Assessing Target Acquisition and Tracking Performance for Complex Moving Targets in the Presence of Latency and Jitter

Andriy Pavlovych and Carl Gutwin
University of Saskatchewan

ABSTRACT

Many modern games and game systems allow for networked remote participation. In such networks latency variability is a commonly encountered factor, but there is still little information available to designers about how human performance changes in the presence of delay. To add to our understanding of performance thresholds for mouse-based tasks that are common in real-time games, we carried out a study of human target acquisition and target tracking in the presence of latency and jitter (variance in latency), for various target velocities and trajectories. Our study indicates critical thresholds at which human performance decreases in the presence of delay. Target acquisition accuracy drops very quickly for latencies over 50 ms and for high velocities. Tracking error, however, is only slightly affected by latency, with deterioration starting at around 110 ms. The effects of latency and target velocity on errors are close to linear, and transverse error is usually smaller than longitudinal error. These results help to quantify the effects of delay on closely-coupled interactive tasks in networked games and real-time groupware systems. They also aid designers in determining when it is critical to improve system parameters and when to apply prediction and delay-compensation algorithms to improve quality of interaction.

Keywords: Latency, jitter, acquisition, tracking, moving targets.

Index Terms: H.5.2. [Information Interfaces and Presentation]: User Interfaces – Theory and methods; evaluation/ methodology.

1 INTRODUCTION

Tracking and acquiring moving targets are common activities in human computer interaction (especially in entertainment applications) and have been studied for quite some time [8, 11, 19, 23, 25]. Many games include animated objects with which the user can interact using a pointing device. Following or clicking on a moving object can change the behaviour of the object (e.g., a player in a soccer game), or may be counted as an achievement in itself (military games where "hits" are important).

Many games are becoming increasingly collaborative, with collaboration typically taking place in replicated clients that share data across a network like the Internet. This organization of collaborative systems introduces the possibility of network delays between the local representations seen by a user, and the 'true' locations of the objects (e.g., as determined by the remote game server, see Figure 1). Although many researchers have looked at various issues in network delay, there is still little evidence about exactly how human performance is affected under different delay conditions and for different tasks. Among the factors present in networked systems are communication latency, jitter (variance in latency), and packet losses. These factors can be minor (e.g., in local wired networks) or major (e.g., in long-distance networks or wireless networks), and may exhibit large variations over time.

Figure 1. Frame from *Counter-Strike: Source*. Wireframe 'hitboxes' show server location of avatar; local representation is lagged.

To provide a better understanding of how delays affect human performance in tracking and acquisition tasks, we carried out a quantitative study that asked people to carry out mouse-based tasks under several simulated delay conditions. We investigated the effects of latency, jitter, target speed, and motion path on tracking accuracy and acquisition accuracy. Our study builds on previous work [22] that looked at controller lag, simple motion paths, and tracking-only motion. In the new study, we consider three new issues that more closely reflect the realities of distributed collaborative interaction: we use remote lag (as would be seen on a real-world network) instead of local controller lag; we use motion paths with additional complexity, both in terms of movement and velocity; and we add target acquisition to the tracking task, which is more common in real-world interactions.

Our study both confirms existing knowledge about how latency and jitter affect interaction in visual workspaces, and also adds new understanding. The study showed that, as seen in previous work, latency starts to affect both target acquisition and tracking noticeably above 100 ms, especially for faster-moving targets, but the effects of jitter are small. The novel results of the study are: that the number of clicks needed to acquire a target increases approximately linearly with target speed; that targets moving at non-constant speed were *easier* to acquire than the same targets moving at a corresponding constant speed; and that for target tracking, participants were good at predicting the target path (relatively low transverse errors), but not the target position on the path (high longitudinal errors).

Our research makes two main contributions. First, we provide new results about critical performance thresholds for tracking and acquisition tasks in several realistic delay scenarios involving multiple target speeds and novel movement paths of varying complexity. These thresholds add important new information to results of previous research, and provide designers of networked systems with understanding about whether different kinds of real-

Email: <firstname.lastname>@usask.ca

Graphics Interface Conference 2012
28-30 May, Toronto, Ontario, Canada

time interaction will be possible under the network conditions predicted for a given game or groupware system. Second, our study identifies several unexpected interactions between performance and the characteristics of the task. For example, our results point to the importance of motion complexity in considerations of the effects of delay, and suggest that users can exploit variations in target speed to reduce tracking and acquisition difficulties as delays increase. Similarly, our results show that jitter can actually allow for improvements in performance, possibly because jitter variations sometimes reduce the effects of latency.

2 BACKGROUND

2.1 Latency and jitter in computing systems

Latency can be defined as the time from when an input device is physically moved to the time a corresponding update appears on the screen, or as the delay in device position updates [9]. Latency is present to some degree in virtually all systems. As an example, detecting a mouse motion with an optical sensor takes a few milliseconds, sending this information to a host computer takes a few more milliseconds or more if network communication is involved; the operating system may need time to "wake up" the application, which has been waiting for input, the processing of the data itself will then require some time, and, finally, displaying the result will require again a bit of time. In a modern desktop computer with an optical mouse all these delays add up to a total end-to-end delay of 30–35 ms [28]. Latencies present in other modern systems range from 67 to 200 ms in gaming consoles [4], and from 20 to 400 ms in wide-area networks, with about 150 ms being the average value [2, 13]. Current round-trip delays across the Atlantic are 100–150 ms between households, with typical jitter of not more than 50 ms RMS[1].

