Effects of Varying Basic Affine Transformations on Scores Generated by Iterated Function Systems

Michael Gogins
michael.gogins@gmail.com

October 5, 2021

The elementary affine transformations — scale, move, shear, rotate – of a simple iterated function system (IFS) are systematically varied to illustrate the effects of such variations on a score generator based on the IFS.

This document presents a system for composing scores for Csound [6] using the deterministic algorithm for computing iterated function systems (IFS) [5], also known as the multiple copy reducing machine (MCRM) [1], and with that system demonstrating the effects of systematically varying the affine transformations that make up the Hutchinson operator of the IFS.

The motivation for this study is that an IFS is produced by recursively shrinking a set towards a fixed point using a Hutchinson operator, a set of contractive affine transformations, and this is a *global* operation in that each of the affine transformations in the operator affects the entire score. And that means that changes in the operator are not, at first sight, intuitive. The goal of this study is to make such changes *more* intuitive.

1 Method

The system used for these experiments is Csound [4], in which the Clang/LLVM just-in-time (JIT) compiler [3, 2] has been embedded as opcodes.

In this study, scores are generated by recursively applying a Hutchinson operator 8 times to a single musical note, which actually represents a single phase-synchronous cosine grain

of sound. The Hutchinson operator simply maps the unit square to each of its four quarters. This type of grain is used because such grains can be mixed with a minimum of artifacts. This, in turn, means that viewing a spectrogram of the resulting soundfile clearly illustrates the attractor of the IFS. The durations of the grains are adjusted so that grains of the same pitch that are successive in time overlap with a minimum of artifacts.

Each score that is generated in this way is one minute long. The dimensions of the space acted upon by the operator are instrument number, time, duration, MIDI key number, MIDI velocity, and stereo pan. A single note or grain is a homogeneous column vector in this space. The operator for the initial unit square, implemented using homogeneous affine transformation matrices, is:

Listing 1: Hutchinson Operator for Unit Square.

```
/*                   i    t    d    k    v    p    T         */
hutchinson[0] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   0, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */

hutchinson[1] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   1, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   0, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */

hutchinson[2] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */

hutchinson[3] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   1, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
```

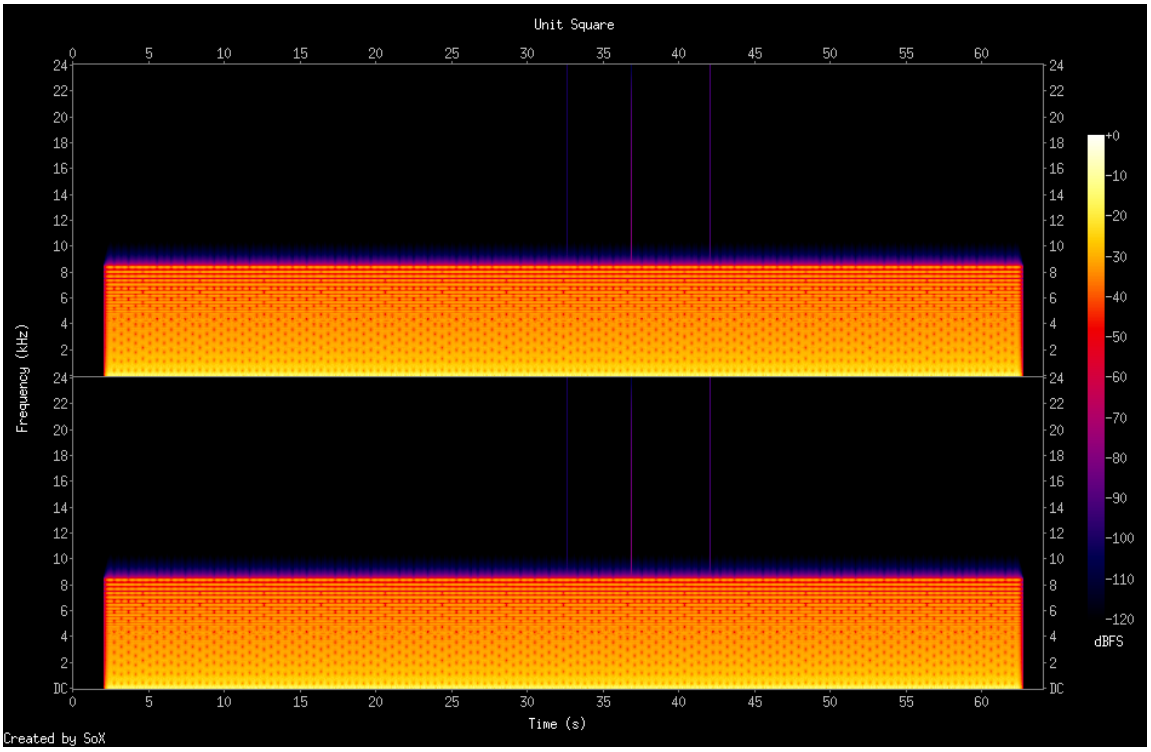


Figure 1: Unit Square.

```
0,  0,  0, .5,  0,  0,  1, /* k */
0,  0,  0,  0, .5,  0,  0, /* v */
0,  0,  0,  0,  0, .5,  0, /* p */
0,  0,  0,  0,  0,  0,  1; /* H */
```

In this operator, the diagonals of the transformations are scaling factors, and the rightmost columns are translations.

In an actual piece, often the grains would be replaced by notes for more distinctive instruments, and pitch might be rounded off to 12-tone equal temperament. In that case, the Hutchinson operator is not iterated all the way to its fixed point, i.e. its attractor, but only far enough to produce a reasonable density of notes. And in such cases, of course, the transformations have to be set up to create some sort of musical interest, and that generally means introducing various asymmetries into the action of the Hutchinson operator.

1.1 Scaling

Here we present the results of varying the scaling of time and pitch, quarter by quarter.

When rescaling time, it is easy to see how the rescaling interacts with the translations in quarters 2 through 4.

Now we consider rescaling of pitch in the same way.

1.2 Translations

1.3 Shearings

1.4 Rotations

These are 15 degree rotations of time into pitch. Actually, they are rotation-dilations because they must include the scaling by 0.5.

2 Discussion

As can be seen, it is challenging to predict the effects upon score generation of even simple modifications of a simple transformation in a simple Hutchinson operator. This is the other side of the great power of this method of composition.

These examples do, however, show some basic principles that can be useful in music composition:

References

[1] Heinz-Otto Peitgen, Hartmut Jürgens, and Dietmar Saupe. "Chaos and Fractals: New Frontiers of Science". In: *Chaos and Fractals: New Frontiers of Science*. Springer-Verlag, 1992. Chap. 5, pp. 229–296. ISBN: 0-387-97903-4 (cit. on p. 1).

[2] LLVM. *ORC Design and Implementation*. Available online at: https://www.llvm.org/docs/ORCv2.html. Accessed 27 September 2021. 2021 (cit. on p. 1).

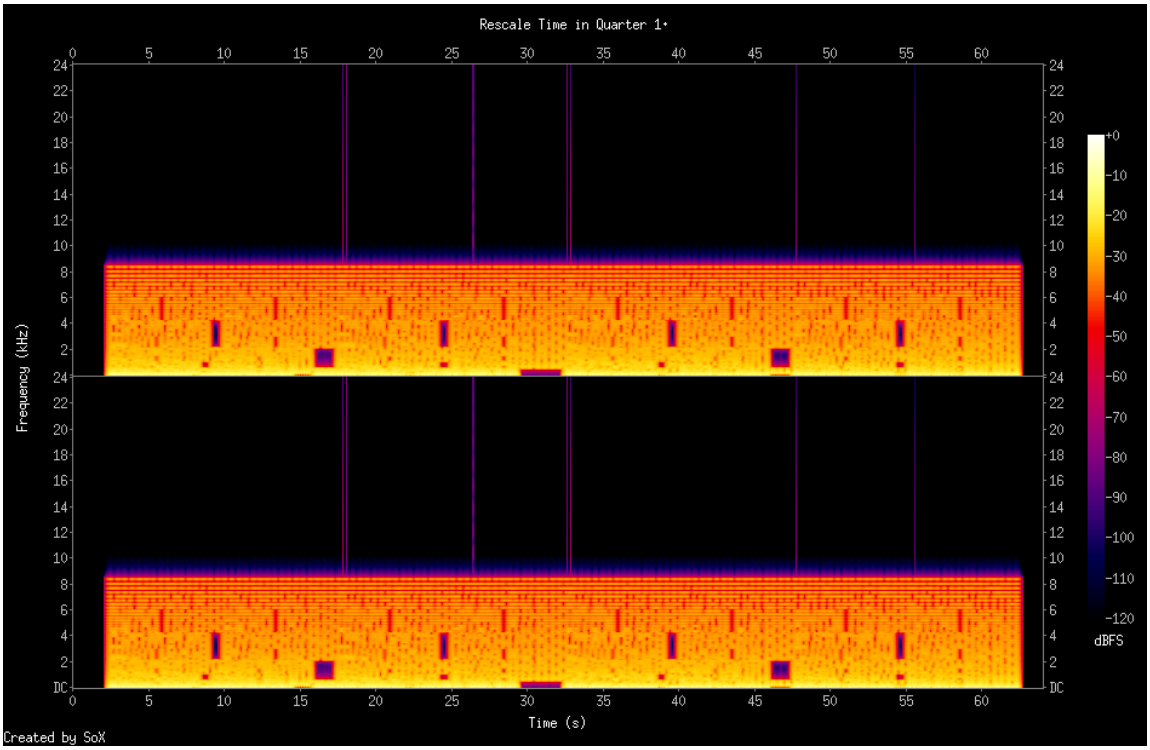
[3] LLVM. *The LLVM Compiler Infrastructure*. Available online at https://llvm.org/. Accessed 27 September 2021. 2021 (cit. on p. 1).

[4] Richard Boulanger. *csounds.com... almost everything Csound*. http://www.csounds.com (cit. on p. 1).

[5] Michael F. Barnsley. *Fractals Everywhere*. 2nd. Boston: Academic Press Professional, 1993 [1988] (cit. on p. 1).

