Design of a Financial Application Driven Multivariate Gaussian Random Number Generator for an FPGA

Chalermpol Saiprasert, Christos-Savvas Bouganis and George A. Constantinides

Department of Electrical & Electronic Engineering, Imperial College London
Exhibition Road, London SW7 2BT, United Kingdom.
{cs405,christos-savvas.bouganis,g.constantinides}@imperial.ac.uk

Abstract. A Multivariate Gaussian random number generator (MVGRNG) is a pre-requisite for most Monte Carlo simulations for financial applications, especially those that involve many correlated assets. In recent years, Field Programmable Gate Arrays (FPGAs) have received a lot of attention as a target platform for the implementation of such a generator due to the high throughput performance that can be achieved. In this work it is demonstrated that the choice of the objective function employed for the hardware optimization of the MVRNG core, has a considerable impact on the final performance of the application of interest. Two of the most important financial applications, Value-at-Risk estimation and option pricing are considered in this paper. Experimental results have shown that the suitability of the chosen objective function for the optimization of the hardware MVRNG core depends on the structure of the targeted distribution. An improvement in performance of up to 96% is reported for VaR calculation while up to 81% improvement is observed for option pricing when a suitable objective function for the optimization of the MVRNG core is considered while maintaining the same level of hardware resources.

1 Introduction

Monte Carlo simulation plays an important role in many scientific applications, one of which is financial mathematics. The multivariate Gaussian distribution is a pre-requisite for such simulations as it captures the correlation between sources of uncertainties that affect the values of the financial instruments. As the number of financial instruments continues to increase, the computation of these simulations has been intensified. Field Programmable Gate Arrays (FPGAs) have been demonstrated to be a good candidate for the acceleration of random number generators due to their fine grain parallelism, and many works have been presented in recent years on the acceleration of many financial applications [1],[2]. However, to the best of the authors' knowledge, no published work in the literature has addressed the question regarding the relative performance of the various optimization objective functions in the design of a random number generator block focusing on the impact that this has on the performance of a financial application.

In this work, we focus on the implementation of the Multivariate Gaussian random number generator (MVGRNG) on an FPGA platform and we investigate the impact of the design decisions taken for the optimization of the MVGRNG to the performance of a financial application. The application under investigation are the estimation of the Value-at-Risk (VaR) of a financial portfolio and option pricing, two of the most widely used applications in financial industry. Existing approaches in the literature regarding an FPGA based MVGRNG can be found in [3], [4] and [5]. The work presented in this paper is based on the framework proposed in [5], as it can accommodate any objective function for the hardware optimization of a MVGRNG design. Three design criteria are proposed for the design of a MVGRNG targeting an FPGA device, and their impact to the estimation of VaR calculation and option pricing are investigated for a set of hardware resources.

2 Related Work

The first FPGA-based multivariate Gaussian random number generator was presented in [3]. The authors decompose the input correlation matrix, which encapsulates the correlation of the distribution of interest, using Cholesky decomposition in order to take advantage of the lower triangular property of the resulting matrix. The resulting design is able to serially generate a vector of multivariate Gaussian random numbers every N clock cycles, where N denotes the dimensionality of the distribution. In [3], DSP48 blocks are used for the implementation of a MVGRNG on an FPGA platform, requiring N blocks for an N-dimensional Gaussian distribution. However, the drawback of this approach is the restriction in resource allocation since the dimensionality of the distribution dictates the number of DSP48 blocks to be utilized. In their approach, the minimization of the mean square error between the approximated correlation matrix and the target one is implemented.

An alternative method which addresses the problem encountered in [3] is presented in [4]. An algorithm, based on the use of Singular Value Decomposition, is introduced to approximate the lower triangular matrix, the result of applying Cholesky decomposition on the correlation matrix, by trading off the error in the approximation of the input correlation matrix for an improved resource usage. The approach in [4] requires $2K$ DSP48 blocks to produce a vector of size N, where K denotes the number of decomposition levels required to approximate the lower triangular matrix while maintaining the same throughput as in [3]. In addition to an improved resource utilization, [4] offers the flexibility to produce a hardware system that meets any given resource constraint. However, it has been shown in [4] that allocating a fixed precision to all of the computation paths of the architecture does not lead to the optimum resource utilization.

