

Exploring an Information Framework for Consistency Maintenance in Distributed Interactive Applications

Xin Zhang, Tomás Ward, Séamus Mcloone

*Department of Electronic Engineering, National University of Ireland Maynooth
National University of Ireland Maynooth
Maynooth, Co, Kildare, Ireland
Email: xzhang@eeng.nuim.ie*

Abstract—Consistency maintenance in Distributed Interactive Applications (DIAs) is subjected to network characteristics such as limited bandwidth and latency. Predictive contract mechanisms are techniques that compensate for the effect of network latency by extrapolating future entity states from historical records. These approaches trade inconsistency within human perceptual limits for reduced network traffic and latency. This paper explores the use of an information metric to analyse the effect of network latency on remote consistency and thus establishes a novel framework to model predictive contract mechanisms as a lossy information sharing process. Such a perspective facilitates a novel explicit analysis of the trade-off between network traffic and inconsistency.

Keywords—consistency; Distributed Interactive Applications; information; Consistency-Throughput Trade-off

I. INTRODUCTION

Modern computer network technology allows geographically distant end-users to exchange information and collaboratively interact with each other in a shared networked virtual environment. Such applications can be described as Distributed Interactive Applications (DIAs). DIAs are widely deployed in various domains such as military simulations (SIMNET [1], DIS [2]), distributed virtual environments (NPSNET [3]) and multiplayer computer games (Quake [4]). Despite the diversity of this application class, one of the most persistent problems has been the issue of maintaining a sufficiently synchronized and dynamic view of the simulation state for all users across the network, i.e. DIAs require some level of consistency.

In a DIA, a virtual entity or object that represents a participant is typically described by a number of state variables, which are synchronized through sending update messages across the network. Consistency has been defined in diverse ways [5]. In the context of this paper, it refers to spatial similarity between the state of an local object and its remote replication. Unfortunately, absolute consistency in DIAs is impossible to achieve because of network latency [6]. In fact, increasing update frequency for finer remote state replication and better consistency leads to more data being transmitted and worse latency if the traffic exceeds the limited network bandwidth, and thus ironically compromises consistency. This “*Consistency-Throughput Trade-*

off” requires DIA designers to employ various consistency maintenance mechanisms to maintain a sufficient level of consistency through optimal use of network resources.

Generally, techniques and approaches used by these mechanisms can be classified into three classes [6]: information management techniques which reduce the amount of data transmitted across the network to optimize the bandwidth usage so that latency can be minimized; time management techniques that manage consistency by manipulating time to mask the effect of network latency and system architecture techniques which seek to improve the efficiency of data processing and dissemination.

One important group of information management techniques is predictive contract mechanisms, which have been widely utilized in military training simulations and computer games [1], [7]. Predictive contract mechanisms compare the actual state of a local entity to the estimate made by some prediction scheme from contextual dynamics. Entity state updates (ESUs) are only generated when the prediction error, i.e. the difference between the actual and estimated states, violates a given threshold. The remote hosts, on receiving ESUs, predict the entity state using the same prediction scheme. In doing so, predictive contract mechanisms maintain a controlled inconsistency for reduced network traffic. There is significant research focused on designing prediction models that better fit entity dynamics [5], [8], [9]. In order to balance the trade-off between the number of update transmissions and the resultant inconsistency, suitable error thresholds must be carefully chosen [10]–[14]. In so far as the authors are aware, there exists no analytical measure of the contribution of the mechanisms in reducing data transmission.

Our previous work on this issue applies information theory to establish a novel framework for predictive contract mechanisms [15]. Mutual information [16] is employed to measure the dependence between the real state dynamic and the approximated state dynamic on the local host. The inconsistency induced by discarding prediction errors within the threshold limit is measured as the information loss in the local approximation. Such a perspective facilitates an analytical study of the trade-off between consistency and

throughput.

In this paper, we extend our previous model [15] by including the remote inconsistency in this information framework. The effect of network latency on fidelity degradation is also measured from an information perspective. This completed information model is a compact representation of key aspects of predictive contract mechanisms, namely entity dynamics, prediction models, threshold and latency. Furthermore the model explicitly shows how tuning the threshold affects the system in the “*Consistency-Throughput Trade-off*”. Analyses of our model are presented through an experimental study.

The remainder of this paper is organized as follows. For the convenience of the reader, fundamental principles of predictive contract mechanisms are outlined in the next section. This is followed by detailed explanations of the complete information model in Section III. Section IV presents our results and discussion from an experimental study perspective. Finally, the paper ends with conclusions and directions for future work in Section V.

