A guide to understand and build a sequencer for a rhythm game using Stepmania's SSC definitions

Pedro G. Bascoy
pepo_gonba@hotmail.com

March 6, 2023

This guide aims to assist anyone in creating a sequencer for a rhythm game. A sequencer is a software that manages two crucial aspects of any rhythm game:

1. The timing of when the notes should be tapped concerning the beginning of the song.
2. The placement of the notes at any given time.

In this guide, we will delve into these two aspects when introducing gimmicks in the game. Gimmicks are alterations in these two aspects that provide additional (usually aesthetic) possibilities to step makers. Specifically, we will examine the StepMania approach to defining the settings required to set up a sequencer. Each definition will be analyzed in its own separate section, with examples provided, and mathematical formalism that can be easily implemented in the programming language of your choice at the end of each section.

This document's sections will focus on two or, at most, three mathematical spaces and the gimmicks that allow transformation from one space to another.

1 From song time to sequencer time

1.1 Introduction

The SSC format provides three gimmicks to artificially skip or pause the note scrolling when the song is playing. Those gimmicks are #WARPS, #STOPS, and #DELAYS. WARP gimmicks are concerned with skipping part of the song along with its notes (rendering them as fake notes). STOPS and DELAYS are gimmicks that allow to stop the scrolling of the song for a certain amount of time. Although there is a subtle difference between them, they are essentially identical w.r.t. the effect that produces. Let us review them one by one to see what each one of them is doing.

One STOPS definition might look as follows:

```
#STOPS: 4=5,6=2;
```

Let us convert the definition into a JSON-like structure:

```
{
  [
    beat: 4,
    stop: 5
  ],
  [
    beat: 6,
    stop: 2
  ]
}
```

This #STOPS gimmick is telling us a couple of things.

1. At beat 4, stop the scrolling for 5 seconds. Then resume.

2. At beat 6, stop the scrolling for 2 seconds. Then resume.

Similarly, one DELAYS defition could look like this:

```
#DELAYS: 4=5,6=2;
```

And after converting it into the friendly structure:

```
{
  [
    beat: 4,
    delay: 5
  ],
  [
    beat: 6,
    delay: 2
  ]
}
```

The `#DELAYS` definition above is telling us essentially the same story as the STOPS definition shown before, i.e.:

1. At beat 4, stop the scrolling for 5 seconds. Then resume.
2. At beat 6, stop the scrolling for 2 seconds. Then resume.

The difference between these two is that notes that lie exactly at the stop and delay beat, will need to be tapped before and after the waiting time for the stops and delays, respectively.

Finally, suppose we have the following WARPS definition:

```
#WARPS:2=1,3=2;
```

It is equivalent to this one:

```
{
  [
    beat: 2,
    warp: 1
  ],
  [
    beat: 3,
    warp: 2
  ]
}
```

As it turns out, this gimmick is telling us the following information:

1. At beat 2, warp over the next 1 beat, i.e. skip one beat. Notes with beats in the range $[2, 3+1) = [2, 3)$ will become fake notes.
2. At beat 3, warp over the next 2 following beats, i.e. skip two beats. Notes with beats in the range $[3, 3+2) = [3, 5)$ will become fake notes.

Note that with this example definition, we could write an equivalente WARP definition that would produce the same result:

```
#WARPS:2=3;
```

1.2 Challenge

First, we would like to have a function $q : \mathbb{R} \to \mathbb{R}$ that maps the song time into the warped song time (i.e. song time with skips), as well as its inverse $q^{-1} : \mathbb{R} \to \mathbb{R}$ that maps from the warped song time into the song time. Additionally, we would like to have a pair of functions $t_{(s)} : \mathbb{R} \to \mathbb{R}$, and $t_{(d)} : \mathbb{R} \to \mathbb{R}$ that would retrieve the song time after stops and delays (sequencer time) from the song time. Additionally, we would like to have two inverse functions of $t_{(s)}, t_{(d)}$, $t_{(s)}^{-1}$ for STOPS and $t_{(d)}^{-1}$ for DELAYS so we are able to map from the sequencer time into the song time.

1.3 Solution

To do so, let us imagine that we have a function $f^{-1} : \mathbb{R} \to \mathbb{R}$ that given a beat calculates the song time at that beat. To simplyfy things further, imagine that this song is has constant BPM of 60, so each second is worth 1 beat. Then $f^{-1}(b) = b$, being b a beat.

Alright, under this assumption, we can start working out a function for the WARPS gimmick. One problem that we might have is while WARPS are defined in the beat space, as we saw earlier, they act in the song time space. This is very counterintuitive – We are not actually skipping beats, but we are actually skipping seconds. That is why we would need a function (in the real life) f^{-1} to convert from one space to the other. Actually, all the functions that we are going to define in the "Formalization" sections throughout this document are just mappings from one space to the other. This function f^{-1} that I am referring to will be described later on.

Anyways, given that our BPM is 60, beats are equivalent to seconds – this is just to keep it simple enough. Knowing this, we can write the function

$$q(x) = \begin{cases} x\,, & \text{if } x \leq 2\,; \\ x + 1\,, & \text{if } 2 - 0 < x \leq 3 - 1\,; \\ x + 1 + 2\,, & \text{if } x > 3 - 1\,; \end{cases} \tag{1}$$

that will skip from beat 2 to beat 3, and from beat 3 to beat 5. You can see the plot of this function in Figure 1.

