Proceedings of the 1997 Winter Simulation Conference
ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson

METHODOLOGY FOR THE INCREASED COMPUTATIONAL EFFICIENCY OF DISCRETE-EVENT SIMULATION IN 3 DIMENSIONAL SPACE

Eugene P. Paulo
Linda C. Malone

Department of Industrial Engineering and Management Systems
in the College of Engineering at the
University of Central Florida
Orlando, Florida 32816, U.S.A.

ABSTRACT

Users of simulation continue to demand more realism and accuracy. This has been addressed by the development of new simulation languages, better simulation software, more user-friendly interface, and more advanced computers. Despite this advance in general simulation capability, only recently has significant research addressed the efficiency of the simulation of the motion of autonomous spatial objects in discrete simulation. The overall goal of this paper is to provide an improvement in the efficiency of a simulation of moving autonomous objects by the use of various forms of dynamic sectoring. Research has shown that no single discrete event simulation methodology has combined the movement and tracking of autonomous objects in three-dimensional space with computational efficiency. The focus of this paper is to provide a summary of previous sectoring research, to consider 3D, to propose a new sectoring approach, and to analytically show why the new approach should provide an improvement.

1 INTRODUCTION

Users of simulation continue to demand more realism and accuracy. This has been addressed by the development of new simulation languages, better simulation software, more user-friendly interface, and more advanced computers. Despite this advance in general simulation capability, only recently has significant research addressed the efficiency of the simulation of the motion of autonomous spatial objects in discrete simulation.

The autonomous movement of spatial and temporal objects has not traditionally been a concern of discrete simulation, which generally focuses on areas in which spatial and temporal precision is not a priority (such as queueing and manufacturing problems) or else simulates movement through the use of a network with specific start and finish nodes. Yet spatial and temporal precision are necessary in many simulations. Examples of these systems are military combat and the air traffic patterns around any airport, as well as in the medical field, where movement of fluids, such as blood or serum, through the body must be monitored.

The overall goal of this paper is to provide an improvement in the efficiency of a simulation of moving autonomous objects by the use of various forms of dynamic sectoring. Research has shown that no single discrete event simulation methodology has combined the movement and tracking of autonomous objects in three-dimensional space with computational efficiency. The focus of this paper is to provide a summary of previous sectoring research, to propose a new sectoring approach to consider 3D, and to analytically show why the new approach should provide an improvement.

2 SECTORING OVERVIEW

Recent research into simulation efficiency has found merit in the concept of sectoring. For spatial objects to move autonomously in a simulation, the location vector of all other objects within the simulation boundaries must be tracked. This has routinely meant that scheduled events are the result of each object querying every other object in the simulation. This basic technique is computationally inefficient when the number of objects gets large. The idea of sectoring allows the simulation space to be split into segments that an object is required to query only other objects that are in its sector or on the boundary of adjacent sectors. While sectoring reduces the amount of queries between objects, the crossing of sector boundaries becomes an additional event to track. Therefore, in order for sectoring to be beneficial, the reduction in overall queries between objects must overcome the additional computational load of tracking the objects as they cross sector boundaries.

Research into sectoring has resulted in several important discoveries. First, the plausibility of two particular sectoring methods, known as fixed sectoring and dynamic sectoring, has been demonstrated in the two-dimensional billiard balls problem. Fixed sectoring provides a fixed or stable amount of identical sized sectors throughout the entire simulation area for the entire duration of the simulation. Dynamic sectoring allows the size of the sectors to change depending on the number of objects currently within a sector, attempting to efficiently account for the contrast in number of objects to query versus the necessity of accounting for the crossing of sector boundaries. Dynamic sectoring is thought to be most effective when the objects are not randomly or uniformly distributed throughout the simulation area.

3 RECENT SECTORING RESEARCH

Several individuals have recently performed research in the area of discrete simulation and spatial object control. Contributions include using different languages and operating systems, sectoring techniques, and improved representation of moving objects (billiard balls, pucks, or polygons) within the simulation. Each of these researchers used a simulation that portrayed objects and their movement in two dimensions. More in depth explanation of these contributions begins with Harless [1995].