II. PREDICTIVE CONTRACT MECHANISMS

Predictive contract mechanisms make use of prediction models to extrapolate entity states and therefore reduce the number of messages transmitted across the network. The operation of these mechanisms is commonly divided into two main components: prediction and convergence [2], [17]. The local and remote hosts in the application apply the same prediction model to generate their own approximations of the entity dynamic. Convergence algorithms are only used on the remote hosts to correct large prediction errors on arrivals of the ESUs, such that more natural motion is obtained.

The prediction algorithms estimate future entity states from existing ESUs. The local host that controls the entity keeps the estimation as a local model of the entity dynamic. The model is compared with the actual motion and whenever the prediction error exceeds a pre-defined threshold, an ESU

containing accurate information about the instantaneous dynamic is generated for further extrapolation. On receiving the ESU, the remote hosts extrapolate the remote model accordingly using the same prediction algorithm until the next ESU arrives. In standard Dead Reckoning and its various extensions, linear and polynomial functions are used as the extrapolation equations [2], [18]. More complicated methods involving statistical learning, such as Kalman filters [9] and Neural Networks [8], have been employed to improve the performance of prediction.

The convergence algorithms define how prediction errors are corrected on the remote hosts on receiving an ESU, so that the remote model looks more natural and smooth. Currently, polynomial equations are the most commonly used convergence algorithms [17], [18]. The convergence is the post-operation taken for the sole purpose of better perceptual experience [17], and is thus independent of ESU generation and bandwidth consumption.

Fig. 1 illustrates the operation of predictive contract mechanisms. Notice that the network latency induces extra transmission error in the remote model and the remote inconsistency may exceed the local threshold. Consider the simulation tick $k = 0$ from when the local model error reaches the threshold h and a new update is generated. The local model is corrected immediately but the remote model keeps diverging from the real dynamic over the period of the network latency of L ticks, because the now out-of-date ESU is still in use on the remote site. The remote inconsistency R_e can be described as in

$$R_e = h + \sum_{k=1}^L \Delta v(k)T_s = h + LT_s\overline{\Delta v}, \quad (1)$$

where T_s is the simulation interval, $\Delta v(k)$ is the difference in velocity between the real dynamic and the remote extrapolation at the simulation tick k , and $\overline{\Delta v}$ is the average difference in velocity over the time of the update transmission. Among these aspects that determine the remote

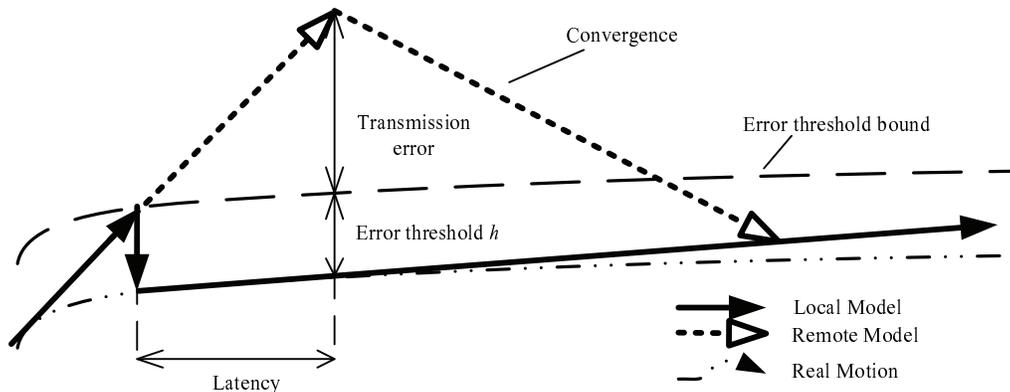


Figure 1. Visual illustration of predictive contract mechanisms. In this case, we use linear extrapolation and first-order convergence where the remote model converges to an extrapolated future point over several steps along a straight line, instead of replacing the predicted value immediately.

inconsistency, the local threshold is deterministic, the network latency is uncontrollable, and the difference in velocity depends on how the prediction model fits the entity dynamic. Therefore, the prediction algorithm is the core of the whole mechanism. Simple or complicated, having closed form equations or not, these algorithms are essentially functions or mappings $f(\cdot)$ that relate the contextual dynamics to the predicted states. More accurate mappings minimize the difference in velocity by making better prediction, and the resultant reduction in update transmission leads to shorter latency, which makes the remote inconsistency more controllable. However there lacks measurements that relate prediction accuracy and the resulting traffic saving.