Jitter is defined as variance in latency. More specifically, this is *temporal jitter*, as opposed to other types such as spatial jitter [20]. Ellis *et al.* [6] report that people can detect very small fluctuations in lag, even as low as 16 ms. Hence, when examining system lag, one must also ensure that latency jitter is minimized, or at least known. The most common causes of jitter in human-computer interfaces are related to application algorithms (e.g., when the processing time depends on the current number of objects present in the game scene), to network conditions (e.g., traffic at an intermediate router), or to scheduling schemes for the operating system. The effects of *spatial* jitter on 2D pointing have also been investigated [21], but the experiment described below focuses only on latency jitter.

It has been established that latency and jitter adversely affect human performance in both 2D pointing tasks with stationary targets, in experiments resembling the ISO 9241–9 tapping task [12, 16, 21, 28], as well as in 3D pointing [6, 17, 27, 30]. Latency, for example, manifests itself in larger movement times and correspondingly lower device throughputs, while spatial jitter decreases accuracy. Latency and update rate have both been shown to impact task performance or user response (e.g. [1, 6, 7, 16]). In one study that investigated the effect of network latency in a multiplayer game [20], 50 ms delay was virtually imperceptible, 100 ms was noticeable (and reduced realism slightly), 200 ms was clearly visible, making the play unrealistic, but still possible, and 500 ms delay made the game unusable. A study by Pavlovych *et al*. [21] looked at the effects of latency and spatial jitter on pointing performance in the ISO tapping task. Both latency and spatial jitter reduced the throughput; however, latency levels up to 50 ms showed no negative impact. MacKenzie and Ware [16] looked at latency in pointing and found that 225ms latency results in very high error rates. Our experience also indicates that latencies greater than 200 ms make interaction uncomfortable and unusable for competitive play. In some cases, latency can also cause nausea and simulator sickness in virtual reality setups [3, 5].

Tracking and selection of moving targets have also been studied. Hasan *et al.* [10] evaluate several techniques for selecting moving targets. The novel technique they introduce enhances the targets based on their speed and direction. Effectively, the targets are made larger in the direction of movement. No latency was reported in their system. Pavlovych *et al.* [22] investigate tracking of simple moving targets under the influence of disruptions in the local controller signal (latency, jitter and dropouts). Among their findings were that latency of up to 110 ms, jitter of up to 40 ms, and dropouts of up to 10 % were well tolerated and caused minimal effect on accuracy of tracking. Large amounts of jitter, however, greatly affected fast-moving targets.

Previous work, particularly Pavlovych's previous study [22], was limited in three ways, and these limitations motivate the design of the study described below.

- The previous study [22] considered local controller lag rather than network lag: that is, the visual representation of the cursor was lagged, rather than the visual representation of the target. In real networked interaction, the delay in the remote target will generally be much higher than controller lag, and it is important to determine whether this different type of latency has similar or different effects on performance thresholds.
- The previous work considered only tracking, but acquisition of the target after tracking it is an important part of many scenarios in distributed interaction (e.g., shooting at a moving target). The need to carry out two successive tasks may change people's performance in delayed environments. It has been shown that for tasks involving pointing at static targets, latency has a measurable impact at magnitudes over 50 ms [21]. When tracking moving targets, this thresholds shifts upwards to 110 ms (suggesting that tracking moving targets is more tolerant to the presence of latency). Little is known, however, about the effects of latency on the combined task of moving target acquisition (i.e., tracking and then selecting).
- The previous study looked at simple motion paths, and it is possible that the motion was predictable by participants. The predictability of the paths may partially explain the increase in latency tolerance for tracking (from 50 ms in pointing tasks). In real-world networked interaction, motion paths are complex and unpredictable, and this may change the effect of delay.

The study described below looks at each of these three factors that were not considered in previous work. The current study investigates the *selection* of *moving* targets under the influence of latency and jitter. We do not add spatial jitter or dropouts, but do animate the targets along paths of varying complexity (including variance in both movement and velocity). Spatial jitter was not considered in the study, because it is usually very small for typical pointing devices compared to network delays; dropouts were also left out of the study, because of the small effects seen in previous work [22].

2.2 Characterizing System Latency and Jitter

Before commencing the experiments, we confirmed the latency and jitter properties of our experimental platform. We never observed dropouts in our system.

We used a variation of Mine's method [18] to characterize the end-to-end latency of the mouse used as input device, the computer, the simulation system, and an LCD monitor with a response time of 5 ms. Such latency is comparable to the latency of CRT displays. The baseline latency of our system was

[1] Network latencies quoted here are based on trials collected during 2009–2011 using [13] and other network utilities.

measured at 20±4 ms. We also looked at the baseline jitter behaviour of our system and found it to be negligible, as in [21].

3 EXPERIMENT

The purpose of our study is to investigate how accuracy in tracking and acquisition degrades in the presence of latency and jitter. Once we know the baseline parameters of the system, we can introduce different amounts of these independent factors and observe how the user performance changes for targets moving at different speeds. As described below, we use Lissajous curves to simulate the movements of targets due to their simplicity and ease of analysis, and because they resemble many motions seen in the kinds of tracking and acquisition tasks encountered in games, as they include a mix of mostly straight segments with different corner curves. Unlike earlier studies [16, 21, 22, 28], in this study we delayed the 'remote' target as it was shown to the user, and the click events, by the same amount. The sum of the two delays represents the network round-trip time and is reported in the results. The 'local controller lag' scenario, where the cursor is delayed rather than the target, exists if the lag were only present in the pointing device. While this is possible for some input devices, in this paper we choose to concentrate on the effects of network delay, which needs to be modelled differently.

3.1 Participants

Twelve students from a local university participated in the experiment, with ages ranging from 22 to 32 (mean 26). Six were male, and six female. All were right handed, or otherwise used the mouse with their right hand. The study lasted about 45 minutes.