Harless built on the billiard ball problem first simulated by Goldberg, and then refined by Lubachevsky, Rogers, and Toleti. Harless considered the issue of non-uniformly distributed autonomous objects and interjected the concept of dynamic sectoring. While fixed sectoring had previously been demonstrated on an arbitrary basis to provide improved efficiency in simulating the movement and collisions of spatial objects, no complete effort had been given to the investigation of non-uniformly distributed objects. Harless used dynamic sectoring, so that sectors were subdivided whenever a sector contained “too many” spatial objects and were rejoined whenever the sector contained “too few” spatial objects. Thresholds for determining when to subdivide were determined with some preliminary testing. Using billiard ball simulation, Harless demonstrated that the dynamic sectoring methodology can provide smaller mean execution time for non-uniform distribution of spatial objects than the fixed sectoring methodology [Harless, 1995].

Using the object-oriented simulation developed by Harless for both fixed and dynamic sectoring, Doescher attempted to validate the work of previous sectoring researchers by establishing analytically that fixed sectoring, with uniformly distributed objects, is more efficient than no sectoring, and to investigate which factors affect optimal sectoring. Specifically, previous researchers had shown in isolated cases that sectoring was an improvement over no sectoring, but had not investigated statistical steps to demonstrate the relationships between sectoring, number of objects, speed of objects, size of objects, and resulting efficiency. While Doescher was not able to determine which factors clearly affect optimal sectoring , he demonstrated improved simulation efficiency using sectoring for certain ranges of number of objects and object size, which he labeled density.

4 NEW SECTORING APPROACH

The goal of this paper is to consider a modified dynamic sectoring approach and to demonstrate that dynamic sectoring provides increased efficiency in discrete event simulation methodology for the movement of autonomous spatial objects in *three* dimensions. Previous research focused on a two-dimensional simulation region, with the movement of objects being like those of billiard balls on a table. Three dimensional space seems intuitively to be more realistic and representative of the physical systems that are modeled.

Until now, the sector shapes used in both fixed and dynamic sectoring have been square, and the distribution of objects has been either uniform or non-uniform (meaning that clusters of objects occur). It is quite possible that there are better forms of sectoring that will provide greater computational savings. For example, certain combinations of object distribution and object density may be more responsive to some modified version of fixed or dynamic sectoring with a given number of sectors. In three dimensions, while cubic sectors are the simplest to conceptualize in a simulation, the overall shape and distribution of objects in the simulation may make other shapes, such as concentric spheres, more efficient. This is investigated more fully in the analytical model described below.

5 METHODOLOGIES FOR 3-D SECTORING ANALYTICAL MODELS

5.1 Overview

In order to conduct a comparison of various sectoring techniques, an analytical way must be formulated to measure the necessary effort. A standard of measure that has traditionally been used for computational complexity is known as “order of work”, or often “big-oh” notation, and is denoted in the form of **O(N)** [Aho, 1987]. This is a measure in general terms of the scale of computational complexity, and means that as N increases, the simulation running time increases at a rate of N.

An example of how this measuring tool is used is in the simulation of autonomous objects without sectoring. This methodology is described in the general terms of order of work by $\mathbf{O(N^2)}$, meaning that the complexity of the computations is on the scale of the number of objects in the simulation squared [Harless, 1995]. This approximation provides anticipated efficiency, as well as identification of certain conditions (e.g. type of sectoring, object distribution, number of sectors) that may demonstrate sectoring benefits.

The two-stage process used to analytically determine computational complexity is described below. The general process is valid for all cases considered, and is therefore applied to each methodology. An equation and example of the computational complexity for each specific sectoring methodology is provided.

5.2 Two-stage process for model development

The analytical model is representative of a simulation that is in steady state. Thus, a master event list has been created which portrays all currently scheduled events in sequence. The computational work reflects the two-stage process of: 1) determining the next event for two objects that have just collided, and 2) updating the master event list. While there are two events to consider, namely collisions between two objects and sector crossings, we will ignore sector crossings.

For the purpose of deriving an analytical model, the current event is assumed to be a collision. Since collisions involve two objects and are therefore a more computationally demanding event, collisions represent a worst case scenario. As is noted, other assumptions are made that allow for a worst case scenario, since this both narrows the problem and provides insight into the upper boundary of computational complexity.

For stage one, let us assume that the next event to occur is a collision. Once that event has occurred, the next event for each of the two objects involved in the collision must be determined. If the methodology used is a form of sectoring, the model compares trajectories of these two objects with all other objects in the current sector and adjacent sectors. All adjacent sectors are checked due to the possibility that an object may be at or near the sector boundary, and a boundary collision may occur. If no sectoring is used, the model must check for possible collisions of these two objects with all other objects in the simulation. The result will be the next scheduled events for the two objects involved in the previous event. These two new events may invalidate previously scheduled events, thus having a ripple effect.