In the next section, we present a new framework that views predictive contract mechanisms from an information theory perspective. In this framework, the contribution of the prediction model is measured by the ability to make use of the temporal dependence in the entity dynamic. More importantly, given a desired fidelity level, the information metric gives the minimal bandwidth requirement to build the remote model. An incomplete information model that focuses only on local operation was introduced in our earlier work [15]. Here we extend the local information framework by taking latency and remote inconsistency into consideration.

III. INFORMATION MODEL

The information model presented here is based on the idea that ESUs can be used to extrapolate future motion because there is temporal dependence among states at different times, therefore instantaneous entity motion states in the ESUs hold some level of knowledge that can help locate future entity states. In information theory, this dependency or knowledge

is measured by the concept of Mutual Information [16]. A detailed review about the related mathematics and concepts can be found in [15].

The information model redefines the illustration in Fig. 1 into a flow diagram in Fig. 2. Here the entity dynamic yields a discrete time series $d(k) = \{d(1), d(2), \dots\}$, where k is the index of the simulation step and the value of $d(k)$ varies within a finite discrete set of entity state values $S = \{s_i\}$. The remote model $\bar{d}(k) = \{\bar{d}(1), \bar{d}(2), \dots\}$ is state approximation simulated by the remote hosts using predictive contract mechanisms.

The model views predictive contract mechanisms as information processing, in which information about the entity dynamic is included in ESUs at the local host and remotely extracted to reconstruct the remote model. Each ESU, as an information carrier, holds full information about the instantaneous entity state and some information about future states. The full instantaneous information is the entropy $H(d)$ of the dynamic d , which is also the average data rate required for a perfectly accurate approximation where an update $u(k)$ is generated at every simulation tick k and no prediction scheme is applied. The entropy is given as in

$$H(d) = - \sum_{s_i} p_d(s_i) \log p_d(s_i), \quad (2)$$

where $p_d(s_i)$ is the probability function of the entity state.

Extrapolation uses the knowledge about the future states included in ESUs for prediction. Consider predicting the entity state $d(k + \tau)$ from the ESU $u(k)$ generated τ steps earlier. For a certain prediction span τ , the available knowledge or interdependence that can be utilized by prediction schemes for extrapolation is measured by mutual

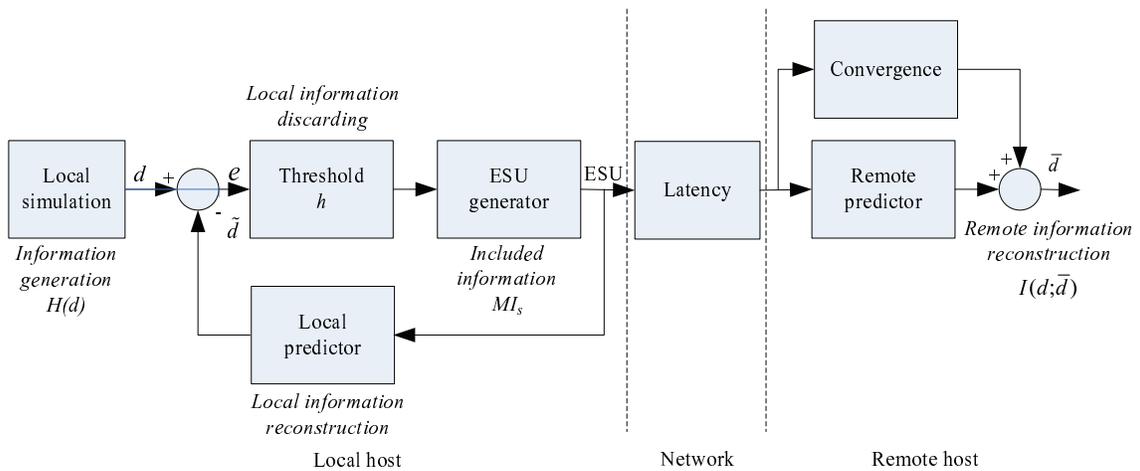


Figure 2. Information model of predictive contract mechanisms. Information generated by the local dynamic d is included with loss into the ESUs, from which the remote model \bar{d} is simulated. The extrapolated dynamic \bar{d} is compared to the real dynamic to decide whether an ESU is needed to correct the prediction error.

