Evaluation analytique de la précision des systèmes en virgule fixe

Romuald ROCHER[1], Daniel MENARD[1], Olivier SENTIEYS[1], Pascal SCALART[1]

[1]ENSSAT/IRISA
Université de Rennes I
6 rue de Kérampont, BP 80518
22305 Lannion Cedex
name@enssat.fr

Résumé – Pour satisfaire les contraintes de coût, les systèmes embarqués utilisent l'arithmétique virgule fixe. Ainsi, les applications définies en virgule flottante doivent être converties en virgule fixe. Cette conversion nécessite de s'assurer que les performances de l'algorithme sont conservées puisque l'arithmétique virgule fixe génère des bruits lors de la modification des formats de données. Ces bruits se propagent dans le système et modifient la précision des calculs. Dans cet article, un modèle d'évaluation de la précision basé sur une approche analytique est présenté pour tous les systèmes composés d'opérations arithmétiques. L'exemple du LMS est développé et la qualité de cet estimateur est vérifiée par des expérimentations.

Abstract – To satisfy cost constraints, application implementation in embedded systems requires fixed-point arithmetic. Thus, applications defined in floating-point arithmetic must be converted into a fixed-point specification. This conversion requires accuracy evaluation to ensure algorithm integrity. Indeed, fixed-point arithmetic generates quantization noises due to bit elimination during a cast operation. These noises are propagated through the system and modify computing accuracy. In this paper, an accuracy evaluation model based on an analytical approach is presented and valid for all systems including arithmetic operations. The LMS algorithm example is developed and its validity is verified through experimentations.

1 Introduction

L'implantation des applications de traitement numérique du signal dans les systèmes embarqués requiert l'utilisation de l'arithmétique virgule fixe pour satisfaire aux contraintes de coût et de consommation. Cependant, ces applications sont spécifiées en virgule flottante pour s'affranchir des problèmes de précision des calculs. Pour réduire le temps de mise sur le marché des applications, des outils de conversion automatique sont nécessaires. Au sein de ces outils, une étape importante concerne l'évaluation de la précision de la spécification en virgule fixe. En effet, l'arithmétique virgule fixe génère des bruits de quantification qui se propagent dans le système et modifient la précision des calculs. Différentes méthodes existent pour évaluer l'influence des bruits de quantification sur la précision des calculs. Tout d'abord, la précision des calculs peut être évaluée au moyen des méthodes basées sur les simulations en virgule fixe [1, 5]. Cependant, ces approches nécessitent une nouvelle simulation dès qu'un format de donnée est modifié. Le temps d'exécution est alors prohibitif. Les autres méthodes existantes sont basées sur des approches analytiques. La précision est exprimée à l'aide d'une relation mathématique. Le temps d'exécution est relativement faible. Cependant, les méthodes existantes ne sont valables que pour des systèmes linéaires et invariants dans le temps (LTI) [3] ou non-LTI et non-récursifs [6] ou alors, font des hypothèses trop restrictives [2]. Ainsi, l'objectif de ce papier est de proposer une méthode analytique évaluant la précision d'un système en virgule fixe composé d'opérations arithmétiques (additions, soustractions, multiplications et divisions). La précision des calculs est évaluée au moyen du Rapport Signal à Bruit de Quantification (RSBQ) et pour toutes les lois de quantification.

L'article est organisé de la manière suivante. Tout d'abord, les bruits de quantification sont introduits. Ensuite, le modèle général est présenté. Celui-ci utilise la réponse impulsionnelle variant dans le temps du système considéré. La puissance du bruit en sortie du système est déduite. Pour valider le modèle, différentes expérimentations sont effectuées. Celles-ci concernent notamment l'application du modèle à l'algorithme LMS et la mesure du temps de calcul de l'expression de la puissance du bruit lors d'un processus de conversion de virgule flottante en virgule fixe. Les résultats obtenus montrent l'intérêt de l'approche par rapport aux méthodes basées sur les simulations.

2 Bruits de quantification

La quantification d'un signal se modélise par une variable aléatoire additive, blanche et uniformément répartie sur son intervalle de définition [7]. La moyenne et la variance de ces bruits dépendent du type de quantification effectué (arrondi ou troncature) et du nombre de bits k éliminés lors du changement de format et du pas de quantification q comme le résume le tableau 1.

