

THE EFFECTS OF TIME ADVANCE MECHANISM ON SIMPLE AGENT BEHAVIORS IN COMBAT SIMULATIONS

Ahmed A. Al Rowaei
Arnold H. Buss
Stephen Lieberman

Naval Postgraduate School
MOVES Institute
700 Dyer Road
Monterey, CA 93943 USA

ABSTRACT

We investigate the effects of time advance mechanisms on the behavior of agents in combat simulations using some simple scenarios relevant to combat and agent-based models. We implement these simulation designs in two modeling packages that illustrate the differences between discrete-time simulation (DTS) and discrete-event simulation (DES) methodologies. Many combat models use DTS as their simulation time advance mechanism. We demonstrate that the presence and size of the time step as a modeling component can have a substantial impact on the basic characteristics of agent and simulation performance. We show that the use of a DTS method can degrade the modeling accuracy of changes in agent sensor range and detection outcomes, and also can compromise the ability of agents to travel to specific target destinations in a spatial simulation environment. We conclude that DES methodology successfully addresses these problems and is preferred as a time advance mechanism in these situations.

1 INTRODUCTION

The time advance mechanism (TAM) for simulations can be categorized as discrete event simulation (DES) models, often referred to as “next-event” (Law and Kelton 2000) or discrete time simulation (DTS), commonly referred to as “time-step.” DTS is the most commonly used time advance mechanism in military models across a wide range of modeling domains (Page and Smith 1998). Previous work has shown the impact of TAM on queuing modeling relevant to the military and academic modeling and simulation (M&S) communities (Buss and Al Rowaei 2010). Here we demonstrate that the DTS approach can introduce additional artifacts into a model, having to do with the size of the time step Δt , which can have a substantial impact on the outcomes of combat simulations and their interpretations. Specifically, we show that the use of DTS can compromise the modeling accuracy of changes in agent sensor range and detection, as well as the ability of agents to travel to specific target destinations in a spatial simulation environment.

In most applications using DTS, modelers cannot change or “adapt” the time step window size Δt and this introduces the opportunity for simulation results to be unknowingly affected by an innate and largely uncontrolled modeling constraint. When a time step model is used, there is a possibility that the results are in large part due to the size of the time step Δt rather than the other inherent characteristics of the simulation model.

To date there has been little systematic investigation of the effects of Δt in such simulations, nor have there been extensive studies of the differences between models of the same scenario constructed using different time advance mechanisms. Although potentially harmful effects of the size of the time step are

well-known in the numerical methods literature (Golub and Ortega 1992), there has been little work in traditional simulation modeling domains. What has been done appears to confirm the fact that the use of time steps affects the accuracy of the results (Park and Fishwick 2008).

The remainder of this paper is organized as follows. First we discuss the DTS and DES simulation approaches and their accompanying time-advance mechanisms, then describe the elements of the current combat models. We follow this with explanations of the specific combat simulation scenarios and a discussion of empirical investigations of the simulation outcomes as a function of the TAM. In the case of DTS approaches, we also discuss the choices of Δt and their impact on the simulation outcomes. We end with a discussion of the impact of time advance mechanisms on generalized military combat simulation and potential avenues for future work.

2 TIME STEP AND NEXT EVENT APPROACHES TO TIME ADVANCE MECHANISMS

In this section we will briefly present the two methods of time advance that will be considered: Discrete Time Simulation (DTS) and Discrete Event Simulation (DES). We note first that both methods involve modeling a system using state variables, and the dynamics of the model execution are represented by a sequence of state transitions. The difference is that for DTS all these state transitions occur at the same time, which must be a multiple of the time step Δt , whereas for DES they occur at arbitrary times (events).

2.1 Discrete Time Simulation

In Discrete Time Simulation (DTS) there is a fixed quantity, the time step Δt , that is the uniform increment by which time is advanced throughout the simulation (See Figure 1 a). From simulation instantiation, time is advanced in increments by Δt , then every state variable (and entity if used) is updated according to the logic defined by the model. The time-step method is used in many agent-based frameworks and combat simulations such as MANA, Pythagoras, IWARS, and JCATS (Alexander 2009). Some authors argue that the simplicity of discrete time simulation (DTS) explains its popularity across many application areas in military and academic M&S communities (Page and Smith 1998). Here we use the MANA 5, a modeling package popular among the military modeling and simulation community, as our DTS platform.

Within the time-step advance framework, simulation time is defined entirely in multiples of the time-step window Δt . Consequently, many state transitions appear to take place simultaneously when the simulation is “updated” at the next time-step. This is especially true when the pre-defined Δt is relatively large and there are many possible events (state transitions) that can take place in the simulation. Since all state transitions occur simultaneously in a DTS model, a “tiebreaking” mechanism must be used to mediate the ordering of these state changes. While it is beyond the scope of this paper to explore the variety of tie-breaking mechanisms used in various DTS packages, it is worth noting that the mechanisms generally employed to address ordering problems can impact the accuracy of the simulation interpretation of continuous systems, particularly with complex scenarios.