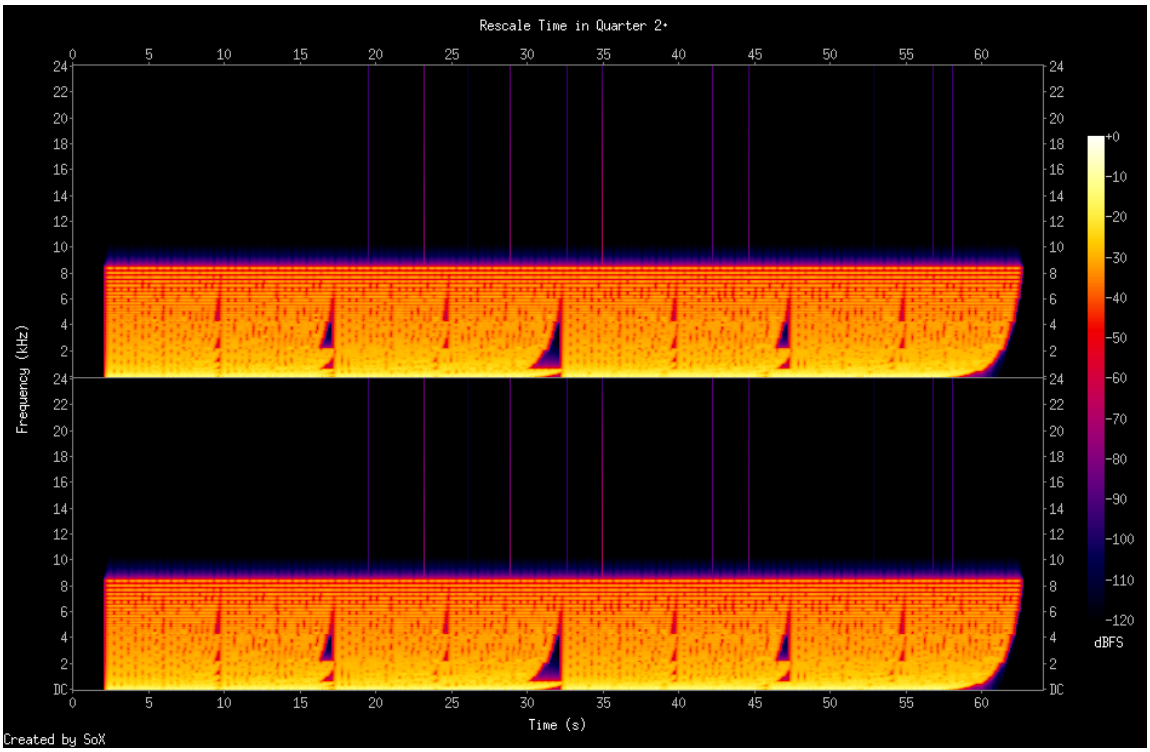
Listing 2: Rescale Time in First Quarter.

```
/*                    i    t    d    k    v    p    T        */
hutchinson[0] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .45,   0,   0,   0,   0,   0, /* t */
                    0,   0,  .5,   0,   0,   0,   0, /* d */
                    0,   0,   0,  .5,   0,   0,   0, /* k */
                    0,   0,   0,   0,  .5,   0,   0, /* v */
                    0,   0,   0,   0,   0,  .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



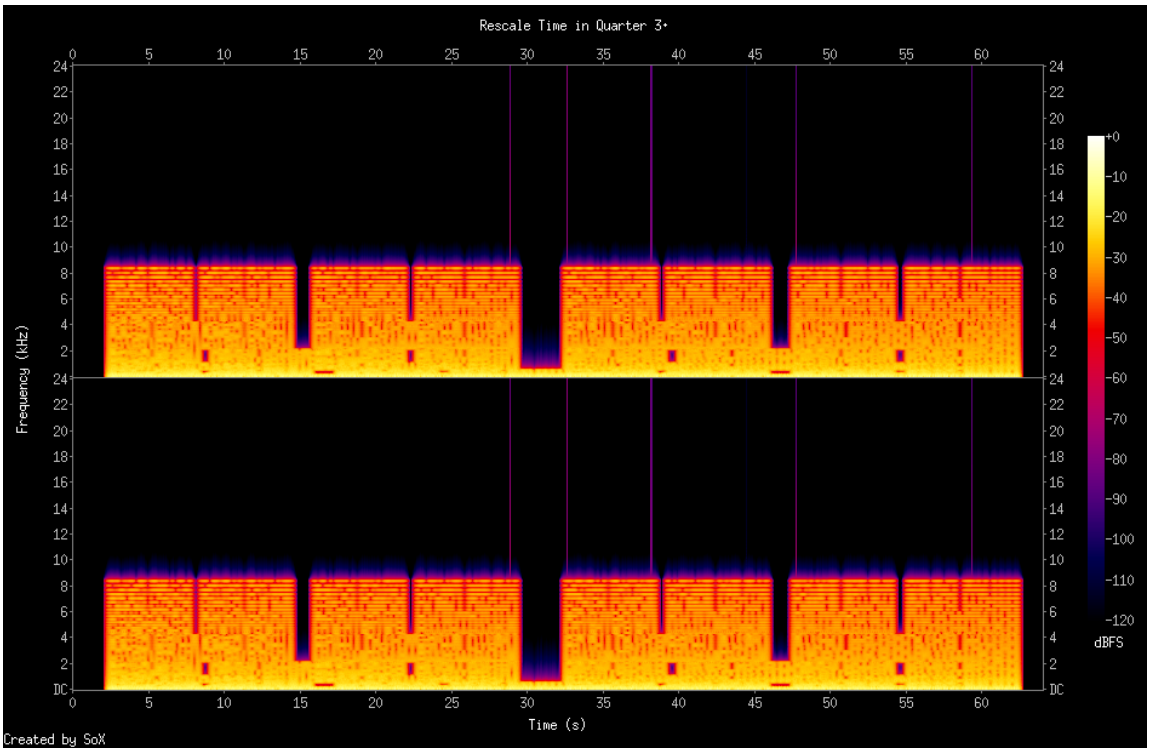
Listing 3: Rescale Time in Second Quarter.

```
/*                    i    t    d    k    v    p    T         */
hutchinson[1] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .45,   0,   0,   0,   0,   1, /* t */
                    0,   0,  .5,   0,   0,   0,   0, /* d */
                    0,   0,   0,  .5,   0,   0,   0, /* k */
                    0,   0,   0,   0,  .5,   0,   0, /* v */
                    0,   0,   0,   0,   0,  .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



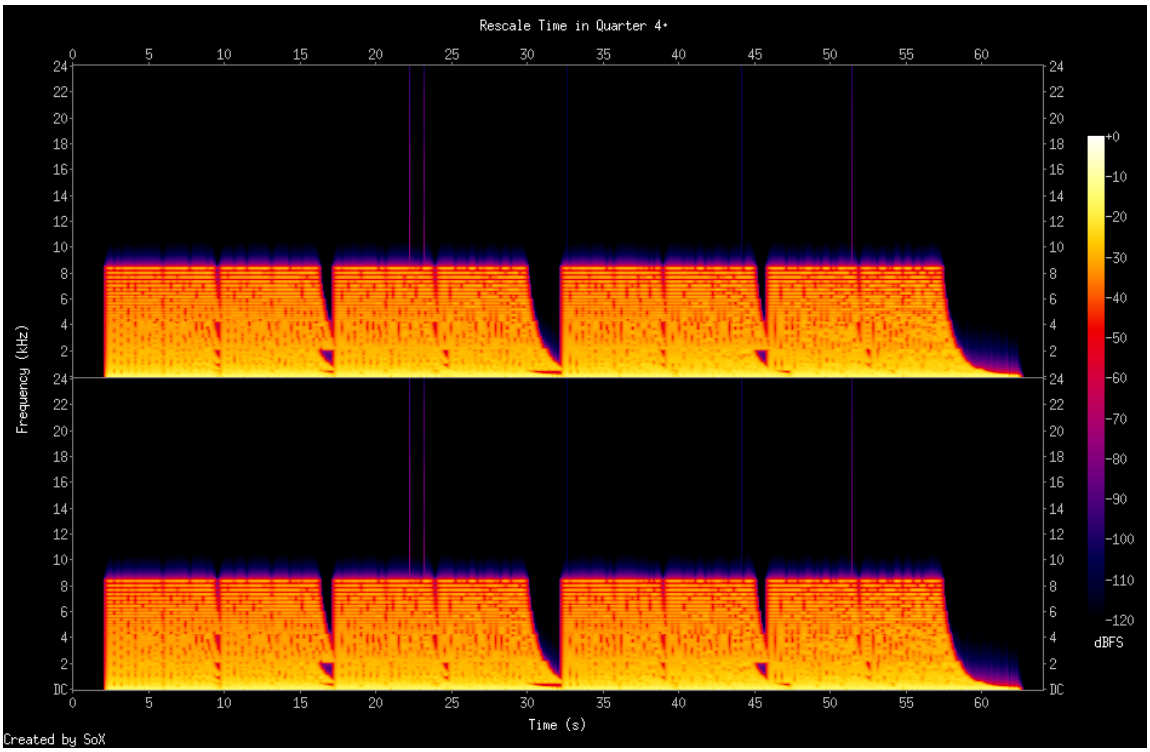
Listing 4: Rescale Time in Third Quarter.

```
/*                    i    t    d    k    v    p    T        */
hutchinson[2] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                   0, .45,   0,   0,   0,   0,   1, /* t */
                   0,   0,  .5,   0,   0,   0,   0, /* d */
                   0,   0,   0,  .5,   0,   0,   0, /* k */
                   0,   0,   0,   0,  .5,   0,   0, /* v */
                   0,   0,   0,   0,   0,  .5,   0, /* p */
                   0,   0,   0,   0,   0,   0,   1; /* H */
```



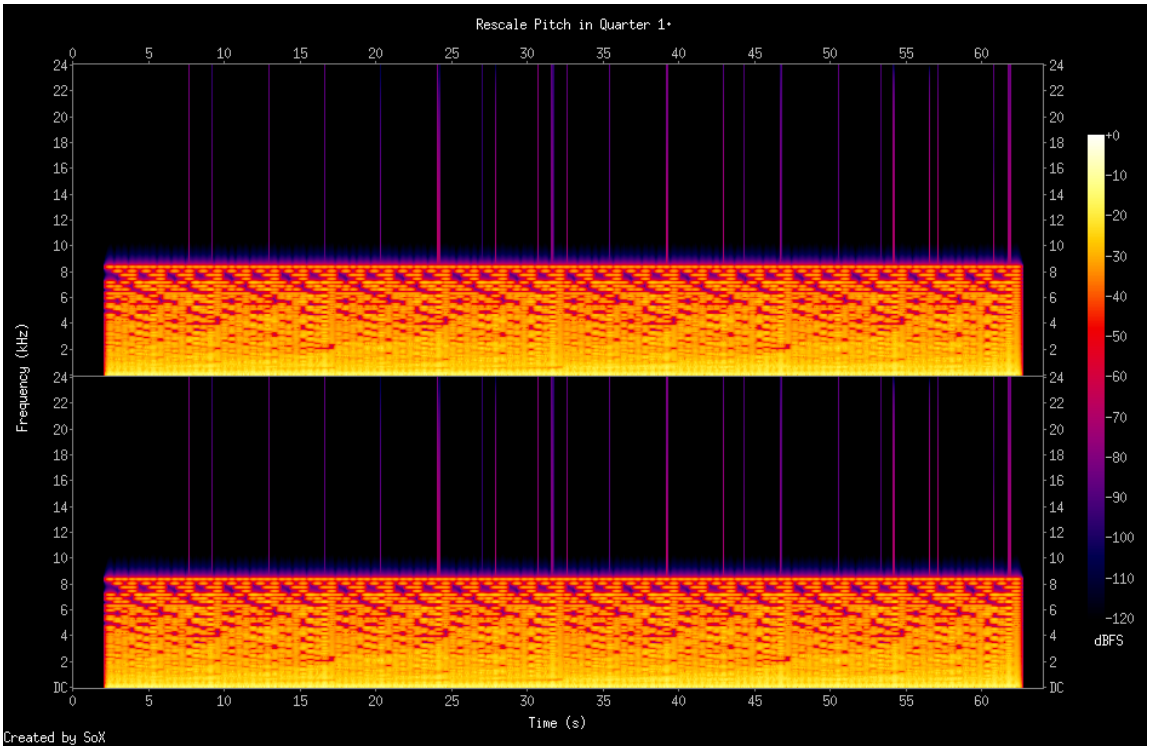
Listing 5: Rescale Time in Fourth Quarter.

```
/*                    i    t    d    k    v    p    T        */
hutchinson[3] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                   0, .45,   0,   0,   0,   0,   1, /* t */
                   0,   0,  .5,   0,   0,   0,   0, /* d */
                   0,   0,   0,  .5,   0,   0,   1, /* k */
                   0,   0,   0,   0,  .5,   0,   0, /* v */
                   0,   0,   0,   0,   0,  .5,   0, /* p */
                   0,   0,   0,   0,   0,   0,   1; /* H */
```