The second stage is then to compare the time to the events in the master event list. The model searches for the proper place in the event sequence for the two new events and places them there. As the event list could have a length equivalent to N/2, assuming every object is scheduled to collide with another object, the search for the two new events is approximately 2 times N/2, or N.

This entire process repeats itself until the event list is updated for every object, which can be considered a ripple effect. Specifically, this means that the objects that collided with these two objects must also have new events calculated. This cycle of continuously updating the event list, as a worst case, could occur for every object, which leads to the equations described below. The result is that the second stage of the process dominates resources in the sectoring problem.

5.3 No Sectoring Methodology

Assuming N objects in the simulation, the work associated with determining the next event for the two objects involved in the latest collision is approximately 2N. This comes from the requirement of each of the two objects to check their new trajectories with those of every other object in the simulation for a possible new collision. To update the master list for each new collision, the upper limit of the amount of work is approximately N, as described above. Assuming that this cycle must occur for every pair of objects in the simulation, the estimated number of possible total work for no sectoring (W_{No}) for the total simulation region is

$$W_{No} = \frac{N}{2}(2N + N) = \frac{3N^2}{2} \quad (1)$$

5.4 Fixed Sectoring Methodology, Uniform or Random Object Distribution

Calculating order of work for fixed cubic sectoring is somewhat more complicated. This methodology divides the trajectory space into **k** cubic sectors of equal dimension. Spatial objects are then distributed uniformly or randomly throughout the simulation region, with the expected number of objects in each sector to be **N/k**, or **n**.

After a collision, the determination of the next event follows the two-stage process described above. Basically, the velocity vectors of the two objects that just collided in a particular sector are examined and compared to the vectors of the other objects in that sector and adjacent sectors, and new events for these objects are scheduled. The amount of work necessary to complete these comparisons is bounded by

$$2n + 2(n)(\mathbf{AS}) = 2n(1+\mathbf{AS}) \quad (2)$$

where **AS** is the number of adjacent sectors. These two new events are inserted in the master event list, which triggers the cycle of determining a new schedule of events.

To determine **AS**, consider a two-dimensional trajectory space as in the figure below. The maximum number of adjacent sectors is eight (which occurs when the sector under consideration is not adjacent to the simulation boundary). In three-dimensional space, the maximum number of adjacent sectors is 26 (assuming a center cubic sector), while the minimum number of adjacent sectors is 7 for corner sectors. Since the number of adjacent sectors varies, it will be designated as the variable **AS**.

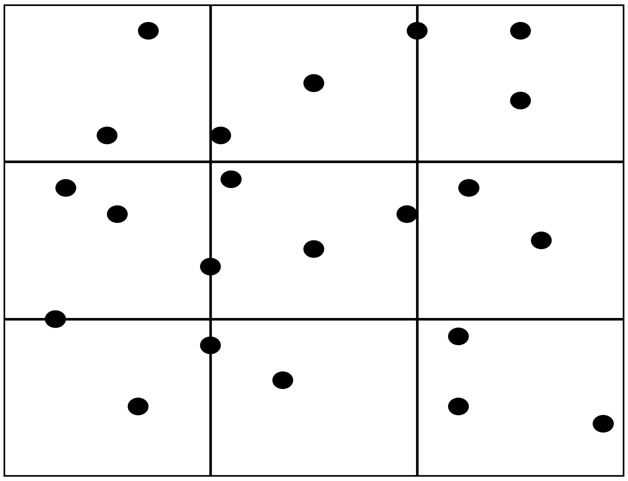

Figure 1: Fixed Sectoring

The remainder of the equation for calculating the amount of work for fixed sectoring is the same as for no sectoring. Checking the entire master event list in order to place the new events in their time sequence could result in work approximating N. Repeating this cycle for each new set of events resulting from this ripple effect results in the following equation, as an upper bound for fixed sectoring, uniform distribution:

$$W_{F,U} = \left(\frac{N}{2}\right)\left[2n(1+AS)+N\right] \quad (3)$$

5.5 Dynamic Sectoring (Cubic Sectors) Methodology

Dynamic sectoring with cubic sectors requires a slightly different approach. Its usefulness is apparent when objects are no longer distributed uniformly throughout the simulation region, but rather are clustered in certain areas. A pair of thresholds are introduced into the simulation, including the upper limit of objects allowed in the sector before that sector sub-divides into smaller square sectors, and the lower limit allowed in sub-divided sectors before the smaller sectors consolidate into their previous larger sector.

