# A Psycho-Perceptual Comparison of the Dead Reckoning and the Hybrid Strategy Model Entity State Update Prediction Techniques

**Séamus McLoone, Alan Kenny, Tomás Ward and Declan Delaney**

*Department of Electronic Engineering,*
*National University of Ireland Maynooth,*
*Maynooth, Co. Kildare, IRELAND*
*E-mail: seamus.mcloone; akenny; tomas.ward@eeng.nuim.ie*

## ABSTRACT

**Distributed Interactive Applications (DIAs) typically employ entity prediction mechanisms in order to reduce the number of packets sent between clients across the network. This in turn counters the effect of network latency and can improve the consistency of the distributed application. Dead Reckoning (DR) is currently the most commonly used entity state prediction mechanism but a more recent technique called the Hybrid Strategy Model (HSM) has been proposed in the research literature. This alternative method has been shown to further reduce the number of update packets required to maintain a consistent state in a DIA. However, there is a distinct lack of end-user perceptual analysis of these techniques. In other words, does the HSM method improve the gaming experience of the user compared to DR? A reduction in packet count may improve issues with latency but can adversely degrade the modelling quality and therefore the overall level of consistency is unknown. Hence, this paper proposes the novel use of user perception as a means to determine the quality of a given entity state update mechanism. Here, we compare DR and HSM from a user perceptual viewpoint by collecting linguistic feedback on short scenes recorded from a racing game. Details of the experiment and the obtained results are presented within.**

**Keywords** – Distributed Interactive Applications, Entity Update Mechanisms, Psycho-Perceptual Analysis, Dead Reckoning, Hybrid Strategy Model

## I INTRODUCTION

One of the main aims of a distributed interactive application (DIA) is to maintain a high level of consistency amongst the client participants. This is of particular importance in the gaming world where networked multiplayer games are becoming more and more dominant. However, issues such as network latency and jitter are causing major concerns as they conspire to reduce the level of consistency. As a result, it is desirable to minimise the number of packets that must be sent across a network in order to assist in the maintenance of a consistent view for each remote user. This counters the effect of latency in the network and therefore can improve the consistency achieved. Various methods for achieving this reduction in packets exist, including entity state prediction mechanisms, area of interest management, data compression and dynamic load balancing [1 - 6].

One of the most popular techniques used to date is the entity state prediction mechanism known as Dead Reckoning (DR). This was formalised in the IEEE Distributed Interactive Simulation (DIS) standard [3] and has become the standard for commercial games, such as Doom, Quake and Tribes II. Dead Reckoning is a method of predicting a user's future actions based on their dynamics, which results in the transmission of less data to remote nodes.

Another, more recent, entity state prediction method has been proposed in the research literature. This method is known as the Hybrid Strategy Model (HSM) and differs from DR by using a priori knowledge of user behaviour to build a set of strategies by which to model the user [4, 5]. It is purported that this method outperforms DR by significantly reducing the number of update packets required to maintain a consistent state in the distributed application [4]. However, there is currently a distinct lack of psycho-perceptual analysis of such mechanisms. In other words, there is no analysis available to determine if the HSM or the DR mechanisms perform better from a user's viewpoint. What is the impact on the gaming experience of the end-user as a result of these mechanisms?

A reduction in packet count may improve issues with latency but, unfortunately, this can also adversely degrade the performance quality of the underlying predictive models and, as a result, the overall level of consistency remains largely unknown. Hence, this paper proposes the novel use of user perception to subjectively compare and contrast the DR and HSM prediction techniques. This involves collecting linguistic user feedback from a series of short video recordings from a racing game. The analysis of this data is presented later.

The rest of this paper is structured as follows. The next section provides a brief description of the DR and the HSM methods. Section III details the design and implementation of the experiment used to collect information pertaining to the end-user perceptual experience. The obtained data is then analysed and discussed in section IV. Finally, the paper ends with some concluding remarks and suggestions for future work in section V.

## II DEAD RECKONING AND HYBRID STRATEGY MODEL

For the convenience of the reader, a brief description of the Dead Reckoning and the Hybrid Strategy Model entity state prediction techniques are now presented. The basic convergence algorithm used in each case is also given. At this stage, it is important to mention that these methods rely on an underlying error threshold value that determines when an update packet needs to be transmitted. There are two different threshold metrics that can be used and both of these are also briefly described.

