Game Input with Delay – A Model of the Time Distribution for Selecting a Moving Target with a Mouse

Shengmei Liu and Mark Claypool

Worcester Polytechnic Institute, Worcester, MA 01609, USA
{sliu7,claypool}@wpi.edu

Abstract. Computer game player performance can degrade with delays from both the local system and network. Analytic models and simulations have the potential to enable exploration of player performance with delay as an alternative to time-intensive user studies. This paper presents an analytic model for the distributions of elapsed times for players doing a common game action – selecting a moving target with a mouse with delay – derived from results from prior user studies. We develop and validate our model, then demonstrate the use of our model via simulation, exploring player performance with different game configurations and delays.

Keywords: moving target selection, game player, modeling, Fitts' law

1 Introduction

Computer games require timely responses to player actions in order to provide for an immersive, interactive experience. Unfortunately, computer input hardware and software always have some delay from when a player inputs a game command until the result is processed and rendered on the screen. Delays on most local computer system are at least 20 milliseconds and can range much higher, up to about 250 milliseconds for some platforms and game systems [11]. Games played over a network, such as multiplayer game systems or cloud-based game streaming, have additional delays from the network and server processing as game data has to be transmitted to and from a server. Both local delays and network delays impact the player, degrading both player performance and player Quality of Experience (QoE) as total delay increases [3, 5, 4].

Research that studies delay and games typically conducts user studies with participants playing a game with controlled amounts of delay. While these studies can be effective for ascertaining player performance for specific games [1, 2], and specific game actions [15], they are time-intensive, requiring months of time to design, setup, conduct and analyze. Moreover, user studies often can only gather data over the small range of game and system configurations tested because of the limited number of configuration parameters users can experience during a single test session. Moreover, some societal conditions (e.g., social distancing during a pandemic) can make organizing and executing traditional user studies impossible.

As an alternative approach, analytic models of player performance and simulations of computer games can provide for a broad exploration of the impact of delay on game conditions without costly user studies. However, such an approach can only be effective if it accurately represents player performance. Data from studies that isolate “atoms” of game actions have the potential to provide the foundation for accurate analytic models of player performance, and resultant simulations that use them can help explain and predict the effects of delay for a wide range of games and delay conditions. This paper makes just such a contribution – an analytic model of a game action, based on user studies, validated and then demonstrated through simulation.

We use data gathered from two previous users studies with over 80 people that measured user performance for an atomic game action – selecting a moving target with a mouse. Users played a game that had them select targets that moved around the screen with different speeds and with different amounts of added delay. The studies recorded the elapsed time to select the target, coupled with a player-provided self-rating of skill.

We use the data from these studies in multivariate, multiple regression to derive models of: 1) the expected elapsed times for selecting a moving target with a mouse, and 2) the distribution of the elapsed times. Used together, these models provide an accurate representation of target selection times over a range of delays and target speeds and two levels of player skill and can be used both to predict performance directly from the model (using the expected value), but also put into discrete event simulations (using the distribution of values). We demonstrate the use of the models in analytic analysis and simulations of player performance for basic shooting games to predict the impact of delay for several game and system configurations.

The rest of this paper is organized as follows: Section 2 presents work related to this paper; Section 3 describes the user study datasets; Section 4 details the derivation and validation of our models; Section 5 evaluates game performance using analytic models and simulation; Section 6 describes some limitations of our work; and Section 7 summarizes our conclusions and possible future work.

2 Related Work

This section describes work related to the problem of modeling distributions for the time needed for users to select moving targets with delay.

2.1 Models of User Input

Fitts' law is a seminal work in human-computer interaction and ergonomics that describes the time to select a stationary target based on the target distance and target width [7]. Fitts' law has been shown to be applicable to a variety of conditions and input devices and has been extended to two dimensions [18], making it suitable for modeling target selection with a mouse [23]. However, Fitts' law by itself accounts for neither moving targets nor delay.

Jagacinski et al. [12], Hajri et al. [9], and Hoffmann [10] extended Fitts' law to moving targets, adding target speed to the model. Mackenzie and Ware [19] measured selection time and error rates when selecting stationary targets with delay, and Jota et al. [13] studied target selection and dragging with touch devices with delay. They found a pronounced multiplicative effect between delay and Fitts' Index of Difficulty

(ID) [7], and proposed a modification to Fitts' law that incorporates delay by including an additional term. Teather et al. [24] measured selection time for stationary targets of different sizes and delays and jitter, and confirmed similar findings to MacKenzie and Ware [19] for Fitts' law's computations for ID. Friston et al. [8] confirmed earlier results of Fitts' law for low delay systems and compared their model to others.