2.2 Discrete Event Simulation

Discrete event simulation (DES) uses the Next Event time advance mechanism, and starts by restricting state variables to have piecewise constant trajectories, and then identifying the state transitions with the instantaneous transitions, or “jumps”, made by them. Since these state transitions can occur at any time, and since all states are constant from any current simulation time until the time of the earliest state transition, time can advance to the next earliest state transition time. That is, simulation time progresses to the next event in the simulation (see Figure 1 b). Thus, time advances in non-constant intervals, potentially jumping over relatively long periods of time. Here we use the well-established Simkit (Buss 2011) modeling package for our DES environment.

DES methodology can be understood in terms of event graphs for a more intuitive visual manner. Nodes represent events while edges represent scheduling or cancelling relationships between the events

(Schruben 1983). The prototypes for Event Graphs are shown in Figure 2. The scheduling edge shown on the left is interpreted as follows: when event A occurs, then if Boolean condition (i) is true, event B is placed on the event list t time units in the future with priority P. When event B occurs, the value of its argument is equal to the value of the expression j when it had been scheduled. The canceling edge on the right is interpreted: the first scheduled occurrence of event B whose argument k equals the expression j is removed from the event list, providing Boolean condition (i) is true.

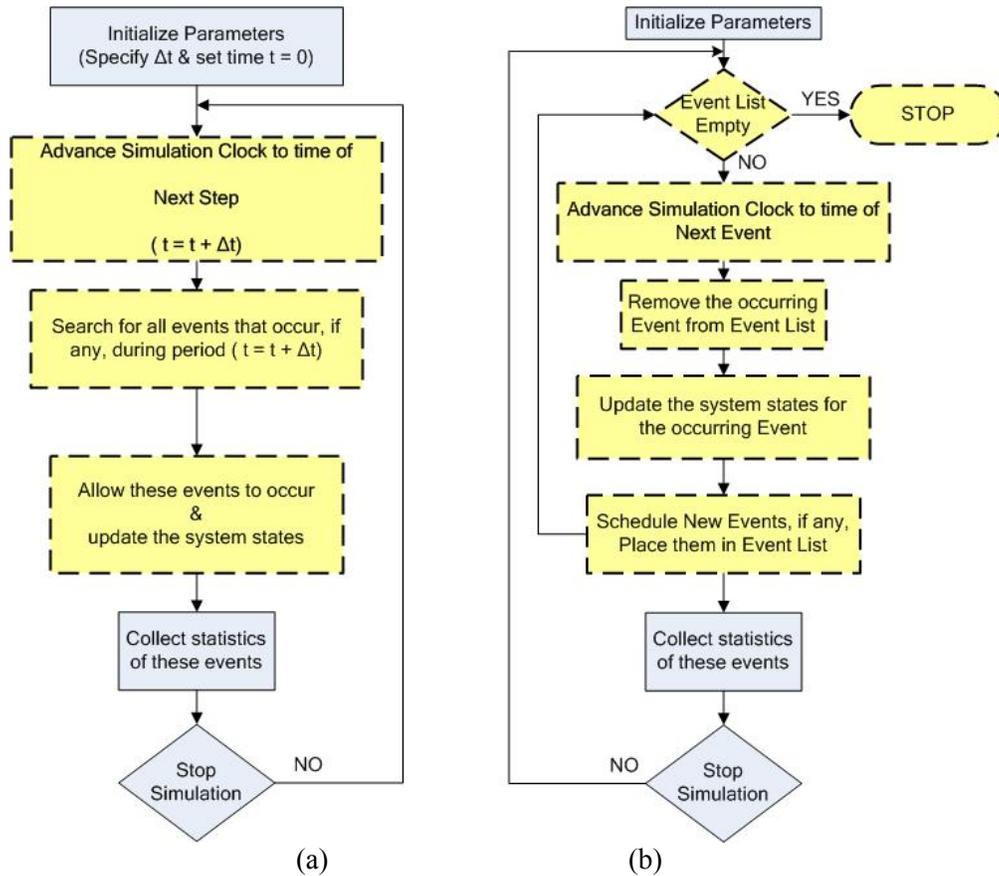


Figure 1: Overview of DTS (a) and DES (b).

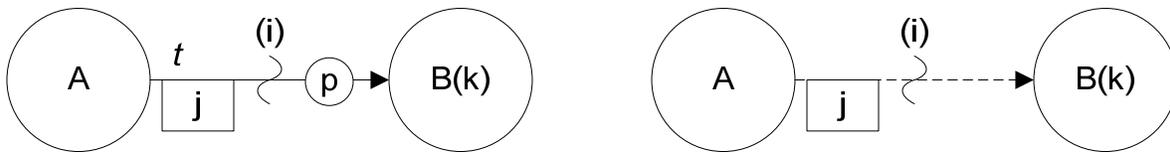


Figure 2: Event Graph prototypes (after Schruben 1983); scheduling (left) and canceling (right).

