A Stochastic Approach For The Range Evaluation

Andrei Banciu

N^o d'ordre :2012REN1E002 ANNÉE 2012

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du signal, télécommunications

Ecole doctorale : Matisse

présentée par

Andrei Banciu

préparée à l'unité de recherche : IRISA-Cairn Lannion

A Stochastic Approach For The Range Evaluation

Thèse soutenue à Lannion le 29.02.2012

devant le jury composé de :

Emmanuel BOUTILLON
Professeur des Universités
Université de Bretagne Sud / rapporteur

Laurent-Stéphane DIDIER
Maître de Conférences, HDR
Université Pierre et Marie Curie / rapporteur

Jean-François NEZAN
Professeur des Universités
INSA Rennes / examinateur

Thierry MICHEL
STMicroelectronics / examinateur

Emmanuel CASSEAU
Professeur des Universités
Université de Rennes1 - ENSSAT / directeur de thèse

Daniel MENARD
Maître de Confrences
Université de Rennes1 - ENSSAT / co-directeur de thèse

To my wonderful wife Eldda

Abstract

Digital Signal Processing algorithms are implemented in VLSI systems using fixed-point arithmetic in order to obtain cost-effective hardware with optimized throughput and power consumption. However, the finite wordlength of the fixed-point representation introduces a quantization error that generates a degradation of the computational precision. The fixed-point implementation has to guarantee the performance constraints required by the application while minimizing the cost. The manual conversion of the floating-point algortihm using fixed-point data is error prone and time consuming and continues to be one of the most important steps of the design. An automatic source code tranformation from the floating-point representation to a fixed-point implemenation can significantly reduce the developement time. The goal of this thesis is to provide a method for the static analysis of the fixed-point performance that can be integrated in an automatic floating-point to fixed-point transformation tool. Our aim is to obtain an analytical representation that characterizes the variability of the signal through the datapath that avoids any type of simulation and manual code instrumentation.

At the beggining, a probabilistic approach for the dynamic range estimation is developped. Some applications can accept occasional overflows if their probability of occurrence is small enough. In this case, the integer part wordlength of the fixed-point variables is optimized in compliance with their statistical description, based on the overflow probability criteria. A real test case from the field of digital communications is analyzed as a validation procedure. The orthogonal frequency-division multiplexing (OFDM) is a transmission technique characterized by a high peak-to-average power ratio (PAPR). As a consequence, choosing the proper wordlength for the fixed-point data types is a difficult task. To avoid overdimensioning the implementation, a trade-off between the dynamic range that is covered by the fixed-point representation and the cost of the implementation has to be made.

The rest of the work is separated in two main parts. First, the case of linear-time invariant systems is adressed. The Karhunen-Loevè Expansion (KLE) is used as a discretization procedure to represent the variability of the input signal. The KLE representation of the output is further determined using the impulse response of the system. The dynamic range is computed from the probability density function (PDF) with respect to a coverage probability. The same KLE discretization approach is applied to evaluate the quantization noise, extending the method to the numerical accuracy analysis.

The Polynomial Chaos Expansion (PCE) is introduced to treat the non-linear operations. It is a mathematical representation that captures the entire probabilistic description of a variable. As a first step, the random behaviour of the input is represented in the form of a PCE representation. The variability of the input is statically propagated through the data-flow graph (DFG) of the application and the analytical representation of the output is obtained. As opposed to the KLE, this method can be applied to any type of system that is composed of arithmetic operations making it possible to treat non-linear systems. Using the same probabilistic methodology that has been introduced, the dynamic range is computed in a similar manner to the KLE method.

The probabilistic approach for the range determination is evaluated for several typical applications. The results show that the PDF of the signal and the probability of overflow estimated using our method follow to a great degree of accuracy the ones obtained using Monte Carlo simulation. Furthermore, a comparison with tradional methods for range estimation shows that the interval of variation can be significantly reduced using our method so that the datapath of the fixed-point application can be optimized and the cost of the implementation reduced.

Resumé

Les applications de traitement du signal ont connu un très fort développement dans les dernières décennies, bénéficiant des avancées majeures de l'industrie des semi-conducteurs. Aujourd'hui, elles sont présentes dans une grande variété de domaines d'activité, tels que les télécommunications, le multimédia, l'électronique grand public, le transport, la médecine, les applications militaires, etc. Les améliorations technologiques continues ont conduit à l'apparition de nouveaux produits qui utilisent des algorithmes complexes de traitement du signal afin de répondre aux exigences de l'application. Pour améliorer la productivité et pour satisfaire les contraintes de temps de commercialisation, des nombreux outils de haut niveau ont été mis au point pour toutes les étapes de la conception. Ils permettent le passage d'une description de haut niveau de l'application à une description de bas niveau avec une exploration rapide des solutions disponibles pour l‘implémentation.

La spécification de l'application détermine les critères de performance qui doivent être garantis par le système. Un algorithme approprié est mis au point pour répondre à ces besoins. Dans un premier temps, une description de haut niveau de l'algorithme est spécifiée en utilisant une précision importante pour surpasser les problèmes liés à la précision du calcul. Il s'agit d'un processus qui valide la fiabilité de l'algorithme pour le problème donné. Même si l'erreur inhérente à la précision de calcul existe encore, l'arithmétique en virgule flottante garantit une précision et une plage de dynamique suffisantes dans la plupart des cas. Des environnements de calcul numérique comme Matlab, Mathematica ou Scilab sont utilisés pour simuler cette description de haut niveau.

Toutes les implémentations pratiques utilisent l‘arithmétique en virgule fixe afin de réduire la surface et la consommation d'énergie. En conséquence, une conversion de la description en virgule flottante de l'algorithme en une version implémentable en virgule fixe, ajustant la largeur du chemin de données, doit être réalisée. C'est un processus d'optimisation qui consiste à trouver les parties fractionnaire (évaluation de la précision numérique) et entière (estimation de la dynamique) minimales qui satisfassent les contraintes de performance.

L'apparition d'outils de synthèse de haut niveau qui génèrent des implémentations RTL directement à partir d'une spécification C/C++ qui utilise l'arithmétique en virgule fixe permet de réduire le temps de développement tout en permettant une bonne exploration de l'espace de conception. Toutefois, l'étape de conversion entre la description en virgule flottante de l'algorithme et celle en virgule fixe doit être

faite à la main et continue d'être l'une des parties les plus difficiles et fastidieuse de la conception des circuits intégrés numériques. C'est un problème qui demande beaucoup de temps et qui est sujet aux erreurs. Trouver un bon compromis entre le coût de l'implémentation et la précision des calculs qui doit être respectée est une tâche très difficile. Il a été montré que cela peut prendre jusqu'à 30 % de la durée totale du développement.

Un outil de conversion automatique virgule flottante - virgule fixe qui permet d'optimiser la surface de l'implémentation et le débit sous une contrainte de performance est obligatoire afin de réduire l'écart entre la description en virgule flottante des algorithmes et l'implémentation matérielle, de maîtriser la complexité et de réduire le temps de développement. Il s'agit d'une transformation de code source qui peut être facilement intégrée dans le flot de conception des circuits numériques. L'application d'entrée est décrite comme une implémentation C/C + + qui utilise la représentation en virgule flottante pour toutes les variables. Séparément, les contraintes de performance (précision de calcul) qui devraient être satisfaites par l'implémentation en virgule fixe sont fournies par l'utilisateur. Le résultat est obtenu par la génération d'une implémentation qui utilise l'arithmétique en virgule fixe et qui permet de régler tous les tailles des variables du chemin de données.

En pratique, la taille de chaque variable représentée dans un format virgule fixe est limitée. Cela produit une dégradation de la précision mathématique du résultat obtenu. La précision du résultat est donnée par le nombre de bits utilisés pour sa représentation. L'augmentation de la longueur des mots du chemin de donnée, améliore la précision mais introduit un coût matériel supplémentaire. Un bruit de quantification est introduit chaque fois que des bits sont éliminés par des opérations de quantification (arrondi ou troncature). De plus, cela provoque l'apparition de débordements chaque fois que la longueur de la partie entière est insuffisante pour représenter la variation de la dynamique.

La conversion virgule flottante - virgule fixe devient un processus d'optimisation qui minimise le coût de l'implémentation pour une dégradation de performances acceptable. En d'autres termes il faut trouver les tailles minimales pour la partie entière et la partie fractionnaire de la représentation virgule fixe de chaque variable qui continuent de satisfaire la précision de calcul globale, requise par l'application (en général le rapport signal-à-bruit (SNR) ou le taux d'erreur binaire (BER) du système). Ainsi, le problème de la transformation peut être séparé en deux parties différentes qui sont réalisées de faon indépendante. La longueur de la partie fractionnaire donne la précision du calcul tandis que la longueur de la partie entière détermine la variation de la dynamique maximale qui est autorisé par cette représentation :

- l'Analyse de la précision numérique : optimisation de la partie fractionnaire de la représentation

- l'Estimation de dynamique : optimisation de la partie entière de la représentation

L'analyse de la précision numérique est liée à la notion de bruit de quantification. Elle étudie la sensibilité de la sortie par rapport aux changements légers des valeurs de l'entrée, traduite en une métrique d'erreur. En fait, des nombreuses méthodes de conversion sont axées seulement sur l'optimisation de la longueur de la partie fractionnaire en utilisant la puissance du bruit de quantification comme critère de performance. La taille minimale de chaque mot se trouve en réalisant un compromis entre la précision nécessaire et le coût du circuit.

L'estimation de dynamique calcule le nombre minimal de bits nécessaires pour la partie entière d'une variable en fonction de ses valeurs maximale et minimale. Les méthodes classiques de calcul de l'estimation se basent sur des limites théoriques absolues (qui ne seront jamais dépassées dans la pratique) pour éviter l'apparition de débordements. En faisant ainsi, on obtient des intervalles de variation qui sont très pessimistes et le coût de l'implémentation est largement augmenté. Comme l'absence de débordements est garantie, l'optimisation de la longueur de mot de la partie entière sous des contraintes de performance devient impossible et le compromis précision-coût de l'implémentation est considéré uniquement pour la partie fractionnaire.

Cependant, certaines applications peuvent toutefois accepter des débordements occasionnels, si la probabilité d'occurrence est assez petite pour ne pas trop dégrader les performances globales du circuit. La méthode d'estimation de la dynamique devrait être en mesure de prendre en compte cette information dans le but de réduire les coûts (la surface et la puissance consommée). Traditionnellement, il s'agit d'un processus qui peut être réalisé en utilisant un nombre de simulations important. Toutefois, il s'agit d'un processus itératif, qui doit être fait à chaque fois qu'un paramètre de l'implémentation a changé. Cette méthode devient vite très complexe, en prenant beaucoup de temps et reste une source d'erreurs si les simulations ne sont pas exhaustives.

Les méthodes classiques d'analyse, telles que l'arithmétique d'intervalle et l'arithmétique affine ne fournissent pas d'informations supplémentaires sur la variation du signal à l'intérieur de l'intervalle de valeurs possibles. De ce fait elles restent une mauvaise approximation de l'incertitude réelle des signaux. Les signaux qui ont de grandes variations, mais qui ont de faibles probabilités au niveau de la queue de leur distribution de probabilité ne sont pas bien représentés.

Dans cette thèse, une approche stochastique pour l'évaluation de la dynamique des données est présentée. Le but est de fournir un cadre probabiliste qui permet de réaliser une estimation de la dynamique à l'aide des critères statistiques et qui peut être facilement intégrée dans un outil automatique de transformation virgule flottante en virgule fixe. Nous sommes intéressés par l'optimisation de la longueur de la partie entière des données, lorsqu'une légère dégradation des performances est acceptable. En fait, les débordements sont autorisés si leur probabilité d'apparition est suffisamment faible pour l'application donnée. La dynamique ne couvre plus tout l'intervalle théorique de variation, et des débordements sont autorisés avec une contrainte quant à leur probabilité d'apparition. Les signaux qui ont des variations importantes de leur amplitude sont approximés avec des intervalles serrés pour réduire le coût de l'implémentation.

Au lieu de représenter la variation d'un signal comme les méthodes classiques d'analyse le font, en utilisant uniquement les limites maximales et minimales (x_{min} et x_{max}), notre objectif est d'obtenir une représentation complète de la variabilité qui intègre son comportement probabiliste. L'intervalle de variation des valeur d'une variable est donc représenté par sa fonction de densité de probabilité (FDP).

Nous allons démarrer par la détermination d'une représentation stochastique (qui intègre la FDP) de chaque entrée d'un système. Cette caractérisation de la variabilité est ensuite propagée à travers le système, de façon à obtenir les représentations correspondantes à chaque variable du système.

Ensuite, nous proposons un critère d'optimisation de la taille de la partie entière basé sur la probabilité de débordement. L'intervalle de variation autorisée pour toutes les variables est calculé à partir de leurs FPD pour correspondre à une probabilité de débordement souhaitée. De cette façon, nous allons fournir plus d'informations sur la variation des signaux que des simples limites maximales et minimales. En effet, une approche qui capte toute la distribution et la corrélation entre les données peut considérablement améliorer les résultats par rapport aux approches classiques comme l'arithmétique d'intervalle et l'arithmétique affine.

Un exemple réel, constitué d'un émetteur OFDM (Orthogonal frequency division multiplexing) est utilisé comme test pour motiver et ensuite valider notre approche probabiliste. Il a été choisi parce que c'est un exemple typique d'application qui met en avant le problème d'un facteur de crête très important (appelé souvent le PAPR (Peak-to-Average Power Ratio)). Il est défini comme le rapport entre l'amplitude du pic du signal et sa valeur moyenne. Lorsque les signaux ont une grande variation de leur amplitude tout au long de l'exécution, le dimensionnement de la longueur des mots du chemin de données devient une tâche extrêmement difficile. Si tout l'intervalle de variation théorique est assuré, le coût de l'implémentation matérielle peut être augmenté significativement. Pour se conformer aux exigences de haut débit nécessaires pour l'application et en même temps obtenir un coût raisonnable, la longueur de la partie entière de la représentation virgule fixe doit être réduite sans couvrir tout l'intervalle de variation possible même si cela va introduire des débordements occasionnels.

La conception du modulateur OFDM a été réalisée en utilisant l'outil de synthèse de haut niveau de Mentor Graphics CatapultC. Il permet d'obtenir rapidement des implémentations matérielles avec des tailles de données différentes pour le chemin de données. Cela nous a permis d'analyser les effets des débordements sur le taux d'erreur binaire de l'application. Nous avons analysé aussi le gain obtenu en termes de surface et de puissance consommée par le circuit en diminuant la taille de la partie entière. Cela se traduit par l'apparition des débordements et donc une réduction des performances du circuit.

Nous avons conclu à partir de cette partie pratique qu'il est possible de diminuer largement les coûts de l'implémentation en virgule fixe et en même temps augmenter le débit obtenu en optimisant la taille de la partie entière de la représentation. L'apparition des débordements peut être tolérée si les limites maximales et minimales de l'intervalle de variation autorisées sont choisis pour satisfaire une probabilité de débordement qui convient pour l'application globale (c'est-à-dire que le taux d'erreur binaire va être conforme au standard de communication).

Dans un premier temps, une méthode pour l'évaluation de l'intervalle de variation par rapport à une probabilité de débordement correspondante est présentée pour les systèmes linéaires et invariants dans le temps (LTI). C'est le cas de nombreux systèmes de traitement du signal et notamment de l'émetteur OFDM que nous avons considéré. La méthode est basée sur le développement de Karhunen-Loève pour la représentation de la variabilité des signaux.

Dans les applications de traitement numériques du signal, souvent les signaux d'entrée ont une correspondance à de processus physiques réels qui varient dans le temps. La structure de corrélation du signal d'entrée va ainsi modifier la description statistique des variables internes et des sorties et la forme de leurs fonctions de densité de probabilité va être fortement modifiée. En conséquence, la dimension temporelle doit être prise en compte afin de fournir des résultats fiables dans la pratique.

La notion de processus aléatoire devient le modèle mathématique qui est le plus approprié pour représenter la variabilité inhérente de l'entrée. Le signal d'entrée en virgule flottante est modélisé comme un processus aléatoire discrèt $x(t, \theta)$, appliqué sur un intervalle de temps $[0, T]$ (c'est à dire une séquence de variables aléatoires). Cela signifie que, à chaque instant de temps $t_0 = 1, 2, 3, ..., n$, la valeur du signal $x(t_0, \theta)$ est représentée par une variable aléatoire. θ désigne le résultat de la variable aléatoire dans l'espace aléatoire et sera omis à partir de maintenant pour la clarté. Le caractère aléatoire de l'entrée se propage dans tous le système et les variables d'état et les sorties deviennent aussi des processus aléatoires.

En général, les processus aléatoires ont une dimension infinie. Afin de les représenter dans la pratique, une procédure de discrétisation doit être réalisée. Le but étant de les représenter par une combinaison d'un nombre fini de variables aléatoires qui est plus facile à gérer en pratique. Plusieurs techniques de discrétisation ont été présentées dans la littérature. Parmis eux, les développements en série sont les plus utilisées.

Dans notre approche, le développement de Karhunen-Loève (KLE) est utilisé comme moyen de discrétisation pour les signaux d'entrée du système. La KLE permet de représenter un processus aléatoire par une combinaison linéaire des fonctions déterministes avec des coefficients aléatoires orthogonaux (non corrélés) (coefficients qui représentent le contenu probabiliste, ou la dimension stochastique).

Nous avons choisi cette méthode de discrétisation pour l'estimation de la dynamique parce qu'elle permet de minimiser l'erreur quadratique moyenne. En fait, la KLE est une série convergente pour tous les processus aléatoires de second ordre (processus avec l'énergie finie) et qui minimise l'erreur de troncature. En d'autres termes, cela veut dire qu'il n'y a pas d'autres développements en séries qui se rapproche mieux du processus aléatoire avec le même nombre de termes que le développement KLE.

Puisque nous nous intéressons seulement aux systèmes LTI dans ce chapitre, il est possible d'utiliser la propriété de superposition pour propager la variabilité des entrées (décrite avec des KLEs) dans tout le système. Par opposition à la méthode basée sur la simulation qui a été déjà présentée, nous montrons ici comment la variabilité peut être propagée statiquement à travers les systèmes LTI en utilisant la réponse impulsionnelle. Il devient donc possible de déterminer la représentation KLE de chaque variable du système sans aucune simulation.

Comme décrit dans la partie pratique, nous utilisons une approche stochastique pour l'estimation de la dynamique. L'intervalle de variation est donc calculé à partir de la FDP par rapport à une probabilité de débordement souhaitée. Pour cela nous proposons plusieurs méthodes pour l'estimation de la FDP de chaque variable à partir de la KLE et notament la méthode kernell density estimation (KDE).

Les résultats pour plusieurs exemples pratiques sont présentés ensuite. Le cas d'un filtre FIR, un filtre IIR et une IFFT 512 points sont traités. La précision de la méthode est comparée tout d'abord par rapport à la simulation pour prouver que les résultats sont conformes à la pratique. Ensuite nous comparons notre méthode avec des méthodes d'estimation de la dynamique traditionnelles comme l'arithmétique d'intervalle et nous montrons qu'en utilisant notre approche, le coût de l'implémentation peut être largement diminué. Cela montre l'intérêt d'utiliser une méthode stochastique pour l'estimation de la dynamique.

Comme un objectif secondaire, le problème de l'estimation de la précision des calcules est adressé. Dans le cas des opérateurs de décision, les approches traditionnelles de l'analyse de la précision numérique qui calculent la puissance du bruit de quantification ont prouvé leurs limites et toute la FDP du bruit de quantification doit être déterminée. Afin de résoudre le problème, nous montrons comment il est possible d'adapter l'approche stochastique pour évaluer le bruit de quantification.

En utilisant la même méthode de discrétisation (KLE) pour le bruit de quantification, la méthodologie peut être modifiée pour obtenir la FDP de la sortie d'un système LTI. Il devient donc possible d'évaluer le bruit de quantification directement. Le SNR est estimé à partir de la variance du bruit de quantification. Si besoin, la FDP complète du bruit peut être calculée. La méthode a été testée sur les mêmes exemples pratiques et les résultats ont montré sa précision.

Par la suite nous avons introduit le développement en polynômes de chaos

(PCE : Polynomial Chaos Expansion) afin de traiter des opérations non-linéaires. De manière similaire au cas des systèmes LTI, tout d'abord nous avons montré comment le comportement aléatoire des entrées peut être représenté sous la forme d'une PCE. Nous avons montré ensuite comment la PCE peut être adaptée pour traiter le cas des variables aléatoires. Ensuite, le cas des entrées corrélées a fait l'objet d'une analyse. Nous avons montré qu'en utilisant la transformée de Nataf il devient possible de décorreler les entrées.

La variabilité de l'entrée est statiquement propagée à travers le graphe des données en utilisant des formules de propagation pour chaque opération arithmétique. De cette manière, la représentation analytique de la sortie est obtenue statiquement.

Par opposition à la KLE, la méthode PCE peut être appliquée à tout type de système qui se compose des opérations arithmétiques et permet également de traiter les systèmes non linéaires.

En utilisant la même méthodologie probabiliste qui a été introduite pour la méthode KLE, l'intervalle de variation est calculé à partir de le FDP par rapport à une probabilité de débordement souhaitée. Les résultats montrent que les distributions obtenues sont proches des résultats obtenus en simulation. En plus, en utilisant notre analyse probabiliste, la taille de l'intervalle est significativement réduite par rapport à la méthode traditionnelle d'arithmétique d'intervalle.

Par rapport à la méthode KLE, l'utilisation des PCEs introduit une complexité plus importante. De ce fait, son applicabilité aux systèmes LTI est moins intéressante. Le nombre de termes qui sont utilisés pour une représentation précise PCE peut augmenter de manière significative avec la dimension et l'ordre choisis pour la représentation. Cela veut dire que pour les applications complexes et non-linéaires il peut devenir un facteur prohibitif dans le processus d'automatisation.

Ensuite, le développement en polynômes de chaos généralisé (gPCE : generalized Polynomial Chaos Expansion) a été introduit en vue de sélectionner une base de polynômes de chaos appropriée en fonction de la distribution du signal d'entrée. Nous avons montré comment le type de polynômes de chaos peut être choisi en fonction de la distribution de l'entrée afin de réduire le nombre de termes qui doivent être utilisés pour une représentation précise.

Enfin, l'évaluation de la précision numérique peut être faite en utilisant la même méthode. Le SNR est calculé à partir de la puissance du bruit de quantification. Dans ce cas l'utilisation des polynômes de Lagrange peut avoir une importance très grande parce que le bruit de quantification a une distribution uniforme.

En tant que perspectives, la complexité de la méthode PCE doit être réduite en utilisant uniquement une structure creuse des polynmes qui fournit seulement les termes les plus importants dans le développement tout en négligeant les autres termes.

Un autre aspect important qui doit être considéré est la mise en oeuvre du développement adaptatif en polynômes de chaos, basé sur le schéma d'Askey. Comme il a été présenté, le développement en polynômes de chaos classique qui utilise les polynmes d'Hermite n'est optimal que pour la représentation de la répartition gaussienne. Pour des distributions fortement non-gaussiennes, le taux de convergence peut être faible et un nombre important de termes est nécessaire. Un développement adaptatif qui modifie automatiquement les bases de ses polynômes, en fonction de la distribution de l'entrée peut significativement réduire la complexité et devrait être mis en oeuvre dans l'avenir.

Acknowledgements

My research would not have been possible without the partnership between INRIA and ST Microelectronics, therefore I have a special appreciation for all those who made it possible.

I would like to express my deepest gratitude to my supervisor, Prof. Emmanuel Casseau for giving me the opportunity to realize this PhD and also for his support and guidance throughout my thesis that allowed me to bring my research to an end.

I would also like to acknowledge my co-supervisor, Prof. Daniel Menard, for giving me the benefit of his knowledge without which I could not have advanced with my work.

I would like to give a special thank to Thierry Michel for his extremely valuable advices, encouragement and friendship.

Furthermore I am very grateful to Pascal Urard for allowing me to become a member of his team.

I wish to express my gratitude to all the committee members for agreeing to judge my work, for their time and effort.

Thanks go to the all my friends and colleagues for their advices and encouragement.

Last but not least, I wish to thank my family for the support they provided me throughout this thesis. Most of all, I must acknowledge my wife Eldda, for her love, presence and encouragement.

Contents

List of Figures

List of Tables

Glossary

16QAM 16-Quadrature Amplitude Modulation.

ADC Analog-to-Digital Converter.

ASIC Application-Specific Integrated Circuit.

BER Bit Error Rate.

DAC Digital-to-Analog Converter.

DFG Data-Flow Graph.

DSP Digital Signal Processing.

FFT Fast Fourier Transform.

FIR Finite Impulse Response.

HLS High-Level Synthesis.

IIR Infinite Impulse Response.

IFFT Inverse Fast Fourier Transform.

KLE Karhunen-Loève Expansion.

LTI Linear Time-Invariant.

OFDM Orthogonal Frequency-Division Multiplexing.

QPSK Quadrature Phase Shift Keying.

PAPR Peak-to Average Power Ratio.

PCE Polynomial Chaos Expansion.

PDF Probability Density Function.

SQNR Signal-to-Quantization-Noise Ratio.

VLSI Very-Large-Scale Integration.

Chapter 1

Introduction

1.1 Background And Motivation

1.2 Background And Motivation

Digital Signal Processing applications have experienced a very strong development in the last decades, benefiting from the major advances of the semiconductor industry. Nowadays, they can be found in a large variety of fields of activity, anywhere from telecommunications, multimedia, consumer electronics, transportation, medicine, military applications etc. The continued technological improvements have allowed the emergence of new products that use complex signal processing algorithms in order to meet the application demands. To improve the productivity and to satisfy the time-to-market constraints, various high-level tools have been developed at all stages of the design. They enable the transition from a high-level description of the application to a low-level description with a rapid exploration of the available solutions for the implementation.

The application specification determines the performance criteria that must be guaranteed by the system. An appropriate algorithm is developed to satisfy these needs. As a first step, a high-level description of the algorithm is specified using a theoretical infinite precision to alleviate problems related to the computational accuracy. This allows the validation of the mathematical algorithm. Numerical computing environments like Matlab [44], Mathematica [77] or Scilab [66] are used to simulate the high-level description.

However, most of the practical DSP implementations use fixed-point arithmetic to reduce the area and power consumption and obtain a cost-effective hardware. A conversion process from the floating-point description of the algorithm to a fixed-point implementation that customizes every wordlength in the datapath has to be realized.

The emergence of High-Level Synthesis tools like *Catapult C* from Mentor Graphics [50], *Cynthesizer* from Forte Design Systems [23] or *Symphony C* Compiler from Synopsys [71] that generate RTL implementations directly from a C/C++ fixed-point specification of the application, reduces the development time while allowing a good design space exploration. However, the floating-point to fixed-point conversion still needs to be done by hand and continues to be one of the most difficult part of the design. It is a time-consuming and error prone problem and finding a good trade-off is a very difficult task. It has been shown it can take up to 30% of the total development time [3, 11, 26, 29]. In order to reduce the gap between the algorithm description and the hardware implementation, to control the complexity and reduce the development time, an automatic floating-point to fixed-point conversion tool that optimizes the area and timing under performance constraint is mandatory. It is a source code transformation that can be then easily integrated into the digital hardware design flow (Figure 1.1).

The limited bit-width of the fixed-point data types will introduce a quantization error which generates a degradation of the computational accuracy. The accuracy of the result is given by the number of bits used for its representation. Increasing the wordlength of the datapath improves the accuracy at the expense of additional hardware cost (area, power consumption and delay). The fixed-point conversion becomes an optimization process [62] that minimizes the implementation cost for an acceptable degradation of the performance. In other words it must find the

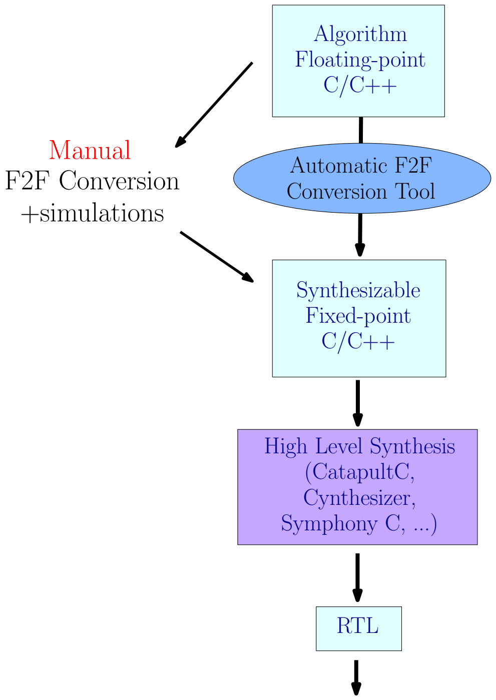


Figure 1.1: Digital Hardware Desing Flow

minimum integer and fractional part wordlengths for every fixed-point variable that still satisfy the overall computation accuracy required by the application (usually the SNR or the BER of the system). So the transformation problem can be separated in two different parts which are handled independently. The first part determines the fractional part wordlength and the second the integer part wordlength:

- the numerical accuracy analysis
- the range estimation

The numerical accuracy analysis is linked to the notion of quantization noise. It studies the sensitivity of the output to slight changes of the input translated to a certain error metric (typically the application signal-to-quantization-noise-ratio (SQNR)). In fact, number of works are focused on optimizing the fractional part wordlength using the power of the quantization noise as a performance criteria. The minimal bit-width is found based on a trade-off between the accuracy needed and the circuit cost.

The range estimation computes the minimum number of integer bits for a variable from its maximal and minimal values. Classical range estimation methods compute theoretical absolute bounds that will never be exceeded in practice to avoid the appearance of overflows. In doing so, they provide ranges that are pessimistic and the implementation cost will be largely increased. As the absence of overflows is guaranteed, the optimization of the integer part wordlength under performance constraints becomes impossible and the trade-off accuracy-implementation cost is considered only for the fractional part.

Some applications can however accept occasional overflows if their probability of occurrence is small enough not to affect the overall performance. As a result, the range estimation method should be able to take this property into account. In addition, methods like the interval and affine arithmetic do not provide additional information about the signal variation inside the interval of possible values making it a poor approximation of the real uncertainty. Signals that have large variations but have small probabilities at the tails of their probability distribution are not well taken into account. Moreover, existing methods of numerical accuracy analysis evaluate only the power of the output quantization noise. In some cases, like the evaluation of the performance in systems with unsmooth operators [58], this limited information proves to be insufficient and the entire probability density function of the noise should be determined.

1.3 Objectives

The floating-point to fixed-point conversion has been an active research project in the CAIRN/IRISA research laboratory. A framework for the automatic floating-point to fixed-point transformation has been developed [28, 48, 49]. Its synoptic is described in Figure 1.2.

