Frame Perfect Probability - A new John

Sjokoladepapir

October 2017

1 Introduction

There has been a lot of talk of dash back, UCF and polling the last couple of months. This have inspired me to look into what frame perfect actually means, and if there can be more to it than just timing. I came up with a theory and probability model, describing how a frame perfect input is handled if certain conditions is set.

2 Theoretical background

Read up on polling and synced polling on Kadanos website about input lag [1]. Some probability theory wouldn't hurt.

Everything described here is only true if the polling is synced. In this post I will use something i call "Perfect polling", which is a simplified version of synced polling. What this means is that every input in one frame will be updated within that frame. The only difference between a input for perfect polling and realistic synced polling, is some minor lag, and probably some weird stuff happening if one hits two inputs at once, at a poll.

3 Theory

A player tries to do a frame perfect input, without knowledge of what's happening on the screen, sound or any other feedback from the game.

This could be a frame perfect wavedash straight down. While holding the gray stick down, input A would be jump (x/y), and input B would be air dodge (L/R). Marth and Fox have different frames they need to do input B on, but the mechanism is the same for every frame perfect input. In fact every input that needs one frame of precision works the same way. Therefore the model used can be simplified to: Input A happens in frame 1 (and will always happen in frame 1). Input B needs to be within frame 2.

Because our blind and deaf player have no feedback from the game, the input of A will land anywhere within frame 1.

4 Math

The timing between input A and input B can be described as $B - A = \Delta t$, where Δt is describing the time between input A and input B in frames. In Smash one frame lasts 0,01666 seconds, so $\frac{1}{60}s = 1F$, where s is second and F is a frame unit. Frame 1 starts at 0F and ends at 1F, frame 2 starts at 1F and ends at 2F, and so on.

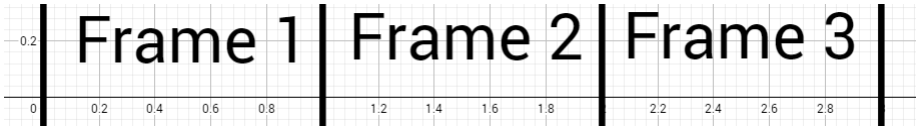

Figure 1: Frames 1, 2 and 3, in a frame unit domain.

If A starts in 0.5F and B in 1.5F then there is one frame unit between, or $1.5F - 0.5F = 1F$.

4.1 The extremes

Consider $\Delta t = 1F$.
Input A is always within frame 1, but since the player doesn't know where in the frame he makes the input, the whole frame needs to be considered as a starting point. $0F \leq A < 1F$ because of this. Since B is one frame after A then $1F < B < 2F$, hence the input will always be a frame perfect input, independent of where in the frame the player inputs A.

Consider $\Delta t = 2F$. In this case B will be shifted one frame ahead, compared to $\Delta t = 1F$. What this means is that $2F < B < 3F$, which always is outside of frame 2, and a input will never be frame perfect.

Consider $\Delta t = 0F$. What this means is that A and B is pressed at the same time, resulting in B always being polled at frame 1. This input will never be frame perfect.

4.2 The in between

Now that the extremes have been defined, some tests for the probability of getting a successful input between $\Delta t = 0F$ and $\Delta t = 2F$, is studied to find out how the continuous probability distribution looks like. This part is somewhat harder to explain, and therefore to harder understand, but bear with me.

If $\Delta t = 0.5F$, then $0 \leq A < 1F$ still holds true, but B will be both in frame 1 and 2, specifically $0.5F < B < 1.5F$. What this means is that there is a 50/50

chance of getting the input right, because only half of the input lands within frame 2 ($1F \leq B \leq 1.5F$).

If $\Delta t = 0.1F$, then $0.1F < B < 1.1F$, B is only a frame perfect input for $1F < B < 1.1F$, hence there is a 10% probability that the input will read as frame perfect.