Event graphs of specific model components can be combined through the concept of listener patterns in Simkit. The concept of Listener Event Graph Object (LEGO) shown in Figure 3 enables the connection of disparate components to maximize the potential to reuse code objects and event components (Buss and Sanchez 2005, Buss 2001). The DES method is currently used in military combat simulations such as JDAFS, Combat XXI, and OneSAF (Alexander 2009).



Figure 3: LEGO Pattern (after Buss 2001, Buss and Sanchez 2002).

2.3 Discussion

As noted above, in DTS, the simulation must often employ a tie-breaking mechanism to deal with state transitions that occur simultaneously. In DES models this is not encountered as frequently since events can occur at arbitrary times rather than only at uniform intervals. Although it may be important in some situations to implement a tiebreaking priority scheme for events that happen to occur at exactly identical times, these are generally quite rare in DES, and thus even when they exist will tend to not have a profound impact on the results. The ordering of events in DES inherently model continuous systems more naturally than in DTS given that events are generally prioritized by the time they occur. This can be done explicitly by the modeler by assigning priorities to events (Schruben 1983, Buss 2011).

DES models tend to be more efficient than corresponding time step models (Banks et al. 2004). Much of this comes from the fact that “uninteresting” periods (i.e., ones for which there are few state changes) are quickly moved through in a DES model. In such circumstances, the typical time step size is too small. Correspondingly, when the model has many rapid state transitions, the DES model “slows down” accordingly, whereas the time step model tends to steps over these places and may not be able to capture the rate at which the changes are occurring. In these circumstances the time step is too big. Since the size of Δt must often be set very small to capture essential elements of the model, execution times can become much greater than for corresponding DES models. Likewise, when the time window is large in a DTS model, the execution time of the corresponding DES model will be greater.

3 SIMULATED COMBAT MODELING ELEMENTS

We now explore two essential scenarios in a combat simulation contexts that illustrate the effects of the time step Δt across a number of metrics. The results of DTS methods are compared to corresponding DES models. We do not attempt here to explore optimality of Δt values; instead we simply observe that DTS models can introduce a variety of conflating factors that result in substantial negative impacts on the usefulness of simulation results, even when simple modeling scenarios are used. In this case, we examine the three fundamental elements universal to many combat and agent simulations: agent movement, sensors and detections, and combat engagement.

3.1 Agent Movement

Agent movement takes place across two-dimensions and uses standard kinematic equations of motion with constant speed in both DTS and DES. All agents are homogenous in respect to movement. The equations of motion used in MANA for the DTS is (McIntosh 2009):

$$S = S_0 + v\Delta t + \frac{a\Delta t^2}{2} \tag{1}$$

where

$$v = v_0 + a\Delta t \tag{2}$$

and S = current location, S_0 = initial agent location, v = agent velocity, a = acceleration.

The equation of motion we use for movement in the DES model is (Buss and Sanchez 2005):

$$X(t) = X_0 + v(t - t_0) \tag{3}$$

where, $X(t)$ = current location at time t , X_0 = initial object location, v = object velocity, t_0 = time at initial location.

The specific architecture is shown in Figure 4, which utilizes the LEGO component framework (Buss and Sanchez 2002).

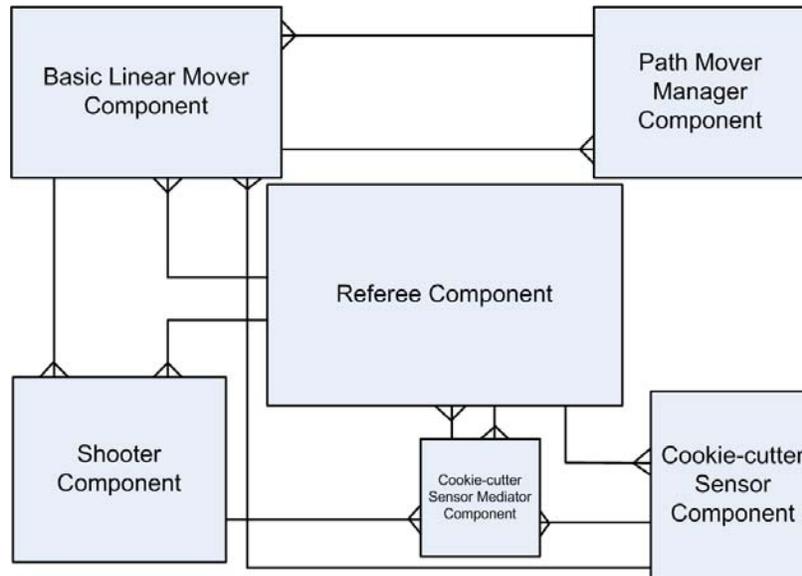


Figure 4: Component level architecture for discrete event combat agent LEGO.

3.2 Sensors and Detection

All agents have the same “cookie-cutter” type of sensors. That is, detection occurs with 100 percent probability within the complete effective range of each sensor. The sensors and detections are modeled in this simplistic fashion because we are not examining sensor models; rather we are testing the TAM effect on fundamental combat modeling elements. The sensor’s range of detection is a circle with the agent in the center of it. In both DTS and DES models all agents have a single cutoff range in which all contacts are detected and classified.