3.2 Apparatus

The computer was an Intel Core i7-based desktop with a 24″ LCD display, running at a resolution of 1920 × 1080 pixels refreshed at 60 Hz. We used a 800 dpi Logitech optical mouse, connected via USB and polled at 125 Hz. The study software, written in C# and run under Microsoft Windows 7, implemented the task of a moving target that had to be followed with a mouse. The width of the target was 45 pixels (0.55″, 14 mm, or ~1.3 degrees of visual field at 60 cm viewing distance). The target trajectory was based on Lissajous curves with a width of 640 pixels (8″, 20 cm) in each dimension, with four different average frequencies. The individual horizontal and vertical frequencies were chosen in such a way so that the repetition period was substantially longer than the duration of the trial, while each of them was close to the average frequency quoted in the results. The motion of the target was delayed (buffered) for the time associated with the current condition. In situations where latency jitter was present, it was distributed normally. The software logged the target and the mouse trails, the parameters for all of the independent variables, and also computed various error metrics. Storing the mouse and target trails enables post-experiment analysis not foreseen before the experiment. During the experiment the users were able to pause their sessions via the space key, which let the current trial complete and then suspended the program. After a short break the users could resume the study with the same space key.

3.2.1 Motion Paths: Lissajous Curves

We used Lissajous curves for modeling the trajectories of our followed targets. Lissajous curves are graphs of a system of the two parametric equations:

$$x = A \cdot \sin(a \cdot t + \varphi)$$
$$y = B \cdot \sin(b \cdot t)$$

The ratio of the curve frequencies a and b affects the appearance of the curves. For equal frequencies and amplitudes, and a phase of $\pi/2$, the curve becomes a circle. For other ratios, the curve becomes a complex harmonic pattern. We also used variations based on Lissajous curves, where we replaced single sinusoids with two and three with other frequencies – to create the paths of higher complexity (i.e., complexities 2 and 3 in Figure 7). For higher complexity paths we adjusted the parameters in such a way that the amplitude and the average velocity of the target motions would be the same as those for one-sinusoid condition. Some of the patterns used in our experiment are shown in Figure 2.

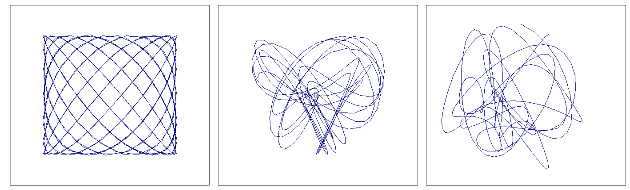

Figure 2. Curves used in the experiment, of complexities 1, 2, and 3. Participants followed *parts* of these curves during the study.

Despite their mathematical simplicity, Lissajous curves mimic a large class of tracking actions. These curves have both slower sections with high curvature and faster ones with low curvature. Also, the spatial frequency of the curves varies, and this makes the curves reasonably similar to real object movements. Real objects obey the laws of physics (such as inertia, where object mass limits acceleration) and conform to the world (real road bends have varying curvature, i.e. varying spatial frequencies). In other words, moving at a constant speed along varying curvatures is difficult and unnatural, and hence, even computer games have to animate objects realistically.

Another argument for Lissajous curves is that for tasks like weapon aiming, the user has to keep the target under the cursor in the center of the screen. Assuming continuous compensation, Lissajous curves are a good approximation to such motions. Finally, such curves artificially constrain the range of action to a square (or rectangular) region. This corresponds to what occurs in many actual scenarios, as the area available for tracking is naturally limited by the boundaries of a display. Previous studies also used Lissajous curves, but only with complexity 1 (Figure 2).

3.3 Procedure

After signing informed consent forms, participants were seated in front of the computer display at a distance of about 0.6 m. Participants were given a brief introduction to the system and were allowed to try the system and find the most comfortable seating position. After that, they were directed to proceed with the task, in which they were instructed to click on the target with a mouse cursor as quickly as possible and with the fewest number of clicks (acquisition phase) and then to follow the moving targets as accurately as possible (tracking phase). Participants were informed about the pause functionality of the program and were encouraged to use it any time they felt the need for a short rest, as the task was highly repetitive and of low entertainment value. None of the participants reported any fatigue at the end of the test. Figure 3 shows a screenshot of the running program.

3.4 Design

The experiment was *within subjects*, and the order in which the combinations of the factors were presented was randomized (without replacement) to compensate for possible asymmetric transfer of learning effects. Similar to [22], we used a pursuit tracking task, where the target moved and had to be followed with the cursor. Unlike in [22], however, there was an acquisition phase where participants also had to click on the target.

3.4.1 Independent Variables

The experiment had three independent variables in a (1+2+3+4) × 4 × 3 = 10 × 4 × 3 arrangement, for a total of 120 combinations:

- *Latency* (round-trip, constant part): 20*, 50, 110, and 170 ms;
- *Jitter* (normally distributed, in addition to the baseline):
  - σ = 0* ms for 20 ms latency,
  - σ = 0, ±20 ms for 50 ms latency,
  - σ = 0, ±20, ±40 ms for 110 ms latency,
  - σ = 0, ±20, ±40, ±60 ms for 170 ms latency;
- *Target speed*: 4, 8, 12, 16 cycles per minute;
- *Path complexity*: 1, 2, 3 harmonic motions.

Although we refer to the fourth factor variable as *speed*, technically, it is *frequency*, measured in cycles per time interval. We will only use term *frequency* in plots and *speed* elsewhere in the text. Four cycles per minute corresponds to about 4 target widths per second *peak* target velocity, or about 2.5 widths *on average*, based on the sinusoidal motion pattern, and the width and amplitude values used in the experiment. Target speed was the average of the two frequencies used to generate the Lissajous path. More precisely, the two frequencies were ±1 Hz relative to the average frequency. The frequencies tested were only half of those in [22]; with the original values the acquisition task was almost impossible to perform. Jitter values, although listed separately, are not truly independent, as they are bound to latency values. The alternative would have been to quantify jitter as a percentage of the latency; however, we chose to use the absolute values, in part because such choice was employed previously [22], and in part due to the small effects of moderate values of jitter in that paper. Finally, in the above list, * denotes the baseline condition, i.e., minimum latency, and no jitter.