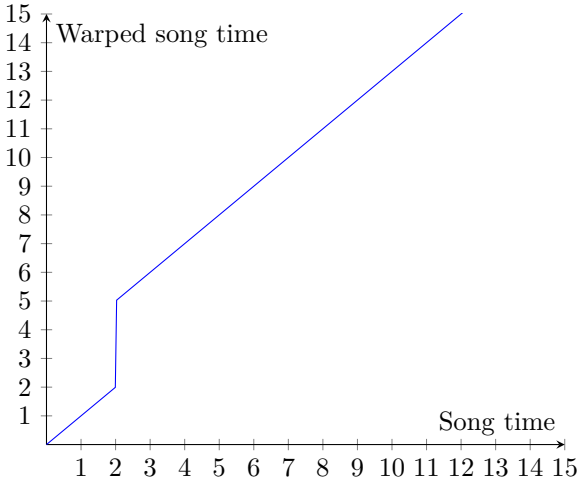

Figure 1: Plot of q

Great! The inverse function

$$q^{-1}(x) = \begin{cases} x\,, & \text{if } x \leq 2\,; \\ 2\,, & \text{if } 2 < x \leq 2 + 1\,; \\ 2\,, & \text{if } 3 < x \leq 3 + 2\,; \\ x - 2 - 1\,, & \text{if } x > 3 + 2\,; \end{cases} \tag{2}$$

will just do the opposite, i.e. unskipping the skipped beats. Next, let us deal with STOPS and DELAYS.

Keeping the assumption that BPM is 60, then we can write the piecewise functions

$$t_{(s)}(x) = t_{(d)}(x) = \begin{cases} x\,, & \text{if } x \leq 4\,; \\ 4\,, & \text{if } 4 < x \leq 4+5\,; \\ x-5\,, & \text{if } 4+5 < x \leq 6+5\,; \\ 6+5\,, & \text{if } 6+5 < x \leq 6+5+2\,; \\ x-2-5\,, & \text{if } x > 6+5+2\,; \end{cases} \tag{3}$$

that will map from the warped song time to sequencer time for the STOPS and DELAYS, respectively. A plot of this function can be see in Figure 2.

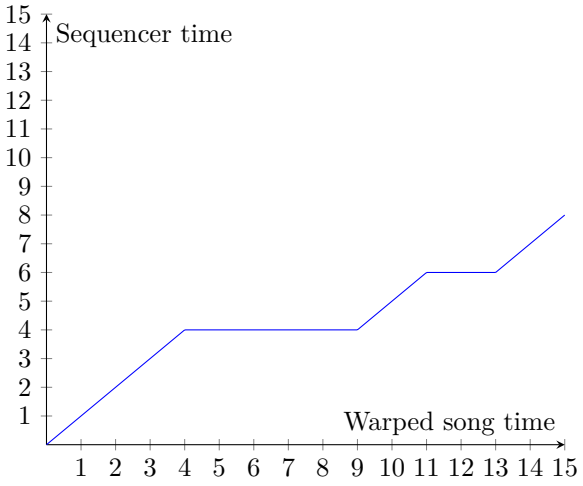

Figure 2: Plot of $t_{(d)}$

We can also easily calculate the two different inverse functions to model when notes at stops or delays should be tapped w.r.t. the song time. On the one hand side, we write the function

$$t^{-1}_{(s)}(x) = \begin{cases} x\,, & \text{if } x \leq 4\,; \\ x+5\,, & \text{if } 4 < x \leq 6\,; \\ x+5+2\,, & \text{if } x > 6\,; \end{cases} \tag{4}$$

to map from the sequencer time into the warped song time for STOPS. Similarly, the function

$$t^{-1}_{(d)}(x) = \begin{cases} x\,, & \text{if } x < 4\,; \\ x+5\,, & \text{if } 4 \leq x < 6\,; \\ x+5+2\,, & \text{if } x \geq 6\,; \end{cases} \tag{5}$$

will map from the sequencer time into the warped song time for DELAYS. Note that the signs at the conditions are slightly different.

1.4 Formalization

Warps Let $\mathcal{W} = \left\{ \left(b_i^{(w)}, w_i \right) \right\}_{i=1}^{n}$ be a sequence of WARPS, where w_i is the warp (measured in beats) at beat $b_i^{(w)}$ and let $f^{-1} : \mathbb{B} \to \mathbb{S}^*$ be a function that maps from the beat space into the sequencer time space. We define a new set

$$\mathcal{W}' = \{(b'_i, w'_i)\} = \bigcup_{i=2}^{n} \left\{ \left(f^{-1}\left(b_i^{(w)} \right), z\left(w_i, b_i^{(w)} \right) \right) \right\} \tag{6}$$