In DTS detection and classification of targets (i.e., friendly or enemy) occur at the same time step when the entity enters the effective sensor range. Given the present Δt , detection and classification can occur either on the periphery of the sensor range or within the sensor area, depending on the speed of the entities. In the event-driven methodology targets are detection and classified immediately upon entering the sensor range. This occurs because events in DES are pre-calculated.

3.3 Combat Engagement

All combat engagements are modeled as direct fire scenarios where the firing range of a simulated weapon is equal or greater than the effective sensor range of each agent. This means that agents can fire with a direct line of sight on their targets immediately upon detection and classification of enemy entities. While these events do not take place simultaneously in reality, many military combat simulations simplify the processes and consider these actions simultaneous.

For all combat engagement in the scenarios considered here, we use a probability of hit $P_h = 0.1$ and the probability of kill P_K upon a given hit is 100 percent. In DTS the firing rate is one shot per each time step Δt . For comparison reasons the firing rate is made to be equivalent with a Δt delay between each shot.

4 IMPACT OF TIME ADVANCE MECHANISMS ON THE MODELING ACCURACY OF CHANGES TO AGENT SENSOR RANGE AND DETECTION

In this section we create models of a simple scenario involving only two agents and examine the differences between the two time advance mechanisms.

4.1 Simulation Methods

The first experiment involved two homogenous agents that were identical in all respects except for sensor range. These agents moved in a linear fashion at 6 meters per second towards each other and engaged in direct fire when an enemy agent was detected. For the red agent, we varied the sensor range in unit increments from 30 to 60 meters, while for the blue agent the sensor range was kept at a constant 30 meters in all simulation runs. Thus, there were 30 scenario conditions, each of which was replicated 500 times in both the DTS and DES frameworks, for a total of 30,000 simulation runs. We collected metrics on the number of kills for each type of agent across all 30 scenarios conditions to elucidate the impact of sensor range changes.

In our simplified physics model, we did not account for acceleration or mass. In MANA 5, an agent mass of 0.1 is equivalent to its being able to reach cruising speed immediately without having to accelerate (McIntosh 2009). In this model, agents started movement at their origin waypoint and moved directly towards the other agent's origin waypoint at the opposite end of the simulation space. Thus, agents directly encountered each other during every simulation run without exception. In every run, either the red agent or the blue agent was killed. In some runs both agents fired their weapons; however there were no runs in which both agents were killed. For the DTS environment, the time step Δt was fixed at 0.1 seconds, the smallest Δt available within MANA 5 (McIntosh, 2009). This Δt was specifically selected to minimize the likelihood of DTS discretization errors, and the MANA simulations were executed in batch mode to minimize computation time.

A graphical display of the scenario implemented in Simkit and in MANA 5 are shown in Figure 5.

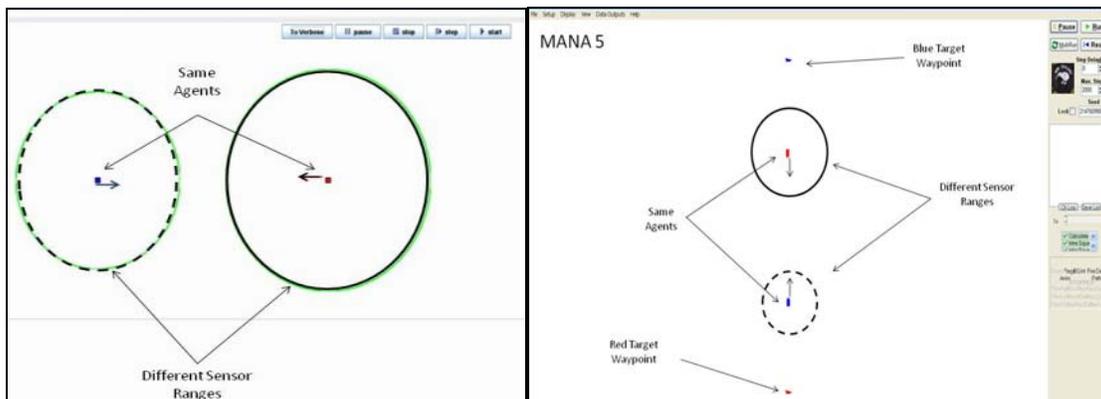


Figure 5: Simple agent combat in Simkit environment (left) and MANA 5 environment (right)

4.2 Results

In both the DTS and DES environments, when both agents had the same sensor range (i.e., 30 meters), they had an equal chance of killing the opposing agent, as seen in Figure 6. For all DES simulation runs, a change in the red agent's sensor range had a corresponding effect on the probability that the red agent would engage and kill the blue agent. In the event-driven model, even a modest increase in the red agent's sensor range had a positive impact on the probability that the red agent would successfully engage and kill the blue agent.

