

Figure 1: General feedback loop

Digital filter implementation of the QTA model for Mandarin F0 modeling

Reiner Wilhelms-Tricarico @ Fonix

last update: 4/06/2006

The F0 control mechanism in the work by Yi Xu's group¹ 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. The system as shown in the original publication and similarly in Fig. 1 above is described by the following transfer function, $\mathbf{H}(s)$, in the Laplace domain:

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

In the proposed algorithm the second order linear dynamics system is described as a linear second order system. This would be:

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

where $\mathbf{c}(t)$ would be some normalized control (with units of acceleration). The transfer function $\mathbf{F}(s)$ is therefore:

$$\mathbf{F}(s) = \frac{1}{s^2 + 2\zeta\omega_0s + \omega_0^2} \quad (3)$$

0.1 Underpinnings of the model

From an engineering view point the model is a linear 2nd order system with delayed feedback, which makes it at minimum a third order model if the feedback is only one sample time. The model's underpinnings are best understood when writing it down in the time domain.

The feed-back transfer function $\mathbf{G}(s)$ in the original work is a delay by τ , which would be by its Laplace transform: $\exp(-\tau s)$. If we rewrite the above transfer function as $\mathbf{F}(s) = 1/\mathbf{B}(s)$, the complete input output relation in the Laplace domain is:

$$\mathbf{Y}(s) = \mathbf{H}(s)\mathbf{U}(s) = \frac{\frac{1}{\mathbf{B}(s)}}{1 + \frac{e^{-\tau s}}{\mathbf{F}(s)}}\mathbf{U}(s) \quad (4)$$

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

Hereby $\mathbf{Y}(s)$ and $\mathbf{U}(s)$ refer to the Laplace transformations of the output $\mathbf{y}(t)$ and input $\mathbf{u}(t)$ of the model, and the delay operator $\mathbf{G}(s)$ is written in not approximated form. This can be rewritten as:

$$(\mathbf{B}(s) + e^{-\tau s})\mathbf{Y}(s) = \mathbf{U}(s) \quad (5)$$

This system corresponds to the 2nd order differential equation with delayed input:

$$\ddot{\mathbf{y}} + 2\zeta\omega_0\dot{\mathbf{y}} + \omega_0^2\mathbf{y} = \omega_0^2(\mathbf{u}(t) - \mathbf{y}(t - \tau)) \quad (6)$$

The “physical” meaning of this the behavior of a damped mass-spring system that is pushed by a force proportional to the difference between the control input and the recently observed state of the system. If ω_0 , the characteristic frequency is small, the system is sluggish (large mass, weak spring), and if ω_0 is large, the system follows adjusts to the input $\mathbf{u}(t)$ quickly. The damping coefficient ζ needs to be chosen so that the resulting system is stable. This is the case as long as ζ is bigger than 1. The authors of the method used $\zeta = 1.5$.

So we are dealing with an inhomogenous linear 2nd order system to which the input is the difference between an externally specified control signal $\mathbf{u}(t)$ and the state \mathbf{y} of the second order system delayed by τ . Hence, the forcing function that enters the “damped mass-spring system” is a (normalized) force obtained by taking the difference between

The control $\mathbf{u}(t)$, in this model, specifies goal trajectory of F0.

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):

$$s \rightarrow \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (7)$$

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

Considering the feed-back delay with Laplace representation $\mathbf{G}(s) = e^{-\tau s}$, if we choose τ a multiple of the sampling rate T - the authors used $\tau = 5ms$ - the delay represented in digital filter forms as a simple delay operator, described in the z -transform domain as z^{-1} . This is 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:

$$\exp(-\tau s) \approx \frac{1 - \frac{\tau s}{2}}{1 + \frac{\tau s}{2}} \quad (8)$$

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

$$\exp(-Ts) \approx \frac{1 - \frac{Ts}{2}}{1 + \frac{Ts}{2}} = \frac{1 - \frac{1-z^{-1}}{1+z^{-1}}}{1 + \frac{1-z^{-1}}{1+z^{-1}}} = z^{-1} \quad (9)$$

Longer delays are similarly represented as z^{-m} . Using the bilinear transform, the above system $\mathbf{F}(s)$ becomes in the z -domain:

$$\mathbf{F}(z) = \frac{(1 + z^{-1})^2}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \quad (10)$$