where

$$z(w, b) = f^{-1}(b + w) - f^{-1}(b)\,. \tag{7}$$

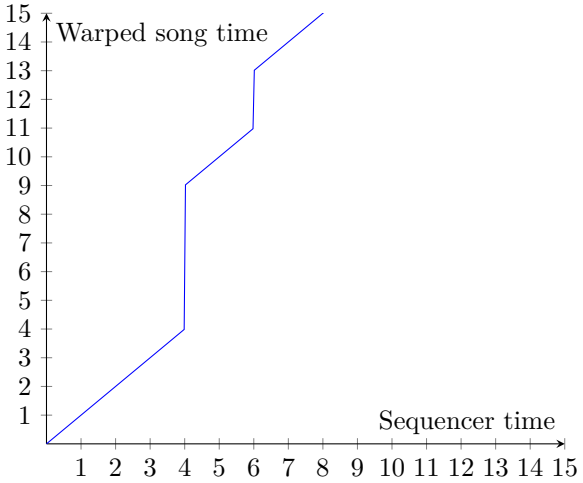

Figure 3: Plot of t^{-1}

We define the function $q : \mathbb{S} \to \mathbb{W}$

$$q(x) = \begin{cases} x\,, & \text{if } x \leq b'_1\,; \\ x + \sum_{j=1}^{i} w'_j\,, & \text{if } b'_i - \sum_{j=1}^{i-1} w'_j < x \leq b'_{i+1} - \sum_{j=1}^{i} w'_j\,, \quad \forall i = 1, \ldots, n\,; \end{cases} \tag{8}$$

where $b'_{n+1} := \infty$ which maps from the song time space into the warped time space, if $\mathcal{W} \neq \emptyset$.

We define the function $q^{-1} : \mathbb{W} \to \mathbb{S}$

$$q^{-1}(x) = \begin{cases} x\,, & \text{if } x \leq b'_1\,; \\ b'_i - \sum_{j=1}^{i-1} w'_j\,, & \text{if } b'_i < x \leq b'_i + w'_i\,, \quad \forall i = 1, \ldots, n\,; \\ x - \sum_{j=1}^{i} w'_j\,, & \text{if } b'_i + w'_i < x \leq b'_{i+1}\,, \quad \forall i = 1, \ldots, n\,; \end{cases} \tag{9}$$

when $\mathcal{W} \neq \emptyset$, where $b_0^{(w)} := -\infty$, that maps from the warped time spate into the song time space. If $\mathcal{W} = \emptyset$, then $q(x) = q^{-1}(x) = x$.

Stops and Delays Let $\mathcal{T} = \left\{\left(b_i^{(t)}, r_i\right)\right\}_{i=1}^{n}$ be a sequence of STOPS or DELAYS, where r_i is the stop or delay (measured in seconds) at beat $b_i^{(t)}$. Let $f^{-1} : \mathbb{B} \to \mathbb{S}^*$ be a function that retrieves the sequencer time given a beat, where $\mathbb{B} = \mathbb{R}$, and $\mathbb{S}^* = \mathbb{R}$.

We define a new set

$$\mathcal{T}' = \{(c_i, r_i)\}_{i=1}^{n} = \left\{\left(f^{-1}\left(b_i^{(t)}\right), r_i\right)\right\}_{i=1}^{n} \tag{10}$$

where c_i is the second from the start of the song of beat $b_i^{(t)}$. We define the functions $t_{(s)} : \mathbb{D} \to \mathbb{S}^*$ and $t_{(d)} : \mathbb{W} \to \mathbb{D}$

$$t_{(s)}(x) = t_{(d)}(x) = \begin{cases} x - \sum_{j=1}^{i-1} r_j\,, & \text{if } c_{i-1} + \sum_{j=1}^{i-1} r_j < x \leq c_i + \sum_{j=1}^{i-1} r_j\,, \quad \forall i = 1, \ldots, n\,; \\ c_i\,, & \text{if } c_i + \sum_{j=1}^{i-1} r_j < x \leq c_i + \sum_{j=1}^{i} r_j\,, \quad \forall i = 1, \ldots, n\,; \\ x - \sum_{j=1}^{n} r_j & \text{if } x > c_n + \sum_{j=1}^{n} r_j\,; \end{cases} \tag{11}$$

when $\mathcal{T}' \neq \emptyset$, where $c_0 := -\infty$, that maps from the warped song time into the sequencer time.

We define the function $t_{(s)}^{-1} : \mathbb{S}^* \to \mathbb{D}$

$$t_{(s)}^{-1}(x) = \begin{cases} x\,, & \text{if } x \leq c_1\,; \\ x + \sum_{j=1}^{i} r_i\,, & \text{if } c_i < x \leq c_{i+1}\,, \quad \forall i = 1, \ldots, n\,; \end{cases} \tag{12}$$

with $c_{n+1} := \infty$ which maps from the sequencer time into the delayed song time for the STOPS, if $\mathcal{T}' \neq \emptyset$, and the function $t_{(d)}^{-1} : \mathbb{D} \to \mathbb{S}$

$$t_{(d)}^{-1}(x) = \begin{cases} x\,, & \text{if } x < c_1\,; \\ x + \sum_{j=1}^{i} r_i\,, & \text{if } c_i \leq x < c_{i+1}\,, \quad \forall i = 1, \ldots, n\,; \end{cases} \tag{13}$$

that maps from the delayed time into the warped song time for the DELAYS, if $\mathcal{T}' \neq \emptyset$. If $\mathcal{T}' = \emptyset$, then $t_{(s)}(x) = t_{(s)}^{-1}(x) = x$ and $t_{(d)}(x) = t_{(d)}^{-1}(x) = x$, for STOPS and DELAYS respectively.