Mode de quantification	Troncature	Arrondi classique	Arrondi convergent
Moyenne	$\frac{q}{2}(1-2^{-k})$	$\frac{q}{2}(2^{-k})$	0
Variance	$\frac{q^2}{12}(1-2^{-2k})$	$\frac{q^2}{12}(1-2^{-2k})$	$\frac{q^2}{12}(1+2^{-2k+1})$

TAB. 1 – Moyenne et variance des bruits de quantification.

Ces bruits se propagent ensuite dans le système selon les différentes opérations rencontrées. La propagation de deux bruits scalaires b_x et b_y associés aux deux entrées X et Y de l'opération génèrent un bruit de sortie b_z exprimé comme la somme des deux sources de bruit multipliées par des termes de signaux α comme expliqué dans [6].

$$b_z = \alpha_1 b_x + \alpha_2 b_y \tag{1}$$

Dans le cas de bruits non-scalaires (vectoriels ou matriciels), la multiplication par les termes de signaux peut s'effectuer soit à gauche soit à droite. Ainsi, chaque source de bruit est multipliée par deux termes de signaux A et D.

$$b_z = A_x b_x D_x + A_y b_y D_y \tag{2}$$

3 Modélisation du système

Comme les termes croisés n'apparaissent pas dans l'expression (2), le bruit en sortie du système $b_y(n)$ s'exprime comme la somme des contributions $b'_i(n)$ de chaque source de bruit $b_i(n)$ comme le montre la figure 1.

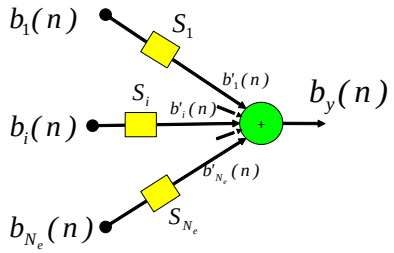

FIG. 1 – Modélisation du système

$$b_y(n) = \sum_{i=1}^{N_e} b'_i(n) \tag{3}$$

où N_e désigne le nombre de sources de bruit. La contribution $b'_i(n)$ dépend de $b_i(n)$ mais aussi de ces échantillons précédents $b_i(n-k)$ pour $k \in [1 : Q_i]$ en raison des retards présents. De plus, la contribution $b'_i(n)$ dépend de ses échantillons précédents $b'_i(n-m)$ pour $m \in [1 : P_i]$ en raison des récursions dans le système aboutissant à l'expression suivante :

$$b'_i(n) = \sum_{k=0}^{Q_i} g_i(n-k) b_i(n-k) + \sum_{m=1}^{P_i} f_i(n-m) b'_i(n-m) \tag{4}$$

où $g_i(n-k)$ représente la contribution de la source de bruit b_i à l'instant $(n-k)$ et $f_i(n-m)$ celle du bruit b'_i à l'instant $(n-m)$. Ces termes f_i et g_i varient dans le temps et dépendent du système. En développant cette expression, la relation entre la contribution $b'_i(n)$ et la source $b_i(n)$ devient

$$b'_i(n) = \sum_{k=0}^{n} h_i(k) b_i(k) \tag{5}$$

où h_i représente la réponse impulsionnelle variant dans le temps du système. Comme la propagation d'une source de bruit se modélise par la multiplication par deux termes de signaux A et D, la réponse impulsionnelle h_i est équivalente à la multiplication par deux termes de signaux.

$$b'_i(n) = \sum_{k=0}^{n} A_i(k) b_i(k) D_i(k) \tag{6}$$

Le bruit de sortie $b_y(n)$ est alors la somme de toutes les contributions $b'_i(n)$

$$b_y(n) = \sum_{i=1}^{N_e} \sum_{k=0}^{n} A_i(k) b_i(k) D_i(k) \tag{7}$$

4 Expression de la puissance du bruit de sortie

La puissance du bruit de sortie Pb est obtenue au moyen du moment d'ordre deux de l'expression (7). La non-corrélation entre les bruits et les signaux et entre les bruits entre eux conduit à l'expression suivante

$$\begin{aligned} P_b &= E[b_y^2(n)] \\ &= \sum_{i=1}^{Ne} \sigma_{b_i}^2 K_i + \sum_{i=1}^{Ne} \sum_{j=1}^{Ne} m_{b_i} m_{bj} L_{ij} \end{aligned} \tag{8}$$

où m_{b_i} et $\sigma_{b_i}^2$ représentent la moyenne et la variance du bruit $b_i(n)$. De plus, K_i et L_{ij} sont des termes de signaux définis par

