01 May 1990

The Parallel Implementation of the Waveform Relaxation Method for the Simulation of Structure-Preserved Power Systems

Marija D. Ilić

Mariesa Crow
Missouri University of Science and Technology, crow@mst.edu

Recommended Citation

M. D. Ilić and M. Crow, "The Parallel Implementation of the Waveform Relaxation Method for the Simulation of Structure-Preserved Power Systems," *Proceedings of the IEEE International Symposium on Circuits and Systems (1990, New Orleans, LA)*, vol. 2, pp. 1285-1288, Institute of Electrical and Electronics Engineers (IEEE), May 1990.
The definitive version is available at https://doi.org/10.1109/ISCAS.1990.112365

The Parallel Implementation of the Waveform Relaxation Method for the Simulation of Structure–Preserved Power Systems

by

M. L. Crow

Department of Electrical and Computer Engineering
University of Illinois at Urbana–Champaign
Urbana, IL 61801

M. Ilić

Laboratory for Electromagnetic and Electronic Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

In this paper, several results pertaining to the partitioning of the waveform relaxation algorithm for dynamic simulation are presented. The WR algorithm is extended to a structure–preserving power system model in which the loads are retained. This results in a system of differential/algebraic equations (DAEs). Power systems are shown to exhibit several dynamic characteristics which make them suitable for simulation by the WR method. One of the main contributions of this paper is a heuristic method for determining a fault–dependent partitioning of the power system for parallel implementation.

1 Introduction

In this paper, the authors extend the results of their earlier paper on *waveform relaxation* [6], which is a parallel algorithm for transient stability analysis. The waveform relaxation (WR) algorithm has not proven to be as effective as anticipated for VLSI circuit simulation for which it was originally developed, but this is due to various characteristics of VLSI circuits, such as floating capacitors, which are not encountered in power system problems. In fact, power systems exhibit several physical characteristics which may be exploited in the numerical simulation by waveform relaxation [5]. The "classical model" of an interconnected power system is often used in frequency studies. This model is represented by a system of ordinary differential equations (ODEs), and was the object of the previous paper by the authors. Often however, it is necessary to know the voltage profile of the entire system during and following a fault on the system. Severe contingencies may result in low–voltage problems at load terminals near the fault. To study the voltage characteristics of the system, the load terminals must be kept intact, adding a set of algebraic constraints to the original ODE system. In the next section, the authors present the waveform relaxation algorithm for systems of differential/algebraic equations (WRDAE). In section 3, the method of *adaptive partitioning* is introduced and then generalized and combined with the *textured model approach* [11] for static partitioning, to yield a near–optimal method for determining a fault–dependent dynamic partitioning for parallel implementation.

2 The WR Algorithm

In the usual power system simulation, either a "power flow" is executed or a "transient stability" analysis is performed. A power flow generally requires the solution of a nonlinear set of equations of the form

$$0 = g(x, y) \tag{1}$$

for x, where y are the independent voltages, the generated power, and the load power. The solution of equation (1) involves the repeated solution of large sets of linear equations of the form

$$Ax = b \tag{2}$$

where A is usually sparse, symmetric, and has between 3 to 10 nonzero entries in a row which does not depend on the size of A. Similarly, transient stability simulations involve large systems of differential/algebraic equations (DAEs):

$$\dot{x} = f(x, y) \tag{3}$$

$$0 = g(x, y) \tag{4}$$

where x now corresponds to the state variables of the generating units, and y to the network variables, and the loads are modeled as static loads. By using an integration technique, such as any of the backwards difference methods, equations (3) and (4) can also be transformed into a system of (2) which must be solved repeatedly at each integration time interval.

The waveform relaxation method is an iterative method for the numerical integration of the system of DAEs over a finite time interval. It is based on the Gauss–Seidel and Gauss–Jacobi relaxation methods used for solving large systems of algebraic equations. In the WR approach the DAE system is broken into subsystems which are solved independently, each subsystem using the previous iterate waveforms as "guesses" about the behavior of the state and algebraic variables in other subsystems. Waveforms are then exchanged between subsystems, and the subsystems are resolved with updated information about the other subsystems. This process is repeated until convergence is achieved. The waveform relaxation algorithm was first explored in context with power systems in [6]. This then led to the further development of the WR method for power system transient stability analysis [5].

