

# Towards Statistical Machine Learning for Edge Analytics in Large Scale Networks: Real-Time Gaussian Function Generation with Generic DSP

Emmanuel Oyekanlu<sup>1</sup>, Samuel Onidare<sup>2</sup>, Paul Oladele<sup>3</sup>

<sup>1</sup> Electrical and Computer Engr. Dept., Drexel University, Philadelphia, Pennsylvania, USA, eao48@drexel.edu

<sup>2</sup> Dept. of Computing & Communication, Lancaster University, Lancaster, United Kingdom

<sup>3</sup> Dept. of Comp Sc. & Space Tech, Luleå University of Technology, Luleå, Sweden

**Abstract**— The smart grid (SG) is a large-scale network and it is an integral part of the Internet of Things (IoT). For a more effective big data analytic in large-scale IoT networks, reliable solutions are being designed such that many real-time decisions will be taken at the edge of the network close to where data is being generated. Gaussian functions are extensively applied in the field of statistical machine learning, pattern recognition, adaptive algorithms for function approximation, etc. It is envisaged that soon, some of these machine learning solutions and other gaussian function based applications that have low computation and low-memory footprint will be deployed for edge analytics in large-scale IoT networks. Hence, it will be of immense benefit if an adaptive, low-cost, method of designing gaussian functions becomes available. In this paper, gaussian distribution functions are designed using C28x real-time digital signal processor (DSP) that is embedded in the TMS320C2000 modem designed for powerline communication (PLC) at the low voltage distribution end of the smart grid, where numerous devices that generate massive amount of data exist. Open-source embedded C programming language is used to program the C28x for real-time gaussian function generation. The designed gaussian waveforms are stored in lookup tables (LUTs) in the C28x embedded DSP, and could be deployed for a variety of applications at the edge of the SG and IoT network. The novelty of the design is that the gaussian functions are designed with a generic, low-cost, fixed-point DSP, different from state of the art in which gaussian functions are designed using expensive arbitrary waveform generators and other specialized circuits. C28x DSP is selected for this design since it is already existing as an embedded DSP in many smart grid applications and in other numerous industrial systems that are part of the large scale IoT network, hence it is envisaged that integration of any gaussian function based solution using this DSP in the smart grid and other IoT systems may not be too challenging.

**Index Terms**— *digital signal processor, powerline communication, smart grid, machine learning, open-source, embedded C, big data*

## I. INTRODUCTION

The SG is an integral part of the IoT [1]. Together with other systems such as industrial, financial, transportation and healthcare system that makes up the entire IoT, it has been projected that devices connected on IoT will be up to 28 billion by 2020 [2], resulting to generation of big data with associated serious shortages in storage memory and computing power. With memory and computing power

shortages, adequate real-time analytics, inference and decisions that will lead to IoT system optimization will be almost impossible [2], [3]. One of the suggested solutions to this problem is that various communications systems, such as wireless, wired (including PLC [4]) and fiber optics should be made to work together (as shown in Fig. 1) to provide adequate communication bandwidth for interacting IoT devices [5], [6], [7]. Edge computing, an IoT computing paradigm by which analytics and decision making will be made at the edge of the SG and IoT systems to ameliorate bandwidth and storage memory shortage and thus, institute real-time inferencing and decision is another solution being suggested. The edge is any part of the network, where data is being generated. For example, as shown in Fig. 1, for the SG, the edge could be the low voltage distribution end where numerous consumer devices are connected.

With the advent of edge analytics, much research is ongoing so that advanced analytics, including machine learning applications could be shifted to the edge of the network using low memory DSPs [8]. The aim of this paper is to contribute to the possibility of machine learning based edge analytics in SG, using low-memory, fixed-point DSP. In this paper, gaussian distributions, because of its wide applications in statistical machine learning, pattern recognition, pattern classification, anomaly detection in SG, machine vision etc., [9], [10], [11], [12], are designed using widely available, industrial, real-time DSP. In literature, lot of research effort have been geared towards the design of gaussian distribution functions for industrial use [11], [12], [13], [14], [15], [16], [17]; however, in most cases, the gaussian function designed are based on expensive and complicated circuits, which may not be readily applicable for edge analytics for SG and IoT. The remaining part of this paper is organized as follows. In Section II, a review of state of the art regarding gaussian distribution function design is briefly explored. Section III is a detailed discussion of the method of gaussian distribution function presented in this paper and main results of the paper. Section IV presents a comparison of the gaussian distribution function designed in this paper with an equivalent Matlab based gaussian function distribution. Section V is conclusion of the paper.