The models proposed provide for expected elapsed times and, in some cases, errors, but do not model the distributions of the elapsed times. The latter is needed for simulations where player response times are selected from a range of possible values based on a model and then used in the simulation, as in the case of our work and needed by other simulators.

2.2 Game Actions

An overlapping research area studies the effects of delay on individual (aka *atomic*) game actions.

Raaen and Eg [22] conducted experiments with a simple button and dial interface, letting users adjust delay based on their perceptions. They found users are capable of perceiving even low amounts of delay (around 66 milliseconds). Long and Gutwin [15] studied the effects of delay on intercepting a moving target. They found target speed directly affects the impact of delay, with fast targets affected by delays as low as 50 ms, but slower targets resilient to delays as high as 150 ms. Pavloyvych and Stuerzlinger [21] and Pavloyvych and Gutwin [20] studied target tracking for objects moving along Lissajous curves (smooth curves, with varying turns within the curve). They found tracking errors increase quickly for delays over 110 ms, but the effects of target velocity on errors is close to linear. Long and Gutwin [16] measured selection time for different sized moving targets. They found that the effects of delay are exacerbated by fast target speeds.

In general, while these approaches have helped understand delay and fundamental game actions, they generally have not applied a model to the data gathered, or if they have, the models are for average (expected) values and not the distributions of the values.

3 Datasets

We use two sets of data obtained from prior user studies [6]: *Set-A* and *Set-B*. Each data set was obtained from users playing a game with controlled amounts of delay where the game focused on a single player's action – selecting a moving target with a mouse. Selecting a moving target is an action common to many PC game genres, such as shooters (e.g., *Nuclear Throne*, Vlambeer, 2015 and *Call of Duty*, Activision, 2003).

3.1 Game

Both datasets are obtained from users playing a custom game called *Puck Hunt* that allows for the study of selecting a moving target with controlled amounts of delay. In Puck Hunt, depicted in Figure 1, the user proceeds through a series of short rounds,

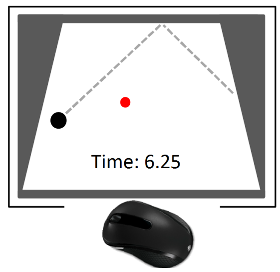

Fig. 1. *Puck Hunt*. Users click on a moving target (the puck) with the mouse cursor (a red ball). The game adds delay to the mouse input and varies the target speed between rounds.

where each round has a large black ball, the puck/target, that moves with kinematics, bouncing off the edges of the screen. The user moves the mouse to control the small red ball (a.k.a., the cursor) and attempts to select the target by moving the ball over the target and clicking the mouse button. Once the user has successfully selected the target, the target disappears and a notification pops up telling the user to prepare for the next round. Thereupon pressing any key, a new round starts, with the target at a new starting location with a new orientation and speed. The user is scored via a timer that counts up from zero at the beginning of each round, stopping when the target is selected. The game settings and parameters were chosen based on pilot tests so as to provide for a range of difficulties – easy to hard. The size of the target is a constant 28 mm, not dissimilar to the size of a "duck" in the video game *Duck Hunt* (Nintendo, 1984).

In dataset Set-A, users select targets with three different speeds (150, 300 and 450 pixels/s) under 11 different delays (100 to 500 ms), with each combination of delay and speed encountered 5 times. In dataset Set-B, users select targets with three different speeds (550, 1100 and 1550 pixels/s) under 11 different delays (20 to 420 ms), with each combination of delay and speed encountered 5 times.

The game records objective measures of performance, including the elapsed time it took for the user to select the target.

3.2 Procedure

All user studies were conducted in dedicated computer labs with computer hardware more than adequate to support the games and LCD monitors.

For each study, participants first completed informed consent and demographic questionnaire forms before starting the game. The demographic questionnaire include the question "rate yourself as a computer gamer" with responses given on a 5 point scale (1 - low to 5 - high). The self-rating question was mandatory. The demographic questionnaire also included an optional gender question with choices for "male", "female", "other" and "prefer not to say" – only one user did not specify either male or female.

Table 3.2 summarizes the major dataset variables, with the bottom row, "Both", showing the users, gender, rounds and conditions for both datasets combined into one.