The input application is described as a C/C++ implementation that uses floating-point representations for the variables. In addition, the performance constraints

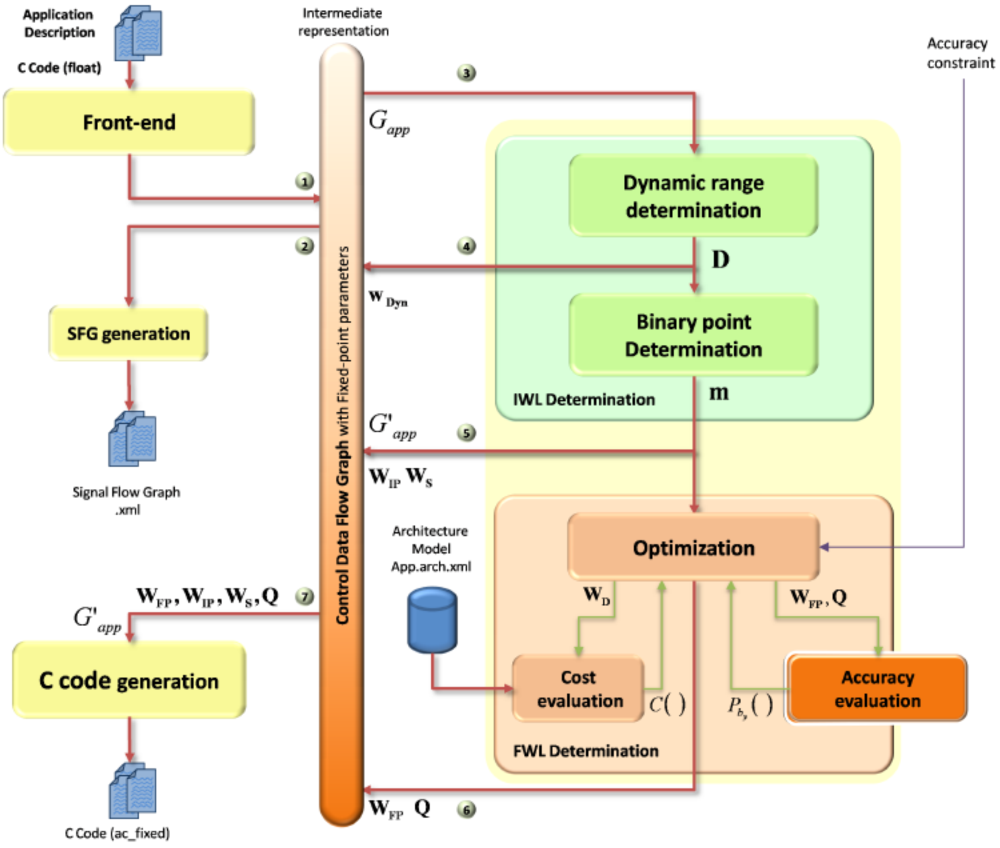

Application Description
C Code (float)
Front-end
Intermediate representation
Accuracy constraint
Control Data Flow Graph with Fixed-point parameters
1
2
3
4
5
6
7
G_{app}
Dynamic range determination
D
$\mathbf{W}_{Dyn}$
Binary point Determination
m
IWL Determination
SFG generation
Signal Flow Graph .xml
G'_{app}
$\mathbf{W}_{IP}\ \mathbf{W}_S$
Architecture Model App.arch.xml
Optimization
$\mathbf{W}_D$
$\mathbf{W}_{FP}, \mathbf{Q}$
Cost evaluation
$C(\ )$
$P_{b_y}(\ )$
Accuracy evaluation
FWL Determination
$\mathbf{W}_{FP}, \mathbf{W}_{IP}, \mathbf{W}_S, \mathbf{Q}$
G'_{app}
C code generation
C Code (ac_fixed)
$\mathbf{W}_{FP}\ \mathbf{Q}$

Figure 1.2: Fixed-point conversion tool developed by CAIRN/IRISA

(computational accuracy) that should be satisfied by the fixed-point implementation is provided by the user. It is used in the wordlength optimization part when the cost evaluation is realized .

The flow is separated in two parts. As a first step, the dynamic range of the variables is determined. The theoretical absolute minimal and maximal bounds are computed using the L1 norm or the interval arithmetic. They guarantee the absence of overflows. The number of bits necessary for the integer wordlength representation is directly computed afterwards.

The second part realizes the actual wordlength optimization under performance constraints. The quantization noise introduced by the limited bit-width of the data representation generates a degradation of the overall system performance. Increasing the width of the datapath reduces the finite wordlength effects at the expense of additional hardware. The accuracy evaluation is computed by evaluating the signal-to-quantization-noise ratio (SQNR). The optimization part consists in finding the minimal fractional part size that still satisfies the performance constraints. Finally, a fixed-point specification of the application is generated as a result using the data types that have been determined.

Following the research already done by the CAIRN team, the purpose of this thesis is to provide a probabilistic framework that solves the range estimation using a statistical criteria and which can be easily integrated in the automatic floating-point to fixed-point transformation tool. We are interested in optimizing the integer part wordlength when a slight degradation of the performances is acceptable. In fact the occurrence of overflows is allowed when their probability is sufficiently low for the given application. The integer wordlength doesn't cover anymore the entire theoretical dynamic range, instead it adapts its width to the application needs from a probabilistic stand point. It is computed from the probability density function (PDF) using a statistical analysis. In this way, more information about the variation of the signal than simple bounds is provided. Indeed, an approach that captures the entire distribution and the correlation between data can significantly improve results compared to the classical approaches like the interval and affine interval [13, 18, 51]. A real example consisting of an OFDM transmitter is used as a test case to motivate and validate the probabilistic approach.

As a secondary goal, the problem of numerical accuracy estimation is addressed. In the case of unsmooth operators, the traditional approaches to the numerical accuracy analysis that compute the power of the quantization noise have proved their limitations. In order to solve the problem, additional information about the noise variation is needed. The same probabilistic approach can be extended to evaluate the quantization noise, with the interest of computing the entire PDF of the output noise.

1.4 Dissertation Outline

The thesis is organized as follows. In Chapter 2, the effects of the finite wordlength representation of numbers on the accuracy of the result in digital computation are presented. Starting from a description of the floating-point and fixed-point number representations, the quantization process is introduced. The rounding/truncation and overflow degrade the accuracy of the computation. A state-of-the art review of the existing methods for the wordlength optimization under performance constraints is made.

In Chapter 3, we present our approach for the range estimation problem. The optimization of the datapath is made in compliance with the performance requirements of the application and with the statistical description of the input. Using a probabilistic framework, the necessary number of bits for the integer part representation are computed using a desired probability of overflow. A real test case is presented as a practical example that validates our method.

The case of linear time-invariant (LTI) systems is considered in Chapter 4. The Karhunen-Loeѐve Expansion (KLE) is used as a means of discretization for the input of the system. Using the superposition property and the transfer function of the system under investigation, the output KLE description can be computed. The overflow probability is computed from an estimation of the PDF. The numerical accuracy is analyzed using the KLE representation of the quantization noise.

The Polynomial Chaos Expansion (PCE) is introduced in Chapter 5. Representing every variable with a PCE, the variability can be propagated through the Data Flow Graph (DFG) from the input to the output. The advantage of the PCE representation is the fact that the PCE arithmetic can be applied for non-linear operations also. As a result the range and the numerical accuracy estimation problems is solved for all types of systems with arithmetic operations.

Chapter 6 presents the conclusion of the work and proposes some perspectives.

Chapter 2

Finite Wordlength Effects

In this Chapter the floating-point and fixed-point number representations are presented. A comparison between the two is realized in order to analyze the finite wordlength effects. The floating-point to fixed-point conversion process under performance constraints is introduced. The problem is divided in two separate parts that can be treated independently: the dynamic range estimation and the numerical accuracy analysis. A literature review of the related work is presented.

2.1 Number Representation

In digital computation, the numeral system specifies the way numbers are represented as a sequence of binary digits and the rules for performing arithmetic operations (e.g. addition, multiplication etc.) between them. Most of the times, the scientific computations provide only an approximation of the exact value (that would be obtained having an infinite precision). This is a consequence of the limited number of bits that can be used in practice by the numeral system. Whether the floating-point or the fixed-point arithmetic is employed, only a finite number of bits are used for the representation of real numbers.

The limited precision of the coding standard can be evaluated from two different perspectives. The accuracy of the computation is given by the quantization step of the numeral system (the distance between two successive numbers). The second aspect is the maximal dynamic variation that is allowed by the representation. The dynamic range variation of a numeral system is given by the domain of possible values that can be represented. It is evaluated by the ratio between the largest (X_{MAX}) and the smallest (X_{MIN}) magnitude that can be represented by the coding standard using a logarithmic scale as in equation (2.1). As a result, the comparison between the floating-point and the fixed-point standards is made by analyzing the numerical accuracy and the dynamic range variation that they ensure.

$$D_{dB} = 20 \log_{10} \left(\frac{X_{MAX}}{X_{MIN}} \right) \tag{2.1}$$

For embedded systems, algorithms are generally developed using the floating-point arithmetic, in order to avoid all the problems related to the finite wordlength. This is a process that validates the reliability of the algorithm solution for the given problem. Even though the inherent error in the computational accuracy still exists, it is very small compared to the fixed-point arithmetic. As a result, the floating-point computation guarantees an accuracy and a dynamic range variation that is sufficient in most of the cases.

Nevertheless, most of all VLSI implementations use fixed-point arithmetic to reduce the area and power consumption and obtain a cost-effective hardware. As a consequence of the limited bit-width of the data representation, a degradation of the computational accuracy is produced. The use of fixed-point data types introduces a quantization noise when bits are eliminated through rounding/truncation operations. In addition, it causes the appearance of overflows whenever the integer part wordlength is insufficient to represent the entire dynamic range variation.

To better understand the problem, a description of the two coding standards is made. A comparison between them is made with an emphasis on the dynamic range variation that they allow and on the computational accuracy that is guaranteed.

2.2 Floating-point Representation

The floating-point number system is the most common coding standard when a high computational accuracy is required. It represents a real number in a scientific notation, with a fractional part called the mantissa (or the significant) and a scale factor called the exponent. The exponent is defined as the power of the base (typically two or ten) and is used as an explicit scale factor that changes during computations, allowing a wide dynamic range of values to be represented. The mantissa determines the accuracy of the represented number. The general representation of a floating-point number can be seen in Figure 2.1 and the associated value is given by the expression in (2.2). S represents the sign of the number, M is the mantissa, E is the exponent and b is the base of the representation.

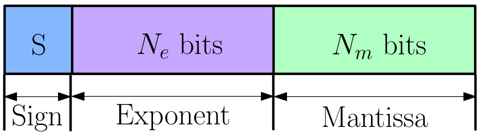


Figure 2.1: Floating-point Number Representation Format

$$x = (-1)^{(S)} \times M \times b^{E} \tag{2.2}$$

As there is a large number of possible values for N_m and N_e, a standardized computer floating-point format has been introduced. The *IEEE Standard for Binary Floating-Point Arithmetic* (IEEE 754-2008) is used by almost all of today CPUs. It specifies the floating-point formats as well as the rounding modes, it describes how arithmetic operations should be realized and the exception handling (division by zero, overflows).

The value of a number represented in the binary IEEE 754 floating-point format is computed using equation (2.3):

$$x = (-1)^{(S)} \times 1.M \times 2^{E-bias} \tag{2.3}$$

The mantissa is normalized to represent a value in the interval [1:2). As a consequence the value of its first bit is fixed to 1 and becomes implicit, meaning that it is not necessary to be stored. The value of the exponent is encoded as an unsigned number, so in order to represent numbers that are smaller than 1, a bias is introduced. The bias depends on the number of bits that are allocated for the representation of exponent: $bias = 2^{N_e-1} - 1$. In the case of the single precision format, it is 127 and the exponent range for normalized numbers is [-126, 127]. For the double precision, the bias is 1023, and the range of the exponent is [-1022, 1023].

From the basic formats, the single precision (32 bits) and the double precision (64 bits) are the most widely used. They are presented in Table 2.1.

	Sign (S)	Exponent (N_e)	Mantissa(N_m)	Bias
Single Precision	1	8	23	127
Double Precision	1	11	52	1023

Table 2.1: IEEE 754 Standard

2.2.1 Dynamic Range Variation

The dynamic range variation of the floating-point representation can be determined as in (2.4).

$$\begin{gathered} D_{dB} = 20\log_{10}\left(\tfrac{X_{MAX}}{X_{MIN}}\right) \backsimeq 20\log_{10}\left(2^{2K+1}\right) \\ \text{with } K = 2^{N_e-1} - 1 \end{gathered} \tag{2.4}$$

For a *single precision* number that has the exponent represented with 8 bits, the dynamic range variation becomes:

$$D_{dB} = 20\log_{10}(2^{(2^8-1)}) = 20\log_{10}(2^{255}) \backsimeq 1535\ dB \tag{2.5}$$

2.2.2 Computation Accuracy

Because of its inherent scientific representation, as the value of exponent increases, the distance between two successive numbers becomes larger. This means that the quantization step of the floating-point coding standard depends on the value that is represented. As a consequence, the computational accuracy of the floating-point representation is proportional to the magnitude of the number that is encoded. As the magnitude of the number increases, the round-off error gets larger.

The maximal and minimal bounds of the quantization step (q) relative to the value that is represented (x) can be determined using equation (2.6). It shows how the quantization step is adapted to the magnitude of the number. When the value is small, the quantization step is also small, and when the value of the number is large, the quantization step becomes also large.

$$2^{-(M+1)} < \frac{q}{|x|} < 2^{-M} \tag{2.6}$$

The analysis of the floating-point quantization noise is made in [74, 75]. When some appropriate requirements are met, which the authors call the ”pseudo quantization noise” model, the floating-point quantization noise e_{fp} has a zero mean and it is uncorrelated with the input signal, x. Its second order moment can be computed as in (2.7). As expected, the value of the quantization error is a function of the value of the signal.

$$E[e_{fp}] = 0.180 \times 2^{-2M} \times E[x^2] \tag{2.7}$$

2.3 Fixed-point Representation

The fixed-point format is a binary code word where numbers are represented using an integer and a fractional part. The general form of a signed fixed-point number is presented in Figure 2.2. One bit is used for the sign (S), m bits are used for the encoding of the integer part and n bits for the fractional part.

Every bit is associated to a weight corresponding to a power of two. The fractional part provides the subunit representation of the number and coincides with the negative powers of two $(2^{-1}, 2^{-2}, \cdots)$. The position of the radix point is fixed during the processing, so the implicit scale factor used by the representation is constant and the range of values that can be represented does not change during computations.

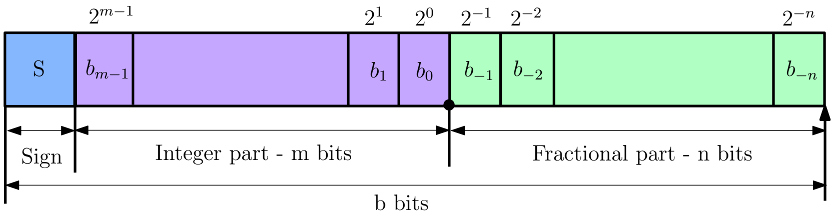


Figure 2.2: Fixed-point Representation of a Number

Generally, fixed-point numbers are encoded using two's complement standard. The value of a number is given by the expression presented in equation (2.8). It possesses some interesting arithmetical properties regarding the addition and the subtraction operations and it also has the advantage of allowing only one possible representation for 0. As a consequence the domain of possible values is not symmetrical to the origin, having $2^{(m+n)}$ negative values and $2^{(m+n)} - 1$ positive values.

$$x = -2^m S + \textstyle\sum_{i=-n}^{m-1} b_i 2^i \tag{2.8}$$

The maximal and minimal values that can be represented are given by the location of the binary point (equation 2.9). In addition, the quantization in fixed-point arithmetic is uniform and the quantization step is not proportional to the value that is represented, being constant for the entire dynamic scale: $q = 2^{-n}$.

$$\begin{gathered} -2^m \leq x < 2^m \\ x \in [-2^m : 2^m - 2^{-n}] \end{gathered} \tag{2.9}$$

As a consequence, the finite wordlength effect in the case of fixed-point numbers can be separated in two different problems that are represented in Figure 2.3. The increase of the integer part wordlength will extend the dynamic range that is covered by the representation because of the implicit multiplication of the scale factor. At the other side, enlarging the fractional part wordlength will enhance the accuracy of the number representation as the quantization step is decreased.

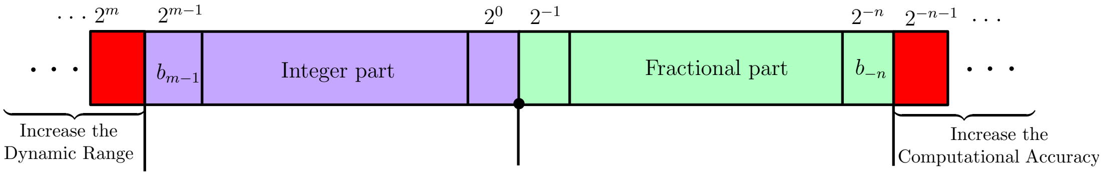


Figure 2.3: Effects of fixed-point wordlength variation

2.3.1 Dynamic Range Variation

For the fixed-point format, the dynamic range variation is linear with the number of bits, b, used for the representation:

$$\begin{gathered} D_{dB} = 20\log_{10}\left(\frac{X_{MAX}}{X_{MIN}}\right) = 20\log_{10}\left(2^{b-1}\right)\text{dB} \\ D_{dB} = 20(b-1)\log_{10}(2) \approx 6.02(b-1) \end{gathered} \tag{2.10}$$

The increase of the dynamic range variation with the wordlength is much larger for floating-point numbers than for fixed-point numbers.

As an example, in Table 2.2 the single precision format is compared with various fixed-point data types. An important difference between them can be observed, even the 128 bits fixed-point number has a significantly smaller dynamic range variation than the 32 bits floating-point representation.

	Dynamic Range (dB)
Single Precision	1535
Fixed-point 16 bits	90
Fixed-point 32 bits	186
Fixed-point 64 bits	379
Fixed-point 128 bits	764

Table 2.2: Dynamic range variation comparison

In Figure 2.4 the evolution of the dynamic range variation for the floating-point and fixed-point data types is presented. In this example, the size of the exponent is fixed to $\frac{1}{4}$ of the total wordlength of the representation. When the wordlength exceeds 16 bits, the dynamic range variation for the floating-point representation becomes larger than in the case of the fixed-point. As a result, the 32-bit floating-point representation can be used in most applications without any risk of overflow ocurence.

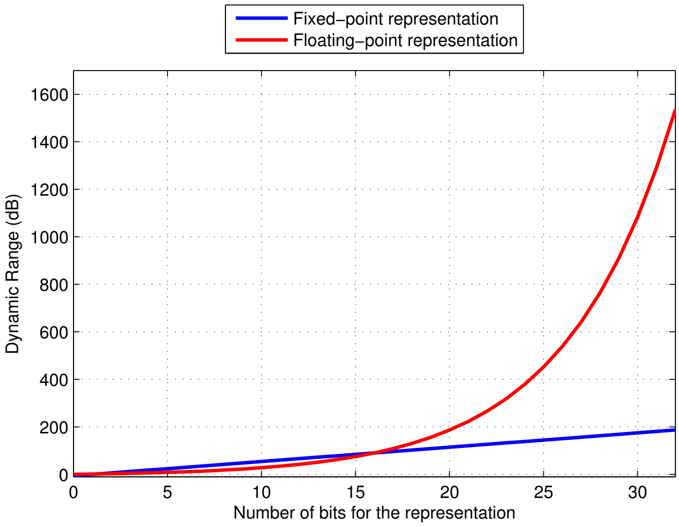


Figure 2.4: Dynamic range variation comparison between the floating-point and the fixed-point representations

The coding process of a fixed-point number can be defined as the representation of a real value x with another value $\widehat{x}$ from the coding domain. Every time a real number exceeds the allowed range of values defined by the coding standard, meaning that $x \notin [\widehat{x}_{min}, \widehat{x}_{max}]$, an overflow occurs and an important error is introduced. The overflow handling describes how a code word is assigned when such an event takes place. There are two methods that can be used for the treatment of overflows. The natural way of dealing with the problem results in a wrap-around of the value. The process can be seen in Figure 2.5. The technique is equivalent to a modular arithmetic as the value that exceeds the bounds is replaced with its value modulo 2^b.

The second method that can be applied is the saturation arithmetic. In this case, any value that exceeds the coding domain is replaced with its closest representable number (the maximal or minimal bound). The process is represented in Figure 2.6. The error that is introduced is smaller than in the case of the modular arithmetic. However, as opposed to the wrap-around technique, the implementation of the saturation arithmetic requires additional hardware so its use is limited in

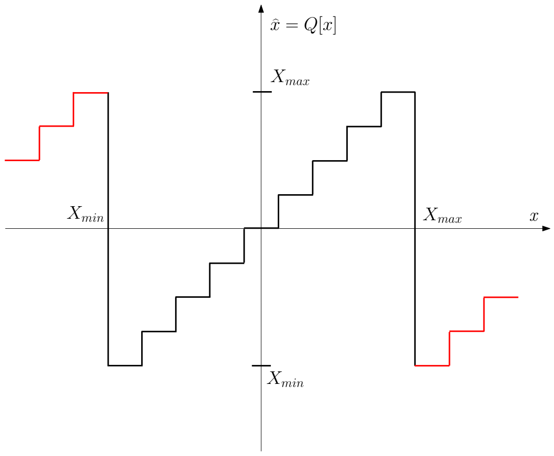


Figure 2.5: Overflow effects using the wrap-around technique

practice.

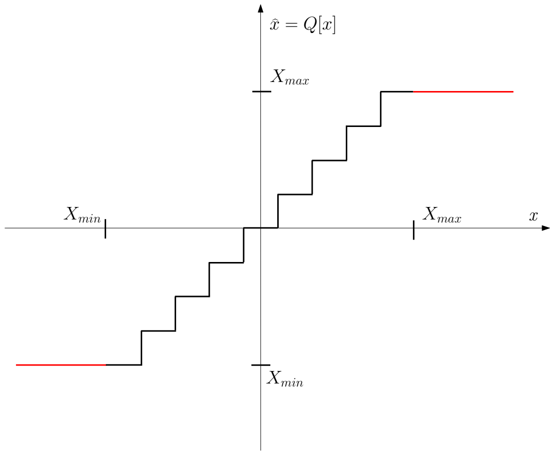


Figure 2.6: Overflow effects using the saturation technique

2.3.2 Numerical Accuracy Analysis

2.3.2.1 Quantization process

The mechanism of assigning a sequence of binary digits for the representation of a real (analogous) value x, is realized by the quantization process. The operation is presented in (2.11), where the value of the signal x is transformed into a fixed-point representation denoted by $\widehat{x}$.

$$x \rightarrow \widehat{x} = Q(x) \tag{2.11}$$

It is a nonlinear procedure that generates a loss in the precision as only a finite number of possible values can be represented. More exactly, when b bits are used for the fixed-point number, 2^b distinct values can be represented. The error that results from the difference between the real value x and the fixed-point representation $\widehat{x}$ is called the quantization noise.

$$e(x) = \widehat{x} - x \tag{2.12}$$

The resolution of the representation is given by the difference between two consecutive numbers and is denoted by q. Its value is determined by the position of the least significant bit (LSB) (2^{-n}). The most widely used quantization modes are the round-off and the truncation.

Rounding quantization

When the round-off quantization is applied (Figure 2.7), the magnitude of the signal is rounded to the nearest quantization level. The maximum error that is introduced is $\pm\frac{1}{2}LSB$. This means that the quantization error that is introduces $(e(x))$ in this case is bounded in the interval $[-\frac{q}{2}, \frac{q}{2}]$.

$$\widehat{x} = Q(x) = \Delta_i + \tfrac{q}{2}, \quad \forall x \in [\Delta_i, \Delta_{i+1}] \tag{2.13}$$

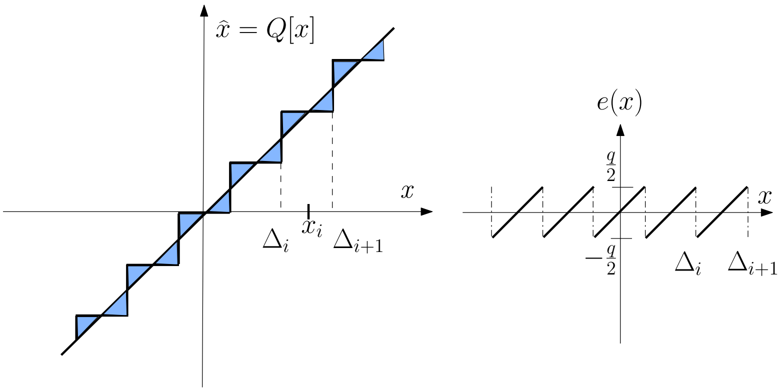


Figure 2.7: Rounding quantization process

Truncation quantization

The truncation method (Figure 2.8) consists in choosing the inferior quantization level for the representation of the signal. As a result the quantization error is always positive, $e(x) \in [0, q]$ and an offset is introduced.

$$\widehat{x} = \Delta_i, \quad \forall x \in [\Delta_i, \Delta_{i+1}] \tag{2.14}$$

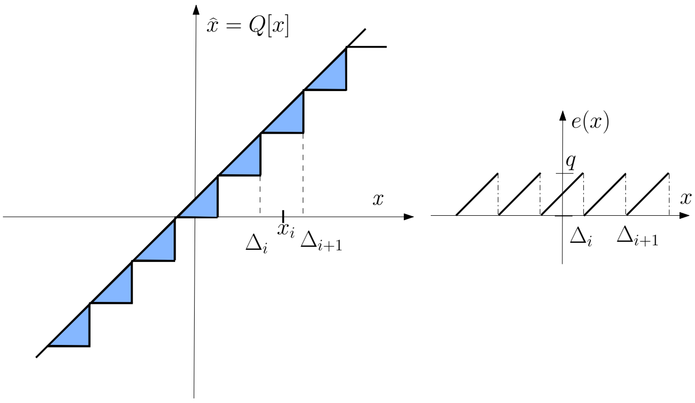


Figure 2.8: Truncation quantization process

Analysis of the quantization noise

The results presented by Widrow [72, 73] show that the quantization process can be modelled by the introduction of an additive noise. The output of a quantizer is equal to the input signal x, plus a random variable e, that represents the quantization error as it can be seen in Figure 2.9.

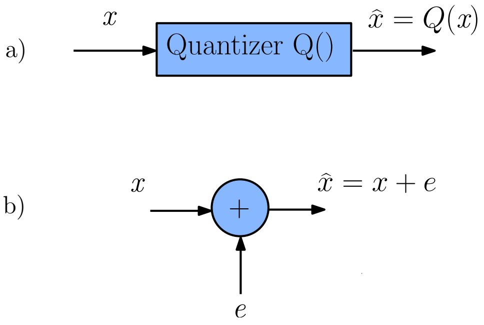


Figure 2.9: Addtive quantization noise model

In the case of the round-off quantization, the authors showed that the error is uniformly distributed in the interval $[-\frac{q}{2}, \frac{q}{2}]$. Therefore, it has a mean (μ_e) that is equal to zero (2.15), and a variance (σ_e^2) that can be computed using equation (2.16) with f_e the PDF of the noise.

$$\mu_e = \int_{-\infty}^{\infty} e f_e(e) de = \int_{-\frac{q}{2}}^{\frac{q}{2}} \frac{1}{q}\ e de = 0 \tag{2.15}$$

$$\sigma_e^2 = \int_{-\infty}^{\infty} (e - \mu_e) f_e(e) de = \int_{-\frac{q}{2}}^{\frac{q}{2}} \frac{1}{q}\ e^2 de = \frac{q^2}{12} \tag{2.16}$$

When the truncation is used, the error is uniformly distributed in the interval [0, q] and it has a mean equal to $\frac{q}{2}$ (the offset). The variance is given in (2.18).

$$\mu_e = \int_{-\infty}^{\infty} e f_e(e) de = \int_{0}^{q} \frac{1}{q}\ e de = \frac{q}{2} \tag{2.17}$$

$$\sigma_e^2 = \int_{-\infty}^{\infty} (e - \mu_e) f_e(e) de = \int_{0}^{q} \frac{1}{q}\ (e - \frac{q}{2})^2 de = \frac{q^2}{12} \tag{2.18}$$

In addition, the autocorrelation function and the correlation with the signal are also analyzed in [75]. It results that the quantization noise can be considered to be a white noise, non-correlated with the signal and independent from other noise sources.

Signal-to-quantization noise ratio

In DSP applications, the most common performance criteria that describes the computational accuracy is the signal-to-quantization-noise ratio(SQNR). The SQNR is defined as the ratio between the power of the signal (P_x) and the power of the quantization noise (P_e) and is often expressed using the logarithmic scale:

$$SQNR_{dB} = 10 \log_{10} \left(\frac{P_x}{P_e} \right) = 10 \log_{10} \frac{E[x^2]}{E[e^2]} \tag{2.19}$$

As the fixed-point representation has a uniform quantization, the SQNR is *linearly dependent on the signal amplitude.* When the amplitude of the signal increases, the quantization noise ratio becomes larger and the SQNR is improved.

Consider the case of a full scale sinusoïdal signal with the amplitude $A = 2^n$. The variance of the signal is then: $\sigma_x^2 = \frac{A^2}{2}$. The SQNR becomes:

$$\begin{aligned} SQNR &= 20\log_{10}\left(2^n\sqrt{\tfrac{3}{2}}\right) \\ &\approx 1.76 + 6.02n\ dB \end{aligned} \tag{2.20}$$

As a conclusion, in the case of the fixed-point representation with a signal that is fully scaled, each additional bit increases the SQNR with approximately 6 dB.

As opposed to the fixed-point case, the floating-point representation has the advantage of having a quantization step that is proportional to the amplitude of the signal. The value of the SQNR using a logarithmic scale is given by the expression in (2.21). It depends on the number of bits that are used for the representation of the mantissa. However, the SQNR of the floating-point representation is not a function of the amplitude of the signal, and can be considered constant for all the values of x.

$$\begin{aligned} SQNR = 10\log_{10}(\tfrac{E[x^2]}{e_{fp}}) &= 10\log_{10}(5.55 \times 2^{2m}) \\ SQNR &\approx 7.44 + 6.02m \end{aligned} \tag{2.21}$$

In Table (2.3), a comparison between the fixed-point data types and the single and double precision is presented. It can be seen that for an equivalent number of bits, the fixed-point representation can guarantee a larger SQNR than the floating-point number if it is *properly scaled.*

	SQNR (dB)
Single Precision	151
Double precision	326
Fixed-point 32 bits	194
Fixed-point 64 bits	387

Table 2.3: SQNR comparison

2.4 Wordlength Optimization

In embedded systems the floating-point number system provides a good environment for the development and the validation of DSP algorithms as all the problems related to the finite wordlength effects can be mitigated. Because of the sizeable dimension of the allowed dynamic range variation, overflows are almost inexistent. In addition, even though it introduces a representation error, the floating-point format ensures a SQNR (especially in double precision) that is sufficiently large for most applications. However, all the advantages come at the expense of an increased implementation cost. When compared to the fixed-point arithmetic, the floating-point operations are more complex to realize because of the inherent structure of the representation. Since the exponent varies during computations an alignment of the fractional part of both operands has to be realized. Moreover, the mantissa must be stored in a normalized form, so a re-normalization is required after each operation. The cost of a simple addition in floating-point arithmetic is increased in a large extent due to the complexity of the supplementary procedures. Multiplications do not demand an alignment of the operands but the re-normalization of the mantissa is still needed. As a consequence, in applications that have high throughput or area and power consumption constraints the additional cost becomes inacceptable in most cases.