The first step in the WR algorithm is to partition the system into subsystems in which tightly coupled variables are grouped together. In particular, the system is decomposed into r subsystems as:

$$\dot{x}_1 = f_1(x_1, \ldots, x_r, y_1, \ldots, y_r) \quad x_1(0) = x_{10} \tag{5}$$

$$0 = g_1(x_1, \ldots, x_r, y_1, \ldots, y_r) \quad y_1(0) = y_{10} \tag{6}$$

$$\vdots$$

$$\dot{x}_r = f_r(x_1, \ldots, x_r, y_1, \ldots, y_r) \quad x_r(0) = x_{r0} \tag{7}$$

$$0 = g_r(x_1, \ldots, x_r, y_1, \ldots, y_r) \quad y_r(0) = y_{r0} \tag{8}$$

where $x_i \in R^{n_i}, y_i \in R^{m_i}, \sum_{i=1}^{r} n_i = n, \sum_{i=1}^{r} m_i = m, f_i : R^{n+m} \rightarrow R^{n_i}$, and $g_i : R^{n+m} \rightarrow R^{m_i}$. The WRDAE algorithm for solving the system of equations (5) through (8) is summarized below:

Algorithm 1 – Gauss-Jacobi WRDAE algorithm.

$k \leftarrow 0$

Guess some $x_i^0(t)$ such that $x_i^{k+1}(0) = x_i(0)$.

Guess some $y_i^0(t)$ such that $y_i^{k+1}(0) = y_i(0)$.

repeat {

$k \leftarrow k+1$.

for each $(i \in \{i, \ldots, r\})$ solve on $[0, T]$

$$\begin{aligned} \dot{x}_i^{k+1} &= f_i(x_1^k, ..., x_i^{k+1}, ..., x_r^k, y_1^k, ..., y_i^{k+1}, ..., y_r^k) \\ 0 &= g_i(x_1^k, ..., x_i^{k+1}, ..., x_r^k, y_1^k, ..., y_i^{k+1}, ..., y_r^k) \end{aligned}$$

} **until** $(\|x^{k+1} - x^k\| \leq \epsilon_x$ and $\|y^{k+1} - y^k\| \leq \epsilon_y)$

Note that each subsystem may be solved independently of the other subsystems; the integration of each subsystem may proceed in parallel. At this point it is appropriate to point out that although the WRDAE algorithm will converge for any chosen partitioning [10], the choice of partitioning greatly influences the rate of convergence of the WR method. If tightly coupled nodes are not grouped together or if too many nodes are placed in the same partition, the rate of convergence may be greatly decreased and the WR method is no longer efficient. The choice of partitions will be discussed in depth in the following section. A slightly different WR algorithm for DAEs was introduced in [7] for VLSI circuits, but this particular algorithm is not explicitly suited for power system applications. The presence of the floating capacitors causes a "nesting" of iterations which causes the convergence to progress slowly. The system structure which causes this phenomena is absent in power systems. The interested reader is referred to [2] for a more complete comparison of these two algorithms.

3 Power System Applications of the WR Algorithm

The basic premise of the waveform relaxation algorithm is that a large system may be decomposed into several smaller weakly coupled subsystems, each of which may exhibit vastly different behaviors. Although the WR method will still converge in spite of poorly chosen subsystems, the rate at which the method converges is greatly affected by the choice of subsystems [10]. Because of this consequence, the first step in any WR–based simulation is to *partition* the system, that is, to determine which nodes in the system should be put together and solved in a direct manner. As the size and complexity of the systems to be simulated increase, it becomes imperative for rapid convergence to partition the network into subsystems which are weakly coupled.

Several approaches have been taken to find a "good" partitioning. The first and most straightforward method is to have the user partition the system, but often a partition which appeals to the user in a physical sense, may not be a good partitioning for the WR algorithm. A second approach is known as the *diagonally dominant loop* partitioning [10] which will lump two nodes (i, j) together if

$$\left| \frac{a_{ij} a_{ji}}{a_{ii} a_{jj}} \right| > \alpha \tag{9}$$

is satisfied, where α is a problem dependent parameter. This criterion sometimes produces misleading results since it focuses only on how strongly coupled the two nodes are to each other, while neglecting their coupling to other nodes. In other words, if two nodes are very strongly coupled, all other couplings to these nodes will appear weak by comparison, even though they may be strong enough to slow down the WR process.

