Frequency-domain intrachip communication schemes for CNN

Citation for published version (APA):

Mondragon-Torres, A., Gonzalez-Carvajal, R., Pineda de Gyvez, J., & Sanchez-Sinencio, E. (1998). Frequency-domain intrachip communication schemes for CNN. In *Proceedings of the Fifth IEEE International Workshop on Cellular Neural Networks and Their Applications, 14-17 April 1998, London, United Kingdom* (pp. 398-403). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/CNNA.1998.685411>

DOI:

[10.1109/CNNA.1998.685411](https://doi.org/10.1109/CNNA.1998.685411)

Document status and date:

Published: 01/01/1998

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Frequency-Domain Intrachip Communication Schemes for CNN

Antonio Mondragón-Torres, Ramón González-Carvajal¹,

José Pineda de Gyvez, Edgar Sánchez-Sinencio

Department of Electrical Engineering,

Texas A&M University

College Station, Texas, 77843, USA

e-mail: antonio@ee.tamu.edu Phone: (409) 845-8799 Fax: (409) 845-7161

ABSTRACT: A frequency-domain scheme to share a communications channel among the cells of a CNN is proposed. The scheme is based on a modification of the Wave-Parallel-Computing technique and addresses the problem of reducing the number of communication links. Reduction in the communication paths is achieved by frequency multiplexing. This makes it possible to have simultaneous full-parallel access to all the cells of the array. The approach also takes advantage of the parallelism inherent in Wave-Parallel-Computing to solve part of the state equation within the same channel during a transmission operation. Moreover, with this architecture, the CNN array is not required to have a physical matrix array of cells, providing in this form even more flexibility for the hardware implementation. A system level simulation was done and operating ranges were found as an aid to propose a final system architecture of a tentative VLSI IC.

1. Introduction

Two of the main problems in the hardware implementation of Cellular Neural Networks (CNN's) are the required number of communication paths, and the size of the cell. However, the main bottleneck to speed up the computation time is the input/output communication delay with each cell. This can be even several times higher than the actual computation time of the CNN array. Several attempts to carry out the I/O communications in the frequency domain have already been pursued [2-4]. The technique that seems more viable of being implement in a CMOS IC with a relatively large number of CNN cells is the Wave-Parallel-Computing approach proposed by Yuminaka et al [4]. This technique is based on Frequency Division Multiple Access Amplitude Modulation (FDMA-AM). In this scheme, all the information in a communications channel is modulated by a finite set of different carriers. Additionally, all arithmetic operations can be performed on the modulated waveforms, leading in this form to an inner product computation in real time and without the need to demodulate. This clearly permits a highly parallelism within the system by transmitting, multiplying and accumulating at the same time. This conveys us to the concept of Transmit Multiply and Accumulate (TMAC) similar to the concept used in the DSP terminology and this approach is not only simple, but also it leads to a much simpler hardware implementation that the one required with traditional techniques.

For near real-time video applications, a continuous time CNN usually requires that its inputs be present at all times. However, the communication links can take a large proportion of the integrated circuit area, that is why current implementations do not have a very large number of CNN cells in spite of the fact that the CNN cells can be efficiently implemented. A time-multiplexing architecture has been proposed to alleviate the problem of communicating the external input data with the array [5]. Nevertheless, under these conventional implementations full parallel access to the entire CNN array is not possible. It will be shown that with the scheme hereby proposed, full parallel access is possible using a reduced number of wires by assigning multiple frequency bands to each connection wire. The main advantages to the system hereby proposed are the following:

- Reconnection programmability. CNN cells are not required to occupy a physical matrix array. Each cell has its own characteristic frequency of operation. This allows interconnecting cells with any other cell in the array. By assigning contiguous frequency channels to both the templates and the cells a simple implementation can be obtain to mimic the conventional CNN array.
- Easy expandability. The CNN array can be easily expanded by adding another CNN chip and programming its frequencies. This is an advantage compared with conventional CNN implementations since the implementation of large arrays in a single chip do not have direct external pin connections with the neighboring cells that makes not possible to add an additional array interacting in real-time with the first array.
- Dynamic programming of the template with different sizes. This will allow to expand and to reprogram the neighborhood dynamically. Neighborhoods with radius greater than 1 can be easily programmed This is a major advantage compared with any of the conventional arrays since their connections can not be programmed because they are limited to the physical hardware connections.