The advantage of the fixed-point arithmetic is that the wordlengths of all the operands can be optimized so that the memory and bus sizes can be reduced. In addition, the operations are less complex to execute so the overall implementation cost (area and power consumption) is greatly decreased in comparison to the floating-point. As a result, most of all practical DSP applications use fixed-point arithmetic and a conversion process from the floating-point representation of the algorithm to the corresponding fixed-point implementation has to be made.

The conversion process has been mathematically formulated [62] as an optimization problem, where the hardware cost must be minimized with a constraint on the performance criteria for the fixed-point application. In order to determine if the performance criteria is satisfied, the evaluation of both the numerical accuracy and the dynamic range of the fixed-point application has to be realized. The process is thus translated into the determination of the fractional part wordlength that ensures a sufficiently large SQNR for the application and the integer part wordlength that avoids the occurrence of overflows.

The SQNR is proportional to the dynamic of the signal. If the input signal is not appropriately scaled for the fixed-point data types, it can be significantly reduced. However, by increasing the amplitude of the signal, the probability of overflow events becomes larger. When the wordlength of the datapath is limited to a fixed bit-size and the signal has a large variation of its amplitude, a trade-off between a high SQNR and the appearance of unwanted overflows has to be done.

2.5 State of the art

In this section, a review of the existing methods for the floating-point to fixed-point conversion is presented. As it has been shown, the problem is divided in two different parts. At the beginning, the range estimation problem is presented. Afterwards, the case of the numerical accuracy evaluation is addressed.

2.5.1 Range Estimation

In order to avoid the occurrence of overflows, the integer part wordlength has to cover the entire range of possible values. If the extreme values (maxima and minima) are known, the minimum integer wordlength (I_{WL}) for a signed variable in two's complement representation can be calculated as:

$$I_{WL} = \begin{cases} \lceil \log_2(|x_{MIN}|) + 1 \rceil & \text{if } |x_{MIN}| > |x_{MAX}| \\ \lceil \log_2(|x_{MAX}| + 1) + 1 \rceil & \text{otherwise} \end{cases} \tag{2.22}$$

where $\lceil x \rceil$ represents the smallest integer not less than x.

Two different cases may arise in practice if the size of the integer part is incorrect. If fewer bits are used for the representation, the overflows will degrade the computational performance of the implementation. If, on the contrary, the bitwidth exceeds the needs, the hardware implementation costs is unnecessary increased.

The existing methods can be separated in two categories:

- **simulation-based methods**, which estimate the range of values for each variable using the extreme values obtained in simulation
- **analytical methods**, which are purely deterministic procedures that provide theoretical results using a description of the input variability

Simulation-based methods

Methodologies proposed in [33, 35, 37, 69] for the automatic range estimation problem are based on Monte Carlo simulation. Large amount of input stimuli are processed and the variable bounds are estimated using the extreme values obtained from simulation of the floating-point model.

The basic method extracts the range of the signals directly from peak-to-peak values obtain by simulation. Improved methods consider that all data are random variables and they try to estimate the range using their statistics estimated from simulation. In [33] the floating-point model is simulated and the mean and standard deviation are calculated from the sum and the squared-sum of the samples. The actual range is estimated for every variable in the program as follows:

$$R(x) = |\mu(x) + n\sigma(x)| \tag{2.23}$$

where n is a user specified integer that is usually in the interval $[4, 16]$. A larger value for n will give a more conservative estimation of the range. This will decrease the possibility of overflows at the expense of larger wordlengths.

In [35] a more elaborated statistical procedure is proposed to calculate the ranges where the signals are differentiated from a probability density function stand point:

- unimodal/multimodal
- symmetric/non symmetric
- zero mean/non zero mean

The symmetry of a probability distribution can be determined using the skewness coefficient (2.24). A nonzero skewness implies an asymmetrical distribution function.

$$s = \frac{\mu_3}{\sigma^3} \tag{2.24}$$

where μ_3 is the 3^{rd} order moment and σ is the standard deviation.

A function is unimodal if it has only one local maxima. This property cannot be directly determined, so the authors propose an heuristic method. A distribution is unimodal if its kurtosis, expressed in (2.25), is in the interval $[-1.2, 5]$.

$$k = \frac{\mu_4}{\sigma^4} - 3 \tag{2.25}$$

where μ_4 is the 4^{th} order moment and σ is the standard deviation.

For an unimodal and symmetrical probability distributions the range can be calculated as in (2.23). Knowing that it is dependent on the kurtosis, n is chosen in practice to be $k + 4$. For all other types of distributions the above formula can not be applied anymore. So the authors introduce a new computation method:

$$R(x) = R_{99,9\%}(x) + g \tag{2.26}$$

where $g = (R_{100\%} - R_{99,9\%})r_R$ is a guard value and $R_{99,9\%}$ is a sub maximal value which covers 99,9% of the entire samples.

This method [35] needs a large amount of data in order to obtain a reliable estimation and thus the simulation time can be extremely long. In addition, correctness for non simulated conditions is unknown. If the sequence of input patterns is chosen to be too short or incorrectly distributed the extreme values that are encountered in practice are not discovered. The possibility of overflows for rare events exists and their probability cannot be determined. Its most important advantage is that it can be applied to any type of system.

Analytical methods are based on the principle that every input data has a defined range of possible values which can be statically propagated through the system. At the end, the corresponding range for every intermediate and output variable is obtained. in other word, this means that the variability of the result of arithmetic operations can be analytically determined from the range of the operands.

L_p norm and transfer function based methods

In [7, 31] a methodology for Linear Time-Invariant (LTI) systems is described based on the L_1 norm and using the transfer function. A LTI system can be completely characterized by its impulse response function. For a system with N inputs, let $h_{ik}(n)$ be the impulse response from the input x_i to a certain variable y_k. Then:

$$y_k(n) = \sum_{i=0}^{N-1} h_{ik} * x_i(n) \tag{2.27}$$

This means that its absolute value is:

$$\max(|y_k(n)|) = \sum_{m=-\infty}^{m=\infty} |h_{ik}| \sum_{i=0}^{N-1} \max(|x_i(n)|) \tag{2.28}$$

Or, in a more abstract form:

$$\|y_k(n)\|_\infty = \|h_{ik}\|_1 \times \|x_i(n)\|_\infty \tag{2.29}$$

As a result, if the maximal and minimal values of the input are known, the dynamic range can be computed for every variable in the system. This method can be used for any type of input signal and gives theoretical bounds for the output that guarantee no overflow will occur. Taking in consideration only the maximum values of the signals and not its statistics, it will generally give conservative results. As an example, in [33] it is shown that for a fourth order IIR filter, the L1 norm will give results 4 bits larger than the results obtained by simulation for a real speech signal.

Interval Arithmetic

The interval arithmetic (IA) method was originally proposed by Moore [51] in the 1960s. Every signal is represented by an interval of possible values $[x_{min}, x_{max}]$, meaning that the true value of x varies between the two bounds.

$$[x_{min}, x_{max}] = \{x \in \Re | x_{min} \leq x \leq x_{max}\} \tag{2.30}$$

For every basic arithmetic operation a propagation rule is defined which provides the interval of possible values of the output variable. As an example, the addition and the multiplication can be computed as in (2.31) and (2.32) respectively.

$$\begin{gathered} x = [x_{min}, x_{max}] \ ; \ y = [y_{min}, y_{max}] \\ z = x + y = [x_{min} + y_{min}, x_{max} + y_{max}] \end{gathered} \tag{2.31}$$

$$\begin{gathered} x = [x_{min}, x_{max}] \ ; \ y = [y_{min}, y_{max}] \\ z = x \times y = [\min(E), \max(E)] \\ E = (x_{min} \times y_{min}, x_{min} \times y_{max}, x_{max} \times y_{min}, x_{max} \times y_{max}) \end{gathered} \tag{2.32}$$

It can be shown that IA is equivalent to the L1 norm method for non-recursive LTI systems. The advantage of this method is that it computes the variable ranges at compilation time and it is not data dependent, thus providing guaranteed accuracy. On the other hand this method considers that all the signals are independent and may take any value in their given interval. However, if there is a correlation between the operands, not all the values in the obtained interval are truly possible and thus the method will provide overestimated bounds. This is particularly important in systems with long datapaths or feedback loops where the bounds grow with every iteration.

As an example, lets determine the range for $y = x - x$, with $x = [-1, 1]$. The result that is obtained is $y = [-2, 2]$. So instead of being 0 the interval of possible values has length that is twice as large as the size of the operands.

An improvement of the IA method that has been proposed is the Multi-Interval Arithmetic [4, 8]. The method is based on the interval arithmetic but splits each interval into P disjoint subintervals:

$$[x_{min}, x_{max}] = \bigcup_{i=1}^{P} [x_{i_1}, x_{i_2}] \tag{2.33}$$

For each combination of subintervals a basic single-interval propagation is performed and the total dynamic range is determined by merging all the intermediate intervals. Because the operations are performed on smaller intervals the dimensions of the final results is reduced in comparison to the traditional IA method. However it does not address the correlation problem.

Affine arithmetic

One of the solutions proposed to solve the dependency problem is the affine arithmetic (AA) method [13, 18, 21, 22]. The authors extend the classical interval arithmetic integrating the source and the sign amplitude of all uncertainties. A variable $\widehat{x}$ will take the form of an affine equation (first degree polynomial)(2.34) between variables.

$$\widehat{x} = x_0 + x_1 \times \epsilon_1 + \cdots + x_n \times \epsilon_n \tag{2.34}$$

where ϵ_i is an independent source of uncertainty or error in the interval $\in [-1, 1]$ which adds to the total uncertainty of the variable $\widehat{x}$. x_0 is called the center value of the variable while $x_1, x_2, \cdots x_n$ are called partial deviations associated with the noise symbols.

For any variable that is represented with an affine form, the corresponding interval of values is determined as:

$$x \in [x_{min}, x_{max}] = [x_0 - r_x, x_0 + r_x] \tag{2.35}$$

with $r_x = |x_1| + |x_2| + \cdots + |x_n|$

The most important property of the method is that a noise coefficient can be shared between variables, keeping track of first order correlation (also called spatial dependency) between them. Similarly to the IA, using the affine arithmetic the variability can be propagated through the arithmetic operations, from the input to the output. This step is straightforward for all affine operations as they will preserve the affine property for the result (2.36).

$$\begin{aligned} \widehat{x} &= x_0 + x_1 \times \epsilon_1 + \cdots + x_n \times \epsilon_n \\ \widehat{y} &= y_0 + y_1 \times \epsilon_1 + \cdots + y_n \times \epsilon_n \\ \widehat{z} = \widehat{x} + \widehat{y} &= x_0 + y_0 + \textstyle\sum_{i=1}^{n}(x_i + y_i) \times \epsilon_i \end{aligned} \tag{2.36}$$

The example from IA is considered, $y = x - x$. Using the AA the value of the results is correctly determined:

$$\begin{aligned} \widehat{x} &= x_0 + x_1 \times \epsilon_1 \\ y = x - x &= x_0 + x_1 \times \epsilon_1 - x_0 - x_1 \times \epsilon_1 = 0 \end{aligned} \tag{2.37}$$

However non-affine operations will not conserve the affine form and the result is required to be linearized resulting in the loss of information and oversized bounds. For example, the multiplication operation is realized as in (2.38). Other non-affine operations can be treated as well [22].

$$\begin{aligned} \widehat{z} = \widehat{x} \times \widehat{y} &= (x_0 + \textstyle\sum_{i=1}^{n} x_i \times \epsilon_i)(y_0 + \sum_{i=1}^{n} y_i \times \epsilon_i) \\ \widehat{z} &= (x_0 \times y_0) + \textstyle\sum_{i=1}^{n}(x_0 \times y_i + y_0 \times x_i) \times \epsilon_i + z_k \times \epsilon_k \end{aligned} \tag{2.38}$$

with $z_k = \sum_{i=1}^{n} |x_i| \times \sum_{i=1}^{n} |x_i|$

The number of noise variables will increase with each non-linear operator. As each one of these uncertainties is independent from others, the correlation between signals will be lost. In conclusion, due to the limited additional information about signals variation, the correlations between signals is not very well used and the range will explode for complex applications.

Probabilistic interval-valued computation

While analytical range estimation methods like IA and AA provide a way to compute the dynamic range in a purely deterministic manner, they provide limited information about signals variation. As a consequence temporal and spatial correlations between signals are not very well managed and the range of values may explode for complex applications. In [64, 65] a novel interval algebra is proposed, refining the affine model from a statistical stand point. Range uncertainties are replaced with confidence intervals referred to as probabilistic intervals.

The authors identify three important problems in the basic interval methods that they try to resolve:

- symmetrical interval bounds
- large operator bounds especially for non-linear operations
- absence of a statistical foundation

To allow asymmetric ranges, the authors add two enforced bounds to the affine model which can be computed if the result is known not to exceed certain values. So the new representation for a variable becomes: (2.39). $[x_l, x_h]$ are the enforced bounds that are imposed to the affine model $\widehat{x}$. Therefore, $\widehat{x}$ cannot have any value that is outside of the interval $[x_l, x_h]$.

$$\{\widehat{x}, [x]\} = \{x_0 + \sum_{i=1}^{n} x_i \times \epsilon_i, x_l, x_h\} \tag{2.39}$$

The computation method for a variable becomes then a two step process:

- compute the symmetrical interval from the affine from $\widehat{x}$

- find the bounds of the results $[x]$

In addition they propose a new method for the linearization of non-affine operations that they call the minvolume approximation which reduces the error. Every non-affine binary function is transformed into the following form:

$$\widehat{z} = A\widehat{x} + B\widehat{y} + C + D\epsilon \tag{2.40}$$

where ϵ is the new error term and A, B, C, D are constants that are determined for the least error.

The last problem they try to solve is to provide a probabilistic foundation for the dynamic variation of a variable. As opposed to finding maximal theoretical bounds, the goal of their approach is to obtain tighter results with a certain probability for the number of times the magnitude will be outside of the predicted interval. The probabilistic nature of a variable $\{\widehat{x}, [x]\}$ comes from the randomness of the error symbols ϵ_i. Supposing that all noise terms are independent and identically distributed random variables with uniform distributions in [-1,+1] and using the Central Limit Theorem, the probability distribution of x is shown to converge to a normal distribution if the number of noise terms, N is large enough. As a result, the range of a variable can be computed for a chosen confidence level, p:

$$[\underline{x_p}, \overline{x_p}] = x_0 + [-\sigma_x\Phi^{-1}(p), \sigma_x\Phi^{-1}(p)] \tag{2.41}$$

where Φ is the normal cumulative density function and σ_x is the standard deviation of x.

These values take the form of new enforced bounds for the range of a variable and provide tighter intervals compared to the deterministic range obtained using the AA method.

However, in DSP applications, where the delay operations are very frequent, it is essential to capture the temporal correlation in order to obtain thigh range intervals and the method doesn't provide a way to track the temporal correlation of the data.

Extreme Value Theory Method

The Extreme Value Theory is a statistical analysis branch concerned with the extreme deviations from the mean of the probability density function. Its purpose is to give theoretical description of the distribution of extreme values that can be applied to model the probability and magnitude of rare events. It has been shown that the maxima and minima of a collection of independent and identically distributed random variables converge in distribution to the generalized extreme value (GEV) distribution.

The Extreme Value Theory has been applied to the range estimation problem in [10, 56, 57, 81]. Between the 3 families of distributions that compose the GEV (Gumbel, Fréchet and Weibull), the Gumbel [27] distribution (or the type I extreme value distribution), is used in this case. Its probability distribution has the following form:

$$f(x) = \frac{1}{\beta} e^{\frac{-(x-\mu)}{\beta}} e^{-e^{\frac{-(x-\mu)}{\beta}}} \tag{2.42}$$

where $\beta = \frac{s\sqrt{6}}{\pi}$ is the scale parameter, $\mu = \overline{x} - \beta\gamma$ is the location parameter, $\overline{x}$ is the mean, s is the standard deviation and $\gamma \approx 0.5772$ is Euler constant.

The method is based on lightweight simulations for statistical data analysis that provides theoretical probabilities for an overflow event. The probability of overflow is defined as the probability that the value of a variable exceeds its assigned range. As a consequence of the fact that the distribution of rare events has a infinite support, there is always a non-zero probability of overflow. The method provides the possibility to reduce this probability to small values, consistent with the application needs.

N sets of random samples are generated as inputs for the program. After simulating N times the program, N minima and N maxima are extracted for each variable. Using the obtained results, the parameters of the Gumbel distribution are estimated.

The user specifies the in-range probability for every variable of the application, which will give the maximal and minimal bounds.

$$\begin{aligned} P_r &= P(X \leq x) = e^{-e^{\frac{-(x-\mu)}{\beta}}} \\ x_{max} &= \mu - \beta ln(ln(1/P_r)) \end{aligned} \tag{2.43}$$

The larger the number of samples N is, the more accurate the statistical analysis becomes. However the number of samples that should be provided for an application is determined empirically. In [56] the authors find that 650 samples are sufficient for reliable results for all applications while in [57] the number is raised to 8000 input samples and in [81] the number of samples that is used varies from 300 to 10000.

Another problem that may arise if the sample size is not large enough, is that not all the possible execution traces in the program are covered. In [57] the problem is treated using an unique number that identifies every variable in every path in the internal representation. When a variable has no value assigned to it, the estimation will not be done and the default bit-width will be left.

On the positive side, the method can be applied to any kind of system and experimental results show that this method provides good results, outperforming AA based methods in range estimation and area reduction especially for non-linear applications [81].

Stochastic method

A new approach is presented in [78, 79] for dealing with the range estimation problem that takes advantage of both the random and temporal dimensions that characterize the uncertainty of data in signal processing applications. The input of the system is considered to be a random process that varies in time. As a consequence all the variables in the system become also random processes. The method is based on a stochastic discretization of the input process in terms of random variables using the Karhunen-Loeve̖ Expansion (KLE) and the Polynomial Chaos Expansion (PCE). As a result of the solid stochastical foundation of the KLE and PCE, the method can capture the temporal and spatial correlation of the signals.

As opposed to all the previous range representations, the KLE is a complete statistical description of the input process $x[n]$ that can be used to determine the statistical moments or the entire probability distribution. Using the superposition property of the LTI systems, the authors showed how it is possible to determine the corresponding KLE description of the output using a limited number of simulations.

For non-linear systems, the superposition cannot be applied anymore, so they proposed the use of the PCE instead. With the help of a projection method, the authors show how the PCE of the input is obtained from the corresponding KLE representation. Introducing a PCE arithmetic, the variability of the input can be statically propagated through the data-flow graph of the application even in non-linear systems. At the end, the PCE representation for all the variables is obtained and their statistics can be derived from there.

Furthermore, the authors propose a wordlength optimization criteria under SNR constraints. However, when the overflows occur in the middle of the computation path, this evaluation may become inaccurate. It is thus not obvious how the number of bits for the integer part wordlength can be computed directly using their method.

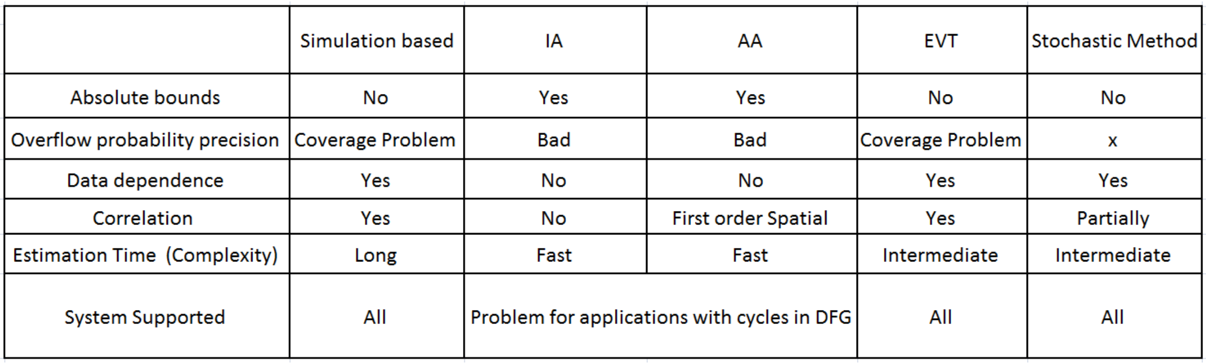
Conclusion

In order to realize a comparison between the methods that have been presented, we will analyze them in terms of the accuracy of the estimation, the time of the evaluation and the types of system that are supported. Furthermore, the precision of the estimation is analyzed using 4 evaluation criterias. We first examine if the absence of overflows is guaranteed, and if the minimal and maximal bounds are absolute or not. If the absence of overflows is not ensured, the precision of the overflow probability estimation is analyzed. The next criteria is the data dependence of the results and finally we examine if the estimation method takes into account the correlation of the data.

A summary of the comparison is presented in Figure 2.10. It is to be noticed that only the IA and the AA are not data dependent and provide absolute bounds. All the other methods provide local bounds and their accuracy is limited by the amount of input data that is provided. As the stochastic method uses the SQNR of the application for the range determination, the overflow probability is not directly determined.

The advantage of the AI and AA methods is that the evaluation time needed is shorter compared to the other methods. The simulation based method has an estimation time that can be extremely long that can become prohibitive while the stochastic and the EVT methods have an intermediate evaluation time that can be accepted in practice.

All the methods that have been presented here can theoretically be applied to any type of system. However, the IA and the AA may not converge to a finite value for applications that have cycles in their DFGs.



	Simulation based	IA	AA	EVT	Stochastic Method
Absolute bounds	No	Yes	Yes	No	No
Overflow probability precision	Coverage Problem	Bad	Bad	Coverage Problem	x
Data dependence	Yes	No	No	Yes	Yes
Correlation	Yes	No	First order Spatial	Yes	Partially
Estimation Time (Complexity)	Long	Fast	Fast	Intermediate	Intermediate
System Supported	All	Problem for applications with cycles in DFG		All	All

Figure 2.10: Estimation methods comparison

2.5.2 Numerical Accuracy analysis

The fractional part wordlength of the fixed-point data types is found by evaluating the quantization noise effects based on a compromise between the accuracy needed and the circuit cost. The evaluation of the computational accuracy can be made using several error metrics. Most of the times, the SQNR is chosen as the precision criteria. It guarantees that the power of the quantization noise does not exceed a certain threshold compared to the signal power. The method is especially attractive in signal processing applications, where a minimal difference between the useful signal and the level of the noise is desired.

One of the alternatives is to chose a maximal quantization error bound. As a result the evaluation of the computational accuracy is made using the interval of possible values that the error can take $e \in [e_{min}, e_{max}]$. Its advantage is that it ensures an absolute maximal value for the quantization noise which cannot be made using the SQNR criteria.

Simulation-based methods

Simulation based methods [32, 34, 36] evaluate the output of a bit-true fixed-point model of the system to random inputs. The results that are obtained are compared to the floating-point simulation, which is considered to be a reference model (as the computational error that is introduced by the floating-point representations is sufficiently small for most applications). The power of the quantization noise is directly obtained from the second order moment of the difference between the two. A new simulation has to be done for each different numerical accuracy evaluation. The method can provide good results but requires a long time in order to guarantee the accuracy. The approach is presented in Figure (3.14).

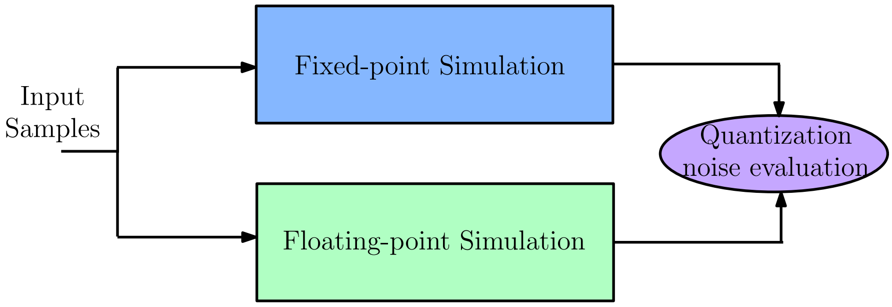


Figure 2.11: Computing the range from the PDF

The simulation of the fixed-point implementation requires an emulator of the fixed-point arithmetic. In [34] the gFix type is introduced by the means of the C++ operator overloading. However, all the mechanism that is constructed is very heavy, and the execution time is largely increased compared to a classic floating-point simulation. An optimization, called pFix type is presented in [36]. It uses the mantissa of the floating-point data for the representation of the fixed-point variables. As a result, the maximal wordlength is limited by the size of the mantissa (53 bits for double precision). The simulation time is greatly reduced compared to the previous version. However, it still remains largely superior to the floating-point execution time. As an example, for a 4^{th} order IIR filter, the simulation time for the fixed-point implementation is 7.5 times larger than the floating-point version.

Another approach has been adopted in [32]. The fixed-point variables are encoded using the integer data types with the purpose of diminuating the simulation time. The method is based on the optimization of the data alignment before arithmetical operations and the implementation of the quantification and overflow operations. FRIDGE tool [32] can reduce the execution time compared to the method based on the operator overload. Nonetheless, the simulation time remains 3.6 times greater than the floating-point.

Affine Arithmetic

In [18, 19, 43] a method based on the affine arithmetic was proposed. The evaluation of the accuracy is made using the absolute quantization error. Based on the fact that the quantization noise introduced by a rounding or truncation operation is bounded (Section 2.3.2.1), the range of the error can be further propagated using the affine arithmetic described in the previous Section. The method can be applied to the analysis of the precision in both the fixed-point [18] and floating-point [19] systems.

As it has been shown, the problem with the AA is the linearization for non-affine operations and the relative poor treatment of the correlation.

Perturbation Method

An approach based on the perturbation theory was presented in [62, 63]. The quantization noise is modeled as a small deviation from the infinite precision signal. The perturbation of the operands of arithmetic operations generates a perturbation of the result. The first and second-order statistics of the output noise can thus be computed. Considering a function with n input variables (x_i) and n associated noise terms (ϵ_i):

$$y = f_t(x_1, x_2, \cdots, x_n, \epsilon_1, \epsilon_2, \cdots, \epsilon_n) \tag{2.44}$$

The result of the fixed-point computation, y_{FP} is computed using a Taylor expansion at the second order:

$$\begin{gathered} f_t(x_1, x_2, \cdots, x_n, \epsilon_1, \epsilon_2, \cdots, \epsilon_n) = f_t(x_1, x_2, \cdots, x_n, 0, 0, \cdots, 0)+ \\ + \sum_{i=1}^{n} \epsilon_i \frac{df_t}{d\epsilon_i} + \sum_{i,j=1}^{n} \epsilon_i \epsilon_j \frac{df_t}{d\epsilon_i d\epsilon_j} \end{gathered} \tag{2.45}$$

The power of the quantization noise can then be determined as:

$$P_e = \overline{\mu}^t B \overline{\mu} + \sum_i C_i 2^{-2n_i} \tag{2.46}$$

where n_i is the number of bits used for the fractional part wordlength, C_i is a constant, μ is a vector that contains the expected values of the noise and B is a N_e size matrix (with N_e representing the number of noise sources).

However, the propagation of the noise requires a statistical evaluation through simulation in order to compute the terms B and C_i. The number of simulations that has to be done is proportional to the number of noise sources N_e^2 of the system. As a result, the computation time can become prohibitive if the number of noise sources is large.

Impulse response based method

In [46, 48, 49] a method based on the transfer function was presented for the case of LTI systems. The approach is based on the automatic determination of the transfer function from the signal flow graph (SFG) of the application and on the quantization noise model. As a result, it can provide the power of the output quantization noise.

Considering a system with N_e inputs $x_i[n]$ and one output $y[n]$, and let h_i be the impulse response from the input $x_i[n]$ to $y[n]$. The output can then be determined using the equation (2.47).

$$y[n] = \sum_{i=0}^{N_e-1} h_i * x_i[n] \tag{2.47}$$

The quantization of each input produces a noise source $b_i(n)$. Each internal operation that generates an elimination of bits (through rounding or truncation) introduces an additional noise term $b_{g_j}[n]$ with an associated impulse response h_{g_j}. As a result, the output quantization noise has the following expression:

$$b_y[n] = \sum_{i=0}^{N_e-1} h_i * b_i[n] + \sum_{j=0}^{N_g-1} h_{g_j} * b_{g_j}[n] \tag{2.48}$$

Using the quantization noise model, the power of the output noise can be computed as in equation (2.49), where μ_{b_I} and $\sigma^2_{b_i}$ represent the mean and the variance of the noise, and $H_i(e^{j\Omega})$ the corresponding transfer function.

$$P_{b_y} = \sum_{i=0}^{N_e+N_g} \sigma^2_{b_i} \frac{1}{2\pi} \int_{-\pi}^{\pi} |H_i(e^{j\Omega})|^2 d\Omega + (\mu_{b_i} H_i(1))^2 \tag{2.49}$$

2.6 Conclusion

This part introduced the floating-point and fixed-point representations and presented the finite wordlength paradigm. The floating-point to fixed-point conversion is transformed into an optimization process where the minimal number of bits for the representation of the integer and the fractional parts is determined under performance constraints. The fractional part wordlength determines the numerical accuracy of the application, while the integer bit-width ensures the dynamic range to avoid the appearance of overflows. A review of the existing methods for both the range estimation and the precision analysis has been made.

In order to solve the problem related to the overestimation of the dynamic range, the objective of this thesis is to develop a probabilistic framework for the optimization of the integer part wordlength with a constraint on the probability of overflow. Our approach for the range analysis is described in Chapter 3. In Chapter 4 a method for the range estimation in LTI systems is presented. Chapter 5 extends the analysis to all types of system.