Since the intent of the partitioning is to improve the speed of convergence of the relaxation, it is sensible to try to exploit the natural decoupling of the physical system. An approach specifically tailored to power system simulation would be to exploit the *coherency* characteristics inherent in power system behavior. The coherency property of power systems is generally defined as the tendency of some synchronous generators to "swing together" during the multimachine transients following a disturbance [1]. This may be loosely interpreted as the coherent group of generators being most affected by the same slow or dominant modes of the system, such that their global movement is similar.

It turns out that there is indeed a correlation between coherency grouping and the partitioning for the waveform relaxation method. Two subsystems have been defined to be *weakly coupled* if the dynamic properties of the overall system, such as its eigenvalues, can be determined, at least approximately, from the subsystems treated separately [1]. This coupling effect is a very important property that the waveform relaxation method must exploit to ensure the efficiency of the algorithm. Much research effort has been expended on ways to determine coherent groups, such that the coherent groups may be replaced by a reasonable approximation of an aggregate machine. One of the most accurate and widely used methods for determining coherent groups is the *slow coherency* method [1], in which the slowest modes of a system are used as a basis around which sets of generators are grouped. The basic goal of the slow coherency method is to obtain groups of generators which are strongly coupled among themselves and weakly coupled to other portions of the system. Although proving the mathematical relationship between coherency and WR partitioning in a rigorous manner is quite difficult, the criteria on which they are both based is the same. Thus, it is a natural extension to use a coherency based partitioning for the WR method applied to power systems. The coherency of generating units in a property which arises from a system being modeled as a set of ODEs. For a system of DAEs, the coherency method is used to identify groups of differential variables (corresponding to the generating units) to which the algebraic variables are assigned. This is accomplished by assigning "electrically close" load buses to the groups defined by the coherent generating units. Indeed, simulation results of the waveform relaxation method applied to the simulation of the classical model power system illustrate that the iterates consistently converge more rapidly for a coherency based partitioning than for any other partitioning [3].

In the WR algorithm, several factors contribute to the rate at which the method converges. It is a well–known fact that in block relaxation methods, the larger the blocks are, the more rapidly the iterates converge. On the other hand, the larger the blocks are, the more computation is necessary per iteration and thus the advantages of using the relaxation methods are lost. To have a reasonable convergence speed, it is required to partition the system at places where the coupling is weak, while keeping the size of the partitions to a minimum. Therefore, finding a partitioning which lumps together tightly coupled nodes without lumping together too many nodes is difficult. One approach to partitioning power systems is to use coherent groups as a basis for the partitions.

First, coherent groups may vary depending on the location and severity of the disturbance, especially if the fault causes a change in the system structure by forcing lines to be removed. Second, a coherent group may consist of several areas in which two of the areas are coupled with the third area, but are weakly coupled to each other. Most partitioning algorithms would group all of the areas into one large moderately coupled partition. This scenario is illustrated in Figure 1. If groups of moderately coupled generators are grouped into one partition, the size of some of the blocks may grow considerably. This may result in a considerable loss of savings due to the increased size

of the blocks. These cases may lead to the need for *adaptive partitioning*. This is the process of dynamically changing the partitioning as the WR iterates progress.

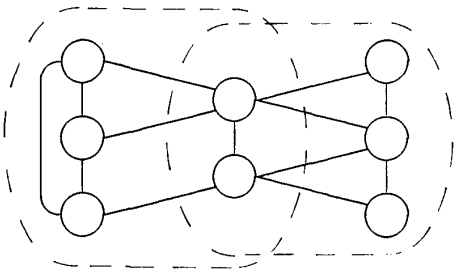

Figure 1: Two Overlapped Coherent Groups

3.1 Adaptive Partitioning

Adaptive partitioning could be used in the case where the pre-fault coherent groups were altered by the fault, and the convergence of the WR algorithm is progressing too slowly for maximum efficiency. The loss of rapid convergence indicates that the weak coupling between partitions is not being maintained. In order to correct this situation, the generators may be grouped into a new set of partitions. One rapid method for determining a new partitioning is the "strict definition" approach. This method groups the generators strictly according to the definition of coherency, that is, two generators are considered to be coherent if

$$|\delta_i(t) - \delta_j(t)| < \epsilon \qquad (10)$$

where ϵ is some small parameter. This method has the characteristic of being receptive to system structure changes.

