

Notes on SYNTAL

Chapter 5: Changing Tempo and Other Things (V.00.2) or OSX

Wayne Slawson

May 21, 2004

How to Change the Tempo.

A New-Tempo Specification. You aren't stuck with the tempo that you specified in the `START` command. Anywhere in the SYNTAL input file, you can use the **mm** command to reset the tempo—or, equivalently, to redefine the internal parameter, *beat*. The **mm** command requires an argument with a decimal point, as do essentially all the commands that take numeric arguments.

You might have composed something using the **tv** command, for example:

```
/* My piece */
#include "/usr/local/bin/qmSYNTAL99.dfs"
/*****START HERE*****/

START(100.)
r( 0.60)
tv( 0.50,AA,AA, FF, P,P5P,DN3, Z, Z)
r( 2.10) /* 3.20 * b1 */
tv( 1.10,II,UU, FF, P,P5P,DN3, Z, Z)
r( 0.20) /* 4.50 */
r( 0.20)
tv( 0.90,EE,00, FF, P,P5P,DN3, Z, Z)
r( 0.50) /* 6.10 */
r( 0.80)
tv( 0.70,EH,AW, FF, P,P5P,DN3, Z, Z)
/* 7.6 */
END
```

But what you really want is for the tempo to pick up to 160 beats per minute just before the third **tv**. You could always change the duration of the events

themselves, but you may not wish to do that for various reasons. Instead you can just insert an **mm** command:

```
/*      My piece with a change of tempo      */
#include "/usr/local/bin/qmSYNTAL99.dfs"
/*****START HERE*****/

START(100.)
r(  0.60)
tv(  0.50,AA,AA, FF,  P,P5P,DN3,  Z,  Z)
r(  2.10)      /* 3.20 * b1 */
tv(  1.10,II,UU, FF,  P,P5P,DN3,  Z,  Z)
r(  0.20)      /* 4.50 */
r(  0.20)
mm(160.)
tv(  0.90,EE,00, FF,  P,P5P,DN3,  Z,  Z)
r(  0.50)      /* 6.10 */
r(  0.80)
tv(  0.70,EH,AW, FF,  P,P5P,DN3,  Z,  Z)
/* 7.6 */

END
```

One obvious way to use **mm** is to specify a rhythmic pattern with the durations in the events and then have varied repetitions at (perhaps drastically) different tempos. You just need to copy the pattern as many times as you want, and put the **mm** command, with the new tempo, before each repetition.

Remember: **mm** must have a number with a decimal point in it.

Gradual Changes in Tempo. You can also change tempos gradually over time something close to the way you change tempo in ordinary music with *accelerando* and *ritardando* markings. You do this with a single command, **glmm**. The **gl** comes from the musical term *glissando*, but in this case it is generalized to mean, not just a continuous change in pitch, but a continuous change in tempo (and, getting ahead a little, in a whole raft of other things—see below and in Chapter 6) or the “metronome mark”, **mm**.

The general form of all **glxx** commands is to supply two numbers, the first of which tells the duration in beats of the action, and the second of which specifies the *final* value of the parameter in question—in this case, the metronome marking—at the end of that duration.

Both numbers require decimal points.

Suppose we have a bunch of **tv** statements like those in the example above, but now instead of suddenly changing to a tempo of 160 beats per minute, we want gradually to increase the tempo to mm 160, starting just after the first **tv**, and continuing through the fourth and last **tv**. We’d write:

```

/*      My piece again      */
#include "/usr/local/bin/qmSYNTAL99.dfs"
/*****START HERE*****/

START(100.)
r(  0.60)
tv(  0.50,AA,AA, FF,  P,P5P,DN3,  Z,  Z)
glmm(6.5,160.)
r(  2.10)      /* 3.20 * b1 */
tv(  1.10,II,UU, FF,  P,P5P,DN3,  Z,  Z)
r(  0.20)      /* 4.50 */
r(  0.20)
tv(  0.90,EE,00, FF,  P,P5P,DN3,  Z,  Z)
r(  0.50)      /* 6.10 */
r(  0.80)
tv(  0.70,EH,AW, FF,  P,P5P,DN3,  Z,  Z)
      /* 7.6 */
/* end of the accelerando */
r(  0.80)
tv(  0.70,EH,AW, FF,  P,P5P,DN3,  Z,  Z)
r(  0.80)
tv(  0.70,EH,AW, FF,  P,P5P,DN3,  Z,  Z)
r(  0.80)
END

```

You'll notice that I added some copies of the last **tv** statement and the comment showing where the **glmm** stops having an effect—exactly 6.5 beats after the first **tv**. The following **tv** statement durations will *stay* in the new tempo of 160 beats per minute until you write another **glmm** or **mm** command.