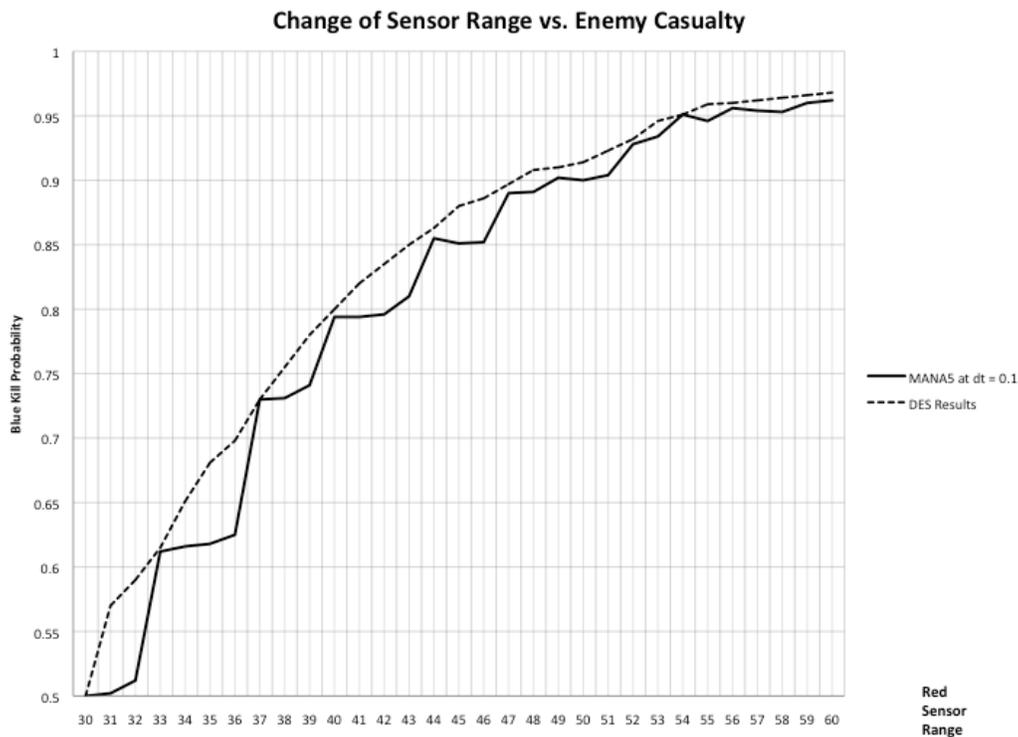


Figure 6: Time step size impact on sensor range change in two agents combat simulation model.

In DTS, the relationship between sensor range and blue kill probability was much more erratic. Sensor range changes appeared to impact blue kill probability in the DTS environment in a more step-wise fashion. The incremental changes to sensor range exhibit a non-monotonic relationship where increases in relative sensor range had no impact and sometimes actually decreased the average effectiveness of the agent.

4.3 Discussion

The choice of time advance mechanism in combat simulations clearly affects the accuracy of modeling results having to do with changes in sensor ranges. Given these results, it appears that discrete time simulations often fail to record important impacts of sensor range changes on the behavior of agents in a combat simulation. This has significant implications for the use of combat simulations in decision-making contexts. Recommendations from simulation results made regarding changes to sensor ranges in operating environments may be flawed depending on the time advance mechanism chosen.

As an example, suppose that an operational commander would like to increase the probability of detection for a combat element by 25%. The results here would indicate that using a DES model would suggest that small increases in sensor range would achieve the improvements to the measure, whereas a DTS model would not indicate such improvements. Conversely, a DTS model is likely to suggest that a potentially suitable sensor range change would have little or no actual effect on combat effectiveness. Since simulation models are typically just one element in a decision-making process, the choice of time advance mechanism in simulation may have a large impact on combat effectiveness as well as the final costs of military operations.

5 IMPACT OF TIME ADVANCE MECHANISMS ON AGENT MOVEMENTS AND TARGET LOCATION RECOGNITION

Our next example is even simpler involving a single agent attempting to move to a desired waypoint.

5.1 Simulation Methods

In our second experiment, we used a single agent moving in a linear fashion at a rate of 12 meters per second toward a single target destination 10,000 meters away from the agent at the opposite end of the simulation space. A movement speed of 12 meters per second was chosen to eliminate the effects of simulated mass in the MANA environment. In MANA 5, agent speed must be kept below certain thresholds otherwise the explorations of time step size in the current simulation are affected by uncontrollable confounding factors that are inherent to the MANA modeling environment. In all simulation conditions, the agents moved at a constant fixed speed toward a fixed destination waypoint. We compared the event-driven and time-step modeling environments and varied the time step size Δt in the DTS simulations in increments of 0.2 seconds from 0.1 to 1.3 seconds to explore the impacts of agent movements toward the target waypoint.

Since this simulation experiment is completely deterministic, we report results for single-run scenarios. In both the DTS and DES environments, the agent knows the exact location of the target destination in advance prior to movement. As in the first experiment, we used a simplified physics model that did not incorporate the modeling of acceleration or mass. Likewise, in this simulation, agents started movement at their origin waypoint and moved directly towards the target.