information $I(u; d_\tau)$ between $u(k)$ and $d(k + \tau)$, as in

$$\begin{aligned} I(u; d_\tau) &= I(u(k); d(k + \tau)) \\ &= \sum_{u(k), d(k+\tau)} p_{ud}(u(k), d(k + \tau)) \\ &\quad \cdot \log \frac{p_{ud}(u(k), d(k + \tau))}{p_u(u(k))p_d(d(k + \tau))}, \end{aligned} \quad (3)$$

where p_u and p_d are corresponding probability functions of the sequences, and p_{ud} is the joint probability of the two. This mutual information is the average amount of knowledge in an ESU that can be used to predict the entity state τ steps later than its generation. Here we define the functioning period T_f of an ESU as the period during which this ESU is referenced by prediction schemes to generate modelled states. The local ESU functioning period is the time interval between successive ESU generations, and remotely it is between ESU arrivals. On average, the information MI_s stored in an ESU for extrapolating one modelled state is the mutual information averaged over the functioning period.

$$MI_s = \frac{\sum_{\tau \in T_f} I(u; d_\tau)}{T_f} \quad (4)$$

The stored information MI_s in (4) is mostly the characteristic of the entity dynamic in that it characterizes the capacity of the ESU to carry information for extrapolation. An ESU could also have full information about all the future states in deterministic dynamics, or none in cases of completely random motions.

With the information stored in the ESUs, predictive contract mechanisms extrapolate the remote approximation by interpreting that information using extrapolation equations. The mutual information $I(d; \bar{d})$ in (5) reflects the amount of information about the entity dynamic that is actually used in extrapolating the remote model, and thus measures the information utilization. It is then apparent that the information utilization is bounded by the stored information. Given the ESUs, better algorithms could utilize a higher percentage of the stored information to produce more accurate predictions.

$$\begin{aligned} I(d; \bar{d}) &= I(d(k); \bar{d}(k)) \\ &= \sum_{d(k), \bar{d}(k)} p_{d\bar{d}}(d(k), \bar{d}(k)) \\ &\quad \cdot \log \frac{p_{d\bar{d}}(d(k), \bar{d}(k))}{p_d(d(k))p_{\bar{d}}(\bar{d}(k))} \end{aligned} \quad (5)$$

The spatial analysis of the remote inconsistency in (1) can also be redefined from this information perspective. By allowing for local model error within the threshold, prediction contract mechanisms prune the original information generation rate $H(d)$ of the entity dynamic to the storing rate MI_s of the ESUs. The local information loss is controlled by the threshold. The network latency endured

by the transmission adds further information loss and only a reduced information rate $I(d; \bar{d})$ is reconstructed remotely.

In the next section, we apply this model to a practical navigation motion dataset in a virtual environment [19]. The scenario settings are representative of virtual environment interactions.

IV. EXPERIMENTS AND RESULTS

The practical test scenario we use here to illustrate how the proposed framework works is a head dataset taken from the StepWim navigation technique [19], where the experiment subject explores the virtual environment and his/her movement is recorded. The discrete entity state unit is 4 mm. For the convenience of illustration, we only consider the x-coordinate positions as the entity states. Therefore, the motion of the entity is a one-dimensional dynamic and is recorded as a scalar time sequence $d(k) = \{d(1), d(2), \dots\}$. Analysis to higher-dimensional movements can be conducted in a similar manner.

The entity dynamic is sampled at the frequency of 1607 Hz. Our experiments were conducted for varying error thresholds and constant latencies. The extrapolation equation we examine here is linear. At each simulation step k , the linear extrapolation for the current entity state is

$$\tilde{d}(k) = d(k_u) + (k - k_u)v(k_u), \quad (6)$$

where $d(k_u)$ and $v(k_u)$ are the sampled position and velocity included in the latest ESU generated at time-step k_u , i.e. $u(k) = \{d(k), v(k)\}$. For ease of analysis, no convergence algorithm is applied, since user perception is beyond the purpose of this paper.

Fig. 3 shows the actual entity motion, along with the local and remote models. A local error threshold of 32 units and a constant network latency of 60 ms were used. The expanded view shows the delay of remote corrections and transmission error induced by the network latency.