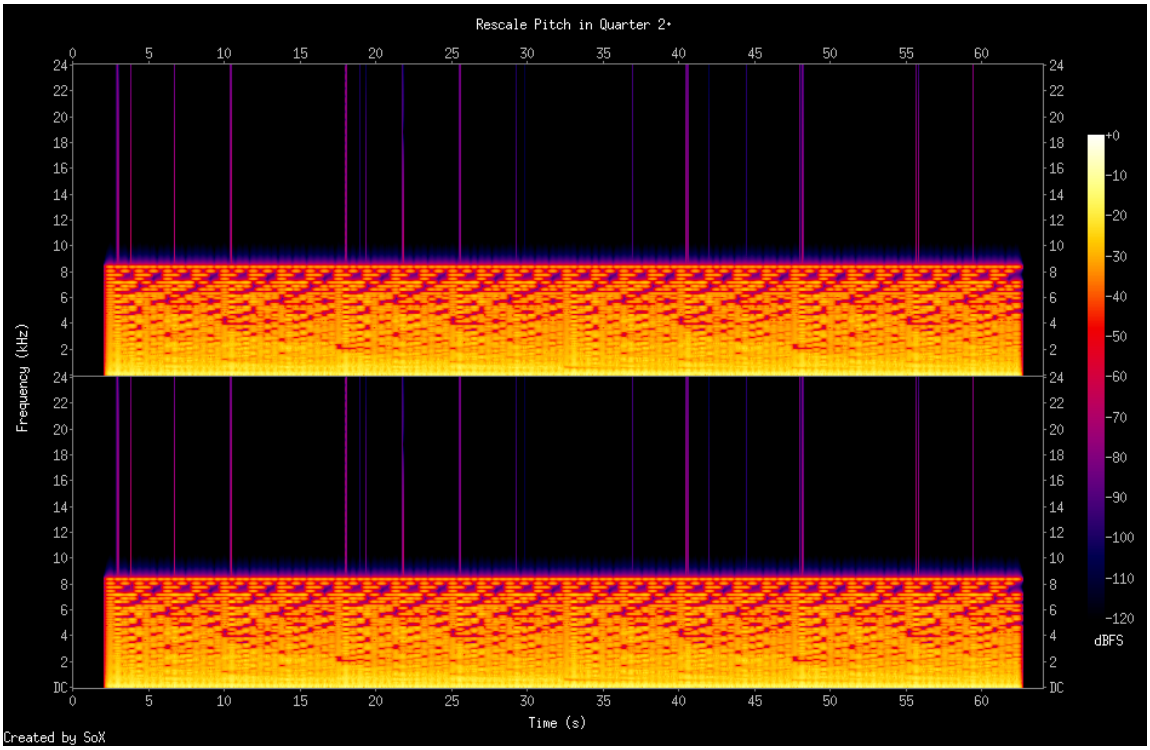
Listing 6: Rescale Pitch in First Quarter.

```
/*                      i    t    d    k    v    p    T        */
hutchinson[0] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .45,  0,   0,   0, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



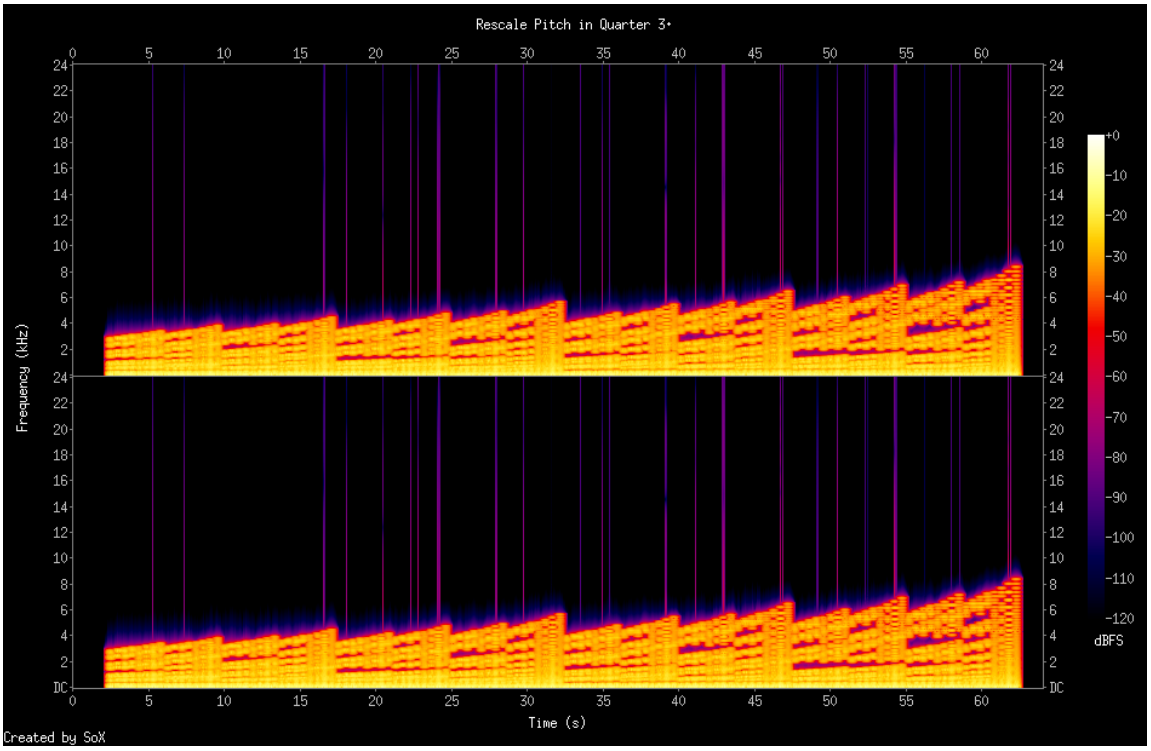
Listing 7: Rescale Pitch in Second Quarter.

```
/*                    i    t    d    k    v    p    T        */
hutchinson[1] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   1, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .45,  0,   0,   0, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



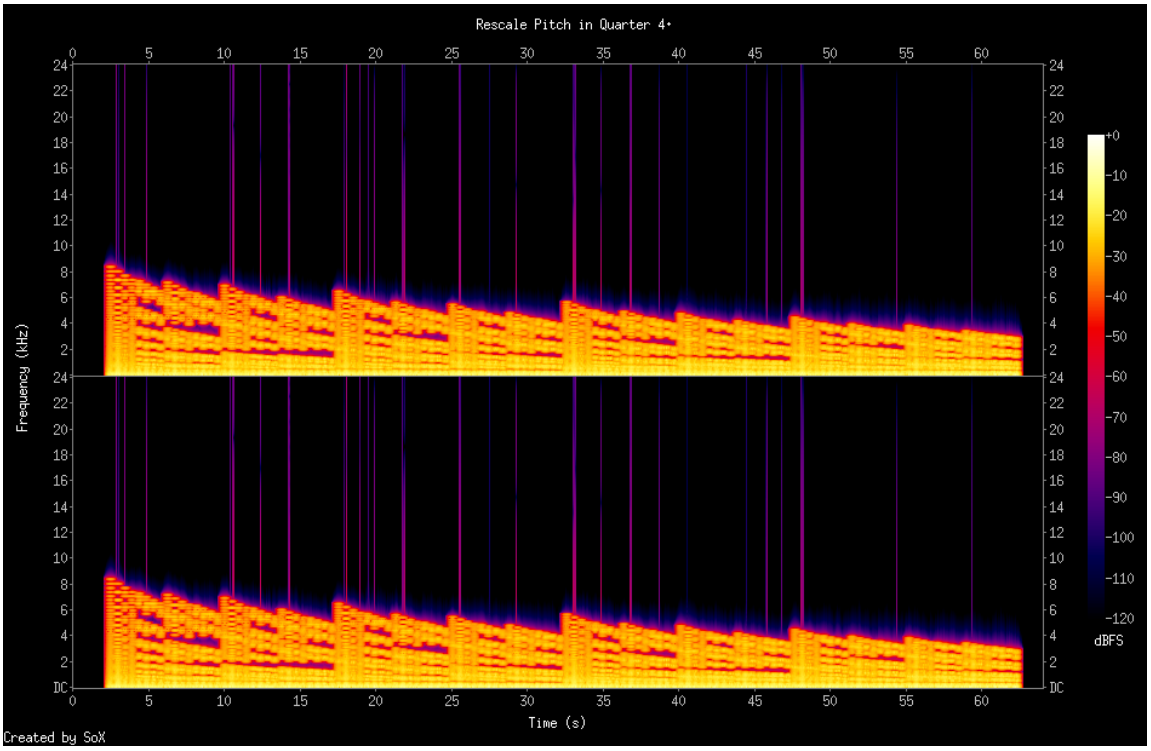
Listing 8: Rescale Pitch in Third Quarter.

```
/*                    i    t    d    k    v    p    T        */
hutchinson[2] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .45,  0,   0,   1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



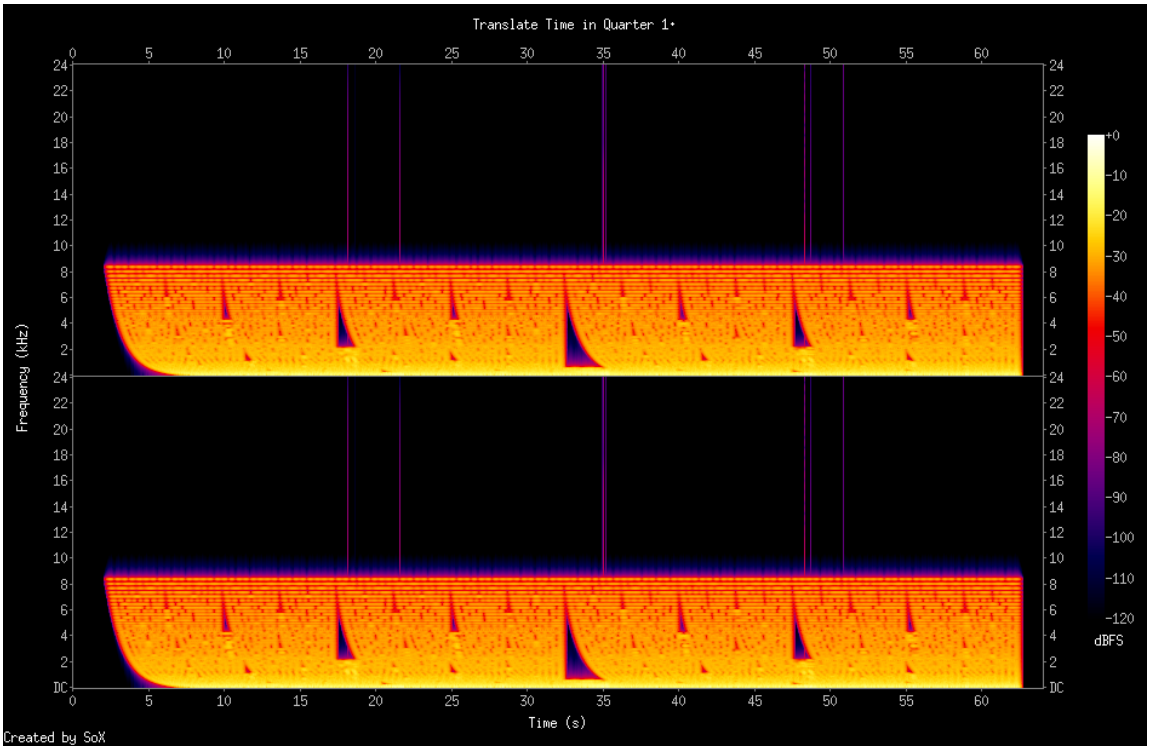
Listing 9: Rescale Pitch in Fourth Quarter