5.2 Results

For all simulation runs, we collected statistics about the time it took for the agent to reach the target waypoint “trip time”, and the number of meters by which the agent passes or overshoots the target destination “miss distance” when this occurred. The DES environment provided the shortest trip time to the target destination at 13.88 seconds, identical to the analytical solution for this movement. In the time-step environment, each increased time step window Δt extended the trip time for the agent. This is illustrated in Figure 7.

As the size of the time step increases, we observe an emergent phenomenon of agent movement in discrete time simulations. We see that the agent must “backtrack” to reach the target waypoint as the time step increases, although the agent will eventually reach the destination for time step sizes under a critical threshold. However, starting at a Δt of 1.3 seconds, the agent never stops at the target destination, instead oscillating back and forth between points on either side of the target waypoint forever.

We also observe a monotonic increasing relationship between the time step size Δt and the miss distance of the agents. Figure 8 below reports the cumulative miss distance for each DTS agent as a function of time step size. The cumulative miss distance is calculated by adding together all miss distances travelled by the agent before reaching the target waypoint. That is, in simulations with the time step window below the critical threshold described above, the agents overstep (oscillate) to either side of the target at decreasing intervals before they reach the target destination. Both the number and distance of these oversteps increase as a function of increase in time step size Δt .

5.3 Discussion

The choice of time advance mechanism in combat simulations greatly impacts agent movement as well as target location recognition. Moreover, the size of the time step Δt has important implications for the outcomes of combat simulations, including the ability to generate emergent simulation behavior when Δt exceeds a critical threshold. In the current simulation, that critical threshold is $\Delta t = 1.3$, but this depends upon all time-sensitive elements of the simulation and almost certainly depends also on inherent properties of the simulation environment that are not under the direct control of the modeler.

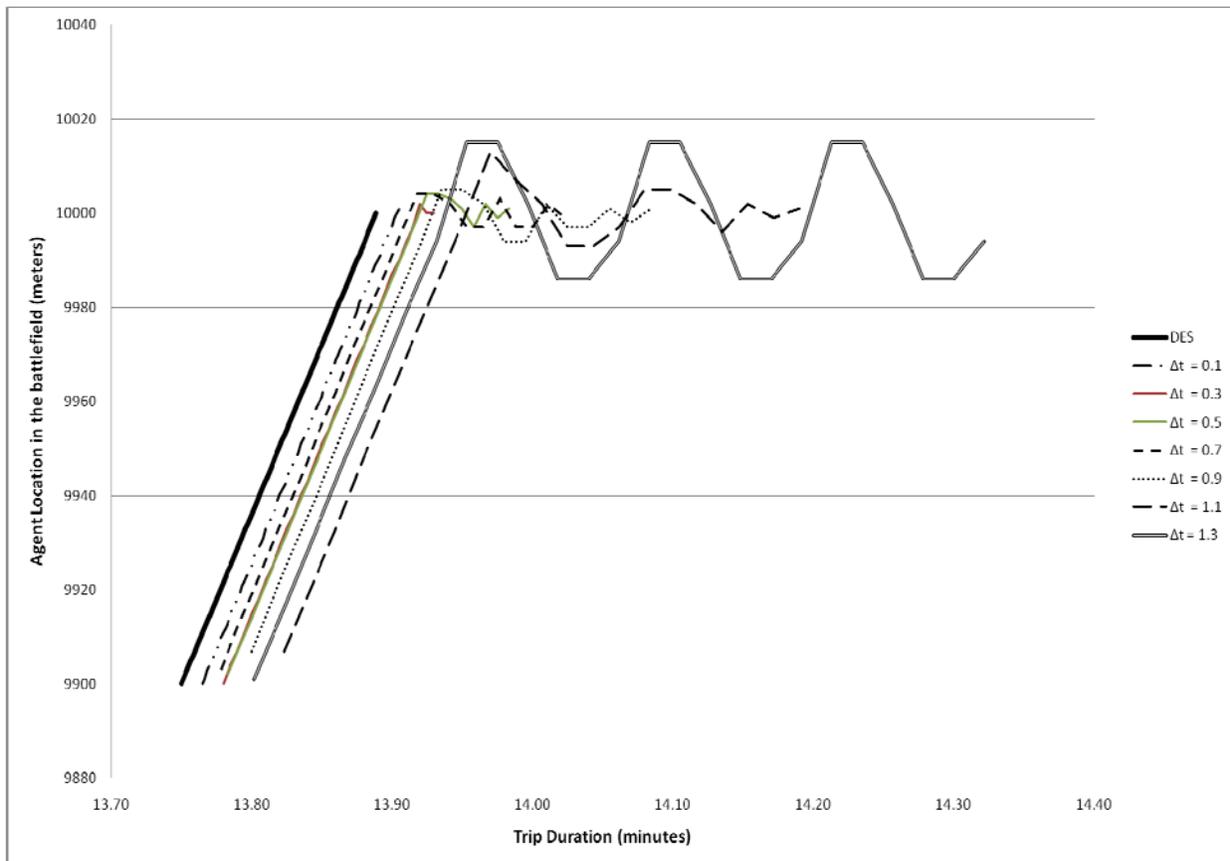
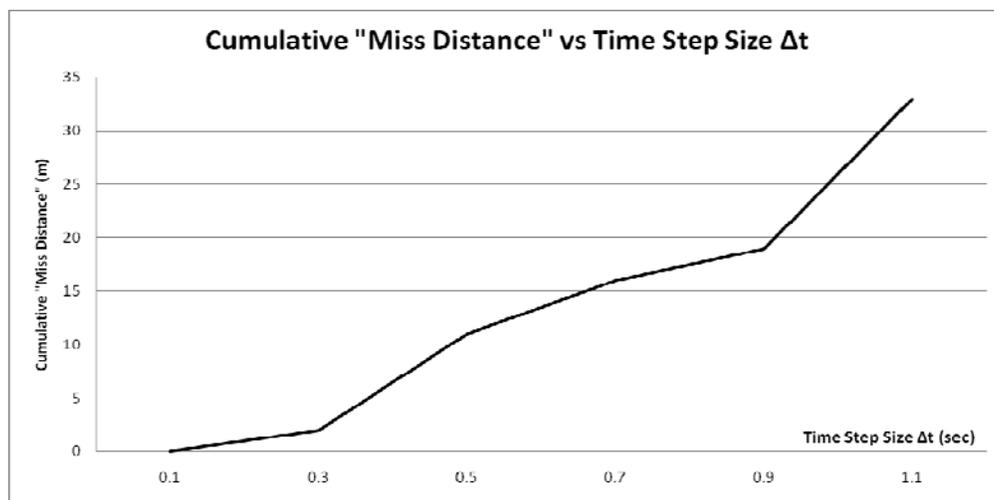


