Stem-ML: Language-Independent Prosody Description

Greg P. Kochanski and Chilin Shih
Lucent Technologies, Bell Laboratories

Abstract

Stem-ML is a tagging system with a completely defined algorithm for translating the tags into quantitative prosody in any language. It separates the description of prosodic intentions from their execution, by modeling the interactions between accents. We designed Stem-ML to allow automated training of accent shapes and parameters from acoustic databases.

Stem-ML is linguistically neutral: it allows a description of any physiologically realizable prosody in terms of linguistic concepts, without imposing a restrictive theory on the data. The tag set and algorithm make no assumptions about the number of distinct types of accents or tones, or their scope. Accents and tones are treated interchangeably. Stem-ML allows, but does not require, descriptions involving phrase curves.

The model begins with soft templates for tone or accent shapes that are specified by the user or obtained by automated training. These soft templates interact because of physically and physiologically motivated constraints that model the smooth and continuous motions of the muscles that control prosody.

Introduction

Stem-ML bridges the gap from linguistic theories to the physical reality of a glottal oscillator. The tags are robust, and general enough so that they can be used to compare different theories. This paper focuses on accents, and largely ignores phrase curves. A companion paper in these proceedings[1] shows examples of Stem-ML application to actual speech data. A more formal and complete description of the Stem-ML tag set can be found at http://www.bell–labs.com/project/tts/stem.html.

Stem-ML marks accents compatibly with standard linguistic assumptions: accents are local, with a scope of a stress group[2], a word or a syllable. Far from the center of a word, they have little effect, except perhaps for a shift in pitch. The phrase curve, on the other hand, has no assumption of locality, and is appropriate for pitch changes on scopes larger than a word.

Stem-ML assumes that humans are capable of pre-planning of pitch contours inside a phrase, so the pitch will be affected by future tags up to the end of the phrase. Pre-planning of other aspects of speech has been shown, such as inspired lung volume[3,4,5].

The modeling in Stem-ML was inspired by tone languages such as Mandarin, but also applies well to languages like English where accents have a word scope. Isolated syllables in tone languages have pitch contours close to the ideal shapes of their tones, while in sentences, tones interact due to their close proximity to each other. As a result, in natural speech or in complex sentences, tone shapes can be far from ideal. Syllables in weak positions can even display inverted tone shapes. Stem-ML explains the changes in tone shapes in terms of interactions with nearby syllables.

Stem-ML assumes that the prosodic trajectory is continuous and smooth over short time scales. We know that all aspects of prosody are controlled by muscle actions[6]. Muscles cannot respond fast enough to discontinuously change prosody between phonemes[7]. Reflecting the constraint that pitch changes are gradual, the model compromises between nearby templates to guarantee smooth connections.

Tone Interaction Modeling

Communication is a two-ended process, a mixture of generation and perception. We assume that the speaker balances the physiological energy cost of adjusting muscle positions against the need to produce unambiguous speech by matching the tone/accent templates. At prosodically strong positions in a sentence, the speaker is generally willing to expend the effort to produce precise prosody. Since energy costs increase with muscle velocities and accelerations, slow, smooth, and small motions are less costly. Thus, between strong positions, the speaker tends to minimize effort by smoothly preparing for the next strong tone/accent, and by ignoring the ideal shape of the syllable in a weak position. Intermediate strengths yield intermediate results.

We represent this process as an optimization problem where we maximize the sum of two functions, one, *G*, representing ease of production, and the other, *R*, representing the speaker's estimate of the extent to which the prosody will have the desired effect on the listener. We approximate the ease of production by $G = -\sum_t \dot{p}_t^2 + (\pi\tau/2)^2 \ddot{p}_t^2$, where τ is the smoothing time, *t* is time, and the raised dots indicate time derivatives. *G* is largest for a flat pitch contour, and becomes negative as the pitch becomes more variable. It can be interpreted as a simple approximation to the energy expenditure in the muscles controlling prosody.