```
/*                    i    t    d    k    v    p    T         */
hutchinson[3] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0,  .5,   0,   0,   0,   0,   1, /* t */
                    0,   0,  .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .45,   0,   0,   1, /* k */
                    0,   0,   0,   0,  .5,   0,   0, /* v */
                    0,   0,   0,   0,   0,  .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



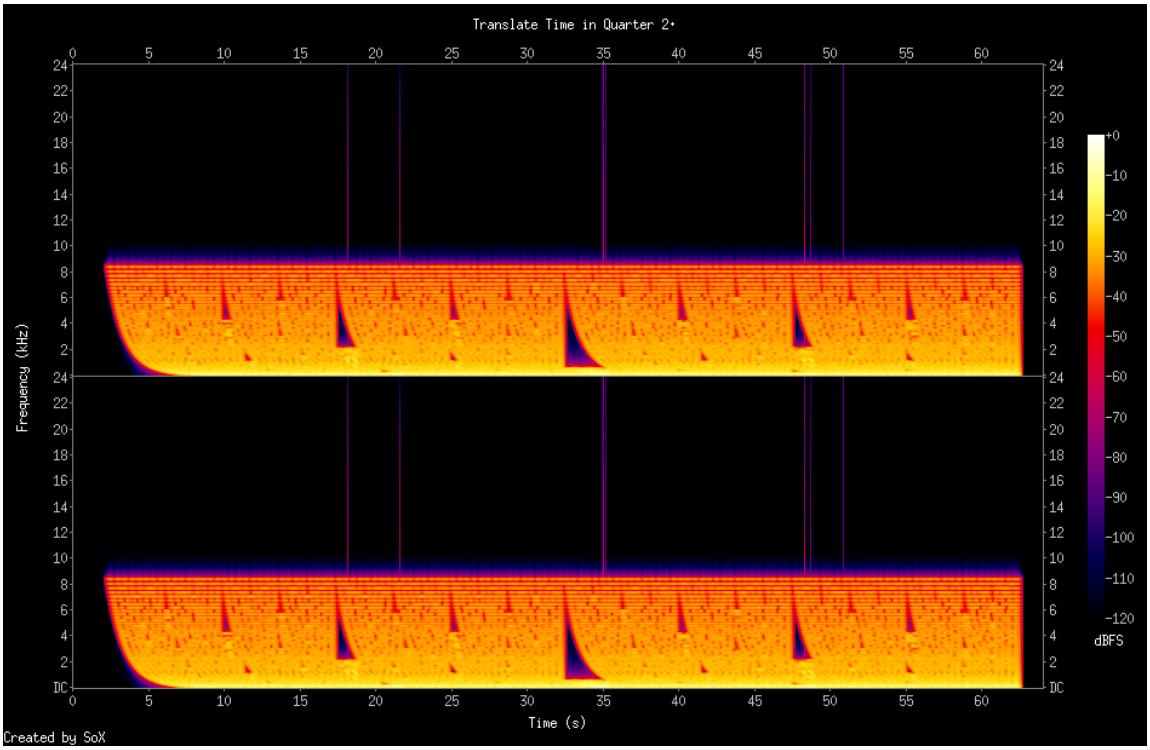
Listing 10: Translate Time in First Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[0] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,  -1, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   0, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



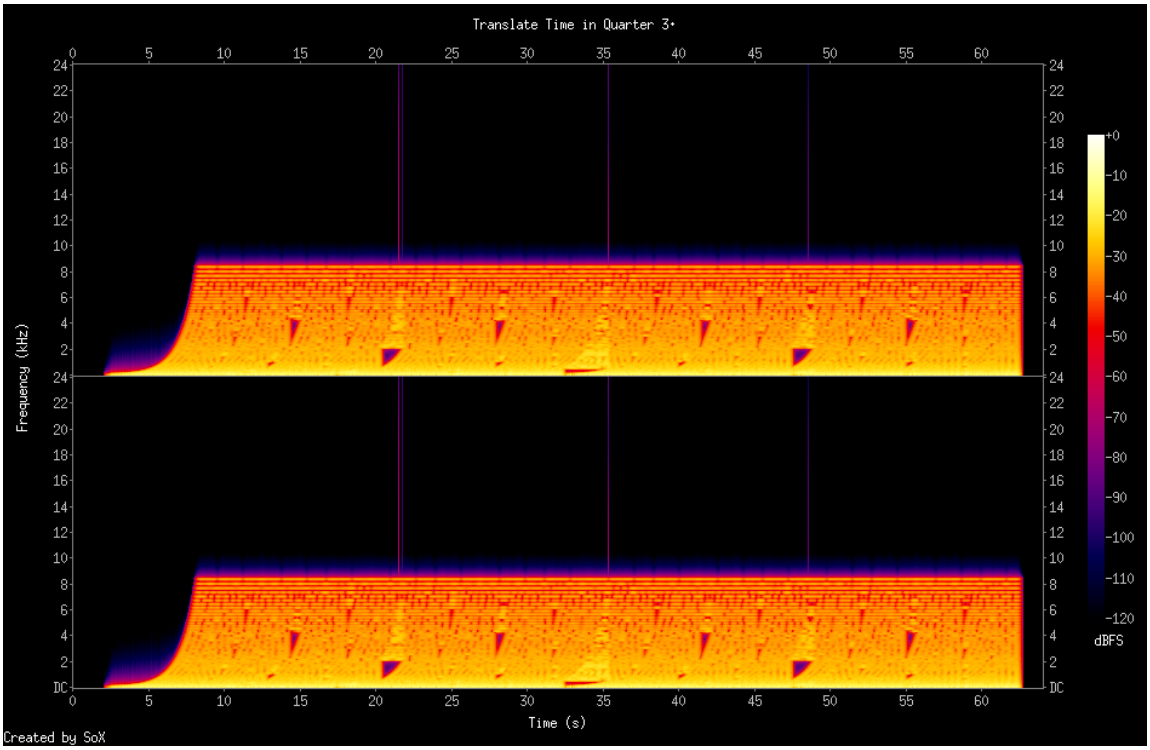
Listing 11: Translate Time in Second Quarter

```
/*                    i    t    d    k    v    p    T        */
 hutchinson[1] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                     0, .5,   0,   0,   0,   0,1.1, /* t */
                     0,   0, .5,   0,   0,   0,   0, /* d */
                     0,   0,   0, .5,   0,   0,   0, /* k */
                     0,   0,   0,   0, .5,   0,   0, /* v */
                     0,   0,   0,   0,   0, .5,   0, /* p */
                     0,   0,   0,   0,   0,   0,   1; /* H */
```



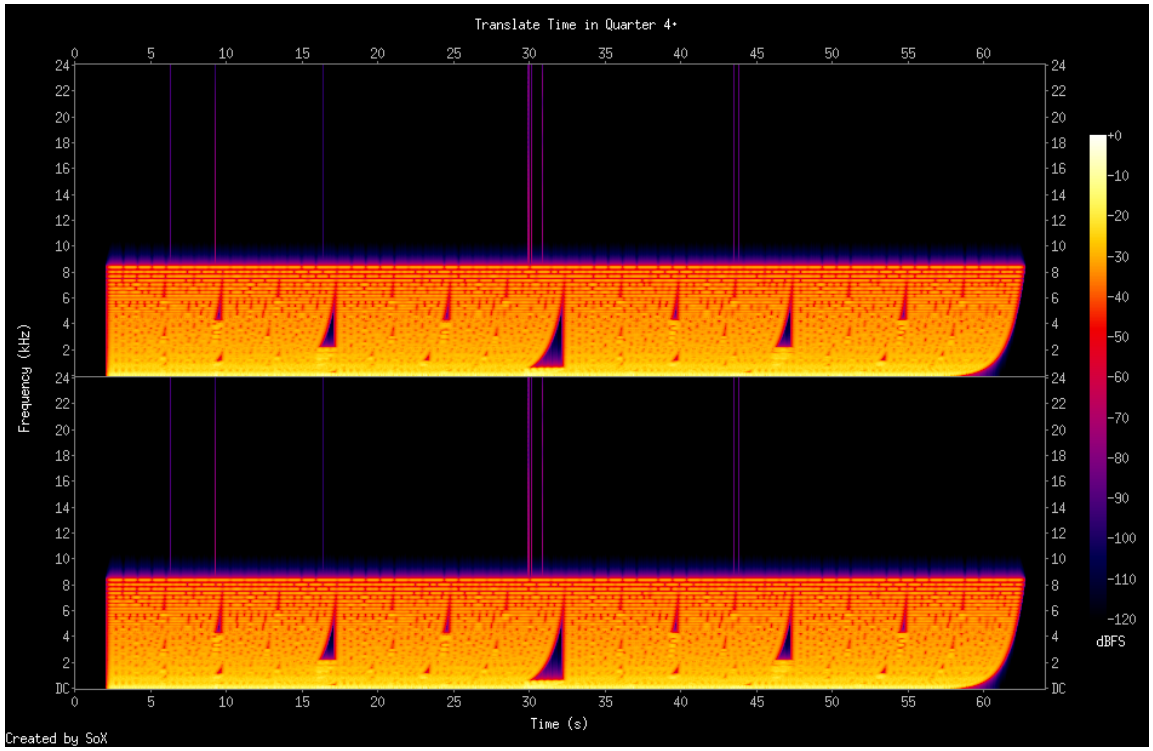
Listing 12: Translate Time in Third Quarter

```
/*                    i    t    d    k    v    p    T         */
hutchinson[2] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0, .1, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



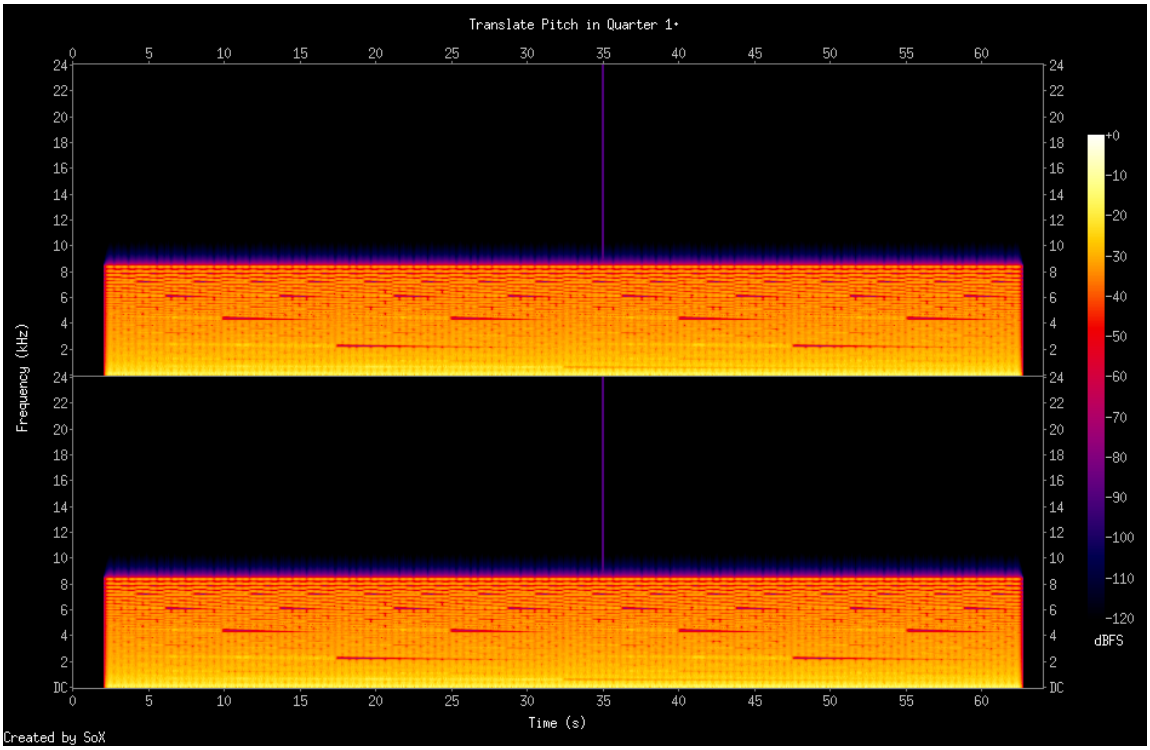
Listing 13: Translate Time in Fourth Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[3] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,1.1, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
```