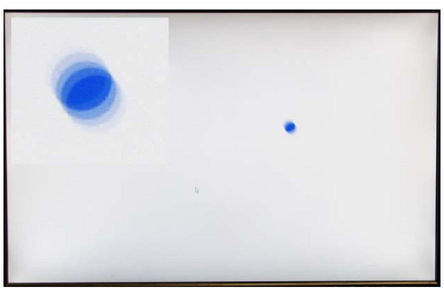

Figure 3. Program screen output. Top left corner shows a magnified view of the target. Target speed is 12 cpm. Blur is due to 1/15 sec. camera exposure time and 60 fps image update time.

The latency and latency jitter values chosen for the experiment are representative of the values common for both local and networked systems. In local environments the values are dominated by input and output device lag and by processing jitter. For systems distributed over the Internet, they are dominated by network properties.

3.4.2 Dependent Variables

We measured three dependent variables: acquisition accuracy and number of clicks to acquire the target (for the acquisition phase), and tracking accuracy (for the tracking phase of the task).

Positional Tracking Accuracy. The error distance (i.e., the instant Euclidean distance between the centre of the target and the tip of the mouse cursor) was divided into two components: a transverse one and a longitudinal one (Figure 4).

The transverse component indicates how far off the *target path* the cursor deviates, and the longitudinal – how far *ahead* or *behind* the cursor is. This separation was done for the following two reasons: the targets are often non-symmetric, but are aligned with the movement direction (e.g., a bus on the road), highlighting the value of computing errors in at least two dimensions; the probability distribution function is normal for both orthogonal components (as was verified subsequently), but it would not be normal for the Euclidean distance, affecting some of the assumptions needed for a statistical analysis. The values were computed over the whole trial, during the one second before the target was successfully acquired, and during each of the subsequent five seconds. These are the values that are reported in this paper.

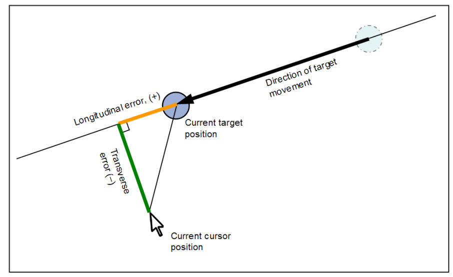

Figure 4. Diagram illustrating the two types of errors measured in the experiment.

Acquisition Accuracy. The second factor studied was the number of clicks required to successfully acquire the moving target. The clicks that missed the targets, preceding the successful one, were also recorded and analysed.

Finally, we computed the values corresponding to cursor deviation from the target position. The notion of longitudinal or transverse direction still applies: the movement vector is defined as the vector from the previous target position to the next one.

4 Results

Each participant completed two sets of 120 rounds, approximately 7 seconds each, with different latencies, target speeds, and path complexities. The duration of the experiment was around 45 minutes, including instructions and short breaks. Given that there were 12 participants, this gave a total of 2×120×12 = 2880 trials. The 7-second duration of the trial was a compromise determined via pilot experiments; it reduces the variability to acceptable levels, while keeping the total duration of the experiment reasonably short.

The error metrics were computed from the target and mouse motion trail logs. For the tracking phase, the errors were averaged along the curve. For the target acquisition phase, the errors were computed at the mouse click positions. The results were analyzed using repeated measures ANOVA. There were significant main effects on the number of clicks and on acquisition and tracking errors for all of the independent variables. Also, there were some interactions between them. We once again note that in this study we delayed the remote targets and the click events, rather than the local cursor. Thus, although we use the same values for some of the experimental variables, the results cannot be directly compared with those from [16, 21, 22], which delayed the local cursor.

4.1 Main Effects on Number of Clicks

4.1.1 Latency

There was a significant main effect of latency on the number of clicks needed to acquire the target, $F_{3,33} = 24.35$, $p < .0001$. According to a Tukey-Kramer test, only the 170 ms latency response was statistically different from the rest. The interaction between latency and target speed was significant, $F_{9,99} = 4.21$, $p < .001$. See Figure 5 for results. No other statistically significant interactions were observed. As with tracking errors in the earlier study [22], the performance degrades more rapidly after 110 ms.

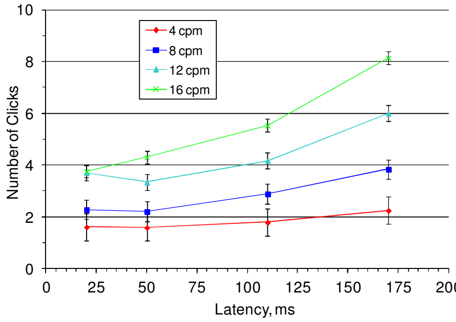

Figure 5. Interaction between latency and target speed on number of clicks.

4.1.2 Jitter

The main effect of jitter on the number of clicks observed only for base latencies of 50 and 110 ms, and no such evidence was seen for 170 ms latency: for 50 ms, $F_{1,11} = 8.06$, $p < .05$; for 110 ms, $F_{2,22} = 5.90$, $p < .01$; for 170 ms, $F_{3,33} = 0.78$, *ns*. Where present, the difference between the means was less than one click. This contrasts with the tracking study [22]: although small amounts of jitter had little effect, at 60 ms it greatly affected tracking accuracy, increasing the tracking error by about 50 %.