The coefficients are obtained by algebraic manipulations:

$$\mathbf{a}_0 = \frac{4}{T^2} + \frac{4\omega_0\zeta}{T} + \omega_0^2 \quad (11)$$

$$\mathbf{a}_1 = 2\omega_0^2 - \frac{8}{T^2} \quad (12)$$

$$\mathbf{a}_2 = \frac{4}{T^2} - \frac{4\omega_0\zeta}{T} + \omega_0^2 \quad (13)$$

Pulling the coefficient \mathbf{a}_0 out, normalizing $\mathbf{c}_1 = \mathbf{a}_1/\mathbf{a}_0$ and $\mathbf{c}_2 = \mathbf{a}_2/\mathbf{a}_0$, and defining $\gamma = 1/\mathbf{a}_0$, one gets instead the representation:

$$\mathbf{F}(z) = \frac{\gamma(1+z^{-1})^2}{P(z)} = \frac{\frac{1}{\mathbf{a}_0}(1+z^{-1})^2}{1 + \mathbf{c}_1 z^{-1} + \mathbf{c}_2 z^{-2}} \quad (14)$$

Using a delay of $\mathbf{m} = 1$ sample times, as the authors did, results in $\mathbf{D}(z) = z^{-1}$, hence the z-transform of the entire system becomes:

$$\mathbf{H}(z) = \frac{\frac{\gamma(1+z^{-1})^2}{P(z)}}{1 + z^{-1} \frac{\gamma(1+z^{-1})^2}{P(z)}} \quad (15)$$

It's realizable digital filter form is:

$$\mathbf{H}(z) = \frac{\gamma(1+z^{-1})^2}{P(z) + \gamma z^{-1}(1+z^{-1})^2} \quad (16)$$

The input output relation in the z-domain is therefore:

$$(P(z) + \gamma z^{-1}(1+z^{-1})^2)Y(z) = \gamma(1+z^{-1})^2 U(z) \quad (17)$$

With becomes:

$$(1 + \mathbf{c}_1 z^{-1} + \mathbf{c}_2 z^{-2} + \gamma(z^{-1} + 2z^{-2} + z^{-3}))Y(z) = \gamma(1 + 2z^{-1} + z^{-2})U(z) \quad (18)$$

In the sample time domain, where \mathbf{n} is an index for time steps of length \mathbf{T} , this becomes a recurrence calculation by bringing all delayed terms to the right:

$$\mathbf{y}_n = \gamma(\mathbf{u}_n + 2\mathbf{u}_{n-1} + \mathbf{u}_{n-2}) - \mathbf{c}_1 \mathbf{y}_{n-1} - \mathbf{c}_2 \mathbf{y}_{n-2} - \gamma(\mathbf{y}_{n-1} + 2\mathbf{y}_{n-2} + \mathbf{y}_{n-3}) \quad (19)$$

Or equivalent:

$$\mathbf{y}_n = \gamma((\mathbf{u}_n - \mathbf{y}_{n-1}) + 2(\mathbf{u}_{n-1} - \mathbf{y}_{n-2}) + (\mathbf{u}_{n-2} - \mathbf{y}_{n-3})) - \mathbf{c}_1 \mathbf{y}_{n-1} - \mathbf{c}_2 \mathbf{y}_{n-2} \quad (20)$$

The initial states \mathbf{y}_i of the filter have to be set to the initial F0 value.

Fig. 2 shows an example for the tone sequence high, raising, low, falling, using the male data

0.3 Applications in Dectalk

For Dectalk the procedure was implemented as a “machine” that is called for each frame (6.4 ms) to compute a new F0 value. Typically, control intervals are specified for each Mandarin syllable.

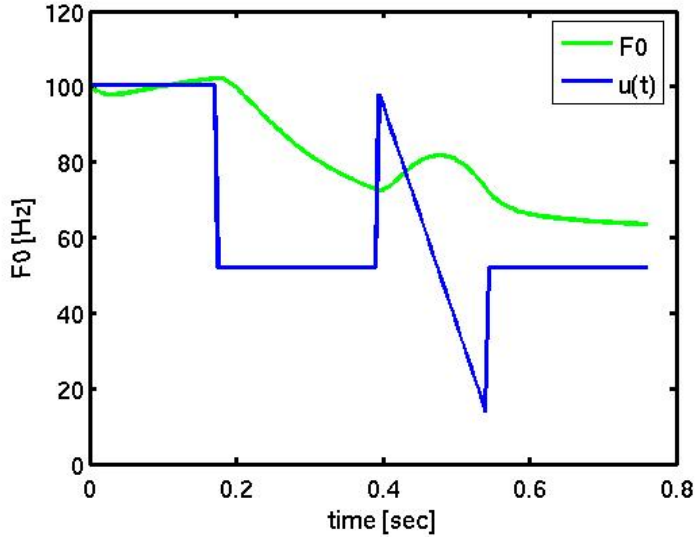


Figure 2: Sequence 'HLRF' using the digital filter implementation of the QTA model. Generated with Matlab