Again, the graphical representation of a two dimensional model can provide some clarity. As mentioned earlier, dynamic sectoring allows for sectors to divide further into smaller sectors when the number of objects in the sector reach a certain threshold. In the figure below, any sector having more than three objects will sub-divide into smaller square sectors such that none of the smaller sectors have more than three objects.

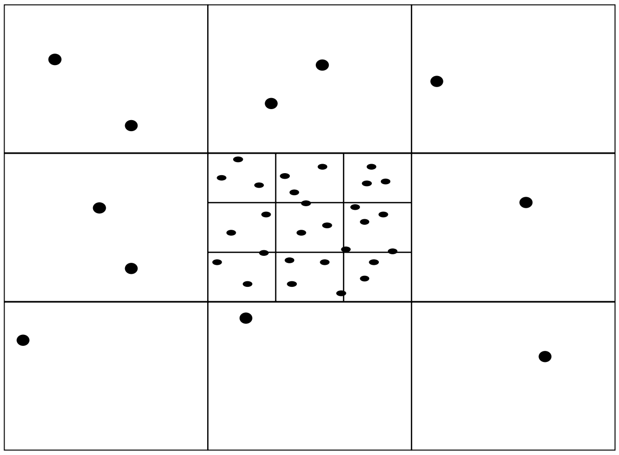

Figure 2: Dynamic Sectoring (Squares)

The calculation of order of work for dynamic sectoring with cubic sectors is similar to fixed sectoring, requiring several small adjustments. First, the average, or expected, number of objects per sector (**n**) is now approximated by the threshold value, which can be called **t**. Additionally, the number of sectors (**k**) becomes a variable **s**, since the number of total sectors changes throughout the simulation as sectors are sub-divided and then consolidated. Finally, the upper bound of the number of possible adjacent sectors, **AS**, changes to infinity. This is the result of a situation where there is continuous subdivision of sectors in one region, while bordering this region are sectors that retain their original size. The following upper bound is therefore used to determine the total work using dynamic sectoring with cubic sectors ($W_{D,C}$):

$$W_{D,C} = \frac{N}{2}\left[2t(1+AS)+N\right] \quad (4)$$

with **AS** having infinity as an upper bound.

5.6 Dynamic Sectoring (Concentric Spheres)

The notion of using concentric spheres may be particularly helpful for certain types of object distribution. Rather than just the occurrence of clusters, where certain areas have a noticeably higher concentration of objects than other areas, most or all of the objects move toward and remain near a particular region of the simulation space. This region, or point in space, is called an attractor, and results in a single concentration of objects in the simulation region. (A practical application of this might be an airport). A two-dimensional depiction of this phenomenon is seen in the figure below.

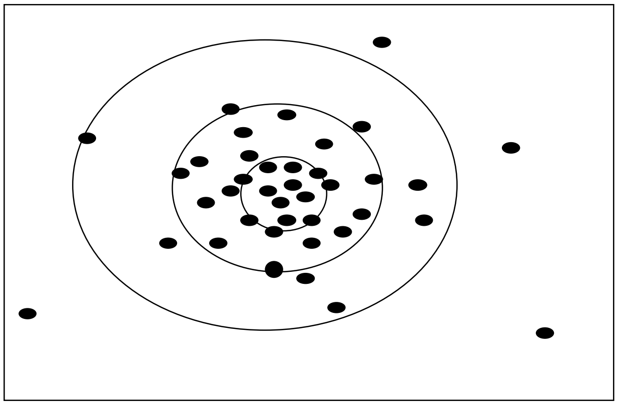

Figure 3: Dynamic Sectoring, Concentric Circles

Since this is a form of dynamic sectoring, the number of sectors varies according to a designated threshold, or **t,** which is the maximum number of objects allowed in the spherical sector before an additional spherical sector is created. The computation of possible comparisons is basically the same as for the previous case, with one exception. The number of adjacent sectors (**AS**) can assume only values of 0,1, or 2 (with 0 implying no sectoring), since the concentric nature of the sectors means that the outer-most and inner-most sectors have one adjacent sector, while the remaining sectors have two adjacent sectors. Thus, letting **AS** equal 2, the upper bound for the total possible work using dynamic sectoring with concentric spheres ($W_{D,S}$) is

$$W_{D,S} = \frac{N}{2}[6t + N] \quad (5)$$

6 CASE STUDIES USING THE ANALYTICAL METHODOLOGIES

6.1 Uniform Distribution

One way to conduct analytical modeling of sectoring in a 3 dimensional simulation region is to address the different object distributions separately, concentrating first on uniform distribution. The two methodologies examined are no sectoring and fixed sectoring. Dynamic sectoring with cubes is not considered because it is the same as fixed sectoring when objects are distributed uniformly or randomly.