Song time to sequencer time mapping The final mapping from song time into sequencer time is $t_{(s)} \circ t_{(d)} \circ q : \mathbb{S} \to \mathbb{S}^*$. Similarly, the mapping from sequencer time into the song time is $q^{-1} \circ t_{(d)}^{-1} \circ 1_{(s)}^{-1} : \mathbb{S}^* \to \mathbb{S}$.

2 From sequencer time to beat

2.1 Introduction

A `SSC` file gives a list of pairs which defines the bpms. The first item in the pair is the target beat, and the second item is the desired BPM from that beat on. Let us imagine we have a `SSC` file with the following definition:

```
#BPMS:0=120,8=70,13=200;
```

Let us convert this cumbersome definition into a friendly structure:

```
{
  [
    beat: 0,
    bpm: 120
  ],
  [
    beat: 8,
    bpm: 180
  ],
  [
    beat: 13,
    bpm: 60
  ]
}
```

This `#BPMS` definition is telling us three things:

1. From beat $-\infty$ to beat 8, the BPM is 120.
2. From beat 8 to beat 13, the BPM is 180.
3. From beat 13 to beat $+\infty$, the BPM is 60.

2.2 Challenge

We want to find a function $f : \mathbb{R} \to \mathbb{R}$ that retrieves the current beat given the a second in the sequencer time space. This function is useful when a song is playing and we want to know at what beat we are at if we know how much time has passed since the start of the song. Also, as we will see later on, notes scroll at a BPM rate, so if we can have the inverse function of f, f^{-1}, we can sort of know when the steps should be tapped as well.

2.3 Solution

First, let us convert BPMS to BPSS (Beats Per Second), since we are going to provide the input in seconds instead of minutes. We can do so by dividing the BPMS by 60, i.e.

$$\text{BPS}(x) = x \times \frac{\text{Beats}}{\text{Minute}} = x \times \frac{1 \times \text{Minute}}{60 \times \text{Seconds}} \frac{\text{Beats}}{\text{Minute}} = \frac{x}{60} \times \frac{\text{Beats}}{\text{Second}}\,. \tag{14}$$

Next, let us define a piecewise function $f' : \mathbb{R} \to \mathbb{R}$ that gives the current BPS given the current Beat. Taking the `#BPMS` toy example from the previous section, we get that

$$f'(x) = \begin{cases} 2\,, & \text{if } x \leq 8\,; \\ 3\,, & \text{if } 8 < x \leq 13\,; \\ 1\,, & \text{if } x > 13\,. \end{cases} \tag{15}$$

In Figure 4 you can see the plot of f' we just defined in (15).

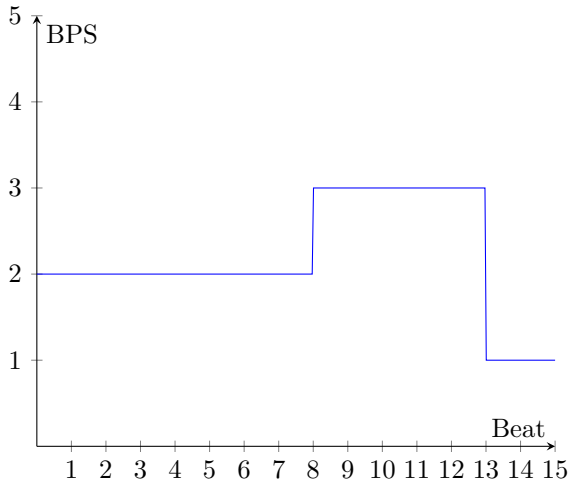

Figure 4: Plot of f'

Note that by using f', we can get the BPS at any beat of the song. This is great, but it does not quite solve our problem.

Next, we can calculate the SPB (Seconds Per Beat) by just inversing the BPS, i.e.

$$\text{SPB} = \frac{1}{\text{BPS}}\,, \tag{16}$$

and therefore we can define a function $t : \mathbb{R} \to \mathbb{R}$

$$t(x) = x \times \text{SPB} \tag{17}$$

that given a beat x retrieves the current second.

Let

$$f^{-1}(x) = \begin{cases} \dfrac{x}{2}\,, & \text{if } x \leq 8\,; \\ \dfrac{8}{2} + \dfrac{x-8}{3}\,, & \text{if } 8 < x \leq 13\,; \\ \dfrac{8}{2} + \dfrac{5}{3} + x - 13\,, & \text{if } x > 13\,; \end{cases} \tag{18}$$

be the function that given a beat x retrieves the current second. This function is the result of plugging (16) and (17) into (15), and can be rewritten recursively as

$$f^{-1}(x) = \begin{cases} \dfrac{x}{2}\,, & \text{if } x \leq 8\,; \\ f^{-1}(8) + \dfrac{x-8}{3}\,, & \text{if } 8 < x \leq 13\,; \\ f^{-1}(13) + x - 13\,, & \text{if } x > 13\,. \end{cases} \tag{19}$$

Figure 5 depicts the function f^{-1}.

