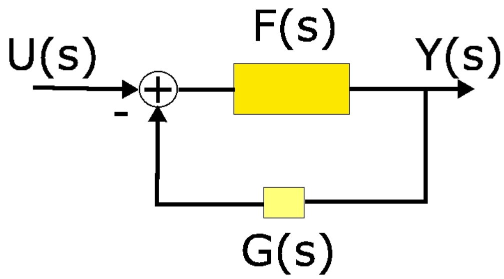

Figure 1: General feedback loop

Digital filter implementation of the QTA model for Mandarin F0 modeling

Reiner Wilhelms-Tricarico @ Fonix

last update: 4/19/2006

The F0 control mechanism in the work by Yi Xu's group[1] uses a 2nd order linear system in combination with a simple feedback control loop for F0 generation. The input to the system is either a piecewise constant function of time or a linearly sloping up or down ramp. In the proposed algorithm, the forward model, which generates the F0 contour, is specified as a second order linear dynamics system, described by a damping ratio ζ and characteristic frequency ω_0. Such a system is described by the following ordinary differential equation.

$$\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2 = c(t) \tag{1}$$

The right side, $c(t)$, would be some normalized control (with units of acceleration).

Written in the generalized frequency domain (Laplace notation) the system can be written as:

$$Y(s)(s^2 + 2\zeta\omega_0 s + \omega_0^2) = C(s) \tag{2}$$

If a second order system of this kind is augmented by a feedback control loop, a system as shown in Fig. 1 is obtained. This is similar to the original publication, but I am not using a factor or 2 for the input, as its purpose remains obscure. Starting from this diagram, the transfer function of the system, $H(s)$, in the Laplace domain is obtained as the rational function in s which relates the input $U(s)$, or control, to the output $Y(s)$.[2]

$$Y(s) = H(s)U(s) = \frac{F(s)}{1 + G(s)F(s)}U(s) \tag{3}$$

It appears that we are to identify the transfer function of the forward model, $F(s)$, as the second order system transfer function according to (2) defined as:

$$F(s) = \frac{1}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{4}$$

The feedback function used in Yi *et al*'s model, $G(s)$ is a delay by τ. In the Laplace domain, it has a representation of

$$G(s) = e^{-\tau s} \tag{5}$$

[1]Functional-oriented articulatory modeling of tones and intonations Santitham Prom-on, Yi Xu and Bundit Thipakorn

[2]How do we get this: In Fig. 1, consider the output of the adder which is the input to $F(s)$, and name it $X(s)$. The signal $X(s)$ goes through the feedback loop, and is as such self-referential, so we have: $X(s) = U(s) - G(s)F(s)X(s)$. From this follows $X(s) = U(s)/(1 + G(s)F(s))$ and from $Y(s) = F(s)X(s)$ follows (3).

0.1 Underpinnings of the model

First, without any approximations, let's see what this model does, or is supposed to do. Starting with (3) and using (4), as well as the representation of the delay by $e^{-\tau s}$, one arrives at the following system equation:

$$Y(s)(s^2 + 2\zeta\omega_0 s + \omega_0^2 + e^{-\tau s}) = U(s) \tag{6}$$

To see the behavior of such a system, it is again usefull to look at it in the time domain. There it corresponds to a 2nd order differential equation with delay:

$$\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2 y(t) = u(t) - y(t-\tau) \tag{7}$$

This system has the behavior of a damped mass spring system responding to the input consisting of the difference between a control signal $u(t)$ and the delayed state y. If ω_0, the characteristic frequency, is small the system behaves sluggish (large mass, weak spring), and if ω_0 is large, the system adjusts to the input $u(t)$ quickly.

However, it is not entirely clear if this is the system the authors originally intended.

In the following I assume that we are seeking a system like above that is driven by a goal function $g(t)$ which is designed such that the system $y(t)$ approaches $g(t)$, within the constraints of its dynamics. If $g(t)$ remains constant for a sufficiently long time at a value g_∞, we expect $y(t)$ also to settle at this value. The question is, what would be the input $u(t)$ to the model, and how does it relate to the goal function $g(t)$?