### *A. Dead Reckoning (DR)*

Under DIS, once an entity is created the information pertaining to this entity is transmitted to all participating remote nodes. Each remote node then attempts to predict this entity movement based on its trajectory data. There are many ways in order to extrapolate a player's position. The most basic and common of these is to set the new position to the transmitted position and the new velocity to the transmitted velocity, which is known as first order DR. Another common DR technique is to use the transmitted position and velocity along with the transmitted acceleration information.

The local node also keeps a model of itself under DR, which is continually compared to its actual position. Once the actual position differs from its predicted position by a set amount, known as its error threshold, an update is sent to all the remote nodes informing them of its updated trajectory. Once the remote nodes receive this new information they update their models to reflect the latest transmitted data. In this paper, we simply make use of the basic first order DR model. Further information on Dead Reckoning can be found in [3, 7].

### *B. Hybrid Strategy Model (HSM)*

HSM is a novel prediction mechanism that uses a priori knowledge of user behaviour to build a set of strategy paths. These can be constructed in various ways: (1) by recording past actual entity movements in the environment, (2) by heuristically identifying possible strategies based on the examination of the environment and (3) by employing automatic path-finding techniques. In this paper, we employ the first method to generate a base strategy. This was generated from analysing a number of path traversals and choosing the one that represented the most common behaviour. This is effectively the centre path of the race track shown in Figure 1. Additional strategies are then generated in real time based on their distance from the main strategy. In other words, if an entity moves a pre-defined distance from the base strategy, a new strategy is created from its new position. This strategy is effectively parallel to the base one, but shifted by a specified value. If the entity moves too far from the base strategy, thus indicating that it is not adopting the desired strategy, than a DR model is employed by default.

The hybrid strategy approach involves dynamically switching between a DR model and one of a number of strategy models. The distance from the actual point to the various models is continuously calculated. When the entity movement is within a defined threshold value of one of the strategy models then that strategy model is employed to describe the entity movement. If no suitable strategy models exist, then the DR model is used by default. The HSM implemented in this paper employs a simple switching mechanism, switching instantaneously between the most suitable models. Further information on the operation of the HSM and generating appropriate strategy paths can be found in [4, 5].

### *C. Convergence*

One of the key problems with predicting a remote entity's position is that once the updated position has been received, the remote entity's position has to be rapidly corrected. If the remote entity is moved directly into the new position this can result in a disjoint and unnatural behaviour from the user. Convergence is an attempt to naturally blend the incorrect current player trajectory into the updated player trajectory. Here, we employ a basic convergence routine.

In the case of DR, when an update packet is received, a linear set of points is generated that connects the current position to a future predicted position. This latter value is based on the received updated position and updated velocity values. The intermediate path is then played out with increased velocity. With HSM, the convergence involves a blending of the underlying strategy paths using a suitable weighting function. Initially, the weighting favours the current strategy path before quickly moving towards the actual strategy path. Further detail can be found in [8].

### *D. Error Threshold Metrics*

Two different error threshold metrics can be employed in both DR and HSM. The first is known as *spatial* and is simply determined by the spatial difference between the actual and modelled position of an entity's motion. The alternative threshold metric is known as *time-space* and uses local absolute inconsistency measures in determining when an update is required [8]. In simple terms, the time-space error is calculated as the integral of the modelling error over time, i.e. the area under the modelling error curve. These two metrics can be used separately or together in the form of a hybrid metric, as outlined in [9].

In this paper, the DR and the HSM methods are analysed from a user perceptual viewpoint. Both error threshold metrics are considered in each case. The next section outlines the design and implementation of the experiment used to obtain the user perceptual feedback.

## III EXPERIMENTATION

### *A. Video Clips*

In order to obtain the required feedback, a set of video clips was created. This was achieved by recording the movements of a 'bot', i.e. a computer controlled entity, under various conditions. The first video clip was used for benchmarking purposes and consisted of the bot's motion under ideal conditions, i.e. the values of latency, jitter and error threshold were all set to zero. Each subsequent video consisted of the bot modelled with either DR or HSM


Figure 1: Java Media Recorder showing Part of elliptical racing Track

using various error threshold metrics and values. In these clips, the bot's motion was subject to latency of 200ms. Jitter was set to ±10% of the latency value. These values reflect typical network conditions [10]. Various subjects were then asked to compare the bot's motion with the perfect one.

