Smoothing of Dead Reckoning Image in Distributed Interactive Simulation

Kuo-Chi Lin,* Morgan Wang,† and Jie Wang‡
University of Central Florida, Orlando, Florida 32816-2370
and
Daniel E. Schab§
U.S. Naval Training Systems Center, Orlando, Florida 32826

I. Introduction

DEAD reckoning (DR) plays a vital role in the continuing development of distributed interactive simulation. 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. In the dead-reckoning process, when a new update of position is received from another entity, a correction in position is usually required so that the entity is depicted in simulation as correctly as possible.[1-4] If the new position is put in the image display system immediately, it can cause jumps in the displayed image. The preferred method is to gradually correct the position of the entity over a period of time; this is called smoothing.

Figure 1 is an example of the smoothing technique. When the updated position is received at time t_n, instead of jumping to the new position, a smoothing model is maintained. First the new dead reckoning model is extrapolated using the updated information at time t_n to find the position at a future time, e.g., $t_n + p\Delta$, where p is the number of smoothing steps ($p = 2$ in the figure) and Δ is the dead-reckoning step time; then the positions from time t_n and $t_n + (p - 1)\Delta$ are obtained by interpolating the positions at $t_n - \Delta$ and $t_n + p\Delta$. In this model, the transition from one dead reckoning model to another is smoother. The number of smoothing steps p is an important parameter. In general, the number used is between 5–15.

II. Smoothing Equations

The smoothing model shown in Fig. 1 is a straight-line (linear) smoothing algorithm. Its formula is given by

$$x_i = x_{-1} + (x_p - x_{-1})i/(p + 1), \qquad i = 1, \ldots, p \tag{1}$$

where $x_i = x[t_n + (i - 1)\Delta]$ is the smoothing position; $x_{-1} = x(t_n - \Delta)$ is the starting position of smoothing, i.e., the last dead-reckoning position before update; and $x_p = x(t_n + p\Delta)$ is the final position of smoothing, i.e., dead-reckoning position p steps after update. The smoothing positions are $x(t_n), \ldots, x[t_n + (p - 1)\Delta]$.

Received June 7, 1995; presented as Paper 95-3398 at the AIAA Flight Simulation Technologies Conference, Baltimore, MD, Aug. 7–10, 1995; revision received Sept. 6, 1995; accepted for publication Sept. 7, 1995.

*Associate Professor, Institute for Simulation and Training, 3280 Progress Drive. Member AIAA.

†Assistant Professor, Department of Statistics.

‡Graduate Student Assistant, Institute for Simulation and Training, 3280 Progress Drive.

§Aerospace Engineer, Simulation Dynamics Branch, 12350 Research Parkway. Member AIAA.

In the previous model, if the straight line is replaced by a cubic spline, a cubic-spline smoothing algorithm is obtained. Its formula is given by

$$\begin{aligned} x_i = {} & \frac{[(p + 1)\Delta(v_{-1} + v_p) + 2(x_{-1} - x_p)]i^3}{(p + 1)^3} \\ & + \frac{[-(p + 1)\Delta(2v_{-1} + v_p) + 3(x_p - x_{-1})]i^2}{(p + 1)^2} \\ & + i\Delta v_{-1} + x_{-1}, \qquad i = 1, \ldots, p \end{aligned} \tag{2}$$

where $v_{-1} = \dot{x}(t_n - \Delta)$ is the velocity at the beginning of the smoothing (matching the velocity of the last dead reckoning before update) and $v_p = \dot{x}(t_n + p\Delta)$ is the velocity at the end of the smoothing (matching the velocity of the next dead reckoning). The cubic-spline smoothing algorithm has continuity in velocity.

There are other possibilities in smoothing. In this Note, only these two equations are discussed. As will be shown in the next section, the differences among smoothing algorithms are not as significant as whether smoothing is applied or not.

III. Smoothness Rating

To test the effectiveness of the smoothing equations introduced in the last section, a method to rate the smoothness is needed. The authors have developed a new rating system based upon multinomial logit models (MLM). The new rating system consists of a set of predict functions for image smoothness. The detailed description of the derivation of this rating system can be found in another paper by the authors.[5] One major advantage of MLM is that it can predict a subject's choice from a discrete set of options, i.e., the choice of smoothness rating based on

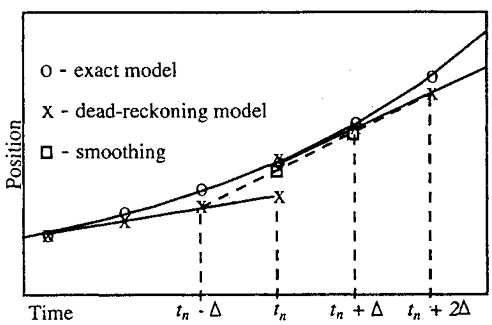

Fig. 1 Smoothing concept.

