1

Event-by-event simulation of quantum phenomena

H. De Raedt[1], K. De Raedt[2], and K. Michielsen[1]

[1]Applied Physics - Computational Physics, Materials Science Centre, University of Groningen, Nijenborgh 4, NL-9747 AG,Groningen, The Netherlands
[2]Department of Computer Science, University of Groningen, Blauwborgje 3, NL-9747 AC Groningen, The Netherlands

Summary. In various basic experiments in quantum physics, observations are recorded event-by-event. The final outcome of such experiments can be computed according to the rules of quantum theory but quantum theory does not describe single events. In this paper, we describe a stimulation approach that does not rely on concepts of quantum theory but nevertheless generates events with frequencies that agree with quantum theory. In particular, we demonstrate that locally connected networks of processing units that have primitive learning capabilities can be used to perform a deterministic, event-based simulation of single-photon Mach-Zehnder interferometer experiments.

1.1 Introduction

Recent advances in nanotechnology, make it possible to control individual ions, atoms, photons and the like. These technological developments facilitate the study of single quantum systems at the level of individual events [1, 2]. Such experiments address the most fundamental aspects of quantum theory. Quantum theory gives a recipe to compute the frequencies for observing events but it does not describe individual events, such as the arrival of a single electron at a particular position on the detection screen [2–5]. Reconciling the mathematical formalism, not describing single events, with the experimental fact that each observation yields a definite outcome is often referred to as the quantum measurement paradox [3, 4].

Computer simulation is a powerful methodology, complementary to theory and experiment, to model physical phenomena [6]. In this approach, we start from the basic equations of physics and use numerical algorithms to solve these equations. But what if the basic equation to describe the individual events is unknown?

From a computational viewpoint, quantum theory provides a set of rules to compute probability distributions [3, 7], that is to compute the final, collective outcome of an experiment in which single events are recorded. However, quantum theory does not provide algorithms to perform an event-based simulation of such experiments. Methods based on the solution of the (time-dependent) Schrödinger equation are inappropriate for this purpose. Hence, a completely new computer simulation method is required to simulate the event-based observations in quantum physics experiments.

Elsewhere, we have already demonstrated that locally-connected networks of processing units with a primitive learning capability can generate events at a rate that agrees with the quantum mechanical probability distribution [8–11]. In this paper we discuss the basic elements of our approach and, as an illustration, we present deterministic event-based simulation results of single-photon Mach-Zehnder interferometer experiments.

1.2 Deterministic learning machines (DLMs)

In quantum physics, an event corresponds to the detection of a photon, electron or the like. In our deterministic, event-based simulation approach an event is the arrival of a message at the input port of a processing unit. This processing unit typically contains three components: an input DLM, a transformation unit and an output DLM. A DLM is a classical (but non-Hamiltonian), deterministic, local and causal dynamical system with a primitive learning capability. A DLM consists of input and output ports, an internal unit vector, a rule that specifies how this vector changes when an input event is received, and a rule by which the DLM determines the type of output event it generates as a response to the input event.

For simplicity, we assume that the input DLM can accept $L = 2$ types of events, but only one at a time. We label the events by 0 and 1. Event 0 carries a message represented by a unit vector $\mathbf{y}_0 = (y_0, y_1)$ containing two real numbers. Similarly, event 1 carries a message represented by $\mathbf{y}_1 = (y_2, y_3)$. In this particular case the internal vectors of the input and output DLM have length four and can be represented by $\mathbf{x} = (x_0, x_1, x_2, x_3)$, where x_i, $i = 0, \ldots, 3$ are real numbers. In general the length of the internal vector is 2×2^L. The initial value of the internal vector is irrelevant and can be chosen at random.

The learning algorithm of the input DLM, DLMi, is defined as follows:

- DLMi constructs a vector $\hat{\mathbf{x}}$ of length four using information from the incoming event and from its own internal vector $\mathbf{x}$. If the DLM receives a 0 event then $\hat{\mathbf{x}} = (y_0, y_1, x_2, x_3)$. If it receives a 1 event then $\hat{\mathbf{x}} = (x_0, x_1, y_2, y_3)$.
- Based on its own internal vector DLMi computes eight candidate internal vectors containing the elements

$$w_{j,j} = \pm\sqrt{1 + \alpha^2(x_j^2 - 1)}, \quad w_{j,i} = \alpha x_i, \quad \text{if} \quad i \neq j, \tag{1.1}$$

where $i, j = 0, \ldots, 3$ and i is the first running index. The parameter $0 < \alpha < 1$ controls the learning process.