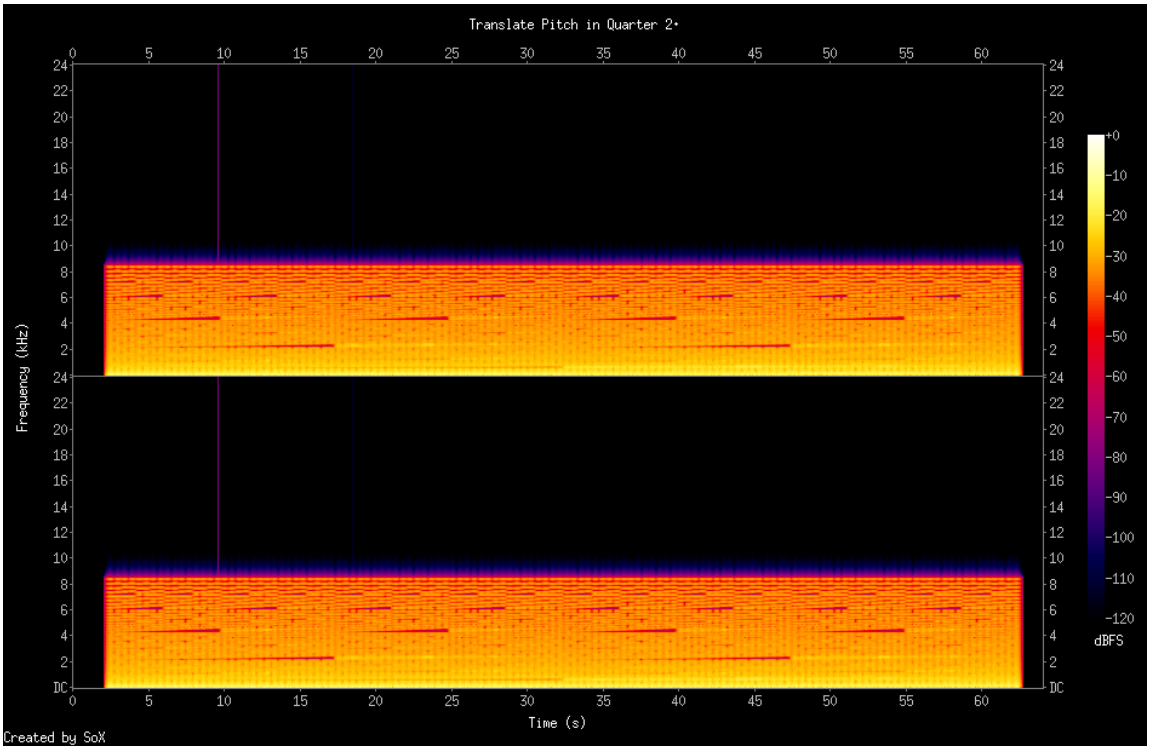
Listing 14: Translate Pitch in First Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[0] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0, .1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



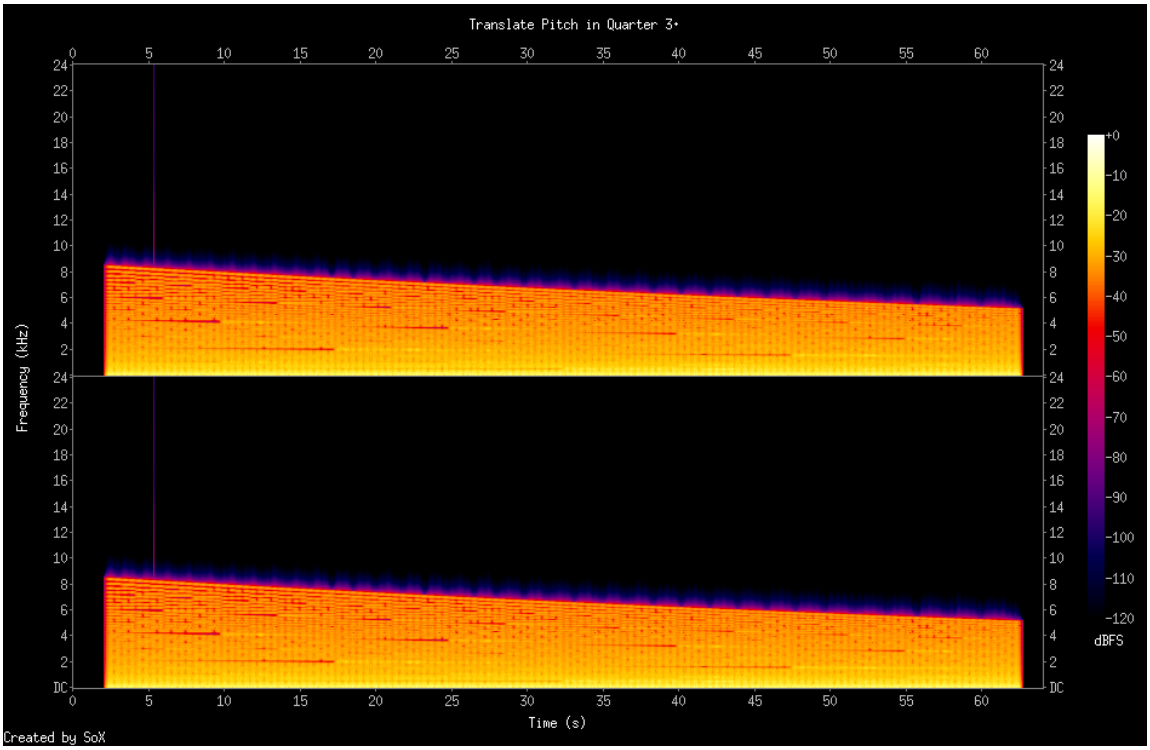
Listing 15: Translate Pitch in Second Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[1] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   1, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0, .1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



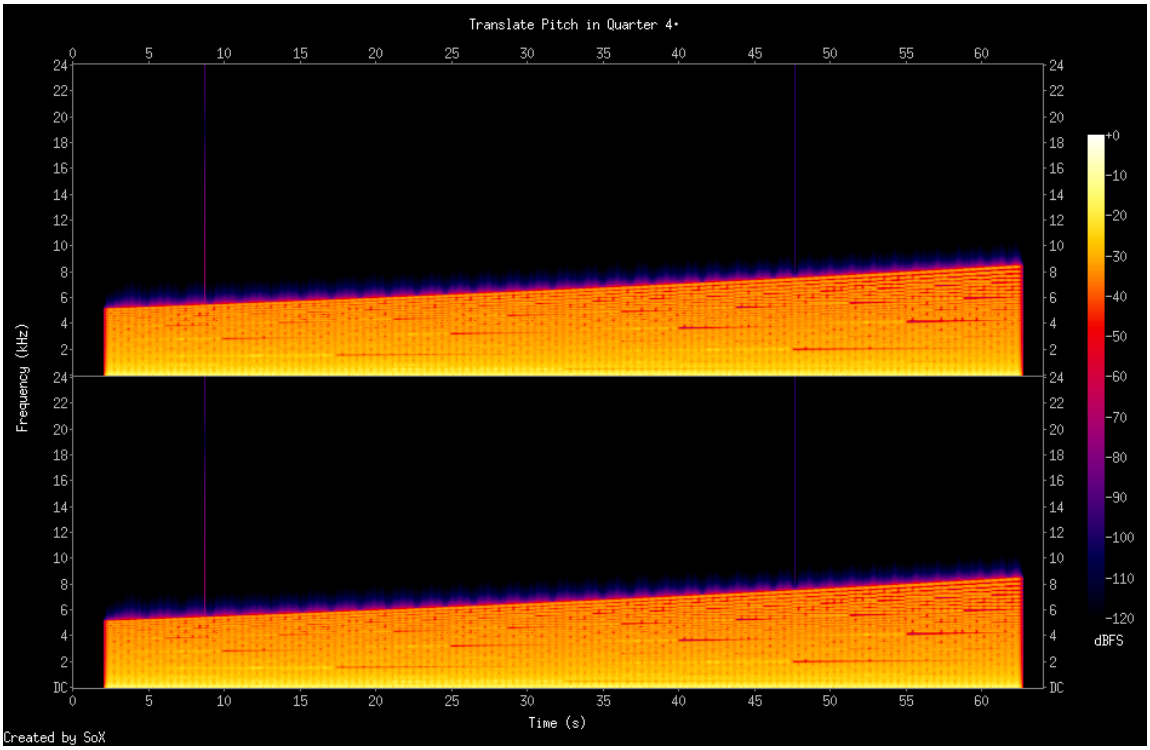
Listing 16: Translate Pitch in Third Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[2] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   1.1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