$$K_i = \sum_{k=0}^{n\to\infty} E\Big[Tr\big(D_i(k) D_i^t(k)\big) Tr\big(A_i(k) A_i^t(k)\big)\Big] \tag{9}$$

$$L_{ij} = \sum_{k=0}^{n\to\infty} \sum_{m=0}^{n\to\infty} E\Big[Tr\big(A_i(k) 1_N D_i(k) D_j^t(m) 1_N A_j^t(m)\big)\Big] \tag{10}$$

avec 1_N désigne la matrice composée de 1. Les expressions de K et L sont des constantes obtenues à partir d'une simulation unique en virgule flottante. Les statistiques des bruits m et σ^2 dépendent des formats virgule fixe et représentent les variables de l'équation (8). Les termes K et L sont représentés par des sommes infinies. En pratique, les sommes sont tronquées après un certain nombre d'échantillons p. Les diverses expérimentations effectuées montrent qu'un nombre p égal à 500 permet d'aboutir à des résultats satisfaisants dans l'ensemble des cas testés.

Pour réduire la complexité de l'approche, une méthode basée sur la prédiction linéaire est introduite. L'expression liant les termes de réponse impulsionnelle d'un système est linéarisée avec des coefficients de prédiction minimisant l'erreur quadratique entre les termes de réponse impulsionnelle et leurs estimés. Cette approche permet de modéliser les sommes infinies et de réduire la complexité du modèle.

5 Expérimentations

5.1 Algorithme LMS

Considérons l'exemple de l'algorithme LMS. Cet algorithme consiste à estimer une séquence de scalaires $y(n)$ à partir de données d'observations $X(n) = [x(n), x(n-1)...x(n-N+1)]^t$ [4]. L'estimée de $y(n)$ est $W^t(n)X(n)$ où $W(n)$ est un vecteur de taille N convergeant vers sa valeur optimale W_{opt} au sens de l'erreur quadratique moyenne selon l'expression suivante :

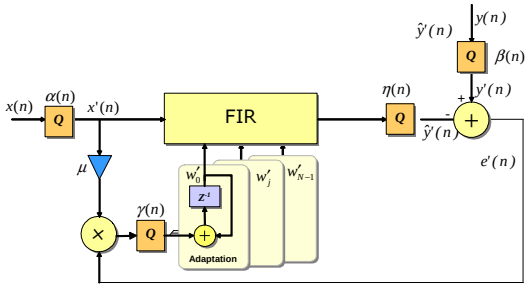

FIG. 2 – Algorithme LMS

$$W(n+1) = W(n) + \mu X(n)(y(n) - W^t(n)X(n)) \quad (11)$$

où μ est une constante positive représentant le pas d'adaptation. Lors d'une implémentation virgule fixe, quatre sources de bruit sont générées. Les bruits $\alpha(n)$ et $\beta(n)$ sont générés par la quantification de $X(n)$ et de $y(n)$. Le terme $\gamma(n)$ provient du produit entre $\mu X(n)$ et l'erreur $e(n) = y(n) - W^t(n)X(n)$. Le bruit $\eta(n)$ est généré par le produit $W^t(n)X(n)$. Le termes m et σ^2 désignent la moyenne et la variance de ces bruits.

Le système parcouru par chaque source est déterminé. Le bruit $\gamma(n)$ est analysé en détails pour illustrer notre approche. La fonction de transfert modélisant sa propagation est la suivante

$$H_\gamma(z) = X^t(n)\frac{z^{-1}}{1 - (I_N - \mu X(n-1)X^t(n-1))\, z^{-1}} \quad (12)$$

où I_N est la matrice identité de taille N. Sa contribution $\gamma'(n)$ est obtenue en utilisant sa réponse impulsionnelle h_γ

$$\gamma'(n) = \sum_{k=0}^{n-1} h_\gamma(k)\gamma(k) = \sum_{k=0}^{n-1} A_\gamma(k)\gamma(k)$$

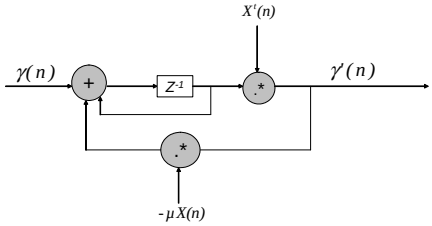

FIG. 3 – Propagation du bruit $\gamma(n)$

$$= \sum_{k=0}^{n-1} X^t(n) \underbrace{\prod_{m=k+1}^{n-1} \Big(I_N - \mu X(m)X^t(m)\Big)}_{F(n,k)} \gamma(k) \quad (13)$$

