U^\top(k)U(k)\right\}$$

Filtered noise

Then the filtered noise Y satisfies

$$\mu_Y = G(0)\mu_U, \quad \sigma_Y^2 = tr\left(\Psi_U(D^\top D + B^\top W_o B)\right)$$

where W_o is the observability Grammian of G, solution of the Lyapunov equation $W_o = A^\top W_o A + C^\top C$

Preleminaries

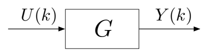
Let's consider a MIMO transfer function G defined by

$$G : z \rightarrow C(zI - A)^{-1}B + D$$

and a noise $U(k)$ with moments

$$\mu_U \triangleq E\left\{U(k)\right\}, \quad \Psi_U \triangleq E\left\{U(k)U^\top(k)\right\}, \quad \sigma_U^2 \triangleq E\left\{U^\top(k)U(k)\right\}$$

Filtered noise



Then the filtered noise Y satisfies

$$\mu_Y = G(0)\mu_U, \quad \sigma_Y^2 = tr\left(\Psi_U(D^\top D + B^\top W_o B)\right)$$

where W_o is the observability Grammian of G, solution of the Lyapunov equation $W_o = A^\top W_o A + C^\top C$

Roundoff Noise Analysis

When implemented, the 3 steps of the computations are modified

$$\begin{aligned} J.T^*(k+1) &\leftarrow M.X^*(k) + N.U(k) + B_T(k) \\ X^*(k+1) &\leftarrow K.T^*(k+1) + P.X^*(k) + Q.U(k) + B_X(k) \\ Y^*(k) &\leftarrow L.T^*(k+1) + R.X^*(k) + S.U(k) + B_Y(k) \end{aligned}$$

The noises depends on

- the way the computations are organized and done
- the fixed-point representation of the inputs, outputs
- the fixed-point representation of the states, intermediate variables
- the fixed-point representation of the constants

Roundoff Noise Analysis

When implemented, the 3 steps of the computations are modified

$$
\begin{aligned}
J.T^*(k+1) &\leftarrow M.X^*(k) + N.U(k) + B_T(k) \\
X^*(k+1) &\leftarrow K.T^*(k+1) + P.X^*(k) + Q.U(k) + B_X(k) \\
Y^*(k) &\leftarrow L.T^*(k+1) + R.X^*(k) + S.U(k) + B_Y(k)
\end{aligned}
$$

The noises depends on

- the way the computations are organized and done
- the fixed-point representation of the inputs, outputs
- the fixed-point representation of the states, intermediate variables
- the fixed-point representation of the constants

Roundoff Noise Analysis

Let B represent all the noises: $B = \begin{pmatrix} B_T \\ B_X \\ B_Y \end{pmatrix}$

Output noise power

The output noise power is given by

$$P = tr\left(\Psi_B\left(M_2^\top M_2 + M_1 W_o M_1^\top\right)\right)$$

where

$$M_1 = \begin{pmatrix} KJ^{-1} & I & 0 \end{pmatrix}, \quad M_2 = \begin{pmatrix} LJ^{-1} & 0 & I \end{pmatrix}$$

Ψ_B depends on the *hardware/software* considerations, whereas M_1 and M_2 depends only on the realization

Roundoff Noise Analysis

Let B represent all the noises: $B = \begin{pmatrix} B_T \\ B_X \\ B_Y \end{pmatrix}$

Output noise power

The output noise power is given by

$$P = tr\left(\Psi_B\left(M_2^\top M_2 + M_1 W_o M_1^\top\right)\right)$$

where

$$M_1 = \begin{pmatrix} KJ^{-1} & I & 0 \end{pmatrix}, \quad M_2 = \begin{pmatrix} LJ^{-1} & 0 & I \end{pmatrix}$$

Ψ_B depends on the *hardware/software* considerations, whereas M_1 and M_2 depends only on the realization

Outline

Roundoff Noise Gain

The RNG is the ouput noise power in a *specific computational scheme*

- the noises appear only after multiplication (*Roundoff After Multiplication*)
- centered white noise
- each noise has the same power σ_0^2