- DLMi chooses from the eight candidate internal vectors $\mathbf{w}$ the one that minimizes the cost function $C = -\mathbf{w}.\hat{\mathbf{x}}$ and replaces its internal vector $\mathbf{x}$ by this vector.
- DLMi puts the four elements of its internal vector on its four output ports and waits for the next event to be processed.
- The four output ports of DLMi are connected to the four input ports of the transformation unit. The transformation unit applies an orthogonal transformation to the vector $\mathbf{x} = (x_0, x_1, x_2, x_3)$ it receives from DLMi. The precise form of the transformation depends on the particular function that the processor has to perform. An example is given in section 1.4. The result of the transformation is the vector $\mathbf{x}' = (x'_0, x'_1, x'_2, x'_3)$.

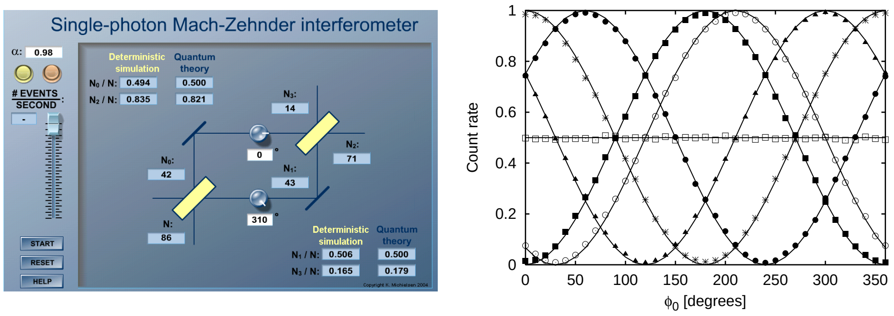
The learning algorithm of the output DLM, DLMo, is defined as follows:

- DLMo takes the vector $\mathbf{x}' = (x'_0, x'_1, x'_2, x'_3)$ as input.
- Just as DLMi, DLMo computes, based on its internal vector eight, candidate internal vectors according to Eq.(1.1).
- DLMo chooses from the eight candidate internal vectors $\mathbf{w}$ the one that minimizes the cost function $C = -\mathbf{w}.\mathbf{x}'$ and replaces its internal vector $\mathbf{x}$ by this vector.
- DLMo generates an output event of type 0 (1) carrying as a message the first (last) two elements of its new internal vector, if an update rule with $j = 0, 1$ ($j = 2, 3$) was chosen.
- DLMo waits for the next event to be processed.

1.3 A DLM is an efficient encoder

We consider as a special case of the DLM described in section 1.2 a DLM which accepts one input event carrying the message $\mathbf{y} = (\cos\phi, \sin\phi)$. In this case, the internal vector $\mathbf{x}$ of the DLM has length two and there are four candidate update internal vectors, given by Eq.(1.1) with $i, j = 0, 1$. The DLM chooses from the four candidate internal vectors $\mathbf{w}$ the one that minimizes the cost function $C = -\mathbf{w}.\mathbf{y}$ and replaces its internal vector $\mathbf{x}$ by this vector. The DLM sends out a 0 (1) event carrying as a message the first (second) element of its new internal vector if an update rule with $j = 0$ ($j = 1$) was chosen. If we count the number of 1 events sent out by the DLM, which we denote by K, and divide by the total number of processed events N, we find $K/N \approx \cos^2\phi$. The DLM generates a fully deterministic sequence of zeroes and ones, that is, the most compact sequence for each K and N, with minimum variance on $K/N \approx \cos^2\phi$. The performance is optimal: the number of distinguishable messages is equal to $N + 1$.

Apparently, the most efficient deterministic encoder (encoding an angle ϕ as a sequence of zeroes and ones) that we can build generates data according



α: 0.98

	Deterministic simulation	Quantum theory
N₀ / N:	0.494	0.500
N₂ / N:	0.835	0.821

N₃: 14
N₂: 71
N₀: 42
N₁: 43
N: 86
0 °
310 °

	Deterministic simulation	Quantum theory
N₁ / N:	0.506	0.500
N₃ / N:	0.165	0.179

Fig. 1.1. Left: Screen dump of an interactive program that performs event-based simulations of a single-photon Mach-Zehnder interferometer [12]. The number of events N_i in channel $i = 0, \ldots, 3$ corresponds to the probability for finding a photon on the corresponding arm of the interferometer. Right: Simulation results for the DLM-network shown on the left. Input channel 0 receives $(\cos\psi_0, \sin\psi_0)$ with probability one. A uniform random number in the range $[0, 360]$ is used to choose the angle ψ_0. Input channel 1 receives no events. Each data point represents 10000 events $(N_0 + N_1 = N_2 + N_3 = 10000)$. Initially the rotation angle $\phi_0 = 0$ and after each set of 10000 events, ϕ_0 is increased by 10°. Markers give the simulation results for the normalized intensities as a function of ϕ_0. Open squares: $N_0/(N_0+N_1)$; Solid squares: $N_2/(N_2+N_3)$ for $\phi_1 = 0$; Open circles: $N_2/(N_2+N_3)$ for $\phi_1 = 30°$; Bullets: $N_2/(N_2+N_3)$ for $\phi_1 = 240°$; Asterisks: $N_3/(N_2+N_3)$ for $\phi_1 = 0$; Solid triangles: $N_3/(N_2+N_3)$ for $\phi_1 = 300°$. Lines represent the results of quantum theory.