During the control intervals, a linear ramp is specified by a slope and offset parameter, so that the control function is calculated as $u(t) = at + b$ during the interval, where the time parameter t is reset to zero at the beginning of each control interval. The slope and offset are calculated in such a way that one assumes that at the end of the syllable a target is reached that corresponds to the value $u(\delta) = a\delta + b + F0_{ref}$, where δ corresponds to the duration of the syllable.

Furthermore, a reference $F0_{ref}$ is specified either as a constant throughout or as a slowly varying function of time. Modeling the slowly raising and then slowly falling phrase F0 function is also accomplished by a simple second order filter (which is essentially a leaky integrator). Fig 5 shows an comparison of the control input and the resulting F0 for the case of constant reference F0 and varying F0 reference.

0.4 Durations

It is clear that overall syllable duration strongly effects the behavior of the algorithm. Currently the variable ζ is held constant for each control interval and ω_0 is somewhat changed, depending on context. It is to be expected that a smaller ω_0 should be used for neutral tone, making the behavior more sluggish during the fifth tone.

The syllable durations are computed by a simple spring model. Initial durations of each phoneme are taken from a table, and the initial total syllable duration is calculated as the sum of these durations. Then the syllable type is determined using 12 types of syllables. This is an adaptation of a model idea by C. Shih and B. Ao². Different reference durations are associated with various degrees of complexity. For example the longest syllable with four phonemes and containing one vowel would be about 1.5 times longer than a single syllable made only of that one vowel. The variation in length is called a syllable scaling factor (syllscale). Once the total duration and the reference duration is calculated, a modification of the length of each phoneme is computed.

²Chilin Shih, Benjamin Ao: Durations Study for the Bell Laboratories Mandarin Text-to-Speech System., in: Jan P.H. van Santen, et.al., editors, Progress in Speech Synthesis, Springer 1997

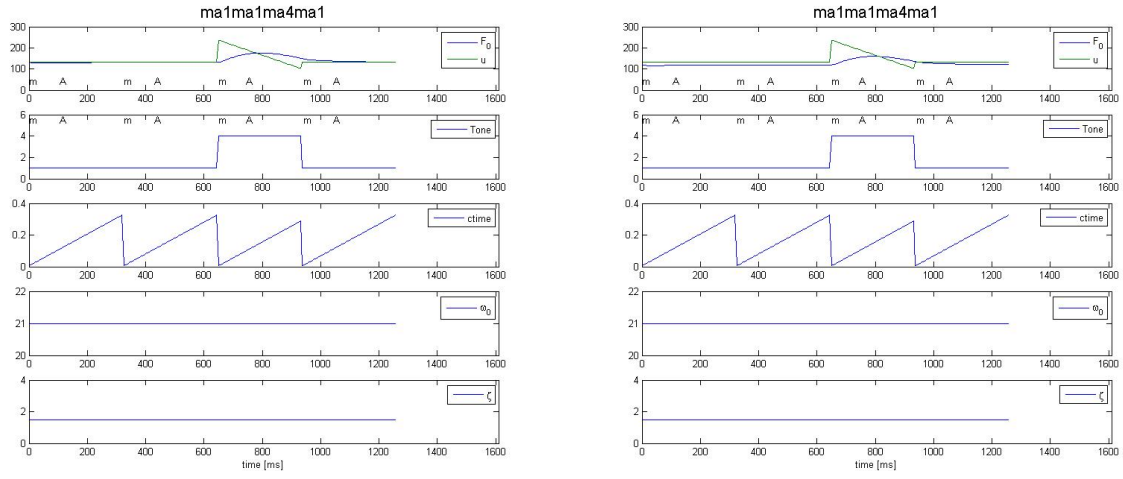


Figure 3: Utterance ma1ma1ma4ma1 with normal ω_0 and on the right with increased ω_0 .