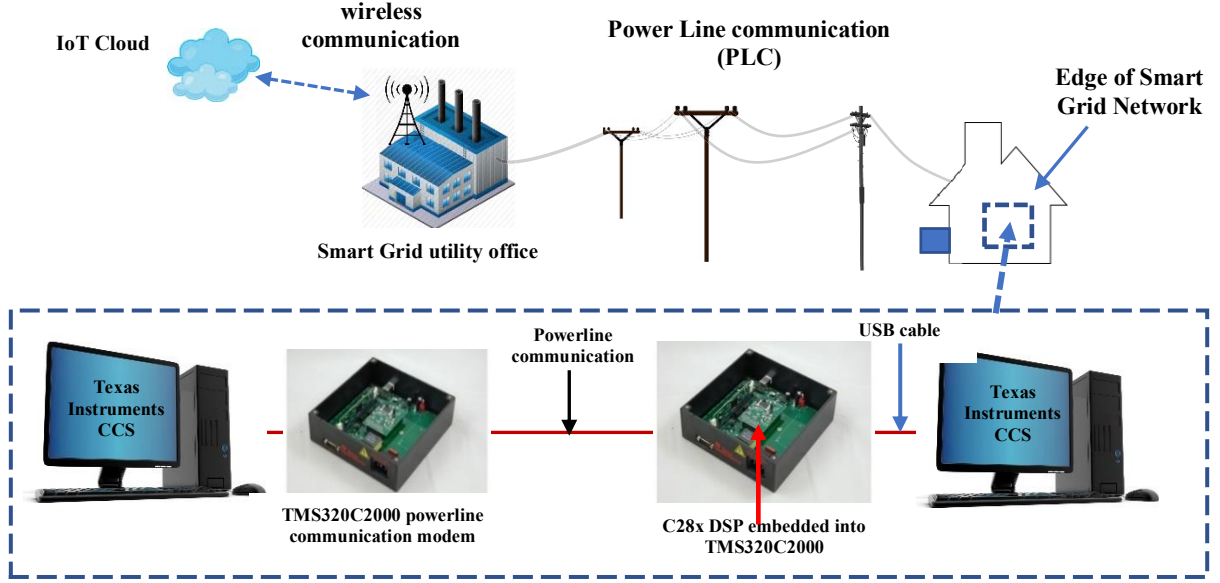


Fig. 1. C28x real-time Embedded DSP could be used as an edge analytic device for the Smart Grid and IoT. Here, C28x is deployed in the TMS320C2000 powerline communication modem. TMS320C2000 is useful for researches on PLC optimization

## II. LITERATURE REVIEW ON THE DESIGN OF GAUSSIAN DISTRIBUTION FUNCTIONS

In [9], a fully programmable very large-scale integration (VLSI) circuit implementing switched current circuit is used to generate gaussian distribution function. The approximate gaussian distribution function that was designed using this circuit is shown in Fig. 2. Also in [13], authors described another method of designing the gaussian distribution function. The method is called  $\text{Cos}^2$  method, and it is fully implementable in a transducer. It also results to an approximate gaussian distribution function shown in Fig. 3. In [18], authors mentioned that an approximate gaussian function is suitable for real time systems with limited memory such as a DSP and intelligent transducers since in most cases, for such systems, memory and computation power is often limited. As such, the design of approximate gaussian function would be helpful in such cases. The case described in [18] regarding an approximate gaussian distribution function would be appropriate for edge analytics in SG and IoT, since memory and computational power is often limited in DSPs at the edges of most large-scale networks such as the SG. It is worthy to note that the approximate gaussian function generated with circuits in [11] and [13] is in most cases, an acceptable approximation of the gaussian function, since as shown in Fig. 4, the 90% to 95% confidence interval, duly covered by these approximations is always used to set data rate limits for gaussian distributions. The gaussian function approximation described in [11] and [13] is also deemed quite useful as discussed in [19]. In [14], a Floating Gate MOS (FGMOS) implementation of the gaussian distribution function is