In [5], the precision issue mentioned above has been exploited and word-length optimization techniques have been introduced to produce an architecture with multiple precisions in its datapath. The algorithm presented in [5] further reduces the required hardware resources in comparison to [3] and [4]. In addition, an analysis of the correlation of errors due to truncation operations in the computation datapath has been presented and its effects have been modeled in the objective function targeting the optimum usage of the hardware resources.

To the best of the authors' knowledge, the existing approaches in the literature have not investigated the impact of the objective function employed in the optimization of the MVGRNG core on the performance of the application that requires such a block and on the hardware requirements of the core. The authors in [3] carried out an evaluation of the performance of their approach for a financial application, the Delta-Gamma asset simulator, but the optimization of the hardware architecture of a MVGRNG over different objective functions has not been considered. This work considers two of the most important financial applications involving Monte Carlo simulation, the estimation of the Value-at-Risk (VaR) and option pricing.

3 Generating Multivariate Gaussian Random Samples

Following from [5], in order to generate random samples from a given multivariate Gaussian distribution with mean $\mathbf{m}$ and correlation $\boldsymbol{\Sigma}$, the eigenvalue decomposition technique is used [6]. In this technique, an algorithm known as Singular Value Decomposition is used to decompose the correlation matrix $\boldsymbol{\Sigma}$. As a result of the decomposition, $\boldsymbol{\Sigma}$ can be expressed as a linear combination of three separable matrices $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ where $\mathbf{U}$ is an orthogonal matrix ($\mathbf{U}\mathbf{U}^T = \mathbf{I}$) containing eigenvectors $\mathbf{u}_1, ..., \mathbf{u}_N$ while $\boldsymbol{\Lambda}$ is a diagonal matrix with diagonal elements being eigenvalues $\lambda_1, ..., \lambda_N$. If we let $\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}$, multivariate Gaussian random samples that follow $N(\mathbf{m}, \boldsymbol{\Sigma})$ can be generated as in (1), where $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$.

$$\begin{aligned}
\mathbf{x} &= \mathbf{A}\mathbf{z} + \mathbf{m} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}\mathbf{z} + \mathbf{m} \\
&= (\sqrt{\lambda_1}\mathbf{u}_1 z_1 + \sqrt{\lambda_2}\mathbf{u}_2 z_2 + ... + \sqrt{\lambda_K}\mathbf{u}_K z_K) + \mathbf{m} \\
&= \sum_{i=1}^{K} (\sqrt{\lambda_i}\mathbf{u}_i z_i) + \mathbf{m}. \qquad (1)
\end{aligned}$$

The full representation of the original correlation matrix $\boldsymbol{\Sigma}$ can be achieved if the number of decomposition levels K is equal to the rank of the correlation matrix. Thus, the correlation matrix of the original distribution can be approximated by taking into account K levels of decomposition where $K \leq rank(\boldsymbol{\Sigma})$.

4 Hardware Architecture

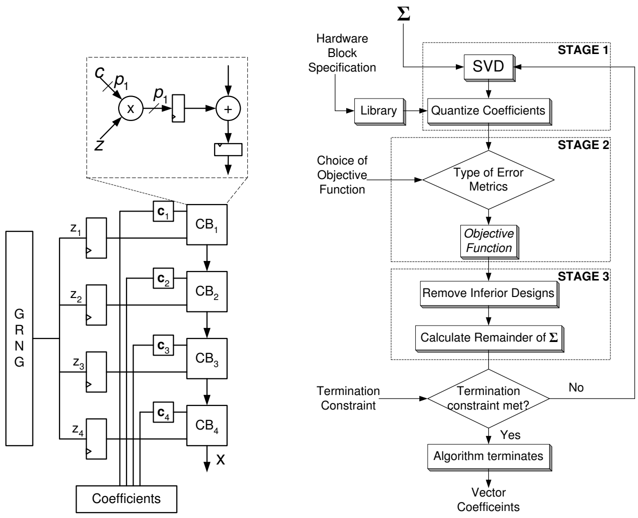
The proposed hardware architecture for a multivariate Gaussian random number generator for an FPGA implementation is based on the eigenvalue decomposition technique. The generation of random samples from a centralized Gaussian distribution, that is a distribution with zero mean, will be the main focus for the remainder of this paper. Any other non-centralized distribution with the same correlation can be produced by a simple offset of the generated vectors. From (1), the correlation matrix under consideration $\boldsymbol{\Sigma}$ is expressed as $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$. Hence, the random samples $\mathbf{x}$ can be generated as $\mathbf{x} = \sum_{i=1}^{K} \sqrt{\lambda_i}\mathbf{u}_i z_i \approx \sum_{i=1}^{K} \mathbf{c}_i z_i$, where $\mathbf{c}_i$ denotes a product of $\sqrt{\lambda_i} \cdot \mathbf{u}_i$ after a quantization step with the desired word-length.

