Network Load in Distributed Interactive Simulation

Kuo-Chi Lin*
University of Central Florida, Orlando, Florida 32826-0544
and
Daniel E. Schab†
U.S. Naval Air Warfare Center, Orlando, Florida 32826-3224

I. Introduction

DEAD reckoning plays a vital role in the continuing development of distributed interactive simulation (DIS). Dead-reckoning algorithms, when formulated to suit the particular needs of a given simulation network, offer optimal realism and speed through a reduction in update traffic within a network.[1]

II. Dead-Reckoning Equations

Dead reckoning is a procedure of extrapolation that uses current or previous state variables (position, velocity, and acceleration) to predict the future position. All the extrapolation formulas and integration algorithms are candidates for dead reckoning. However, most of the formulas derived for the fixed step time situation are not suitable for dead reckoning. In the dead-reckoning process, there are two different step times, as shown in Fig. 1. In the figure, h is the dead-reckoning step time, and Δt are update step times, where $\Delta t_{-1} = t_0 - t_{-1}$, $\Delta t_{-2} = t_{-1} - t_{-2}$, etc., t_0 is the time of current update, t_{-1} is the time of the last update, and t_{-2} is the time of the update before the last update, etc. In general, $h \leq \Delta t$ and $\Delta t_{-1} \neq \Delta t_{-2} \neq \ldots$, etc.

The zero-order equation is given by

$$x_i = x_0 \tag{1}$$

where x_i is the dead-reckoning x (position) at time $t_i = t_0 + ih$, $i = 0, 1, 2, \ldots$, which counts the dead-reckoning time steps from the update time. The vehicle remains at its position until receiving the new position update.

The first-order equation is given by

$$x_i = x_0 + v_0 ih \tag{2}$$

where v_0 is $\dot{x}$ (velocity) at time t_0. An alternative formula for the first-order equation is given by

$$x_i = x_0\left[1 + \frac{ih}{\Delta t_{-1}}\right] - x_{-1}\frac{ih}{\Delta t_{-1}} \tag{3}$$

where x_{-1} is the update position at the last update time t_{-1}.

The second-order equation is given by

$$x_i = x_0 + v_0 ih + a_0 (ih)^2/2 \tag{4}$$

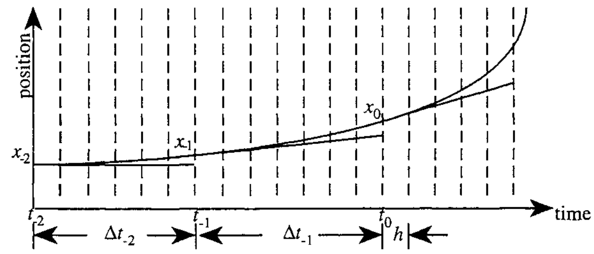

Fig. 1 Update time in dead reckoning.

where a_0 is the $\ddot{x}$ (acceleration) at time t_0. The second-order equation has more alternatives than the first-order equation, as given by the following equations:

$$x_i = x_0 + v_0\left[ih + \frac{(ih)^2}{2\Delta t_{-1}}\right] - v_{-1}\frac{(ih)^2}{2\Delta t_{-1}} \tag{5}$$

$$x_i = x_0\left[1 - \frac{(ih)^2}{\Delta t_{-1}^2}\right] + x_{-1}\frac{(ih)^2}{\Delta t_{-1}^2} + v_0\left[ih + \frac{(ih)^2}{\Delta t_{-1}}\right] \tag{6}$$

$$\begin{aligned} x_i = {} & x_0\left[1 + \frac{ih}{\Delta t_{-1}} + \frac{(ih)^2 + ih\Delta t_{-1}}{\Delta t_{-1}(\Delta t_{-1} + \Delta t_{-2})}\right] \\ & - x_{-1}\frac{(ih)^2 + ih(\Delta t_{-1} + \Delta t_{-2})}{\Delta t_{-1}\Delta t_{-2}} \\ & + x_{-2}\frac{(ih)^2 + ih\Delta t_{-1}}{\Delta t_{-2}(\Delta t_{-1} + \Delta t_{-2})} \end{aligned} \tag{7}$$