4.3 Triangular Distribution

Using $\Delta t = 1.5F$ and $\Delta t = 1.9F$, one can see that it a mirroring function. Since Δt correlates to the probability of getting a frame perfect input linearly, the probability distribution will be a triangular distribution. The formula for the probability distribution function will be

$$f(\Delta t) = \begin{cases} 0 & \text{for } \Delta t < 0F, \\ \Delta t & \text{for } 0F \leq \Delta t < 1F, \\ 1 & \text{for } \Delta t = 1F, \\ 2F - \Delta t & \text{for } 1F < \Delta t \leq 2F, \\ 0 & \text{for } 2F < \Delta t. \end{cases}$$

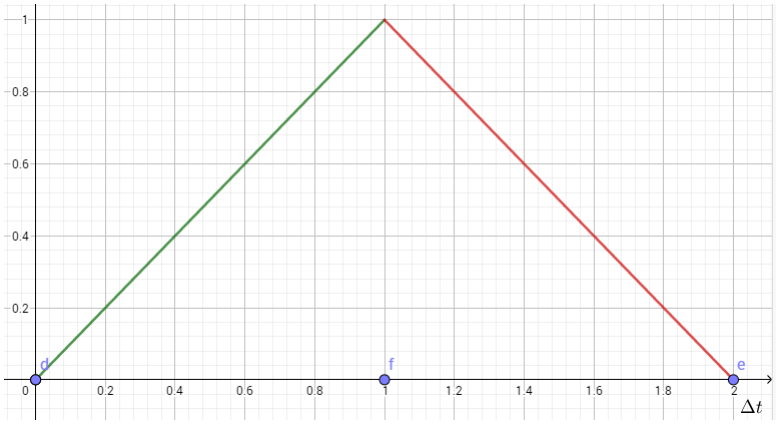

Figure 2: The triangular distribution function $f(\Delta t)$

5 Discussion

What all of this means is that a player never can be consistent with a frame perfect input, if the player doesn't know where in the frame input A is placed. It does also however mean that a frame perfect input have a two frame probability distribution, so it's possible (however not probable) to hit a frame perfect input even if your input is 0.9 frames off to either sides.

In isolated cases this probability distribution should also work the same way on vanilla melee, since one way of looking at the update frame is just to translate it to polling frequency, and apply the same distribution to it. The distribution will however not work if two polls are made in the same frame, within the frame perfect input range.

I suspect that players which is consistent with some frame perfect inputs, is in "sync" with the game, and somehow knows where their first input lands in the frame based on visual cues, auditory cues, and inputs done before the frame perfect input. This also depends on lag from the TV, so a hand warmer makes sense to get a feel for the setup.

If the players do not get in "sync" of the frames, one could argue that every frame perfect input have a probability connected to them, therefore there is a random element of timing in every input one does.

In the case of dash back on vanilla melee, one would be able to get it consistent if the player knew where in the frame case A is, which requires lag analyses on ridiculous levels, where one watches idle animation and compare it to what frame it updates and input lag, while also moving the stick fast and at the right time. This is however not the case, and therefore I can conclude that this probability distribution applies to some frame perfect options in melee. UCF in this case is probably a good option to eliminate the probability of the frame perfect input.

I don't think this post will change the meta at all (I am however not an expert on melee meta, so who knows?), because players will often go for safe non frame perfect options anyways. The only thing it proves is that frame perfect options is less optimal than previous assumed, and failed frame perfect ledge dashing foxes will have a new John.

6 Conclusion

A player with no feedback from the game, cannot be 100% consistent at frame perfect inputs, since it is impossible to input with exactly one frame accuracy (input B 0,01666 sec after input A). A frame perfect input will in this case have a range of two frames, starting at the first input, where the probability of the inputs producing a frame perfect input increases to 100% toward the middle of this range, and 0% to the sides. In some cases, this property can be applied to melee with a feedback from the game.

7 Further development

It could be interesting to make a model based on failure rate of frame perfect inputs, which describes a Gaussian probability distribution (or something similar), that takes in account accuracy and precision. I do however think this would need an external measurement (other than the GC), to get a usable result.

References

[1] Kadano. ssbm input lag on console and dolphin/faster melee. http://kadano.net/SSBM/inputlag/. Accessed: 2017-10-12.