Figure 7: Agent locations and trip time over model runs from DES and DTS models



Time Step Size Δt	0.1	0.3	0.5	0.7	0.9	1.1	1.3
Cumulative Miss-Distance (meters)	0	2	11	16	19	33	Inifinte
Number of Oversteps	0	1	3	5	6	6	Inifinte
Trip Duration Time (minutes)	13.91	13.93	13.98	14.02	14.09	14.19	Inifinte

Figure 8: Agent cumulative miss distance from target location for multiple time step sizes in DTS model

There is a direct relationship between time step size and the amount of time it takes for an agent to reach a target destination. This is true even though the agents are moving at the same speed, and the destination is at the same distance throughout all simulation scenarios. Although the equation of motion is the same throughout, the trip time is impacted by the time step size that is an element of the DTS environment. Given the effects of the time advance mechanism on even the simplest combat simulation scenario depicted here, we propose that the impacts of TAM choice could produce even greater and less understood effects on the more complex scenarios that are common in military M&S.

6 CONCLUSIONS

Our results indicate that the choice of time advance mechanism can have profound impacts on the behavior of agents in combat simulations. The two small examples form the groundwork for a wide variety of more complex simulations. We have shown that the TAM greatly affects the modeling accuracy of changes to agent sensors and detection, as well as agent movements and target location recognition. Moreover, the impact of TAM can produce emergent phenomena and simulation results that cannot be explained by virtue of the elements in the actual system being modeled. This is demonstrated saliently by the oscillation behavior that occurs in the agent movement studies when the time-step window size passes a certain critical threshold.

For DTS models, determining the size of the time-step presents a significant and ongoing problem. To date, there are no generally accepted empirical methods used to identify and choose the appropriate time step size in combat simulations. Instead, modelers often relay on guesswork as well as “trial and error” in selecting the time window size. In general, the approach taken to time step size determination in combat simulation involves starting with a relatively large Δt and reducing the size until there are no readably noticeable visible errors through a portion of the simulation run. However, it is rare for analyses of time step window size effects to be undertaken during the simulation design or before execution of any simulation runs.

Even so, there are many benefits to using a time-step TAM for particular combat simulation models. DTS is generally simple to implement and understand since there are many “off-the-shelf” simulation environments with easily navigable graphical user interfaces, such as the MANA software used here. Additionally, there are few restrictions on the types of variables and trajectories that are possible in many software packages, thus allowing for potentially greater latitude in defining state variables with such a framework. This flexibility makes DTS particularly popular for modeling multi-agent systems, especially in low-resolution models such as MANA (Straver, Vincent and Fournier 2006). We note that in no way are results to be construed as a criticism of the MANA software. Rather, our critique is of the underlying DTS methodology itself.

We found no evidence of odd emergent agent behaviors using the DES methodology for any of the combat simulations studied. While the DES methodology may be less intuitive to simulation designers familiar with DTS environments, it is clear that the potential drawbacks from DTS are too great to use this method with combat simulations in decision-making contexts.