We write the simplest possible form for R, a weighted error measure between prosody targets and the realized prosody. $R = -\sum_{i \in tags} s_i^2 r_i$, where s_i is the strength of tag *i*, where $r_i = \alpha \sum_{t \in tag\ i} (p_t - y_t)^2 + \beta(\overline{p} - \overline{y})^2$ is the mismatch associated with the i^{th} tag. Alpha and beta are constants that depend on the *type* of the tag[8]. R is zero when the pitch curve exactly matches the shape of the tag, and becomes negative (indicating a greater expected likelihood of listener misinterpretation) if the pitch doesn't match the tag.

The Stem-ML prosody solution is then the pitch curve that minimizes *G*+*R*. In short, the algorithm accumulates a set of constraints on the prosody, then calculates the function that best meets the constraints. The constraints come in bunches – tags – that are associated with accented syllables. One can also look at the system as implementing elastic or soft templates that compromise with their neighbors.

Tags

Stem-ML is controlled by the following parameters (set once per phrase) with the **set** tag:

- *smooth*=`float`: sets the smoothing time of the pitch curve, in seconds. This is used to set the width of a pitch step (see the *step* tag).
- *base*=`float`: set's the speaker's baseline.
- *range*=`float`: set's the speaker's pitch range.

The **stress** tag defines the ideal tone shape, locally. Each stress tag has a preferred shape (and a preferred height relative to the phrase curve), but they will bend to compromise with each other. Stress tags will also compromise to meet the requirement that the pitch curve must be smooth.

The stress tag allows you to accent words or syllables in a very general manner. You always specify three things: the ideal 'platonic' shape of the tone/accent, which is the shape it would have without neighbors. Second, you specify the strength of the accent. Strong accents tend to keep their shape; weak accents tend to be dominated by their neighbors. Finally, you give the *type* of the accent.

Arguments:

- *shape*=(`point` ",")* `point`: this specifies the ideal shape of the accent curve as a set of (time, frequency) points.
- *strength*=`float`. Corresponds to the linguistic strength of the accent. Accents with zero strength have no effect on pitch. Accents with $strength >> 1$ will be followed accurately, unless they have strong neighbors.
- *type*=`float`. Controls whether that accent is defined by its mean value relative to the pitch curve, or by its *shape*. If it is important only that the accent should be above or below the pitch curve, but the detailed shape is not important, you should set *type*=1. Alternatively, if the shape is critical (*e.g.*, the accent is a falling tone), but it doesn't matter whether the accent ends up above or below the pitch curve, then you should set *type*=0. Intermediate values let you control both the mean pitch and shape.

The following figure is an example:

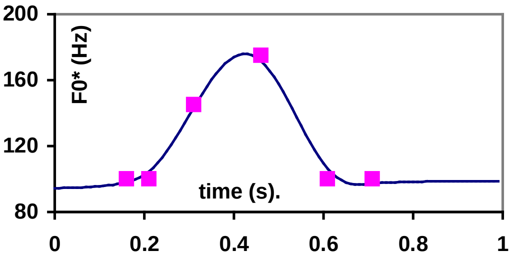

Figure 1 Points that define the shape of a **stress** tag and the resulting pitch trajectory.

The following table shows schematically how accents interact with their neighbors.

Accent interactions vs. strength and *type*.	***Type* ≈ 0**	***Type* ≈ 1**
Strength >> neighbor's & Strength >> 1	The accent keeps its shape precisely. Neighbors will bend to accommodate it.	The accent's average pitch is precisely controlled. Neighbors must accommodate.
Strength ≈ neighbor's	The shape will be a compromise with the neighboring accents. Average pitch will be controlled by the neighbors.	The average pitch will be a compromise with the neighboring accents. The shape will be controlled by the neighbors.
Strength << neighbor's	The accent is relatively weak. The prosody will be dominated by the neighboring accents.	
Strength >> 1	The speaker is willing to expend substantial effort to make the sound match the template.	
Strength ≈ 1	The pitch curve will be a smoothed version of the accent.	
Strength << 1	This accent is unimportant. The speaker is expending minimal effort, and the pitch curve will be smooth and continuous.	

