

Chapter 12

Neural Networks for Measurement and Instrumentation in Virtual Environments

Emil M. Petriu
University of Ottawa, Ottawa, Ont., Canada

Abstract

Neural Networks (NNs), which are able to learn nonlinear behaviors from a limited set of measurement data, can provide efficient modeling solutions for many virtual reality applications. Due to their continuous memory behavior, NNs are able to provide instantaneously an estimation of the output value for input values that were not part of the initial training set. Hardware NNs consisting of a collection of simple neuron circuits provide the massive computational parallelism allowing for higher speed real-time models. A virtual prototyping environment for Electronic Design Automation (EDA) and a NN model for the 3D electromagnetic field are discussed in a representative case study.

1. Introduction

Virtual Reality (VR) is a computer based mirror of the physically reality. Synthetic and sensor-based, computer representations of 3D objects, sounds and other physical reality manifestations are integrated in a multi-media *Virtual Environment (VE)*, or *virtual world*, residing inside the computer. Virtual environments are dynamic representations where objects and phenomena are animated/programmed by scripts, by simulations of the laws of physics, or driven interactively directly by human operators and other real world objects and phenomena, Fig. 1.

The original VR concept has evolved finding practical applications in a variety of domains such as the industrial design, multimedia communications, telerobotics, medicine, and entertainment.

Distributed Virtual Environments (DVEs) run on several computers connected over a network allowing people to interact and collaborate in real time, sharing the same virtual

worlds. Collaborative DVEs require a broad range of networking, database, graphics, world modeling, real-time processing and user interface capabilities, [1].

Fig. 1

*Virtualized Reality*TM *Environment* (VRE), [2], is a generalization of the essentially synthetic VE concept. While still being a computer based world model, the VRE is a conformal representation of the mirrored real world based on sensor information about the real world objects and phenomena. *Augmented Reality* (AR) allows humans to combine their intrinsic reactive-behavior with higher-order world model representations of the immersive VRE systems. A *Human-Computer Interface* (HCI) should be able to couple the human operator and the VRE as transparently as possible. VRE allow for *no-penalty training* of the personnel in a variety industrial, transportation, military, and medical applications.

There are many applications such as remote sensing and telerobotics for hazardous environments requiring complex monitoring and intervention, which cannot be fully automated. A proper control of these operations cannot be accomplished without some AR telepresence capability allowing the human operator to experience the feeling that he/she is virtually immersed in the working environment. In such cases, human operators and intelligent sensing and actuator systems are working together as *symbionts*, Fig. 2, each contributing the best of their specific abilities, [3] and [4].

Fig. 2

VR methods are also successfully used in the *concurrent engineering* design. The traditional approach to the product development is based on a two-step process consisting of a *Computer Aided Design* (CAD) phase followed by a physical prototype-testing phase. The limitations of this approach are getting worse as the design paradigm shifts

from a sequential domain-by-domain optimization to a multi-domain concurrent design exercise. VR methods allow simulating the behavior of complex systems for a wide variety of initial conditions, excitations and systems configurations - often in a much shorter time than would be required to physically build and test a prototype experimentally. *Virtual Prototyping Environment* (VPE) design methods could be used to conduct interactive what-if experiments on a multi-domain virtual workbench. This results in shorter product development process than the classical approach, which requires for a series of physical prototypes to be built and tested.

2. Modeling natural objects, processes, and behaviors for real-time virtual environment applications

VREs and VPEs depend on the ability to develop and handle conformable (i.e., very close to the reality) models of the real world objects and phenomena. The *quality and the degree of the approximation* of these models can be determined only by validation against experimental measurements. The convenience of a model is determined by its ability to allow for extensive parametric studies, in which independent model parameters can be modified over a specified range in order to gain a global understanding of the response. Advanced computation techniques are needed to reduce the execution time of the models used in interactive VPE applications when analysis is coupled with optimization, which may require hundreds of iterations.

Model development problems are compounded by the fact that the physical systems often manifest behaviors that cannot be completely modeled by well-defined analytic techniques. Non-analytical representations obtained by experimental measurements have to be used to complete the description of these systems.

Most of the object models used in virtual environments are discrete. The objects are represented by a finite set of 3D sample points, or by a finite set of parametric curves, stored as *Look Up Tables* (LUTs). The fidelity of these discrete models is proportional with the cardinal of the finite set of samples or parametric curves. The size of the

corresponding LUTs is not a matter of concern thanks to the relatively low cost of today's RAM circuits. However, the main drawback of these discrete models is the need for a supplementary time to calculate by interpolation the parameters of each point that is not a sample point. This will increase the response time of the models, which in turn will affect the real-time performance of the interactive virtual environment.

Higher efficiency models could be implemented using NNs that can learn nonlinear behaviors from a limited set of measurement data, [5] and [6]. Despite the fact that the training set is finite, the resulting *NN model has a continuous behavior* similar to that of an analog computer model. An analog computer *solves the* linear or nonlinear differential and/or integral equations representing mathematical model of a given physical process. The coefficients of these equations must be exactly known as they are used to program the coefficient-potentiometers of the analog computer's computing - elements (OpAmps). The analog computer doesn't follow a sequential computation procedure. All its computing elements perform simultaneously and continuously. Because of the difficulties inherent in analog differentiation, the equation is rearranged so that it can be solved by integration rather than differentiation, [7]. A Neural Network does not require a prior mathematical model. A learning algorithm is used to adjust, sequentially by trail and error during the learning phase, the synaptic-weights of the neurons. Like the analog computer, the NN does not follow a sequential computation, all its neuron performing simultaneously and continuously. The neurons are also integrative-type processing elements.

The NN may take a relatively long time to learn the behavior of the system to be modeled, but this is not critical as it is done off-line. On the other hand, the recall phase, which is what actually counts in this type of interactive applications, is done in real-time. Due to their continuous memory, NNs are able to provide instantaneously an estimation of the output value for input values that were not part of the initial training set. Hardware NNs consisting of a collection of simple neuron circuits provide the massive computational parallelism allowing for even higher speed real-time models.

The following chapter discusses *Neural Network* (NN) techniques for real-time modeling and pattern recognition in VRE and VPE applications.

3. Hardware NN architectures for real-time modeling applications

Random-pulse data representation was proposed by von Neuman in 1956 as a cybernetic model explaining how algebraic operations with analog variables can be performed by simple logical gates, [8]. Due to the simplicity of its circuitry, this data representation was used to build low cost instrumentation during the 60s, when the digital IC technology was still relatively expensive, [9], [10] and [11]. There is a renewed interest in random-pulse data systems as their high functional packing density makes them quite suitable for the VLSI implementation of neural networks, [12], [13] and [14].

Random-pulse data are produced by adding a uniformly distributed dither to an analog input signal V and then passing the result through a 1-bit quantizer. As variables are represented by the statistical averages of random pulse streams, the resulting data processing system has a better tolerance to noise than the classical deterministic systems. The digital technology used to implement these *random-pulse machines*, [15], offers a number of advantages over the analog technology: modular and flexible design, higher internal noise immunity, simpler I/O interfaces. An important drawback is the relatively long time needed to get an acceptable accuracy for these statistical averages. However, the effects of this drawback can be mitigated by using a generalized multi-bit dithered quantization, [16] and [17].

Generalized random data representations are produced by *multi-bit analog/random-data conversion*, or dithered quantization, Fig. 3. The analog input V , supposed to have a relatively low variation rate, is mixed with an analog dither signal R uniformly distributed within a quantization interval, i.e between $+\Delta/2$ and $-\Delta/2$. The resulting analog signal VR is quantified with a b -bit resolution and then sampled by a clock CLK to produce the random sequence VRP of b -bit data.

Fig. 3

The ideal statistical average over an infinite number of samples of the random-data sequence VRP is:

$$E[VRP] = (k-1) \cdot p[(k-1.5)\Delta < VR < (k-0.5)\Delta] + k \cdot p[(k-0.5)\Delta < VR < (k+0.5)\Delta] = (k-1) \cdot \beta + k \cdot (1-\beta) = k - \beta$$

The estimation accuracy of the recovered value for V depends on the quantization resolution Δ , the finite number of samples that are averaged, and on the statistical properties of the dither R.

Because of the computational and functional similarity of a neuron and a correlator, we found useful to consider the following table giving relative speed performance figures for correlators with different quantization levels, [17]:

Quantization levels	Relative mean square error
2	72.23
3	5.75
4	2.75
...	...
8	1.23
...	...
analog	1

For instance, a basic 2-level (1-bit) random-pulse correlator will be 72.23 times slower than an ideal analog correlator calculating with the same accuracy the correlation function of two statistically independent Gaussian noise signals with amplitudes restricted within $\pm 3\sigma$. A 3-level (2-bit) correlator will be 5.75 times, and a 4-level (2-bit) correlator will be 2.75 times, slower than the analog correlator.