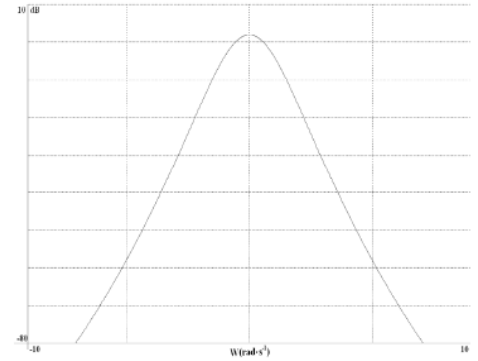


Fig. 2. Approximate gaussian distribution function designed using the large-scale VLSI circuit described in [9].

discussed. In the design, the MOSFET is replaced in the conventional CMOS circuit. In [15], an FGMOS is also used to design the gaussian distribution function, and in the design, the width of the gaussian function could be varied by adjusting the control voltage  $v_c$ . In [16] and [17], the design of a 2-dimensional gaussian distribution function is described using FPGA/ASIC circuits. However, authors pointed out the limitations of the FPGA implementation when compared to a DSP based implementation. It was discussed that FPGA/ASIC implementation may not be as advantageous as a DSP based design of the gaussian distribution function since a DSP is easier to be programmed using open source C program; also, a DSP can easily be used for a floating-point implementation of the gaussian distribution function.

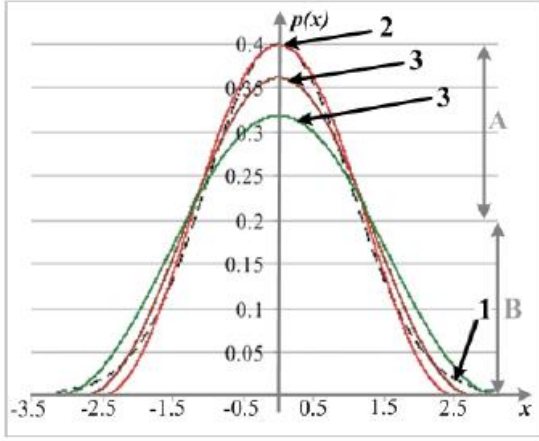


Fig. 3. Several realizations of the approximate gaussian distribution function designed for transducers described in [11].

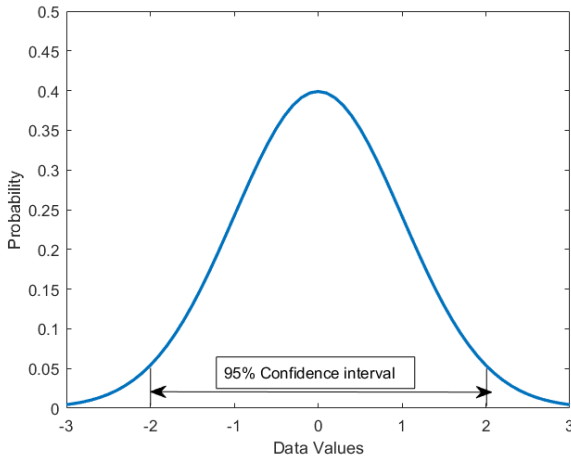


Fig. 4. 90% to 95% confidence interval is always used to set acceptable data rate limits for gaussian distributions

In this paper, a DSP based implementation of the approximate gaussian distribution function using a generic, easily available, widely distributed DSP is achieved. Open source C program is used to implement the design for the C28x, Texas Instruments DSP. The approximate gaussian function designed in this paper is adaptive since the width and the height of the gaussian function could be easily varied to achieve an approximate gaussian distribution of any width and height by varying the parameters of the approximate gaussian function stored in the LUT. The design is simple to implement since it is based on sine wave clipping method. The design is also different from the state of art since it is not based on complicated circuits as reported by other authors. It is believed that with this design, many complex applications, include machine

learning applications could be achieved at the edges of large scale networks such as the SG since the C28x is widely used in many IoT industries [24].