¹ On leave from University of Seville, Spain

2. Intrachip Communications Schemes

A conventional CNN array [1] consists of a matrix array of cells; each of them connected only to the surrounding neighbors. The neighborhood of a cell C_{ij} is denoted as $N(i,j)$. The principle of operation of each cell consists of computing the weighted sum of the outputs of the neighboring cells and the input image including a bias current. This sum is passed through a lossy integrator and a limiting function (1).

$$C \frac{dx_{ij}(t)}{dt} = -\frac{x_{ij}(t)}{R} + \left[\sum_{C(k,l) \in N(i,j)} A(i,j;k,l)y(t) + \sum_{C(k,l) \in N(i,j)} B(i,j;k,l)u_{kl} + I \right] \quad (1)$$

$$= -\frac{x_{ij}(t)}{R} + \langle A, Y \rangle + \langle B, U \rangle + I$$

In this section, an Intrachip Communication Scheme is presented based on a modification of the Wave-Parallel-Computing technique[4] and of the narrow-band system of [6]. One can see from (1) that the equation has two inner product computations $\langle A, Y \rangle$ and $\langle B, U \rangle$, which can be realized as part of the Wave Parallel Multiplication henceforth referred to as TMAC. The input signals are carrier modulated by an α -spaced set of frequency of carriers as well as the template values which are modulated to the respective set depending on the neighborhood configuration. Let $[Y]_{m_a \times n_a}$ be a two-dimensional array corresponding to the output of the cells in a neighborhood $N(i,j)$ surrounding some cell C_{ij} . The array Y is a submatrix of $[\tilde{Y}]_{m_y \times n_y}$ that represents the entire output array. Let $[A]_{m_a \times n_a}$ be a two-dimensional array corresponding to template A . Typically $m_y \gg m_a$, and $n_y \gg n_a$. Each of these arrays is a two dimensional array with elements

$$[A]_{ij} = a_{ij} \cos(\omega^{(a)}_{ij} t) \quad (2a)$$

$$[\tilde{Y}]_{ij} = \tilde{y}_{ij} \cos(\omega^{(y)}_{ij} t), \quad (2b)$$

Let us now represent the 2-D spatial frequency array as a set of contiguous frequency bands indexed by columns j and then by rows i as follows

$$\omega^{(y)}_{ij} = \alpha_{ij} \omega_0 \quad i=1 \dots m_y, j=1 \dots n_y \quad (3a)$$

$$\omega^{(a)}_{ij} = \alpha_{ij} \omega_0; \quad i=1 \dots m_a, j=1 \dots n_a \quad (3b)$$

where ω_0 is the initial frequency and α_{ij} is a frequency scaling factor defined as

$$\alpha_{ij} = ((i-1)m_y + j) \quad i=1 \dots m_y, j=1 \dots n_y \quad (3c)$$

Notice that the row frequency spacing for $\omega^{(a)}$ is given by m_y and not by m_a . This will allow a cell C_{ij} to have A and Y at the same frequencies by just shifting A to the position of the cell C_{ij} , i.e. $\omega^{(a)}_{ij} = \omega^{(y)}_{ij}$. Next, we will discuss the conditions for the implementation of the inner products of eq. (1). Without loss of generality consider a two-dimensional inner product $\langle A, \tilde{Y} \rangle$

$$\langle A, \tilde{Y} \rangle = \sum_{i=1}^{m_a} \sum_{j=1}^{n_a} \sum_{h=1}^{m_y} \sum_{k=1}^{n_y} a_{ij} \tilde{y}_{hk} [\cos(\alpha_{ij} \omega_0 t) \cdot \cos(\alpha_{hk} \omega_0 t)] \quad (4)$$

Expanding the product using the identity $\cos(a)\cos(b) = 1/2 [\cos(a+b) + \cos(a-b)]$ we obtain

$$\langle A, \tilde{Y} \rangle = \sum_{i=1}^{m_a} \sum_{j=1}^{n_a} \sum_{h=1}^{m_y} \sum_{k=1}^{n_y} a_{ij} \tilde{y}_{hk} \left[\cos[(\alpha_{ij} - \alpha_{hk}) \omega_0 t] + \cos[(\alpha_{ij} + \alpha_{hk}) \omega_0 t] \right] \quad (5)$$