The Roundoff Noise Gain is defined by [Mullis76,Gevers93]

$$G = \frac{P}{\sigma_0^2} \tag{1}$$

Roundoff Noise Gain

Let introduce the matrices d_J to d_S. They are diagonal matrices such

$$(d_X)_{ii} \triangleq \text{number of non-trivial parameters in the i}^{\text{th}} \text{ row of } X$$

where trivial parameters are 0, 1 and -1 because they did not imply a multiplication

The RNG is given by

$$\begin{aligned} G &= tr\left((d_M + d_N + d_J)\, J^{-\top} \left(L^\top L + K^\top W_o K \right) J^{-1} \right) \\ &\quad + tr\left((d_K + d_P + d_Q)\, W_o \right) + tr\left(d_L + d_R + d_S \right) \end{aligned}$$

Outline

Optimal design

It is possible to analytically describe equivalent classes of realization (*Inclusion Principle*)

Equivalent realization

Consider a realization $\mathcal{R}_0$. All realizations $\mathcal{R}_1$ such that

$$\begin{pmatrix} -J_1 & M_1 & N_1 \\ K_1 & P_1 & Q_1 \\ L_1 & R_1 & S_1 \end{pmatrix} = \begin{pmatrix} \mathcal{Y} & & \\ & \mathcal{U}^{-1} & \\ & & I_p \end{pmatrix} \begin{pmatrix} -J_0 & M_0 & N_0 \\ K_0 & P_0 & Q_0 \\ L_0 & R_0 & S_0 \end{pmatrix} \begin{pmatrix} \mathcal{W} & & \\ & \mathcal{U} & \\ & & I_m \end{pmatrix}$$

are equivalent (with $\mathcal{U} \in \mathbb{R}^{n\times n}$, $\mathcal{Y} \in \mathbb{R}^{l\times l}$ and $\mathcal{W} \in \mathbb{R}^{l\times l}$ non-singular matrices).

State-space : $(A, B, C, D) \rightarrow (\mathcal{T}^{-1}A\mathcal{T}, \mathcal{T}^{-1}B, C\mathcal{T}, D)$

Outline

Example

We consider the following low-pass filter

$$H(z) = \frac{0.01594(z+1)^3}{z^3 - 1.9749z^2 + 1.5562z - 0.4538}$$

And the following realizations

Z_1: direct form I with shift-operator,

Z_2: RNG-optimal state-space realization,

Z_3: RNG-optimal implicit state-space realization: we consider all the equivalent realizations described by

$$\begin{cases} EX(k+1) &= AX(k) + BU(k), \\ Y(k) &= CX(k) + DU(k). \end{cases}$$

Z_4: RNG-optimal δ-realization, with $\Delta = 2^{-5}$.

Example

The optimizations are done with Adaptative Simulated Annealing method.

realization	**RNG**	**Nb. operations**
Z_1	$27.53dB$	$6+\ 7\times$
Z_2	$16.40dB$	$12+\ 16\times$
Z_3	$12.05dB$	$15+\ 19\times$
Z_4	$13.35dB$	$15+\ 19\times$

Outline

Conclusions and Perspectives

EUSIPCO'07
—
T. Hilaire,
D. Ménard
and
O. Sentieys

Conclusions

- Implicit State-Space as a Unifying Framework
- Output noise power analysis (RNG scheme)
- optimal design on various forms

Perspectives

- Other structurations to study (q/δ mixed realizations, ρDFIIt...)
- More realistic computational scheme
- Methodology to consider other criteria (L_2-sensitivity, pole-sensitivity,...)
- Toolbox to solve theses problems

Conclusions and Perspectives

Conclusions

- Implicit State-Space as a Unifying Framework
- Output noise power analysis (RNG scheme)
- optimal design on various forms

Perspectives

- Other structurations to study (q/δ mixed realizations, ρDFIIt...)
- More realistic computational scheme
- Methodology to consider other criteria (L_2-sensitivity, pole-sensitivity,...)
- Toolbox to solve theses problems

Questions

Any questions ?

Bibliography

M. Gevers and G. Li.
Parametrizations in Control, Estimation and Filtering Probems.
Springer-Verlag, 1993.

R. Istepanian and J.F. Whidborne, editors.
Digital Controller implementation and fragility.
Springer, 2001.

T. Hilaire, P. Chevrel, and Y. Trinquet
Implicit state-space representation : a unifying framework for FWL implementation of LTI systems
IFAC05 Wolrd Congress, July 2005.