In this work, the multivariate Gaussian samples are generated from the sum of products of $\mathbf{c}$ and z and, thus, the various decomposition levels i are mapped onto computational blocks (CB) designed for an FPGA. Each CB performs the multiply-add operation. The architecture is mapped to logic elements only, so that the precision of the datapath can be varied. Using the proposed approach, a MVGRNG design is constructed from any combinations of CBs. Figure 1(a) depicts an example of such architecture where four CBs are used. p_i denotes the precision of the datapath for each level of decomposition while the precision in the adder path p_T is fixed to the maximum precisions of all CBs. Independent univariate Gaussian samples z are produced from the GRNG block.

The multiply-add operation is pipelined so that all the computational blocks operate in parallel in order for an improved throughput. Fixed point number representation is deployed throughout the entire design as it produces designs which consume fewer resources and operate at a higher frequency in comparison to designs that use floating point arithmetic. Similar to other existing approaches in the literature, the elements of each random vector are serially generated resulting to a throughput of one vector per N clock cycles for an N dimensional Gaussian distribution [3] [4] [5].

5 Constructing MVGRNG Blocks

In this section, the methodology of mapping the hardware architecture described in Figure 1(a) to an FPGA is discussed. The main concept behind the methodology used in this work is the exploration of mixed precisions in the computational path of the architecture which was proposed in [4]. The high level overview of the proposed methodology is depicted in Figure 1(b). There are three main stages in the proposed methodology where, in the first stage, the correlation matrix $\boldsymbol{\Sigma}$ is decomposed into a vector $\mathbf{c}$ and its transpose using the Singular Value Decomposition (SVD) algorithm. The resulting vectors are converted into fixed point representation using one of the user-specified word-lengths from the hardware library. In the second stage, the appropriate coefficients are selected in order to minimize the error metric according to the selected objective function which will be described in the next section. The third stage of the proposed approach removes any inferior designs, that is designs which in comparison to another design, use more hardware resources but produce worse approximation error. The last step of this stage calculates the remainder of the original matrix $\boldsymbol{\Sigma}$ which is the starting point for the next iteration. These steps are repeated until the termination condition is met. Examples of the termination condition are a specified approximation error, a given resource constraint or the number



(a) Architecture of Multivariate Gaussian Random Number Generator.

(b) Overview of the Proposed Approach.

Fig. 1. Proposed Methodology

of decomposition levels to be used. The output of the proposed approach is a set of vectors of coefficients $\mathbf{c}_i$ which best approximate the input correlation matrix under a given norm.

The approximated correlation matrix $\overline{\boldsymbol{\Sigma}}$ can be decomposed as $\overline{\boldsymbol{\Sigma}} = \mathbf{C}_K + \mathbf{W}$, where $\mathbf{C}_K$, approximates the input correlation using K levels of decomposition taking into consideration the quantization effects and $\mathbf{W}$ is an expected truncation error matrix, which is a function of the correlation between the error due to truncation operations [5].

In summary, a framework to design a hardware architecture of a MVGRNG with the ability to accommodate different objective functions has been discussed. The coefficients of the correlation matrix of the distribution of interest are chosen based on the objective function of interest. The proposed approach provides an expectation of the generated correlation matrix given the coefficients in the various levels of the architecture taking into consideration all sources of errors injected into the system. It should be noted that the proposed system takes into account the correlation effect due to truncation error in the approximation of the correlation matrix. In the next section, the quality of the random samples produced based on the proposed methodology are used for the estimation of the Value-at-Risk of a financial portfolio and the pricing of options so the impact of the different objective functions on the two financial applications can be investigated.