Table 2 shows the breakdown of self-rated skills for each dataset, with the mean and standard deviation reported by $\bar{x}$ and s, respectively, in the last two columns. The bottom row shows the breakdown of both datasets combined into one. The datasets have a slight skew towards higher skills (mean skill slightly above 3 and mode 4 for each), but there are players of all skill levels in both sets.

Table 1. Summary of dataset variables

Dataset	Usrs	Gender	Rounds	Conditions
Set-A	51	43 ♂, 8 ♀	167	3 speeds, 11 delays
Set-B	32	23 ♂, 8 ♀, 1 ?	167	3 speeds, 11 delays
Both	83	66 ♂, 16 ♀, 1 ?	334	6 speeds, 22 delays

Table 2. Self-rated skill

Dataset	1	2	3	4	5	$\bar{x}$	s
Set-A	1	3	5	24	18	4.1	0.9
Set-B	4	2	9	8	9	3.5	1.3
Both	5	5	14	32	27	3.9	1.1

4 Modeling Selection Time

This section describes our methods used to: a) process and analyze the user study data, and then, b) derive models for the distribution of elapsed times to select moving targets with delay.

4.1 Pre-processing

In Puck Hunt, if the user's time to select the target surpasses 30 seconds, the round ends, and the elapsed time for that round is recorded as a 30. These 30 second values are not the elapsed times that would have been recorded if the trial continued, and so artificially impact any model of selection time that includes them. Thus, we look to replace values of exactly 30 seconds with estimates of the larger values they likely would have had if the round had continued (and the user had kept trying) until the target was selected. In total, the game has 33 different combinations of speed and delay, called *difficulty levels*. For most difficulty levels, there are no elapsed times of 30 seconds. However, the higher difficulty levels (speeds above 500 px/s, and delays above 120 ms) have one or more 30 second values. We replace these 30 second values with randomly generated points above 30 seconds using previously derived models [6]. See our technical report for details [14].

Before modeling, we standardized the delay and target speed by subtracting the means and dividing by the standard deviations. The mean value for delay is 206 ms and the standard deviation is 122 ms. The mean value for target speed is 683 px/s and the standard deviation is 488 px/s.

The distribution of all elapsed times appears log-normal, which makes sense since human actions are impacted by many individual factors that, when put together, have an exponential distribution. We take the natural logarithm of the elapsed time to obtain a probability distribution that appears normal. The probability distribution of elapsed time at each difficulty level follows this pattern, too.

4.2 Modeling

We use multivariate, multiple regression to model the mean and standard deviation of the natural logarithm of the elapsed time. This will allow generation of distributions of elapsed times (using a normal distribution, then taking the exponent) for a given difficulty level, making it usable both for analytically modeling and for simulating game performance.

There are many possible models that fit the elapsed time data. We compare different regression models by using the coefficient of determination (the adjusted R^2) as a measurement of how well observed outcomes are replicated by the model. We fit a model with terms for delay, speed and a combined delay-speed term that is the most parsimonious in providing the desired prediction with as few terms as possible – for both mean and standard deviation. See our technical report for details on the models compared [14].

Our models for mean and standard deviation for the natural log of the elapsed time are:

$$\begin{aligned} mean(\ ln(T)\ ) &= k_1 + k_2 d + k_3 s + k_4 ds \\ stddev(\ ln(T)\ ) &= k_5 + k_6 d + k_7 s + k_8 ds \end{aligned} \tag{1}$$

where the k parameters (e.g., k_1) are constants, d is the standardized delay ($d = \frac{D-206}{122}$) and s is the standardized speed ($s = \frac{S-683}{488}$).

Using Equation 1 on our standardized user study data yields an adjusted R^2 for mean and standard deviation of 0.96 for both. The final model for the mean and standard deviation of the natural log of elapsed time is:

$$\begin{aligned} mean(\ ln(T)\ ) &= 0.685 + 0.506d + 0.605s + 0.196ds \\ stddev(\ ln(T)\ ) &= 0.567 + 0.126d + 0.225s + 0.029ds \end{aligned} \tag{2}$$

With the models predicting mean and standard deviation for $ln(T)$, given speed and delay, the normal distribution with the predicted mean and standard deviation can be used to generate a distribution of logarithmic elapsed times and taking the exponent to get the elapsed times. Our model has excellent fit for the data with R^2 of 0.99 and root mean square error (RMSE) of 0.03.