For example, choose a 3-dimensional simulation with 64 cubic sectors (**k**), and a total number of 512 objects (**N**). The average number of objects per sector (**n**) is therefore equal to 8, and the threshold for sector sub-division is also 8. Only 8 of the 64 are not at least a part of one boundary, therefore just those eight have 26 adjacent cubic sectors. The remaining 56 have between 7 and 17 adjacent sectors, depending on whether the sector being considered forms a corner of the simulation space or merely part of the simulation boundary plane. The average number of adjacent sectors (**AS**) is 14.625. The number of possible comparisons to find the next events after a collision and to update the event list is 195,051.

Similarly, with the same number of objects, but with the number of cubic sectors (**k**) equal to 27, we get a slightly different numbers of possible comparisons. The average number of objects per sector (**n**) is now 18.96. The average number of adjacent sectors decreases to 11.704, since only the center sector has a full 26 cubic adjacent sectors, while the remaining sectors have between 7 and 17. The resulting number of possible comparisons for fixed sectoring is now 254,357.

Finally, computing the number of possible comparisons using 125 cubic sectors and contrasting these three figures with no sectoring reveal the following:

- No sectoring: 393,216
- Fixed Sectoring (27 sectors) 254,357
- Fixed Sectoring (64 sectors) 195,051
- Fixed Sectoring (125 sectors) 168,396

While these numbers show significant savings in computational requirements as the number of sectors increases, they do not show the significance of sector crossings. Intuitively, sector crossings become a more important factor in the total work when there are a greater number of sectors in the simulation, and although they may detract from the apparent improvement from larger numbers of sectors, it is not expected that they will significantly influence results. These numerical results could provide insight for obtaining a starting point for simulation runs.

6.2 Clustering

Consider the same case described above, except that instead of uniform distribution of objects, there is a cluster of objects in the simulation region. In this situation, the two methodologies that may provide improvement over no sectoring are fixed sectoring and dynamic sectoring with cubic sectors. The number of possible comparisons for the no sectoring case remains the same. To more clearly illustrate the difference between the two sectoring methodologies, assume that the cluster contains all the objects in the simulation space and that the fixed sectors (and initial sectors in the dynamic sectoring case) are sized such that all clustered objects reside within one sector.

If the threshold value, **t**, is set equal to **n,** then the following approximates the number of possible combinations for the two methodologies:

- Fixed sectoring (**n = 19**) 393,216
- Dynamic sectoring (**t = 19**) 254,357
- Fixed sectoring (**n = 8**) 393,216
- Dynamic sectoring (**t = 8**) 195,051

These numbers show that fixed sectoring can result in virtually a no sectoring approach, since all objects reside within one sector. Additionally, in this scenario, the dynamic sectoring method has similar results to fixed sectoring in uniform distribution, as the sector containing the cluster can sub-divide until the threshold value is met. Again, while not providing a thorough understanding of what to expect in the simulation scenario, this comparison allows for some guidance in how to set the initial simulation parameters.

6.3 Attractor

Consider the case where the objects are moving generally toward a single attractor in the simulation region. While the objects may not appear as a large congested cluster, they nevertheless may form an approximation of a multivariate normal distribution around the attractor, such that objects continue moving in the direction of the attractor, and objects are closer to each other when nearer to the attractor. In this situation, the two methodologies that may provide improvement over no sectoring are dynamic sectoring with cubic sectors and dynamic sectoring with concentric spheres. The number of possible collisions for the no sectoring case remains the same.

Assume that the location of the attractor is known and it is chosen as the center of the concentric spheres. Also recall that the average number of adjacent sectors, **AS**, hinges on the fact that the inner concentric spheres have two adjacent sectors, while the outermost and innermost concentric spheres have only one. This will result in an AS close to 2, with the actual value depending on the sector containing the next event.