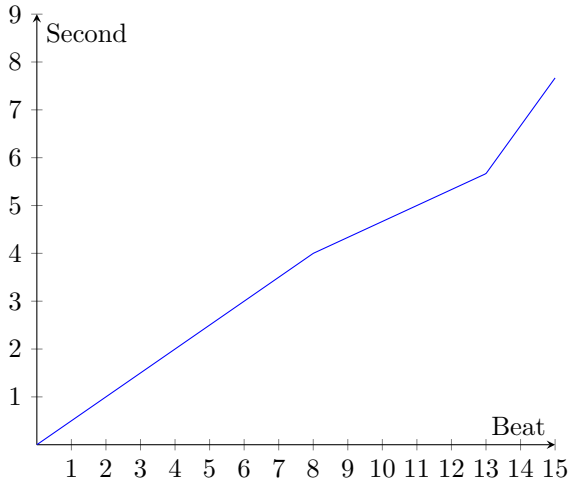

Figure 5: Plot of f^{-1}

As it turns out, the function f that we are looking for is just the inverse function of f^{-1}, thus

$$f(x) = \begin{cases} 2x\,, & \text{if } x \leq 4\,; \\ (x-4) \times 3 + 8\,, & \text{if } 4 < x \leq 5.6\,; \\ x - 5.6 + 13\,, & \text{if } x > 5.6\,. \end{cases} \tag{20}$$

A plot of f can be seen in Figure 6.

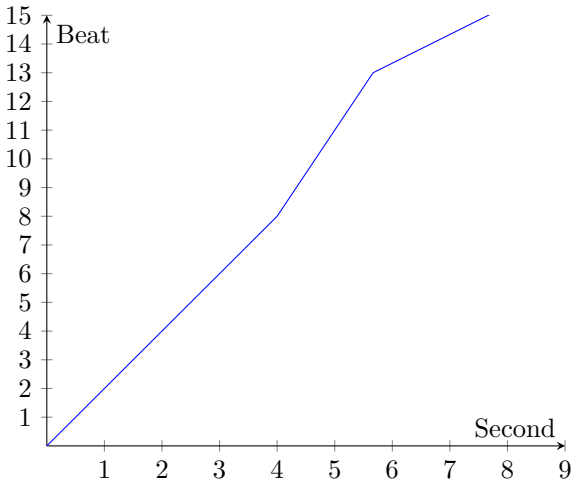

Figure 6: Plot of f

2.4 Formalization

Let $\{(b_i, v_i)\}_{i=1}^{n}$ be a sequence of n beat signatures, where v_i is the BPS value at beat b_i. Let $f^{-1} : \mathbb{B} \to \mathbb{S}^*$ be a function that provided a beat, returns the seconds in the sequencer time space. We define this function as a n-step piecewise function

$$f^{-1}(x) = \begin{cases} \dfrac{x}{v_1}\,, & \text{if } x \leq b_2 \ ; \\ f^{-1}(b_i) + \dfrac{x - b_i}{v_i}\,, & \text{if } b_i < x \leq b_{i+1}\,; \quad \forall i = 2, \ldots, n\,; \end{cases} \tag{21}$$

where $b_1 := 0$, and $b_{n+1} := \infty$.

Analogously, let $f : \mathbb{S}^* \to \mathbb{B}$ be a function that provided a second in the sequencer time space, returns the beat from the zero second. We define this function as a n-step piecewise function

$$f(x) = \begin{cases} v_1 x\,, & \text{if } x \leq f^{-1}(b_2) \ ; \\ \left[x - f^{-1}(b_i)\right] \times v_i + b_i\,, & \text{if } f^{-1}(b_i) < x \leq f^{-1}(b_{i+1})\,; \quad \forall i = 2, \ldots, n\,. \end{cases} \tag{22}$$

3 From beat to note position

3.1 Introduction

There are a pair of stepmania definitions that influence the position where notes should be placed on the screen. One of them is `#BPMS`, which arguably is the rate at what notes travel upwards towards the receptor w.r.t. to the music rhythm. We have already dealt with it in the previous section.

However, there is another gimmick that plays a role in the note positioning: `#SCROLLS`. Let's have a look at an example:

```
#SCROLLS:0=1,4=0,10=2;
```

Again, let us convert this cumbersome definition into a friendly structure:

```
{
  [
    beat: 0,
    scroll: 1
  ],
  [
    beat: 4,
    scroll: 0
  ],
  [
    beat: 10,
    scroll: 2
  ]
}
```

The scroll gimmick changes the effective BPM at a given beat by a rate defined by the scroll value. Thus, in this example, the SCROLLS gimmick is changing the BPMs as follows:

1. From beat 0 to beat 4, the BPM is its value times 1.
2. From beat 4 to beat 10, the BPM is its value times 0. (all the steps in between this beats, will have the same position)
3. From beat 10 on, the BPM is its value times 2.

3.2 Challenge

We would like to have a function $p : \mathbb{R} \to \mathbb{R}$ that given a beat, it retrieves the position w.r.t. the origin (or where the receptor is) where a note at that beat should be drawn.