Notice that the second term contains frequencies above $2\omega_0$. When $i = h$ and $j = k$ the summation indexes go from 1 to the template size (m_a, n_a) and therefore $\tilde{Y}$ becomes Y . Equation (5) becomes:

$$\langle A, Y \rangle = \frac{1}{2} \sum_{i=1}^{m_a} \sum_{j=1}^{n_a} a_{ij} y_{ij} [\cos 0 + \cos(2\alpha_{ij} \omega_0 t)] \quad (6)$$

It can be seen that the first term has a base band component ($\cos 0$). Notice that the coefficients of the base band component are $a_{ij} y_{ij}$, which are the terms corresponding to the inner product $\langle A, Y \rangle$. By using a lowpass filter with a cutoff frequency such that $\omega_{pass} < \omega_0/2$ then we obtain only the base band elements.

$$\langle A, Y \rangle = \frac{1}{2} \sum_{i=1}^{m_a} \sum_{j=1}^{n_a} a_{ij} y_{ij} \quad (7)$$

which is what we want.

A second inner product can be computed simultaneously if the required output is $\langle A, Y \rangle + \langle B, U \rangle$. This can be done as long as the frequency bands assigned to A and $\tilde{Y}$ do not overlap those assigned to B and $\tilde{U}$. As a special issue consider the case when a coefficient is negative. Then we have that $-k \cos(\omega t) = k \cos(\omega t + \pi)$. Notice that the sign information is contained in the phase. This is how the proposed modulation scheme differs from the conventional amplitude modulation in which a DC offset higher than the maximum possible value of the constant is added.

Figure 1. System Block diagram representation of one CNN processor

The system block of the modified proposed is shown in Figure 1. In this diagram Templates A and B and the Input U and Output Y are modulated by a set of different frequencies. Due to the fact that the neighborhood was chosen to be one, the number of different frequencies needed per cell is nine. Special attention must be paid to the fact that all the addition and subtraction of frequencies can lead to terms that can be near the low pass frequency bandwidth. Therefore, a practical frequency assignment is done by separating the carrier frequencies by a proportional factor α , e.g. $f_1 = \omega_0$, $f_2 = 2\omega_0$, $f_3 = 3\omega_0, \dots$. This factor is chosen as a low pass filter design constraint which leads to the specification of the filter order. The number of frequencies required for each individual TMAC cycle depend on the size of the neighborhood, e.g. for radius 1, nine frequencies are required. In Figures 2(a) and 2(b), template B and input U are modulated by a set of carriers leading to an FDMA-AM-DSB system, this is shown as a set of weighted impulses. These waveforms are then multiplied to realize the inner product computation. A similar procedure is done for Template A and the output values Y. From Figure 2(c), it is shown that the cross product has a large DC component (desired result) and a series of cross terms.

Some of the advantages of using this technique are: the oscillators can be shared, the required filters are very simple, the technique is all analog and the building blocks are simple. Some of the disadvantages though are as follows: the required number of frequencies is high, the frequency assignment is dependent on the effect of the harmonics in the system (quality of oscillators), the time response of the system is a function of the frequency assignment (DC filtering), it is also sensible to offsets, and the oscillator reference must be shared among cells.

Figure 2. (a) Carrier modulated Template B signal. (b) Carrier modulated input U. (c) Template B and input U both carrier modulated after multiplication in the frequency domain.

3. Modeling System Non-idealities

All derivations use normalized units to determine the best architecture for the system and to not assume that the processing is in voltage or current mode. Time constants are treated in a similar way. For the purpose of evaluating the approach, we considered the following figures of merit: *i) Algorithm convergence* which is a qualitative solution when the CNN algorithm is met, *ii) State convergence* which indicates when the system arrives at steady state, and *iii) Filter convergence* which corresponds to the steady state convergence of the low pass filter. For simulation purposes, the integration step was set to 0.1 time units and $R=C=1$. As way of example, for an edge detection algorithm applied to the image of Fig. 3a, and without any deviation from the

nominal parameters, the *algorithm convergence* is met in 9 cycles, the *state convergence* is met in 57 cycles using a filter with *filter convergence* to a unit step of 2048 units.

Figure 3. (a) Unprocessed image, (b) after edge detection, (c) after processing with worst case parameters: $\tilde{A}_{offs}=0.1$, $\tilde{M}_{offs}=0.5$ and $THD=20\%$

Figure 3a shows our 10x10 pixel benchmark image, Figure 3b displays the image once the edge detection algorithm was applied, and Figure 3c shows the incorrectly processed image under perturbations in the system. $\tilde{A}_{offs}$ is the offset added at the output of the modulators, $\tilde{M}_{offs}$ is the offset added at the output of the multipliers and THD represent the total harmonic distortion added at the third harmonic.