to Malus's law: $I = I_0 \cos^2 \phi$, where ϕ is the angle between the polarization direction of the light and the transmission axis of the polarizer. Note that we designed the encoder by making use of geometric rules and not by using laws of physics. The $\cos^2 \phi$ law is the result of finding the optimal encoder.

1.4 Single-photon Mach-Zehnder interferometer

We consider a DLM network (see Fig. 1.1) that generates the same interference patterns as those observed in single-photon Mach-Zehnder interferometers [1]. There is a one-to-one correspondence between the processing units in the network and the physical parts of the experimental setup. For the beam splitters we use a network consisting of an input and output DLM as described in section 1.2. To implement the operation of a beam splitter, we use in the transformation unit the transformation $(x_0, x_1, x_2, x_3) \to 1/\sqrt{2}(x_0 - x_3, x_2 + x_1, x_2 - x_1, x_0 + x_3)$. The phase shift is taken care of by two passive devices that perform plane rotations by ϕ_0 and ϕ_1, respectively. According to quantum theory, in the case that input channel 1 receives no input events, the amplitudes (b_0, b_1) of the photons in the output channels N_2 and N_3 of the Mach-Zehnder interferometer are given by

$$b_0 = e^{i\gamma}\sin((\phi_0 - \phi_1)/2) \quad , \quad b_1 = e^{i\gamma}\cos((\phi_0 - \phi_1)/2), \tag{1.2}$$

where γ is an irrelevant phase. From Eq.(1.2), it follows that the probabilities $|b_0|^2$ and $|b_1|^2$ depend on $\phi = \phi_0 - \phi_1$ only. In Fig. 1.1 we present a representative selection of simulation results for the Mach-Zehnder interferometer built from DLMs. It is clear that our event-by-event simulation approach generates events according to the wave mechanical distribution Eq.(1.2).

1.5 Discussion

We have described the basic elements of a procedure to construct algorithms that can be used to simulate quantum processes without solving the Schrödinger equation. We have shown that single-particle quantum interference can be simulated on an event-by-event basis using local and causal processes, without using concepts such as wave fields or particle-wave duality. Elsewhere, we show that the same procedure can be used to perform a deterministic, event-based simulation of universal quantum computation [11, 13]. These results suggest that we may have discovered algorithms that do not rely on concepts of quantum theory, but are able to simulate quantum phenomena using classical, causal, local, and event-based processes

References

1. P. Grangier, R. Roger, and A. Aspect, Europhys. Lett. 1, 173 (1986).
2. A. Tonomura, *The Quantum World Unveiled by Electron Waves*, World Scientific, Singapore (1998).
3. D. Home, *Conceptual Foundations of Quantum Physics*, Plenum Press, New York (1997).
4. R.P. Feynman, R.B. Leighton, M. Sands, *The Feynman lectures on Physics*, Vol. 3, Addison-Wesley, Reading MA, (1996).
5. L.E. Ballentine, *Quantum Mechanics: A Modern Development*, World Scientific, Singapore (2003).
6. D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulation in Statistical Physics*, Cambridge University Press, Cambridge, (2000).
7. N.G. Van Kampen, Physica A 153, 97 (1988).
8. K. De Raedt, H. De Raedt, and K. Michielsen, arXiv: quant-ph/0409213
9. H. De Raedt, K. De Raedt, and K. Michielsen, Europhys. Lett. 69, 861 (2005).
10. H. De Raedt, K. De Raedt, and K. Michielsen, J. Phys. Soc. Jpn. (in press).
11. K. Michielsen, K. De Raedt, and H. De Raedt, J. Comp. Theor. Nanoscience (in press).
12. For additional information visit http://www.compphys.net/dlm
13. K. Michielsen, K. De Raedt, and H. De Raedt, *Deterministic event-based simulation of universal quantum computation*, in Computer Simulation Studies in Condensed-Matter Physics XVIII, eds. D.P. Landau et al., Springer Proceedings in Physics, Volume ?? (Springer, Berlin), ?? (200?)