3.3 Solution

Let us define a function $p : \mathbb{R} \to \mathbb{R}$ that given a beat, retrieves the effective beat (i.e., beat with applied scrolls). For that matter, we just need to check out in what beats the scroll is taking place, and change the beat accordingly to the scroll rate. For our toy example the resultant p function looks like this

$$p(x) = \begin{cases} x\,, & \text{if } x \leq 4\,; \\ (x-4) \times 0 + p(4)\,, & \text{if } 4 < x \leq 10\,; \\ (x-10) \times 2 + p(10)\,, & \text{if } x > 10\,. \end{cases} \tag{23}$$

The function p is depicted in Figure 7. Now, this is great! By asking p, now we have the effective

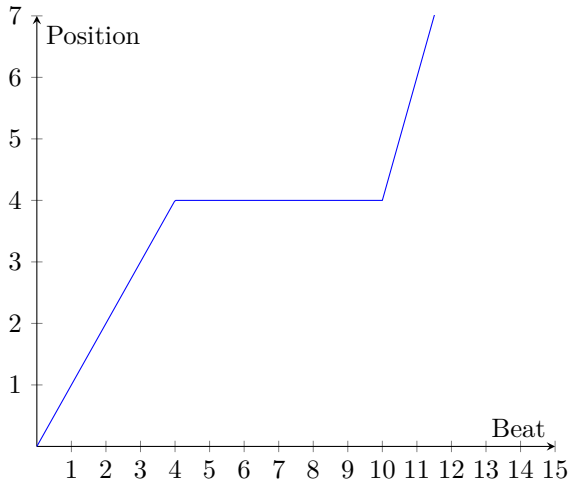

Figure 7: Plot of p

beat, and therefore the position. Note that de drawing position IS NOT the beat for that we are going to use to calculate when the note is needed to be tapped!

3.4 Formalization

Let

$$\left\{ \left(b_i^{(s)}, s_i \right) \right\}_{i=1}^{n} = \mathcal{S} = B^{(s)} \times S \tag{24}$$

be a sequence of m scroll signatures, where $s_i \in S = \{s_j\}_{j=1}^{n}$ is the scroll value at beat $b_i^{(s)} \in B^{(s)} = \{b_j^{(s)}\}_{j=1}^{n}$.

We define the function $p : \mathbb{R} \to \mathbb{R}$

$$p(x) = \begin{cases} s_1 x\,, & \text{if } x \leq b_2^{(s)}\,; \\ p\left(b_i^{(s)}\right) + \left(x - b_i^{(s)}\right) \times s_i\,, & \text{if } b_i^{(s)} < x \leq b_{i+1}^{(s)}\,; \quad \forall i = 2, \ldots, n\,, \end{cases} \tag{25}$$

as the function that retrieves the position given a beat, where $b_{i+1}^{(s)} := \infty$.

4 From beat to scroll

4.1 Introduction

Another issue that we might have is to know how far upwards we should scroll the notes from its original position given the current beat. If we did not have any other gimmicks, this would be very easy to compute. Let us suppose that the notes redendered in the screen are squares of one unity of length and height, and that one beat is worth one distance of separation, as shown in Figure 8. In this set up, the amount of scroll that we have to apply for a given beat x is just $-x$ (if we were scrolling upwards, on the y axis).

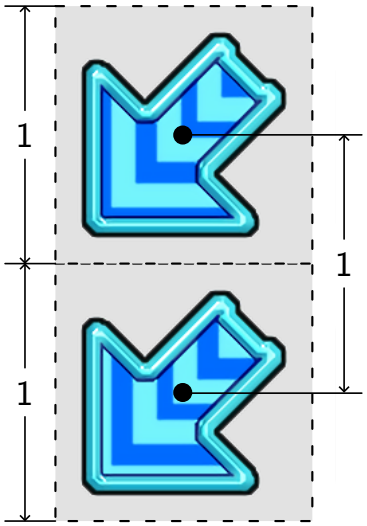

Figure 8: Two notes inside with their respective bounding boxes separeted apart by one beat.

However there are two gimmicks that modify this scroll function w.r.t. the beat space in a number of different ways: `#SPEEDS`, and `#SCROLLS`. Let us see and example to know what these modifiers do.

On the one hand, suppose we have the following `#SPEEDS` definition:

```
#SPEEDS:4=2=1=0,6=0.5=1=1;
```

which is equivalent to this one:

```
{
  [
    beat: 0,
    speed: 1,
    span: 0,
    type: 0
  ],
  [
    beat: 6,
    speed: 7 ,
    span: 1,
    type: 1
  ]
}
```

Well, the information provided with this gimmick is the following:

1. From beat 0 on, the separation between notes increases by a factor of 1, the scrolling speed also increases by a factor of 1, and because the `type` is `0`, this transition is smoothly applied in the span of 0 **beats**.
2. From beat 6 on, the separation between notes increases by a factor of 7, the scrolling speed also increases by a factor of 7, and because the `type` is `1`, this transition is smoothly applied in the span of 1 **second**.