Chapter 3

Stochastic Approach for Dynamic Range Estimation

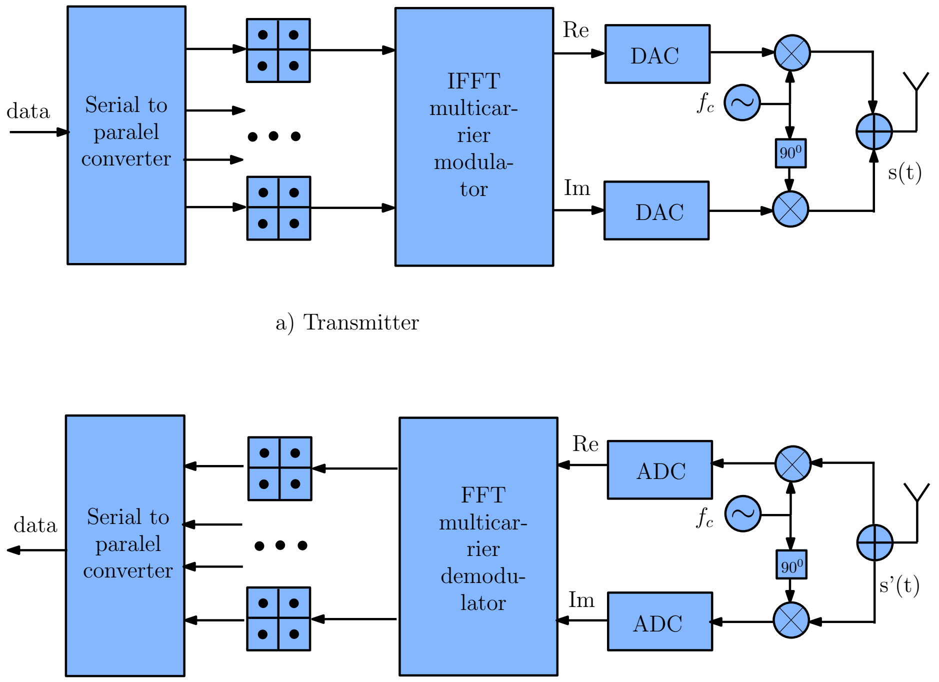
In this Chapter, the range evaluation problem is addressed. An orthogonal frequency-division multiplexing (OFDM) transmitter is presented and the datapath wordlength optimization problem is analyzed. It is a real application example that proves the interest of accepting overflows in order to realize a trade-off between the cost of the implementation and the performance of the system.

Secondly, a stochastic approach for the range estimation is proposed where the interval of variation is determined with a constraint on the overflow probability. For applications that can accept occasional overflows, the integer part wordlength is optimized without covering the entire theoretical dynamic range with the purpose of reducing the implementation cost.

3.1 Test Case Analysis - An OFDM Transmitter

The OFDM is a multi-carrier modulation scheme applied in a wide range of applications, such as digital television, wireless communications or broadband internet access, which became one of the most frequent communication technologies for high data rate transmissions. It is an efficient method for transmitting data over frequency-selective fading channels as the channel division makes it possible to avoid difficult equalization schemes at the receiver.

It is a modulation method that divides the entire frequency channel into many narrow band flat fading orthogonal sub-channels (or sub-carriers). An overview of the OFDM modulation structure can be seen in Figure 3.1. The serial bitstream is first separated into N different sub-carriers with a serial to parallel converter. Each channel is then independently modulated with a traditional modulation scheme (quadrature amplitude modulation or phase-shift keying). The multicarrier modulation is realized through the means of a complexInverse Fast Fourier Transform (IFFT). The real and the imaginary parts are then transformed into analog signals by the digital-to-analog converters (DAC) and the transmitted signal $s(t)$ is obtained.


data
Serial to paralel converter
IFFT multicarrier modulator
Re
Im
DAC
DAC
f_c
90^0
s(t)
data
Serial to paralel converter
FFT multicarrier demodulator
Re
Im
ADC
ADC
f_c
90^0
s'(t)

a) Transmitter

b) Receiver

Figure 3.1: OFDM modulation scheme

The receiver realizes the inverse operation. First, a modulated signal $s'(t)$ is transformed into its baseband correspondent and two digital signals are obtained using analog-to-digital converters (ADC). Using a complex Fast Fourier Transform (FFT), N parallel sub-carriers are obtained in the frequency domain. They are further demodulated and transformed into N binary streams of data. The final bitstream is obtained with a parallel-to-serial converter.

3.1.1 Application Description

The real example that has been chosen as a test case is the modulator of an OFDM transceiver for the WirelessHD standard. This is a technology for multi-gigabit wireless communication at distances of up to 10 meters for consumer electronics products (wireless audio, video and data streaming). The first implementation is designed for data rates of up to 3.0 Gbit/s, but the specification supports a theoretical throughput of 28 Gbit/s. The WirelessHD uses a 7 GHz channel in the 60 GHz radio band.

The standard is based on an OFDM modulation with $N = 512$ subcarriers, each one being modulated with a QPSK or 16QAM scheme. The sampling rate of of the application is 2.538 Gsamples/s.

The development of the entire transceiver has been done using High-level synthesis tools. However, from the entire application, our focus is only on the datapath optimization for the digital signal processing part of the transmitter, described in Figure 3.2.

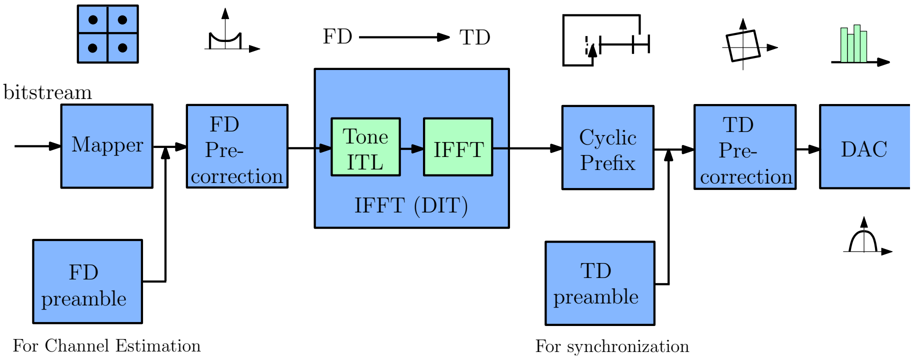


Figure 3.2: Digital Signal Processing Modulator

The main block of the modulator is formed by the 512-point IFFT. From the total of 512 subcarriers, only 336 subcarriers are actually used for the transmission.The Mapper generates the QPSK or 16QAM constellations for each sub-carrier. The number of bits that are transmitted for each OFDM symbol depends on the modulation scheme that is used for the sub-channels:

- QPSK modulation: 672 bits/OFDM symbol
- 16QAM modulation: 1344 bits/OFDM symbol

A frequency domain (FD) preamble is inserted for the channel estimation at the receiver and a time domain (TD) preamble is introduced for synchronization. In addition, a cyclic prefix with a length of 64 is used in order to avoid the intersymbol interference(ISI). The non-linear effects of the DAC, like the I/Q imbalance and the non-flat frequency behaviour are corrected using the frequency and time domain pre-correction blocks.

3.1.2 Peak-to-average power ratio problem

One of the major problems in OFDM communication systems is the high peak-to-average power ratio (PAPR) of the transmitted signal. This means that the peak values that appear are much larger than the mean. Hence, so as to avoid clipping the signal, enough bits need to be provided to cover the entire dynamic in the digital part while in the analog parts linear amplifiers that work linearly on a large range are required. Otherwise the signal will be clipped whenever the value will exceed a certain threshold causing distortions and out-of-band radiation that will degrade the overall bit-error rate performance of the system.

The peak-to-average power ratio is defined as:

$$PAPR = \frac{[x(t) * x(t)]}{E[x(t) * x(t)]} \tag{3.1}$$

where $*$ represents the conjugate

Considering an OFDM signal, consisting of N subcarriers, each symbol being modulated using an M QAM modulation scheme, it can be shown [30] that the maximum PAPR will be:

$$PAPR_{max} = 3N\frac{\sqrt{M}-1}{\sqrt{M}+1} \tag{3.2}$$

However, the probability that this event will occur in practice is very low [55]:

$$P_{PAPR_{max}} = \frac{1}{M^{N-2}} \tag{3.3}$$

Even though PAPR reduction techniques are used to modify the properties of the signal, practical values of the PAPR still remain high. So using the traditional methods for range analysis that guarantee that overflow never occur in practice imply the use of a large integer part wordlength and the implementation cost is largely increased. As the extreme values will rarely arise in practice, an important part of the dynamic variation is almost never used. It is then possible to optimize the hardware implementation without covering all the theoretical range using fewer bits for the integer part representation. A statistical method for the dynamic range determination should be applied, where an occasional overflow is authorized if the overall system performance is still guaranteed in order to reduce the area and power consumption of the application and to decrease the critical path delay.

3.1.3 Overflow Effects

Bit-error-rate (BER) analysis

If the integer part wordlength doesn't cover the entire theoretical dynamic range, the impact of overflows on the system performance should be analyzed. So as to evaluate the computational degradation that is introduced, the entire OFDM modulation scheme must be taken into account. The computational precision is translated into a decoding error and a decrease of the BER of the application.

The amplitude of the modulated signal at the *Mapper* output (with a QPSK or 16QAM scheme) is a parameter of the modulator (Figure 3.2) that can be modified.

Increasing or decreasing its magnitude modifies the range of the signal through the IFFT. As it has been shown in Section 2.3.2.1, the SQNR of the application is linearly dependent on the signal amplitude. As a consequence when the amplitude of the Mapper is increased, the SQNR is improved. However, the wordlength for the datapath is limited by the cost of the implementation so the dynamic range that can be represented by the fixed-point data types is limited. As a result, the increase of the amplitude of the Mapper causes the appearance of overflows because the application has a high PAPR.

The phenomenon has been observed in simulation and is illustrated in Figure 3.3. The bit-width of the datapath of the IFFT has been set to 12 bits. Considering an integer representation, the maximal and minimal values that can be represented with 12 bits are $x_{min} = -2048$ and $x_{max} = 2047$. For the 16QAM and QPSK modulation schemes, the absolute maximal amplitude of the input signal is varied from 400 to 1000. The corresponding probability of overflow in the IFFT datapath is determined in simulation. For low magnitudes of the signal, the absence of overflows is ensured. As the amplitude increases, the number of exceedings starts to grow and is quickly becoming very large after a certain threshold.

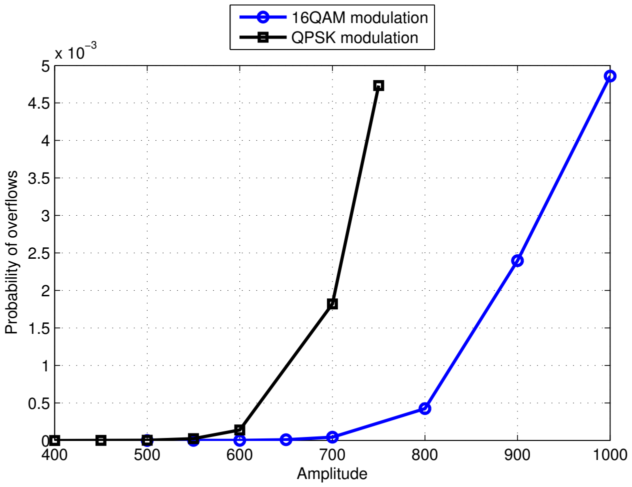


Figure 3.3: Number of overflows variation with the amplitude

The SQNR variation with the amplitude is also considered in Figure 3.4. As expected, the increase in the magnitude of the signal generates an improvement of the SQNR. The first point is the only one where no overflows were detected. As it can be seen, even though the occurrence of an overflow generates an important computational error, the global SQNR level of the application continues to grow when their number is limited. However, when their number becomes very high, a rapid degradation of the SQNR is noticed.

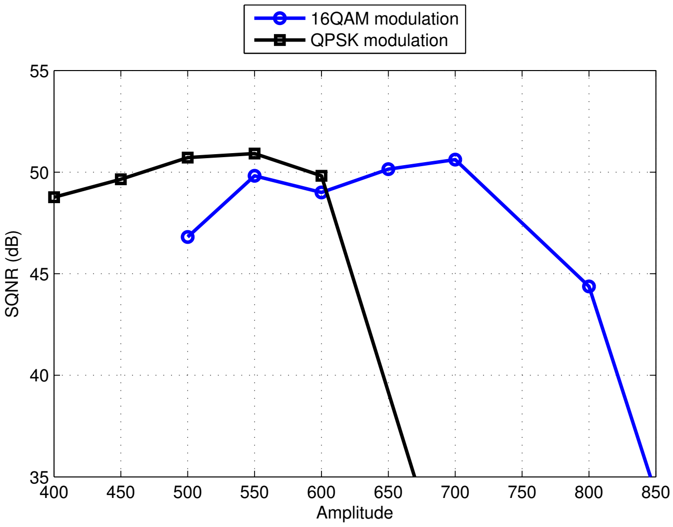


Figure 3.4: SQNR variation with the amplitude

Because the data transmission is realized over noisy channels, the use of a forward error correction (FEC) part is introduced into the communication scheme to correct the decoding errors and improve the overall BER. The channel decoder contains a Viterbi decoder followed by a Reed-Solomon decoder. The evaluation of the BER degradation caused by overflows is analyzed after the channel decoder. The test configuration of the OFDM modulator is presented in Figure 3.5. Several simulations tests are done, each one for a different value of the amplitude of the Mapper. Therefore the number of overflows that is observed increases and a different BER is obtained. Each simulation is done using an input frame of 10^6 bytes length.

The BER results obtained for the 16QAM and QPSK modulation in the presence of overflows are presented in Figure 3.6 and Figure 3.7 respectively.

As it can be seen, the overflow effects are more destructive for the 16QAM than

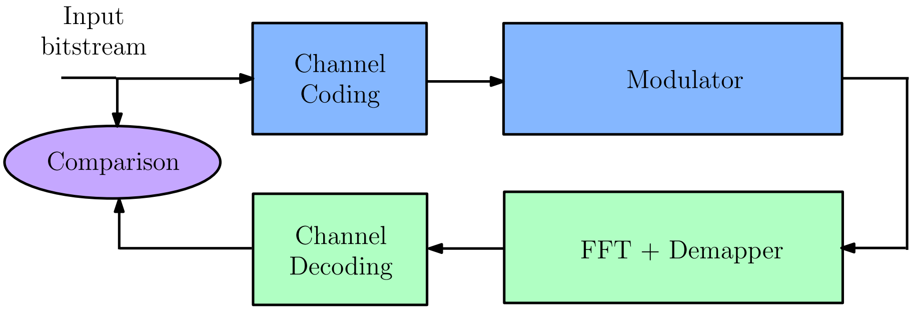


Figure 3.5: OFDM modem test diagram

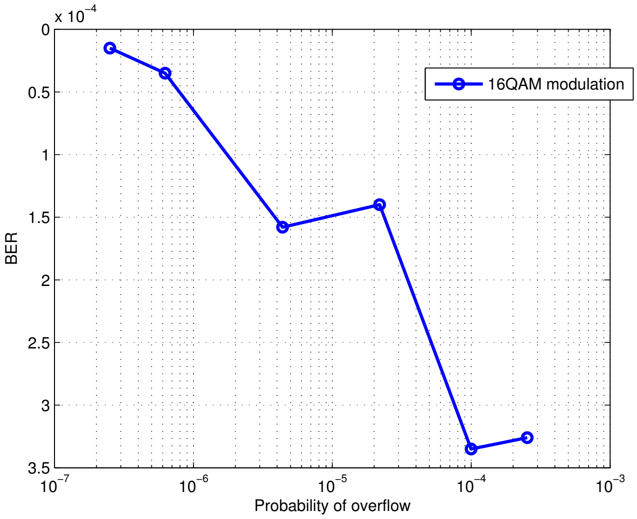


Figure 3.6: BER variation for 16QAM modulation

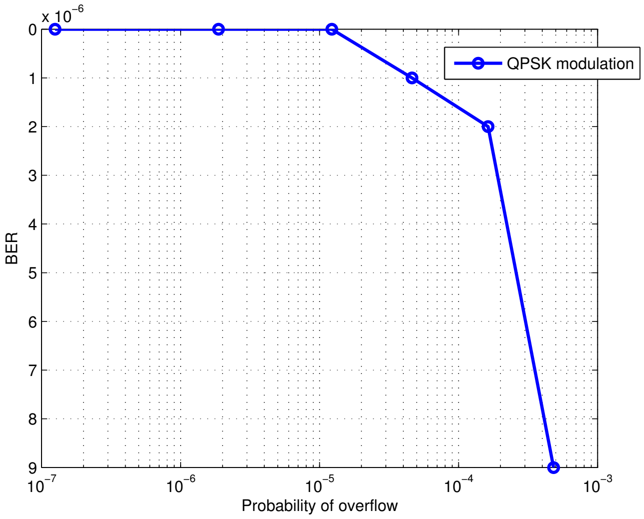


Figure 3.7: BER variation for QPSK modulation

for the QPSK modulation. However, for a desired BER efficiency of the implementation a corresponding overflow probability can be determined. Instead of trying to guarantee the absence of overflows it is possible to allow a limited number of exceedings that still ensure the performance constraint of the application.

When the wordlength of the datapath is limited by the implementation constraints and the signal has a large variation of its amplitude a trade-off between the fractional part and the integer part wordlength should be made. This is translated into a trade-off between an increased SQNR for the application and a probability of overflow.

Frequency spectrum analysis

The second aspect that is analyzed in the presence of overflows is the power spectral density of the transmitted signal. When an overflow occurs in the IFFT, the shape of the signal is changed and the frequency spectrum of the transmitted signal is modified, meaning that the emission mask is not respected anymore. The problem can be observed in Figure 3.8, where the floating-point signal at the output of the IFFT is compared with its fixed-point corresponding, when an overflow was produced.

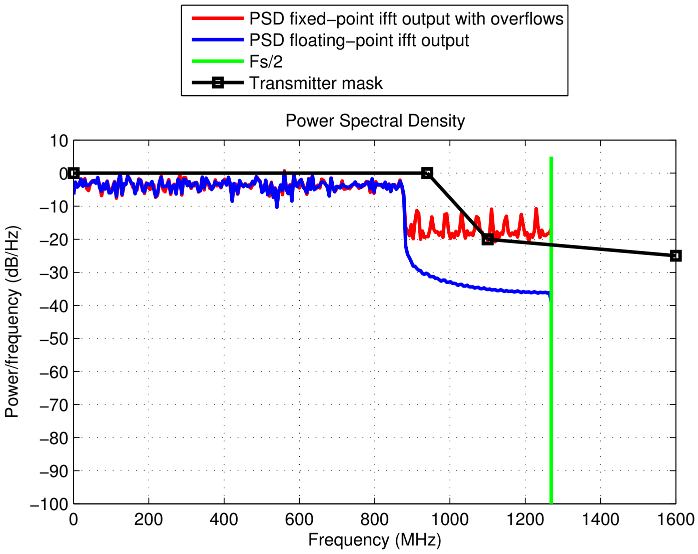


Figure 3.8: IFFT output PSD

Digital-to-analog converters have a non-flat frequency-response, meaning that the high frequencies will be attenuated as they approach $F_s/2$ (where F_s is the sampling frequency). The frequency-response is described by a $sin(x)/x$ (sinc(x)) roll-off that introduces at $F_s/2$ a -3.89 dB attenuation. The transmitted signal must be bandlimitted, so the analog signal is further passed through a reconstruction filter that eliminates the high frequencies.

To see if a relatively inexpensive reconstruction filter can remediate the overflow effects, we considered a 3^rd order Chebyshev low-pass reconstruction filter. The results are presented in Figure 3.9 and show that the low-pass filter is sufficient to guarantee the emission mask when overflows occur.

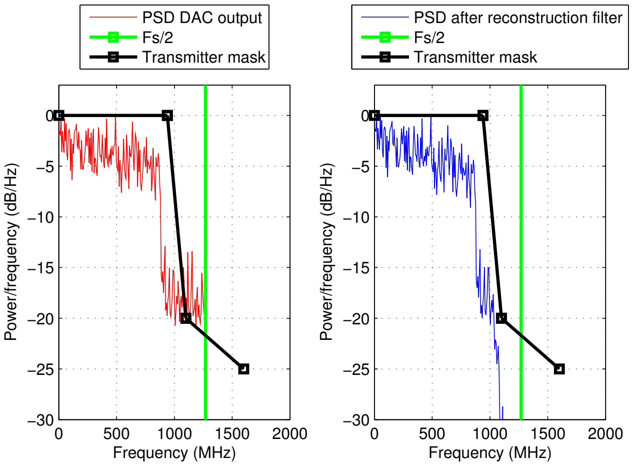


Figure 3.9: Transmitted signal PSD

3.2 Hardware Implementation

The implementation of the OFDM modulator has been realized using Mentor Graphics High-Level Synthesis tool CatapultC [50] and the corresponding fixed-point data types (*ac_fixed* and *ac_int*). The RTL code generated with CatapultC is further synthesized with Design Compiler (DC) [70] for a *65nm LP 1.2V* target technology at different frequencies. The most important part of the design is represented by the 512-point decimation-in-time IFFT. It accounts for 80% of the total area of the modulator and it is the heavily affected by the increase in the size of the datapath. The analysis of the hardware implementation of the IFFT provides an accurate model for the entire modulator, as a consequence only the results for the IFFT are presented in this section .

In order to observe the link between the performance (BER) and the cost (area and power consumption), several implementations have been realized for different datapath wordlengths. The size of the datapath has a direct connection to the performance of the system. The addition of every bit increases the SQNR of the transmitted signal which is translated into a higher BER for the overall application.

In practice, the implementation of the design has to ensure a certain throughput generally given by the standard. The advantage of the HLS tools (like CatapultC) is that the operation frequency can be set as a synthesis constraint in order to respect the desired throughput. It is then possible to compare the area and power consumption of a circuit for different synthesis frequencies very easily.

The results in terms of area after DC synthesis for different datapath wordlengths are presented in Table 3.1 for the 65nm LP technology. The importance of gaining even only 1 bit for the datapath of the IFFT can be easily seen and it becomes a crucial factor in obtaining cost-effective implementations, especially for the high frequencies that are needed for the WirelessHD standard.

The area comparison curves are plotted in Figure 3.10 and the corresponding power consumption comparison is presented in Figure 3.11. It is interesting to observe that there is a larger distance between the 11 and 12 bits implementations than between the 12 and 13 bits. This is an effect of the structure of the multipliers. As a result, passing from 11 to 12 bits has a higher impact on the circuit cost than passing from 12 to 13 bits. Depending on the frequency, the increase of the total area between the 11 and the 12 bits implementations is around 17-18% while the increase from 12 to 13 bits is only 3-5%. The same conclusion can be made about the power consumption, where a difference of about 29-32% is found between the 11 and 12 bits implementations whereas only 2-3% is observed for the 12 and 13 bits.

Another important observation should be made about the operating frequency. For a relatively constant area, the frequency is reduced by approximately a factor of

Datapath size	Frequency (MHz)	Area (μm^2)
10	200	560125
11	200	623589
12	200	735169
13	200	773834
14	200	856169
10	320	661459
11	320	755184
12	320	884238
13	320	912882
14	320	977072

Table 3.1: Area comparison for different wordlength sizes

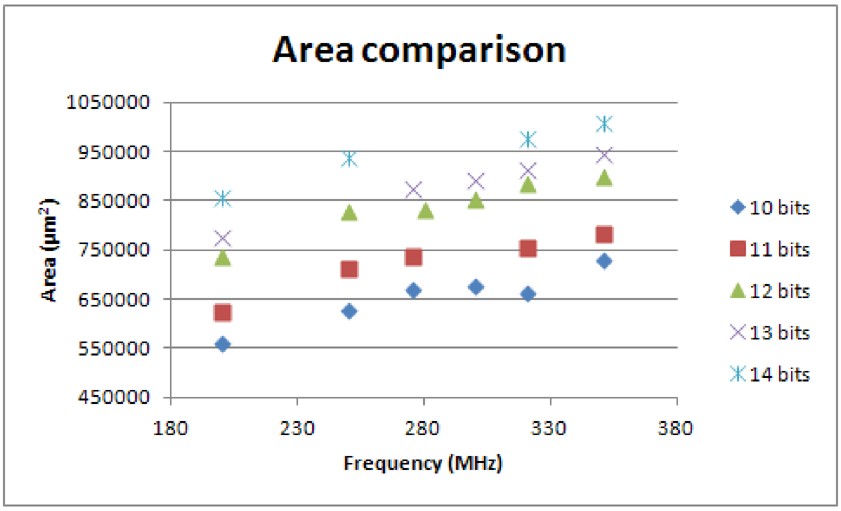


Figure 3.10: IFFT area comparison

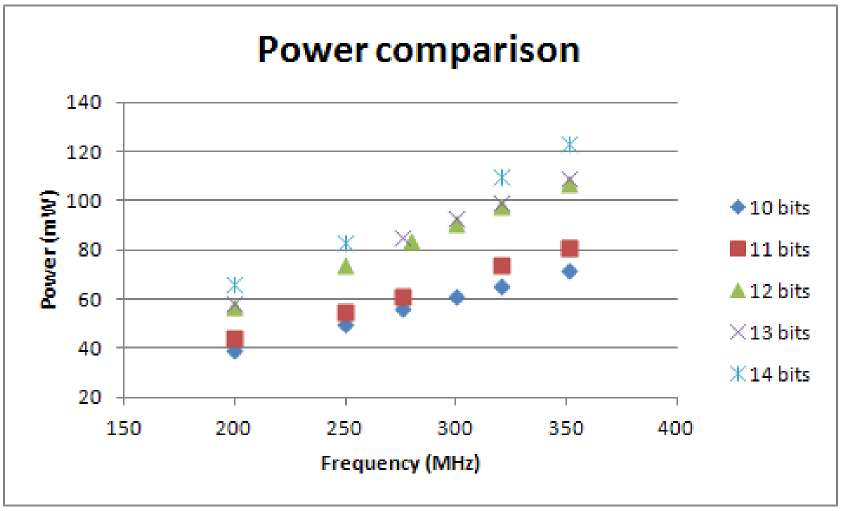


Figure 3.11: IFFT power consumption comparison

two when the size of the datapath is increased with two bits. The results are shown in Table Table 3.2.

Datapath size	Area (μm^2)	Frequency (MHz)
10	724109	375
12	735169	200
14	725277	100
11	663073	200
13	623589	100
11	755184	320
13	724844	162

Table 3.2: Frequency variation at approximately constant area size

The different comparisons that have been presented demonstrate the importance of the datapath optimization to obtain adequate hardware implementations that minimize the area and power consumption. In the following Section, a novel approach for the range determination is introduced, with the aim of reducing the integer part wordlength so that the cost can be decreased.

3.3 Proposed Method for the Range Analysis

As it has been shown in the previous Section, when the signal has a high variation of its amplitude throughout execution, dimensioning the wordlength of the datapath becomes an extremely difficult task. The classical range estimation methods determine absolute variation bounds (Figure 3.12).

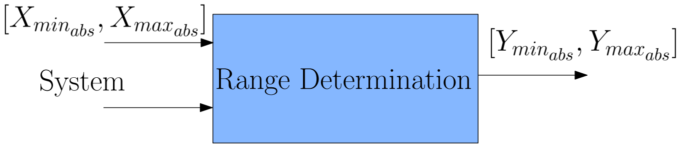

$[X_{min_{abs}}, X_{max_{abs}}]$
System
Range Determination
$[Y_{min_{abs}}, Y_{max_{abs}}]$

Figure 3.12: Classical Range Determination

If the entire theoretical range is ensured, the cost of the hardware implementation can be significantly increases. To comply with the high throughput demands of the application and at the same time obtain a cost effective implementation, the wordlength of the integer part can be reduced so that not the entire interval of variation is covered. As a consequence the occurrence of overflows is authorized with a constraint regarding their probability of appearance. Variables that have long tailed PDFs will be approximated with tight intervals that correspond to a desired coverage probability.

It becomes thus very important to estimate accurately the dynamic range and the probability of appearance of high peaks. Traditionally, this is a process that can be done using extensive simulations. However, this is an iterative process, that has to be done every time a parameter of the implementation has changed. Therefore this is a method that becomes time demanding and error prone.

To solve the problem, an analytical method should be developed. It is an optimization problem that can be separated into two parts. The first one corresponds to the determination of the dynamic range for a given overflow probability while the second is concerned with the analysis of the performance degradation generated by the overflows.

This thesis focuses only on the first part of the integer wordlength optimization. As a result, a probabilistic framework is developed for the determination of the variation interval that corresponds to a desired overflow probability.

3.3.1 Probabilistic description of the system

Instead of representing the variation of a signal like the classical analytic methods do, using only the maximal and minimal bounds $[X_{min}, X_{max}]$, our aim is to obtain a complete representation of the variability of the output of the system that incorporates its probabilistic behavior from a stochastic representation of the input. The range of a variable is thus represented by its PDF. This characterization of the variability is further propagated through the system, obtaining the corresponding representations for each variable in the system. An input-output view of a system can then be represented as in Figure 3.13.

Furthermore, we propose an integer wordlength optimization criteria based on

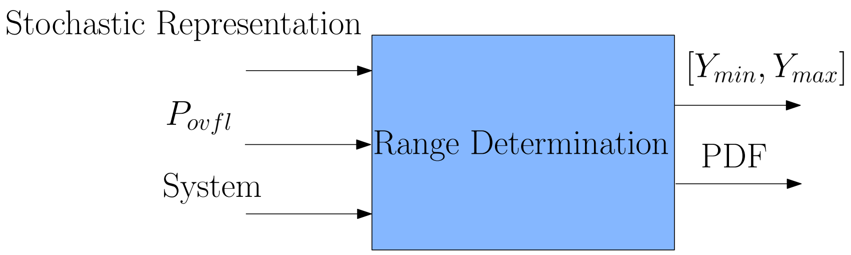


Figure 3.13: Probabilistic Range Determination

the overflow probability. The range for all variables is computed from the PDF with respect to a coverage probability. The probability that the values of a variable will exceed a certain threshold can be computed from the PDF as it is shown in equation (3.4).

$$P_{overflow} = \int_{D} p_Y(y)dy \tag{3.4}$$

where $D = \{y \mid y_{min} > y \text{ and } y < y_{max}\}$

The dynamic range is then determined from the PDF by the integration of its tails in order to correspond to a desired probability of overflow as it can be seen in Figure 3.14.

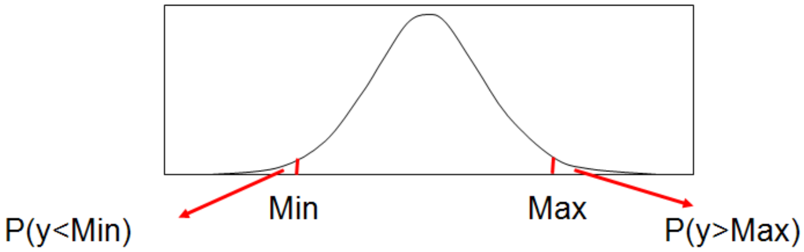


Figure 3.14: Computing the range from the PDF

With our approach, it becomes possible to realize a trade-off that can reduce the implementation cost depending on the application performance specifications. As opposed to the methods that provide fixed minimal and maximal limits and thus overdimension the system, we can determine appropriate intervals of variations by changing the allowed overflow probability. The situation is illustrated in Figure 3.15, where an implementation cost gain can be obtained by adapting the wordlength of the datapath to the bit-error-rate (BER) needed for the application.