At the extremes, the accent *type* parameter separates accents into those where the shape, (or changes in pitch) are critical, or those where the average pitch is critical. If *type*=0, the shape is critical. One example might be "the pitch drops by 50Hz". At the other extreme, *type*=1, the shape doesn't matter, but the average pitch is important. An example might be "the pitch is

50Hz above the phrase curve.” Intermediate types are possible, and give you accents that define both a shape and a mean pitch.

Compromises between Tags - 1

While it is normal to write a phrase curve without conflicting requirements that would cause the system to compromise, compromises abound in tone shapes. It is easy to find situations where the speaker wants to end one tone low, yet start the next one at a high pitch. Somehow, the shapes need to be modified, or the pitch has to be increased between the two tones. Stem-ML can do either.

In the following five figures, we explore the interaction between two nearby tones. The first is a level tone with a well-defined pitch. The second tag is a falling tone. What we’ll see in each figure is how the pitch behaves as we adjust the target pitch of the first tone. The first figure shows a pure falling tone: it has no preferred pitch, but has a strongly preferred shape (*type* = 0). Each following figure will have successively stronger pitch preferences and weaker shape preferences, until in the last figure, the shape is totally unimportant (*type*=1). The centers of both tones are marked with dashed lines.

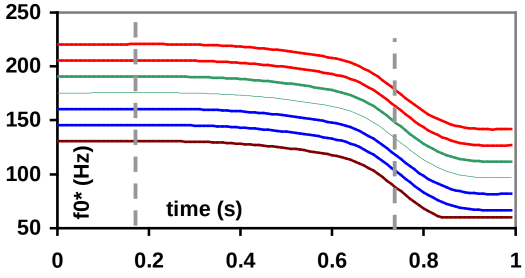

Figure 1: A pure falling tone (*type*=0) following a level tone (*type*=0.8). We vary the target pitch of the level tone. The resulting pitch curves are parallel, because the second tone has its shape constrained, not it’s average pitch.

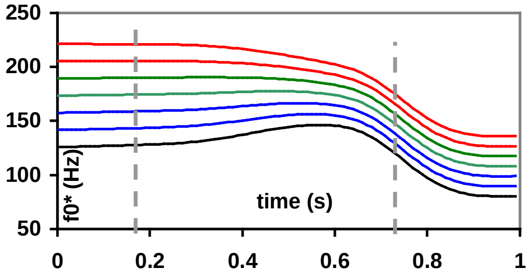

Figure 2: A falling tone with a weak pitch preference (*type*=0.1) following a level tone. The pitch curves start to bunch up on the falling tone, as its pitch preference begins to be felt.

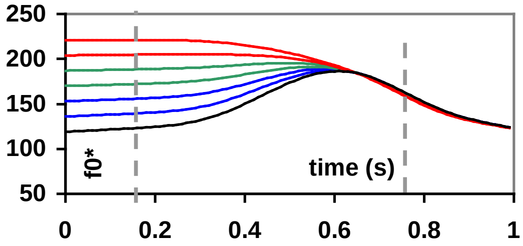

Figure 3: The second tag now has a strong pitch preference (*type*=0.6). It defines both its shape and pitch quite rigidly. Note that when the preceding level tone is low, the pitch now must increase in preparation for the second tone.

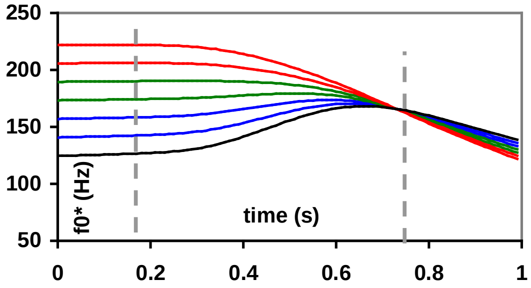

Figure 4: With *type*=0.8, the second tone is primarily defined by its average pitch. The shape is now relatively unimportant, but the tone still manages to enforce a declining pitch near its midpoint. When the first tone has a low pitch, the pitch curve now needs to rise strongly in between the two tones, so that the pitch will be correct at the center of the second tone.