On the other hand, We have already talked about how the `#SCROLLS` definition changes where the notes should be placed on the screen. It turns out that also has an impact on the scrolling function. Taking the same example from Section 3, we can extract the following information:

1. From beat 0 to beat 4, the scrolling 1 times as fast.
2. From beat 4 to beat 10, the scrolling is 0 times as fast.
3. From beat 10 on, the the scrolling is 2 times as fast.

4.2 Challenge

We would like to have a function $e : \mathbb{R} \to \mathbb{R}$ that calculates the speed factor at a given beat, and a function $g : \mathbb{R} \to \mathbb{R}$ that calculates the scroll value.

4.3 Solution

Speeds The function that we need to come up for the speeds is simple as well. However, it gets a bit complicated when dealing with speeds of `type` 1: Since we are working at the beat space, we need somehow to convert a span of seconds into a span of beats so we can perform operations in the same units. We do so by mapping a given beat into the song time space, then add the span of seconds, and then mapping back into the beat space. To keep things simple, let us imagine that we are running a 60 BPM song with no delays or stops. In this particular case, we can write

$$e(x) = \begin{cases} 1\,, & \text{if } x \leq 6\,; \\ \dfrac{7-1}{1}(x-6)+1\,, & \text{if } 6 < x \leq 6+1\,; \\ 7\,, & \text{if } x > 6+1\,. \end{cases} \tag{26}$$

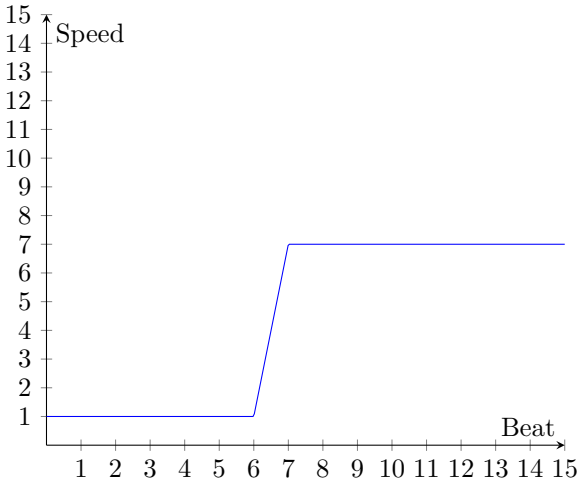

Figure 9: Plot of e

Scrolls There is not much to say here. The function g that we are looking for is just identical to the function p,

$$g(x) = p(x) \tag{27}$$

4.4 Formalization

Speeds Let $\mathcal{E} = \left\{ \left(b_i^{(e)}, s_i, p_i, t_i \right) \right\}_{i=1}^{n}$ be a sequence of SPEEDS, where s_i is the speed change at beat $b_i^{(e)}$ with a span transition of p_i beats if $t_i = 0$, or p_i seconds if $t_i = 1$.

We define a new sequence $\mathcal{E}' = \left\{ \left(b_i^{(e)}, s_i, p_i' \right) \right\}_{i=1}^{n}$ from $\mathcal{E}$ as

$$\mathcal{E}' = \bigcup_{i=1}^{n} \begin{cases} \left\{ \left(b_i^{(e)}, s_i, p_i \right) \right\}\,, & \text{if } t_i = 0\,; \\ \left\{ \left(b_i^{(e)}, s_i, g\left(p_i, b_i^{(e)} \right) \right) \right\}\,, & \text{otherwise}\,, \end{cases} \tag{28}$$

where

$$g(p, b) = \left(f \circ t_{(s)} \circ t_{(d)} \circ q \right) \left(\left(q^{-1} \circ t_{(d)}^{-1} \circ t_{(s)}^{-1} \circ f^{-1} \right)(b) + p \right) - b\,. \tag{29}$$

We define the function $e : \mathbb{B} \to \mathbb{E}$

$$e(x) = \begin{cases} s_1\,, & \text{if } x \leq b_1^{(e)}\,; \\ \dfrac{s_i - s_{i-1}}{p'_i}\left(x - b_i^{(e)}\right) + s_{i-1}\,, & \text{if } b_i^{(e)} < x \leq b_i^{(e)} + p'_i \wedge p'_i \neq 0\,, \quad \forall i = 1, \ldots, n\,; \\ s_i\,, & \text{if } b_i^{(e)} + p'_i < x \leq b_{i+1}^{(e)}\,, \quad \forall i = 1, \ldots, n\,, \end{cases} \tag{30}$$

where $b_{n+1}^{(e)} := \infty$, and $s_0 := s_1$.

Scrolls We define $g : \mathbb{R} \to \mathbb{R}$

$$g(x) = p(x) \tag{31}$$

where p is (25).

5 Positioning and scrolling notes

Alright! Now we have formally defined everything we need to build a sequencer using stepmania's notation. This section will show how to use all functions defined formally in the sections above to determine where to draw the notes in the screen at any given song time.

To ease the understanding, Table 1 gathers all the functions that we are going to need as well as a brief descripcion for each one of them.