4.3 Player Skills

Before modeling player skill, we assess if there is a difference in performance based on self-rated skills.

In the user studies, players rated their skills as computer gamers from 1 (low) to 5 (high). The mean self-rating is about 3.9, showing a slight skew towards having “high ability”. Based on our user sample, we divided users into low skill (24 users with self-rating 1-3) and high skill (59 users with self-rating 4-5).

Since the games and test conditions are slightly different between the two user studies, we normalize the data based on the average performance of all users in the same

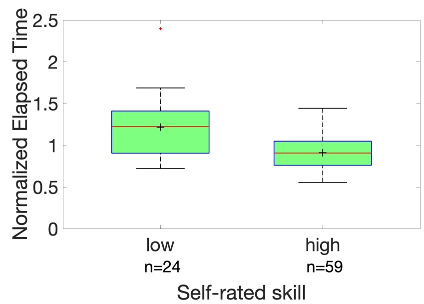

Fig. 2. Combined skill groups

dataset. For example, since the average elapsed time to select a target across all users and all trials for the Set-B dataset is 1.6 seconds, each individual user in the Set-B dataset has their elapsed times divided by 1.6. Users with normalized values below 1 are better than average and values above 1 are worse than average – e.g., a normalized score of 0.9 is 10% better than the average while a 2.0 is twice as bad as the average.

Figure 2 shows boxplots of normalized elapsed time on the y-axis for users clustered by skill group on the x-axis. Each box depicts quartiles and median with the mean shown with a '+'. Points higher or lower than $1.4 \times$ the inter-quartile range are outliers, depicted by the dots. The whiskers span from the minimum non-outlier to the maximum non-outlier. The x-axis "n=" labels indicate the number of participants in each skill group. From the figure, the mean and median of normalized elapsed times decrease (improve) with self-rated skill.

Since the elapsed time data is skewed right, comparisons of the two skill groups was done using a Mann-Whitney U test – a non-parametric test of the null hypothesis that it is equally likely that a randomly selected value from one group will be greater than or less than a randomly selected value from a second group. Effectively, this tests whether two independent self-rated skill group samples come from populations having the same distribution. The test results indicate that the elapsed time was larger for low skill players ($median = 1.22$) than for high skill players ($median = 0.91$), with $U = 321$, and $p < .001$.

Since self-rated skill appears to generally differentiate performance, we derive models for original elapsed time $ln(T)$ parameterized by skill as we did for all users before. The final models are shown in Table 3.

4.4 Validation

To validate our model, we randomly select 20% of data from each difficulty level (combination of target speed and delay) for validation, and train a model on the remaining 80% of the data. Figure 3 depicts the resulting model fit to the training data, showing the elapsed time on the x-axis and the distribution of values on the y-axis. The data are shown with separate colors (blue and pink), with the purple color showing data overlap. The model fits the training data well with an R^2 of 0.99 and a RMSE of 0.03. Figure 4

Table 3. Models of moving target selection time with delay

Skill	Model	Adjusted R^2
All	$mean(\ ln(T)\ ) = 0.685 + 0.506d + 0.605s + 0.196ds$	0.96
	$stddev(\ ln(T)\ ) = 0.567 + 0.126d + 0.225s + 0.029ds$	0.96
Low	$mean(\ ln(T)\ ) = 0.850 + 0.560d + 0.672s + 0.212ds$	0.96
	$stddev(\ ln(T)\ ) = 0.589 + 0.118d + 0.253s + 0.009ds$	0.88
High	$mean(\ ln(T)\ ) = 0.605 + 0.468d + 0.625s + 0.208ds$	0.95
	$stddev(\ ln(T)\ ) = 0.539 + 0.109d + 0.227s + 0.041ds$	0.96

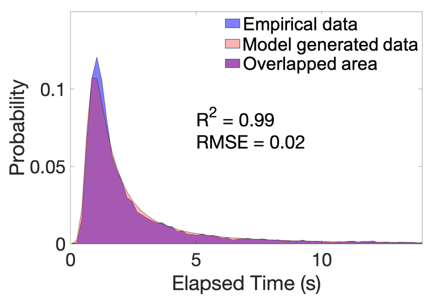

Fig. 3. Training

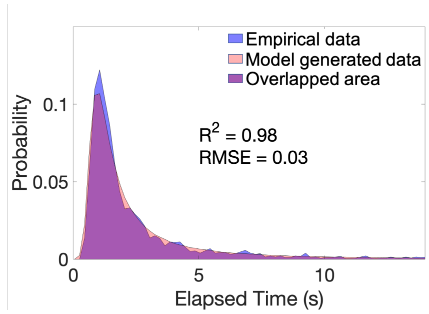

Fig. 4. Validation

depicts the trained model fit to the validation data. The model also fits the validation data well with R^2 at 0.98 and RMSE at 0.03. For the rest of this paper, we use the models trained on all of the data (Table 3) for our analysis and simulations.