We have considered a 6th and a 2nd order low pass Butterworth filter with cutoff frequencies at $\omega_c/4$ and $\omega_c/16$, respectively. Observe that the constant frequency scaling of the carriers allows us to relax the filter specifications. The only difference between the two filter specs is that the algorithm convergence for the second order filter increases to 12 cycles, but the complexity of the filter is greatly reduced. AM is a coherent demodulation, and thus there is a very small range of frequency deviation that is allowed. Our simulations indicate maximum deviations in the range of $\pm 0.1\%$ for correct algorithm convergence. Similarly, the phase difference must be within a small range. This implies that only one set of carriers must be used. It follows then that this reference frequencies must be supplied by one common module to other, or they must be supplied externally to both (this option is preferred to avoid design complexity on either module). From a mathematical standpoint, each carrier can be represented as $\omega_{carrier} = \cos((\alpha\omega_o + \omega_{dev})t)$ where $\omega_{dev} \ll \omega_o$.

Typically, actual hardware implementations of (de)modulators have a small offset. Therefore, an equal magnitude and sign offset is added at the output of each modulator. Basically, this adds a DC level error that affects the Wave-Parallel-Computing technique. Recall that this technique relies on the DC component to estimate the inner product. Fortunately, this offset can be compensated without incurring into complex implementations of the modulator. The compensation technique consists of low pass filtering the modulated signals once they are added together, and of extracting afterwards the DC component and subtracting it from the added signals. The effect of this offset is given in (8) for template A and outputs Y; a similar set of equations applies for template B and input U.

$$\tilde{a}_{ij} = a_{ij} \cos(\omega_{ij}^{(a)} t) + \tilde{A}_{ij} \quad (8a)$$

$$\tilde{y}_{ij} = y_{ij} \cos(\omega_{ij}^{(y)} t) + \tilde{A}_{ij} \quad (8b)$$

where $\tilde{A}_{ij}$ is a uniformly distributed random variable between 1 and -1. When the carrier frequencies are equal, then by multiplication of individual terms and after applying trigonometric identities the corresponding inner product results in

$$\langle \tilde{A}, \tilde{Y} \rangle = \sum_{i=1}^{m_a} \sum_{j=1}^{n_a} \left(\frac{1}{2} a_{ij} y_{ij} + \frac{1}{2} a_{ij} y_{ij} \cos(2\alpha_{ij} \omega_o t) + (a_{ij} + y_{ij}) \cos(\alpha_{ij} \omega_o t) \tilde{A}_{ij} + \tilde{A}_{ij}^2 \right) \quad (9)$$

where $\omega_{ij}^{(a)} = \omega_{ij}^{(y)} = \alpha_{ij} \omega_o$. By filtering out the high frequency terms, the output from the modulator is

$$\langle \tilde{A}, \tilde{Y} \rangle = \frac{1}{2} a_{ij} y_{ij} + ij \tilde{A}_{offs}^2 \quad (10)$$

where for simplicity $\tilde{A}_{offs} = \tilde{A}_{ij}$. Notice that the inner product needs to be scaled by 2 to normalize values. We have assumed that the nominal amplitude of the modulating waveforms is 1. From simulations we found that higher values help to arrive at the *algorithm convergence* faster and lower values tend not to converge to the correct solution. The modulating signal can have a range of $\pm 10\%$ amplitude variations without an appreciable change in the algorithm convergence (for edge detection). These values determine the amplitude quality of the modulator. These variations can be modeled as

$$\tilde{a}_{ij} = \tilde{a} a_{ij} \cos(\alpha_{ij} \omega_o t) \quad -1 \leq \tilde{a} \leq 1 \quad (11a)$$

$$\hat{y}_{ij} = \tilde{y}_{ij} \cos(\alpha_{ij} \omega_o t) \quad -1 \leq \tilde{y} \leq 1 \quad (11b)$$