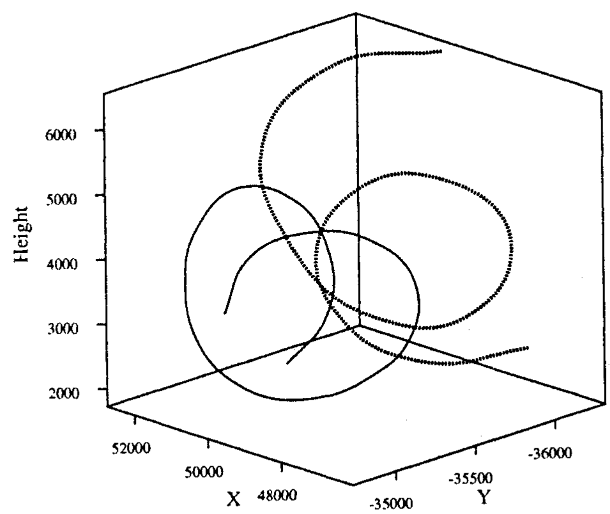

Fig. 2 Trajectories of two F-16 fighters. ····· target and —— pilot.

Table 1 Results of smoothness experiment

Smoothing methods	No. of smoothing steps	DR thresholds, smoothness groups, ft 3	6	12	24	36	48	60
N	0	S	A	A	A	U	U	U
L	5	S	S	S	S	A	U	U
L	10	S	S	S	S	S	A	A
L	15	S	S	S	S	S	A	A
C	5	S	S	S	S	S	A	U
C	10	S	S	S	S	S	A	A
C	15	S	S	S	S	S	S	S

Note: N = no smoothing, L = linear smoothing, C = cubic-spline smoothing, S = smooth, A = acceptable, and U = unsmooth.

Table 2 Maximum rms errors of smoothing algorithms

Smoothing methods	No. of smoothing methods	DR thresholds, smoothness groups, ft 3	6	12	24	36	48
N	0	1.48 (3.00)	2.85 (6.00)	5.50 (11.98)	10.63 (23.96)	15.73 (35.98)	21.46 (47.95)
L	5	1.63 (**3.32**)	3.10 (6.00)	5.90 (11.98)	11.29 (23.96)	16.58 (35.98)	22.44 (47.95)
L	10	1.82 (**4.51**)	3.40 (**6.27**)	6.40 (11.98)	12.11 (23.96)	17.65 (35.98)	23.70 (47.95)
L	15	2.15 (**4.63**)	3.71 (**7.00**)	6.84 (11.98)	12.84 (23.96)	18.62 (35.98)	24.85 (47.95)
C	5	1.69 (**3.43**)	3.18 (6.00)	6.01 (11.98)	11.45 (23.96)	16.77 (35.98)	22.65 (47.95)
C	10	2.03 (**6.05**)	3.67 (**6.80**)	6.73 (**12.44**)	12.60 (**24.50**)	18.24 (**36.56**)	24.33 (**48.37**)
C	15	2.41 (**8.16**)	4.20 (**8.95**)	7.49 (**12.90**)	13.80 (**25.15**)	19.75 (**37.26**)	26.06 (**49.24**)

Note: The boldface numbers represent the errors exceeding the thresholds. N = no smoothing, L = linear smoothing, and C = cubic-spline smoothing.

continuous explanatory variables. The theoretical bases of the MLM method can be found in Agresti's book.[6]

The experiment is conducted using two 35-s trajectories, as shown in Fig. 2, as the test platform. One represents the target airplane and the other one represents the pilot airplane. They were recorded from two high-fidelity F-16 flight simulators when the pilots were engaged in an air combat maneuvering training exercise against each other. The second-order dead reckoning, with or without smoothing, is applied to the target trajectory with different levels of thresholds. The MLM method is used to obtain the smoothness ratings of the resulting trajectories.

Table 1 shows the smoothness ratings. If no smoothing algorithm is applied, the trajectory is rated smooth only for small thresholds, 3 ft and below. For larger thresholds, the smoothness ratings are degraded to acceptable or unsmooth. When the linear smoothing algorithm is applied, the trajectory is rated smooth up to a 24-ft threshold for five smoothing steps, and up to 36 ft for 10 and 15 smoothing steps. It is a significant improvement.