Based on these relative performance figures we have opted for a NN architecture using a 3-level generalized random-data representation, produced by a dithered 2-bit dead-zone quantizer. This gives, in our opinion, a good compromise between the processing speed

and the circuit complexity, [18], [19] and [20].

Random-data/analog conversion allows to estimate the deterministic component V of the random-data sequence as an average V^*_N over the finite set of N random-data $\{VRP_i \mid i=1,2,\dots,N\}$. This can be done using a *moving average* algorithm, [21] and [22]:

$$V^*_N = \frac{1}{N} \sum_{i=1}^N VRP_i = \frac{1}{N} (\sum_{i=1}^{N-1} VRP_i + VRP_N)$$

$$V^*_N = V^*_{N-1} + \frac{VRP_N - VRP_0}{N}$$

While the classical averaging requires the addition of N data, this iterative algorithm requires only an addition and a subtraction. The price for this simplification is the need for a shift register storing the whole set of the most recent N random data. Fig. 4 shows the mean square error of V^*_N , calculated over 256 samples, as function of the size of the moving average window in the case of the 1-bit and respectively 2-bit quantization.

Fig. 4

The analog/random-data and random-data/digital conversions are illustrated in Fig. 5 showing a step-like analog signal x_2 is converted to a sequence of random-pulses x_2RQ which is then reconverted as a moving average over $N=16$ random-pulses to produce the analog estimation $MAVx_2RQ$.

Fig. 5

One of the most attractive features of the random-data representation is that simple logical operations with individual pulses allow arithmetic operations with the analog variable represented by their respective random-pulse sequences to be carried out, [15]. This feature is still present in the case of low bit random-data representations.

The *arithmetic addition* of m signals $\{x_i \mid i=1,2,\dots,m\}$ represented by their b -bit random-data $\{X_i \mid i=1,2,\dots,m\}$ is carried out, as shown in Fig. 6, by time multiplexing the randomly decimated incoming random-data streams. The decimation is controlled by uniformly distributed random signals $\{S_i \mid i=1,2,\dots,m\}$ with $p(S_i) = 1/m$. This random sampling removes unwanted correlations between sequences with similar patterns, [10]. The random-data output sequence $Z = (X_1 + \dots + X_m)/m$ represents the resulting sum signal $z = x_1 + \dots + x_m$

Fig. 6

We will consider further the case of 3-level unbiased random-data produced by a dithered 2-bit dead-zone quantizer. The truth table for the *multiplication* of two signed 2-bit random-data, $Z = X * Y$ is:

		Y		
		0	1	-1
X		<i>00</i>	<i>01</i>	<i>10</i>
	0	<i>00</i>	0 <i>00</i>	0 <i>00</i>
1	<i>01</i>	0 <i>00</i>	1 <i>01</i>	-1 <i>10</i>
-1	<i>10</i>	0 <i>00</i>	-1 <i>10</i>	1 <i>01</i>

Fig. 7 shows the resulting logic circuit for this 3-level 2-bit random data multiplier.

Fig. 7

Neural-network architecture using generalized random-data representation

We have developed a NN hardware architecture based on the described 3-level 2-bit random data processing. Each synapse multiplies an incoming random data streams X_i , where $i=1,2,\dots,m$, by a synaptic-stored weight w_{ij} , which is adjusted during the learning phase. The positive-valued weights are considered *excitatory* and those with negative values are considered *inhibitory*. The neuron-body adds up the $DT_{ij} = X_i * w_{ij}$ signals from all the incoming post-synaptic channels, Fig. 8. The results of this addition are then integrated by a moving average random-data/digital converter. Since the neuron output will be used as a synaptic input to other neurons, a final *digital/random-data converter* stage is used to restore the randomness of Y_j .

Fig. 8

Using this 2-bit random-data neuron structure we implemented an auto-associative memory for pattern recognition applications, Fig. 9, which can learn input-pattern/target $\{P_q, t_q\}$ associations:

$$\{P_1, t_1\}, \{P_2, t_2\}, \dots, \{P_Q, t_Q\}$$

The NN is able to recognize any of the initially taught associations. If it receives an input $P=P_q$ then it produces an output $a = t_q$, for $q = 1,2,\dots,Q$. It can also recognize patterns corrupted by noise: i.e if the input is changed $P = P_q + \mathbf{d}$ the output will still be $a = t_q$.

Fig. 9

Fig. 10 shows as an example three training patterns, which represent the digits $\{0,1,2\}$ displayed as a 6x5 grid. Each white square is represented by a “-1”, and each black

square is represented by a “1”. To create the input vectors, we scan each 6x5 grid one column at a time. The weight matrix in this case is $W = P_1P_1^T + P_2P_2^T + P_3P_3^T$.

Fig. 10

In addition to recognizing all the patterns of the initial training set, the auto-associative NN is also able to deal with up to 30% noise-corrupted patterns as illustrated in Fig. 11.

Fig. 11

4. Case study: NN modeling of electromagnetic radiation for virtual prototyping environments.

The experimental VPE for *Electronic Design Automation* (EDA) developed at the University of Ottawa, [23] and [24], provides the following interactive object specification and manipulation functions:

- (i) updating geometric, electrical and material specifications of circuit components, as illustrated in Fig. 12;
- (ii) 3D manipulation of the position, shape, and size of the circuit components and layout.
- (iii) accounting for 3D EM and thermal field effects in different regions of the complex electronic circuit.

Fig. 12

The VPE scenes are composed of multiple 3D objects: printed circuit boards (PCBs), electronic components, and connectors, Fig. 13. Any object in the VPE is characterized by its 3D geometric shape, material property and *safety-envelopes* defining the 3D geometric space points where the intensity of a given field radiated by that object becomes smaller than a user-specified threshold value. Each type of field (EM, thermal, etc.) will have its own safety-envelope, whereas the geometric safety-envelope is the object shape itself.

Fig. 13

Electronic components are placed on a PCB where they are interconnected according to functional CAD specifications and to design constraints taking in consideration the EM interference between the PCB layout components. However, this design phase does not take in consideration the interference due to the EM and thermal fields radiated by the integrated circuits and other electronic components. These problems are identified and ironed out during the prototyping phase, which may take more *what-if* iterations until an acceptable circuit placement and connection routing solution is found. Traditionally this phase involves building and testing a series of physical prototypes, which may take considerable time.

Such a multi-domain virtual workbench allows conducting more expediently, in a concurrent engineering manner, *what-if* circuit-placement experiments. The VPE is able to detect the collisions between the safety-envelope of the circuit currently manipulated by the manipulator dragger and the safety-envelopes of other objects in the scene. When a collision is detected, the manipulated circuit returns to its last position before the collision.

Virtual prototyping allows a designer to test the prototype's behavior under a wide variety of initial conditions, excitations and systems configurations. This results in shorter product development process than the classical approach, which requires for a series of physical prototypes to be built and tested, Fig. 14.

Fig. 14

A key requirement for any VPE is the development of conformable models for all the physical objects and phenomena involved in that experiment. Neural networks, which can incorporate both analytical representations and descriptions captured by experimental measurements, provide convenient real-time models for the EM fields radiated by a variety of electronic components.

Neural Network modeling of EM fields

As a representative case, we consider the NN model of the 3D EM field radiated by a dielectric-ring resonator antenna, Fig. 15, with a ring radius a and a ring height d both with values from 4 to 8 mm in steps of 1 mm, and the dielectric constant of the ring ϵ_r with values from 20 to 50 in steps of 1, [25].

Fig. 15

The backpropagation NN using the Levenberg –Marquard algorithm consists of two input neurons, two hidden layers having 5 neurons with hyperbolic tangent activation function on each layer, and one output linear neuron.

The training data set, Fig. 16, was obtained analytically by calculating far-field values in 3D space and frequency from near-field data using the finite element method combined with the method of integral absorbing boundary conditions, [26]. Each geometrical configuration was solved using the *Finite Element Method* (FEM) for each of the 31 dielectric constants and for 1400 frequency steps from 2 to 16 GHz. The 200 epochs training took 55 s on a SPARC 10 workstation.

Fig. 16

The resulting NN model, shown in Fig. 17, has a continuous (analog memory) behavior allowing it to render EM field values with a higher sampling resolution than that of the initial training data set. It took only 0.5 s on the same SPARC 10 workstation to render 5,000 points of the 3D EM field model.

Fig. 17

Model Validation

While the VPE idea is gaining wider acceptance, it also becomes apparent the need for calibration techniques able to validate the conformance with reality of the models incorporated in these new prototyping tools.