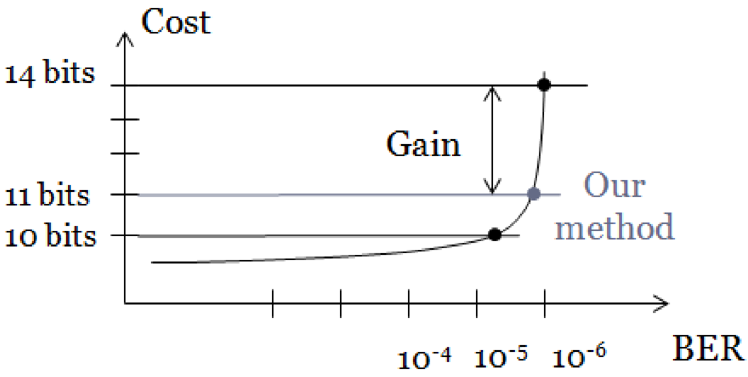


Figure 3.15: Cost-performance trade-off

3.3.2 Range determination methodology

Based on the new probabilistic approach for the variability analysis, a general methodology for the range determination can be seen in Figure 3.16.

The statistical description of the output is obtained by propagating the variability of the input through the system. As a result, the first part of the methodology relies on a stochastic discretization procedure that generates a representation of the PDF for each input variable x_i, denoted here by $\Gamma_i(x_i)$.

The application, originally described using a high-level language like C++, is transformed into a data-flow graph (DFG). The stochastic representation of the output, $\Gamma_i(y_i)$ is computed next, relying on the input model and the DFG.

Finally, the corresponding PDF of each variable y_i is estimated and the dynamic range, $[y_{min}, y_{max}]$ is determined according to an authorized probability of overflow that is given as a parameter.

3.4 Conclusion

In this Chapter a real application has been presented as a validation example for the acceptance of overflows in order to optimize the wordlength of the datapath and reduce the cost of the hardware implementation.

A statistical approach for the range evaluation was proposed, where the necessary

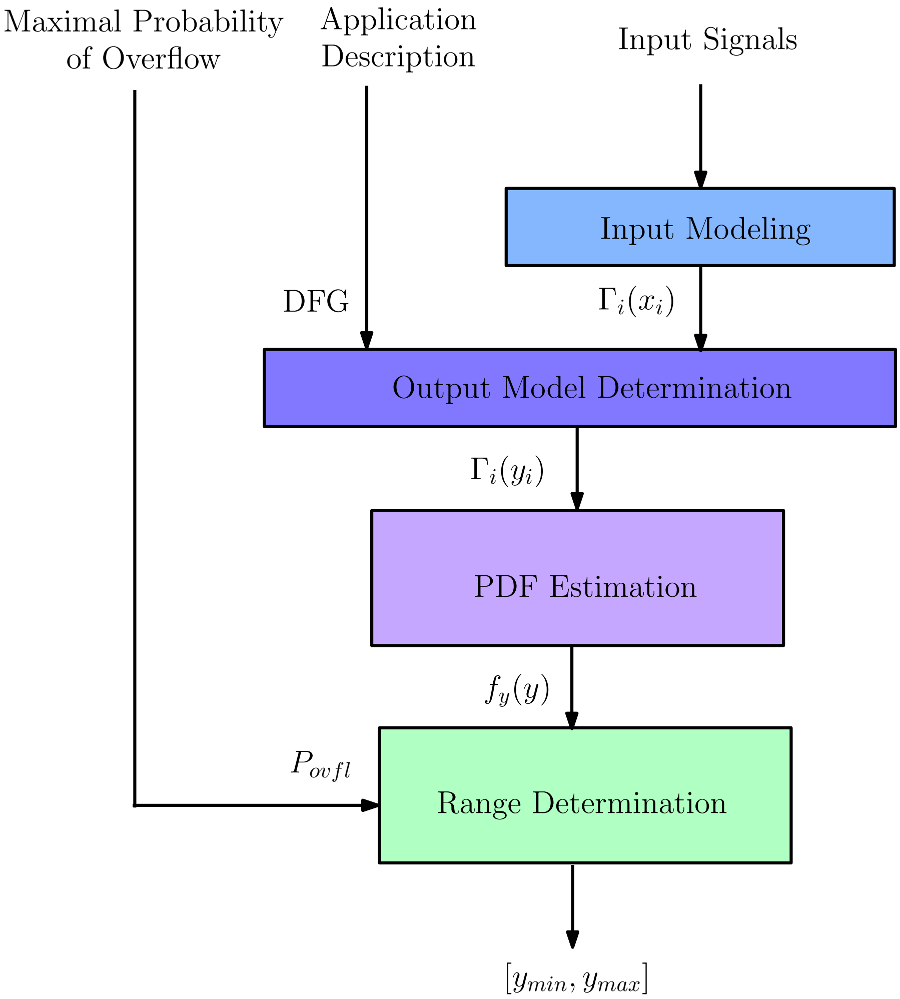


Figure 3.16: Probabilistic range determination methodology

number of bits for the integer part representation are computed with a constraint on the probability of overflow that is allowed.

Chapter 4

Karhunen Loève Expansion Method For Linear Time Invariant Systems

In this Chapter a method for the range evaluation of variables in LTI systems with respect to a corresponding overflow probability is presented. The procedure is based on the Karhunen-Loève Expansion as a means of representation for the variability of signals. Furthermore, we show that the quantization noise estimation can be realized using the same approach. The results obtained for several typical applications are presented.

4.1 The Krahunen-Loève Expansion

4.1.1 Introduction

Random Variable

Let (Ω, F, P) be the probability space with Ω the sample space, F an σ-algebra and P the probability measure. A real random variable is a function $X : (\Omega, F, P) \rightarrow D \subset \mathbb{R}$. For every outcome $\theta \in \Omega$, a real value $X(\theta)$ is assigned. If X has a discrete number of possible values $D = \{x_k,\ k \in \mathbb{N}\}$, the random variable is called discrete.

If the domain of values D is continuous, X is a continuous random variable. Any random variable X is defined through its cumulative distribution function (CDF) $F_X(x)$ as in equation (4.1):

$$\begin{aligned} F_X &: \mathbb{R} \to [0,1] \\ F_X(x) &= P\{X \leq x\} \end{aligned} \tag{4.1}$$

For a discrete random variable, the probability mass function is defined as:

$$P\{X = x_k\} = p_k \tag{4.2}$$

The CDF can then be determined as it follows:

$$F_X(x) = \sum_{x_k \leq x} p_k \tag{4.3}$$

In the case of continuous random variables, the probability density function (PDF) $f_X(x)$ is introduced:

$$f_X(x) = \frac{dF_X(x)}{dx} \tag{4.4}$$

The CDF becomes:

$$F_X(x) = \textstyle\int_{-\infty}^{x} f_X(y)dy \tag{4.5}$$

The probability between any two values of X can then be computed as in equation (4.6):

$$P\{a < X < b\} = \int_a^b f_X(x)dx, \ \forall a < b \tag{4.6}$$

with $\int_{-\infty}^{\infty} f_X(x)dx = 1$.

Moments

The expected value (mean) of a random variable is defined as:

$$\mu = E[X] = \int_{\mathbb{R}} x \, f_X(x)dx \tag{4.7}$$

The variance is:

$$\sigma^2 = E[(X - E[X])^2] = \int_{\mathbb{R}} (X - E[X])^2 f_X(x)dx \tag{4.8}$$

The n-th moment of X is:

$$E[X^n] = \int_{\mathbb{R}} x^n \; f_X(x)dx \tag{4.9}$$

The covariance of 2 random variables X, Y is a measure of the strength of the correlation between the two variables and is given by equation (4.10).

$$\begin{aligned} Cov(X,Y) &= E[(X - E[X])(Y - E[Y])] \\ &= \textstyle\int_{\mathbb{R}} \int_{\mathbb{R}} (X - E[X])(Y - E[Y]) f_{X,Y}(x,y)dxdy \end{aligned} \tag{4.10}$$

with $f_{X,Y}$ the joint probability distribution of the two random variables.

The correlation between two random variables X and Y is given by the correlation coefficient ρ. It is obtained by normalizing the covariance with the standard deviations σ_x and σ_y of each variable (4.11):

$$\rho = \frac{cov(X,Y)}{\sigma_x \, \sigma_y} \tag{4.11}$$

Random Vector

A random vector X is a function $X : (\Omega, F, P) \rightarrow D \subset \mathbb{R}^d$ where d is the size of the vector $X = (X_1, X_2, , X_d)^T$, and whose components are all random variables.

Random Process

A stochastic (random) process is mathematically described as a sequence of random variables indexed by a parameter t, $x(t, \theta) = \{x_t, \; t \in T\}$, defined on the probability space (Ω, F, P). When the set T is countable (e.g. $T = 0, 1, 2, \cdots$), $x(t, \theta)$ is called a discrete random process. Otherwise, if T is an interval (e.g. $T = [a, b] \in \mathbb{R}$), $x(t, \theta)$ is a continuous random process. Usually, the index t represents time, and then x_t represents the process at the time instant t.

For a fixed t_0, $x(t_0, \theta)$ is a random variable while for a fixed $\theta_0, x(t, \omega_0)$ represents a realization (or trajectory) of the process and a curve in the Hilbert space L^2.

A random process is called stationary if its statistics do not depend on the observation interval, meaning that its joint probability distribution is not modified by time shift operations.

The autocovariance of a stochastic process is defined as the covariance between its value at t_1 and its value at t_2:

$$C_{XX}(t_1, t_2) = Cov\big(x(t_1, \theta), x(t_2, \theta)\big) \tag{4.12}$$

4.1.2 Krahunen-Loève Expansion

Generally, random processes have an infinite dimension. In order to represent them in practice, a discretization procedure must be realized. Its purpose is to approximate the process as a combination of a finite set of random variables that is easier to manage. Several discretization techniques have been presented in the literature [40, 67]. Between them, the series expansion methods are the most widely used.

The Karhunen-Loève Expansion (KLE) [41] is a discretization procedure based on the covariance function of the input process. Consider a second order random process $x(t, \theta)$ with mean $m(t)$ and autocovariance function C_{XX}. It is then possible to represent the process using a spectral expansion of its covariance function, in a similar manner to a Fourier series representation, called the Karhunen-Loève Expansion:

$$x(t, \theta) = m(t) + \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(t)\eta_i(\theta) \tag{4.13}$$

where $\{\eta_i(\theta)\}$ is a set of uncorrelated, with zero mean and unit variance random variables. λ_i and ϕ_i are the eigenvalues and eigenfunctions of the covariance function C_{XX}, meaning that they are the solution to the homogeneous Fredholm integral equation of the second kind:

$$\int_T C_{XX}(t_1 t_2)\phi_i(t_1)dt_1 = \lambda_i \phi_i(t_2) \tag{4.14}$$

The main difficulty of the KLE is to compute the equation (4.14). For some particular cases, the eigenproblem can be computed analytically as it is described in [25]. However, in most of the practical cases a numerical solution based on the Cholesky decomposition or the QZ algorithm can be used instead.

In practice, if the mean and the covariance is not known analytically and only a number of realizations of the process are known, the unbiased estimators are computed using the following equation:

$$\begin{aligned} m(t) &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N} x_i \\ &\text{and} \\ C_{XX} &= \tfrac{1}{N-1}\textstyle\sum_{i=1}^{N}(x_i - m(t))^T(x_i - m(t)) \end{aligned} \tag{4.15}$$

where x_i is the i^{th} realization of the random process and N is the total number of realizations used in the estimation.

The random variables $\{\eta_i(\theta)\}$ are orthogonal and have a zero mean:

$$\langle \eta_i(\theta) \rangle = 0 \text{ and } \langle \eta_i(\theta)\eta_j(\theta) \rangle = \delta_{ij} \tag{4.16}$$

with the inner product defined as:

$$\langle \eta_i(\theta)\eta_j(\theta) \rangle = E[\eta_i(\theta)\eta_j(\theta)] \tag{4.17}$$

From the equation (4.13), an expression for each random variable can be determined:

$$\eta_i(\theta) = \frac{1}{\sqrt{\lambda_i}} \int_T (x(t,\theta) - m(t))\phi_i(t)dt \tag{4.18}$$

$\{\phi_i : D \to \Re\}$ is a set of deterministic functions of t and form a complete orthonormal basis in $L_2(D)$. The eigenvalues λ_i are all positive and describe the importance of the corresponding eigenfunction in the process. They can be arranged in a decreasing order. The decay of the eigenvalues depends on the smoothness of the covariance and on the correlation length (the decay increases with the correlation length).

For the specific case when the input process is Gaussian, the random variables $\eta_i(\theta)$ are all independent standard normal random variables. In the general case, however, they have an unknown distribution and are only uncorrelated.

In theory, the KLE representation has an infinite sum of random variables. However, in order to use the expansion in practice, only a finite approximate of the process $x(t,\theta)$ is used, meaning that the KLE is truncated after a certain order M:

$$x(t,\theta) \approx m(t) + \sum_{i=1}^{M} \sqrt{\lambda_i}\phi_i(t)\eta_i(\theta) \tag{4.19}$$

The KLE is a mean square convergent series for all finite second order random processes (processes with finite energy) and it can be shown that it is even optimal in the sense that it minimizes the truncation error for a fixed order M. In other words, there is no other series expansion that approximates better the random process with the same number of terms.

$$\begin{gathered} E\Big[\textstyle\int_D \Big(x(t,\theta) - \big(m(t) + \sum_{i=1}^{M} \sqrt{\lambda_i}\phi_i(t)\eta_i(\theta)\big)\Big)^2 dt\Big] = \\ = \textstyle\sum_{i>M} \lambda_i \to 0 \quad \text{as} \quad M \to \infty \end{gathered} \tag{4.20}$$

One way to compute M is by choosing a truncation error that is sufficiently close to zero:

$$e_{tr} = 1 - \frac{\sum_{i=1}^{M} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \tag{4.21}$$

where $\lambda_1, \lambda_2, \cdots, \lambda_M$ are the eigenvalues kept in the truncated expression from the total of N eigenvalues.

The value M is determined by the decay rate of λ_i. The closer the process is to a white noise, the more terms are needed in the expansion. At the other end, a random variable can be represented by a single term. It is clear that the KLE is more efficient for highly correlated random processes.

Because the random coefficients $\{\eta_i\}$ are uncorrelated, the variance of the KLE approximation of the process can be computed using the Bienaymé formula as the sum of the variances of each term:

$$\sigma^2_{x_{KLE}} = \sum_{i=1}^{M} \left(\sqrt{\lambda_i}\phi_i(t)\eta_i(\theta)\right)^2 = \sum_{i=1}^{M} \lambda_i \phi_i(t)^2 \tag{4.22}$$

given that the variance of $\{\eta_i\}$ is equal to one.

The error of the truncated variance can be thus determined by the following equation:

$$e_{\sigma^2} = \sigma^2_x - \sigma^2_{x_{KLE}} = \sigma^2_x - \sum_{i=1}^{M} \lambda_i \phi_i(t)^2 > 0 \tag{4.23}$$

because all the eigenvalues λ_i are positive.

This points out that the truncation of the KLE will always underestimate the variance of the process.

In conclusion, the KLE approximates a random process by a linear combination of (countable) deterministic functions (also called KL modes) $\{\phi_i\}$ with orthogonal (uncorrelated) random coefficients $\{\eta_i\}$ which represent the probabilistic content (the stochastic dimension).

4.2 Stochastic Modeling

In digital signal processing applications, many times the input signals have a correspondence to real physical processes that vary in time. The probability of overflow for a variable corresponds to the number of times the values of that variable exceed the allowed range during the execution time [0, T]. The situation is represented in Figure 4.1.

For stationary processes, the overflow probability can be estimated directly from the PDF integrating its tails. Reciprocally, for a chosen probability of overflow the

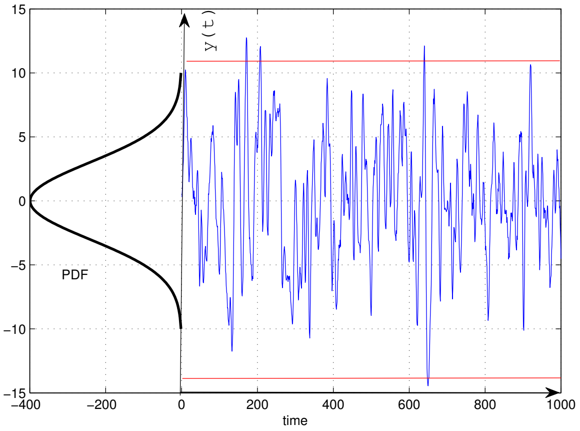


Figure 4.1: Overflow occurrence

corresponding maximal and minimal bounds can be determined. Therefore, the integer part wordlength optimization under a probability of overflow constraint can be realized using a probabilistic approach that characterizes the dynamic variability by associating a PDF to every variable in the design.

Digital signal processing algorithms often use delay operations. As a result, the samples of the process are computed at different time instants, as in the case of a FIR filter: $x(n)$, $x(n-1)$, $x(n-2), \cdots$. The values of the signal at a particular point in time are found to be more or less correlated with the values that proceed and succeed them. As a result, the statistical description (PDF) of the internal and output variables is dependent of the correlation structure of the input signal.

Consider the example of a random process with the PDF presented in Figure 4.2 and the covariance function given in Figure 4.3.

A comparison between the sum of: $x(n)+x(n)$, $x(n)+x(n-1)$, $x(n)+x(n-4)$ and $x(n)+x(n-10)$ is presented in Figure 4.4. Because the correlation between the samples varies with the delay, the PDF of the result is different each time. Even though the theoretical absolute minimal and maximal values obtained with the interval arithmetic are the same, their probability of occurrence changes. With our approach the range is determined from the PDF with respect to an overflow probability, so the obtained interval will be different in all of the four cases. The

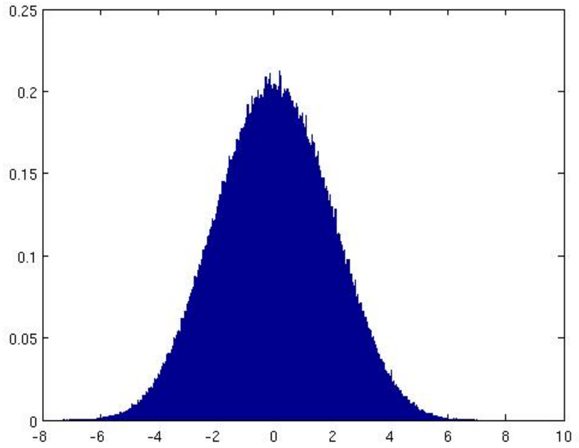


Figure 4.2: Input PDF

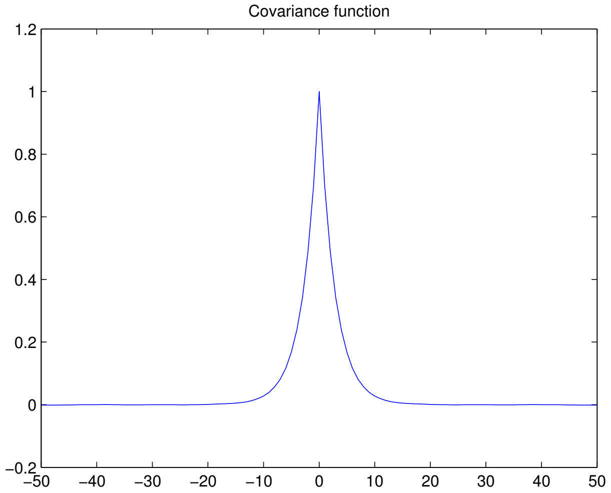


Figure 4.3: Input covariance function

importance of the temporal correlation between data is thus primordial in order to obtain an accurate range estimation. This is why the traditional methods like the interval arithmetic and the affine arithmetic are not adapted to this kind of situations.

In order to incorporate the temporal correlation the notion of random process becomes the mathematical model that is the most appropriate. The randomness of the input propagates through the system and the state variables and the output become also random processes. The problem of range estimation is thus equivalent to evaluating the response of a system modeled as a deterministic function with

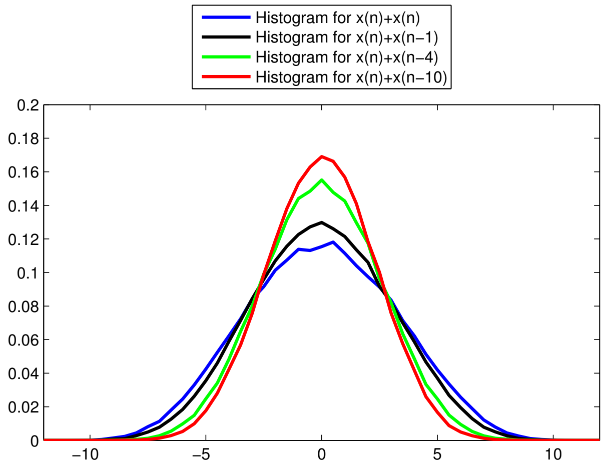


Figure 4.4: PDF of the sum of delayed samples

random inputs: $y(t) = f(x_1(t), x_2(t), \cdots, x_n(t))$.

The methodology can then be divided into 3 different parts:

1. Represent the input variability using the KLE discretization procedure
2. Propagate the uncertainty through the system and obtain the KLE representation for every variable
3. Range determination from the PDF and according to a probability of overflow

4.2.1 Input Representation

As a primary step, the variability of the input signal is represented by the means of the KLE:

$$x(t,\theta) \approx m(t) + \sum_{i=1}^{M} \sqrt{\lambda_i}\phi_i(t)\eta_i(\theta) \text{ with } t = 0, 1, 2, \cdots \tag{4.24}$$

Synthetic signal

In order to model the correlation of the input process the auto-regressive (AR) time series model is used. This method is frequently applied in DSP applications to model real physical process. It is a linear regression of the current value of the time series against its past values. The AR of order p is described by the following equation:

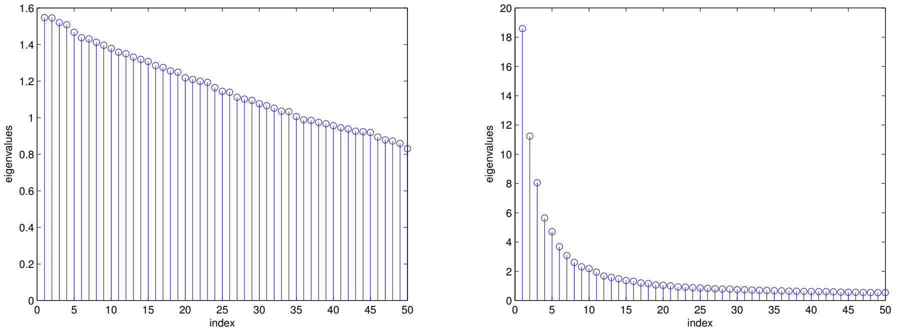
$$x_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_p x_{t-p} + \epsilon_t \tag{4.25}$$

where φ_i are the parameters of the model and ϵ_t is a white noise.

The AR of order 1 (AR(1)) is used as a test for the generation of the input process.

$$x_t = \varphi_1 x_{t-1} + \epsilon_t \tag{4.26}$$

The correlation of the process is modified by the parameter φ_1. The temporal length of analysis depends on the dimension of the system in order to obtain the steady state of the output. It will give the size of the covariance matrix C_{XX} and it is set here to 50 time points. The number of terms that should be kept in the expansion is given by the decay rate of the eigenvalues. This is a function of the correlation length, for the higher correlation the decay is steeper. As an example, the first 50 eigenvalues of the AR(1) process for two different cases: $\varphi_1 = 0.2$ and $\varphi_1 = 0.95$ are represented in decreasing order in Figure 4.5a and in Figure 4.5b) respectively. The values of the eigenvalues decrease much faster for the case when $\varphi_1 = 0.95$ as the correlation is more important.



(a) $\varphi_1 = 0.2$

(b) $\varphi_1 = 0.95$

Figure 4.5: Eigenvalues arranged in decreasing order

The KLE allows to treat the deterministic variable (t) and the random character (θ) of the input separately. The expression of the input at different time instants is given by the value of the eigenfunctions $\phi_i(t)$. The first 4 eigenfunctions for the AR process with $\varphi_1 = 0.95$ can be seen in Figure 4.6.

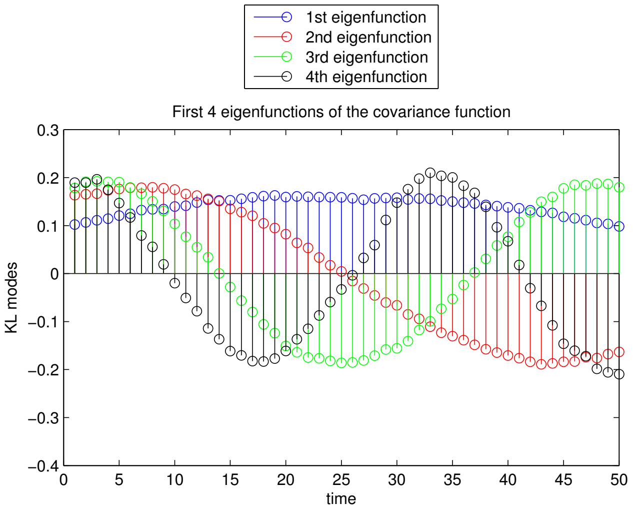
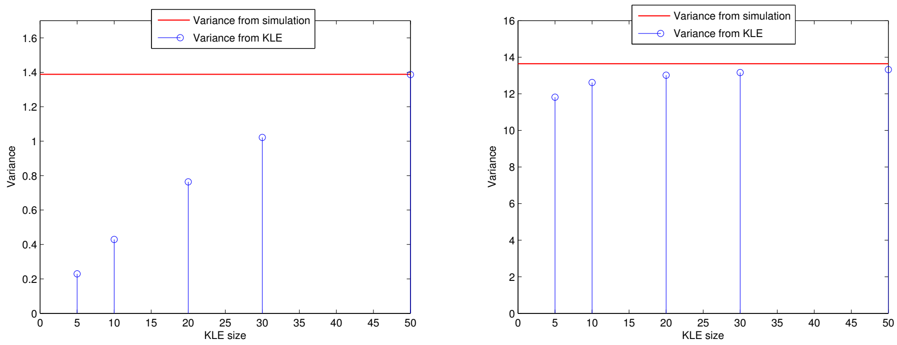


Figure 4.6: First 4 eigenfunctions

If the input has a low correlation, the truncation of the KLE will generate a loss in precision because every eigenvalue is important in the expansion. On the other hand, if the process is highly correlated, the complexity can be reduced as most of the energy is captured with only a few terms. The effects of truncation can be seen by computing the variance for different KLE sizes. The results are plotted in Figure 4.7. As it can be seen, the error is significantly greater for the low correlation process. In order to obtain the same accuracy an increased number of terms is needed in this case.

The effects of the KLE truncation can also be seen on the PDF estimation. The results are presented in Figure 4.8 and Figure 4.9. The PDF for the highly correlated process can be approximated with fewer terms.



(a) $\varphi_1 = 0.2$

(b) $\varphi_1 = 0.95$

Figure 4.7: Variance variation with the KLE size

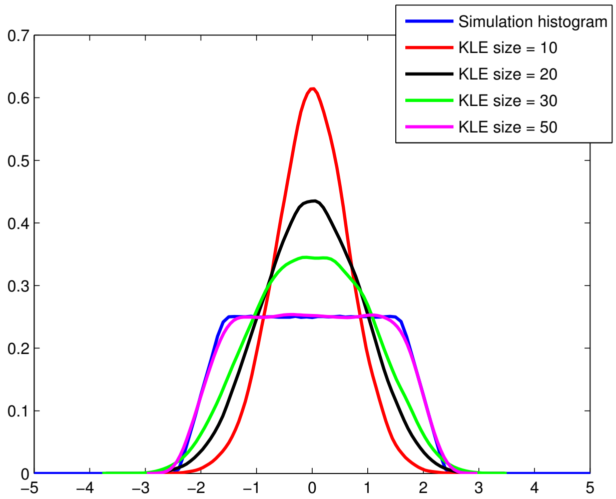


Figure 4.8: PDF estimation for $\varphi_1 = 0.2$ with different KLE sizes

4.2.2 Variability propagation in LTI systems

A system is called LTI if it satisfies the superposition property (4.27) and it is invariant to time shifts (4.28). Generally a system is LTI if it does not have any non-linear operations (e.g. multiplications between variables, divisions etc.). All the other systems will be non-linear.

The superposition property is defined as:

$$f(a_1x_1(t) + a_2x_2(t)) = a_1f(x_1(t)) + a_2f(x_2(t)) \tag{4.27}$$

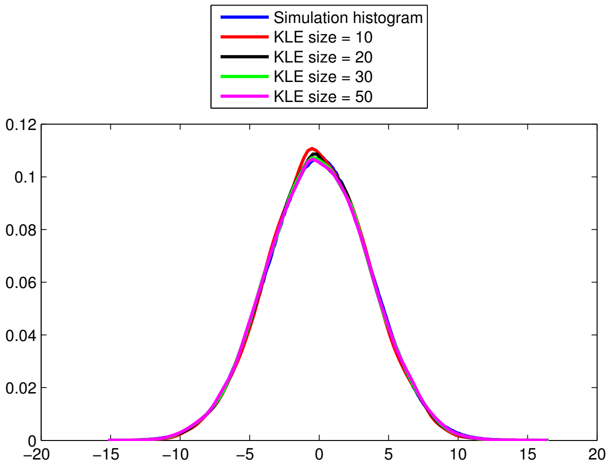


Figure 4.9: PDF estimation for $\varphi_1 = 0.95$ with different KLE sizes

A system is time invariant if:

$$f(x(t)) = y(t) \Rightarrow f(x(t-\tau)) = y(t-\tau) \tag{4.28}$$

In this Chapter, we only focus on LTI systems. We want to propagate the variability of the input (represented by the means of the KLE) through the system in order to obtain a representation of the variability for all the variables in the design.

The authors in [79] presented a method for the KLE propagation in the case of LTI systems based on the superposition property. They showed how the corresponding description of the output can be obtained using a limited number of simulations. Considering the case of a system that has one input $x(t,\theta))$ and one output $y(t,\theta))$, that is mathematically defined by a function $\mathbb{L}$:

$$\begin{aligned} & y(t,\theta) = \mathbb{L}\big(x(t,\theta)\big) = y_0(t) + \textstyle\sum_{i=1}^{M} y_i(t)\eta_i(\theta) \\ & \text{where } y_i(t) = \begin{cases} \mathbb{L}\big(m(t)\big), & \text{i = 0} \\ \mathbb{L}\big(\sqrt{\lambda_i}\phi_i(t)\big), & \text{i = \{ 1, ... , M \}} \end{cases} \end{aligned} \tag{4.29}$$

As a result, the KLE representation of the output can be computed using (M+1) simulations of the system.

In our work, we use the same KLE discretization approach to represent the variability of the input signal. However, we show that when the operands of linear operations are represented with KLEs, the results can be computed analytically and thus the need of simulation is completely removed.

A linear system is completely described by its impulse response. When the inputs are represented with KLEs, the output can be computed as the convolution of the system impulse response and its KLE representation:

$$y(t) = x(t) * h(t) = (m(t) + \sum_{i=1}^{M} \sqrt{\lambda_i}\phi_i(t)\eta_i) * h(t) \tag{4.30}$$

Furthermore, using the distribution property, the output becomes:

$$y(t) = (m(t) * h(t) + \sum_{i=1}^{M} \left(\sqrt{\lambda_i}\eta_i\phi_i(t) * h(t)\right) \tag{4.31}$$

The impulse response of a system can be computed using the approach proposed in [49]. The output can then be computed analytically by a simple convolution and thus the need of simulation is completely removed.