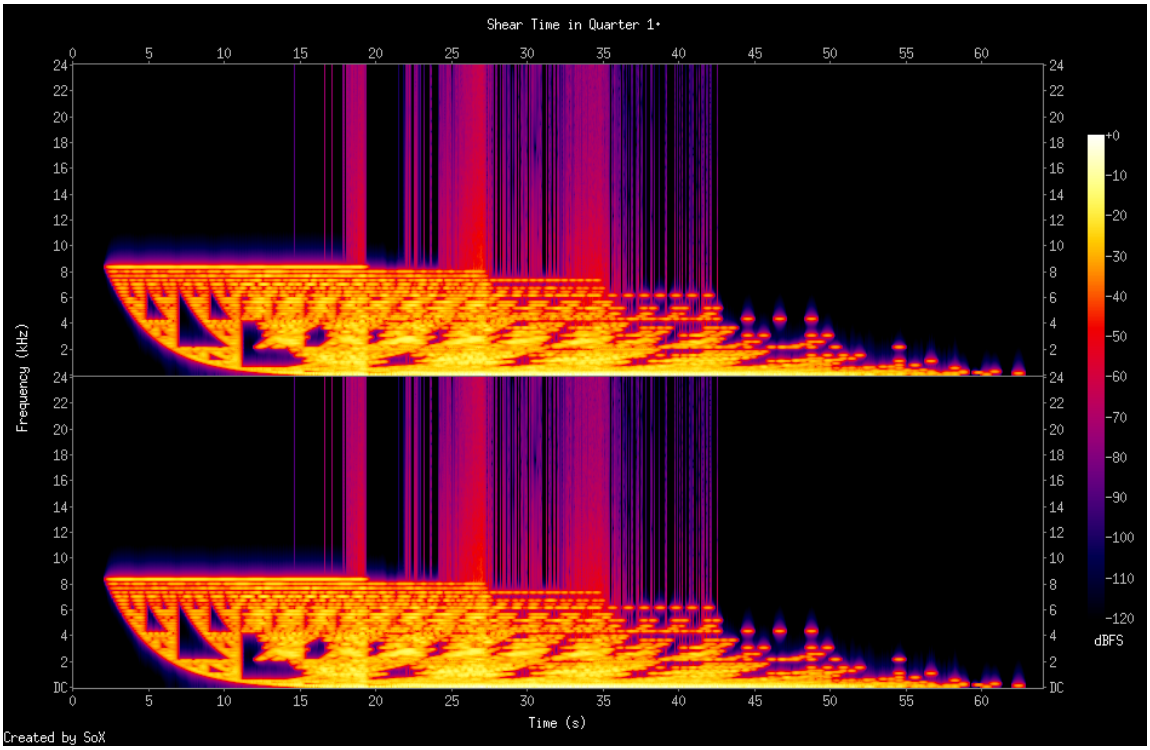
Listing 17: Translate Pitch in Fourth Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[3] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   1, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,1.1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



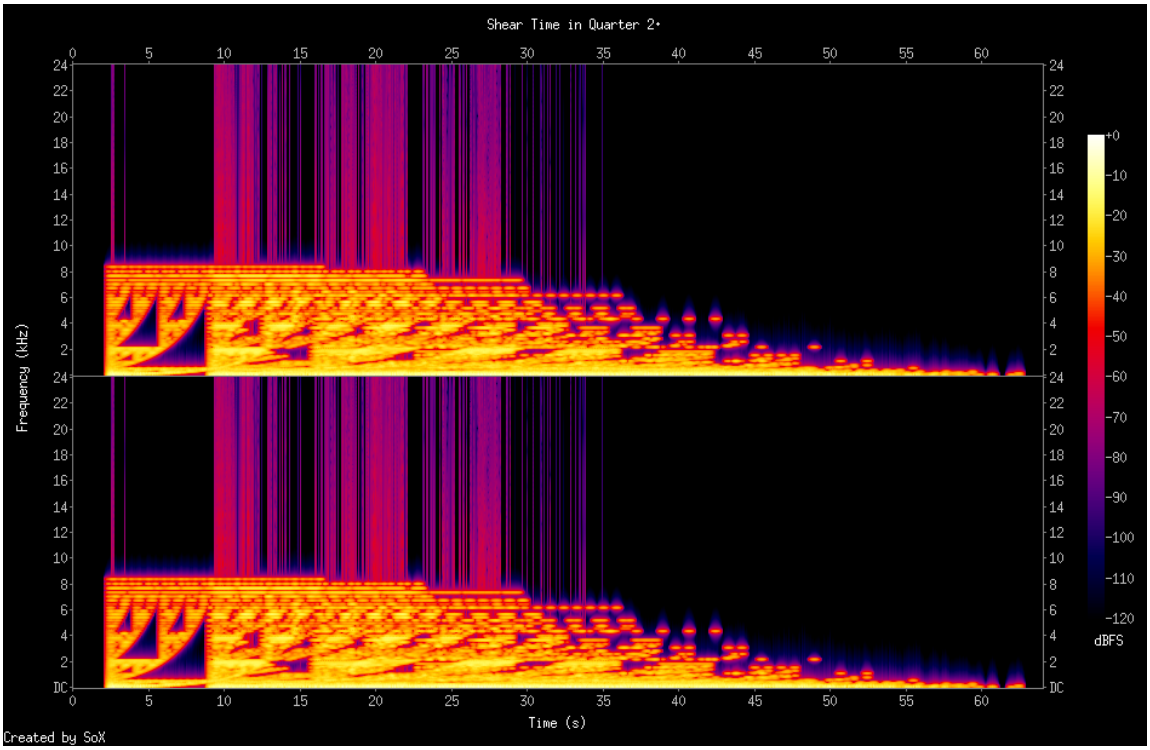
Listing 18: Shear Time in First Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[0] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   1,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   0, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



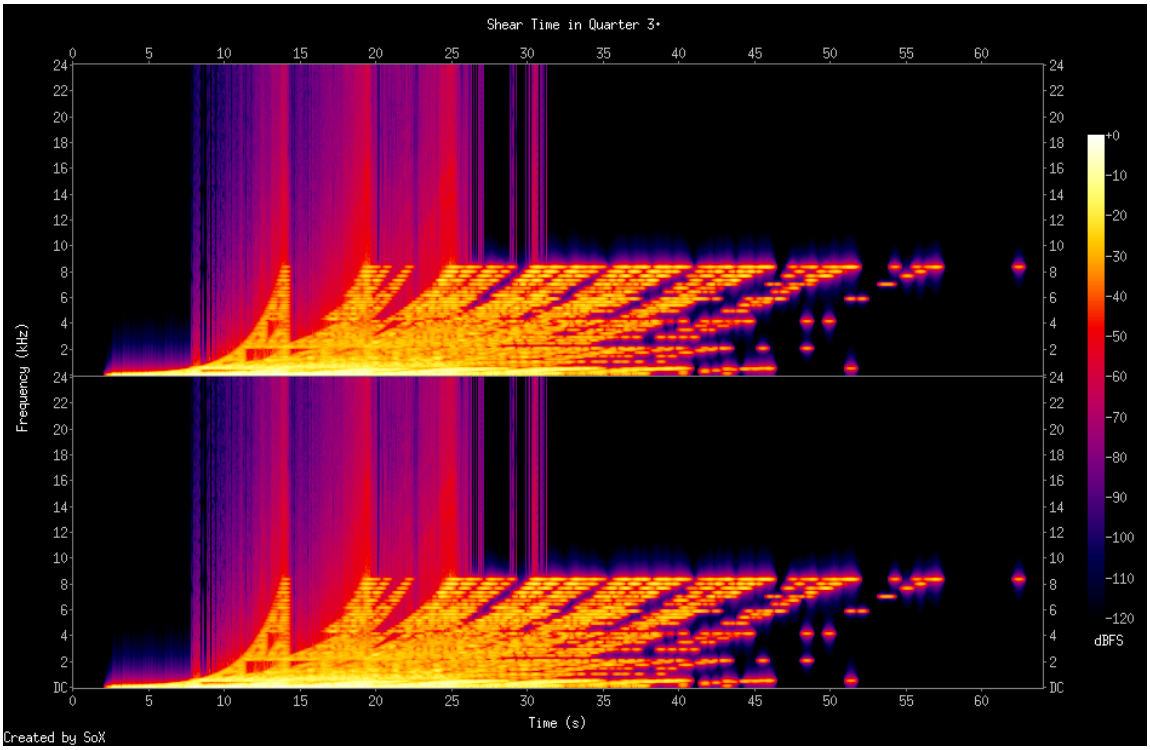
Listing 19: Shear Time in Second Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[0] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   1,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   0, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



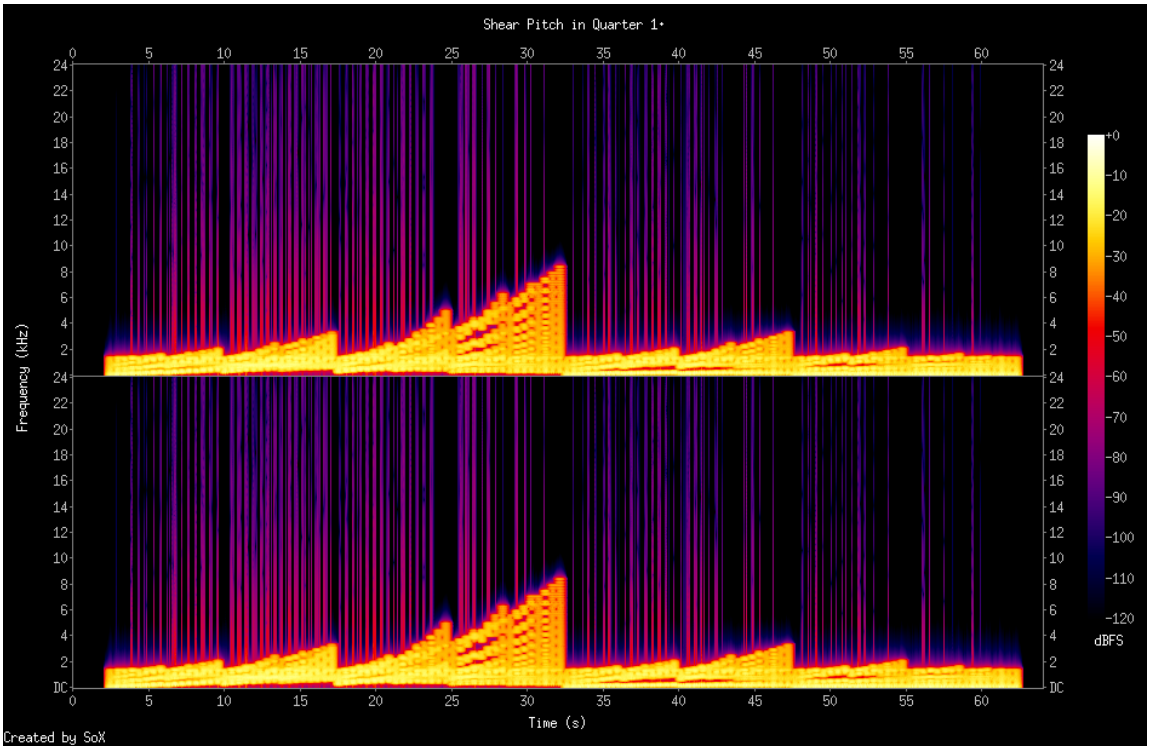
Listing 20: Shear Time in Third Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[2] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   1,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   0,   0, .5,   0,   0,   1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