where $\tilde{a}$ and $\tilde{y}$ are uniformly distributed random variables. There exists also a possibility of having offset at the output of the activation function. However, our simulations indicate that this perturbation does not have a large impact on algorithm convergence, but instead it does on the convergence of the ± 1 saturation levels. The perturbation can be modeled by adding a uniformly distributed random variable, $\tilde{y}_{offs}$, between 1 and -1, to y_{ij} in (8b). To process the inner product, say $\langle \mathbf{A}, \mathbf{Y} \rangle$, the modulated waveforms of $\mathbf{A}$ and $\mathbf{Y}$ need to be multiplied together. A DC offset at the output of the multiplier has a direct impact on the correct convergence of the algorithm. Also it is important to note that this perturbation has a different impact depending on which template is used. This new offset effect can be modeled as

$$2\langle \tilde{\mathbf{A}}, \tilde{\mathbf{Y}} \rangle + \tilde{M}_{offs} = 2\left(\frac{1}{2} a_{ij} y_{ij} + ij \tilde{A}_{offs}^2\right) + \tilde{M}_{offs} \quad (12)$$

where M_{offs} is the multiplier's offset. The effect of the offset at the output of the modulators plus the offset at the output of the multiplier after low-pass filtering and summing both inner products is

$$x_{ij}(t) = 2\tilde{M}_{offs} + \langle \tilde{\mathbf{A}}, \tilde{\mathbf{Y}} \rangle + \langle \tilde{\mathbf{B}}, \tilde{\mathbf{U}} \rangle + I = 2\tilde{M}_{offs} + 4ij\tilde{A}_{offs}^2 + \langle \mathbf{A}, \mathbf{Y} \rangle + \langle \mathbf{B}, \mathbf{U} \rangle + I \quad (13)$$

This equation represents those parameters that show an overall impact in the correct solution of the algorithm. It is shown that they are additive terms that can be comparable in magnitude to the maximum values or dynamic range that the system can sustain under limited power supply and noise floor. To meet the dynamic range imposed from both the power supply and noise floor, a method of scaling the signals is needed to let the system work within linear ranges to minimize the distortion. The scaling is done by multiplying the templates by the square root of a scaling factor (β), and by limiting the values of the activation function and the bias term I to the same scale factor. The output $\mathbf{Y}$ is then multiplied by the inverse of the scale factor to obtain the ± 1 range expected for the solution. Equation (1) is modified to give the desired scaled version as shown next

$$C \frac{dx_{ij}(t)}{dt} = -\frac{x_{ij}(t)}{R} + \langle \sqrt{\beta} \mathbf{A}, \hat{\mathbf{Y}} \rangle + \langle \sqrt{\beta} \mathbf{B}, \sqrt{\beta} \mathbf{U} \rangle + \sqrt{\beta} I \quad (14)$$

$$\hat{\mathbf{Y}} = \frac{1}{2} \left(X + \sqrt{\beta} \right) = \left| X - \sqrt{\beta} \right|$$

The nonlinear distortion added by clipping the signals to the power supplies affects the algorithm convergence. Essentially, the clipping effect adds unwanted harmonics to the signal. A value of $\beta=0.01$ (i.e. scaling by 100) is enough to meet a safe power supply value within ± 2 units. This scaling must also be evaluated with respect to the noise floor. Observe that a 1 unit amplitude carrier is scaled to 0.1 units, and after multiplying by a similar value carrier the signal is further reduced to 0.01 units. Therefore, once the signal acquires a value comparable to that of noise, it is imperative to consider the signal to noise ratio. In other words, a noise analysis must be made in order to determine the noise floor and thus the minimum scaling that can be set in the system. The result of this analysis sets the power supply range and if this is fixed, other parameters need to be modified accordingly and evaluated. To evaluate the SNR an additive white gaussian noise (AWGN) vector is added to the FDMA channel. Because the wanted portion of the signal is the DC level, a noise with mean 0 has very little effect on *algorithm convergence* and the SNR can be as high as -5dB. Total harmonic distortion is evaluated by sweeping the amplitudes from the 2nd harmonic to the 9th harmonic. We performed this sweep for a range from 0% to 30% of the nominal amplitude's value. We found that the largest distortion that can be tolerated in odd harmonics is 20%. The system cannot tolerate even harmonics so a fully differential architecture is suggested. Table 2 lists the individual parameter variation and the ranges for which was algorithm convergence obtained.

Table 2 Individual Parameter variation.