The second case of overlapped coupling outlined above leads to a slightly more complex approach to adaptive partitioning. In this situation, it is difficult to ascertain which sets of generators are strongly coupled and which couplings can be neglected. Two sets of generators may be strongly coupled to a third set, but are also relatively insensitive to each other. Placing all three groups into one large partition defeats the purpose of the waveform relaxation method. One way to bypass this drawback is to alternately group the third set with the first and second sets. Thus, this leads to two strongly coupled *overlapping* partitions. This idea of overlapped partitioning was first explored in [9] in context with unidirectional VLSI circuits.

An adaptive partitioning scheme based on the overlapped groups would switch back and forth between partitionings as the WR method iterates progressed. The primary advantage of the adaptive partitioning would be to reduce the overall size of the partitions while still exploiting the strong coupling between adjacent subsystems. The convergence of the adaptive partitioning scheme was established in [8] for linear time invariant systems. The conditions for convergence are given in the following theorem.

Theorem 1 *If the product of the spectral radii of each of the iteration matrices for each partitioning is less than one, then the relaxation will converge.*

Proof: The proof given here is summarized from [8].
Let $M_1 - N_1, M_2 - N_2, \ldots, M_j - N_j$ be j splittings of A, where each splitting corresponds to a distinct partitioning of P_j of the system

$$\dot{x}(t) = Ax(t)$$

and $M_j - N_j = A$ for all j splittings. Let $K_i(z) \triangleq (zI + M_i)^{-1} N_i$ which is the iteration operator for the i^{th} partitioning. Then

$$x^k = \left(\prod_{i=1}^{k} \prod^{j} K_i(z) \right) x^0 = \left(\prod^{k} (K_1(z) K_2(z) \ldots K_j(z)) \right) x^0$$

This will converge as $k \to \infty$ if and only if

$$lim_{k \to \infty} \quad \sigma \left(\prod_{i=1}^{k} \prod^{j} K_i(z) \right) \to 0$$

which occurs if and only if

$$\sigma \left(\prod_{i=1}^{j} K_i(z) \right) < 1 \qquad \square$$

This result can be applied to the two scenarios previously discussed. In the first case, all that is required is that the iteration operator of the second partitioning have a spectral radius less than one. Even if the first partitioning is nonconvergent ($\sigma > 1$), switching permanently to the new partitioning at some finite iteration will eventually drive any incurred error to zero. In the second situation, the effect is more subtle, yet still powerful enough to maintain convergence. The product of the iteration operators $\prod_{i=1}^{j} K_i(z)$ can be thought of as an iteration operator in itself. Thus, if some of the iteration operators which comprise the product operator have a spectral radius greater than one, their effect can be negated by those operators which have spectral radii less than one. This results in a conglomerate iteration operator whose spectral radius is the product of the iteration operators of which it is comprised. Thus, if it has a spectral radius less then one, the relaxation will converge.

In the above explanation of adaptive partitioning, it is assumed that every node is processed at every iteration, while alternating the partitioning, or grouping, of each node. In many large systems, however, several groups of nodes may be *latent*, or relatively inactive. In this case, it is not necessary to process all of the nodes at every iteration, only those groups of nodes which are currently active need to be processed. One promising method of adaptive partitioning which has been proposed to optimize the parallel solution of nonlinear algebraic equations is the *textured model approach* [11].

3.2 The Textured Model Approach to Partitioning

The basic premise of the textured model approach to solving the nonlinear algebraic system is to arrange the nodes of the system into groups around active inputs, assuming the influence of the active inputs upon the outputs drops below a preset threshold outside of the group boundaries. The groups or partitions are uniformly sized for minimum computation time of the intended algorithm. Nonoverlapping groups are then assigned to several *leaves* in roughly equal numbers. Thus, each partition on any leaf may be solved in parallel if the interaction with the rest of the system is frozen at the group boundaries. The results obtained for one leaf may then be transferred to the next leaf directly. Thus, this method is capable of using the same number of processors continually and with minimal intervening sequential steps. This results in an almost complete elimination of idle time for the processors.

The textured model approach can be directly extended to the waveform relaxation method for power systems. The waveform relaxation method is an iterative algorithm which must meet the same objectives as iterative methods for solving large

nonlinear algebraic equations. The waveform relaxation algorithm is based on the reasoning that each partition of the system contains strongly coupled states which are weakly coupled to the states outside the boundaries of the partition, and the interaction across these may be frozen.