An equivalent approach is to statically propagate the KLE representation through the data-flow graph of the application. When the operands of a linear operation are represented with KLEs, the result can be computed using arithmetic operations between the coefficients:

Scalar multiplication

$$\begin{aligned} x &= m + \textstyle\sum_{i=1}^{M} x_i\eta_i \\ z &= a \times x = ma + \textstyle\sum_{i=1}^{M}(ax_i)\eta_i \end{aligned} \tag{4.32}$$

Addition/Subtraction

$$\begin{aligned}&x = m_x + \textstyle\sum_{i=1}^{M_1} x_i\eta_i \quad\quad y = m_y + \textstyle\sum_{j=1}^{M_2} x_j\eta_j \\ &z = x + y = (m_x + m_y) + \textstyle\sum_{i=1}^{max(M_1,M_2)} (x_i + y_i)\eta_i\end{aligned} \tag{4.33}$$

Using one of the two proposed methods, the output KLE representation can be computed without any sort of simulation.

4.2.3 Probability density function estimation

In [79] a trade-off between the wordlength and the application SQNR is proposed. When the overflows occur in the middle of the computation path, this evaluation may become inaccurate. We propose an integer wordlength optimization criteria based on the overflow probability. The range for all variables is computed from the probability density function (PDF) with respect to a coverage probability.

Propagating the input variability through the system, a KLE representation is obtained for every variable in the system:

$$y(t, \theta) = y_0(t) + \sum_{i=1}^{M} y_i(t)\eta_i(\theta) \tag{4.34}$$

If the input process is Gaussian, then all $\{\eta_i\}$ become independent standard Gaussian random variables. The output y can then be simulated directly by generating samples for $\{\eta_i\}$ from a Gaussian distribution. For Non-Gaussian inputs, the $\{\eta_i\}$ are mutually dependent and have unknown PDFs. However, it is still possible to obtain the corresponding samples from the input process using the equation (4.18).

The PDF can be approximated using one of the methods:

- Histogram

  The simplest method of PDF estimation is to generate a histogram from N samples of the output.

- Kernel Density Estimation

  This method [59] approximates the density function f_X by:

$$\widehat{f}_X(x) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big) \tag{4.35}$$

  Where K is the kernel, h is the bandwidth and n is the total number of samples. They are parameters that are chosen depending on the shape of the distribution.

- Edgeworth Expansion

  If the output distribution is weakly non-Gaussian, the Edgeworth Expansion [5] provides a good approximation for the PDF in terms of its cumulants using the Gaussian density as a reference function. It is a truly asymptotic expansion that allows controlling the error. As an example, the Edgeworth expansion with the first three terms is shown in equation 4.36:

$$\begin{aligned}\widehat{f}_X(x) = \Psi(x)\Big(1 + \tfrac{\gamma_1}{3!\sigma^3}H_3\big(\tfrac{x-\mu}{\sigma}\big) + \tfrac{\gamma_2}{4!\sigma^4}H_4\big(\tfrac{x-\mu}{\sigma}\big) \\ + \tfrac{10\gamma_1^2}{6!\sigma^4}H_6\big(\tfrac{x-\mu}{\sigma}\big)\Big)\end{aligned} \tag{4.36}$$

  where $\Psi(x)$ is the standard normal density, H_3, H_4, H_6 are the Hermite polynomials and γ_1 and γ_2 are the skewness and kurtosis respectively.

4.2.4 Comparison with the Affine Arithmetic

The Affine Arithmetic [18] was presented in Chapter 2.5.1. It is a model that keeps track of the first-order correlation between variables by representing a variable x with an affine form:

$$\widehat{x} = x_0 + x_1\epsilon_1 + x_2\epsilon_2 + \cdots + x_n\epsilon_n \tag{4.37}$$

where $\epsilon_i \in [-1, +1]$ are independent symbolic variables that represents an uncertainty component. As they can appear in the expression of several variables in the program, the AA model can remove the spatial dependence between the operands.

However, in the case of signal processing many variables are the result of a delay in time of the input $(x[n], x[n-1], \cdots)$. Using the AA they are supposed to be independent, so this method cannot keep track of the temporal correlation.

Because the noise terms ϵ_i are represented only by their maximal and minimal values, the shape of the PDF of x cannot be determined and its variability can only be characterized by its maximal and minimal bounds.

Similarly to the AA, the KLE represents a variable in an affine form as in equation 4.19. For the linear operations, the KLE operations are computed in a similar manner to the AA. The difference is that the random variables that appear in the expansion have an unknown distribution and generally have an infinite support. As a result, the variability is represented by the entire PDF. In addition, the KLE incorporates the temporal correlation of the input process also.

4.3 Range Evaluation Methodology

The methodology for the range determination in LTI systems is summarized in Figure 4.10.

The input application is described as a C/C++ code that uses floating-point representations for the variables. Using the framework for the automatic floating-point to fixed-point transformation that has been developed by the CAIRN/IRISA [28, 48, 49] (called *ID.fix*), the application is transformed into a Signal Flow Graph. The impulse response is determined using the method described in [49].

Separately, the range evaluation using the KLE method was developed in Matlab [44] and has been further integrated in the automatic floating-point to fixed-point transformation tool.

The KLE discretization of the input is realized in the following manner:

- If the covariance is not known, the unbiased estimators for the mean and the covariance are determined

- The eigenproblem is resolved using standard techniques

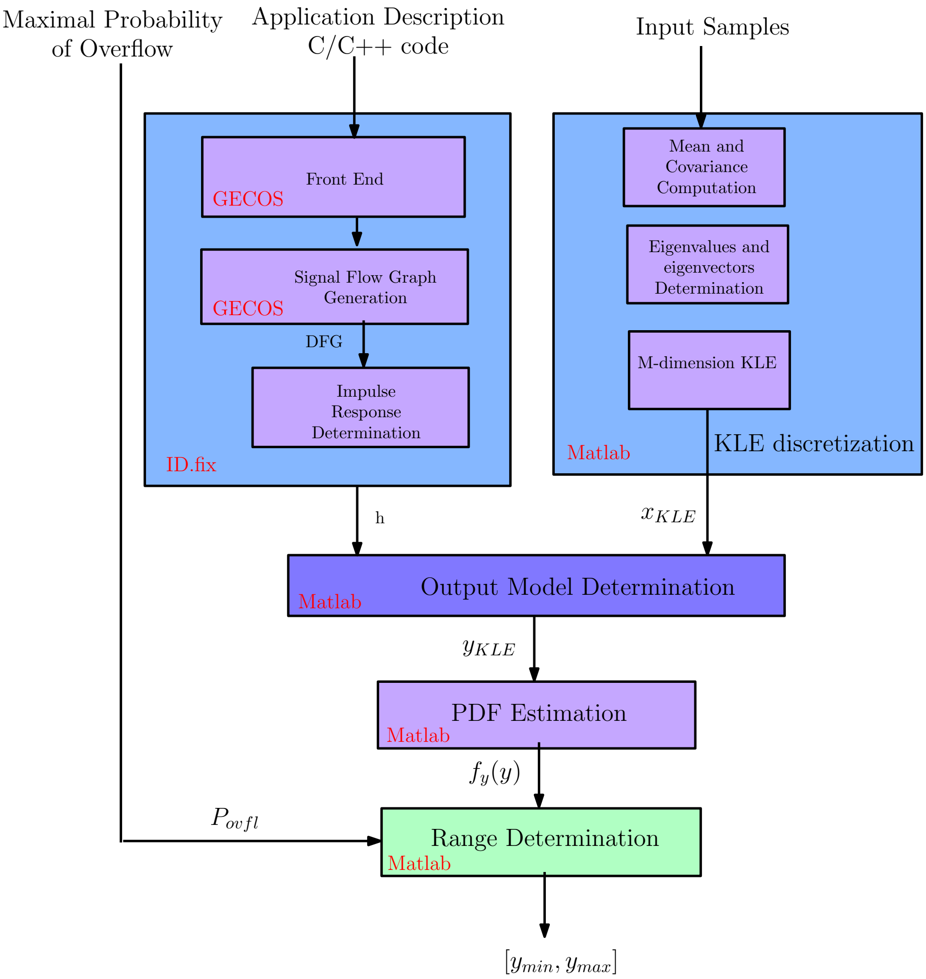

Maximal Probability of Overflow
Application Description C/C++ code
Input Samples
Front End
GECOS
Signal Flow Graph Generation
GECOS
DFG
Impulse Response Determination
ID.fix
Mean and Covariance Computation
Eigenvalues and eigenvectors Determination
M-dimension KLE
Matlab
KLE discretization
h
x_{KLE}
Output Model Determination
Matlab
y_{KLE}
PDF Estimation
Matlab
$f_y(y)$
P_{ovfl}
Range Determination
Matlab
$[y_{min}, y_{max}]$

Figure 4.10: Methodology Description for the Range Determination

- From equation 4.21 the M most important terms of the KLE are determined.

Then, then KLE representation of the input is propagated through the system analytically and the corresponding PDF is estimated for every variable in the system. Finally, the range of the output is further determined according the authorized overflow probability.

4.3.1 Experimental Results

In this part, we present the results obtained for several DSP applications. A 31-tap FIR filter, a 4^{th} order IIR filter and a 512-point IFFT are used for the tests. The input samples are generated using the AR(1) model as described in Section 4.2.1. The PDF is estimated using the Kernel Density Estimation method. Furthermore, the PDFs and the overflow probabilities are compared with the results obtained by simulation. The size of the range interval is also measured with the traditional L_1 norm method.

In order to compute the probability of overflow, the length of the impulse response has to be taken into account. In the case of non-recursive systems, like the FIR filters, the transient response has a finite duration and the values of the output at the steady-state can be used for the computation of the PDF. The PDF of the output of the FIR is plotted in Figure 4.11. As it can be seen, the PDF determined using the KLE method is very close to the histogram obtained in simulation.

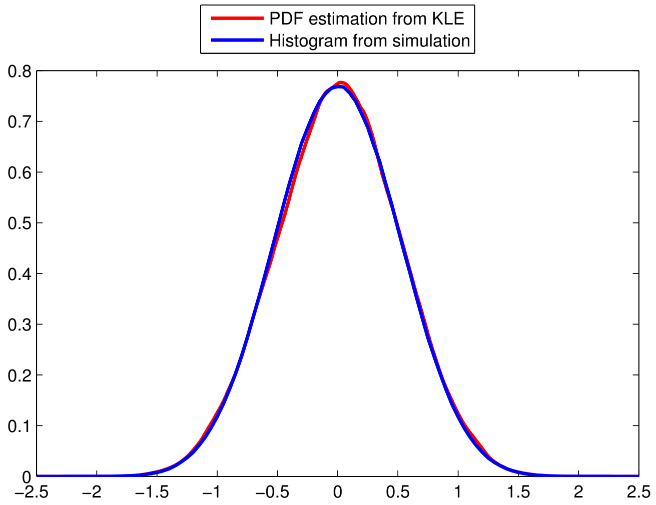


Figure 4.11: PDF of the FIR filter output

Recursive systems have a theoretical infinite impulse response. However, practical recursive systems will have a decay of the impulse response and its computation is made possible [49]. The variation of the output PDF for an IIR filter estimated with our method in time is plotted in Figure 4.12. It can be seen that it will converge after a finite time. This reflects the fact that the recursive filter has a stable behavior and the output will not diverge.

Comparison with simulation

First, the variance of the output signals for the three different examples is pre-

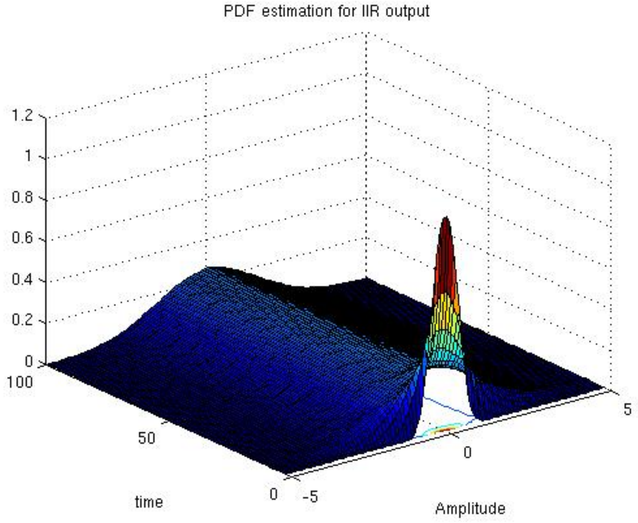


Figure 4.12: IIR filter output

sented in Table 4.1. The values obtained from the KLE are close to the simulation results.

Variance	KLE	Simulation	Error
FIR31	0.257	0.257	0
IIR4	2.137	2.146	0.41%
512 IFFT	0.8879	0.8881	0.02%

Table 4.1: Variance comparison

For a chosen probability of overflow the corresponding minimal and maximal bounds of the signal are determined. In order to test the accuracy of the results, the overflow probability for the obtained interval is computed from a simulation with 10^7 samples. The results are presented in Table 4.2. In all of the cases, the probabilities are in the same spectrum.

Comparison with L_1 norm

Next, the range evaluation is realized using the L_1 norm or interval arithmetic. The PDF from the KLE of the output of the FIR filter along with the maximal and minimal bounds found with the classical method can be seen in Figure 4.13.

As it can be seen in Table 4.3, the classical method overestimates the ranges for

	Overflow Probability KLE	Overflow Probability Simulation
	10^{-3}	$0.94 * 10^{-3}$
FIR31	10^{-4}	$1.14 * 10^{-4}$
	10^{-5}	$0.74 * 10^{-5}$
	10^{-3}	$0.963 * 10^{-3}$
IIR4	10^{-4}	$0.971 * 10^{-4}$
	10^{-5}	$1.98 * 10^{-5}$
	10^{-3}	$0.975 * 10^{-3}$
512 IFFT	10^{-4}	$1.08 * 10^{-4}$
	10^{-5}	$1.13 * 10^{-5}$

Table 4.2: Overflow Probability comparison between KLE and simulation

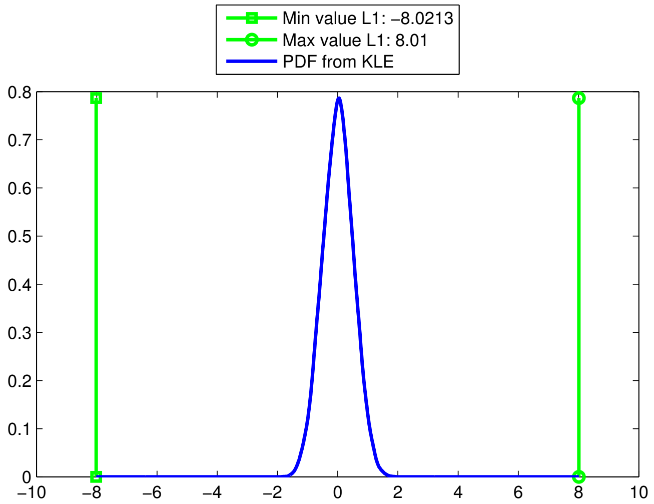


Figure 4.13: FIR filter output

all the applications that have been considered. Translated to the number of bits for the integer part, the L_1 norm increases the wordlength with 1 bit in the case of the IIR filter or even 3 bits in the case of the IFFT.

Implementation cost

Using the bounds computed with the L_1 norm method, the system would be over-dimensioned. Based on the synthesis results already presented in Section 3.2 the increase in the implementation cost introduced by the over-estimation is analyzed.

For the 65nm LP 1.2 V target technology, the additional cost in terms of area and power consumption of the increase of 3 bits for the datapath of the 512-point

	Overflow Probability	KLE Range	L_1 norm range
FIR31	10^{-3}	[-1.5659:1.6127]	[-8.021 :8.01]
	10^{-4}	[-1.7814 :1.8282]	
	10^{-5}	[-1.9969 :2.0437]	
IIR4	10^{-3}	[-4.4967:4.73572]	[-14.918:14.896]
	10^{-4}	[-5.1897: 5.4288]	
	10^{-5}	[-5.5690 :5.8080]	
512 IFFT	10^{-3}	[-4.3524 :4.1433]	[-60.01:60.28]
	10^{-4}	[-5.3485:5.1394]	
	10^{-5}	[-6.2420:6.0329]	

Table 4.3: Range comparison between KLE and L1 norm

IFFT can be seen in Table 4.4 and Table 4.5.

Method	Number of bits	Frequency (MHz)	Area (μm^2)
KLE	10	320	661459
L_1	13	320	912882
Gain			$\approx 27\%$
KLE	11	320	755184
L_1	14	320	977072
Gain			$\approx 22\%$

Table 4.4: Area comparison

Number of bits	Frequency (MHz)	Power (mW)
10	320	65
13	320	99
Gain		$\approx 34\%$
11	320	74
14	320	110
Gain		$\approx 32\%$

Table 4.5: Power consumption comparison

This proves that the gain in terms of area and power consumption that can be obtained using our range evaluation method is substantial, and shows the motivation that stands behind our probabilistic approach.

4.4 Quantization Noise And Numerical Accuracy Evaluation

We apply the same computational approach to evaluate the quantization noise, extending the method to the numerical accuracy estimation. Previous methods [18, 49, 62] evaluate only the variance of the output quantization noise. In addition, we will show that the entire PDF of the noise can be determined. This supplementary information is required in the characterization of unsmooth operators for which the model based on perturbation theory is no longer valid [58].

Every quantization operation realized when an infinite precision value is replaced with a fixed-point representation introduces an error that can be modeled as an additive uniform white noise as it was shown in Section 2.3.2.1. In this work, only the case of rounding operations is considered, where the quantization noise is distributed in the interval $\left[\frac{-2^n}{2}, \frac{2^n}{2}\right]$ with n the number of bits for the fractional part. The truncation can be treated in a similar manner.

The fixed-point input can thus be replaced by the expression $x[t] + q_x[t]$, where $x[t]$ represents the infinite precision value and $q_x[t]$ the quantization noise (Figure 4.14).

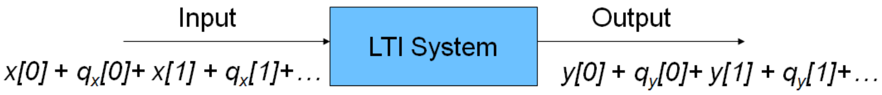

Input
x[0] + q_x[0]+ x[1] + q_x[1]+…
LTI System
Output
y[0] + q_y[0]+ y[1] + q_y[1]+…

Figure 4.14: Input/output view of the system

Because the quantization noise is uncorrelated with the signal, and we are only dealing with LTI systems, the superposition property can be applied and the signal and the noise can be analyzed separately.

As a result, the precision analysis can be formulated in a similar manner with the range estimation: $q_y(t) = f(q_{x1}(t), q_{x2}(t), \cdots, q_{xn}(t))$, where the random input is represented by the quantization noise.

For all the three applications presented in Section 4.3.1, the SQNR is computed with our method for various wordlengths of the data-path and is compared with the values obtained with a fixed-point simulation. As it can be seen in Table 4.6, 4.7 and 4.8 the values are close to the experimental results obtained in simulation.

Wordlength	6 bits	7 bits	8 bits
SQNR KLE	39.76 dB	45.78 dB	51.79 dB
SQNR reference	39.99 dB	46.00 dB	52.03 dB

Table 4.6: FIR SQNR comparison

Wordlength	6 bits	7 bits	8 bits
SQNR KLE	33.68 dB	39.73 dB	45.77 dB
SQNR reference	33.86 dB	39.88 dB	45.88 dB

Table 4.7: IIR SQNR comparison

Wordlength	10 bits	11 bits	12 bits
SQNR KLE	35.24 dB	41.27 dB	47.29 dB
SQNR reference	35.28 dB	41.56 dB	47.77 dB

Table 4.8: IFFT SQNR comparison

In addition, the PDF of the FIR output quantization noise is determined with our method and it is compared with the one obtained in simulation Figure 4.15. The two match very well. It can be seen that they do not have a Gaussian PDF.

With our approach, both the range determination and the numerical accuracy evaluation can be realized. A trade-off between an occasional error (integer part width) and a global SQNR (fractional part width) can be made. It can be very useful when the implementation has a limited width for the data-path and the application has a high peak-to-average power as in the case of the OFDM transmitter.

4.5 Conclusion

In this Chapter, a method for the range evaluation in the context of LTI systems was presented. The Karhunen-Love Expansion is used as a means of representation for the variability of the input signal. Furthermore, we showed how the variability can

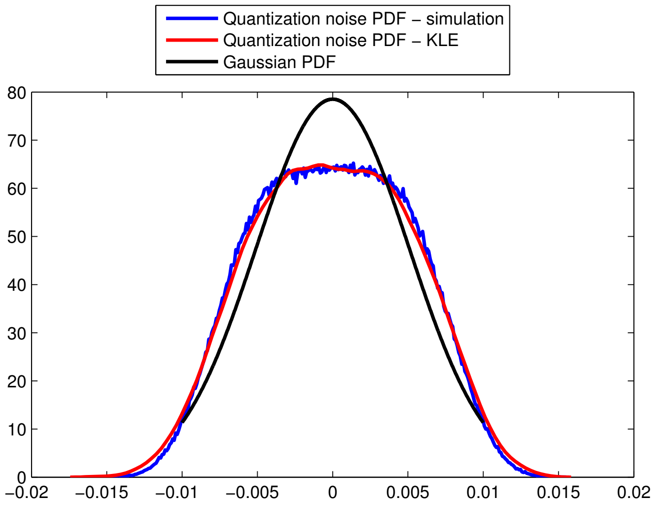


Figure 4.15: Output quantization noise PDF

be statically propagated through LTI systems and how the corresponding output representation is obtained.

Several methods for the PDF determination were presented which allow the computation of a large class of PDF shapes. For systems where occasional overflows are accepted, the dynamic range of all variables is computed from their corresponding PDFs with respect to a desired overflow probability.

As a secondary goal, we used the same computational approach to solve the numerical accuracy evaluation problem. The SQNR of the application is determined from the quantization noise variance. In addition, the complete noise PDF can be estimated if needed. The experiments show the accuracy of the method.

Chapter 5

Polynomial Chaos Expansion Method

In this Chapter, a method for the range determination based on the Polynomial Chaos Expansion is presented. It will be shown how the PCE can be used to acquire a representation of the input of the system and how the corresponding PCE of each variable can be obtained. The advantage of the PCE representation is the fact that the PCE arithmetic can be applied for non-linear operations also. As a result the range and the numerical accuracy estimation problems is solved for all types of systems with arithmetic operations. In comparison to the KLE method it has an increased complexity so its applicability to LTI systems is less interesting.

5.1 Polynomial Chaos Expansion Introduction

5.1.1 1-dimensional Hermite polynomials

The Hermite polynomials are an orthogonal polynomial sequence, defined in equation (5.1):

$$H_n(x) = (-1)^n \frac{1}{\phi(x)} \frac{d^n \phi(x)}{dx^n} \tag{5.1}$$

with $\phi(x) = \frac{1}{\sqrt{(2\pi)}} e^{-\frac{-x^2}{2}}$

They become:

$$H_n(x) = (-1)^n e^{-\frac{x^2}{2}} \frac{d^n}{dx^n} e^{-\frac{-x^2}{2}} \ \ , \ n = 0, 1, 2, \cdots \tag{5.2}$$

However, it is easier to compute them using the following recursion relation:

$$\begin{aligned} H_{-1}(x) &= H_0(x) = 1 \\ H_{n+1}(x) &= xH_n(x) - nH_{n-1}(x) \end{aligned} \tag{5.3}$$

As an example, the first 10 polynomials are:

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= x \\ H_2(x) &= x^2 - 1 \\ H_3(x) &= x^3 - 3x \\ H_4(x) &= x^4 - 6x^2 + 3 \\ H_5(x) &= x^5 - 10x^3 + 15x \\ H_6(x) &= x^6 - 15x^4 + 45x^2 - 15 \\ H_7(x) &= x^7 - 21x^5 + 105x^3 - 105x \\ H_8(x) &= x^8 - 28x^6 + 210x^4 - 420x^2 + 105 \\ H_9(x) &= x^9 - 36x^7 + 378x^5 - 1260x^3 + 945x \\ H_{10}(x) &= x^{10} - 45x^8 + 360x^6 - 3150x^4 + 4725x^2 - 945 \end{aligned} \tag{5.4}$$

The Hermite polynomials form an orthogonal basis of the Hilbert space of square integrable functions with respect to the inner product:

$$\langle H_i H_j \rangle = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} H_i(x) H_j(x) w(x) dx = \delta_{ij} \langle H_i^2 \rangle \tag{5.5}$$

where δ_{ij} is the Kronecker delta and $w(x) = e^{\frac{-x^2}{2}}$ is the weighting function.

5.1.2 Multi-dimensional Polynomial Chaos Expansion

The original Homogeneous Chaos was introduced by Wiener [76] as a means of representation for the Gaussian stochastic processes using the multi-dimensional Hermite polynomials in terms of Gaussian random variables as a basis of the random space. Let $L_2(\Phi, F, P)$ be the Hilbert space of random variables with finite variance. The Cameron-Martin theorem [6] proves that any second-order (L_2) random variable (or random process) can be represented as a mean-square convergent series of infinite-dimensional Hermite polynomials in terms of Gaussian random variables, called the Polynomial Chaos Expansion (PCE):

$$\begin{aligned} x(\theta) = \widehat{x}_0 H_0 &+ \underbrace{\sum_{i_1=1}^{\infty} \widehat{x}_{i_1} H_1(\xi_{i_1}(\theta))}_{1^{st} \text{ order terms}} + \underbrace{\sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \widehat{x}_{i_1 i_2} H_2(\xi_{i_1}(\theta), \xi_{i_2}(\theta))}_{2^{nd} \text{ order terms}} \\ &+ \underbrace{\sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} \widehat{x}_{i_1 i_2 i_3} H_3(\xi_{i_1}(\theta), \xi_{i_2}(\theta), \xi_{i_3}(\theta))}_{3^{rd} \text{ order terms}} + \cdots \end{aligned} \tag{5.6}$$

where $\{\widehat{x}_{i_1 i_2 \cdots}\}$ are the coefficients, $H_n(\xi_{i_1}(\theta), \cdots, \xi_{i_n}(\theta))$ are the multi-dimensional Hermite polynomials of order n in terms of the random vector $\boldsymbol{\xi} = \{\xi_{i_1}(\theta), \xi_{i_2}(\theta), \cdots, \xi_{i_n}(\theta)\}$ of independent standard Gaussian random variables. They are defined in equation (5.7). The non-Gaussian behaviour is represented by the terms that have a degree superior to one.

$$H_n(\xi_{i_1}(\theta), \cdots, \xi_{i_n}(\theta)) = e^{\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi}} (-1)^n \frac{\partial^n}{\partial \xi_{i_1} \cdots \partial \xi_{i_n}} e^{-\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi}} \tag{5.7}$$

For notational convenience the PCE representation is rewritten as:

$$x(\theta) = \sum_{j=0}^{\infty} x_j \Psi_j(\boldsymbol{\xi}(\theta)) \tag{5.8}$$

This is simply a reordering of the terms in the summation, with a one-to-one correspondence between the Hermite polynomials $H_n(\xi_{i_1}(\theta), \cdots, \xi_{i_n}(\theta))$ and $\Psi_j(\boldsymbol{\xi}(\theta))$ and between the coefficients $\widehat{x}_{i_1 \cdots i_n}$ and x_j.

The PCE forms a complete orthogonal basis of the L_2 space of random variables:

$$\langle \Psi_i \Psi_j \rangle = \langle \Psi_i^2 \rangle \delta_{ij} \tag{5.9}$$

where δ_{ij} is the Kronecker delta and $\langle , \rangle$ is the inner product defined as:

$$\langle \Psi_i \Psi_j \rangle = \int_{-\infty}^{\infty} \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) W(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{5.10}$$

with the weighting function $W(\boldsymbol{\xi}) = \frac{1}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi}}$

The PCE representation has a theoretically infinite number of terms, however in practice the expansion is truncated to a limited number of terms. The number of random variables M of the random vector $\boldsymbol{\xi} = \{\xi_1, \xi_2, \cdots, \xi_M\}$ is called the dimension and the highest power p is called the order of the PCE. It becomes:

$$x(\theta) = \sum_{j=0}^{N} x_j \Psi_j(\xi_1, \xi_2, \cdots, \xi_M) \tag{5.11}$$

where the number of terms N is a function of the order and the dimension:

$$N = \frac{(M+p)!}{M!p!} - 1 \tag{5.12}$$

Using the PCE, the computation of the random variable $x(\theta)$ is replaced with the computation of the deterministic spectral coefficients x_j, as they characterize the entire stochastic dimension of the input. A higher dimension takes into account higher frequency random fluctuations while a higher order represents better the non-linearities. However, the number of terms of the PCE basis increases very fast with the order and the dimension so they have to be limited in practice. As an example, the 2-dimensional 4^{th} order PCE that has 15 terms is represented in Table 5.1.

A construction procedure for the M-dimensional p-order PCE basis was first proposed in [24] based on the relation from equation (5.7). A different approach was presented in [67, 68] that uses the fact that the M-dimensional Hermite polynomials $\Psi_j(\xi_1, \xi_2, \cdots, \xi_M)$ are in fact a tensor product of the M 1-dimensional Hermite

index (j)	order (p)	Ψ_j	$E[\Psi_j^2]$
0	0	1	1
1	1	ξ_1	1
2	1	ξ_2	1
3	2	$\xi_1^2 - 1$	1
4	2	$\xi_1\xi_2$	1
5	2	$\xi_2^2{-}1$	1
6	3	$\xi_1^3 - 3\xi_1$	1
7	3	$\xi_2(\xi_1^2 - 1)$	1
8	3	$\xi_1(\xi_2^2 - 1)$	1
9	3	$\xi_2^3 - 3\xi_2$	1
10	4	$\xi_1^4 - 6\xi_1^2 + 3$	1
11	4	$\xi_2(\xi_1^3 - 3\xi_1)$	1
12	4	$(\xi_1^2 - 1)(\xi_2^2 - 1)$	1
13	4	$\xi_1(\xi_2^3 - 3\xi_2)$	1
14	4	$\xi_2^4 - 6\xi_2^2 + 3$	1

Table 5.1: 2-dimensional 4^{th} order PCE

polynomials:

$$\Psi_\alpha(\boldsymbol{\xi}) = \prod_{i=1}^{M} H_{\alpha_i}(\xi_i) \tag{5.13}$$

Each polynomial of the PCE basis Ψ_α, is thus completely defined by a sequence of M integers $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \cdots, \alpha_M\}$. Because the order of the PCE is set to p, the sum $\alpha_1 + \alpha_2 + \cdots + \alpha_M \leq p$ and $\alpha_1 \leq 0$.