(a) Given the above model, if $u(t)$ is constant, namely u_∞, and the system has settled (accelerations and velocities have a approached zero), then we have the equation:

$$\omega_0^2 y_\infty + y_\infty = u_\infty$$

. Since we want y_∞ to become g_∞ we need to use the control

$$u_\infty = (\omega_0^2 + 1)g_\infty$$

to achieve the goal g_∞.

If we have in addition the goal that the velocity the system is prescribed, that is, after a long duration of applying a ramped control $g(t) = tb_\infty + const$, we expect the velocity of the system to become $\dot{y}_\infty = b_\infty$. Comparing velocities can be done by writing the system down for time t and $t + \tau$ and subtracting, while using a ramped input $u(t) = at + const$. This results in

$$\tau(\omega_0^2 + 1) = \tau a$$

and hence we recover the same type of scaling as for the stationary case.

(b) The above may not be what was intended, since we would need some funny scaling of the input, to go from the goal function $g(t)$ to the input of the system. So let's try another one. What if the authors mean a system like the following:

$$\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2 y(t) = \omega_0^2(u(t) - y(t-\tau)) \tag{8}$$

Using similar arguments as before, for constant input u_∞ it would settle as $\omega_0^2 y_\infty = \omega_0^2(u_\infty - y_\infty)$, or $u_\infty = 2y_\infty$.

A question I have is if this is where the funny factor 2 actually came from.

(c) Finally, is there any advantage to use one of the systems (a) or (b) over the following:

$$\ddot{y} + 2\zeta\omega_0\dot{y} = \omega_0^2(u(t) - y(t-\tau)) \tag{9}$$

This system is behaves, like the other ones, as a damped mass-spring system, however, the input can be directly interpreted as the goal function $u(t)$, and no other scaling is required. It can be easily seen that for constant input u_∞ the output settles at $y_\infty = u_\infty$, and as well, for a ramp $u(t) = const + at$ it will settle at with a constant velocity $\dot{y} = a$.

From these considerations, I decided that the version (c) is probably what the authors intended. Now let's get to the implementation as digital filter, starting with the representation in the Laplace domain:

$$Y(s)(s^2 + 2\zeta\omega_0 s + \omega_0^2 e^{-\tau s}) = \omega_0^2 U(s) \tag{10}$$

Remark: I convinced myself that this system is stable, simply by using typical values of the coefficients and calculating the poles numerically (it is possible to calculate the poles analytically for a third order polynomial but the algebra is daunting). For the analogue system (in the Laplace domain) the real parts of the poles are all negative, and for the below investigated digital representation, they are inside the unit circle. So it's stable, and the implementation hasn't blown up thus far, which is the proof in the pudding.

0.2 Representation as digital filter

The goal is now to convert the above system into a digital filter for numerical iteration. A straightforward way to do this, is to replace the Laplace variable s by the bilinear transform, which is a two-point approximation of the differentiation operator (whose Laplace transform is s):

$$s \longrightarrow \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}} \tag{11}$$

where T is the sampling rate (time between sampling points).

Considering the feed-back delay with Laplace representation $G(s) = e^{-\tau s}$, if we choose τ a multiple of the sampling rate T - the authors used $\tau = 5ms$ - the delay represented in digital filter is by means of a simple delay operator, described in the z-transform domain as z^{-1}. This may be behind the explanation that the Padé approximation of the delay operator is being used. In particular, the [1,1] Padé approximation of the delay is:

$$\exp(-\tau s) \approx \frac{1 - \frac{\tau s}{2}}{1 + \frac{\tau s}{2}} \tag{12}$$

Using the above bilinear transform[3], and the special case $\tau = T$, it can easily be seen that the approximated delay operator becomes:

$$\exp(-Ts) \approx \frac{1 - \frac{Ts}{2}}{1 + \frac{Ts}{2}} = \frac{1 - \frac{1-z^{-1}}{1+z^{-1}}}{1 + \frac{1-z^{-1}}{1+z^{-1}}} = z^{-1} \tag{13}$$

Longer delays, by m sampling periods, are accordingly represented as z^{-m}. It is also possible to make intermediate delays with delay smaller than the sampling period, which in principle allows to form any delay by combination of partial and integer delays.