To this end, the authors present the following theorem about the convergence of the textured model approach applied to the WR algorithm.

Theorem 2 *Consider a WR algorithm to which the textured model approach has been applied*

$$\begin{aligned} \dot{x}^k &= \tilde{f}(x^k, x^{k-1}, ..., x^{k-L}, y^k, y^{k-1}, ..., y^{k-L}) \quad k \geq L \\ 0 &= \tilde{g}(x^k, x^{k-1}, ..., x^{k-L}, y^k, y^{k-1}, ..., y^{k-L}) \end{aligned}$$

where L is the number of leaves, $x \in R^n$, and $y \in R^m$. If all the nodes are covered, then for any initial guesses $(x^0(t), y^0(t); t \in [0,T]), (x^1(t), y^1(t); t \in [0,T]), \ldots, (x^L(t), y^L(t); t \in [0,T])$ the sequence $\left\{(\dot{x}^k(t), x^k(t), y^k(t); t \in [0,T])\right\}_{k=L}^{\infty}$ generated by the WR algorithm converges uniformly to $(\dot{\hat{x}}(t), \hat{x}(t), \hat{y}(t); t \in [0,T])$.

The proof of this theorem may be found in [2] and is therefore omitted for brevity. The functions $\tilde{f}$ and $\tilde{g}$ may be different from the functions f and g of equations (3) and (4). In this case, however, it is not required that the functions $\tilde{f}$ and $\tilde{g}$ be found explicitly, only that they exist. A systematic method for finding this "canonical form" exists for the linear case, and may be generalized to the nonlinear case.

The primary goal of the textured model approach is to establish a set of criteria by which partitionings may be chosen such that the simulation progresses with maximum efficiency. In the case of a global integration time step, this corresponds to a set of groups which are roughly the same size, so that the computation per group per iteration is equivalent. This will then minimize the idle time of each processor. It is not essential that all the groups are the same size, but that all the groups assigned to a particular leaf must contain essentially the same size group. The size of the groups is determined by the size of the sensitivity threshold and by merging overlapping groups until uniformly sized groups are achieved. If the group sizes are larger than desired, the threshold may be raised. If the groups sizes are too small for the desired convergence rates, the threshold may be lowered to increase the size of the groups.

When taking multirate integration into account, the task of determining group size and leaf assignment is more complex. Groups which are chosen according to a global time step criteria are roughly the same size, but when integrated with differing time steps may result in vastly different simulation times. This drawback may be overcome by lowering the threshold criteria for those buses and generators far from the fault. This will result in larger groups of buses being associated with generators which are distant and therefore considered to be latent with respect to generators nearer the fault. In [4], Crow and Ilić extended the static textured model approach to the dynamic problem of grouping and leaf assignment of the partitions which arise the the WR method. The preliminary results, reported in [4], suggest that the combination of the textured model approach and the waveform relaxation method together may lead to near optimal parallel simulation of power system dynamic behavior. For practical uses, the initial grouping stage must be performed off-line, according to some nominal sensitivity threshold, but as the simulation progresses, adaptive grouping may be implemented to account for system structure changes due to a contingency. When a fault occurs, these nominal size groups may be merged to form larger groups corresponding to their distance from the fault and stratified according to the number of precessors available. This process may be performed rapidly at run-time without serious loss of efficiency.

4 Conclusions and Acknowledgements

In this paper, the parallel implementation of the waveform relaxation algorithm is explored using the dynamic characteristics of the power system as a basis for additional simulation efficiency. The physical structure of the power system may be exploited in determining a fault-dependent partitioning of the power system for the parallel implementation of the WR method.

The authors acknowledge the research support under the NSF grant NSF-ECS83-52221 and the support of the Power Affiliates Program of the University of Illinois. The authors would also like to thank J. White, M. A. Pai, and P. Sauer for their many conversations and invaluable advice.

References

[1] J. Chow and P. Kokotovic, "Time scale modeling of sparse dynamic networks," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 8, pp. 714-722, August 1985.

[2] M. L. Crow, "Waveform relaxation methods for the simulation of systems of differential/algebraic equations with power system transient stability applications", *Ph.D. dissertation*, University of Illinois, Urbana, IL, 1990.