Symbol	Description
$\mathbb{S}$	Song time space (s) $\mathbb{S} = \mathbb{R}$.
$\mathbb{D}$	Delayed time space (s) $\mathbb{D} = \mathbb{R}$.
$\mathbb{S}^*$	Sequencer time space (s) $\mathbb{S}^* = \mathbb{R}$.
$\mathbb{B}$	Beat space (beats) $\mathbb{B} = \mathbb{R}$.
$\mathbb{P}$	Position space (units) $\mathbb{P} = \mathbb{R}$.
$\mathbb{T}$	Scroll space (units) $\mathbb{T} = \mathbb{R}$.
$\mathbb{E}$	Speed space (speed factor units) $\mathbb{E} = \mathbb{R}$.
$\mathbb{W}$	Warped song time space (units) $\mathbb{W} = \mathbb{R}$.
$t_{(d)}$	Function $t_{(s)} : \mathbb{W} \to \mathbb{D}$ that maps from the song time space into the delayed time space.
$t_{(d)}^{-1}$	Function $t_{(d)}^{-1} : \mathbb{D} \to \mathbb{W}$ that maps from the the delayed time space into the song time space.
$t_{(s)}$	Function $t_{(s)} : \mathbb{D} \to \mathbb{S}^*$ that maps from the delayed time space into the sequencer time space.
$t_{(s)}^{-1}$	Function $t_{(s)}^{-1} : \mathbb{S}^* \to \mathbb{D}$ that maps from the the sequencer time space into the delayed time space.
f	Function $f : \mathbb{S}^* \to \mathbb{B}$ that maps from the sequencer time space into the beat space.
f^{-1}	Function $f^{-1} : \mathbb{B} \to \mathbb{S}^*$ that maps from the beat space into the sequencer time space.
p	Function $p : \mathbb{B} \to \mathbb{P}$ that maps from the beat space into the position space.
q	Function $q : \mathbb{S} \to \mathbb{W}$ that maps from the beat space into the warped beat space.
q^{-1}	Function $q^{-1} : \mathbb{W} \to \mathbb{S}$ that maps from the warped beat space into the beat space.
e	Function $e : \mathbb{B} \to \mathbb{E}$ that maps from the beat space into the speed space.
g	Function $g : \mathbb{W} \to \mathbb{T}$ that maps from the warped beat space into the scroll space.

Table 1: Table of functions and spaces.

We will take the following assumptions:

1. We are modeling a rhythm game whose notes scroll upwards.

2. The receptor is placed at the origin of the scrolling axis.

3. The beat 0 has a position 0 in the scrolling axis.

Let $\mathcal{N} = \{(u_i)\}_{i=1}^{n}$ be a sequence of notes where u_i is the beat when the i-th note should be tapped. We define a new set

$$\mathcal{N}' = \{(u_i, v_i, w_i)\} = \bigcup_{\{u\}\in\mathcal{N}} \{(u, z(u), p(u))\} \tag{32}$$

where $z : \mathbb{B} \to \mathbb{S}$

$$z(x) = \left(q^{-1} \circ t_{(d)}^{-1} \circ t_{(s)}^{-1} \circ f^{-1}\right)(x)\,, \tag{33}$$

and thus v_i is the exact time where the i-th note should be tapped, and w_i is its relative position at beat 0 in the scrolling axis. At a given moment in time t w.r.t. to the begginig of the song, the i-th note should be positioned at

$$\left[w_i - \left(g \circ f \circ t_{(s)} \circ t_{(d)} \circ q\right)(t)\right] \times \left(e \circ q \circ f \circ t_{(s)} \circ t_{(d)} \circ q\right)(t)\,. \tag{34}$$

A Tickcounts and Piu-style holds

Let

$$\mathcal{U} = \{(b_i, t_i)\}_{i=1}^{n} \tag{35}$$

be a sequence of TICKCOUNTS, where t_i is the tick count at beat b_i.

For each piu-style hold in the span of beats h_1, h_2, where h_1 is the first beat of the hold and h_2 is the last beat, we construct a new set

$$\mathcal{U}' = \{(b_i', t_i')\}_{i=1}^{n'} = \{(h_1, g(h_1))\} \tag{36}$$

$$\cup \bigcup_{(b,t)\in\mathcal{U}} \begin{cases} (b,t)\,, & \text{if } b > h1 \wedge b < h_2\,; \\ \emptyset\,, & \text{otherwise}\,; \end{cases} \tag{37}$$

$$\cup \{(h_2, 0)\}\,, \tag{38}$$

where

$$g(x) = \min_{\substack{(b,t)\in\mathcal{U};\\ x\le b}} x - b\,. \tag{39}$$

We define the sequence of beats where fake tapnotes should be place as

$$\mathcal{T} = \begin{cases} \{h_1\}\,, & \text{if } a_1 \neq 0\,; \\ \emptyset & \text{otherwise}\,, \end{cases} \tag{40}$$

$$\cup \bigcup_{i=1}^{n'-1} \bigcup_{k=1}^{s_i} \left\{ a_i + \sum_{j=1}^{k-1} \frac{1}{t_i'} \right\} \tag{41}$$

$$\cup\, h_2\,, \tag{42}$$

where

$$a_i = b_i' + b_i' \mod \frac{1}{t_i'} \tag{43}$$

and

$$s_i = t_i'(b_{i+1}' - a_i)\,. \tag{44}$$