4.1.3 Target Speed

The main effect of target speed on number of clicks was significant, $F_{3,33} = 63.79$, $p < .0001$. Speed had an almost linear effect on number of clicks (Figure 6), and from the chart it may be expected that for some very low target velocities the number of clicks will converge to one. Also, there was an interaction between the target speed and path complexity, $F_{6,66} = 10.32$, $p < .0001$. The result resembles the behaviour of the tracking errors in the earlier study [22].

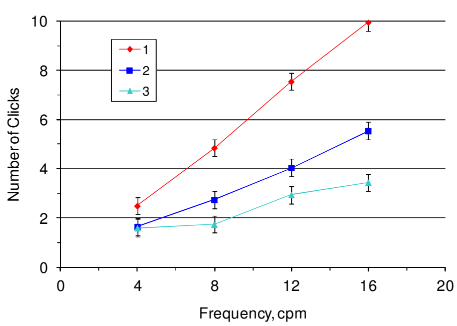

Figure 6. Interaction between target speed and path complexity on number of clicks.

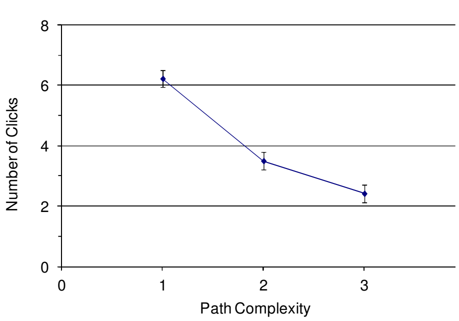

Figure 7. Main effects of path complexity on number of clicks.

4.1.4 Path complexity

The main effect of path complexity on number of clicks was significant, $F_{3,33} = 39.73$, $p < .0001$. Contrary to expectations, more complex paths made the selection easier. This can be attributed to the fact that in more complex paths the speed varies much more than in the simple paths, and this temporary drop in target speed may aid acquisition. See Figure 7 for details.

4.2 Main Effects on Acquisition Error

Here we consider both successful and unsuccessful clicks on targets. As one can observe, "unsuccessful" hits on small targets (i.e., misses) can be "successful" hits for targets of larger size. Thus, one may consider the error distances between the cursor and the target centre irrespective of whether the cursor was within the target boundaries.

4.2.1 Latency

The effect of latency on acquisition error was significant for both transverse and longitudinal errors, $F_{3,33} = 7.86$, $p < .001$, $F_{3,33} = 81.14$, $p < .00001$. While the transverse errors dominate at low latencies, they change very little as latency increases, and at higher latencies it is the longitudinal errors that affect accuracy the most.

The interaction between latency and path complexity was significant for longitudinal errors, $F_{6,66} = 24.64$, $p < .0001$, but not for transverse, $F_{6,66} = 1.35$, $p = .25$. The interaction between latency and target speed for longitudinal errors was also significant, $F_{9,99} = 5.23$, $p < .0001$. No other statistically significant interactions were observed. Figures 8–9 illustrate the results. Here, we have two unexpected effects: the errors change very little for the slowest targets (no threshold effect); and for faster targets they change immediately after 50 ms (the threshold is 50 ms, rather than 110 ms in [22].

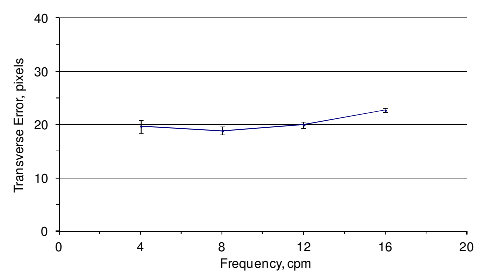

Figure 8. Main effects of latency on transverse error.

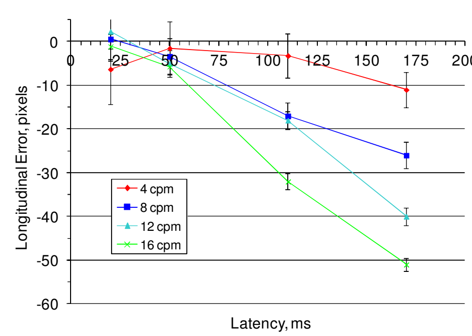

Figure 9. Main effects of latency on longitudinal error[2].

[2] Here and further, negative longitudinal error means that the cursor is behind the moving target.

4.2.2 Jitter

No evidence of the main effect of jitter on acquisition errors was observed for any of the base latencies, where the jitter was present (50, 110, 170 ms). While in [22] the effects of jitter could be observed starting at 60 ms, we do not see the same behaviour here.

4.2.3 Target Speed

The main effect of target speed on acquisition errors was significant for both longitudinal and transverse errors, $F_{3,33} = 11.48$, $p < .00001$; $F_{3,33} = 47.53$, $p < .00001$. Speed had an almost linear effect on errors (see Figure 10). For transverse errors there was an interaction effect of target speed and path complexity $F_{6,66} = 12.34$, $p < .00001$; the paths containing a pair of simple sinusoids was affected by speed increase the most (similar to findings of [22]), while paths composed of multiple superimposed sinusoids had a much less pronounced increase.

Looking at Figure 11, it may be expected that for some very low target velocities the errors stop decreasing with the decrease in velocity. This agrees with an observation that even stationary targets are almost never hit exactly in the centre. The overall drop in accuracy is dominated by an almost linear increase of longitudinal errors with speed.

4.2.4 Path complexity

The main effect of path complexity on acquisition errors was significant, $F_{3,33} = 102.93$, $p < .00001$. Contrary to expectations, more complex paths actually lowered the error rates. This can be attributed to the fact that in more complex paths the speed varies much more than in the simple paths, and this temporary drop in target speed may aid acquisition. Such result is partially visible in Figure 11.