Better experimental test-beds and validation methodologies are needed to check the performance of the computer models against the ultimate standard, which is the physical reality. Some of the challenges to be faced are:

- (i) the development of an experimental setup which should allow the desired manipulation of multi-domain (geometric, mechanical, electric, thermal, material, etc) design parameters;
- (ii) the identification and measurement of multi-domain phenomena which are considered to be behavioral characteristics for a given circuit design;
- (iii) finding the minimum set of experimental setups, cause-effect analytical/correlation methods, and calibration methodologies that provides a guaranty by interpolation (within acceptable error margins) the performance of the VPE computer models over wide ranges of multi-domain design parameters.

The analysis in homogeneous space simplifies greatly the problem of calculating *Far Field* (FF) EM values from *Near Field* (NF) measurements, [26]. The radiating *Device*

Under Test (DUT) is modeled as an array of short dipoles sitting on top of a table. The equation to solve for the electromagnetic fields is Helmholtz' wave equation:

$$\nabla^2 \vec{H} + k^2 \vec{H} = 0$$

in a homogeneous volume V bounded on one side by a surface where the magnetic field values of H are known through measurements and on the other side by the ground plane.

An explicit solution allowing to evaluate the magnetic field H anywhere in the volume V from its field values and its derivatives on a surface S_1 as proposed in [27]:

$$H(r') = \frac{1}{4\pi} \int_{S_1} \left[G(r, r') \frac{\partial H(r)}{\partial n} - H(r) \frac{\partial G(r, r')}{\partial n} \right] dS_1$$

where S_1 is the closed surface on which measurements are made, n is the normal to S_1 , and $G(r, r')$ is the free space Green's function.

This algorithm is independent of the type of radiation. While it shares some sources of error with other transform algorithms, the integral transform employed here is more immune to aliasing errors than the FFT-based algorithms. Another advantage over conventional FFT transforms is that the far-field results are available everywhere and not only at discrete points.

The EM field measurement system, [28], is shown in Fig. 18. It consists of a turning table with a highly conducting grounded surface on which the DUT is resting. The EM field probe can be positioned anywhere on a 90° arc of circle above the turning table.

Fig. 18

A special interface was developed for the control of the probe positioning and the collection of the measurement data via a spectrum analyzer. The turning table and the probe can be positioned as desired by steering them with position-servoed cables driven by motors placed outside an anechoic enclosure. The probe positioning system and the

steering cables are made out of non-magnetic and non-conductive material in order to minimize disturbance of the DUT's fields. EM field measurements are taken on both hemispherical surfaces, providing data for the interpolative calculation of the derivative's variation on the surface S_1 . The surfaces are closed with their symmetric image halves. This is possible due to the presence of the ground plane.

The actual angular positions of the table and that of the probe are measured using a video camera placed outside the enclosure. The azimuth angle j is recovered by encoding the periphery of the turning table with the terms of a 63-bit pseudorandom binary sequence, [29]. This arrangement allows to completely identify the 3D position parameters of the EM probe while it scans the NF around the DUT.

5. Conclusions

Virtual environment technology has found applications in a variety of domains such as the industrial design, multimedia communications, telerobotics, medicine, and entertainment.

*Virtualized Reality*TM environments, which provide more conformal representations of the mirrored physical world objects and phenomena, are valuable experimental tools for many industrial applications. Virtual environment efficiency depends on the ability to develop and handle conformable models of the physical objects and phenomena. As real world objects and phenomena more often than not manifest behaviors that cannot be completely modeled by analytic techniques, there is a need for non-analytical models driven by experimental measurements. Neural networks, which are able to learn nonlinear behaviors from a limited set of measurement data, can provide efficient modeling solutions for many virtualized reality applications.

Due to their continuous, analog-like, behavior, NNs are able to provide instantaneously an estimation of the output value for input values that were not part of the initial training set. Hardware NNs consisting of a collection of simple neuron circuits provide the massive computational parallelism allowing for even higher speed real-time models.

As a representative case of the *Virtualized Reality*TM technology, the virtual prototyping environments for EDA allow developers to interactively test complex electronic systems by running concurrently *what-if* multi-domain (mechanical, electrical, thermal, etc) experiments. For many an electromagnetic radiation problem it is not possible to derive an analytic solution. In this case, the only practical solution would be to use NN models trained by experimental measurement data. It is worthwhile to note that even when analytic solutions could be derived, the NN models still have better real time performance than classical numerical EM field methods.

As the performance of the NN models can be decided only by a validation against experimental measurements, there is need for a concerted effort to characterize and catalogue the EM radiated field signatures of all the integrated circuits and electronic components used for EDA.

Many real-world objects are characterized by a multitude of parameters of different nature. In order to capture this complexity, future efforts are need in order to develop composite models integrating 3D geometry, radiated fields, and other material properties. This could achieved by using a multi-sensor fusion system that integrates a variety of sensors covering all four phases in the perception process: far-away, near-to, touching, and manipulation.

Acknowledgment

This work was funded in part by NSERC, the Natural Sciences and Engineering Research Council of Canada, and CITO - Communications and Information Technology Ontario. The author gratefully acknowledges the contributions and assistance provided over the years by his collaborators Hassan Ali , Igor Ratner, Marius Cordea, Lichen Zhao, Arto Chubukjian, and Jie Mao.

References

- [1] R.C. Waters and J.W. Barrus, "The Rise of Shared Virtual Environments," *IEEE Spectrum*, Vol.34, No. 3, pp. 18-25, March 1997.
- [2] T. Kanade, *Virtualized Reality™*, <http://www.cs.cmu.edu/~virtualized-reality/>, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.
- [3] R.W. Picard, "Human-Computer Coupling," *Proc. IEEE*, Vol.86, No.8, pp. 1803-1807, Aug. 1998.
- [4] E.M. Petriu and T.E. Whalen, "Computer-Controlled Human Operators," *IEEE Instrum. Meas. Mag.*, Vol. 5, No. 1, pp. 35 -38, 2002.
- [5] C. Alippi and V. Piuri, "Neural Methodology for Prediction and Identification of Non-linear Dynamic Systems," in *Instrumentation and Measurement Technology and Applications*, (E.M. Petriu, Ed.), pp. 477-485, IEEE Technology Update Series, 1998.
- [6] C. Citterio, A. Pelagotti, V. Piuri, and L. Roca, "Function Approximation – A Fast-Convergence Neural Approach Based on Spectral Analysis," *IEEE Tr. Neural Networks*, Vol. 10, No. 4, pp. 725-740, July 1999.
- [7] A.S. Jackson, *Analog Computation*, McGraw-Hill Book Co., 1960.
- [8] J. von Neuman, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies*, (C.E. Shannon, Ed.), Princeton, NJ, Princeton University press, 1956
- [9] B.P.T. Veltman and H. Kwakernaak, "Theorie und Technik der Polaritatkorrelation fur die dynamische Analyse niederfrequenter Signale und Systeme," *Regelungstechnik*, vol. 9, pp. 357-364, Sept. 1961.
- [10] B.R. Gaines, "Stochastic computer thrives on noise," *Electronics*, pp.72-79, July 1967.
- [11] ***, "*SEM - Electronic Correlator*," NORMA Messtechnik Tech. Doc., PM 4707E.
- [12] A. Hamilton, A.F. Murray, D.J. Baxter, S. Churcher, H.M. Reekie, and L. Tarasenko, "Integrated pulse stream neural networks: results, issues, and pointers," *IEEE Trans. Neural Networks*, vol. 3, no. 3, pp. 385-393, May 1992.
- [13] M. van Daalen, T. Kosel, P. Jeavons, and J. Shawe-Taylor, "Emergent activation functions from a stochastic bit-stream neuron," *Electron Lett.*, vol. 30, no. 4, pp. 331-333, Feb. 1994.
- [14] E. Petriu, K. Watanabe, T. Yeap, "Applications of Random-Pulse Machine Concept to Neural Network Design," *IEEE Trans. Instrum. Meas.*, Vol. 45, No.2, pp.665-669, 1996.
- [15] S.T. Ribeiro, "Random-pulse machines," *IEEE Trans. Electron. Comp.*, vol. EC-16, no. 3, pp. 261-276, June 1967.
- [16] F. Castanie, "Signal processing by random reference quantizing," *Signal Processing*, vol. 1, no. 1, pp. 27-43, 1979.
- [17] K.-Y. Chang and D. Moore, "Modified digital correlator and its estimation errors," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 6, pp. 699-706, 1970.