[3] M. L. Crow and M. Ilić, "Effects of system decomposition and waveform relaxation windowing of the efficiency of large scale power systems simulation," *Proceedings of the 12th IMACS World Congress on Scientific Computation*, Paris, France, July 1988.

[4] M. L. Crow and M. Ilić, "The parallel implementation of the waveform relaxation method for transient stability simulation," *IEEE Transactions on Power Systems*, paper # 90 WM 240-2 PWRS, to appear.

[5] M. L. Crow, M. Ilić, J. White, "Convergence properties of the waveform relaxation method as applied to electric power systems", *Proceedings of the 1989 IEEE Int'l Symp. on Circuits and Systems*, Portland, OR, May 1989.

[6] M. Ilić-Spong, M. L. Crow, M. A. Pai, "Transient Stability Simulation by Waveform Relaxation Methods," *IEEE Trans. of Power Systems*, vol. PWRS-2, no. 4, Nov. 1987.

[7] E. Lelarasmee, "The waveform relaxation method for the time domain analysis of large scale nonlinear dynamical systems," *Ph.D. dissertation*, University of California, Berkeley, California, 1982.

[8] U. Miekkala, and O. Nevanlinna, "Sets of convergence and stability regions," Report-MAT-A239, Institute of Mathematics, Helsinki University of Technology, Helsinki, Finland, 1986.

[9] M. E. Mokari-Bolhassan, D. Smart and T. N. Trick, "A new robust relaxation technique for VLSI circuit simulation," *Proc. of the IEEE Int'l Conference on Computer Aided Design*, Santa Clara, CA, pp. 26-28, Nov. 1985.

[10] J. K. White, A. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits*, Kluwer Academic Publishers, Boston, MA, 1987.

[11] J. Zaborsky, G. Huang, K. W. Lu, "A textured model for computationally efficient reactive power control and management", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-104, no. 7, pp. 1718-1727, July 1985.

PANEL TOPIC: "What Neural Network Chips Are Worth Building?"

PANEL MEMBERS	ASSOCIATION
Les Atlas	University of Washington
Mark Holler	Intel
Lawrence Jackel	AT&T Bell Labs
Paul Mueller	University of Pennsylvania
Andrew Penz	Texas Instruments, Chairman
John Wyatt	MIT

The panel will address the issue of neural network chips within the context of possible marriages to algorithms that are likely to produce practical capabilities not currently obtainable. Figure 1 is an attempt to set the stage for this analysis. It shows a qualitative view of current chip technology difficulty and algorithm utilization in simulation as a function of analog content. For instance, backpropagation is very popular in simulation and requires extensive analog capability, putting it in the top right hand corner of the figure. On the other hand sparse distributed memory [SDM] algorithms are highly binary in flavor but are seldom used by the NN community. The tendency of algorithm utilization, therefore is from the lower left to the upper right of the diagram.

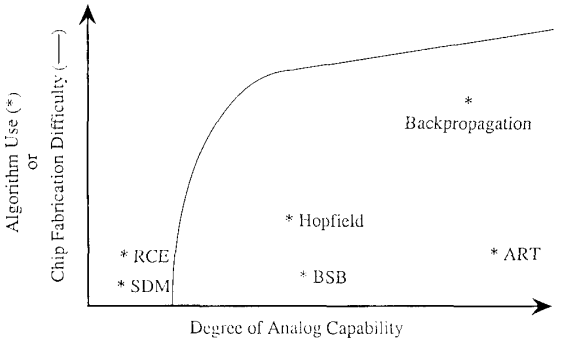

Figure 1. Chip complexity and NN algorithm popularity vs. analog capability.

The difficulty of fabrication analogy semiconductors is plotted as a curve which follows the algorithm usage trend as a function of analog capability. VLSI production is very heavily weighted in favor of binary technology, i.e., binary hardware is easy to build. Analog devices are difficult to build. The more popular the algorithm, the more difficult it is to make a chip. The figure indicates the dilemma which faces the NN community: VLSI technology does not match well to most popular NN algorithms. But it is broadly assumed that semiconductor technology must be brought into execution of NN algorithms, as opposed to simulation of these algorithms, for the current NN enthusiasm to result in the creation of a new computational methodology. "Neurons" must be very cheap in order to afford large numbers, and only binary semiconductor methods currently offer a hope of truly inexpensive artificial neurons. The panel will discuss this dilemma and hopefully enlighten themselves and the audience on ways to effect chips that are "worth building".