The cubic-spline smoothing algorithm performs even better. For 5 and 10 smoothing steps, the trajectory is rated smooth up to a 36-ft threshold, and up to a 60-ft threshold for 15 smoothing steps.

The tradeoff of the use of a smoothing algorithm is the increase of errors. When a new protocol data unit (PDU) is received it indicates that the error will exceed the threshold in the next step time. In the case of no smoothing, the dead-reckoning trajectory is corrected to the true trajectory, and the error is reset to zero. In the case of smoothing, the smoothing trajectory takes the number of smoothing steps to catch up with the dead reckoning trajectory. These smoothing steps add errors to the trajectory.

Table 2 shows the rms and maximum (in parentheses) errors of the smoothing trajectories of several levels of thresholds. All show the same trend: the trajectories with no smoothing have the lowest errors. The linear smoothing trajectories have higher errors and the cubic-spline smoothing algorithm produces the highest errors. Also, the trajectories with more smoothing steps have higher errors than the ones with less steps, with all other conditions the same. This trend is directly opposite to the smoothness ratings. In other words, the algorithm with better smoothness rating has higher error.

In the table, the maximum errors that exceed the thresholds are listed in boldface typesets. In the case of no smoothing, none of the maximum errors exceeds the thresholds, as it should be for dead reckoning. Applying smoothing to a dead-reckoning trajectory can cause the errors to exceed the threshold in some instances. When longer smoothing times are used (more smoothing steps), the chances that a new PDU arrives before the smoothing trajectory catch up with the dead-reckoning trajectory are increased. Trying to do smoothing from the previous smoothing trajectory, instead of previous dead-reckoning trajectory, can increase the error. This is the major reason that errors exceeding thresholds are observed more often at smaller thresholds, as the table shows.

IV. Conclusions

The smoothing algorithm can significantly improve the visual smoothness under a distributed interactive simulation environment. The tradeoff for smoothing is that the error of the trajectory will increase. The visual smoothness difference between smoothing and no smoothing is significant; but the differences among different smoothing algorithms are minor. Cubic-spline smoothing algorithm not only has higher error but also consumes more computer resources to execute the procedure. Therefore, the linear smoothing algorithm is recommended for most applications.

The number of smoothing steps has a similar tradeoff consideration. More smoothing steps can make the visual image smoother; however, the error is higher. The choice of the smoothing steps depends on the accuracy (fidelity) requirement of the exercise.

Acknowledgments

This research was partially supported by National Science Foundation Grant DMS-9305864, and the University of Central Florida Division of Sponsored Research Grant 11-92-402.

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*, Chinese-American Scholar Association of Florida, Miami, FL, 1992, pp. 1–5.

[2]Lin, K. C., and Schab, D., "The Performance Assessment of the Dead Reckoning Algorithms in DIS," *Simulation*, Vol. 63, No. 5, 1994, pp. 318–325.

[3]Lin, K. C., "Dead Reckoning and Distributed Interactive Simulation," *Proceedings of SPIE's International Symposium on Aerospace, CR58* (Orlando, FL), SPIE Optical Engineering Press, Bellingham, WA, 1995, pp. 16–36.

[4]Lin, K. C., and Schab, D., "Network Load in Distributed Interactive Simulation," *Journal of Aircraft*, Vol. 32, No. 6, 1995, pp. 1392, 1393.

[5]Lin, K. C., Wang, M., Wang, J., and Schab, D., "Rating of Image Smoothness in Distributed Interactive Simulation," *International Journal in Computer Simulation* (submitted for publication).

[6]Agresti, A., *Categorical Data Analysis*, Wiley, New York, 1990.

Simple Prediction Method for Location of Vortex Breakdown on Delta Wings

Lance W. Traub*

Texas A&M University, College Station, Texas 77843

Nomenclature

c	=	chord
g	=	similarity function
K	=	Sychev similarity parameter
U_∞	=	freestream velocity
x	=	chordwise direction
α	=	angle of attack
Γ	=	vortex circulation
$\Gamma_{\alpha > BD\text{-}TE}$	=	circulation at trailing edge at angles of attack beyond that at which breakdown occurs
ε	=	wing apex half-angle
Λ	=	leading-edge sweep angle