- [18] E. Petriu, "Contributions to the improvement of correlator performance," *Dr. Eng. Thesis*, Polytechnic Institute of Timisoara, Romania, (in Romanian), 1978.
- [19] E. Pop, E.M. Petriu, "Influence of Reference Domain Instability Upon the Precision of Random Reference Quantizer with Uniformly Distributed Auxiliary Source," *Signal Processing (EURASIP)*, North Holland, Vol. 5, pp.87-96, 1983.
- [20] L. Zhao, "Random Pulse Artificial Neural Network Architecture," M.A.Sc. Thesis, OCIECE, University of Ottawa, Canada, 1998.
- [21] A.J. Miller, A.W. Brown, and P. Mars, "Moving-average output interface for digital stochastic computers," *Electron Lett.*, vol. 10, no. 20, pp. 419-420, Oct. 1974.
- [22] A.J. Miller and P. Mars, "Optimal estimation of digital stochastic sequences," *Int. J. Syst. Sci.*, vol. 8, no. 6, pp. 683-696, 1977.
- [23] E.M. Petriu, M. Cordea, and D.C. Petriu, "Virtual Prototyping Tools for Electronic Design Automation," *IEEE Instrum. Meas. Mag.*, Vol. 2, No. 2, pp. 28-31, 1999.
- [24] E.M. Petriu, M. Cordea, D.C. Petriu, Lou McNamee, "Modelling Issues in Virtual Prototyping Environments," *Proc. VIMS'99, IEEE Workshop on Virtual and Intelligent Measurement Systems*, pp. 1-5, Venice, Italy, May 1999.
- [25] I. Ratner, H.O. Ali, and E. Petriu, "Neural Network Simulation of a Dielectric Ring Resonator Antenna," *J. Systems Architecture*, vol. 44, pp. 569-581, 1998.
- [26] R. Laroussi and G.I. Costache, "Far-Field Predictions from Near-Field Measurements," *IEEE Tr. Electromagnetic Compatibility*, Vol. 36, No.3, pp. 189-195, 1994.
- [27] A.J. Poggio, and E.K. Miller, "Integral equation solutions of three dimensional scattering problems", in *Computer Techniques for Electro-magnetics*, Mittra R., ed., Pergamon Press, Oxford, 1973.
- [28] A. Rocznik, E. Petriu, and G.I. Costache, "3-D Electromagnetic Field Modeling Based on Near Field Measurements," *Proc. IMTC/96, IEEE Instrum. Meas. Technol. Conf.*, pp.1124-1127, Brussels, Belgium, 1996.
- [29] E. Petriu, W.S. McMath, S.K. Yeung, N. Trif, and T. Bieseman "Two-Dimensional Position Recovery for a Free-Ranging Automated Guided Vehicle," *IEEE Trans. Instrum. Meas.*, Vol. 42, No. 3, pp. 701-706, 1993.

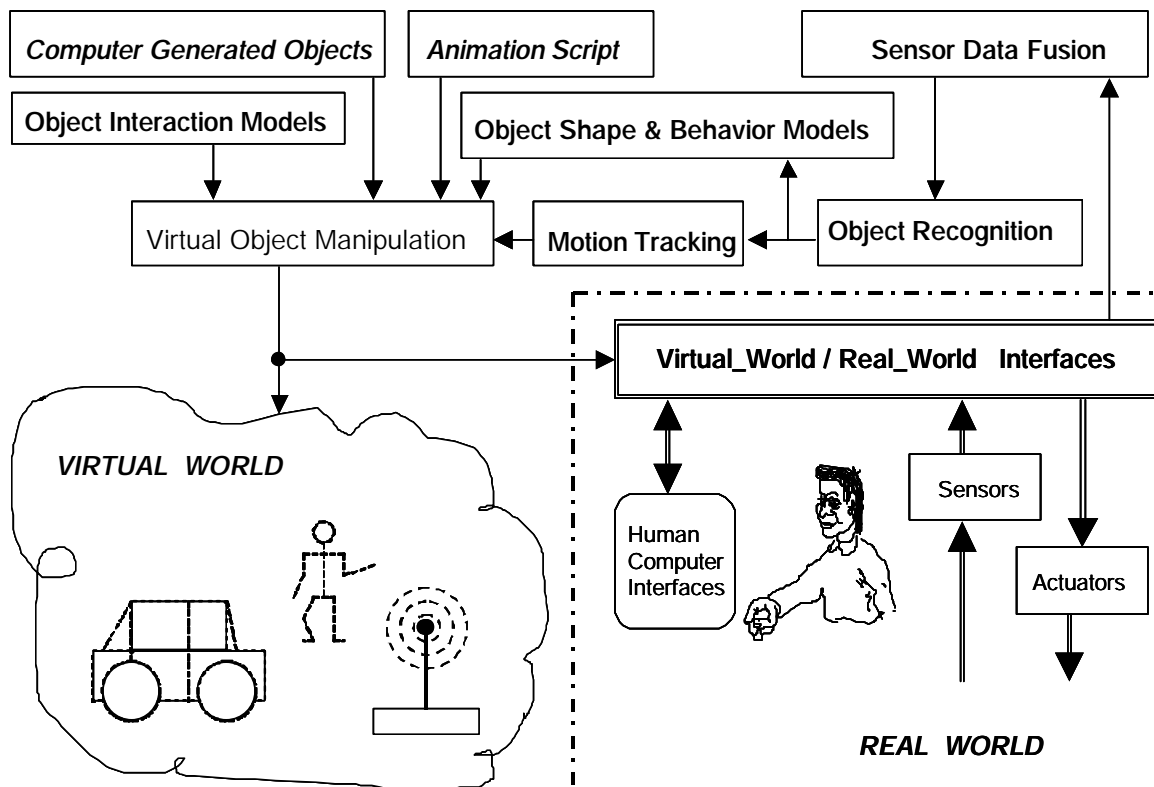


Fig. 1. Synthetic and sensor-based computer representations of natural objects and phenomena are integrated in a dynamic multi-media *Virtual Environment*.

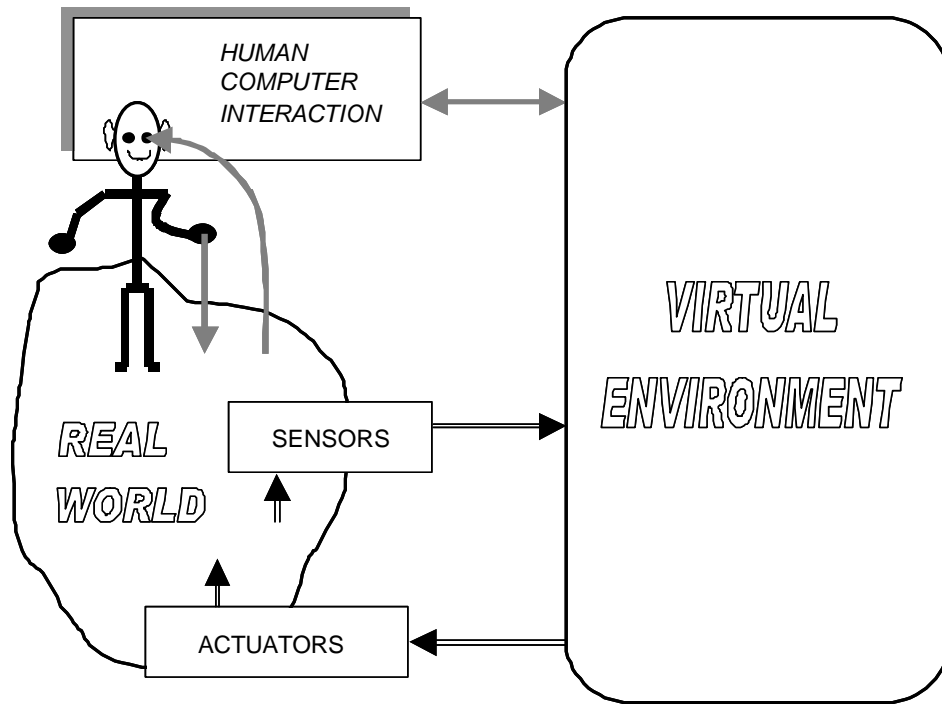


Fig. 2. Human-Computer-Real_World interaction in the augmented virtual reality.