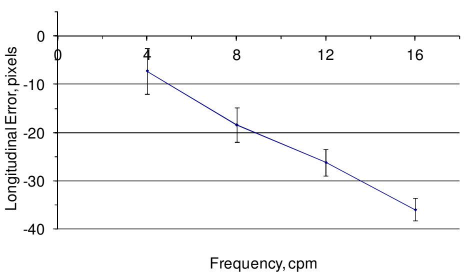

Figure 10. Main effect of target speed on longitudinal errors.

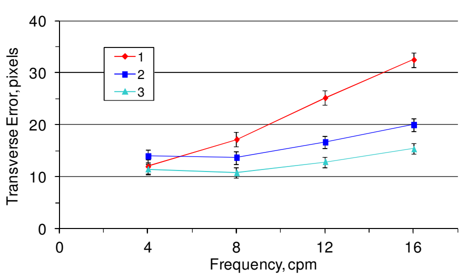

Figure 11. Interaction between path complexity and target speed for transverse errors.

4.3 Main Effects on Tracking Error

4.3.1 Latency, Target Speed, and Path Complexity

The main effect of latency on longitudinal tracking error was significant, $F_{3,33} = 48.72$, $p < .0001$, but not for transverse error, $F_{3,33} = 0.11$, *ns*. According to a Tukey-Kramer test, all pairs were confirmed to be significantly different, except the 20-50 ms pair.

The interaction between latency and target speed was also significant, $F_{9,99} = 7.17$, $p < .0001$. So was the interaction between latency and path complexity, $F_{9,99} = 7.22$, $p < .0001$. No other interaction effects were observed. See Figures 12 and 13.

It can be seen from the graphs that users consistently overpredicted the target motions, and this effect diminishes only towards the highest levels of latency. Small levels of latency are not distinguishable up to 50 ms round-trip time. Following faster targets is affected by latency more, with accuracy starting to drop after 50 ms (the threshold is lower than the 110 ms seen in [22]) . As with selection, higher complexity of the path seems to make the tracking more accurate; possibly due to presense of portions with lower instant speed.

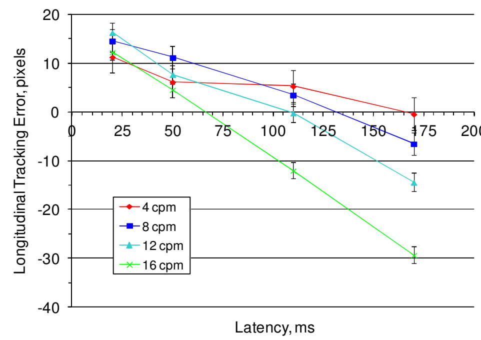

Figure 12. Interaction between target speed and latency for longitudinal errors.

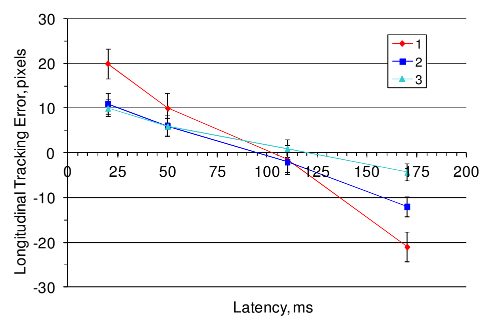

Figure 13. Interaction between path complexity and latency for longitudinal errors.

4.3.2 Jitter

No evidence of the main effect of jitter on tracking errors was observed for any of the base latencies, where the jitter was present (50, 110, 170 ms). In [22] the effects of jitter could be observed starting at 60 ms, and we do not see the same behaviour here.

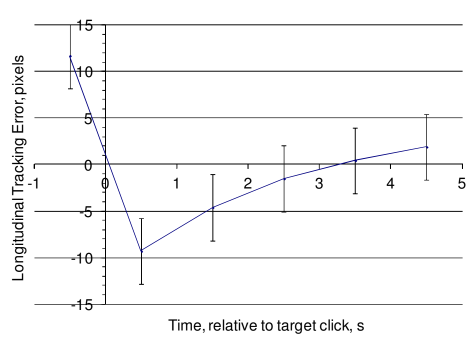

Figure 14. Change of tracking error during trial.

4.3.3 Change of error distance with time

Figure 14 illustrates how the distance between the cursor and the target changes with time, starting with one second-interval before the successful click and ending 5 seconds after the click.

From the graph it can be seen that the users lead the target before the click, most probably are very close to target centre when the click takes place (time = 0), and then momentarily jump behind the target before slowly recovering back to zero-lead. However, the variability of these data is rather high, influenced by lack of any feedback as to the relative position of the cursor relative to the *real* (i.e., non-delayed) target.

4.3.4 Target Speed and Path Complexity

The main effect of target speed on tracking errors was significant only for longitudinal errors, $F_{3,33} = 13.39$, $p < .0001$. According to a Tukey-Kramer test, only the highest speed condition at −13 ms was different from others (at −2.6–3.8 ms). There was also significant interaction between target speed and path complexity, $F_{6,66} = 3.57$, $p < .01$, with higher speeds affecting simpler paths more. No new significant interactions were observed. Figure 15 shows the means of the errors.

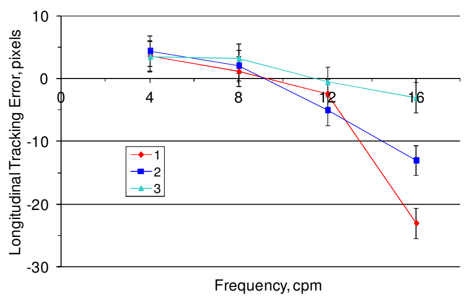

Figure 15. Interaction between path complexity and target speed for longitudinal tracking errors.