[3] which by itself is a consequence of approximating the inverse of the mapping $z = e^{sT}$, or $s = \frac{1}{T}\log z$ by a [1,1]-Padé approximation about $z = 1$

For now, let's stay with the simple delay, z^{-1}. The following filter can be obtained, from (10), using the bilinear transform (11) and the digitial approximation of the delay:

$$B(z)Y(z) = A(z)U(z) \tag{14}$$

with:

$$B(z) = 1 + \frac{T^2\omega_0^2 - 8}{4T\zeta\omega_0 + 4}z^{-1} + \frac{2\left(T^2\omega_0^2 - 2T\zeta\omega_0 + 2\right)}{4T\zeta\omega_0 + 4}z^{-2} + \frac{T^2\omega_0^2}{4T\zeta\omega_0 + 4}z^{-3} \tag{15}$$

and

$$A(z) = \frac{T^2\omega_0^2}{4T\zeta\omega_0 + 4}(1 + z^{-1})^2 \tag{16}$$

So for implementation purposes, we write the filter:

$$Y(z)(1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}) = a_0(1 + 2z^{-1} + z^{-2})U(z) \tag{17}$$

$$\begin{aligned} d &= 4T\zeta\omega_0 + 4 \\ a_0 &= T^2\omega_0^2/d \\ b_1 &= a_0 - 8/d \\ b_2 &= 2a_0 - (4T\zeta\omega_0 - 4)/d \\ b_3 &= a_0 \end{aligned}$$

And implemented it as the recursion:

$$y_n = a_0(u_n + 2u_{n-1} + u_{n-2}) - b_1 y_{n-1} - b_2 y_{n-2} - b_3 y_{n-3} \tag{18}$$

0.3 Tricks and fixes

In the application I used the original values from the tables but eventually decided to divide the slopes in half. Using the original slopes in this implementation resulted in too large extensions of the pitch contour, exceeding the usual phyisiological range.

Actually, I also halfed the offsets, since I believe that the factor of two is not necessary. However, later I also shifted the offsets somewhat, as for example the 4th tone always appeared too high. I also modified some of the ω_0-s from the table, giving the raising and the falling tone sharper tuning with higher characteristic frequency.