In a complex simulation, the execution time of a DES model could possibly be slower than the corresponding DTS model with a large time-step Δt . While large time steps in a DTS model may execute faster than a corresponding DES model, our results here suggest that the outcomes of the DTS model could be seriously flawed. Perhaps more importantly, the modeler may be unaware that the model is producing erroneous results because of the size of the time step. The probability for quickly visible substantial errors produced in the agent behavior and simulation results will drive modelers to choose smaller time-steps. However, even if the problems identified with DTS here were resolved, the size of the time-step window required to match the measures obtained via most corresponding DES models will cause the DTS model to take substantially longer to execute than the DES model.

Perhaps the most significant implication of the research presented here is the fact that most modelers in the military and academic M&S communities are unaware of the significant impacts that TAM and time-step size can have on simulations. Even in situations where the potential for effects of TAM are

acknowledged, there is bound to be a large amount of uncertainty as to whether the TAM is actually influencing the results of complex simulations. The scenarios examined here were chosen for their simplicity and near universal applicability. They were constructed in such a way that analytic results are available to verify the accuracy and behavior of the agents in the simulation. In general, combat simulations are used to aid decision-makers in situations where the analytic results are unknown, or unavailable.

Complex combat simulations are used in scenarios that resist direct analytic methods. We have seen here that even the simplest of simulations are affected by the TAM in ways that make simulation results unreliable in exactly those contexts in which modelers use complex simulations to gain insight. Given the benefits of discrete-event simulations and the inherent limitations of the time-step methodology demonstrated here, these results suggest that the use of DES models is strongly preferable to DTS models in combat simulations, particularly in military decision-making contexts.

Further work in this area is ongoing; specifically, we are examining more complex models in the combat simulation domain, including modeling different types of sensors, scenarios involving multiple agent interactions, as well as scenarios involving naval warfare. Another rich area in which DES has not been heavily employed is agent-based models, specifically models of human behaviors and social interactions. The use of DES to model each agent state transitions as events occurs and the logic of local clock to each agent will be examined in future work.

REFERENCES

- Alexander, R. 2009. "Models, Gaming and Simulation." In: *Class notes in Military and Simulation SYST-683 and OR-649*, George Mason University.
- Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2004. *Discrete-Event System Simulation*. 4th ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.
- Buss, A. 2001. "Discrete Event Programming with Simkit." *Simulation News Europe* 32(33):15-25.
- Buss, A. 2011. Simkit website. Accessed March 30. <http://diana.nps.edu/Simkit/>.
- Buss, A., and A. Al Rowaei. 2010. "A Comparison of the Accuracy of Discrete Event and Discrete Time." In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Huan, and E. Yücesan, 1468–1477 Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Buss, A. and P. J. Sanchez. 2002. "Modeling Very Large Scale Systems: Building Complex Models with LEGOs (Listener Event Graph Objects)." In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, 732-737. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Buss, A., and P. J. Sanchez. 2005. "Simple Movement and Detection in Discrete Event Simulation." In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 992–1000 Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Golub, G. H., and J. M. Ortega. 1992. *Scientific Computing and Differential Equations*. San Diego, CA: Academic Press.
- Law, A., and D. Kelton. 2000. *Simulation Modeling and Analysis*. Third Edition. New York: McGraw Hill.
- McIntosh, G. C. 2009. *MANA-V (Map Aware Non-Uniform Automata – Vector) Supplementary Manual*. Provided with MANA Software Package.
- Park, H., and P. A. Fishwick. 2008. "A Fast Hybrid Time-Synchronous/Event Approach to Parallel Discrete Event Simulation of Queuing Networks." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 795–803. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Page, E. H., and R. Smith. 1998. "Introduction to Military Training Simulation: A Guide for Discrete Event Simulationists." In *Proceedings of the 1998 Winter Simulation Conference*, edited by D. J.

Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, 53–60 Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Straver, M. C., E. Vincent, and P. Fournier. 2006. “Experiences with the MANA simulation tool.” Defence Research and Development Canada (DRDC) Valcartier Operational Research Team. Technical Memorandum DRDC Valcartier TM 2006-404.

Schruben, L. 1983. “Simulation Modeling with Event Graphs.” *Communications of the ACM* 26:957-963.

AUTHOR BIOGRAPHIES

AHMED “ALI” AL ROWAEI is a Ph.D. candidate in the MOVES Institute at the Naval Postgraduate School. The work presented here is part of his Ph.D. dissertation in the application of discrete event simulation. His e-mail address is aalrowae@nps.edu.

ARNOLD BUSS is a Research Associate Professor in the MOVES Institute at the Naval Postgraduate School. His research is in the area of Discrete Event Simulation modeling, particularly in developing effective and reusable frameworks for developing models. His e-mail address is abuss@nps.edu.

STEPHEN LIEBERMAN is a research scientist at the Naval Postgraduate School specialized in analytical modeling and simulation of modern military operations and decision support. His research centers on simulation analysis, multi-agent systems, and complex adaptive systems. His e-mail address is stephen-lieberman@gmail.com.