Subscripts

BD	=	breakdown
BD-TE	=	breakdown at trailing edge
r	=	root

Introduction

BEYOND moderate angles of attack the flowfield over slender delta wings is usually characterized by the formation of leading-edge vortices. These vortices form and are strengthened by vorticity emanating from the wing's leading edge. However, as the angle of attack of the wing increases, the leading-edge vortices may break down, a phenomena associated with a marked deceleration of the vortex core, and increase in the diameter of the resulting structure. Unwelcome characteristics of vortex breakdown (e.g., a flattening of the lift curve and a destabilizing nose up pitching moment, etc.) have resulted in numerous studies of the phenomena.[1–3] Generally, attempts to find a universal scaling parameter to correlate breakdown have not been satisfactory.[4] However, studies have suggested that breakdown is associated with the concentration and distribution of vorticity, and the ability of the vortex to convect it axially downstream.[5] The apparent success of the Rossby number[6] (i.e., the ratio of the axial velocity to the maximum rotary velocity at the edge of the viscous core), as a criterion for the prediction of the onset of breakdown, also supports the concept of a threshold level for the convection of axial momentum or vorticity downstream.

Although numerical predictions of the onset of vortex breakdown are feasible, it is still useful to have simple analytic expressions for predicting the location of the breakdown,

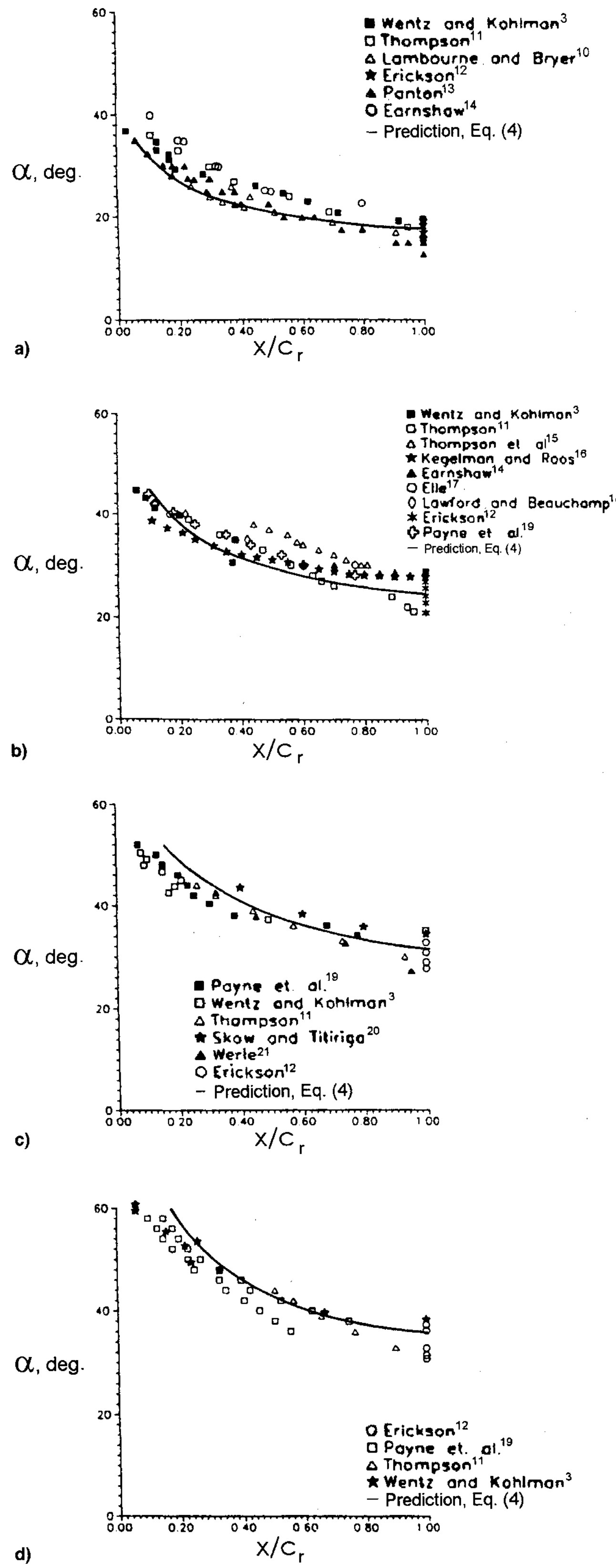

Fig. 1 Location of vortex breakdown position for various leading-edge sweep angles. Λ = a) 65, b) 70, c) 75, and d) 80 deg.

Received Oct. 18, 1995; revision received Nov. 22, 1995; accepted for publication Nov. 27, 1995.

*Graduate Student, Aerospace Engineering Department.