First, the 1-dimensional Hermite polynomials up to p^{th} order are computed using the recurrence relation from equation (5.3). Then, the M-dimensional p-order PCE basis can be generated by computing all the sequences of $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_M\}$ whose sum $\alpha_1 + \alpha_2 + \cdots + \alpha_M \leq p$.

This PCE construction method is more adapted to the case when the output of a system is represented as a function of several random variables that are expanded separately using 1-dimensional Hermite polynomials. The output will then be represented as a joint-expansion of the variables in terms of a multi-dimensional PCE.

5.2 Probabilistic framework

The range evaluation is realized using a probabilistic framework where the input $\boldsymbol{x} = \{x^{(1)}, \cdots, x^{(N)}\}$ is modeled as a N-variate random vector. As a result, the variability of each variable is computed by evaluating a function of random variables.

In [79] a method for the range estimation based on the PCE was proposed. It represents the temporal variability when the input has a correspondence to a physical random process that varies in time. The authors used the KLE discretization in order to obtain a reduced-order representation of the input that still captures the probabilistic content that characterizes the uncertainty of data. In this way the correlation introduced by the delay operations that exist in many DSP applications is incorporated into the PCE representation.

However, in some applications the operands do not represent the values of a process at different time instants. As a result of its inherent relation with the use with the KLE and the temporal discretization, the method is not adapted to the case where the operands come from different signal sources, have different probabilistic distributions and may be correlated. In this case, each operands should be represented by a different random variable. Furthermore, their approach for the integer wordlength determination based on a SQNR trade-off is not always accurate.

In this Chapter, we will show how the PCE can be adapted to treat the case of random variables. The PCE representation is obtained for every input variable and an analytical description of the variability of the output is determined. Furthermore, the correlation of the inputs is captured using the Nataf transform. The range is computed using a probabilistic analysis from the PDF in the same manner as for the KLE method.

5.3 Input representation

5.3.1 PCE representation for independent random variables

Any L_2 random variable (with finite variance) can be represented with a mean-square convergent series of 1- dimensional Hermite polynomials as in (5.14). The order of the expansion needed for an accurate approximation is given by the non-Gaussian character of the distribution.

$$x = \sum_{i=0}^{\infty} x_i H_i(\xi) \tag{5.14}$$

where H_n are the 1-dimension Hermite polynomials and ξ is a standard Gaussian random variable.

The problem of computing x is replaced with finding the coefficients x_i. One of the methods proposed in the literature to obtain the PCE coefficients is the Galerkin projection [68]. It is based on the fact that the Hermite polynomials are orthogonal. This means that if we multiply on each side by H_i and take the expectations, the coefficients are:

$$x_i = \frac{\langle x H_i(\xi) \rangle}{\langle H_i^2(\xi) \rangle} \tag{5.15}$$

The denominator can be computed analytically:

$$\langle H_i^2(\xi) \rangle = E[H_i^2(\xi)] = i! \tag{5.16}$$

The numerator is:

$$\langle x H_i(\xi) \rangle = E[x H_i(\xi)] = \int_{\Re} x H_i(\xi) w(x) dx \tag{5.17}$$

where $w(x) = \frac{1}{\sqrt{(2\pi)}} e^{\frac{-x^2}{2}}$ is the weight function.

The random variable ξ has a Gaussian PDF $g(\xi)$ and a CDF denoted by $G(\xi)$. Let the CDF of x be $F_X(x)$. Considering the isoprobabilistic transformation $F_X(x) =$

$G(\xi)$, then $x = F_X^{-1}(G(\xi))$ and:

$$x_i = \frac{1}{i!} \int_{\Re} F_X^{-1}(G(t)) H_i(t) \phi(t) dt \tag{5.18}$$

If the CDF $F_X(x)$ is not known analytically, it can be estimated from samples. In the particular case when x has a Gaussian or uniform distribution, the coefficients can be computed analytically [68]. Otherwise, the integral can be solved using Monte Carlo techniques.

For the Gaussian distribution:

$$N(\mu, \sigma) \rightarrow x_0 = \mu,\ x_1 = \sigma,\ x_i = 0, \quad \text{for} \quad i \geqslant 2 \tag{5.19}$$

The PCE for x becomes:

$$x = \mu H_0(\xi) + \sigma H_1(\xi) \tag{5.20}$$

For the uniform distribution:

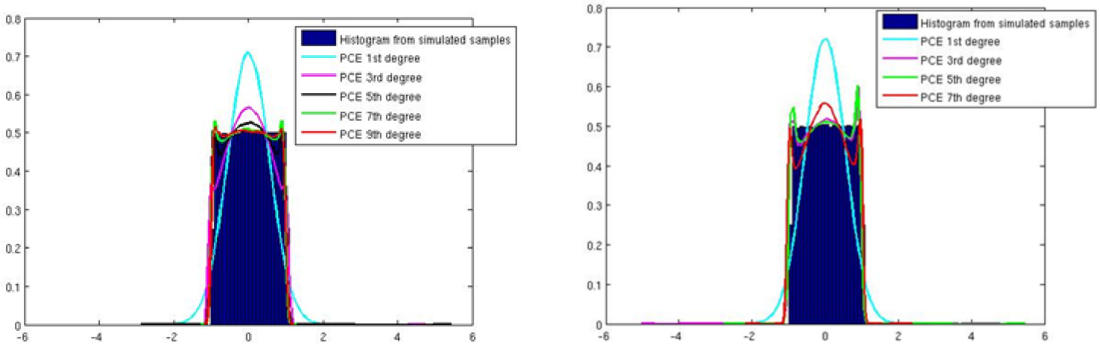
$$U(a, b) \rightarrow x_0 = \tfrac{a+b}{2},\ x_{2i} = 0,\ x_{2i+1} = \tfrac{(-1)^i (b-a)}{2^{2i+1}\sqrt{\pi}(2i+1)i!} \tag{5.21}$$

The PCE for x becomes:

$$x = \frac{a+b}{2} H_0(\xi) + \frac{(b-a)}{2\sqrt{\pi}} H_1(\xi) + \frac{(-1)(b-a)}{24\sqrt{\pi}} H_3(\xi) + \cdots \tag{5.22}$$

Let x be a uniform random variable in the interval [-1,1]. The coefficients for the PCE representation are computed using the Monte Carlo method using 10^6 samples. The PDFs determined using both the analytical and Monte Carlo coefficients for different expansion orders are presented in Figure 5.1.

Let x be a random variable that follows a gamma distribution with the shape parameter $k = 2$ and the scale $\theta = 1$. The PCE coefficients are determined using a Monte Carlo simulation with 10^6 samples. The PDF estimation for several PCE



(a) Theoretical coefficients

(b) Monte Carlo determined coefficients

Figure 5.1: PDF comparison for a uniform random variable

orders can be seen in Figure 5.2. Compared to the uniform distribution, it can be noticed that a lower PCE order is needed for an accurate approximation.

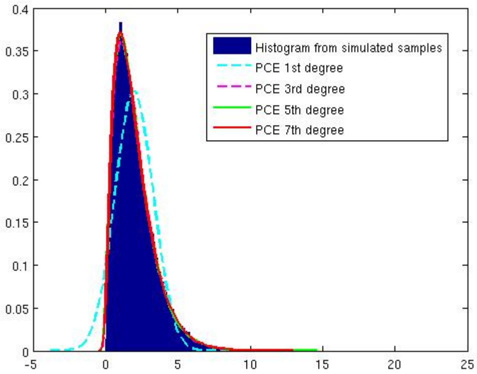


Figure 5.2: PDF comparison for a gamma random variable

5.3.2 Correlated Random Variables

Nataf transform

Let $\boldsymbol{x} = \{x^{(1)}, x^{(2)}, \cdots, x^{(n)}\}$ be the random vector that contains all the inputs variables. If the variables are correlated, it is not possible to expand them in a PCE independently as proposed earlier. In order to solve this problem, a decorrelation procedure has to be applied. A procedure that transforms the random vector $\boldsymbol{x}$ into another random vector $\boldsymbol{z}$ with the same dimension but with independent standard Gaussian components is employed. The transformation was introduced by Nataf [52]. The advantage of the method in comparison to other approaches (e.g. the Rosenblatt transform [61]) is that it only needs the marginal distributions and the correlation structure of the random vector in order to be applied. This occurs in most practical cases, where the joint PDF is unknown or is difficult to estimate and only the marginal PDFs and the correlation matrix can be determined from samples.

Let the correlation matrix of $\boldsymbol{x}$ be $\boldsymbol{C}$. The marginal PDF of each random variable $x^{(i)}$ is $f_i(x)$ and the corresponding CDF is $F_i(x)$. The isoprobabilistic transformation is realized in two steps:

T1:

$$u^{(i)} = \Phi^{-1}(F_i(x^{(i)})) \tag{5.23}$$

$\boldsymbol{x}$ is transformed using the marginal distributions into a Gaussian vector $\boldsymbol{u}$ with standard normal marginal distributions and correlation matrix $\boldsymbol{C_U}$.

T2:

$$\boldsymbol{z} = \boldsymbol{u}\Gamma \tag{5.24}$$

where Γ is the Cholesky factor of $\boldsymbol{C}_U$: $\Gamma^T\Gamma = \boldsymbol{C}_U^{-1}$.

The second step is a linear transformation that is performed in order to decorrelate the components of $\boldsymbol{u}$. As it is a Gaussian random vector, they will be independent.

Once the independent standard Gaussian vector $\boldsymbol{z}$ has been determined, each input random variable can be represented using the Polynomial Chaos basis:

$$x^{(i)} = x_{i_0}\Psi_0 + x_{i_1}\Psi_1(z_1, z_2, \cdots, z_n) + \cdots + x_{i_N}\Psi_N(z_1, z_2, \cdots, z_n) \tag{5.25}$$

5.3.3 Construction of an M-dimensional PCE for random processes

If the input is a random process, the KLE can be used to obtain a reduced order representation in terms of M random variables as it was presented in Chapter 4.1.2.

$$x(t) = m(t) + \sum_{i=1}^{M} \sqrt{\lambda_i}\phi_i(t)\eta_i \tag{5.26}$$

As it was shown in [79], the KLE obtained previously can be transformed into an M-dimensional p-order PCE:

$$x(t) = m(t) + \sum_{i=1}^{M} \sqrt{\lambda_i}\phi_i(t)\eta_i \quad \rightarrow \quad x(t) = \sum_{i=0}^{N} x_i(t)\Psi(\xi_1, \xi_2, \cdots, \xi_M) \tag{5.27}$$

In the particular case, where the process $x(t)$ is Gaussian, the set of random variables η_i becomes a set of M independent standard Gaussian random variables. It results that in this case the KLE is exactly the M-dimensional first order PCE.

In the general case, the random variables are non Gaussian and only uncorrelated. However, if the distribution is not too far from the Gaussian, the independence property can be assumed and each variable $\{\eta_i\}$ is transformed independently into a 1-dimension p_i-order PCE in an analogous mode to Section 5.3.1:

$$\eta_i = \sum_{j=0}^{p_i} a_j H_j(\xi_i) \tag{5.28}$$

As a consequence of the fact that all $\{\eta_i\}$ are supposed independent, all the coefficients that correspond to cross terms in the PCE are zero and the expansion is not a true M-dimensional.

Because the independency property of $\{\xi_i\}$ will not be guaranteed for highly non-Gaussian stochastic processes, the error that is introduced in the PCE representation in this case may be significant. A procedure should be used to transform the set of uncorrelated non-Gaussian random variables into another set of independent random variables that can be further projected into a PCE [38].

5.4 PCE Arithmetics

From the previous Section, the PCE representation of all the inputs is obtained. Next, the arithmetic operations can be implemented using the PCE arithmetic presented in [14].

1. Scalar multiplication

   Let u be a variable with the PCE representation:

$$u = \sum_{i=0}^{N} u_i \Psi_i(\xi_1, \xi_2, \cdots, \xi_M) \tag{5.29}$$

   Then:

$$z = c \times u = \sum_{i=0}^{N} c \times u_i \Psi_i(\xi_1, \xi_2, \cdots, \xi_M) \tag{5.30}$$

2. Addition/Subtraction

   Consider the following two variables u and v:

$$u = \sum_{i=0}^{N} u_i \Psi_i(\xi_1, \xi_2, \cdots, \xi_M) \quad v = \sum_{i=0}^{N} v_i \Psi_i(\xi_1, \xi_2, \cdots, \xi_M) \tag{5.31}$$

   The addition/subtraction is realized as:

$$z = \Big(\sum_{i=0}^{N} u_i \Psi_i(\xi_1, \xi_2, \cdots, \xi_M) + \sum_{i=0}^{N} v_i \Psi_i(\xi_1, \xi_2, \cdots, \xi_M) \Big) \tag{5.32}$$

$$= \sum_{i=0}^{N} (u_i + v_i) \Psi_i(\xi_1, \xi_2, \cdots, \xi_M) \tag{5.33}$$

As an example, the PDF of the addition of two independent uniform random variables in $[-1, 1]$ can be seen in Figure 5.3. A comparison of the variance obtained for different PCE orders is presented in Table 5.2.

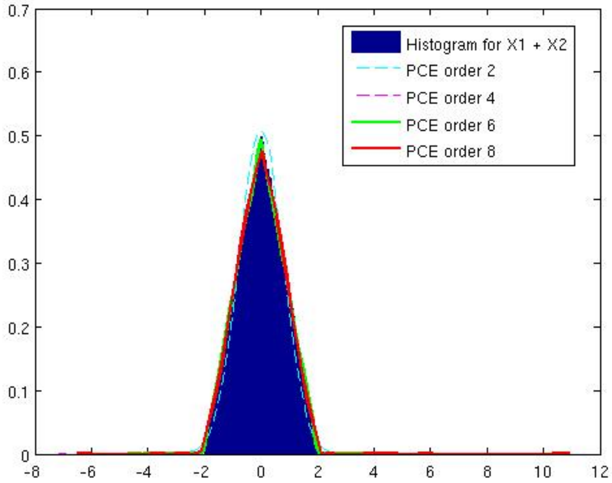


Figure 5.3: Addition of two uniform random variables

PCE order	Var PCE	Var Simulation	Difference	Nb terms
2	0.6113	0.6669	8.33%	6
4	0.6805	0.6669	-2.05%	15
6	0.6806	0.6669	-2.06%	28
8	0.6841	0.6669	-2.58%	45

Table 5.2: Variance comparison for the addition operation

3. Multiplication

Consider the same random variables as above. The multiplication is defined as follows:

$$z = u \times v = (\sum_{i=0}^{N} u_i \Psi_i(\xi_1, \xi_2, \cdots, \xi_M))(\sum_{j=0}^{N} v_j \Psi_j(\xi_1, \xi_2, \cdots, \xi_M)) \tag{5.34}$$

$$= \sum_{k=0}^{N} z_k \Psi(\xi_1, \xi_2, \cdots, \xi_M)) \tag{5.35}$$

The coefficients z_k are determined using the equation:

$$z_k = \sum_{i=1}^{N} \sum_{j=0}^{N} u_i v_j \frac{E[\Psi_i \Psi_j \Psi_k]}{E[\Psi_k^2]}, \quad k \in \{0, 1, \cdots, N\} \tag{5.36}$$

This is a Galerkin projection that minimizes the error of the resulting PCE representation on the space spanned by the polynomial basis up to the order N. The expectations $E[\Psi_k^2]$ and $E[\Psi_i \Psi_j \Psi_k]$ can be computed analytically as a pre-processing step and stored in a table as it is detailed in Section 5.4.1.

As an example, the PDF of the multiplication of two uniform random variables is shown in Figure 5.4 and a comparison of the variance is presented in Table 5.3.

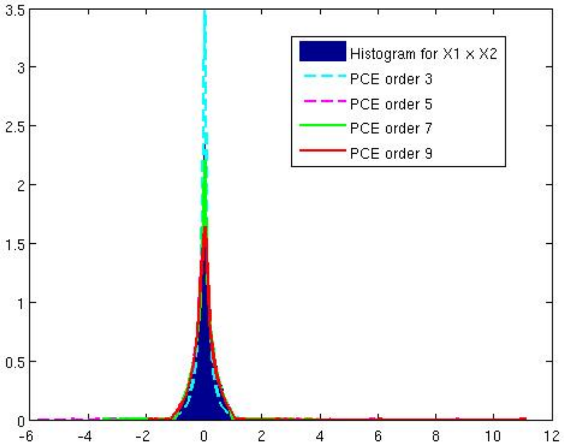


Figure 5.4: Multiplication of two uniform random variables

PCE order	Var PCE	Var Simulation	Difference	Nb terms
3	0.0933	0.1110	15.88%	10
5	0.1145	0.1110	-3.14%	21
7	0.1157	0.1110	-4.23%	36
9	0.1167	0.1110	-5.19%	55

Table 5.3: Variance comparison for the multiplication operation

As the PCE representations are in fact only approximations, the truncation error may become important when computing multiple multiplications.

The division operation can be computed considering that $z = u/v$ is in fact equivalent to $u = zv$. As u and v have known PCE representations, a system of $(N+1)$ linear equations can be created. Solving the system will give the PCE coefficients of the result. Even other types of non-polynomials operations (exponentials or logarithms) may be computed if needed [14].

As a conclusion, the PCE arithmetic can be used in order to statically propagate the variability of the input through the Data Flow Graph of the application.

5.4.1 Statistical analysis

The expectations $E[\Psi_i\Psi_j]$ and $E[\Psi_i\Psi_j\Psi_k]$ can be analytically computed using the fact that $\{\xi_i\}$ are all independent standard Gaussian random variables. As a result:

$$E[\xi_i^{2k}] = \frac{(2k)!}{2^k k!} \text{ and } E[\xi_i^{2k+1}] = 0 \tag{5.37}$$

Furthermore, the PDF can be estimated using the same approaches presented in Section 4.2.3.

5.5 Range Evaluation Methodology

The methodology for the range determination using the PCE is summarized in Figure 5.5.

- The first step is represented by the input representation. Each operand is discretized using the PCE. When delay operations appear in the data-path of the application: $x[n], x[n-1], \cdots$, the input is modeled as a random process. In this way the temporal correlation that exists between $x[n]$ and $x[n-1]$ can be captured. When the operands do not represent the values of the same process at different time instants, the input should be modeled as a random variable. Only independent random variables can be treated directly. If two inputs are correlated, the Nataf transform should be used to decorrelate them.

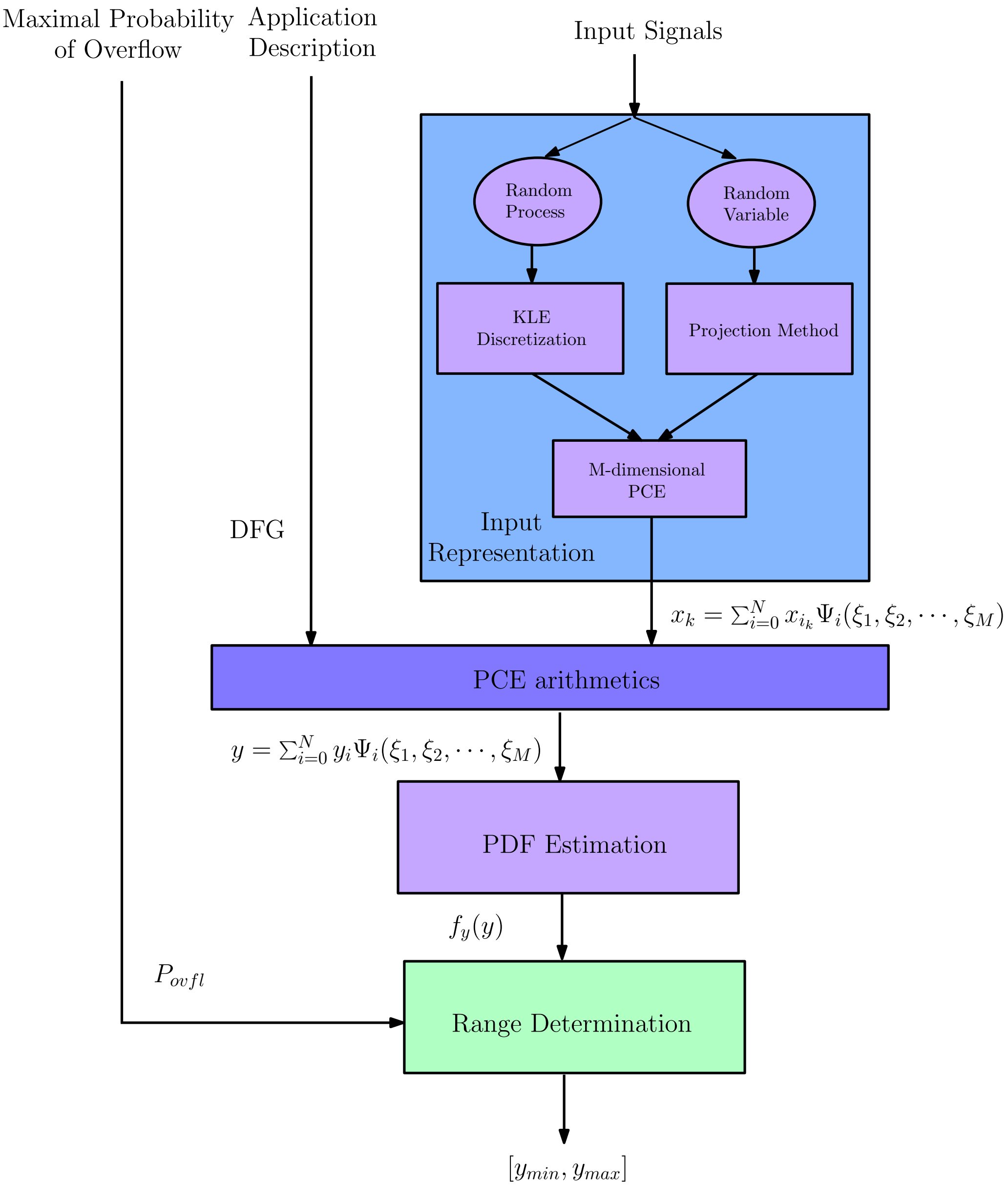


Figure 5.5: Methodology Description for the Range Determination using PCE

The corresponding PCE coefficients are computed as described in Section 5.3.3 or Section 5.3.1 depending if it is a random process or a random variable. At the end, the dimension M of the PCE should represent the number of all uncertainties that influence its random behavior.

- The PCE propagation is realized by applying the PCE arithmetic. The output is the result of a function of M variables and it will be represented using an M-dimensional PCE.

- PDF determination
- From the PDF and the allowed overflow probability, compute the maximal values and the number of integer part bits

The PCE method for the range evaluation has been implemented in Matlab based on the methodology that has been presented here. However, it has not been integrated into the automatic floating-point to fixed-point transformation tool yet.

5.6 Experimental Results

In this section we present the results obtained for some practical examples. We concentrate on two important aspects of the range estimation: non-linear operations and statistical correlation of the operands.

As a first example, we examine the ability of the PCE to evaluate the dynamic variations in the case of the approximation of a non-linear function that depends on only one random variable. Let x be a Gamma random variable with the scale parameter $k = 20$ and shape $\theta = 0.1$. The exponential function is evaluated with a 5^{th} order Taylor expansion:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + O(x^6) \tag{5.38}$$

The range for different overflow probabilities is computed. In order to see if the interval corresponds to the desired overflow probability in practice, the simulation of the Taylor expansion with 10^7 samples is made. For a chosen overflow probability, the number of times the result exceeds the corresponding interval is determined. The results are presented in Table 5.4. In all of the cases, the simulation evaluation is found to be in the same class of values. This proves that with the PCE representation, the tails of the distribution are accurately estimated.

As a comparison, the range of the output is computed using IA. The Gamma distribution has an infinite support. In order to propagate the variability using the

IA, the bounds of the input variable x are set to the minimal and maximal values found with a 10^7 samples simulation. The results show that our range analysis approach provides tighter range intervals in comparison to the IA.

Overflow Probability	Simulation	Obtained Range	IA
10^{-2}	1.04*10-2	[4.19 : 80.25]	[1.59:100.86]
10^{-3}	0.938*10-3	[2.43 : 81.61]	[1.59:100.86]
10^{-4}	0.915*10-4	[2.43 : 81.61]	[1.59:100.86]
10^{-5}	0.78*10-5	[1.81 : 82.67]	[1.59:100.86]

Table 5.4: Range Comparison For Different Overflow Probabilities

Correlated random variables

In order to generate random inputs that have different correlation structures, the copulas theory is used [53]. Copulas are functions that describe the dependence structure between the random variables.

In this example, a Gaussian copula is used to simulate the correlation between the inputs. When other types of copulas are used, the Nataf transform becomes less adapted for the situation. In this case, a generalized Nataf transform was presented [17] and can be used depending on the corresponding copula.

As a first example, let us consider two random variables, each one following a uniform distribution ($x_1 \sim U(-2, 2)$,$x_2 \sim U(-1, 1)$). The correlation coefficient r is varied modifying the dependence between them. When $r = 0$ the variables are independent and when $r = 1$ they are totally correlated. The addition operation between the two variables is realized using a 5^{th} order PCE. The PDF of the result with and without the Nataf transform for the case when $r = 0$ is shown in Figure 5.6. Because the variables are independent, the two distributions are similar and match the histogram from simulation. For $r = 0.8$ the result is presented in Figure 5.7. Without the Nataf transform, the two variables are supposed to be independent and the distribution is not very well approximated. As a consequence of applying the Nataf transform, the PDFs can be approximated more accurately.

As a second example, consider two Gaussian variables $x_1 \sim N(\mu = \mathit{0.3}, \sigma^2 = \mathit{0.2})$

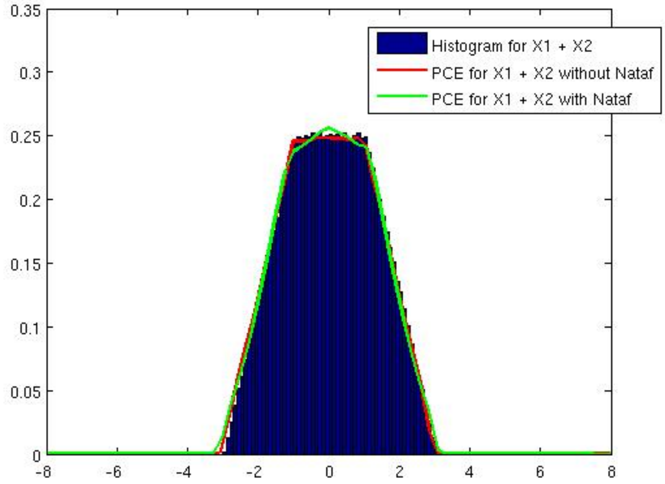


Figure 5.6: Addition of independent uniform random variables

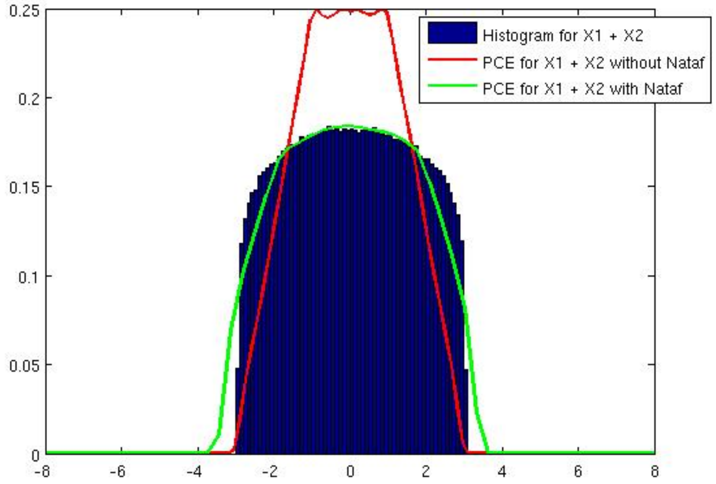


Figure 5.7: Addition of correlated uniform random variables with r=0.8

and $x_2 \sim N(\mu = 0.4, \sigma^2 = 0.4)$ and the following polynomial evaluation:

$$y = \left(0.3 + 1.7x_1 + 0.5x_1^2\right)\left(0.2 + 2.7x_2 + 0.5x_2^2\right) \tag{5.39}$$

In order to see the influence of the correlation on the output range, the correlation coefficient (r) between the two variables is set from 0 to 0.75. The range interval is computed using the PCE with the Nataf transform. As it can be seen in Figure 5.8, the size of the interval increases with the correlation of the variables. In order to find range intervals that are adapted for the application, the correlation must be taken into account. Otherwise, either the range will be overestimated or it will not guarantee the performance requirements. The result obtained using IA (as in the first example, the support of the inputs is cut to the maximal and minimal values found by simulation) is also presented. The size of the interval is largely increased compared to our approach.

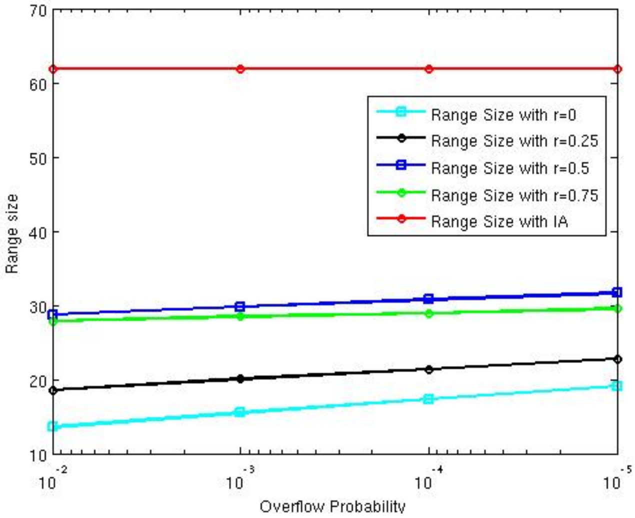


Figure 5.8: Range variation with the correlation and the overflow probability

The CDF and the PDF of the result are estimated from a 10^7 samples simulation. The Kolmogorov-Smirnov statistic is computed between the empirical CDF ($F_{2n}(x)$) and the CDF obtained using PCE ($F_{1n}(x)$) (with and without the Nataf transform). It measures the maximal difference between the two distributions:

$$D = sup_x|F_{1n}(x) - F_{2n}(x)| \tag{5.40}$$

The results are presented in Table 5.5. It shows that the distance (D) between the distributions obtained using the Nataf transform is smaller with at least one order of magnitude. Furthermore, the PDFs are presented in Figure 5.9 for $r = 0.75$. The PDF obtained using the Nataf transform approaches more accurately the histogram obtained by simulation. If the independence property is assumed, the interval that is obtained will not correspond to the real overflow probability.

	r=0	r=0.25	r = 0.5	r =0.75
D_{with_Nataf}	—	0.0029	0.0022	0.0021
$D_{without\_Nataf}$	3.253*10-4	0.0306	0.0565	0.0795

Table 5.5: Kolmogorov-Smirnov Statistic test

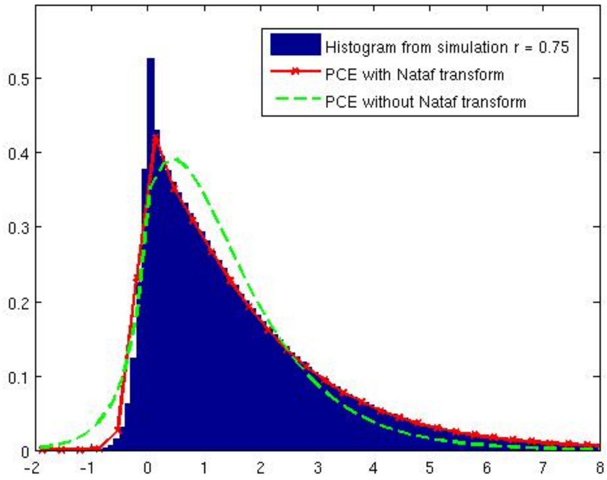


Figure 5.9: PDF comparison for y with r=0.75

5.7 The Askey scheme

It has been shown that the Hermite Polynomial Chaos in terms of Gaussian random variables is the best way to represent a Gaussian distribution. However, for the non-Gaussian case, the convergence rate may be slow. A generalization of the original Wiener Chaos has been introduced [80] to solve the problem and provide a more efficient representation for the non-Gaussian processes. This generalized polynomial chaos (gPC) uses several types of orthogonal polynomials from the Askey scheme that are optimal for different types of distributions.