5 Evaluation

As a demonstration of our model's use, we use the model to analytically model and simulate player performance for some game configurations. This allows us to explore the impact of game and system configuration on player performance over a range of delays not studied by the previous user studies.

5.1 Player Performance versus Delay

We begin by comparing the impact of delay on performance for target selection compared to reaction time actions where a player responds immediately (e.g., by a key press or mouse click) to an event in the game. Using our model, selection time performance with delay is modeled analytically by $\frac{T(0)}{T(n)}$, where $T(n)$ is the mean elapsed time with delay at an average target speed at 450 px/s, calculated with our model. Reaction time actions are similarly modeled, but using the mathematical response time derivation by Ma et al. [17].

Figure 5 depicts the effects of delay on player performance for both actions. The x-axis is delay, in milliseconds, with a "0" representing the ideal case, and the y-axis

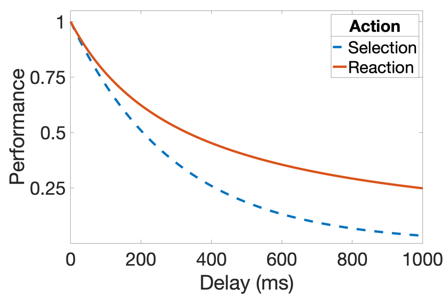

Fig. 5. Player performance versus delay

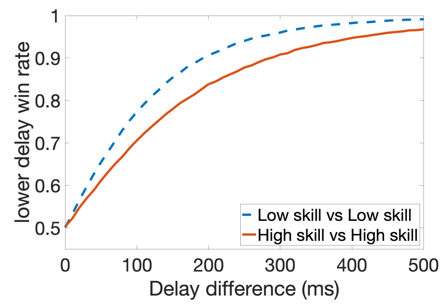

Fig. 6. Win rate versus delay – Skill

the normalized performance with a "1" representing performance in the best case (0 delay). The dashed blue line shows how player performance selecting a 450 px/s target decays with an increase in delay. The solid orange line shows how player reaction time performance decays with an increase in delay. From the figure, both actions have degraded by about 25% at a modest delay of 100 ms. However, the dashed blue line has a steeper decreasing trend than the solid orange line, indicating that delay has more impact on selection actions than on reaction actions.

5.2 Win Rate versus Delay – Skill

We simulate a shooter game where two players try to select (shoot) a target before their opponent. Both players have an equal base delay, but one player has extra network delay to evaluate the effects of unequal amounts of delay on matches with players of equal skill. We simulate 100k iterations of the game for each combination of added delay and player skill. Figure 6 depicts the results. The x-axis is the delay difference for the two players, and the y-axis is the win rate of the player with lower delay. The dashed blue line is games with two low skill players, while the solid orange line is games with two high skill players. From the figure, even a modest delay difference of 100 ms makes it about 50% harder for the delayed player to win. The dashed blue line increases faster than the solid orange line, indicating delay impacts games with higher skill players less.

5.3 Win Rate versus Delay – Target Speed

Figure 7 depicts how delay impacts win rate with different target speeds. This simulation is the same as before, but both players are of average skill and the target speed varies for each game. The x-axis is the delay difference for the two players, and the y-axis is the win rate of the player with lower delay. The dashed blue line is games with slow target speeds of 150 px/s, while the solid orange line is games with fast target speeds of 1150 px/s. The solid orange line increases faster than the dashed blue line, indicating delay impacts player performance more for higher target speeds (i.e., harder games).