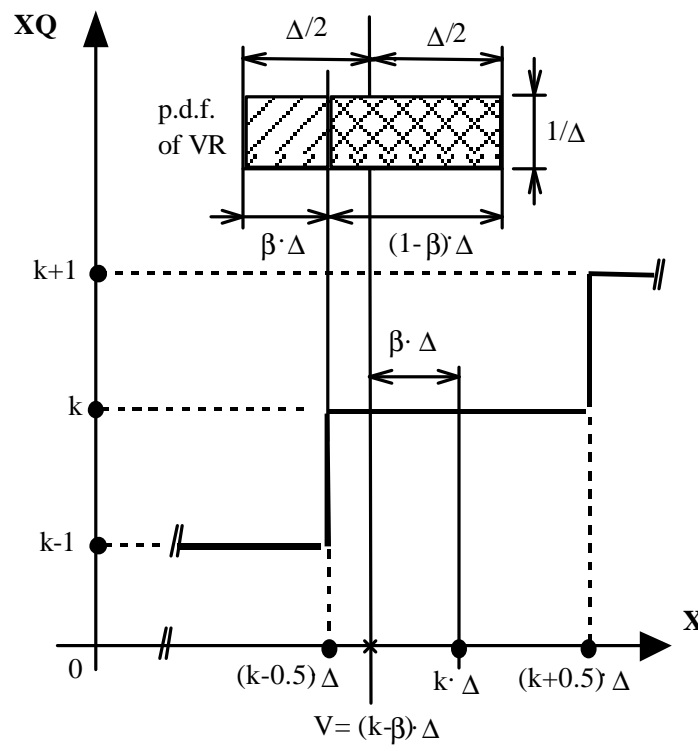
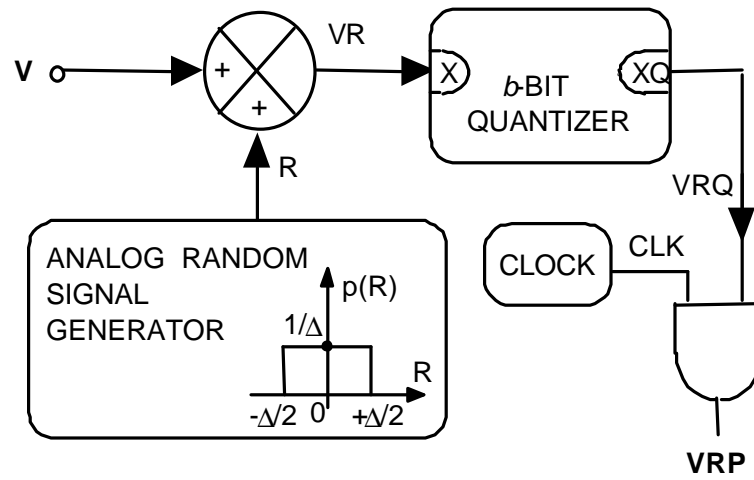


Fig. 3. Multi-bit analog/random-data conversion.

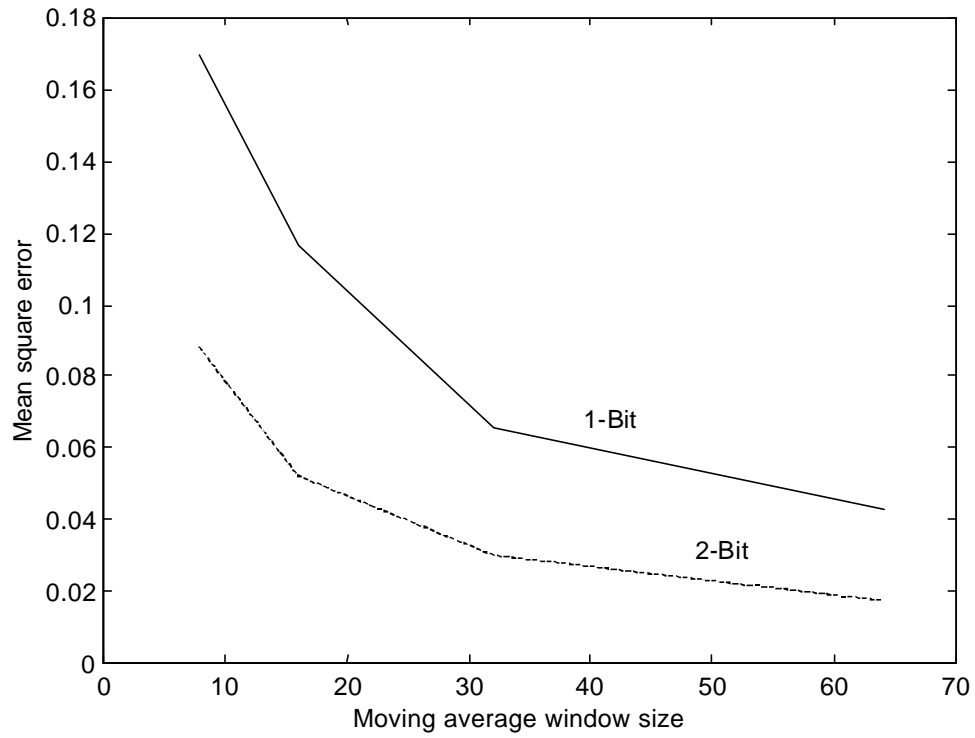


Fig. 4. Mean square errors of the moving average algorithm function of the size of the window.

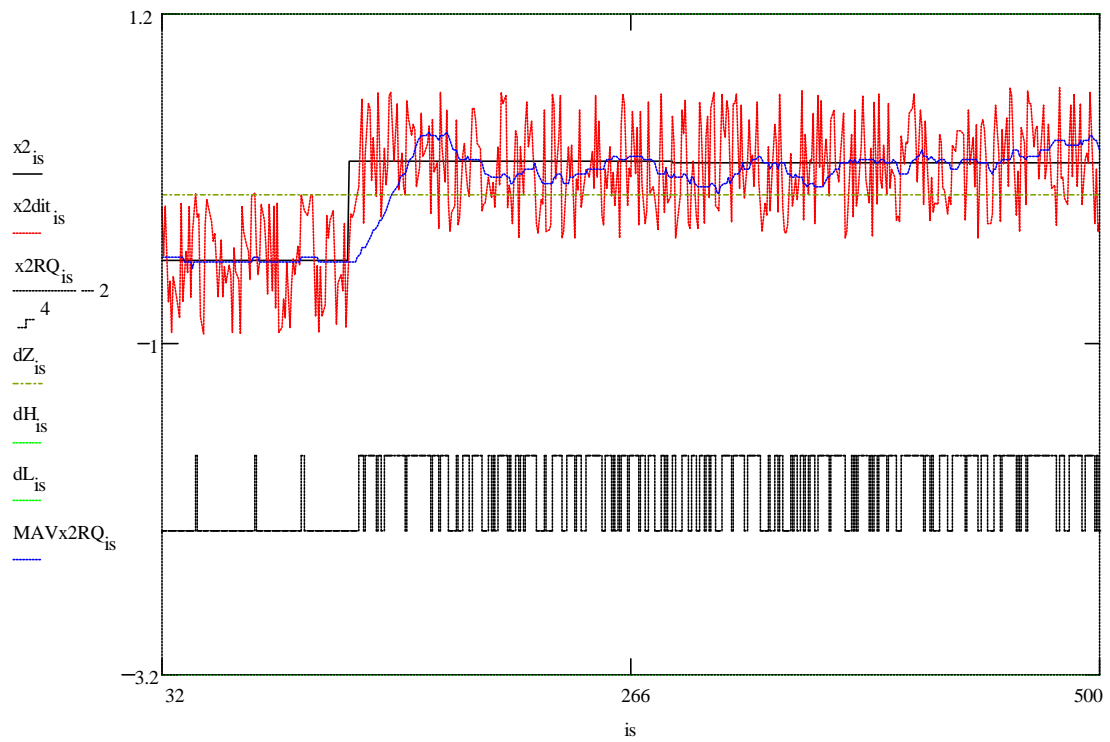


Fig. 5. Simulations illustrating the *analog/random-data* and *random-data/digital* conversions.

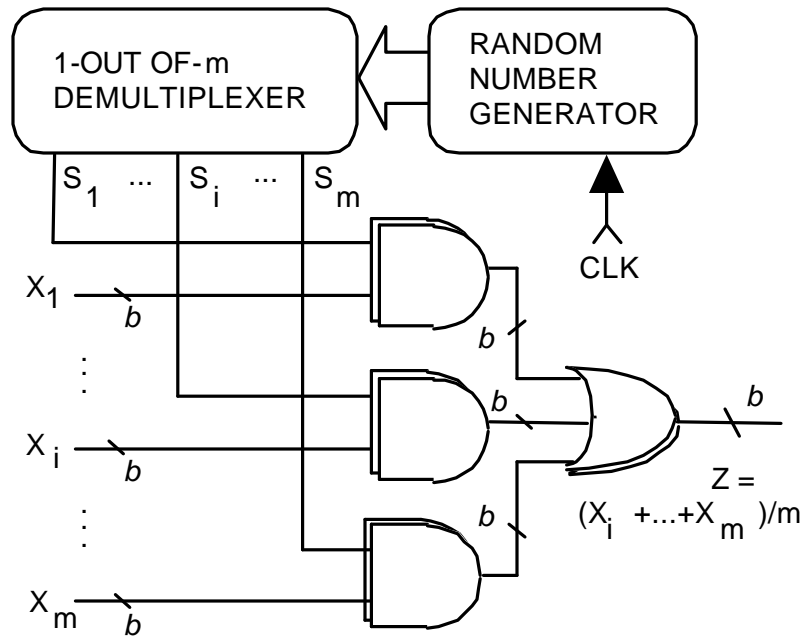


Fig. 6. Circuit performing the arithmetic addition of m signals represented by the b -bit random-data streams X_1, X_2, \dots, X_m