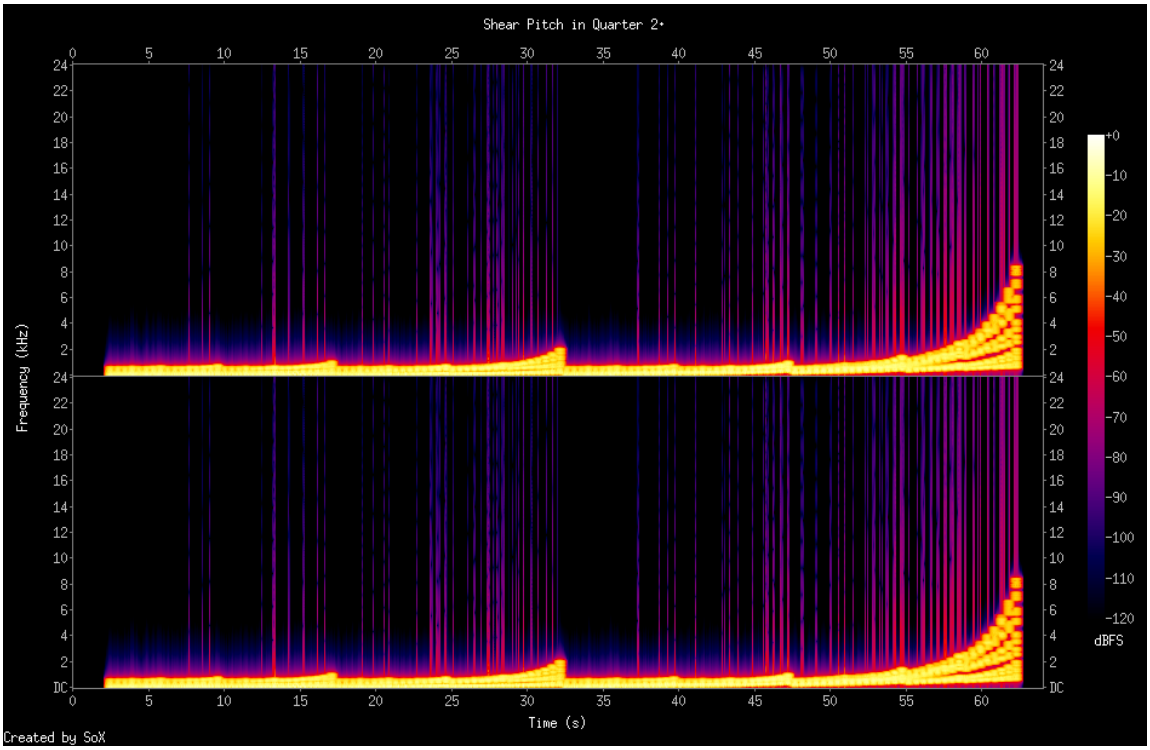
Listing 21: Shear Pitch in First Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[0] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   1,   0, .5,   0,   0,   0, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



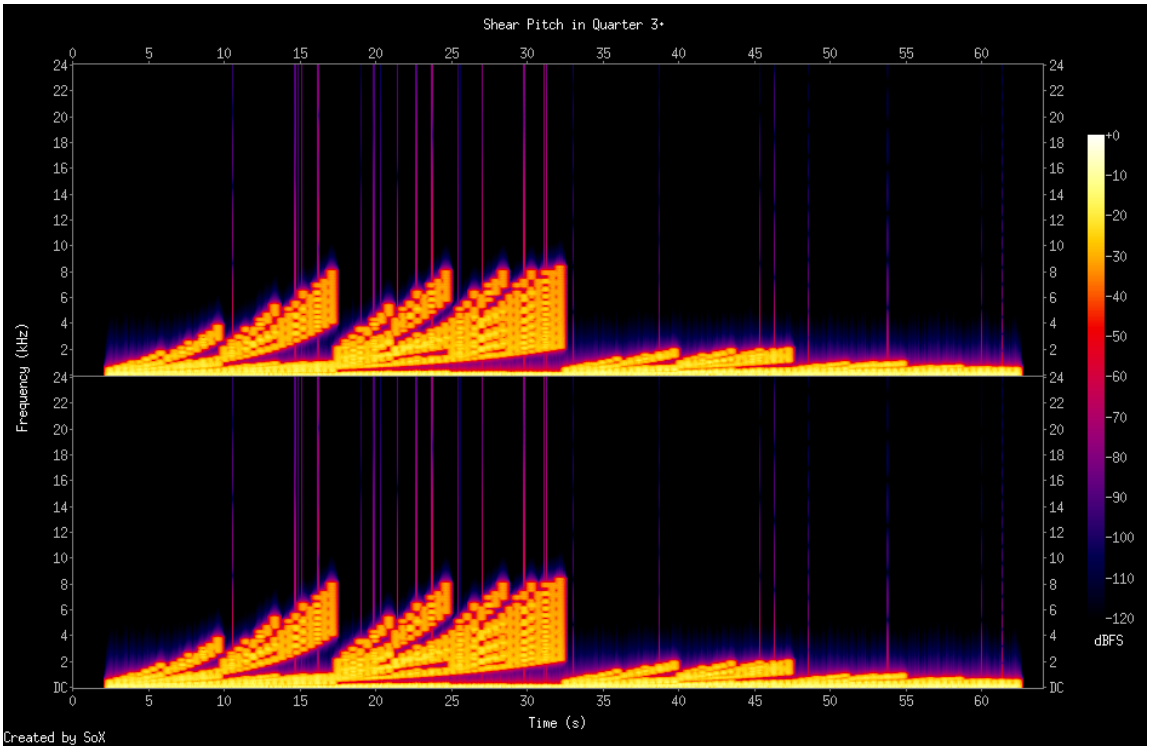
Listing 22: Shear Pitch in Second Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[0] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   1,   0, .5,   0,   0,   0, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



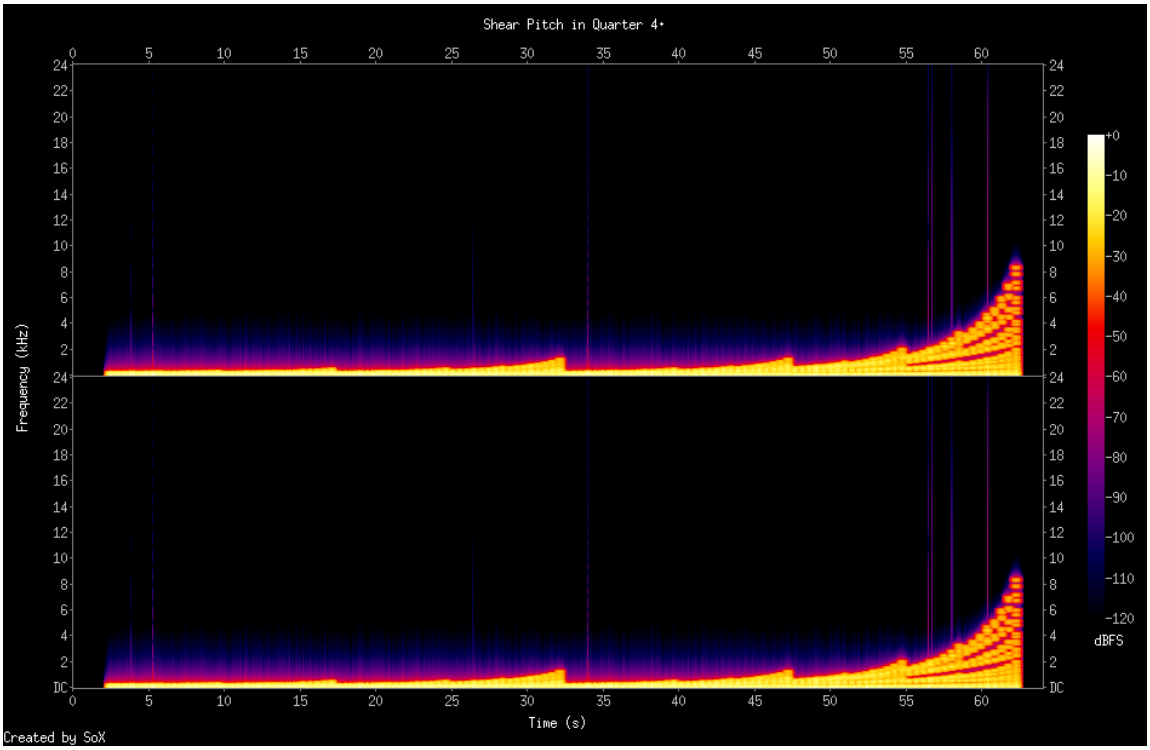
Listing 23: Shear Pitch in Third Quarter

```
/*                    i    t    d    k    v    p    T         */
hutchinson[2] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   1,   0, .5,   0,   0,   1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



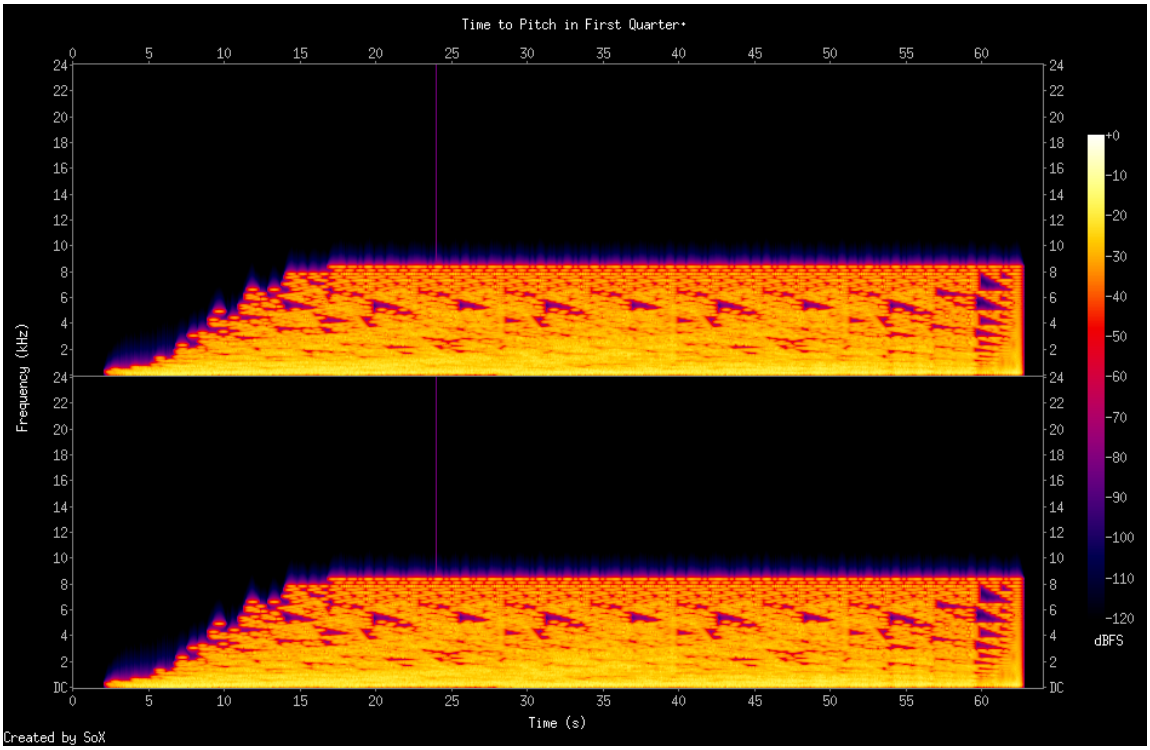
Listing 24: Shear Pitch in Fourth Quarter