La réponse impulsionnelle h_γ est définie comme le produit de deux termes de signaux A_γ et D_γ. Ce dernier n'apparaît pas car toutes les multiplications sont effectuées à gauche. Les contributions des autres sources de bruit sont obtenues de la même façon. Comme les bruits $\eta(n)$ et $\beta(n)$ sont scalaires, les termes A le sont également, et dans ce cas $Tr(AA^t) = A^2$ pour ces deux termes. La puissance du bruit de sortie est obtenue en appliquant la relation (8)

$$\begin{aligned} E[b_y^2(n)] &= \sum_{k=0}^{n} \sigma_\alpha^2 E\Big[Tr\big(A_\alpha(k)A_\alpha^t(k)\big)\Big] + \sum_{k=0}^{n} \sigma_\eta^2 E[A_\eta^2(k)] \\ &+ \sum_{k=0}^{n} \sigma_\beta^2 E[A_\beta^2(k)] + \sum_{k=0}^{n} \sigma_\gamma^2 E\Big[Tr\big(A_\gamma(k)A_\gamma^t(k)\big)\Big] \\ &+ \sum_{k=0}^{n}\sum_{l=0}^{n} E\Big[Tr\big(M(k)M^t(l)\big)\Big] \end{aligned} \quad (14)$$

avec

$$\begin{aligned} M(k) &= A_\alpha(k)m_\alpha + A_\beta(k)m_\beta + A_\eta(k)m_\eta + A_\gamma(k)m_\gamma \\ A_\alpha(k) &= \mu X^t(n)F(n,k)\Big(e(k) - X(k)W^t(k)\Big) + W^t(n)\Delta(n-k) \\ A_\beta(k) &= \mu X^t(n)F(n,k)X(k) \\ A_\eta(k) &= -\mu X^t(n)F(n,k)X(k) + \Delta(n-k) \end{aligned} \quad (15)$$

où Δ est le symbol de Kronecker.

5.2 Qualité de l'estimation

Pour évaluer la qualité de notre modèle, différentes expérimentations ont été menées. L'erreur relative entre la puissance du bruit estimée par notre modèle et sa valeur réelle déterminée par une simulation est évaluée. La figure 4 montre l'erreur relative commise par notre modèle sur un algorithme LMS de taille 32. Les résultats sont présentés en fonction du nombre p choisi pour déterminer les sommes infinies et de la corrélation du signal d'entrée $x(n)$. Cette dernière est déterminée par le coefficient de corrélation δ. Le signal peut être blanc ($\delta = 0$), moyennement corrélé ($\delta = 0.5$) ou très corrélé ($\delta = 0.95$). Si p augmente, l'erreur relative diminue. En effet, l'augmentation de p conduit à un nombre de termes plus important

dans le calcul des sommes infinies donc plus représentatif. De plus, la convergence de l'erreur relative dépend de la corrélation des données d'entrée. Pour des données non-corrélées, la convergence de l'erreur relative est plus lente que pour des données très corrélées. Pour obtenir une erreur inférieure à 20%, 300 points sont nécessaires pour des données très corrélés contre 550 points pour des données non-corrélées.

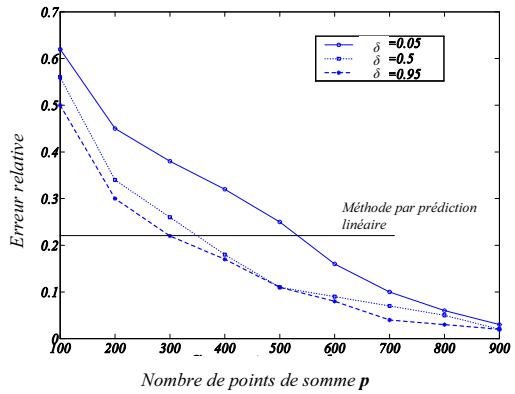

FIG. 4 – Erreur relative commise sur le LMS

Ainsi, le nombre de points p utilisé pour le calcul des sommes infinies dépend de la corrélation des données d'entrée. Néanmoins, un nombre p égal à 500 assure d'obtenir une erreur relative inférieure à 25% dans tous les cas. Ceci représente un écart de 1 dB entre la puissance du bruit obtenue par notre modèle et la puissance réelle. La méthode par prédiction linéaire aboutit à une erreur relative de 21%. Des résultats du même ordre sont obtenus pour d'autres tailles de l'algorithme LMS.