For third-order position dead-reckoning equations, the time derivative of acceleration $\dot{a}_0$ is not available from the update protocol data unit (PDU), therefore, extrapolations from the previous updates are required for all cases. Some examples are given by the following equations:

$$x_i = x_0 + v_0 ih + a_0\left[\frac{(ih)^2}{2} + \frac{(ih)^3}{6\Delta t_{-1}}\right] - a_{-1}\frac{(ih)^3}{6\Delta t_{-1}} \tag{8}$$

$$\begin{aligned} x_i = {} & x_0 + v_0\left[ih - \frac{(ih)^3}{3\Delta t_{-1}^2}\right] + v_{-1}\frac{(ih)^3}{3\Delta t_{-1}^2} \\ & + a_0\left[\frac{(ih)^2}{2} + \frac{(ih)^3}{3\Delta t_{-1}}\right] \end{aligned} \tag{9}$$

$$\begin{aligned} x_i = {} & x_0\left[1 - \frac{3(ih)^2}{\Delta t_{-1}^2} - \frac{2(ih)^3}{\Delta t_{-1}^3}\right] + x_{-1}\left[\frac{3(ih)^2}{\Delta t_{-1}^2} + \frac{2(ih)^3}{\Delta t_{-1}^3}\right] \\ & + v_0\left[ih + \frac{2(ih)^2}{\Delta t_{-1}} + \frac{(ih)^3}{\Delta t_{-1}^2}\right] + v_{-1}\left[\frac{(ih)^2}{\Delta t_{-1}} + \frac{(ih)^3}{\Delta t_{-1}^2}\right] \end{aligned} \tag{10}$$

$$\begin{aligned} x_i = {} & x_0 + v_0\left[ih + \frac{2(ih)^3 + 3(ih)^2(2\Delta t_{-1} + \Delta t_{-2})}{6\Delta t_{-1}(\Delta t_{-1} + \Delta t_{-2})}\right] \\ & - v_{-1}\frac{2(ih)^3 + 3(ih)^2(\Delta t_{-1} + \Delta t_{-2})}{6\Delta t_{-1}\Delta t_{-2}} \\ & + v_{-2}\frac{2(ih)^3 + 3(ih)^2\Delta t_{-1}}{6\Delta t_{-2}(\Delta t_{-1} + \Delta t_{-2})} \end{aligned} \tag{11}$$

$$\begin{aligned} x_i = {} & x_0\left[1 + \frac{(ih)^3}{\Delta t_{-1}^3}\right] - x_{-1}\frac{(ih)^3}{\Delta t_{-1}^3} + v_0\left[ih - \frac{(ih)^3}{\Delta t_{-1}^2}\right] \\ & + a_0\left[\frac{(ih)^2}{2} + \frac{(ih)^3}{2\Delta t_{-1}}\right] \end{aligned} \tag{12}$$

Received Feb. 24, 1994; revision received March 5, 1995; accepted for publication March 10, 1995.

*Assistant Professor of Aerospace Engineering. Member AIAA.

†Aerospace Engineer, Fighter Attack Branch, Training Systems Division. Member AIAA.

Table 1 Number of PDUs (PDU/s) vs threshold, dead reckoning on position only

Equation	Order	60 in.	24 in.	12 in.	6 in.
1	0	214 (6.1)	487 (13.9)	837 (23.9)	1315 (37.6)
2	1	35 (1.0)	56 (1.6)	79 (2.3)	112 (3.2
3	1	49 (1.4)	77 (2.2)	108 (3.1)	152 (4.3)
4	2	16 (0.46)	23 (0.66)	30 (0.86)	39 (1.1)
5	2	22 (0.63)	29 (0.83)	38 (1.1)	49 (1.4)
6	2	20 (0.57)	28 (0.80)	35 (1.0)	49 (1.4)
7	2	29 (0.83)	39 (1.1)	50 (1.4)	64 (1.8
8	3	15 (0.43)	19 (0.54)	25 (0.71)	31 (0.89
9	3	14 (0.40)	19 (0.54)	25 (0.71)	30 (0.86)
10	3	16 (0.46)	20 (0.57)	30 (0.86)	38 (1.1)
11	3	19 (0.54)	24 (0.69)	31 (0.89)	38 (1.1)
12	3	15 (0.43)	17 (0.49)	23 (0.66)	31 (0.89)

Table 2 Number of PDUs (PDU/s) vs threshold, dead reckoning on orientation only

Equation	Order	20 deg	10 deg	5 deg	2 deg
1	0	41 (1.2)	77 (2.2)	149 (4.3)	320 (9.1)
2	1	17 (0.49)	27 (0.77)	36 (1.0)	62 (1.8)
3	1	27 (0.77)	38 (1.1)	51 (1.5)	81 (2.3)
5	2	21 (0.60)	31 (0.89)	44 (1.3)	60 (1.7)
6	2	22 (0.63)	28 (0.80)	39 (1.1)	58 (1.7)
7	2	29 (0.83)	41 (1.2)	50 (1.4)	79 (2.3)
10	3	29 (0.83)	35 (1.0)	50 (1.4)	69 (2.0)
11	3	29 (0.83)	37 (1.1)	50 (1.4)	69 (2.0)