In Fig. 4(a), the information measurement in (3) shows the capacity of the ESUs to carry information about future states for varying prediction spans. An entropy $H(d)$ of 8.8699 bits is given by (2) for the navigation dataset, which is the average information required to reconstruct the entity motion with absolute accuracy. An ESU holds this full information for the immediate state ($\tau = 0$), and only partial information for the future ($\tau > 0$). The decreasing feature of the mutual information indicates that given the knowledge about instantaneous entity motion, the future entity state becomes less predictable as the prediction span increases. By using the incomplete information, predictive contract mechanisms trade consistency for less network traffic. The information carried by an ESU for extrapolating the local and remote models depends on its functioning period. The situation for a threshold of 8 units and a latency of 40 ms is shown in Fig. 4(a). The average ESU functioning period

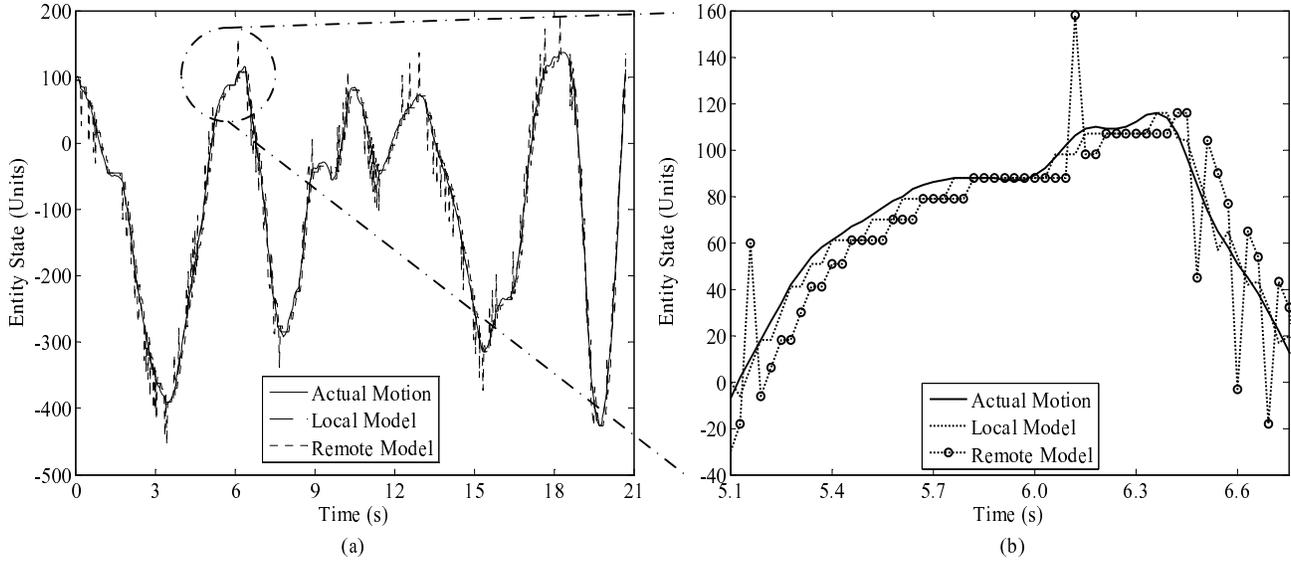


Figure 3. The actual entity motion and the local and remote models for an error threshold of 32 units and a constant latency of 60 ms. A zoomed-in section in (a) is shown in (b).