```
/*                    i    t    d    k    v    p    T        */
hutchinson[2] <<   .5,   0,   0,   0,   0,   0,   0, /* i */
                    0, .5,   0,   0,   0,   0,   0, /* t */
                    0,   0, .5,   0,   0,   0,   0, /* d */
                    0,   1,   0, .5,   0,   0,   1, /* k */
                    0,   0,   0,   0, .5,   0,   0, /* v */
                    0,   0,   0,   0,   0, .5,   0, /* p */
                    0,   0,   0,   0,   0,   0,   1; /* H */
```



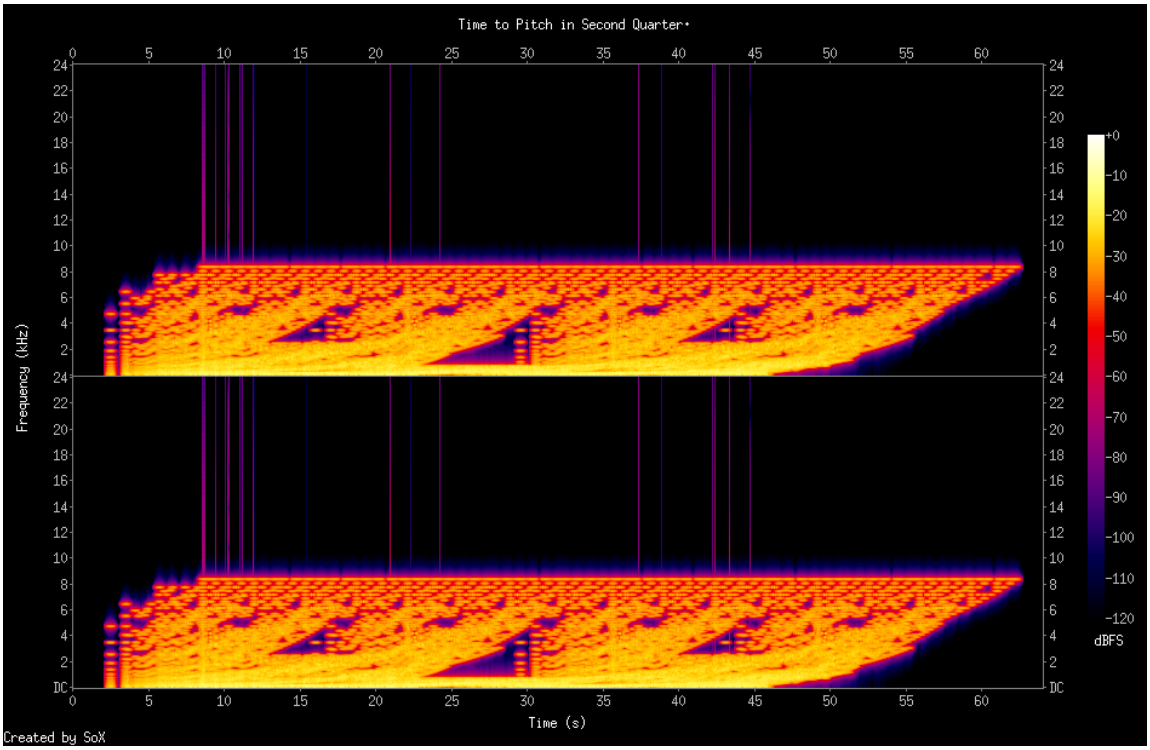
Listing 25: Time into Pitch in First Quarter

```
/*                    i    t         d    k         v    p    T        */
hutchinson[0] <<   .5,   0,        0,   0,        0,   0,   0, /* i */
                   0,   0.482,     0, -0.129,     0,   0,   0, /* t */
                   0,   0,       .5,   0,        0,   0,   0, /* d */
                   0,   0.129,     0,   0.482,    0,   0,   0, /* k */
                   0,   0,         0,   0,      .5,   0,   0, /* v */
                   0,   0,         0,   0,        0, .5,   0, /* p */
                   0,   0,         0,   0,        0,   0,   1; /* H */
```



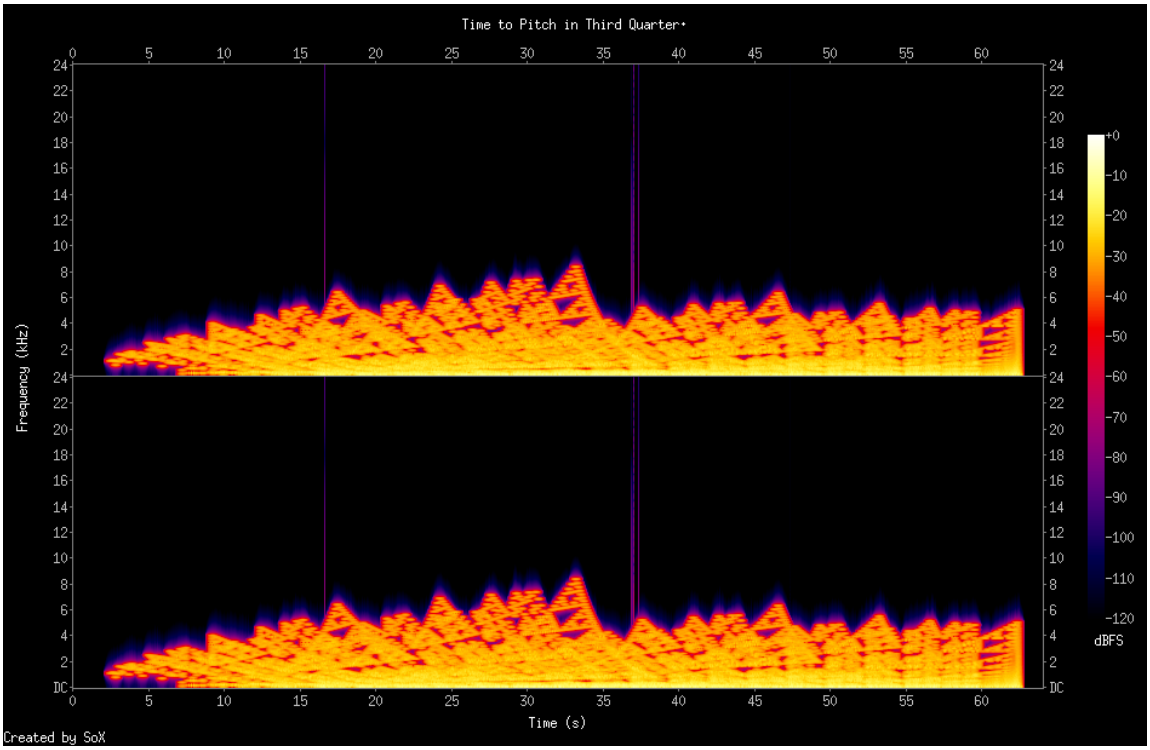
Listing 26: Time into Pitch in Second Quarter

```
/*                    i    t         d    k         v    p    T         */
hutchinson[1] <<   .5,   0,        0,   0,        0,   0,   0, /* i */
                   0,   0.482,     0, -0.129,     0,   0,   1, /* t */
                   0,   0,       .5,   0,        0,   0,   0, /* d */
                   0,   0.129,     0,   0.482,    0,   0,   0, /* k */
                   0,   0,         0,   0,      .5,   0,   0, /* v */
                   0,   0,         0,   0,        0, .5,   0, /* p */
                   0,   0,         0,   0,        0,   0,   1; /* H */
```



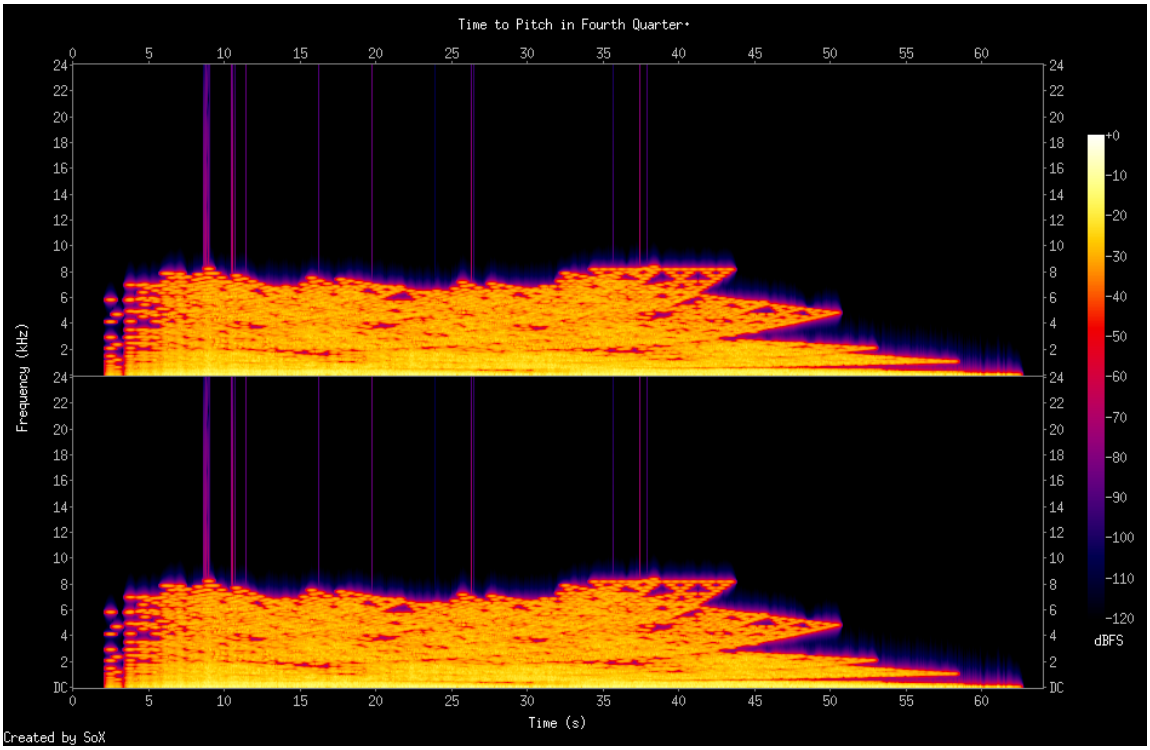
Listing 27: Time into Pitch in Third Quarter

```
/*                    i    t         d    k         v    p    T          */
hutchinson[2] <<   .5,   0,        0,   0,        0,   0,   0, /* i */
                   0,   0.482,     0,  -0.129,    0,   0,   0, /* t */
                   0,   0,       .5,   0,        0,   0,   0, /* d */
                   0,   0.129,     0,   0.482,    0,   0,   1, /* k */
                   0,   0,         0,   0,      .5,   0,   0, /* v */
                   0,   0,         0,   0,        0, .5,   0, /* p */
                   0,   0,         0,   0,        0,   0,   1; /* H */
```



Listing 28: Time into Pitch in Fourth Quarter

```
/*                    i    t         d    k         v    p    T         */
hutchinson[3] <<   .5,   0,        0,   0,        0,   0,   0, /* i */
                   0,   0.482,     0, -0.129,     0,   0,   1, /* t */
                   0,   0,       .5,   0,        0,   0,   0, /* d */
                   0,   0.129,     0,   0.482,    0,   0,   1, /* k */
                   0,   0,         0,   0,      .5,   0,   0, /* v */
                   0,   0,         0,   0,        0, .5,   0, /* p */
                   0,   0,         0,   0,        0,   0,   1; /* H */
```



[6] Barry Vercoe, John ffitch, et al. *Csound*. Available online at: http://csound.com. Accessed April 22, 2021. 2021 [1985] (cit. on p. 1).