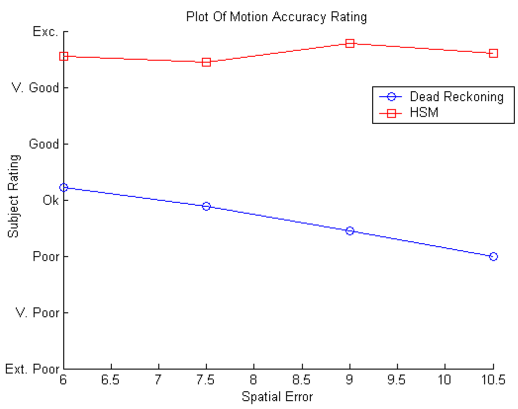
An elliptical racing course was chosen for simplicity, as shown in Figure 1. This was created using the Torque Game Engine [11]. Here, the game measurements, such as the spatial distance, are calculated in Torque Game Units (tgu). As a reference point, the track used in this experiment is approximately 100 tgu in width and 250 tgu in length.

The main strategy path used for the HSM approach in this paper was generated from analysing a number of path traversals and choosing the one that represented the most common behaviour. Additional strategies are then generated in real time based on their distance from the main strategy. If this distance exceeds a pre-defined limit, than the DR model is employed. The base strategy model used here is effectively the centre path of the track. This is a simple implementation of the HSM approach. In less spatially restricted scenarios the development of strategy models may be non-trivial.

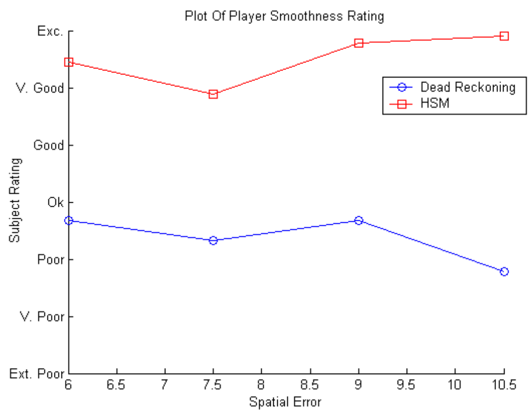
Finally, the various games scenes were recorded as AVI files using FRAPS (http:// www.fraps.com), a utility designed for recording game footage.

### *B. User Feedback*

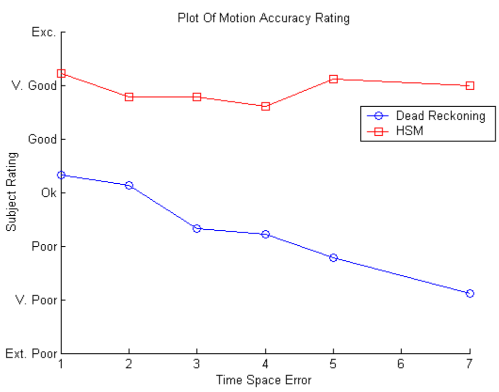
The goal of this experiment was to perceptually compare the DR and HSM methods for various error metrics and threshold values. In order to achieve this, feedback relating to these models had to be obtained, specifically in terms of how accurate a subject perceived the model to be relative to the perfect scenario. Previous experimentation [12] revealed that in the case where a bot



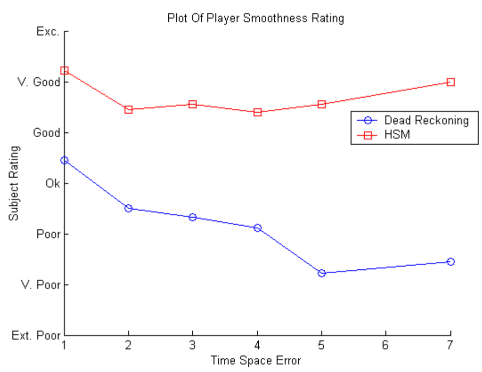

(a)




(a)




(b)




(b)

Figure 2: Average Motion Accuracy rating for DR and HSM when (a) a spatial and (b) a time-space error metric was employed

Figure 3: Average Player Smoothness rating for DR and HSM when (a) a spatial and (b) a time-space error metric was employed

followed a particular path accurately but appeared to be slightly 'jumpy' in its motion, subjects rated this as very poor in terms of accuracy. In an attempt to alleviate this problem, users were asked, in this experiment, to provide separate feedback on both the smoothness and accuracy of the bot's motion.