### III. DESIGN OF APPROXIMATE GAUSSIAN FUNCTION DISTRIBUTION USING C28X REAL TIME DSP

To design the gaussian distribution function, a TMS320C2000 modem is connected to a PC having Code Composer Studio (CCS). CCS is a software interface produced by Texas Instruments, and it is primarily for real-time programming of DSPs using C programming language, embedded C or the assembly language code of the DSP of interest. The connection schematic is shown in the dashed block part of Fig. 1. TMS320C2000 is designed for PLC at the low voltage end of the SG. The signal processing method used for gaussian distribution function design for the C28x DSP involves using C language program in the CCS for signal clipping and storage in C28x LUT [20]. A sine wave that has a frequency corresponding to the width of the gaussian waveform is first designed. The sine wave part corresponding to the height and width of the gaussian distribution is clipped [21] and stored in C28x LUT. The stored waveform can be recalled and modified to generate gaussian waveforms of different amplitudes and widths. Using C language affords the real-time programmer the benefit of designing waveforms without the rigors of understanding the underlying assembly code of the particular processor [20]. Schematics of the process sine wave design and clipping for gaussian waveform generation at the edge of the SG is shown in Fig. 5. It is worthy to note that the amplitude of the approximate gaussian distribution function generated is conditioned on the maximum voltage level that is supported by the C28x digital to analog converter (DAC). The amplitude of the C28x DAC is given in (1) as

$$\text{Waveform}_{\text{voltage}} = V_{\text{ref}} \left( \frac{\text{Waveform Amplitude}}{\text{Range of 16 bit DAC}} \right) \quad (1)$$

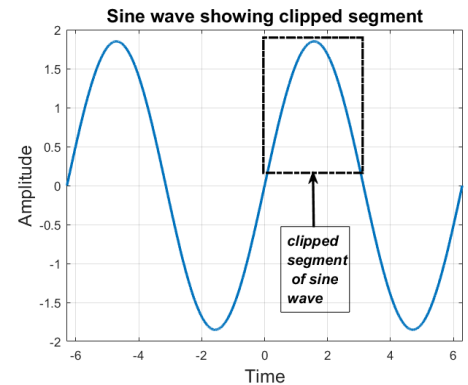


Fig. 5. Method of sinewave clipping for gaussian waveform generation useful for outlier detection at the edge of the Smart Grid

In (1), the range of the 16-bit DAC is  $(\frac{2^{16}}{2})/2 = 32,768$ .

According to [22], the analog output voltage of the stored waveform can range from voltage reference (Vref) value of 0 V to 3.3 V. A 1 V reference DAC voltage is used as the maximum DAC waveform output voltage in this paper. Thus, the maximum amplitude that any generated [22], [23] gaussian waveform could have in this implementation is

$$V_{\text{Analog}} = 1.0 \left( \frac{30,000}{32,768} \right) \approx 1.0 \text{ V} \quad (2)$$

The reason for fixing this amplitude is to be able to adjust the amplitude and width of the stored gaussian waveform in the LUT with respect to a fixed threshold. With this method, any approximate gaussian waveform with varying amplitude and width could be generated (with respect to this threshold) using the stored waveform in the LUT. Using the method of sinewave clipping for gaussian waveform described, a random gaussian distribution function with parameter  $\mu = 95$  and  $\sigma = 30$  is designed. The Matlab design is first shown in Fig. 6 and an approximate gaussian function reflecting this parameter is designed for C28x DSP using C program in CCS. It is shown in Fig. 7. To accomplish the design, the gaussian distribution function equation