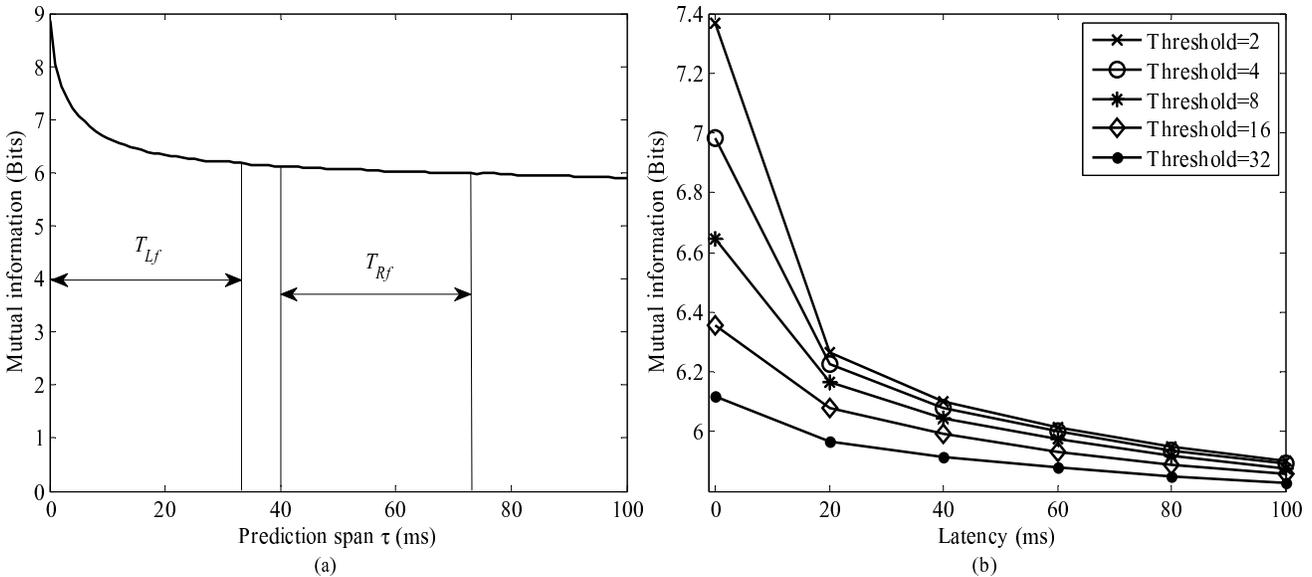


Figure 4. (a) Information capacity of ESU for increasing prediction spans. The local and remote functioning periods T_{Lf} and T_{Rf} are also included. The average ESU functioning period is 33 ms for an error threshold of 8 units and a latency of 40 ms. (b) Average information per simulation tick that is available for extrapolation for different threshold values.

is 33 ms. As in (4), the average included information for varying thresholds and latencies is shown in Fig. 4(b).

The behaviour of the decreasing mutual information measurement highlights the characteristics of the entity dynamic from the perspective of predictability. Application parameters, such as error threshold in the case of dead-reckoning, must be set up accordingly. A quite slower information drop can be expected for entity motions such as a car-racing game, where objects mostly move along their assigned lanes.

In this case, a large error threshold is preferred because the ESUs stay in effect longer without causing much further information loss, and network traffic is reduced. In other scenarios, for example, where players shoot enemies out of random hiding places, mutual information would drop faster, and the ESU functioning period has to be short to include timely information and thus a tight threshold is better. Given the mutual information chart and the network conditions, a proper threshold can be chosen such that

sufficient information is carried by the ESUs to maintain acceptable inconsistency while minimizing network traffic.

The network latency is reflected in the information chart in Fig. 4(a) by the shifting of the remote functioning period. This shifting causes further information loss in ESU because the mutual information decreases as the latency grows.

Average remote inconsistency measured in spatial distance is compared with the mutual information between the remote model and the actual dynamic for varying thresholds and increasing constant latencies in Fig. 5. The mutual information measurement agrees with the inconsistency in that low mutual information comes with high inconsistency, indicating less interdependence between the model and actual motion. More importantly, This also indicates the information rate to maintain such inconsistency. For example, the consistency

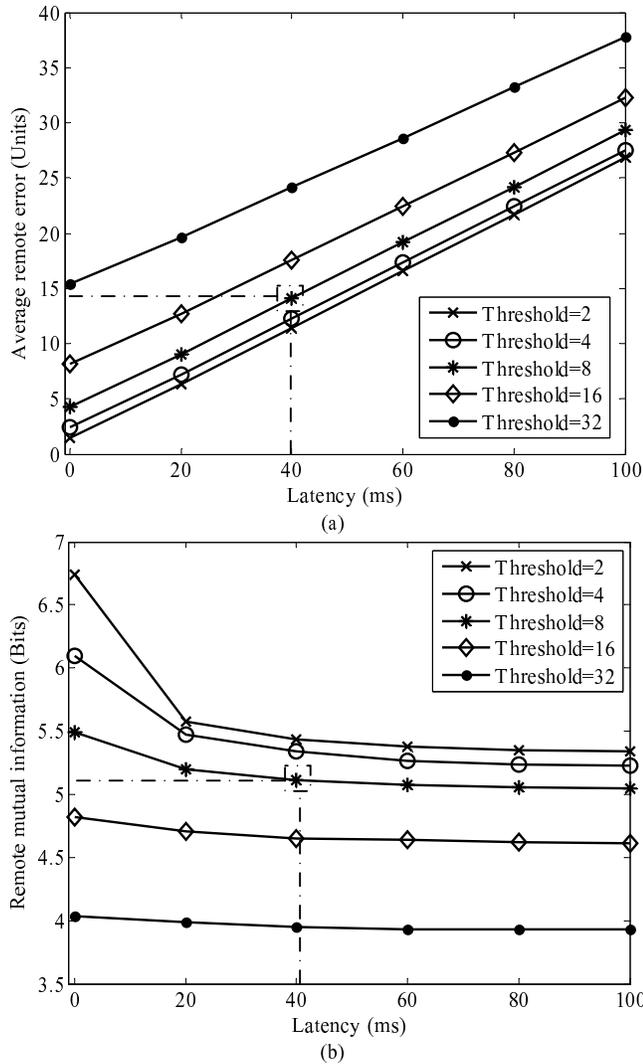


Figure 5. (a) Average remote error for varying thresholds and increasing constant latencies. (b) Mutual information between the remote model and the actual dynamic.

maintenance mechanism used here (linear extrapolation and no convergency) provides an information rate of 5.11 bits per simulation tick, or 8.22 Kbps, to maintain an average remote inconsistency of 14 units for a local error threshold of 8 units and a latency of 40 ms.