Table 3 Number of PDUs (PDU/s) vs threshold, both position and orientation

Equations		Orders		60 in. and	24 in. and	12 in. and	6 in. and
Position	Orientation	Position	Orientation	20 deg	10 deg	5 deg	2 deg
1	1	0	0	218 (6.2)	489 (14.0)	839 (24.0)	1315 (37.6)
2	1	1	0	48 (1.4)	85 (2.4)	155 (4.4)	323 (9.2)
2	2	1	1	41 (1.2)	64 (1.8)	88 (2.5)	129 (3.7)
4	2	2	1	25 (0.71)	34 (1.0)	49 (1.4)	71 (2.0)
9	6	3	2	26 (0.74)	35 (1.0)	46 (1.3)	67 (1.9)

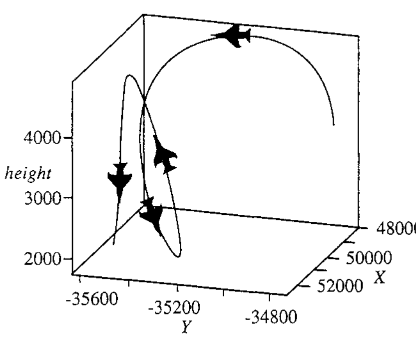

Fig. 2 Test flight trajectory.

For the orientation dead reckoning, the entity state PDU provides Euler angles and angular rates. Since the angular acceleration is not available, all dead-reckoning equations that need acceleration information can not be used. The qualified candidates are Eqs. (1–3), (5–7), (10), and (11).

The previous equations are by no means the complete set of dead-reckoning equation candidates. They are chosen in this study because their truncation errors are smaller than other equations. A more complete list of the dead-reckoning equations can be found in Ref. 2.

III. Network Load

The flight trajectory chosen to test the network loads of the dead-reckoning equations listed in the previous section is a 35-s flight trajectory with a 40-Hz update rate; this flight trajectory was recorded from a high-fidelity F-16 flight simulator when the pilot was engaged in an air combat maneuvering (ACM) training exercise. Figure 2 shows the three-dimensional plot of this trajectory. Without dead reckoning, the simulator has to send out an entity state PDU every 0.025 s to provide the updated information on its position and orientation. There are 1400 PDUs in 35 s.

The dead-reckoning equations discussed in the previous section are used on the flight trajectory with several different thresholds. In Tables 1–3, the numbers of PDUs (with the PDU rates in the parentheses) are listed against the thresholds. As the threshold increases, the number of PDUs generated decreases. From the consideration of network load, a large threshold is preferable. However, a large threshold represents not only large error, but also creates jitters in the dead-reckoning trajectory, which in turn reduces the visual fidelity.

IV. Conclusions

If the network load is the only criterion to choose the dead-reckoning equation, any combination of second-order (or higher-order) equation for position and first-order (or higher-order) equation for orientation is a very good choice. In general, higher-order equations need more computation time. If the simulator has a severe limitation in computation power, it has to use a lower-order equation, provided that the network traffic load is not a problem. Another factor is that the higher-order equations (third-order in position, second-order in orientation) need extrapolation from the variables of the previous updates. This can cause larger errors when there is disturbance in the data. Based on these considerations, the combination of Eq. (4) for position and Eq. (2) for orientation is a better choice.

Acknowledgments

The research reported in this article is sponsored by the Defense Modeling and Simulation Office under the Department of Defense and administered by the Naval Air Warfare Center Training Systems Division, Contract N61339-92-C-0110.

References

[1]Lin, K. C., "Dead-Reckoning in Distributed Interactive Simulation," *Proceedings of the 3rd Annual Conference of the Chinese-American Scholars Association of Florida*, Miami, FL, 1992, pp. 1–5.

[2]Hoffman, L. L., and Lin, K. C., "An Experimental Design for Detecting Optimal Simulated Flight Performance on Distributed Interactive Systems," Inst. for Simulation and Training, Univ. of Central Florida, TR IST-CR-93-16, Orlando, FL, May 1993.