Parameter	Range
Frequency deviation from nominal frequency	$\tilde{F}_{dev} = \pm 0.1\% \text{ max}$
Offset added to the modulated signal	$\tilde{A}_{offs} = \pm 10\%$
Modulator amplitude mismatch from the +1, -1 range	$-20\% \leq \tilde{a}, \tilde{y} \leq 100\%$
Offset added to the output of the activation function	$-1 \leq \tilde{y}_{offs} \leq 1$
Offset added at the output of the multiplier	$\tilde{M}_{offs} \leq 0.48 \text{ units}$

4. Simulation Results

All parameter variations represent a worst case scenario in which only the parameter in the respective analysis is modified and all other parameters remain at their nominal values. The most relevant parameters obtained from

the simulations were the ones generated by offsets at the output of the modulators, at the output of the multipliers and also the total harmonic distortion that can be tolerated on the third harmonic. These parameters define the quality of the block implementation at the transistor level. So, a combined simulation without offset compensation for: $-0.1 \leq \tilde{A}_{offs} \leq 0.1$, $-0.5 \leq \tilde{M}_{offs} \leq 0.5$ and $0 \leq THD \leq 20\%$ was done and a 3-D parameter variation volume was generated, with the mass density representing the number of cycles required for the algorithm to converge. The total black areas represent convergence to an incorrect solution. The volume can be given an interpretation in a simplified way, by fixing one variable at its nominal value and determining the range of variation for the other two (as a transversal slice of the volume), or it can be interpreted as the density that satisfies most of the design requirements with the best compromise for all three variables.

Figure 4. (a) Offset at the multiplier outputs and THD projected on the Offset at the Modulators plane. (b) Offset at the Modulators and THD projected on the Offset at the multiplier outputs plane. (c) Offset at the multiplier outputs and Offset at the Modulators projected on the THD plane.

Figures 4a, b and c represent the volume density or the volume concentration of one parameter projected on a 2-D plane. The graphics are plotted in a Hot colormap where black represents where less density is present and white represents the opposite. These graphics were obtained by adding all the algorithm convergence cycles in two dimensions. These graphics give a very good idea of the parameter variations that can be supported. Evaluating the 3-D volume we can set this ranges as: $-0.025 \leq \tilde{A}_{offs} \leq 0.25$, $-0.5 \leq \tilde{M}_{offs} \leq 0.1$ and $0 \leq THD \leq 20\%$. From Figure 4a it can be seen that a reasonable THD tolerance is up to 7.5% when varying the offset of the multiplier from -0.5 to -0.2 units. From Figure 4b it can be seen that a reasonable tolerance is up to 10% of THD and varying the offset of the modulators from -0.025 to 0.025 units. From Figure 4c it can be seen that a reasonable tolerance is obtained by varying the offset of the multiplier from -0.5 to 0 units and the offset of the modulators from -0.05 to 0.05 units.

5. Conclusions

A modification of the Wave-Parallel computing technique is proposed to solve the communication and parallel processing needed in a real time CNN. Exploiting these characteristics, a parallel processing system was simulated using the concept of Transmit Multiply and Accumulate (TMAC), that led to a system that can realize most of the signal processing algorithm during the communication phase. The simulation resulted in a complete specification of the parameters for the different blocks that compose the system.

6. References

- [1] L. O.Chua and L. Yang: "Cellular Neural Networks: Theory", Transactions on Circuits and Systems, Vol. 35, No. 10, October 1988.
- [2] Craven, M. Curtis, K. and Hayes-Gill, B. "Consideration of multiplexing in neural network hardware", IEE Proc.-Circuits Devices Syst., Vol, 141, No 3, June 1994.
- [3] A. Murray: "Pulse Techniques in Neural VLSI: A Review", Proceedings of the IEEE 1992 Custom Integrated Circuits Conference, pp. 2204-2206.
- [4] Y. Yuminaka, Y Sasaki, T. Aoki and T. Higuchi: "Wave Parallel Computing Technique for Neural Networks Based on Amplitude-Modulated Waves", Proceedings of the 26th International Symposium on Multiple-Value Logic (ISMVL'96), pp. 210-215.
- [5] A.Q. Fong, A. Kanji and J. Pineda de Gyvez, "Time Multiplexing Scheme for Cellular Neural Networks based Image Processing," *Journal of Real-Time Imaging*, Academic Press, vol. 2, no. 3, pp. 231-239, Oct. 1996
- [6] O. Moreira-Tamayo and J. Pineda de Gyvez, "A Multiple Frequency Band Architecture for CNN", private communications.