6 Case Study: MVGRNG in Financial Applications

This section focuses on the two financial applications of interest namely, the estimation of Value-at-Risk of a portfolio and the pricing of an option. The first part of this section provides a background description of the two applications. The discussion is then shifted towards correlated assets where a multivariate Gaussian random number generator is deployed to model the correlation between those underlying assets. Three different objective functions are introduced for the hardware design of the MVGRNG, given a certain constraint on the available resources, and their impact on the performance of the two financial applications is investigated.

6.1 Value-at-Risk of a Financial Portfolio

A portfolio is a collection of financial investments such as stocks, bonds and options. Essentially, the purpose of holding a portfolio is to limit the risk while increasing the possibility of making profit, where the investment is spread over a variety of assets with different degree of risk factors. In order to quantify the expected loss of a portfolio with many correlated assets, the progression of the risk factors affecting this portfolio must be taken into account. Monte Carlo simulation is used to simulate the time evolution of the risk factors due to their stochastic nature. Let us consider a portfolio containing N assets. The market price of each asset is denoted by S_i having drift μ_i and volatility σ_i, where $i = 1 \ldots N$. The correlation between all of the assets in the portfolio under consideration is captured by a correlation matrix $\boldsymbol{\Sigma}$. The dynamics of the price of each asset are modeled as in (2).

$$\ln(S_i(t_E)) = \ln(S_i(t_0)) + \sum_{j=1}^{E}(\mu_i \delta t + x_j), \tag{2}$$

where x_j denotes a multivariate Gaussian random sample that follows $N(0, \boldsymbol{\Sigma})$. The time interval from t_0 to t_E is divided into intervals of length δt [7]. The path taken by the random walk algorithm during the simulation is of interest only for *path-dependent* derivatives such as an Asian-style derivative where the price at maturity is the average price of the path taken over a specified period. A European-style derivative, on the other hand, does not take this into account and the price at maturity is taken at the end of the specified period. Value-at-Risk (VaR) is a measurement used to evaluate a risk of loss of a specific portfolio and describes probabilistically the market risk of a trading portfolio by measuring the worst expected loss over a specific time interval at a given level of confidence.

6.2 Option Pricing

An option is a contract between a buyer and seller which permits a buyer, depending the type of option held, the right to purchase or sell a particular asset at an agreed price on or before the option's expiration time. In this work, the option called "Chooser option" is considered. The strike price, which is a price when the option is being exercised, of a Chooser option with three underlying assets is determined by the maximum of the sum of the two highest priced assets at the end of a specified period of time. Monte Carlo methods are used to calculate the value of these options where multiple underlying assets are involved and sources of uncertainty are captured by the multivariate Gaussian distribution.

6.3 Objective Functions

In this work, we propose three objective functions for the selection of coefficients and the precision of the computational paths for each decomposition level in the design of the MVGRNG core. The following notation is used for all three objective functions. $\mathbf{R}_{K-1}$ denotes the remaining of the original correlation matrix after $K-1$ levels of decomposition with $\mathbf{R}_0$ defined as the original correlation matrix at the start of the algorithm. The max(.) operator is an element-wise operation for a matrix of interest. Table 1 illustrates the three objective functions to be used in this work. The first objective function selects the coefficients based on the minimization of the maximum absolute error in the approximation of the correlation matrix while the second and third objective functions minimize the relative and mean square error in the correlation matrix approximation respectively. All three objective functions take into account the correlation effect between the truncation errors to the final correlation matrix.

Table 1. Objective Functions.

Error to be Minimized	Objective Functions
Maximum Absolute Error	$\max \left\| \mathbf{R}_{K-1} - (\mathbf{c}_K \mathbf{c}_K^T + \mathbf{W}) \right\|$
Maximum Relative Error	$\max \left\| \frac{\mathbf{R}_{K-1} - (\mathbf{c}_K \mathbf{c}_K^T + \mathbf{W})}{\mathbf{R}_{K-1}} \right\|$
Mean Square Error	$\frac{1}{N^2} \left\Vert \mathbf{R}_{K-1} - (\mathbf{c}_K \mathbf{c}_K^T + \mathbf{W}) \right\Vert^2$

6.4 Framework

Figure 2 shows the framework deployed in this work. There are two main components in the system. The first part generates random samples from a given multivariate Gaussian distribution which captures the correlation between the assets of interest. These random numbers are produced by the proposed system where the generator is mapped onto an FPGA using mixed precision in the computational paths. The second part of the framework performs the simulation of the two financial applications under consideration using the randomly generated data.