$$f_x(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3)$$

is considered. In (3), the parameter x is the data value for

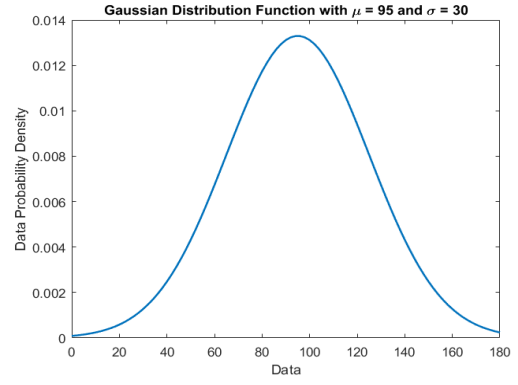


Fig. 6. Gaussian Distribution Function with  $\mu = 95$  and  $\sigma = 30$  implemented in Matlab.

which the corresponding gaussian probability distribution  $f(x)$  is evaluated. For a gaussian waveform  $\mu = 95$  and  $\sigma = 30$ ,  $f(x)$  is  $\approx 0.013$  for 180 data points as shown in Fig. 6. To reflect the full 1 V DAC voltage of the C28x in CCS, the 0.013 amplitude is scaled up by a factor 30000 which will represent the corresponding C28x voltage realization. It is further scale up by a 2-order magnitude as discussed in [22], [23] to compensate for DAC thermal and noise effect. The resulting approximate gaussian function waveform is shown in Fig. 7. To evaluate the adaptivity of the approximate gaussian waveform stored in the LUT, several other approximate gaussian waveforms were produced using the initial waveform of Fig. 7. For example, an approximate Gaussian waveform having only one-third of the amplitude