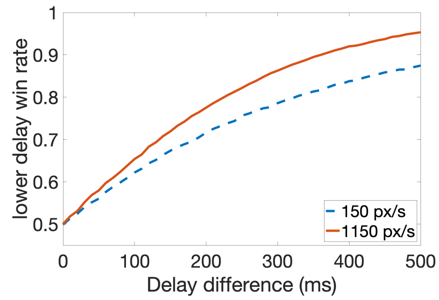

Fig. 7. Win rate versus delay for difference target speeds

6 Limitations

The model is limited by the details recorded and varied by the underlying user studies. As such, the model is only accurate over the range of target speeds and delays tested. For example, this means the model may not be accurate in extrapolating results to very low delays, such as might be encountered in future high end gaming systems, nor may it well-represent extremely small or extremely fast targets.

While target selection is a common action in both 2D and 3D games, the user studies were for a 2D game only so the model may not be accurate in 3D where perspective (and, hence target sizes and on-screen speeds) may change with camera placement.

The self-rated skill is a coarse measure of player experience and a more detailed, multi-question self-assessment may provide for more accuracy in models of performance versus skill.

The studies including 66 males, but only 16 females and the vast majority are relatively young. Gamer demographics are much more evenly split across gender and more widely distributed by age.

a

7 Conclusion

With the growth in networking and cloud services, computer games are increasingly hosted on servers, adding significant delay between user input and rendered results. Understanding the impact of delay on game input can help build systems that better deal with the avoidable delays.

While user studies are effective for measuring the effects of delay on player performance, they are time intensive and, by necessity, typically have a limited range of game parameters they can test. Analytic models and simulations that mimic the behavior of players in games can complement user studies, providing for a broader range of evaluation. This approach is most effective if the game modeled and simulated incorporates observed user behavior.

This work leverages data gathered from two previous user studies [6] to build a model that can be used for just such approaches – analytic models and simulations.

Eighty-three users playing 334 rounds of a game provided data on the time to select a moving target, the target moving with 6 different speeds and the mouse input subjected to 22 different amounts of delay.

We use multivariate, multiple regression to derive a model for the mean and standard deviation of the natural log of the elapsed time, with additive linear terms for delay and speed and a multiplicative interaction term. The same approach is used to model elapsed time distributions based on self-rated user skill – low (self-rated skill 1-3) and high (self-rated skill 4-5). Our derived models have an excellent fit (R^2 around 0.98 and low root mean square error) and can be used to generate a normal distribution of logarithmic elapsed times, then expanded to elapsed time by taking the exponential.

In addition to the main contribution of a model for the distribution of target selection times with delay, we demonstrate use of the model by analytically modeling and simulating player performance in a game that features target selection. The analytic modeling and simulation results show that for the evaluated game:

1 Even delay differences of only 100 ms make it about 50% harder to win.
2 High skill players are less affected by delay than low skill players.
3 Delay affects players more for faster targets.

There are several areas of potential future work.

While the models presented in this paper provide insights into a meaningful and fundamental measure of performance – the elapsed time to select – another measure of performance for selection is accuracy. Future work could model accuracy in a manner similar to the elapsed time models in this work, possibly being combined into a single model of performance.

The atomic action of moving target selection is only one of many fundamental game actions that are affected by delay. Future work could design and conduct user studies to gather data on the effects of delay on other atomic actions, for example, navigation. Such studies could provide data for additional models on these atomic game actions, allowing for richer simulations that can predict the effects of delay for a broader set of games.

References

1. Amin, R., Jackson, F., Gilbert, J.E., Martin, J., Shaw, T.: Assessing the Impact of Latency and Jitter on the Perceived Quality of Call of Duty Modern Warfare 2. In: Proceedings of HCI – Users and Contexts of Use. pp. 97–106. Las Vegas, NV, USA (Jul 2013)
2. Armitage, G.: An Experimental Estimation of Latency Sensitivity in Multiplayer Quake 3. In: Proceedings of the 11th IEEE International Conference on Networks (ICON). Sydney, Australia (Sep 2003)
3. Chen, D.Y., dn Kuan-Ta Chen, H.T.Y.: Dude, the Source of Lags is on Your Computer. In: Proceedings of ACM Network and System Support for Games Workshop (NetGames). Denver, CO, USA (Dec 2013)
4. Chen, K.T., Chang, Y.C., Hsu, H.J., Chen, D.Y., Huang, C.Y., Hsu, C.H.: On the Quality of Service of Cloud Gaming Systems. IEEE Transactions on Multimedia **26**(2) (Feb 2014)
5. Claypool, M., Claypool, K.: Latency and Player Actions in Online Games. Communications of the ACM **49**(11) (Nov 2006)