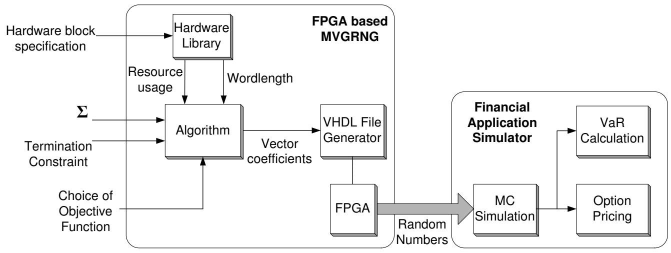


Fig. 2. High Level Overview of Framework.

6.5 Experimental Setup

The purpose of this investigation is to assess the performance of the random number generator core when the already mentioned objective functions are used for its optimization, for a given resource constraint, using the calculation of VaR and option pricing application as testbenches. Since we are interested in the impact of the objective function for the design of the MVGRNG hardware block on the performance of the applications under investigation, it is adequate for this work for the two applications to be implemented on a CPU. The target device for the MVGRNG is a Stratix III EP3SE50F484C2 FPGA from Altera and Quartus II is utilized as a hardware synthesis tool. The precision of the univariate Gaussian random samples z is kept constant throughout the design at 18 bits with 3 bits dedicated for the integer part and 14 bits dedicated for the fractional part. The set of computational blocks (CBs) that is used in the proposed framework is chosen in order to cover the range between 10 to 18 bits precision in an almost uniform way where three CBs with 10, 14 and 18 bits precision have been pre-defined in the hardware library. After the synthesis stage from Quartus II, the resource utilization of 212, 296 and 332 LUTs is reported for CBs using 10, 14 and 18 bits precision respectively. These results are used by the three objective functions under consideration in order to optimize for the desired architecture. In this work two types of input correlation matrices are considered, Type I denotes matrices with high cross correlation between the underlying assets whereas Type II denotes correlation matrices with very low cross correlation.

6.6 Experiment I: Calculation of VaR

In the first experiment, four portfolios each with five correlated assets are considered. The underlying correlations between the five assets for each portfolio are modeled by the four correlation matrices, namely **A**, **B**, **C** and **D**. **A** and **B** are of Type I while **C** and **D** are of Type II. Cross correlation values for Type I matrices are above 0.9 while that of Type II matrices are below 0.01. A fixed resource constraint is used as a terminating condition for the three objective functions of interest where the constraint is set to 1660 LUTs since 1660 is equivalent to five DSP48 blocks which is the amount of hardware resources required by [3], an approach that is based solely on DSP48 blocks for producing multivariate Gaussian random numbers on an FPGA platform, for a 5x5 correlation matrix.

In total, four MVGRNG hardware blocks are investigated and compared. The first three generators are implemented on FPGA using the proposed methodology optimizing for three different error metrics as seen in objective functions 1, 2 and 3 respectively, see Table 1. The fourth MVGRNG is generated using [3]. The generated random numbers are then used for the calculation of VaR of both European-style and Asian-style derivatives and produce the risk assessment for the four portfolios. The results are compared to a reference design implemented on general purpose processor using double precision floating point number representation using the same "seed" for the univariate Gaussian random number generator block.

6.7 Experiment II: Option Pricing

In this experiment two types of Chooser option with 3 assets are considered. The strike price of Chooser option 1 is defined as the sum of all of its assets while that of Chooser option 2 is defined as the maximum of the sum of the two highest priced assets at the end of a specified period of time. Taking into account the two different correlation matrix types I and II we end up with four possible cases of Chooser options, Chooser option 1 and 2 both with high and low cross correlations between their assets. The same procedure as in Experiment I is taken in order to generate the MVGRNG hardware blocks with the exception of the termination condition being set to 996 LUTs for resource constraint which is equivalent to three DSP48 blocks, the amount of hardware required by [3].

6.8 Results: Evaluation of Hardware MVGRNG Blocks

A comparison is made between maturity price of the asset simulated from the random numbers produced from the four different objective functions of interest. We denote the results from the reference design which uses double precision floating point as the actual values. Table 2 summarizes the best objective functions, which produce the closest results for the generation of random numbers from a MVGRNG to the actual values, for each financial application. For both applications where the underlying assets are highly correlated, the best results are obtained by employing the objective function based on the mean square error. On the other hand, with low cross correlation, the objective function which is based on the relative error is the best metric for three out of four cases with an exception of the pricing of Chooser Option 2 where [3] provides the best results.