The remote model is the result of the extrapolation. The mutual information in Fig. 5(b) measures the amount of knowledge in the remote model, and reflects the amount of information actually utilized by the extrapolation equation. Comparative examination of Fig. 4(b) and Fig. 5(b) reveals that the information included in the ESUs is not completely used by the linear extrapolation. The utilization ratio varies from 60% to 90%, and is lower for larger thresholds. This utilization of included information can be a performance measurement of the extrapolation algorithm. More sophisticated prediction schemes could make better use of the included information and thus improve consistency. However designing such schemes generally requires more prior information about the dynamic, such as training data. Furthermore, such algorithms would be application-sensitive.

Finally, it is true that the information metric is computationally complex. However, information metric can achieve more than just inconsistency measurement as spatial difference. As stated above, the information model reframes predictive contract mechanisms as a information communication process. In addition, the information metric suggests optimal interpretation of the transmitted data. In Fig. 5, the mutual information drops very slowly for larger latencies while the remote error keeps growing rapidly. This is because mutual information measures general relationship between two variables rather than spatial similarity only. This difference is illustrated in Fig. 6, which displays a full dataset of 5 ticks for a trivial simulation used for illustrative

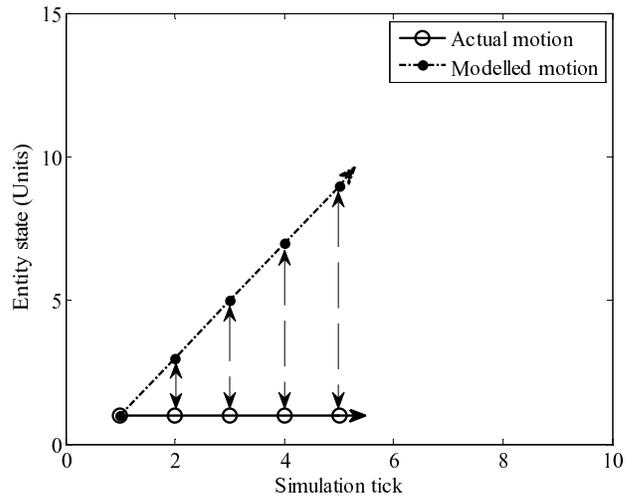


Figure 6. Mutual information measures general interdependence instead of spatial similarity.

purposes. Although the initial extrapolation vector is not perfect and the modelled path diverges from the actual motion, the mutual information is still perfect because the actual motion can be determined with absolute certainty from the modelled path. The modelled path can be transformed to the actual motion with perfect accuracy. Therefore the information metric suggests that there exists a mapping that can transform the results of the linear extrapolation and better interpret the mutual information to present improved consistency. However, designing such a mapping is beyond the scope of this work.

V. CONCLUDING DISCUSSION

This paper explores the use of information theory to reframe predictive contract mechanisms as an information process. In our information model, aspects of consistency maintenance, namely entity dynamics, prediction algorithms, threshold, and network latency are analysed using an integrated information metric. Such analysis facilitates a novel understanding of the “*Consistency-Throughput Trade-off*”, and suggests improvement of application design. Our future work will focus on measuring different extrapolation equations and designing optimal entity state distribution protocol.

ACKNOWLEDGMENT

This work is supported by the Irish Research Council for Science, Engineering and Technology (IRCSET): funded by the National Development Plan.