Vortex Burst Model for the Vortex Lattice Method

Steven A. Brandt*
U.S. Air Force Academy,
Colorado Springs, Colorado 80840

Introduction

RECENT flight testing of aircraft designed to maneuver at very high angles of attack suggest that such capabilities may be useful in future fighter aircraft. Inexpensive computational tools that accurately predict forces and moments at high angles of attack may be useful for the conceptual design of such aircraft. The vortex lattice method (VLM) has long been used as an analysis tool for attached flow cases. More recently,[1,2] VLM has been used with empirical vortex burst data to give good predictions of trim requirements at higher angles of attack. In the current work, a vortex core model is analytically derived in a form that allows it to be easily incorporated into the VLM. Vortex burst locations predicted by the coupled vortex burst model/vortex lattice method (VBM/VLM) for several swept wings at high angles of attack are in good agreement with empirical data. This makes it possible to substitute model-predicted burst points for the empirical data of Refs. 1 and 2.

Formulation

A model for the vortex core is derived from the steady, incompressible Navier–Stokes equations written for a cylindrical coordinate system centered on the vortex core. The derivation follows, in general, the approaches developed by Mager[3] and Krause[4] and extends the method described in Refs. 5 and 6. The vortex is assumed to be slender and axisymmetric. The u, v, and w velocities are defined in the x, r, and θ cylindrical coordinate system aligned with the vortex central axis. The following variables are defined:

$$\bar{x} = \frac{x}{L} \qquad \varepsilon = \frac{\delta_{\mathrm{ref}}}{L} \qquad \bar{r} = \frac{r}{\varepsilon L} = \frac{r}{\delta_{\mathrm{ref}}} \qquad Re = \frac{\rho U_\infty L}{\mu}$$

$$\bar{p} = \frac{2p}{\rho U_\infty^2} \qquad \bar{u} = \frac{u}{U_\infty} \qquad \bar{\nu} = \frac{\nu}{\varepsilon U_\infty} \qquad \bar{w} = \frac{w}{U_\infty}$$

where L is the length of the vortex to the wing trailing edge, δ_{ref} is the vortex core radius at $x = L$, and ρ, μ, and U_∞ are freestream density, viscosity, and velocity, respectively. The overbars denote dimensionless quantities, but will be omitted in the subsequent analysis.

As in Ref. 6, the ratio of the vortex core diameter to the vortex length is assumed to be of the order of the inverse of the square root of the Reynolds number. After eliminating terms that become negligible for large Reynolds numbers, the nondimensional equations of motion become

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{v}{r} = 0 \tag{1a}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial r} = -\frac{\partial p}{\partial x} + \frac{1}{Re\varepsilon^2}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) \tag{1b}$$

$$\frac{w^2}{r} = \frac{\partial p}{\partial r} \tag{1c}$$

$$u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial r} + \frac{vw}{r} = +\frac{1}{Re\varepsilon^2}\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial(rw)}{\partial r}\right] \tag{1d}$$

Algebraic profiles for radial variation of axial and circumferential velocity are chosen and Eq. (1c) is integrated from $r = 0$ to $r = \delta$. For physically reasonable choices of velocity profiles, the integrated form of Eq. (1c), after negligible terms are discarded, is

$$p(x, 0) = p(x, \delta) - K_a(\Gamma^2/a) \tag{2}$$

where

$$a = \pi\delta^2 \quad \text{and} \quad \Gamma = 2\pi\delta w_\delta$$

The subscript δ denotes quantities at the core edge and K_a depends on the choice of the w velocity profile.

The previously chosen u profile is next used in Eq. (1a) to determine the corresponding v profile. Then, Eq. (1d) is integrated over the area of the vortex core to obtain

$$\frac{1}{a}\frac{\mathrm{d}a}{\mathrm{d}x} = \frac{K_1}{\Gamma}\frac{\mathrm{d}\Gamma}{\mathrm{d}x} - \frac{K_2}{u_\delta}\frac{\mathrm{d}u_\delta}{\mathrm{d}x} + \frac{K_3}{Re\varepsilon^2 a u_\delta} \tag{3}$$

where the values of K_1, K_2, and K_3 depend on the velocity profiles chosen. If a simple solid body rotation is used for the w profile, K_2 and K_3 are zero and $K_1 = 1$, and so Eq. (3) can be integrated with respect to x to yield

$$a \propto \Gamma$$

If more complex w profiles are chosen, the expression for a can still be approximated as

$$a = K_0\frac{\Gamma^{K_1}(x - x_0)}{u_\delta^{K_2}} \tag{4}$$

where K_0 depends primarily on the product $Re\varepsilon^2$. Assuming that the core circumferential velocity profile is sufficiently

Received Oct. 4, 1993; presented as Paper 94-0074 at the AIAA 32nd Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 10–13, 1994; revision received Dec. 22, 1994; accepted for publication May 20, 1995.

*Associate Professor of Aeronautics, Department of Aeronautics, 2354 Fairchild Drive, Suite 6H24. Member AIAA.