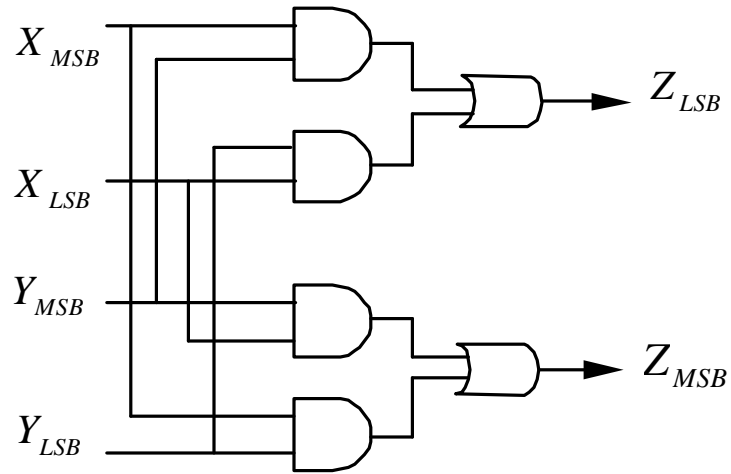


Fig. 7. Two-bit random-data multiplier.

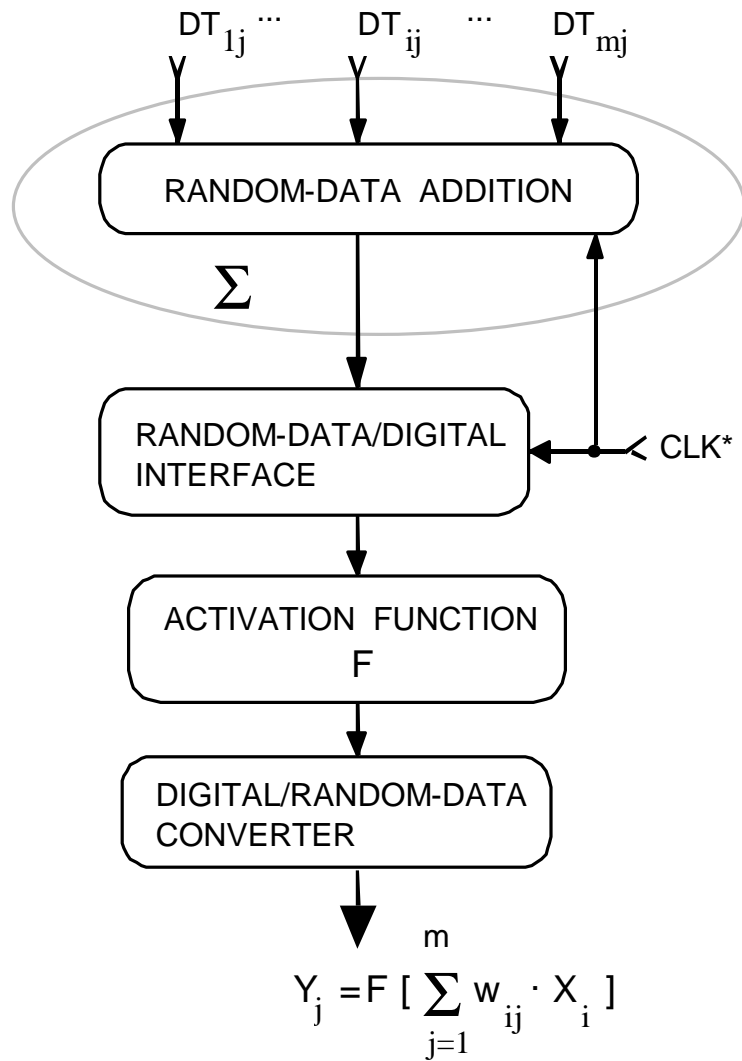


Fig. 8. Random-data implementation of the neuron body.

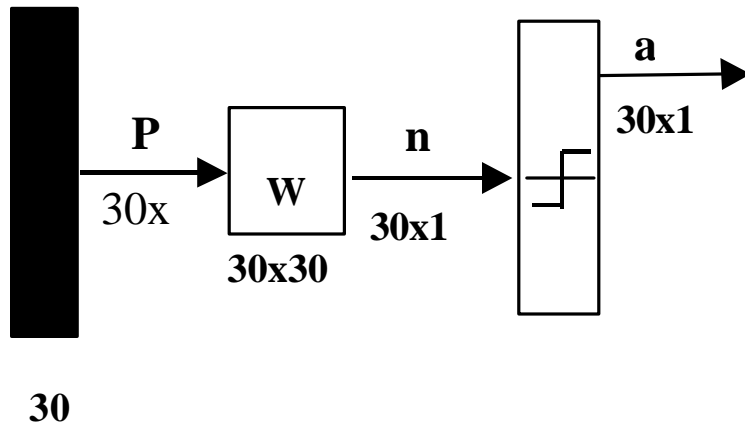


Fig. 9. Auto-associative memory NN architecture.

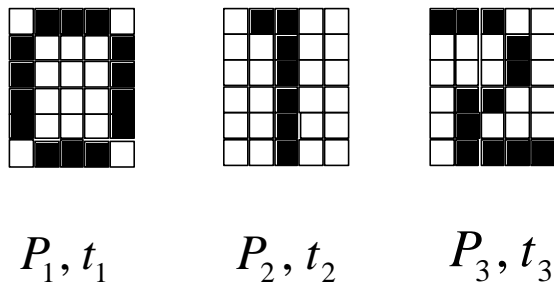


Fig. 10. Training set for the auto-associative NN.

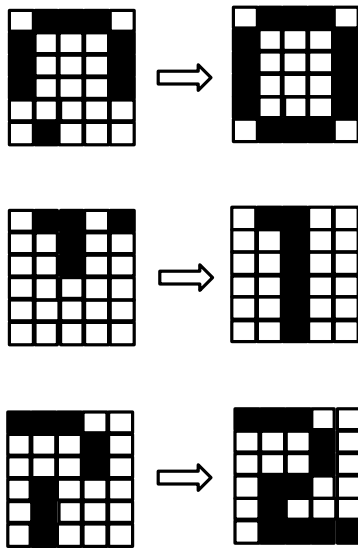


Fig. 11. Recovery of 30% occluded patterns by the auto-associative NN.

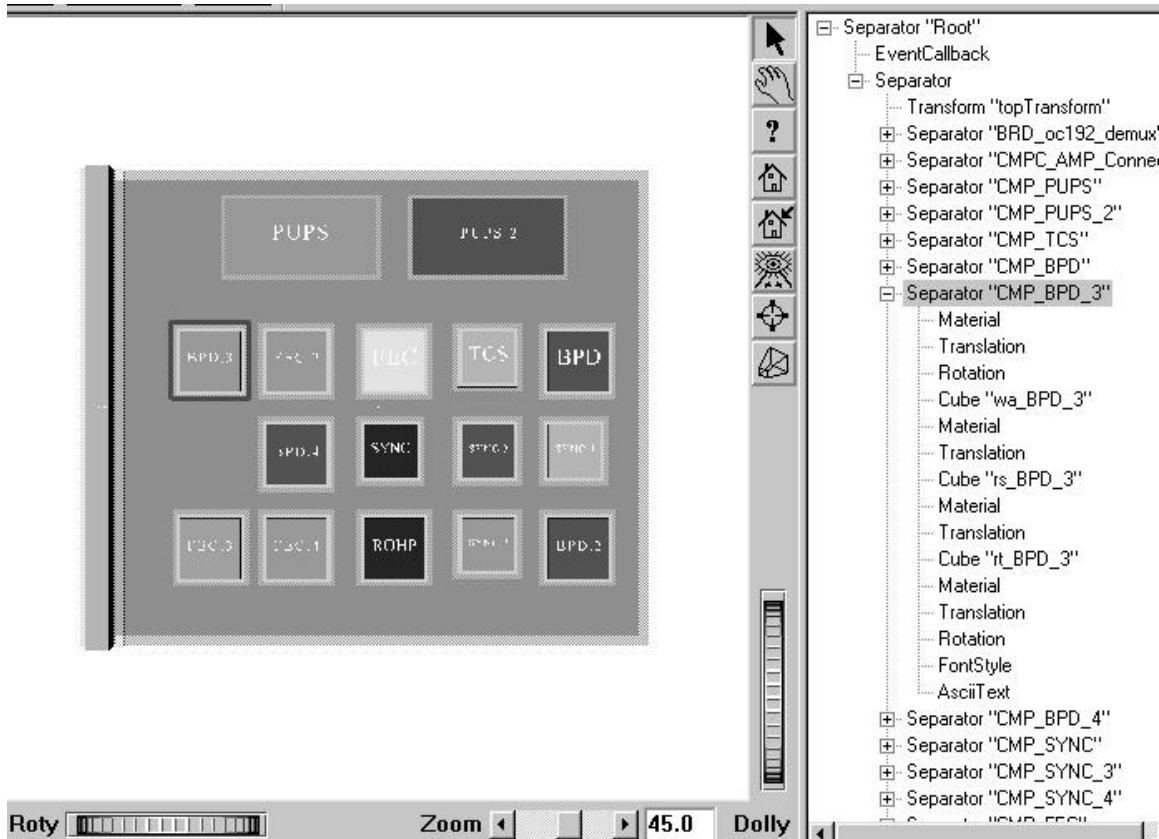


Fig. 12. Editing component properties in VPE.