5 Discussion

Here we discuss several issues raised during the study, and compare our results to previous experiments. We consider each of the three main ways in which our study extends previous work [22]: performance thresholds for remote latency and jitter, the types and amounts of error under the different types of latency, and the effects of the increased complexity of target motion.

5.1 Latency and Jitter Thresholds

Our study considered remote latency and jitter rather than local controller lag, and added both complex paths and the additional action of selection to the previously-studied tracking task. Our results for the effects of delays on acquisition of moving targets provide some confirmation of previous results, but also add new knowledge. First, effects on tracking error are largely in line with previous results (e.g., [22]): at slightly above 100 ms, suggesting that the different types of delay affect tracking in similar ways. There were some differences in our thresholds, but cases where the effects are weaker can perhaps be attributed to a presence of a secondary task, i.e., acquisition.

Our study adds new information to previous work in providing new results about the acquisition phase of the task – that is, the number of clicks to hit the target also exhibited a threshold effect, with performance declining substantially after 110 ms. Overall, our results suggest that for real-world track-and-select tasks, round-trip latency of less than 100 ms can be tolerated with a moderate decrease in interaction efficiency, while higher latencies lead to a sharp increase in time needed to acquire a moving target.

Our study also revealed an unexpected result regarding the effects of latency on selection accuracy. Latency seems to have little effect when selecting slow targets, but the decrease in accuracy is immediate for targets at speeds above 8 cpm (see Figure 9). We plan to examine this behaviour more closely in future work. Another unexpected result was a much lower, or sometimes totally absent effect of jitter. While in the previous study [22] jitter affected errors at a level of 60 ms, the present study showed no such effect.

5.2 Effects of path complexity

Our study used paths of much higher complexity than previous work, and our results both confirm and extend previous work on interaction with moving targets. The linear increase in the number of erroneous clicks with increasing speed also resembles the linear increase of tracking errors [22] and is similar to effects observed in other studies [24]. However, the decrease in errors with higher path complexity was unexpected. A possible cause of this result is the fact that increased complexity adds variance to both the object's motion and to its velocity; and although the motion is less predictable than with simple paths, there are points along the path where a decrease in speed can be exploited by the user. The average speed was adjusted to be equal for all path complexities, but certain parts of the path (particularly corners) were at a consistently lower speed than straight sections. As lower speed aids acquisition, those slower segments might have influenced performance accordingly. It should be noted, however, that in real-world motion, object speed does decrease when it follows a path with increased curvature, and this behaviour is related to human kinematics (e.g. the Two-Thirds Power Law, [15]) and the laws of physics. This is potentially an important result for game design, since designers may be able to mitigate the increasing difficulty of tracking and acquiring faster targets with the characteristic variability in speed that is exhibited by naturalistic physics-governed motion paths. Had we allowed the speed to vary independently from the path complexity in our experiments, the outcomes would likely have been different; this, however, would not represent the real-world behaviour of the targets.

5.3 Longitudinal vs. Transverse Errors

Previous research [22] has found longitudinal errors to be substantially larger than transverse errors, by a factor of 1.5–2.0 at times. In this study we observe a similar effect (e.g., see Figures 10 and 11). It appears that this difference increases with target speed, and similarly should disappear if the target stops. While the transverse errors dominate at low latencies, they change very little as latency increases, and at higher latencies it is the longitudinal errors that affect accuracy the most. This behaviour is somewhat different from the one observed in [22]: there the longitudinal errors were always larger, even at the lowest speed of 8 cpm. While the motion of the targets was predictable in the previous experiment, it appears that participants' sideways accuracy was better. The users could be ahead or stay behind the target, but at least they were close to the target path. One possible reason for this phenomenon is that participants were able to learn the general form of the motion paths we used, but were still affected by latency. This suggests a clear design principle – that if game targets move, for a given area, the size of the target in the direction of the motion needs to be correspondingly larger.

5.4 Generalizability to other Pointing Devices

Although we investigated only mouse-based input, the findings are likely to generalize to other devices, as latency will affect them similarly. However, the quantitative effects are likely to vary for different classes of devices. For example, direct touch interfaces are affected differently by latency, as one can immediately see where one points and thus adjust the trajectory based on non-delayed direct visual observation. On the other hand, an isometric joystick – a relative pointing device – could be more similar to a mouse than to a direct touch panel. However, these conjectures still need to be examined in the future.

In our study, we used only one target size – mainly to limit the number of independent variables. While some results might still apply to smaller or larger targets (e. g., tracking and acquisition errors), others – like number of clicks to acquire the target, will likely be affected by target size.

6 Conclusions and Future Work

We presented an experimental evaluation of latency, jitter, target speed and target path complexity on moving target acquisition and tracking, tasks that are common to interactive entertainment systems and computer games. Our study shows that all of the investigated factors affect tracking performance. Errors start to increase very quickly at latencies over 110 ms; however jitter does not exhibit the same effect seen in earlier studies investigating pure tracking without acquisition. The effects of target velocity on errors are close to linear, and transverse errors are always smaller in magnitude, compared to longitudinal ones. The results of task complexity are unclear, as the change of complexity was associated with change is speed variance (which is also likely to be the case in real-world situations).

The results can be used to better quantify the effects of different factors on moving objects in interactive scenarios, such as video games. They also enable designers to make better-informed choices for selecting target sizes and velocities, as well as for adjusting smoothing, prediction and compensation algorithms to constrain the overall system characteristics to the values above, in order to avoid large losses in performance.

Acknowledgments

We thank Wolfgang Stuerzlinger for his valuable input on this research. Additionally, we thank our GI reviewers for their helpful comments, and NSERC and the GRAND NCE for funding. Also, our thanks to all participants in the experiment.