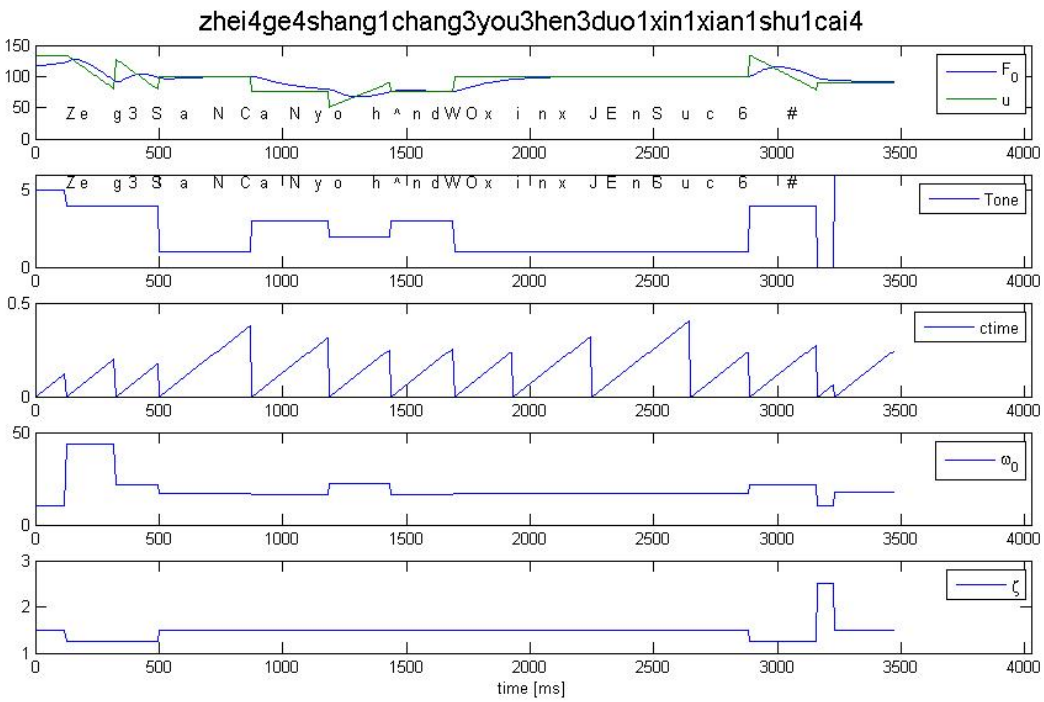

Figure 2: In this utterance, the first syllable received a high characteristic frequency to force the contour filter to immediately follow the control ramp.

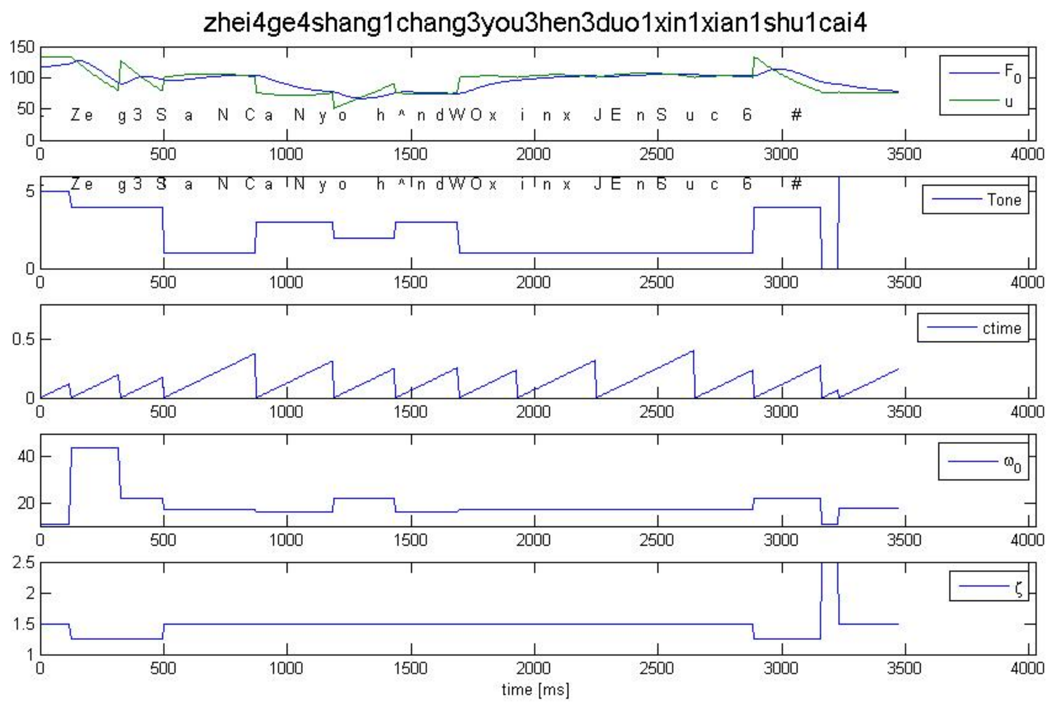

Figure 3: Additional tricks: The final lengthening pause is modeled as slightly falling ramp. Further for the high tone, the low tone, and the falling tone, small parabolas are added to the control. For multiple high tones, this results in a modulation of F0.