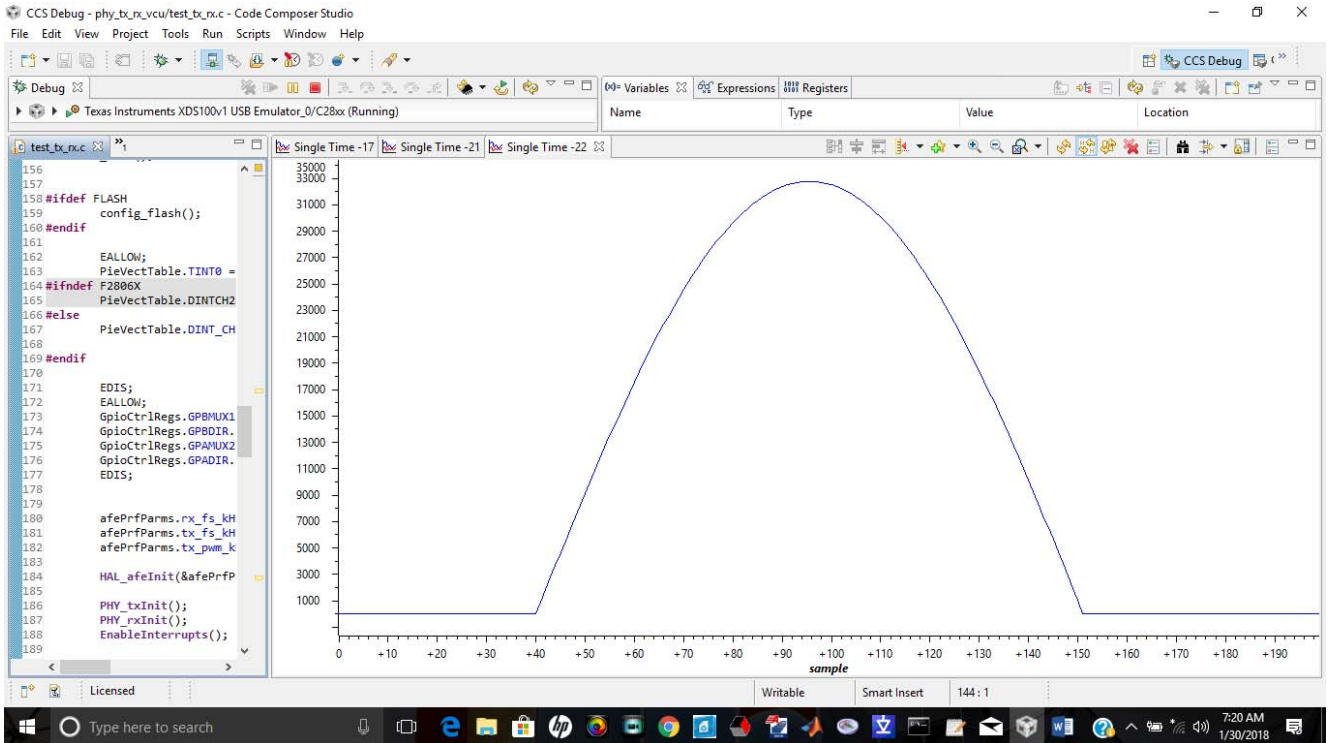


Fig. 7. Approximate gaussian distribution function designed using the C28x DSP. Here, scaling for the parameters  $\mu = 95$  and  $\sigma = 30$  to reflect DSP DAC parameter is could be observed on the y-axis.

of the stored waveform was produced mainly by scaling down the amplitude of the clipped sine wave. It is also shifted on the data axis (x-axis) to reflect another waveform having different gaussian distribution function parameters. The result is displayed in Fig. 8. Also, an approximate discrete gaussian waveform was produced using the C28x DSP and shown in Fig. 9, while yet another type of possible discretization, amplitude and width for approximate gaussian waveform is shown in Fig. 10.

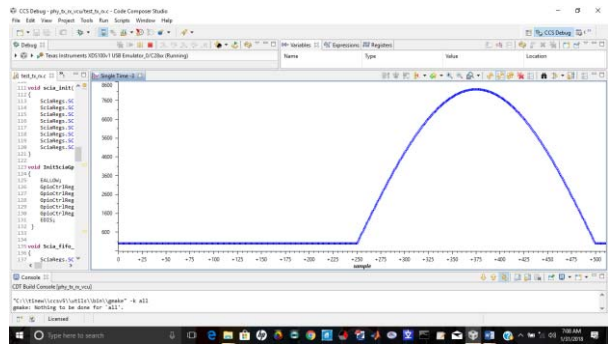


Fig. 8. An approximate gaussian distribution waveform produced by adaptive variation of the gaussian waveform stored in the LUT

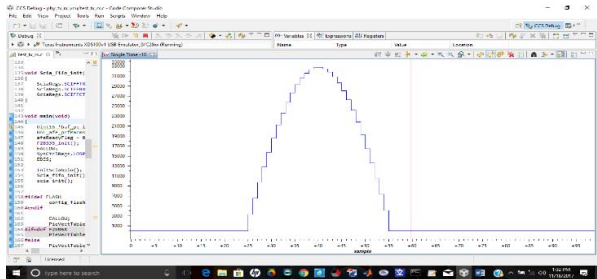


Fig. 9. Approximate discrete gaussian waveform designed using C28x DSP that is part of a PLC modem at the edge of the smart grid.

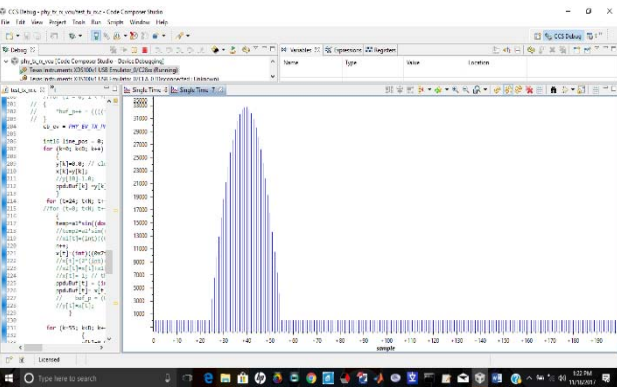


Fig. 10. Approximate discrete gaussian waveform of with different gaussian distribution function parameters produced using the C28x DSP

#### IV. COMPARISON OF THE APPROXIMATE GAUSSIAN FUNCTION DISTRIBUTION DESIGNED USING C28x REAL TIME DSP WITH EQUIVALENT MATLAB BASED GAUSSIAN FUNCTION