6. Claypool, M., Eg, R., Raaen, K.: Modeling User Performance for Moving Target Selection with a Delayed Mouse. In: Proceedings of the 23rd International Conference on MultiMedia Modeling (MMM). Reykjavik, Iceland (Jan 2017)
7. Fitts, P.M.: The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. Journal of Experimental Psychology **47**(6), 381–391 (Jun 1954)
8. Friston, S., Karlström, P., Steed, A.: The Effects of Low Latency on Pointing and Steering Tasks. IEEE Transactions on Visualization and Computer Graphics **22**(5), 1605–1615 (May 2016)
9. Hajri, A.A., Fels, S., Miller, G., Ilich, M.: Moving Target Selection in 2D Graphical User Interfaces. In: Proceeding of IFIP TC Human-Computer Interaction (INTERACT). Lisbon, Portugal (Sep 2011)
10. Hoffmann, E.: Capture of Moving Targets: A Modification of Fitts' Law. Ergonomics **34**(2), 211–220 (1991)
11. Ivkovic, Z., Stavness, I., Gutwin, C., Sutcliffe, S.: Quantifying and Mitigating the Negative Effects of Local Latencies on Aiming in 3D Shooter Games. In: Proceedings of the ACM CHI Human Factors in Computing Systems. pp. 135–144. Seoul, Korea (2015)
12. Jagacinski, R., Repperger, D., Ward, S., Moran, M.: A Test of Fitts' Law with Moving Targets. The Journal of Human Factors and Ergonomics Society **22**(2), 225–233 (Apr 1980)
13. Jota, R., Ng, A., Dietz, P., Wigdor, D.: How Fast is Fast Enough?: A Study of the Effects of Latency in Direct-touch Pointing Tasks. In: Proceedings of the ACM CHI Human Factors in Computing Systems. Paris, France (2013)
14. Liu, S., Claypool, M.: Game Input with Delay - A Model for the Time to Select a Moving Target with a Mouse. Tech. Rep. WPI-CS-TR-20-05, Computer Science Department at Worcester Polytechnic Institute (Jul 2020)
15. Long, M., Gutwin, C.: Characterizing and Modeling the Effects of Local Latency on Game Performance and Experience. In: Proceedings of the ACM Symposium on Computer-Human Interaction in Play (CHI Play). Melbourne, VC, Australia (2018)
16. Long, M., Gutwin, C.: Effects of Local Latency on Game Pointing Devices and Game Pointing Tasks. In: Proceedings of the ACM CHI Human Factors in Computing Systems. Glasgow, Scotland, UK (May 2019)
17. Ma, T., Holden, J., Serota, R.A.: Distribution of Human Response Times. Complexity **21**(6), 61–69 (2013)
18. MacKenzie, I.S., Buxton, W.: Extending Fitts' Law to Two-Dimensional Tasks. In: Proceedings of the ACM CHI Conference on Human Factors in Computing Systems. pp. 219 – 226. Monterey, CA, USA (May 1992)
19. MacKenzie, I.S., Ware, C.: Lag As a Determinant of Human Performance in Interactive Systems. In: Proceedings of ACM CHI Human Factors in Computing Systems (1993)
20. Pavlovych, A., Gutwin, C.: Assessing Target Acquisition and Tracking Performance for Complex Moving Targets in the Presence of Latency and Jitter. In: Proceedings of Graphics Interface. Toronto, ON, Canada (May 2012)
21. Pavlovych, A., Stuerzlinger, W.: Target Following Performance in the Presence of Latency, Jitter, and Signal Dropouts. In: Proceedings of Graphics Interface. pp. 33–40. St. John's, NL, Canada (May 2011)
22. Raaen, K., Eg, R.: Instantaneous Human-Computer Interactions: Button Causes and Screen Effects. In: Proceedings of the HCI International Conf. Los Angeles, CA, USA (Aug 2015)
23. Soukoreff, R.W., MacKenzie, I.S.: Towards a Standard for Pointing Device Evaluation – Perspectives on 27 Years of Fitts' Law Research in HCI. Elsevier International Journal of Human-Computer Studies **61**(6), 751 – 789 (2004)
24. Teather, R.J., Pavlovych, A., Stuerzlinger, W., MacKenzie, I.S.: Effects of Tracking Technology, Latency, and Spatial Jitter on Object Movement. In: Proceedings of the IEEE Symposium on 3D User Interfaces. pp. 43–50. Lafayette, LA, USA (2009)