The representation becomes:

$$\begin{aligned} x(\theta) = a_0 I_0 + \textstyle\sum_{i_1=1}^{\infty} a_{i_1} I_1(\zeta_{i_1}(\theta)) + \\ + \textstyle\sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2} I_2(\zeta_{i_1}(\theta), \zeta_{i_2}(\theta)) \\ + \textstyle\sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} I_3(\zeta_{i_1}(\theta), \zeta_{i_2}(\theta), \zeta_{i_3}(\theta)) + \cdots \end{aligned} \tag{5.41}$$

In this case I_n are the Wiener-Askey polynomials of order n in terms of the random vector $\boldsymbol{\zeta} = \{\zeta_{i_1}(\theta), \zeta_{i_2}(\theta), \cdots, \zeta_{i_n}(\theta)\}$. The correspondence between them is given by the Askey scheme.

	Distribution	gPC basis Polynomials	Support
Continous	Gaussian	Hermite	$(-\infty, \infty)$
	Gamma	Laguere	$[0, \infty]$
	Beta	Jacobi	$[a, b]$
	Uniform	Legendre	[a,b]
Discrete	Poisson	Charlier	$\{0, 1, 2, \cdots, N\}$
	Binomial	Krawtchouk	$\{0, 1, 2, \cdots, N\}$
	Negative Binomial	Meixner	$\{0, 1, 2, \cdots, N\}$
	Hypergeometric	Hahn	$\{0, 1, 2, \cdots, N\}$

Table 5.6: The Askey scheme

5.7.1 Legendre Chaos

From all the polynomials in the Askey scheme, a very useful family is the Legendre Chaos, which is optimal for the representation of the uniform distribution. This means that it can approximate the distributions that have a finite support with only a few terms. A uniform distribution is represented with only 1 term (Figure 5.10).

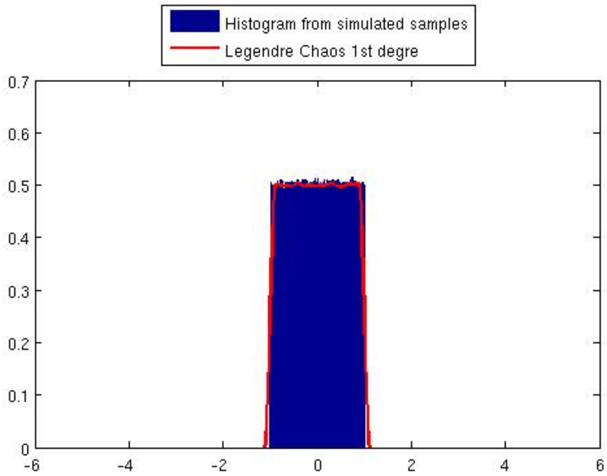


Figure 5.10: Uniform random variable representation with the Legendre Chaos

On the other hand, as the Legendre Chaos has a finite support it cannot represent accurately the long tails of the Gaussian distribution. This aspect can be seen on the PDF approximation in Figure 5.11.

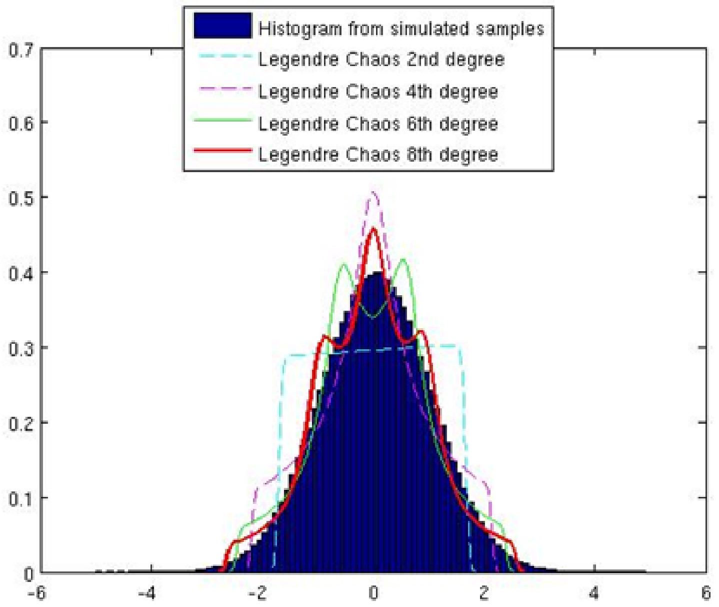


Figure 5.11: Gaussian random variable representation with the Legendre Chaos

An adaptive polynomial chaos methodology for the range evaluation can be used. Depending on the probability distribution of the input, an appropriate polynomial chaos should be chosen in order to optimize the number of terms that are needed for an accurate representation. The computation of the coefficients for the gPC can be made using a Galerkin projection approach as proposed in [80]. Further the methodology remains similar to the classical case of the Hermite Chaos.

5.8 Numerical Accuracy Evaluation

The proposed approach can also be applied to the numerical accuracy evaluation. The quantization noise model proposed by Widrow [72] is adopted in this Section. Every quantization operation introduces an independent source of noise modeled as an uniform random variable. As a result, the example of a quantized system presented in Figure 5.12 is transformed into an equivalent version presented in Figure 5.13, where every quantization operations is replaced with a noise source q_i.

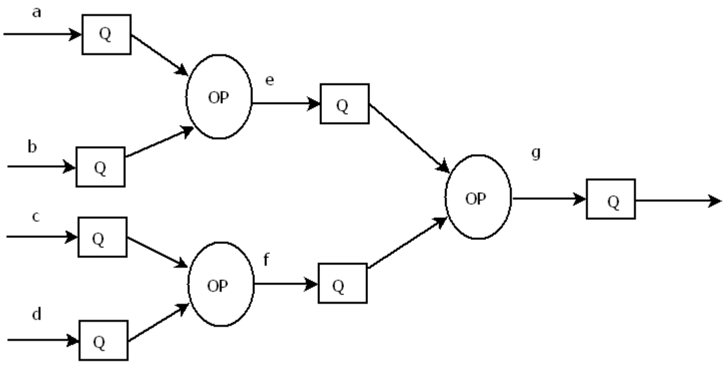


Figure 5.12: Example of quantized system

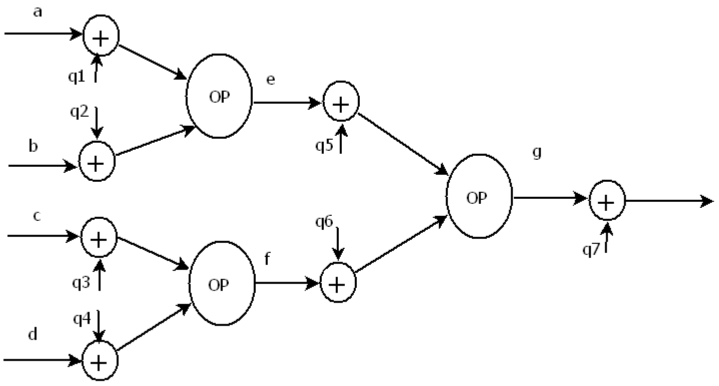


Figure 5.13: Transformed system

Using the projection method presented in Section 5.3.1, a 1-dimensional PCE can be obtained for every quantization operation. As the quantization noise has a uniform distribution, the Legendre polynomials are the optimal representation. Using the Legendre Chaos only one term is needed to represent the quantization noise, while 4 or 5 terms (depending on the accuracy needed) should be used with the Hermite Chaos.

For linear operations, the superposition property can be applied and the quantization noise is analyzed separately from the signal. For M quantization operations, M 1-dimensional PCE are created and the output noise is computed in exactly the same manner as for the case of range estimation, obtaining an M-dimensional PCE.

Let the input quantization noise be: $q_i = \sum_{j=0}^{P_i} q_{i_j} \Psi_j(\xi_i)$.

The output will then be: $q_{out} = \sum_{j=0}^{P} q_{out_j} \Psi_j(\xi_1, \xi_2, \cdots, \xi_M)$.

For non-linear systems, the superposition property cannot be applied anymore and signal values will influence the output noise. As a result, both the signal and the quantization noise will appear in the PCE computation, but the procedure remains very similar.

From the a computational point of view, the numerical accuracy and the range estimation problem are in fact equivalent. The output of the system is the result of arithmetic operations with the operands represented with PCEs.

5.9 Conclusion

In this Chapter we have presented a method for the range evaluation based on the Polynomial Chaos Expansion. It has been shown that the complete probabilistic description of the input can be obtained by the means of the PCE. The case of correlated variables has also been treated using the Nataf transform. The variability is statically propagated through the Data-Flow Graph from the input to the output and the analytical representation for all the variables is obtained. As opposed to the KLE, this method can be applied to any type of system that is composed of arithmetic operations making it possible to treat non-linear systems.

Using the same probabilistic methodology that has been introduced for the KLE method, the range is computed from the PDF with respect to a desired overflow probability. The results show that the obtained distributions are close to the simulation results. Furthermore, using our probabilistic analysis, the size of the range intervals is significantly reduced compared to the IA method.

The generalized polynomial chaos has been presented in order to select an appropriate polynomial chaos basis depending on the distribution of the input signal.

Chapter 6

Conclusions and Perspectives

Conclusions

In this thesis, a probabilistic approach for the dynamic range evaluation has been developed in the context of wordlength optimization. In order to avoid over-dimensioning the system, a trade-off between the dynamic range that is covered by the fixed-point representation and the cost of the implementation has to be made. For applications that accept occasional overflows if their probability of occurrence is small, the integer part wordlength can be reduced without covering the entire theoretical range.

First, the case of linear-time invariant systems was addressed. The Karhunen-Loev̀e Expansion (KLE) was used as a means of representing the variability of the input signal. As opposed to the method based on simulation presented in [79], we showed how the variability can be statically propagated through LTI systems obtaining the corresponding output representation using the impulse response of the system. The range is further computed from the PDF with respect to a coverage probability.

The same KLE discretization approach was also applied to evaluate the quantization noise. The application SQNR is estimated from the quantization noise variance. In addition, the complete noise PDF can be computed. The method has been developed in Matlab and has already been integrated in the automatic conversion tool of the CAIRN team.

Next, we have presented a method for the range estimation based on the Polynomial Chaos Expansion (PCE). As a first step, the random behavior of the input is represented in the form of a PCE. We showed how the PCE can be adapted to treat the case of random variables. Furthermore, the case of correlated inputs has also been covered using the Nataf transform. The variability of the input is statically propagated through the data-flow graph and the analytical representation of the output is obtained. As opposed to the KLE, this method can be applied to any type of system that is composed of arithmetic operations making it possible to treat non-linear systems. The range is computed from the PDF with a probabilistic analysis in similar manner to the KLE method. In comparison to the KLE method it has an increased complexity so its applicability to LTI systems is less interesting.

Furthermore, the generalized Polynomial Chaos has been introduced and it has been shown how the type of the polynomial chaos can be chosen depending on the distribution of the input in order to reduce the number of terms that need to be used for an accurate representation. Finally, the numerical accuracy evaluation can be done using the same method. All the development has been done in Matlab and has not been integrated into the automatic conversion tool yet.

Perspectives

The number of terms that are used for an accurate PCE representation can significantly increase with the dimension and the order. For large non-linear applications this can become a prohibitive factor in the process of automatization. As a future work, the complexity should be reduced by using only a sparse structure of polynomials that provides only the most important terms in the expansion while neglecting the others.

Another important aspect that should be considered is the implementation of an adaptive polynomial chaos based the Askey scheme. As it was presented, the classical polynomial chaos that employs the Hermite polynomials is optimal only for the representation of the Gaussian distribution. For highly non-Gaussian PDFs, the convergence rate may be low and an important number of terms is needed. An adaptive polynomial chaos that automatically modifies its basis polynomials depending on the distribution of the input can significantly reduce the complexity and should be implemented in the future.

Personal publications

- A. Banciu, E. Casseau, D. Menard, T. Michel, “ Dynamic range evaluation using the polynomial chaos expansion and the Nataf transform ”, submitted to IEEE International Symposium on Circuits and Systems, ISCAS 2012, Seoul, Korea, May 20-23 2012.

- A. Banciu, E. Casseau, D. Menard, T. Michel, “ Stochastic Modeling for Floating-point to Fixed-point Conversion”, IEEE Workshop on Signal Processing Systems, SiPS 2011, Beirut, Lebanon, October 4-7 2011.

- A. Banciu, E. Casseau, D. Menard, T. Michel, “ A Case Study Of The Stochastic Modeling Approach For Range Estimation”, Conference on Design and Architectures for Signal and Image Processing, DASIP 2010, Edinburgh, UK, pp.301-308, October 26-28, 2010.

Bibliography

[1] A. Banciu, E. Casseau, D. Menard, T. Michel, "A Case Study Of The Stochastic Modeling Approach For Range Estimation", *Proc. DASIP Conf*, pp. 301-308, Oct. 2010.

[2] A. Banciu, E. Casseau, D. Menard, T. Michel, " Stochastic Modeling for Floating-point to Fixed-point Conversion", IEEE Workshop on Signal Processing Systems, SiPS 2011, Beirut, Lebanon, October 4-7 2011.

[3] M. Barberis and N. Shah, "Migrating signal processing applications from floating-point to fixed-point", White paper, Catalytic Inc, Palo Alto, USA, November 2004.

[4] A. Benedetti and P. Perona, "Bit-width optimization for configurable DSPs by multi-interval analysis", in Proc. Asilomar Conf. Signals, Syst. And Comp., vol. 1, pp 355-359, 2000.

[5] S. Blinnikov and R. Moessner, "Expansions for nearly Gaussian distributions", Astronomy and astrophysics suplement series 130:193-205, May 1998.

[6] R. Cameron and W. Martin, "The orthogonal development of nonlinear functionals in series of Fourier-Hermite functionals", Ann. Math., 48, p. 385, 1947.

[7] J. Carletta, R. Veillette, F. Krach and Z. Fang, "Determining appropriate precision for signals in fixed point IIR filters", *Proc. Design Automation Conf.*, Anaheim, CA, pp.656-661, Jun 2-6 2003.

[8] C. Carreras, J. A. Lopez et al., "Bit-width selection for data-path implementations", in Proc. Int. Symp. Syst. Synthesis, pp. 114-119, 1999.

[9] C. Chatfield, "The Analysis of Time Series: An Introduction", Chapman and Hall, London, 4th edition, 1989.

[10] A. Chapoutot, L.-S. Didier, Fanny Villers, "Range Estimation of Floating-Point Variables in Simulink Models", NSV-II : Second InternationalWorkshop on Numerical Software Verification, 2009.

[11] M. Clark, M. Mulligan, D. Jackson, and D. Linebarger, "Accelerating Fixed-Point Design for MB-OFDM UWB Systems", *CommsDesign*, January 2005.

[12] J. A. Clarke, G. A. Constantinides, P. Y. K. Cheung, "Wordlength Selection for Power Minimization via Nonlinear Optimization", *ACM Trans. Ob Des. Aut. Of Electr. Syst.*, vol. 14, no. 3, art. 39, May 2009.

[13] J. Cong et al., "Evaluation of static analysis techniques for fixed-point precision optimization", *Proc. 17th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 231-234, April 2009.

[14] B. J. Debusschere, H. N. Najm, P. P. Pebay, O.M. Knio, R. G. Ghanem, and O. P. LeMaitre, "Numerical challenges in the use of polynomial chaos representations for stochastic processes", SIAM J. Sci. Comput., 26(2):698719, 2004.

[15] L.-S. Didier, "A statistical method of range estimation for embedded applications", 13th International Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computations - SCAN2008, p. 34-35, 2008.

[16] O. Ditlevsen and H.O. Madsen, "Structural Reliability Methods", Internet edition 2.3.7 http://www.web.mek.dtu.dk/staff/od/books.htm. June-September, 2007.

[17] A. Dutfoy and R. Lebrun, “Modelisation de la dependence par la theorie des copules: une generalization de la transformation de Nataf”, 18eme Congres Francais de Mecanique, aout 2007.

[18] C. F. Fang, R. B. Rutenbar, M. Puschel, T. Chen, “Towards Efficient Static Analysis of Finite-Precision Effects in DSP Applications via Affine Arithmetic Modeling”, *Proc. Design Automation Conference*, pp. 496-501, Jun. 2-6, 2003.

[19] C. Fang Fang, T. Chen, and R.A. Rutenbar, “Floating Point Error Analysis based on Affine Arithmetic”, *In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP03)*, pages 561564, 2003.

[20] R. V. Fields Jr, “Numerical methods to estimate the coefficients of polynomial chaos expansion”, 15th ASCE Engineering Mechanics Conference, June 2002.

[21] L. H. de Figuereido and J. Stolfi, “Self-Validated numerical methods and applications”, Brazilian Mathematics Colloquium monographs, IMPA, Rio de Janeiro, Brazil, Jul. 1997.

[22] L. H. de Figuereido and J. Stolfi, “Affine arithmetic: concepts and applications”, Numerical Algorithms vol. 37,Numbers 1-4, pp. 137-158, Dec. 2004.

[23] http://www.forteds.com/

[24] R. G. Ghanem and P. D. Spanos, “Stochastic Finite Elements: A Spectral Approach”,pringer Verlag, 1991.

[25] R. G. Ghanem and P. D. Spanos, “Stochastic Finite Elements: A Spectral Approach”,Revised Edition, Dover Publications, 2003.

[26] T. Grtker, E. Multhaup, and O.Mauss, “Evaluation of HW/SW Tradeoffs Using Behavioral Synthesis”, *In 7th International Conference on Signal Processing Applications and Technology (ICSPAT96)*, Boston, October 1996.

[27] Emil, J. Gumbel, “Statistics of Extremes”, Columbia University Press, 1958.

[28] N. Hervé, D. Menard, and O. Sentieys, “Data wordlength optimization for fpga synthesis”, *In IEEE International Workshop on Signal Processing Systems (SIPS05)*, pages 623628, Athens, Grece, November 2005.

[29] T. Hill, “AccelDSPs synthesis tool floating-point to fixedpoint conversion of matlab algorithms targeting FPGAs”, *Xilinx, White papers*, April 2006.

[30] S. Hussain, “Peak to Average Power Ratio Analysis and Reduction of Cognitive Radio Signals”, PhD Thesis, Universit de Rennes I, Oct., 2009.

[31] L. B. Jackson, “On the roundoff noise and dynamic range in digital filters”, Bell Syst. Tech. J., vol. 49, pp 159-184, Feb. 1970.

[32] H. Keding, M. Willems, M. Coors, and H. Meyr, “FRIDGE: A fixedpoint design and simulation environment”, *CMConf. on Design,Automation and Test in Europe 1998*, Paris, France, pp. 429435, Mar. 1998.

[33] S. Kim and W. Sung, “A floating-point to fixed-point assembly program translator for the TMS320C25”, *IEEE Trans. Circuits Syst. II*, vol. 41, pp. 730-739, Nov. 1994.

[34] S. Kim and W. Sung, “Fixed-Point-Simulation Utility for C and C++ Based Digital Signal Processing Programs”, *In Twenty-eighth Annual Asilomar Conference on Signals, Systems, and Computer*, October 1994.

[35] S. Kim, K. Kum and S. Wonyang, “Fixed-Point Optimization Utility for C and C++ Based Digital Signal Processing Programs”, *IEEE Transaction on Circuits and Systems II*, vol. 45, no. 11, Nov. 1998.

[36] S. Kim and W. Sung, “Fixed-Point Error Analysis and Word Length Optimization of 8x8 IDCT Architectures”, *IEEE Transactions on Circuits and Systems for Video Technology*, 8(8):935940, December 1998.

[37] K.-I. Kum, W. Sung, “Combined word-length optimization and high level synthesis of digital signal processing systems”, *IEEE Trans. On CAD Of Integr. Circ. And Syst.*, vol. 20, No. 8, Aug, 2001.

[38] N. Lagaros, G. Stefanou and M. Papadrakakis, “An enhanced hybrid method for the simulation of highly skewed non-Gaussian stochastic fields”, Comput. Methods Appl. Mech. Engrg. 194, 4824-4844, 2005.

[39] D.-U. Lee et al., “Accuracy-guaranteed bit-width optimization”, *IEEE Trans. On CAD Des. Of Integr. Circ. And Syst.*, vol. 25, No. 10, Oct., 2006.

[40] C-C Li and A. Der Kiureghian, “Optimal discretization of random fields”, J. Eng. Mech., 119(6) :11361154, 1993.

[41] M. Loève, “Probability Theory”, fourth ed., Springer-Verlag, Berlin 1977.

[42] J. A. Lopez et al, “Fast characterization of the noise bounds derived from coefficient and signal quantization”, in Proc. ISCAS, vol. 4, pp 309-312, 2003.

[43] J. A. Lopez, C. Carreras, O. Nieto-Taladriz, “Improved interval-based Characterization of fixed-point LTI systems with feedback loops”, in IEEE Trans. on CAD of Circ. and Syst., vol. 26, no. 11, pp. 1923-1933, Nov. 2007.

[44] http://www.mathworks.com/

[45] D. Menard and O. Sentieys, “ Automatic Evaluation of the Accuracy of Fixed-point Algorithms”,In Design, Automation and Test in Europe 2002 (DATE02), Paris, March 2002.

[46] D. Menard and O. Sentieys, “ A methodology for evaluating the precision of fixed-point systems”,In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP02), Orlando, May 2002.

[47] D. Menard, “Methodologie de compilation d’algorithmes de traitement du signal pour les processeurs en virgule fixe, sous contrainte de precision”, PhD thesis, Universite de Rennes I, Lannion, Dcembre 2002.

[48] D. Menard, D. Chillet, and O. Sentieys, “Floating-to-fixed-point conversion for digital signal processors”, 2006:Article ID 96421, 25 pages, 2006.

[49] D. Menard, R. Rocher, O. Sentieys, “Analytical Fixed-Point Accuracy Evaluation In Linear-Time Invariant Systems”, *IEEE Trans. On Circ. And Syst.I:Regular Papers*, vol.55, no.1, Nov. 2008.

[50] http://www.mentor.com/esl/

[51] R. E. Moore, “Interval Analysis” Prentice-Hall, 1966.

[52] A. Nataf, “Determination des distributions dont les marges sont donnees”, Comptes rendus de lAcademie des Sciences, 225 :42-43, 1962.

[53] R. B. . Nelsen, “An Introduction to Copulas”, New York: Springer, 1999.

[54] B. Van den Nieuwenhof, “Stochastic Finite Elements For Elastodynamics: Random field and shape uncertainty modelling using direct and modal perturbation-based approaches”, RE Transactions on Circuit Theory, PhD thesis, Universite catholique de Louvain, 2003.

[55] Hideki Ochiai and Hideki Imai, “On the Distribution of the Peak-to-Average Power Ratio in OFDM Signals”, in IEEE Trans. on Communications., vol. 49 no. 2, pp. 282-289, Feb. 2001.

[56] E. Ozer, A. P. Nisbet and D. Gregg, “Stochastic bit-width approximation using extreme value theory for customizable processors”, CC 2004, LNCS 2985, pp. 250264, 2004.

[57] E. Ozer, A. P. Nisbet and D. Gregg, "A stochastic bitwidth estimation technique for compact and low-power custom processors", *ACM Trans. On Embedded Comput. Syst.*, vol. 7, no. 3, pp.1-30, Apr. 2008.

[58] K. Parashar, Romuald Rocher, Daniel Menard, Olivier Sentieys, "Analytical approach for analyzing quantization noise effects on decision operators", *ICASSP 2010*, pp. 1554-1557, 2010.

[59] E. Parzen, "On estimation of a probability density function and mode", Ann. Math. Stat. 33 (3), 1065-1076, 1962.

[60] J. G. Proakis, "Digital Communications", 4th ed., McGraw Hill, 2000.

[61] M. Rosenblatt, "Remarks on multivariate transformation", The Annals of Mathematical Statistics. v23 i3. 470-472, 1952.

[62] C. Shi and R.W. Brodersen, "An automated floating-point to fixed-point conversion methodology", *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP03)*, Vol. 2, pp. 529-532, April 2003.

[63] C. Shi and R.W. Brodersen, "A perturbation theory on statistical quantization effects in fixed-point DSP with non-stationary inputs",*In IEEE International Symposium on Circuits and Systems (ISCAS04)*, 2004.

[64] A. Singhee, C. F. Fang, J. D. Ma, and R. A. Rutenbar, "Probabilistic interval-valued computation : Toward a practical surrogate for statistics inside CAD tools", *Proc. IEEE/ACM Des. Autom. Conf. (DAC)*, pp167-172, Jul. 2006.

[65] A. Singhee, C. F. Fang, J. D. Ma, and R. A. Rutenbar, "Probabilistic Interval-Valued Computation: Toward a Practical Surrogate for Statistics Inside CAD Tools", *IEEE Trans. on CAD of Integrated Circuits and Systems* 27(12): 2317-2330 (2008).

[66] B. Pinon, "Une introduction Scilab", ESIAL Nancy, April 2006.

[67] B. Sudret and A. Der Kiureghian, "Stochastic finite elements and reliability : A state-of-the-art report. Technical Report no UCB/SEMM-2000/08", University of California, Berkeley, 2000.

[68] B. Sudret, M. Berveiller, M. Lemaire, "A stochastic finite element procedure for moment and reliability analysis", Eur. J. Comput. Mech., Vol. 15 (7-8), pp. 825866, 2006.

[69] W. Sung. and K.-I. Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems", *IEEE Trans. Sig. Proc. 43*, 12, 3087-3090, 1995.

[70] http://www.synopsys.com

[71] http://www.synopsys.com/Systems/BlockDesign/HLS/

[72] B. Widrow, "A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory", RE Transactions on Circuit Theory, CT-3(4):266276, Dec. 1956.

[73] B. Widrow, "Statistical Analysis of Amplitude Quantized Sampled-Data Systems", Trans. AIEE, Part. II:Applications and Industry, 79:555-568, 1960.

[74] B. Widrow, I. Kollar and M.-C. Liu, "Statistical Theory of Quantization", in IEEE Trans. on Instrumentation and measurement, vol. 45, no. 2, april, 1996.

[75] B. Widrow and I. Kollar, "Quantization Noise: Round Off Error in Digital Computation", Signal Processing, Control and Communications, Cambridge University Press, 2008.

[76] N. Wiener, 'The homogeneous chaos", Amer. J. Math., 60 (1938), pp. 897-936.

[77] http://www.wolfram.com/

[78] B. Wu, J. Zhu, F.Najm, "An analytical approach for dynamic range estimation", *Proc. Design Automation Conference*, June 7-11, 2004.

[79] B. Wu, J. Zhu, F.Najm, “Dynamic Range Estimation”, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 5(9):16181636, September 2006.

[80] D. Xiu and G. E. Karniadakis, “The Wiener-Askey polynomial chaos for stochastic differential equations”, SIAM Society for Industrial and Applied Mathematics, 24(2):619644, 2002.

[81] L. Zhang, Y. Zhang, W. Zhou, “Floating-point to fixed-point transformation using extreme value theory”, *in Proc. of the Eigth IEEE/ACIS International Conference on Computer and Information Science*, pp. 271-276, 2009.

Resumé

Les applications de traitement du signal ont connu un très fort développement dans les dernières décennies, bénéficiant des avanceés majeures de l'industrie des semi-conducteurs. Toutes les implémentations pratiques utilisent l'arithmétique en virgule fixe afin de réduire la surface et la consommation d'énergie. En conséquence, une conversion de la description en virgule flottante de l'algorithme à une implémentation en virgule fixe qui ajuste la largeur du chemin de données doit être réalisée. C'est un processus d'optimisation qui consiste à trouver les partie fractionnaire (évaluation de la précision numérique) et entière (estimation de la dynamique) minimales qui satisfassent les contraintes de performance.

Dans cette thèse, une approche stochastique pour l'évaluation de la dynamique des données est présentée. Notre objectif est d'obtenir une représentation complète de la variabilité qui intègre le comportement probabiliste et non seulement les limites maximales et minimales. Une méthode basée sur le développement de Karhunen-Loève est développée pour le cas des systèmes linéaires et invariants dans le temps. Ensuite, le développement du chaos polynomial est introduit afin de traiter des opérations non-linéaires. Les méthodes sont appliquées à l'optimisation de la taille de données quand une légère dégradation des performances est acceptable. La dynamique retenue ne couvre plus tout l'intervalle théorique de variation : des débordements sont autorisés avec une contrainte quant à leur probabilité d'apparition. Les signaux qui ont des variations importantes de leur amplitude sont approximées avec des intervalles serrés pour réduire le coût de l'implémentation.

Abstract

Digital Signal Processing (DSP) applications have experienced a very strong development in the last decades, benefiting from the major advances of the semiconductor industry. All practical DSP implementations use fixed-point arithmetic to reduce the area and power consumption and obtain a cost-effective hardware. As a consequence, a conversion from the floating-point description of the algorithm to a fixed-point implementation that adjusts every bit-width in the datapath must be realized. This is an optimization process that consists in finding the minimal fractional part (numerical accuracy evaluation) and integer part (range estimation) wordlengths that still satisfy the performance constraints.

In this thesis a stochastic approach for the range evaluation is presented. Our goal is to obtain a complete representation of the variability that incorporates the probabilistic behaviour and not only the maximal and minimal bounds. A method based on the Karhunen-Love Expansion is developed at the beginning for the case of linear time-invariant systems. Furthermore, the Polynomial Chaos Expansion is introduced in order to treat non-linear operations. The methods are applied to the optimization of the integer part wordlength when a slight degradation of the performances is acceptable. The range doesn't cover anymore the entire theoretical interval of variation, instead the occurrence of overflows is authorized with a constraint regarding their probability of appearance. Signals that have high variations of their amplitude are approximated with tight intervals so that the implementation cost can be reduced.

Index Terms

Range estimation, accuracy evaluation, fixed-point arithmetic, Karhunen-Loève Expansion, Polynomial Chaos Expansion, digital signal processing systems