A selection of plots from the two experiments are shown to illustrate the behaviour of the two financial applications. Figure 3(a) illustrates a plot of the difference in mean return of each portfolio with respect to the mean return of the reference architecture for a European-style VaR. It can be seen that the mean square error metric gives the nearest approximations to the actual values for

Table 2. Summary of Best Performing Objective Function for Financial Applications of Interest.

Experiment	Financial Applications	Best Metric	
		Type I	Type II
I	European-style VaR	MSE	REL
	Asian-style VaR	MSE	REL
II	Pricing of Chooser Option 1	MSE	REL
	Pricing of Chooser Option 2	MSE	[3]

Type I matrices while the relative error is the best metric for Type II matrices. It can be deduced from the plots that an improvement in performance within the range of 5% to 96% is reported for Type I matrices while 24% to 80% improvement is observed for Type II matrices using the four different objective functions for the same hardware resource utilization for the MVGRNG block.

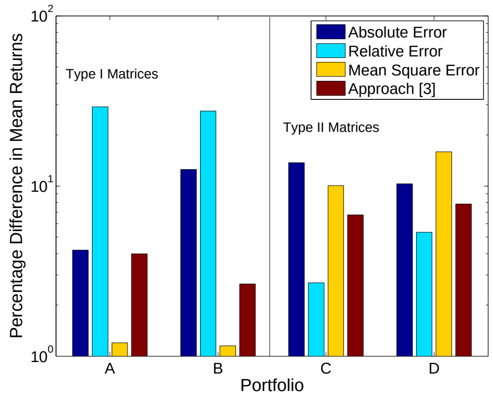
The upper and lower limits of the value of the returns of each portfolio given a 95% level of confidence are investigated. The upper limit is defined as the maximum expected return while the lower limit is the minimum expected return of each portfolio over a specified confidence interval. The difference of the upper and lower limits for Type I and Type II matrices are plotted in Figures 3(b) and 3(c). The plots reinforce the trend previously observed for the best objective functions for Type I and II matrices.

Figures 4 and 5 illustrate plots of the deviation of the strike price from the actual values for Chooser options 1 and 2 over a range of resources. All graphs show that the deviation from the actual strike price decreases as the resource utilization of the MVGRNG hardware block increases. One important point from these plots is the ability for the proposed approach to produce designs across all design space whereas [3] will only offer one design for a given correlation matrix. The plots indicate that for Type I matrix, an improvement in performance within the range of 17% to 81% and 11% to 71% is reported for Chooser options 1 and 2 respectively using the four different objective functions. For Type II matrix, an improvement in performance within the range of 12% to 51% is reported for Chooser option 1 and 8% to 39% for Chooser option 2 using the three four objective functions.

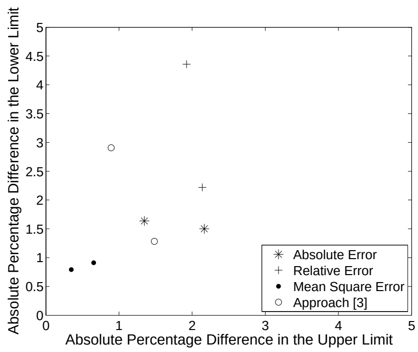
In terms of the hardware performance, the operating frequency of all designs is in the range of 380 to 420 MHz, where the mean is 401.22MHz.

7 Conclusion

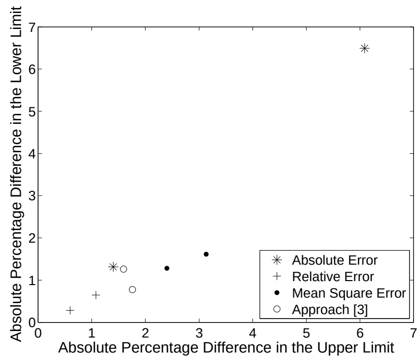
In this paper, a methodology to construct and implement a customized hardware architecture to produce random samples from a multivariate Gaussian distribution tailored made for a specific financial application is described. Three design criteria for the optimization of the MVGRNG are proposed and their impact on the estimation of VaR financial problem and option pricing are investigated. Simulation results have shown that, for an application with highly correlated assets,



(a) Absolute Percentage Difference in the Mean Returns.