If the threshold value, **t**, is set at 8 for one iteration and at 19 for another, then the following approximates the number of possible comparisons for the two methodologies:

- Dynamic sectoring, cubic sectors (**t = 19**) 254,357
- Dynamic sectoring, spherical sectors (**t = 19, AS = 2**) 160,256
- Dynamic sectoring, cubic sectors (**t = 8**) 195,051
- Dynamic sectoring, spherical sectors (**t = 8, AS = 2**) 143,360

This comparison shows the potential for improvement in simulation time using concentric spheres.

7 FINAL INSIGHTS

The above analysis provides some insight into the effects of various sectoring methodologies and object distribution on the number of possible comparisons and subsequent order of work. While some mention has already been made regarding overhead, or sector crossings, that are not part of the analytical computations above, there are also other factors that may affect computational speed. This is evident from the results of Doescher and Harless.

At no time in Harless' simulations was dynamic sectoring more than three times faster than fixed sectoring in a similar object environment. Additionally, it is apparent that the relatively small number of sector crossings in that best case could not have accounted for such a closing of the gap between the two methodologies.

Doescher focused strictly on fixed sectoring. His research showed that in no circumstance was fixed sectoring able to reduce computational speed over no sectoring by more than 52%. Again, the number of sector crossings was not sufficient to explain these results. However, there are at least two possibilities.

The first is what Doescher called object density. This is the product of the number of objects and the size of objects in the simulation. Doescher showed there was some correlation between the object density and the efficiency of fixed sectoring, with greater object density resulting in more significant improvement for fixed sectoring versus no sectoring.

Additionally, the data structures, such as simple arrays or link lists, used in the program could affect how efficiently the simulation progresses. How well the sectors are structured, identified, and how objects are tracked and assigned to sectors influence the efficiency of the simulation.

Finally, there are undoubtedly other factors that have not been considered. The next step in trying to improve the simulation efficiency would appear to be an empirical study to try to gather data to explain the gap between the theory and the practice.

REFERENCES

Aho, A.V., Hopcroft, J.E., and Ullman, J.D., 1987, *Data Structures and Algorithms*, Reading, Massachusetts: Addison-Wesley Publishing Company.

Alder B.J., and Wainwright, T.E. "Studies in Molecular Dynamics. I. General Method." *Journal of Chemical Physics*, 31 (1959): 459-466.

Doescher, Craig, 1996. "A Statistical Verification of the Improvement in Computational Efficiency of Uniformly Distributed Spatial Objects Using Fixed Sectoring Simulation," Master's thesis, Department of Industrial Engineering and Management Sciences, University of Central Florida, Orlando, 1996.

Goldberg, A.P. "Object-Oriented Simulation of Pool Ball Motion." Master's thesis, University of California, Los Angeles, 1984.

Harless, Gary J. “Improving Computational Efficiency in the Discrete Event Simulation of Non-Uniformly Distributed Autonomous Spatial Objects.” Ph.D. dissertation, Department of Industrial Engineering and Management Sciences, University of Central Florida, Orlando, 1995.

Katzenelson, J. “Computational Structure of the N-Body Problem.” *SIAM Journal of Science and Statistical Computing*, 10(4) (1985): 85-103.

Law, Averill, and W. D. Kelton. 1991. *Simulation Modeling and Analysis*, New York, N.Y.: McGraw-Hill.

Lubachevsky, B.D. “Several Unsolved Problems In Large-Scale Discrete Event Simulations.” *Proceedings of the 1993 Workshop on Parallel and Distributed Simulation*, San Diego, CA, (May 1993): 60-67.

Lubachevsky, B.D. “How to Simulate Billiards and Similar Systems.” *Journal of Computational Physics*, 94 (1991): 255-283.

Rogers, R. “Autonomy: Simulation’s Next Events.” *Proceedings of the 1993 Winter Simulation Conference*. Los Angeles, CA, eds. Evans, G., Mollaghasemi, M., Russell, E., and Biles, W. (December 1993a): 1378-1379.

AUTHOR BIOGRAPHIES

EUGENE P. PAULO is a doctoral candidate in the Department of Industrial Engineering and Management Science at the University of Central Florida. He is a Major currently on active duty in the United States Army. He received a B.S. in Engineering from the United States Military Academy in 1981 and an M.S. in Operations Research from the Naval Postgraduate School in 1991. This paper is based on his dissertation.

LINDA C. MALONE is an Associate Professor in the Department of Industrial Engineering and Management Science at the University of Central Florida. She also serves as Director of the Masters Program in the department. She received her PhD from Virginia Tech in 1975. Her research interests include regression, statistics, and simulation.