Linguistic variables were used to obtain subject feedback. This avoided the problem of having to assign an exact numerical value to a given level of accuracy or smoothness. The following linguistic variables were used: *Extremely Poor*, *Very Poor*, *Poor*, *Okay*, *Good*, *Very Good*, *Excellent*. Seven variables were used in accordance with the fact that, as humans, we can reliably perceive up to 7 different states [13]. The test application used in this experiment is shown in Figure 1.

### *C. Experimental Setup*

Prior to the start of the experiment, each subject was given an explanation of the task in hand, and were also shown a demonstration video. The tests began once the subject felt confident that they understood the requirements. The experiment consisted of the subject watching a video clip once and then, on completion, rating it in terms of Player Smoothness and Motion Accuracy.

Each experiment consisted of a total of 28 video clips - 6 perfect play-outs, 20 unique models and 2 duplicates. Each video lasted about 20 seconds and the entire experiment lasted approximately 15 minutes. Subjects were only told that the first video clip was perfect.

## IV ANALYSIS AND DISCUSSION

A total of 10 subjects took part in this experiment, consisting of 6 males and 4 females, ranging in age from 15 to 35 years. All subjects had some level of experience with using a computer, while 6 had some experience with computer games and 4 had experience with networked games.

DR and HSM were both examined using spatial error threshold values ranging from 6 to 10.5 tgu and time space threshold values ranging from 1 to 7 tgu. The relationship between these two metrics is explored in detail in [9].

Figure 2(a) represents the average motion accuracy obtained for the various spatial threshold values for both entity state prediction mechanisms, while Figure 2(b) represents similar results for different time space threshold values. Figures 3(a) and 3(b) present similar results for player smoothness.

It is worth noting that these results are based on a small number of participants. Nevertheless the results obtained are a fair reflection of the performance of both the DR and the HSM for the given set of video clips.

The results show that the rating for DR falls below acceptable between 6 and 8 tgu for both the player smoothness and motion accuracy. On the other hand the HSM exhibits much higher ratings, obtaining a '*very good'* rating for all the measured thresholds. This shows that not only does HSM result in less packets being generated than DR, but also maintains a much higher perceptual rating. Interestingly, at the highest tested error threshold, 10.5 tgu, HSM only generates one packet of data but still remains more than acceptable to the end user. However, this simply reflects the simplicity of the underlying track as well as the high accuracy of the models used as part of the HSM approach. Nevertheless, the same analogy extends to more complex situations provided that suitable and accurate models can be determined for use in the HSM technique.

Like the spatial error threshold, HSM also outperforms DR when using the time space error threshold. The acceptance rating for DR rapidly falls below acceptable levels between 2 and 3 tgu for both the player smoothness and motion accuracy. However, the HSM maintains a rating of approximately '*very good'* throughout.

It is worth noting that HSM performed slightly worse when using a time-space threshold metric compared to when it used a spatial threshold one. The former results in sending more update packets, which is purported to improve absolute consistency [9], yet this is not evident in the obtained results. The results here suggest that updating too frequently can lower the end users experience as they notice the entity having to correct its position more often and these corrections are regarded as inaccurate.

Another point of note rising from these results relates to the operation of the HSM. As previously stated, HSM attempts to choose the best available strategy model for a given circumstance. If no appropriate model exists then a DR model is employed by default. Error threshold values are normally static, i.e. they are fixed to a constant value prior to runtime. Therefore, the HSM strategy models and its associated DR model would end up with the same error threshold. Clearly this experiment shows that different models require different error thresholds in order to get the best packet rate to maintain the overall end user experience. In essence the error threshold should be dynamically chosen depending on the current model employed. From the work presented here, a strategy model could have a significantly larger error threshold, regardless of error metric, than a DR model.

## V CONCLUDING REMARKS

This paper has shown how psycho-perceptual measures can be employed as a useful means to compare different entity update mechanisms. Here, Dead Reckoning and the Hybrid Strategy Model approaches were compared from an end-user's viewpoint. Both spatial and time space threshold metrics were considered for both techniques.