How to Change Loudness

There are ways of changing the general levels of the various sources suddenly to a new level. You'd use the commands **setav**, **setas**, and **setaf**, to change the overall levels of the voiced, aspiration, and fricative sources. They are like **mm**; that is to say, they take a single number (with a decimal point—don't forget!) that sets the appropriate overall level to a new place. The number is interpreted in decibels, and it can be either positive—to increase the levels—or negative—to reduce them. The piece shown below gives an example of each.

You may also want to make continuous changes analogous to **glmm**. These are very hard to do in individual statements. The commands **glav**, **glas**, and **glaf**, allow you to change continuously the voiced, aspiration, and the fricative

sources, without changing the loudness parameters in each statement. These statements are used in the same way **glmm** is used; you say how long (in beats) the **glxx** statement is to be in effect, and where you want to be at the end of that time. The following example illustrates both the **setxx** and **glxx** commands.

```

/*      Another piece */
#include "/usr/local/bin/qmSYNTAL99.dfs"
/*****START HERE*****/

START(100.)
/* start everything at a low level */
    setav(-24.)
    setas(-18.)
    setaf(-18.)
glmm(18.,220.) /* start an accelerando */
tv( 0.50,AA,AA, FF,  P,P5P,DN3,  Z,  Z)
tv( 1.3,OE,OE, FF,  F,PPP,AF2,PPP, FF)
tv( 0.5,AW,EH,  F,  F,PPP,AF2,PPP, FF)
tv( 0.7,OO,EE, FF,  F,PPP,AF2,PPP, FF)
tv( 1.1,UU,II,  F,  F,PPP,AF2,PPP, FF)
r( 0.6) /* 4.2 beats */
glas(4.4,-3.0) /* start a crescendo in aspiration */
glaf(4.4, 0.) /* start a crescendo in affrication */
tv( 0.5,OE,OE,  F,  F,PPP,AF2,PPP, FF)
r( 0.4)
tv( 1.1,AW,EH, FF,  F,PPP,AF2,PPP, FF)
tv( 1.3,OO,EE,  F,  F,PPP,AF2,PPP, FF)
r( 0.2)
tv( 0.7,UU,II, FF,  F,PPP,AF2,PPP, FF)
r( 0.2) /* 4.4 */
glas(9.4,-24.0) /* start a diminuendo in aspiration */
tv( 1.1,OE,OE,  F,  F,PPP,AF2,PPP, FF)
r( 0.6)
tv( 0.7,AW,EH,  F,  F,PPP,AF2,PPP, FF)
tv( 0.5,OO,EE, FF,  F,PPP,AF2,PPP, FF)
r( 0.4)
tv( 1.3,UU,II, FF,  F,PPP,AF2,PPP, FF) /* 4.6 */
glav(9.6,0.0) /* start a crescendo in AV to nearly the end */
tv( 0.7,OE,OE,  F,  F,PPP,AF2,PPP, FF)
tv( 1.3,AW,EH, FF,  F,PPP,AF2,PPP, FF)
r( 0.6)
tv( 1.1,OO,EE,  F,  F,PPP,AF2,PPP, FF)
r( 0.6)

```

```

tv( 0.5,UU,II, FF,  F,PPP,AF2,PPP, FF) /* 4.8 */
glaf(4.8,-24.) /* start a diminuendo in affrication */
/* here the tempo gets to mm=220 and stays there */
tv( 0.7,OE,OE,  F,  F,PPP,AF2,PPP, FF)
tv( 1.3,AW,EH, FF,  F,PPP,AF2,PPP, FF)
r( 0.6)
tv( 1.1,OO,EE,  F,  F,PPP,AF2,PPP, FF)
r( 0.6)
tv( 0.5,UU,II, FF,  F,PPP,AF2,PPP, FF) /* 4.8 */
r(1.0)
END

```

Notice that you can change various of the parameters all at once (the **setav**, etc. statements) or at various times (the **glas**, etc. statements.) For the loudness you need to know that 0.0 dB is “normal”. Depending on how loud the dynamics are in each statement, you can increase the levels a little bit—say up to 3.0 dB—but if you go much higher you’ll get samples out of range. You can go as *low* in level as you like. Minus 24 dB is really quite quiet, but not exactly silence. There’s a bit of variation in the absolute values between AV, AS, and AF. It’s best to try them out and then adjust after you’d heard the first result.

Try these “gliss” commands; you’ll like them!