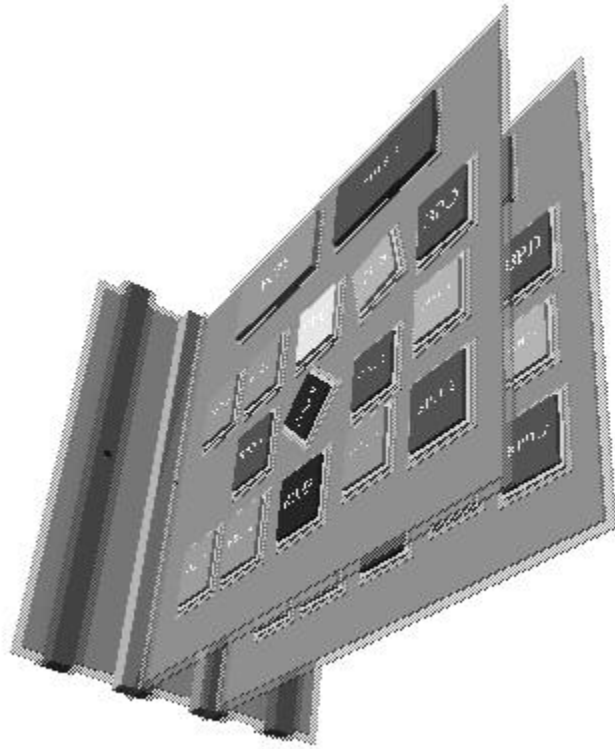


Fig. 13. 3D scene with a complex circuit assembly.

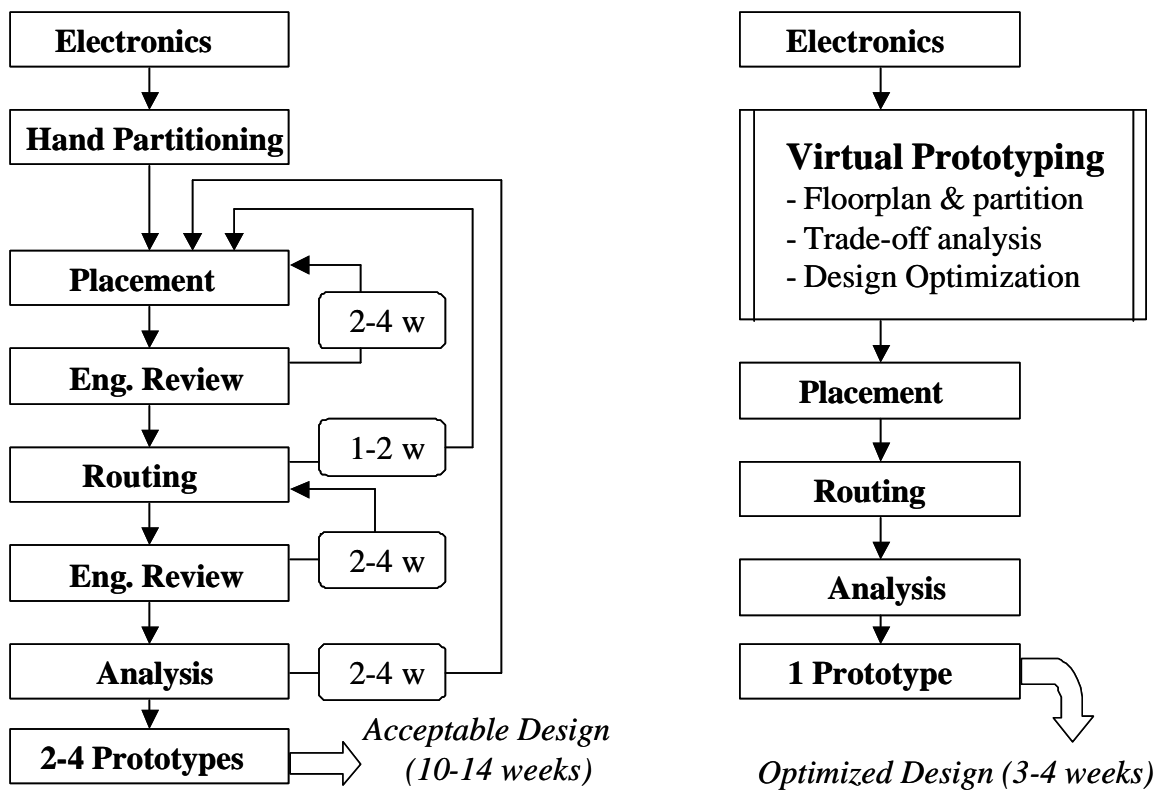


Fig. 14. Product design cycles for the traditional and respectively virtual prototyping.

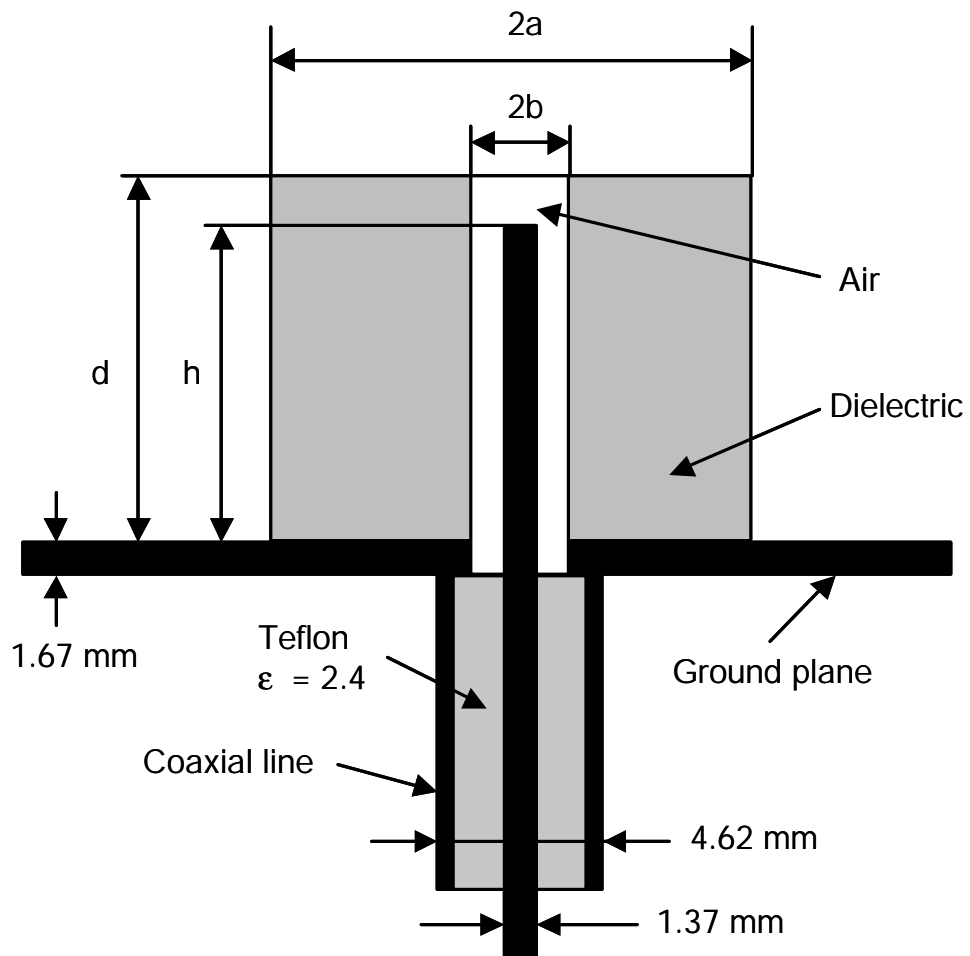


Fig. 15. The dielectric ring resonator antenna.