References

[1] Allison, R. S., Harris, L. R., Jenkin, M., Jasiobedzka, U., and Zacher, J. E. 2001. Tolerance of Temporal Delay in Virtual Environments. *IEEE VR 2001*, 247–254.

[2] Armitage, G. and Stewart, L. 2004. Limitations of using real-world, public servers to estimate jitter tolerance of first person shooter games. *ACM ACE '04*, 257–262.

[3] Bolia, R. S., Morley, R. M., Nelson, T. W., Roe, M. M. 2000. Assessing Simulator Sickness in a See-Through HMD: Effects of Time Delay, Time on Task, and Task Complexity. *Proceedings of the 2000 IMAGE Conference*, available Online at: http://www.spatiald.wpafb.af.mil/Publications/IMAGE00.pdf

[4] Console Gaming: The Lag Factor. http://www.eurogamer.net/articles/digitalfoundry-lag-factor-article.

[5] Craig, S. J., Reid, L., and Kruk, R. 2000. The effect of visual system time delay on helicopter control. *Proceedings of the IEA 2000/HFES 2000 Congress*, 3-69–3-72.

[6] Ellis, S. R., Breant, F., Manges, B., Jacoby, R., and Adelstein, B. D. 1997. Factors influencing operator interaction with virtual objects viewed via head-mounted see-through displays: viewing conditions and rendering latency. *Virtual Reality 1997*, 138–145.

[7] Ellis, S. R., Young, M. J., Adelstein, B. D., and Ehrlich, S. M. 1999. Discrimination of changes of latency during voluntary hand movements of virtual objects. *Human Factors and Ergonomics Society, 1999*, 1182–1186.

[8] Fekete, J.-D., Elmqvist, N., and Guiard, Y. 2009. Motion-pointing: target selection using elliptical motions. *Proc. CHI '09*. 289–298.

[9] Foxlin, E., Motion tracking requirements and technologies. In *Handbook of virtual environments: Design, implementation and applications*, Lawrence Erlbaum, 2002, 163–210.

[10] Hasan, K., Grossman, T., and Irani, P. 2011. Comet and Target Ghost: Techniques for Selecting Moving Targets. *CHI '11*, 839-848.

[11] Hoffmann, E. R. 1991. Capture of moving targets: a modification of Fitts' Law', *Ergonomics*, 34:2, 211–220.

[12] ISO/DIS 9241–9 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 9: Requirements for non-keyboard input devices. International Standard, International Organization for Standardization, 2000.

[13] Just ping – Online web-based ping: remote ping a server with 40 checkpoints worldwide, www.just-ping.com.

[14] Kulikov, S., MacKenzie, I. S., and Stuerzlinger, W. 2005. Measuring the effective parameters of steering motions. *CHI '05 extended abstracts*, 1569–1572.

[15] Lacquaniti, F., Terzuolo, C., and Viviani, P. 1983. The law relating kinematic and figural aspects of drawing movements. *Acta Psychologica*, 54, 115–130.

[16] MacKenzie, I. S., and Ware, C. 1993. Lag as a determinant of human performance in interactive systems. *ACM CHI 1993*, 488–493.

[17] Meehan, M., Razzaque, S., Whitton, M. C., and Brooks, F. P. 2003. Effect of Latency on Presence in Stressful Virtual Environments. In *IEEE VR 2003*, 141–138.

[18] Mine, M. R., Characterization of End-to-End Delays in Head-Mounted Display Systems. Univ. North Carolina Chapel Hill, 1993.

[19] Mould, D. and Gutwin, C. The effects of feedback on targeting with multiple moving targets. *Graphics Interface 2004*, 25–32.

[20] Pantel, L and Wolf, L. C. 2002. On the impact of delay on real-time multiplayer games. In *Proc. NOSSDAV '02*, 23–29.

[21] Pavlovych, A. and Stuerzlinger, W. 2009. The tradeoff between spatial jitter and latency in pointing tasks. *Proc. EICS '09*, 187–196.

[22] Pavlovych, A. and Stuerzlinger, W. 2011. Target Following Performance in the Presence of Latency, Jitter, and Signal Dropouts, *Graphics Interface 2011*, 33–40.

[23] Port, N. L., Kruse, W., Lee, D., and Georgopoulos, A. P. 2001. Motor Cortical Activity during Interception of Moving Targets. *J. Cognitive Neuroscience* 13, 3, 306–318.

[24] Poulton, E. C. 1952. Perceptual anticipation in tracking with two-pointer and one-pointer displays. *British Journal of Psychology*, 43, 222–229.

[25] Poulton, E. C. *Tracking Skill and Manual Control*. Academic Press (1974).

[26] Savery, C. and Graham, N. 2011. It's about time: confronting latency in the development of groupware systems. *CSCW 2011*, 177-186.

[27] So, R. H. Y., and Chung, G. K. M. 2005. Sensory Motor Responses in Virtual Environments: Studying the Effects of Image Latencies for Target-directed Hand Movement. In *Engineering in Medicine and Biology Society, IEEE-EMBS 2005*, 5006–5008.

[28] Teather, R., Pavlovych, A., Stuerzlinger, W., and MacKenzie, S. 2009. Effects of tracking technology, latency, and spatial jitter on object movement, *IEEE 3DUI 2009*, 43–50.

[29] Tumanov, O., Allison, R., and Stuerzlinger, W. 2007. Variability-Aware Latency Amelioration in Distributed Environments, *IEEE VR 2007*, 123–130.

[30] Ware C., and Balakrishnan, R., Reaching for objects in VR displays: lag and frame rate. *ACM TOCHI* 1, 4, 1994, 331–356.