REFERENCES

- [1] J. Calvin, A. Dickens, B. Gaines, P. Metzger, D. Miller, and D. Owen, “The SIMNET virtual world architecture,” IEEE Virtual Reality Annual International Symposium 1993, Sep. 1993, pp. 450–455, doi:10.1109/VRAIS.1993.380745.
- [2] IEEE, “IEEE standard for distributed interactive simulation—application protocols,” IEEE Std 1278.1a-1998, Aug. 1998.
- [3] M. Capps, D. McGregor, D. Brutzman, and M. Zyda, “NPSNET-V. a new beginning for dynamically extensible virtual environments,” Computer Graphics and Applications, IEEE, vol. 20, no. 5, Sep./Oct. 2000, pp. 12–15, doi:10.1109/38.865873.
- [4] D. Kushner, “The wizardry of id,” Spectrum, IEEE, vol. 39, no. 8, Aug. 2002, pp. 42–47, doi:10.1109/MSPEC.2002.1021943.
- [5] J. D. Delaney, “Latency reduction in distributed interactive applications using hybrid strategy-based models,” Ph.D. dissertation, National University of Ireland Maynooth, Maynooth, Ireland, Nov. 2004.
- [6] D. Delaney, T. Wárd, and S. McLoone, “On consistency and network latency in distributed interactive applications: A survey—part I,” Presence: Teleoperators & Virtual Environments, vol. 15, no. 2, Apr. 2006, pp. 218–234.
- [7] Y. Yu, Z. Li, L. Shi, Y.-C. Chen, and H. Xu, “Network-aware state update for large scale mobile games,” Proc. 16th International Conference on Computer Communications and Networks (ICCCN 2007), Aug. 2007, pp. 563–568.
- [8] A. McCoy, T. Wárd, S. McLoone, and D. Delaney, “Multistep-ahead neural-network predictors for network traffic reduction in distributed interactive applications,” ACM Trans. Model. Comput. Simul., vol. 17, no. 4, Sep. 2007, p. 16, doi:http://doi.acm.org/10.1145/1276927.1276929.
- [9] D. Delaney and T. Wárd, “A java tool for exploring state estimation using the kalman filter,” IEEE Irish Signals and Systems Conference (ISSC), Jun. 2004, pp. 679–684.
- [10] A. Kenny, S. McLoone, T. Wárd, and D. Delaney, “Using user perception to determine suitable error thresholds for dead reckoning in distributed interactive applications,” Irish Signals and Systems Conference 2006, Jun. 2006, pp. 49–54, doi:10.1049/cp:20060412.
- [11] D. Marshall, S. McLoone, D. Delaney, and T. Wárd, “Statistical determination of hybrid threshold parameters for entity state update mechanisms in distributed interactive applications,” Proc. 10th IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT ’06), IEEE Computer Society, Oct. 2006, pp. 85–94, doi:10.1109/DS-RT.2006.36.
- [12] D. Roberts, D. Marshall, S. McLoone, D. Delaney, T. Wárd, and R. Aspin, “Exploring the use of local consistency measures as thresholds for dead reckoning update packet generation,” Proc. 9th IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT ’05), IEEE Computer Society, Oct. 2005, pp. 195–202, doi:http://dx.doi.org/10.1109/DISTRA.2005.21.
- [13] S. Zhou, W. Cai, B.-S. Lee, and S. J. Turner, “Time-space consistency in large-scale distributed virtual environments,” ACM Trans. Model. Comput. Simul., vol. 14, no. 1, Jan. 2004, pp. 31–47, doi:http://doi.acm.org/10.1145/974734.974736.
- [14] B.-S. Lee, W. Cai, S. J. Turner, and L. Chen, “Adaptive dead reckoning algorithms for distributed interactive simulation,” International Journal of Simulation Systems, Science & Technology, vol. 1, no. 1–2, Dec. 2000, pp. 21–34.
- [15] X. Zhang, T. Wárd, and S. McLoone, “Towards an information model of consistency maintenance in distributed interactive applications,” International Journal of Computer Games Technology, vol. 2008, no. 4, Apr. 2008, pp. 1–10, doi:10.1155/2008/371872.
- [16] T. M. Cover and J. A. Thomas, Elements of information theory, 2nd ed., New Jersey, USA: John Wiley & Sons, Inc., Hoboken, Jul. 2006.
- [17] S. Singhal and M. Zyda, Networked Virtual Environments: Design and Implementation, 1st ed., New York: Addison-Wesley, 1999.
- [18] A. B. McCoy, “Data-driven modelling approaches for network traffic reduction in distributed interactive applications,” Ph.D. dissertation, National University of Ireland Maynooth, Maynooth, Ireland, Feb. 2007.

- [19] J. J. LaViola Jr., D. A. Feliz, D. F. Keefe, and R. C. Zeleznik, "Hands-free multi-scale navigation in virtual environments," Proc. 2001 symposium on Interactive 3D graphics (I3D '01), ACM, Mar. 2001, pp. 9–15, doi:<http://doi.acm.org/10.1145/364338.364339>.