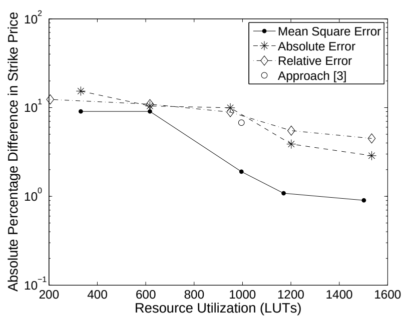


(b) Difference in the Upper and Lower Limits of the 95% Confidence Interval for Type I Matrices **A** and **B**.

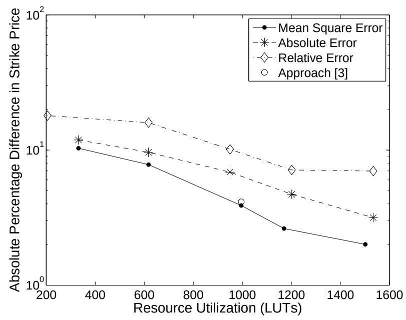


(c) Difference in the Upper and Lower Limits of the 95% Confidence Interval for Type II Matrices **C** and **D**.

Fig. 3. European-style VaR.

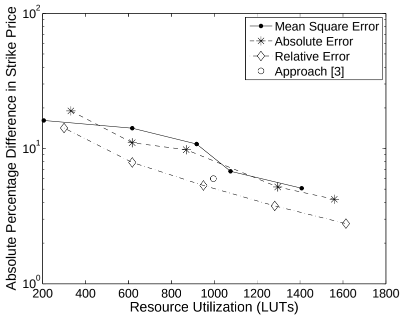


(a) Difference in Strike Price for Chooser Option 1.

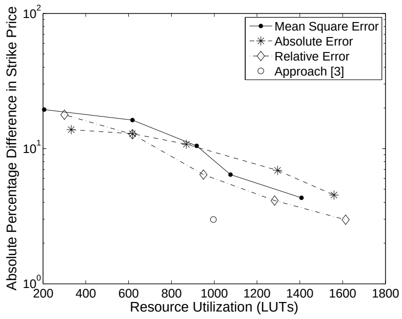


(b) Difference in Strike Price for Chooser Option 2.

Fig. 4. Chooser Options 1 and 2 with Type I Matrix.



(a) Difference in Strike Price for Chooser Option 1.



(b) Difference in Strike Price for Chooser Option 2.

Fig. 5. Chooser Options 1 and 2 with Type II Matrix.

the objective function which optimizes for the mean square error in the approximation of the target correlation matrix of the multivariate Gaussian distribution provides the best performance for the same hardware resource utilization. On the other hand, for an application with low cross correlation between its assets the objective function based on the relative error provides the best results. An improvement in performance of up to 96% is reported for VaR calculation while up to 81% improvement is observed for option pricing using the four different objective functions.

References

1. G. Zhang, P. H. Leong, D.-U. Lee, J. D. Villasenor, R. C. Cheung, and W. Luk, "Ziggurat-based hardware gaussian random number generator," in *Proceedings IEEE International Symposium on Field Programmable Logic and Applications*, 2005, pp. 275–280.
2. N. A. Woods and T. VanCourt, "FPGA acceleration of quasi-monte carlo in finance." in *Proceedings IEEE International Conference on Field Programmable Logic and Applications*, 2008, pp. 335–340.
3. D. B. Thomas and W. Luk, "Multivariate gaussian random number generation targeting reconfigurable hardware," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 1, no. 2, pp. 1–29, 2008.
4. C. Saiprasert, C.-S. Bouganis, and G. A. Constantinides, "Multivariate gaussian random number generator targeting specific resource utilization in an FPGA," in *ARC '08: Proceedings of the 4th international workshop on Reconfigurable Computing*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 233–244.
5. ——, "An optimized hardware architecture of a multivariate gaussian random number generator," *ACM Transactions on Reconfigurable Technology and Systems*, 2009 (to appear).
6. N. H. Chan and H. Y. Wong, *Simulation Techniques in Financial Risk Management.* New Jersey, USA: Wiley, 2006.
7. P. Glasserman, *Monte Carlo Methods in Financial Engineering.* New York, USA: Springer, 2004.