To evaluate the open source (C program) based sine wave clipping method of designing approximate gaussian distribution function for low memory DSP devices at the edge of large scale networks discussed in this paper, data of the approximate gaussian distribution function designed with the C28x DSP in Fig. 7 is exported into an Excel file, scaled down by 3E4 and another 2-order magnitude factor, and compared with the equivalent Matlab based gaussian distribution function shown in Fig. 6. The comparison is shown in Fig. 11. It is to be noted that the comparison is only up to the 95% confidence interval, which is approximately the data values between 40 and 150 in Fig. 11. A mean square error (MSE) comparison between the approximated gaussian distribution function of C28x DSP and the equivalent Matlab based gaussian function shows only a difference of 2.5454e-05, which is a quite minimal difference between the two gaussian shapes.

#### V. CONCLUSION

In this paper, approximate gaussian distribution functions that could be used for a lot of applications, including statistical machine learning applications at the edges of large scale networks, such as the SG is designed using Texas Instruments C28x DSP and open-source C programming language. The approximate gaussian function is useful for DSPs that have low memory and low computing power, synonymous with DSPs that are always available at the edges of large scale network. Statistical comparison, using the MSE between the approximate gaussian distribution function designed using sine wave clipping method for C28x

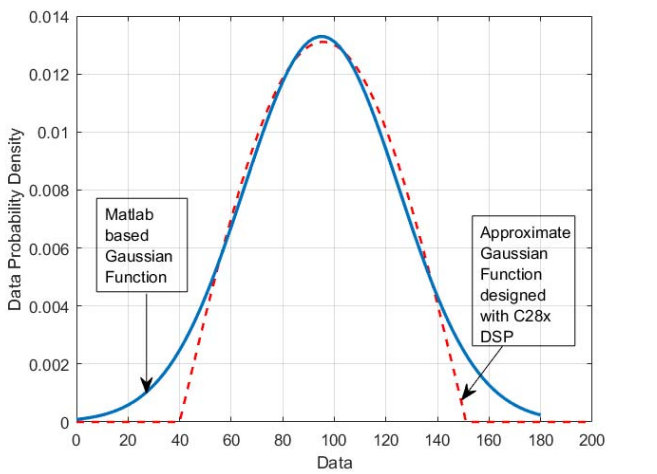


Fig. 11. Comparison of the approximate gaussian distribution function designed using real-time C28x DSP and an equivalent Matlab based design.

real-time DSP and an equivalent Matlab based gaussian distribution function design show a low difference of only  $2.5454e-05$  up to the 95% confidence interval. This indicates that the design of gaussian distribution functions for low-cost, low-computing power and low memory DSPs discussed in this paper could be applied for a lot of edge analytic applications in the SG and other large-scale networks. Future consideration for this method of gaussian distribution function design for real-time DSPs may include devising a method by which this method can be improved to yield an improved gaussian distribution function bell shape with an even lower statistical MSE value.

#### ACKNOWLEDGMENT

Authors would like to thank Schweitzer Engineering Laboratory (SEL), Pullman, Washington, USA, for the donation of equipment that, in part aided this research work.