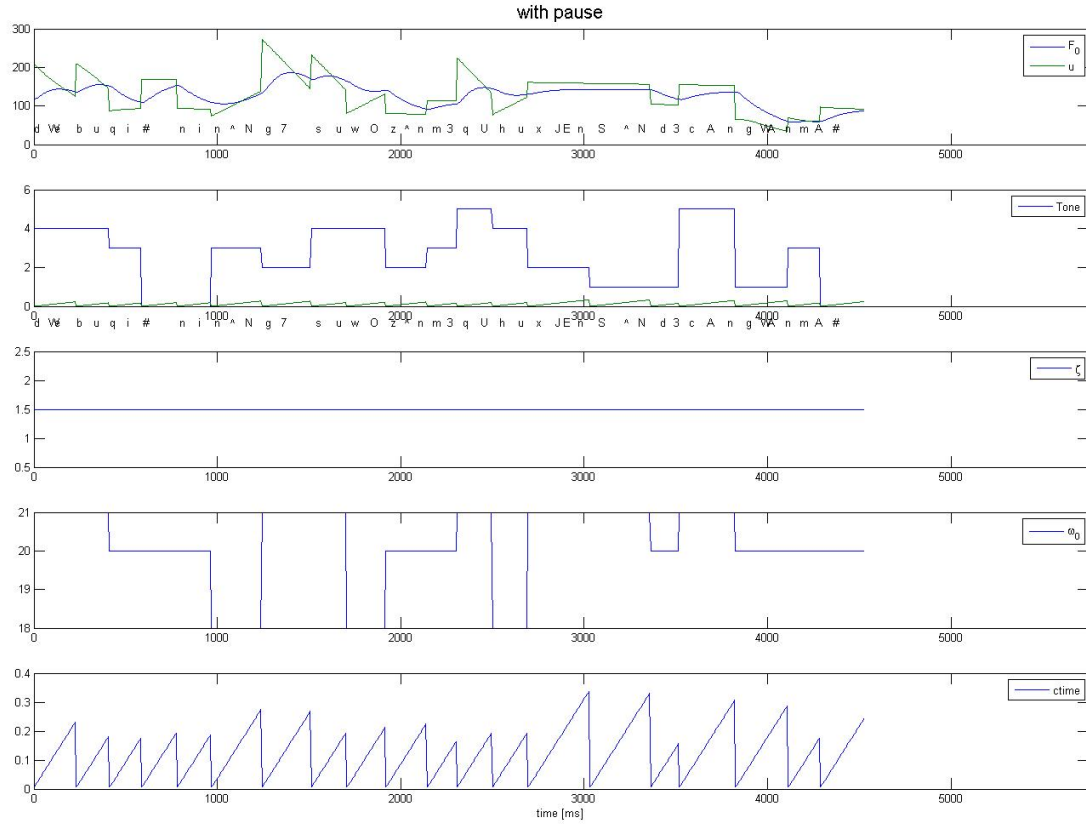


Figure 4: Utterance Excuse me where is Hu's restaurant?

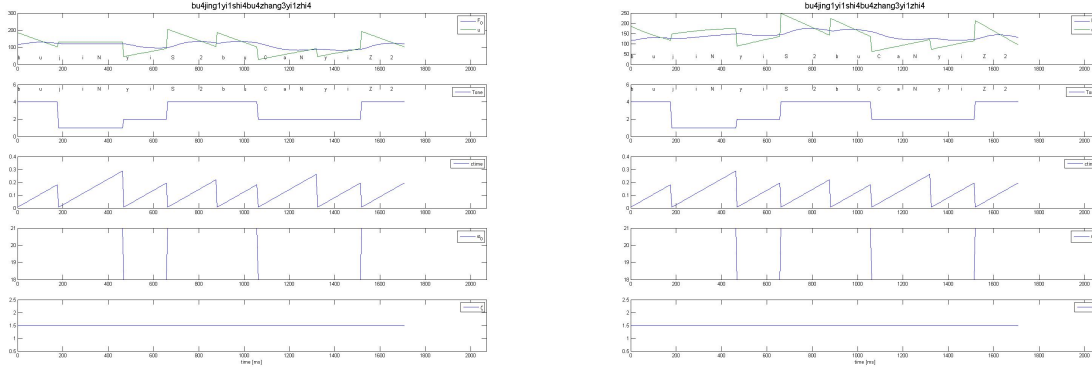


Figure 5: Comparison of control input and F0 contour without phrase contour (left) and with phrase contour (right)

If each sound would have to be scaled the same way, the syllable scaling factor would have to be divided by the number of phonemes and then each phoneme would have instead a length of

$$\text{syllscale} \times \text{total duration} / \text{number of phonemes}$$

However, since some phonemes are more flexible than others, for each phoneme a compliance number is tabled. Vowels have high compliance and plosives have the lowest compliance. The ratio of the phoneme's compliance and the sum of compliances is used as weight to calculate the change of length of the sound.