5.3 Temps d'optimisation

Le modèle a été comparé en terme de temps d'exécution aux méthodes basées sur des simulations lors d'un processus d'optimisation en virgule fixe. Les expérimentations ont été effectuées sur Matlab et les résultats sont présentés sur la figure 5. Dans un premier temps, notre approche analytique consiste à déterminer l'expression de la puissance du bruit. Cette étape initiale requiert 46 secondes pour la méthode basée sur les sommes infinies et 4 secondes pour le modèle de prédiction linéaire. Ensuite, chaque itération du processus correspond à l'évaluation de l'expression pour les formats virgule fixe définis. Cette seconde étape nécessite un temps de calcul négligeable. Pour l'algorithme LMS, notre méthode permet d'obtenir des gains de temps après moins de 100 iterations équivalant à un temps d'exécution de 46 secondes. Pour un processus d'optimisation avec une trentaine de variables, entre 10000 et 100000 itérations sont nécessaires. Avec le modèle utilisant la prédiction linéaire, des gains de temps sont obtenus après seulement 10 itérations par rapport aux méthodes basées sur des simulations. Ce résultat nécessite un temps d'exécution de seulement 4 secondes. L'intérêt de notre modèle est ainsi démontré pour réduire le temps de développement des systèmes en virgule fixe.

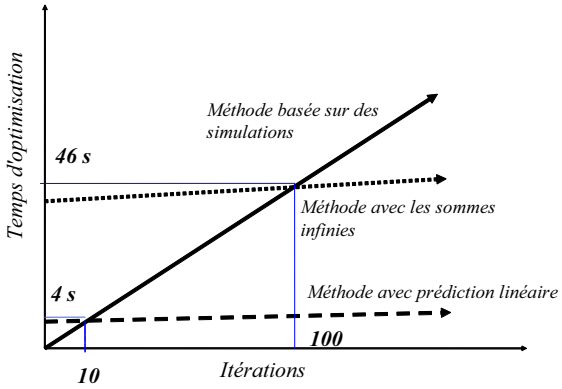

FIG. 5 – Temps d'optimisation pour notre approche et les méthodes basées sur des simulations

6 Conclusion

Dans cet article, un modèle pour déterminer analytiquement la précision d'un système en virgule fixe est présenté. Le modèle est développé pour tous les systèmes composés d'opérations arithmétiques. Celui-ci aboutit à une expression de la puissance du bruit de sortie basée sur des sommes infinies. Pour réduire la complexité de l'approche, un modèle utilisant la prédiction linéaire a été introduit. La méthode a été appliquée à l'algorithme LMS pour illustrer le modèle et vérifier sa validité. De plus, le temps de calcul a été mesuré. Cette approche permet de réduire le temps de conversion de virgule flottante en virgule fixe.

Références

[1] P. Belanovic and M. Rupp, "Automated Floating-point to Fixed-point Conversion with the fixify Environment," *IEEE Rapid System Prototyping*, pp. 172-178, 2005.

[2] J.M. Cheneaux and L.S. Didier and F. Rico, "The Fixed CADNA Library," *Real Number and Computers*, Sep. 2003.

[3] G.A. Constantinides and P.Y.K. Cheung and W.Luk "Synthesis and Optimization of DSP Algorithms," *Kluwer Academic Publishers*, 2004.

[4] S. Haykin, "Adaptive Filter Theory, " *Englewood Cliffs, NJ :Prentice-Hall*, 2^{nd} edition, 1991.

[5] S. Kim and K. Kum and S. Wonyong, "Fixed-Point Optimization Utility for C and C++ Based Digital Signal Processing Programs," *IEEE Transactions on Circuits and Systems II*, vol. 45, no. 11, pp. 1453-1464, Nov. 1998.

[6] D. Menard and R. Rocher and P. Scalart and O. Sentieys, "Automatic SQNR determination in non-linear and non-recursive fixed-point systems," *XII European Signal Processing Conference* , pp. 1349-1352, Sep. 2004.

[7] B. Widrow and I. Kollár and M.-C. Liu, "Statistical Theory of Quantization," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 2, pp. 353-361, Apr. 1996.