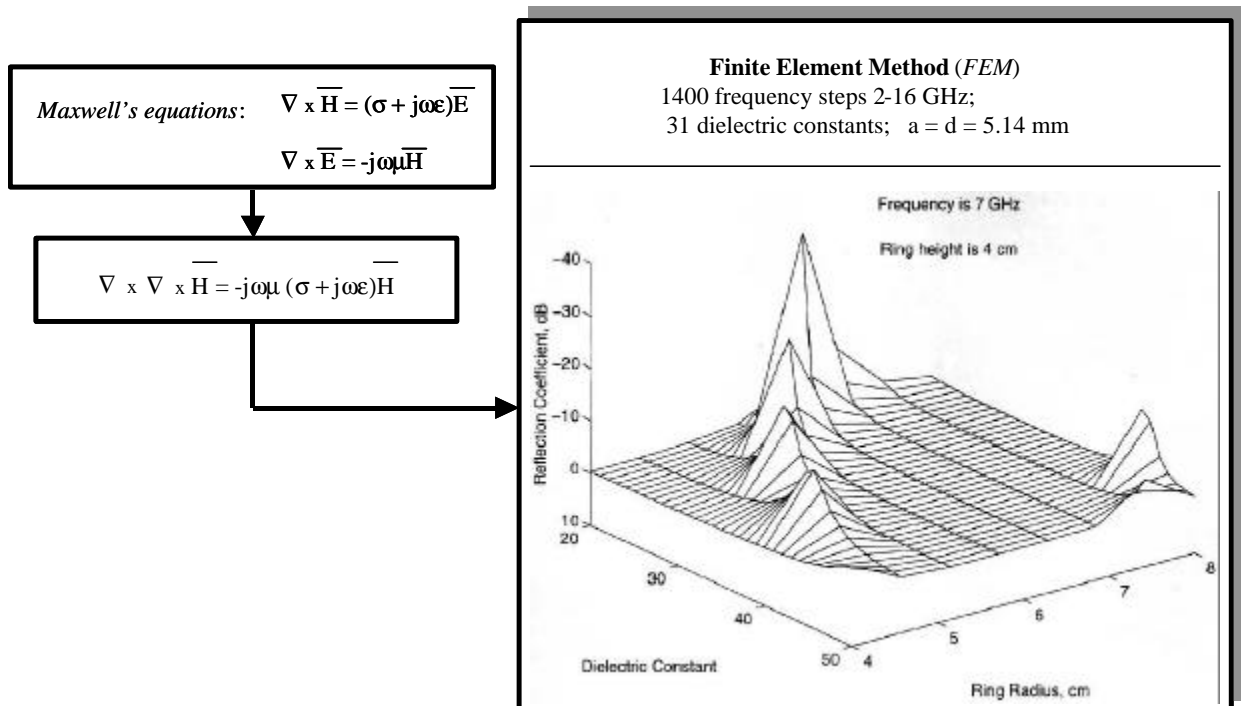


Fig. 16. The training data are obtained as analytically by calculating far-field values from near-field data using the finite element method.

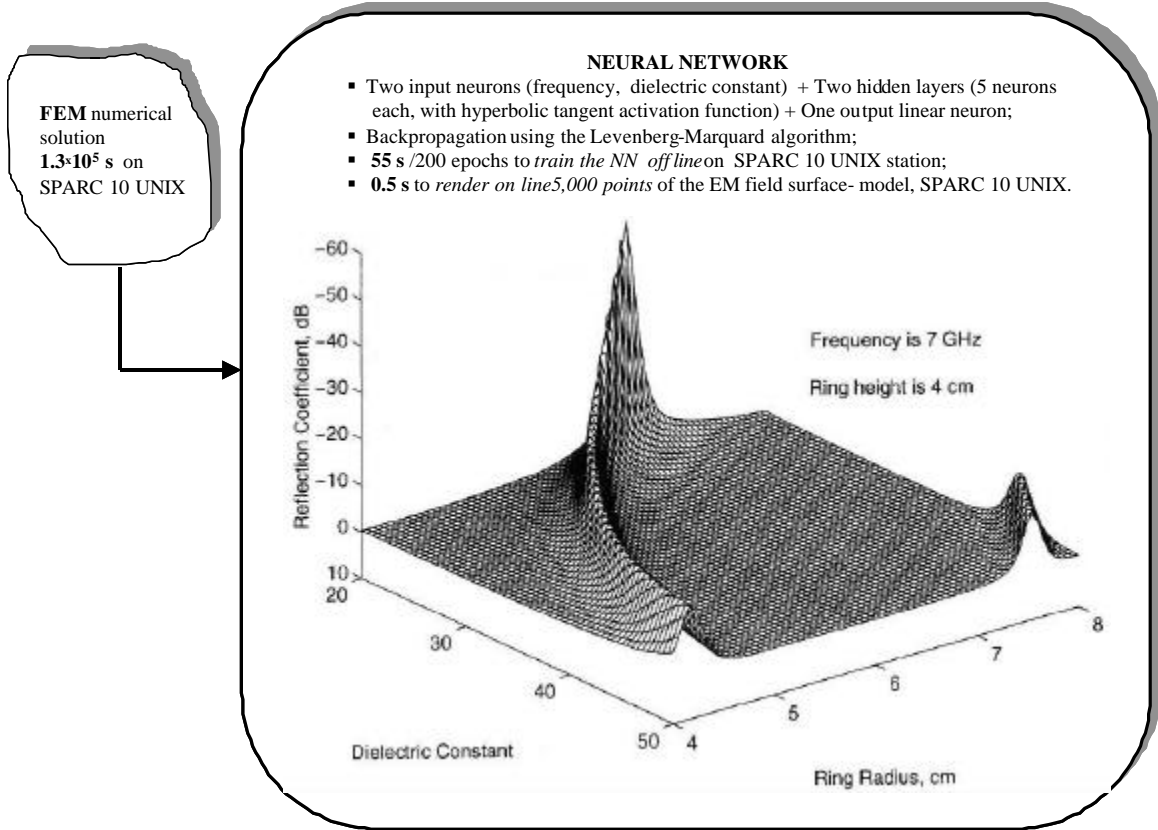


Fig. 17. NN model of the 3D EM field radiated by the dielectric-ring resonator antenna.

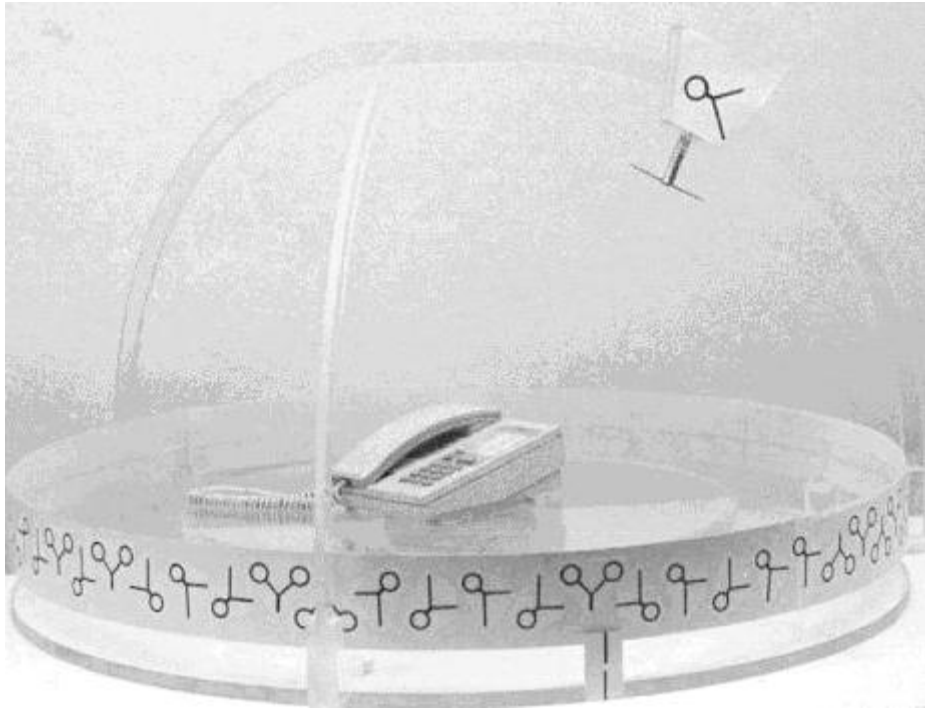


Fig. 18. Experimental setup for measuring the 3D EM near-field signature.