HSM was shown to give far better perceptual ratings than Dead Reckoning for both the spatial and time space error threshold. HSM also requires less update packets to be transmitted across the network. Therefore, the results confirm that for this particular application, namely an online racing game, that HSM is far better suited to delivering a good end user experience. In fact HSM appears to give very good results even at the highest error threshold for this type of application. Other applications, such as fast paced First Person Shooter (FPS) games, have more intense interaction and may require tighter thresholds. Nevertheless it would be expected that HSM would still perform better than DR. This requires further investigation.

Finally, for the HSM, it is proposed that dynamically choosing an error threshold based on the current model would result in the best possible end user experience, while minimizing the number of update packets. Future work will look at this issue also.

## ACKNOWLEDGEMENTS

This work is supported by Science Foundation Ireland and Enterprise Ireland - grant no. IRCSET/SC/04/CS0289.

## REFERENCES

[1] Delaney, D., T. Ward and S. McLoone. "On consistency and network latency in distributed interactive applications: A survey - Part I." *Presence: Teleoperators and Virtual Environments* 15(2), pp. 218-234, April 2006.

[2] Delaney, D., T. Ward and S. McLoone. "On consistency and network latency in distributed interactive applications: A survey - Part II." *Presence: Teleoperators and Virtual Environments* 15(4), pp. 465-482, August 2006.

[3] IEEE. *IEEE Standard for Distributed Interactive Simulation - Application Protocols IEEE Std 1278.1-1995* IEEE, 1995.

[4] Delaney, D., T. Ward and S. Mc Loone "Reducing Update Packets in Distributed Interactive Applications using a Hybrid Model". *16th International Conference on Parallel and Distributed Computing Systems, August 13-15*, Reno, USA, pp. 417-422, 2003.

[5] D. Marshall, D. Delaney, S. McLoone and T. Ward, "Exploring the spatial density of strategy models in a realistic distributed interactive application", DS-RT'04, Budapest, Hungry, 2004.

[6] S. Singhal and M. Zyda, "Networked virtual environments: design and implementation.", ACM Press SIGGRAPH Series, Addison-Wesley, Reading, Massachusetts, 1999.

[8] D. Delaney, S. McLoone and T. Ward, "A novel convergence algorithm for the hybrid strategy model packet reduction technique", *IEE Irish Signals and Systems Conference*, Dublin, Ireland, 1-2 September, pp.118-123, 2005.

[9] Roberts, D., D. Marshall, S. McLoone, D. Delaney, T. Ward and R. Aspin "Exploring the use of local inconsistency measures as thresholds for dead reckoning update packet generation". *Distributed Simulation and Real Time Applications*, Montreal, Canada, pp. 195 - 202, 2005.

[10] T. Defanti and R. Fraser, Teleimmersion in the grid: blueprint for a new computing infrastructure, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. pp. 131-155, 1998.

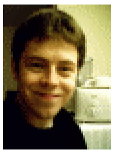
[11] Lloyd, J. "The Torque Game Engine", Game Developer Magazine, 11 (8), pp. 8-9, September 2004.

[12] A. Kenny, S. McLoone, T. Ward and D. Delaney, "Using User Perception to Determine Suitable Error Thresholds for Dead Reckoning in Distributed Interactive Applications", IEEE Irish Signals and Systems Conference, Dublin, Ireland, 2006.

[13] G. Miller, "The magic number 7 plus or minus 2", Psychological Review, USA, 1956.

## AUTHOR BIOGRAPHIES

Dr. Séamus McLoone is a member of the distributed interactive applications group (DIAG) at the Department of Electronic Engineering in the National University of Ireland Maynooth. His research interests varies from multiple model approaches to nonlinear system identification to reducing and masking the effects of latency in distributed interactive applications.



Alan Kenny graduated from NUI Maynooth in computer science and software engineering in 2004. Having spent a short stint working as a web developer he returned to the University to study a PhD in the area of psycho-perceptual measures in networked games.

Dr. Tomas Ward is a member of the distributed interactive applications group at the NUI Maynooth. His main interests are in assistive technology and data management techniques for distributed shared applications. Spare time is spent on pushing out his personal best on Konami DrumMania V.

Declan Delaney received his B.E. and M.Eng.Sc. from University College Dublin in 1991 and 1998 respectively. His PhD was awarded in 2005 by the National University of Ireland, Maynooth. From 1992 to 2000 he worked as a software developer and subsequently as project manager. At present he works in the area of Knowledge management for an Italian multinational, ENI and is based in Milan, Italy.