The durational changes are also made dependent on the tone. Further anticipated features will be to include duration changes based on position in the phrase, for instance, as typically the last syllable of a phrase is spoken with longer duration.

Once the duration of the syllable and the durations of each phoneme are calculated, the control intervals for F0 control for each syllable are defined, and the phonemes are sent to the synthesiser together with duration information.

0.5 Notes on possible modifications of the model (incomplete)

The main point here is to figure out what happens with the model if the duration of the delay τ is changed. In the original model, $\tau = 5\text{ms}$ and I chose to use the same sampling rate. Let's now consider the general case and see how the digital filter implementation changes.

For the case that $\tau = T$, it was already shown that the representation is by means of the digital delay operator z^{-1} . If the delay is an integer multiple of the sampling rate, *i.e.* $\tau = mT$, we have m delays in the digital domain, which are represented by the delay operator z^{-m} .

The intermediate case in which $\tau = mT + \kappa$, results in a somewhat more complicated digital filter. It can be represented by a delay operator z^{-m} followed by a digital filter that represents the delay by κ . The (approximated) delay in the Laplace domain is:

$$D(s, \kappa) = \frac{1 - \frac{\kappa s}{2}}{1 + \frac{\kappa s}{2}} \quad (21)$$

Using the bilinear z-transform (Eqn. 7), this becomes:

$$D(z, \delta) = \frac{\delta + z^{-1}}{1 + \delta z^{-1}} \quad \text{with} \quad \delta = \frac{1 - \kappa/T}{1 + \kappa/T}. \quad (22)$$

For this general case we can now replace the simple delay z^{-1} by the delay $z^{-m}D(z, \delta)$ in Eqn. 16, and obtain:

$$H(z) = \frac{\gamma(1 + z^{-1})^2}{P(z) + \gamma z^{-m} \frac{\delta + z^{-1}}{1 + \delta z^{-1}} (1 + z^{-1})^2} \quad (23)$$

To again obtain a realizable form, this has to be modified first into

$$H(z) = \frac{\gamma(1 + \delta z^{-1})(1 + z^{-1})^2}{P(z)(1 + \delta z^{-1}) + \gamma(\delta + z^{-1})(1 + z^{-1})^2} \quad (24)$$

Explicitely, as input-output relation one obtains

$$\left[P(z)(1 + \delta z^{-1}) + \gamma z^{-m}(\delta + z^{-1})(1 + z^{-1})^2 \right] Y(s) = \gamma(1 + \delta z^{-1})(1 + z^{-1})^2 U(s) \quad (25)$$

For now we will assume that m is at least equal to 1, that is, the delay is larger than then the sampling period. If that is the case, a recursion for y can be found. This time let's write it first in the Laplace domain, all input variables and delayed variables on the right side:

$$Y(s) = \gamma \left[(\delta + z^{-1})(1 + z^{-1})^2 \right] \left(U(z) - z^{-m}Y(z) \right) - \delta z^{-1}P(z)Y(z) - c_1 z^{-1} - c_2 z^{-2}Y(z) \quad (26)$$

Let's split this a little to avoid too much confusion. The resulting recursion will be written as $y_n = v_n - w_n$, and the firs part comes from the first part in the above, namely

$$V(z) = \gamma \left[(\delta + z^{-1})(1 + z^{-1})^2 \right] \left(U(z) - z^{-m}Y(z) \right)$$

results in the contribution:

$$v_n = \gamma \delta (u_n - y_{n-m}) + \gamma (2\delta + 1)(u_{n-1} - y_{n-m-1}) + \gamma (2 + \delta)(u_{n-2} - y_{n-m-2}) + \gamma (u_{n-3} - y_{n-m-3})$$

and the other part is from the subtraced:

$$W(z) = (\delta z^{-1}P(z) + c_1 z^{-1} + c_2 z^{-2})Y(z)$$

written as sequence:

$$w_n = (\delta + c_1)y_{n-1} + (\delta c_1 + c_2)y_{n-2} + \delta c_2 y_{n-3}$$