#### REFERENCES

- [1] J. Jadhav, S. S. Lokhande, V.N. Gohokar, "Energy Management System in Smart Grid Using Internet of Things", ICPEICES, IEEE, 2016
- [2] D. Reinsel, J. Gantz, J. Rydning, "Data Age 2025: The Evolution of Data to Life-Critical Don't Focus on Big Data; Focus on Data That's Big", IDC, Seagate, April, 2017.
- [3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, *Edge Computing: Vision and Challenges*, IEEE Journal of Internet of Things, Vol.3, No. 5, Oct. 2016.
- [4] A. A. Zhilenkov, D. D. Gilyazov, I. I. Matveev and Y. V. Krishtal, "Power line communication in IoT-systems," *2017 IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus)*, St. Petersburg, 2017, pp. 242-245.
- [5] E. Oyeekanlu, "Predictive Edge Computing for Time-Series of Industrial IoT and Large Scale Critical Infrastructure based on Open-Source Software Analytic of Big Data", IEEE Conference on Big Data, Boston, Massachusetts, USA, December 2017.
- [6] USA Dept. of Energy, Communication Requirements of Smart Grid Technologies, DOE White Paper, Oct. 2010.
- [7] E. Oyeekanlu, C. Nelatury, A. Fatade, O. Alaba, O. Abass, "Edge Computing for Industrial IoT and the Smart Grid: Channel Capacity for M2M Communication over the Power Line", IEEE 3<sup>rd</sup> Int'l Conference on Electro-Technology for National Development, November 2017.
- [8] K. S. Bharati, A. Jhunjhunwala, "Implementation of Machine Learning Applications on a Fixed-Point DSP", In Proc., IEEE 28<sup>th</sup> Conf. on Electr. & Computer Engrn, Halifax, Canada, 2015
- [9] C. E. Rasmussen, C. K. I. Williams, "Gaussian Processes for Machine Learning", the MIT Press, 2006, ISBN 026218253X.
- [10] C. Williams, "Gaussian Processes for Machine Learning", Institute of Adaptive and Neural Computation, Univ. of Edinburgh, UK, 2007.
- [11] W. S. Zhao, Y. G. He, J. Y. Huang, S. Z. Qi, G. F. Fang, "A Fully Programmable Analog VLSI For Gaussian Function Generator Using Switched-Current Circuit", In Proc., Int'l Conf. on Wavelet Analysis and Pattern Recognition, Beijing, China, 2007
- [12] N. Okada, M. Takasaki, and K. Tanaka, "A Case of Bad Data Detection for Distribution Feeder Voltage Measurement", IEEE PES Innovative Smart Grid Technologies Europe, Berlin, 2012
- [13] Z. L. Warza, M. J. Korczynski, M. Galovska, "Shifted Up Cosine Function as Model of Probability Distribution", IMEKO World Congress of Fundamental and Applied Metrology, Portugal, 2009.
- [14] R. Srivastava, U. Singh, M. Gupta, "Analog Circuit for Gaussian Function with Improved Performance", IEEE, 2011
- [15] R. Srivastava, M. Gupta, and U. Singh, "Fully Programmable Gaussian Function Generator Using Floating Gate MOS Transistor", Int'l Scholarly Res. Network, Vol. 2012, Article ID 148492.
- [16] M. C. Hanumantharaju, and M. T. Gopalakrishna, "Design of Novel Architectures and FPGA Implementation of 2D Gaussian Surround Function", Int'l Journal of Information Processing, 7(4), 66-75, 2013.
- [17] M. C. Hanumantharaju, M. Ravishankar, D. R. Rameshbabu, "Design and FPGA Implementation of an 2D Gaussian Surround Function with Reduced On-Chip Memory Utilization", IEEE, 2013.
- [18] Z. L. Warza, M. J. Korczynski, "On Line Cleaning of the Raw Data and Uncertainty Type A Evaluation – Development Aspect", Proceedings of 18<sup>th</sup> Symposium on Metrology and Metrology Assurance, Bulgaria, 2008.
- [19] J. D. Cook, "Simple Approximations to Normal Distributions", online, available, [www.johndcook.com](http://www.johndcook.com) April, 2010.
- [20] S. S. Prasad, and S. K. Sanyal, "Design of Arbitrary Waveform Generator based on Direct Digital Synthesis Technique using Code Composer Studio Platform", IEEE, 2007
- [21] M. Pukette, "The Theory and Technique of Electronic Music", World Scientific Publishing, pp. 128, 2007
- [22] A. Singh, D. Hemann, "DC Power-Line Communication Reference Guide", Texas Instruments, Dallas, TX, Rep. no. TIDU160, 2013, pg. 2.
- [23] "FFT Plot in CCS v3.3 – Magnitude Unit", Code Composer Forum, Texas Instruments, 2010, [Online]. Available: [https://e2e.ti.com/support/development\\_tools/code\\_composer\\_studio/f/81/t/55368](https://e2e.ti.com/support/development_tools/code_composer_studio/f/81/t/55368).
- [24] Texas Instruments, "C2000 Real-Time Microcontrollers", Texas Instruments, 2016