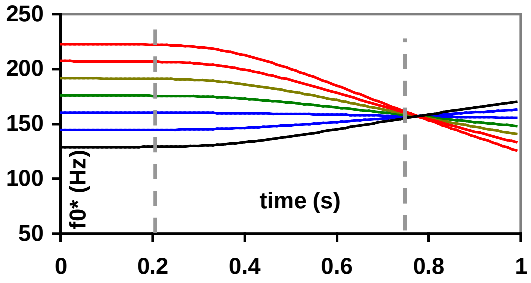

Figure 5: In this last figure in the sequence, the second tone is defined completely by its pitch (*type*=1). In this extreme, the *shape* of that tag becomes irrelevant, and the only constraint that the average value of each pitch trajectory (over the region where the tone is defined) be correct.

Compromises between Tags - 2

If we bring nearby accents together, we can get another example of interactions between tags. Stem-ML is not an additive model: the result of putting two accents on top of each other is less than the sum of the two accents. It corresponds to a single accent of the same *shape* and *type*, but $2^{1/2}$ times the strength.

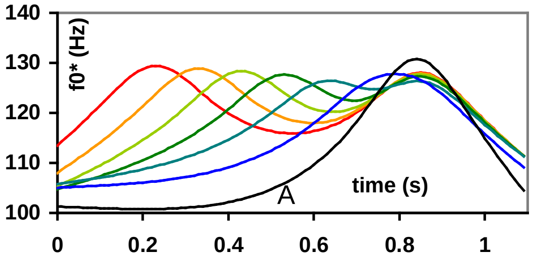

Figure 6: Interaction of two identical accents. One accent comes in from the right; the other is stationary at 0.83s. The curve labeled 'A' shows the accents on top of one another.

Accent Strength

In Stem-ML, all accents/tones have a strength parameter, which is intended to correlate with the linguistic strength of the word. In general, strong accents will keep their shapes, while weak accents will be dominated by their neighbors. The next example shows this effect by simulating three tones: a strong high tone, then a falling tone of varying strength, then a weak high tone. When the falling tone is very weak, it is completely dominated by its neighbors, and is almost invisible. On the other hand, when it is strong, it retains its shape, pushing down the weaker high tone.

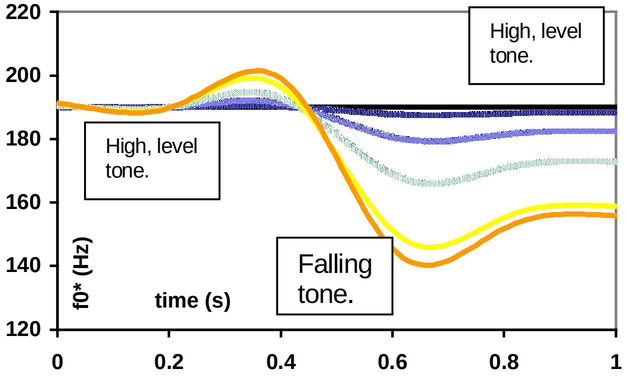

Figure 7 shows interactions between three tones as the strength of the middle one (a falling tone) is varied. The falling tone is unimportant when it has zero strength (topmost curve), and gradually approaches its ideal shape as its strength is increased (successively descending curves). The weak, neighboring level tone (right) is increasingly perturbed and pushed down as the falling tone becomes stronger.

Implementation

The algorithm operates in four stages. First, it expands all macros. Second, it uses **slope** and **step** tags to build a phrase curve. Third, it uses **stress** tags to build a prosody curve based on the phrase curve. Fourth, it maps the prosody onto observable acoustic characteristics. The phrase curve and accents/tones that ride on top of it are calculated similarly.

The maximization is a linear operation, and can be implemented with standard matrix packages like LAPACK. On a 200MHz workstation, our current implementation calculates 10s of prosody per CPU second. The speed is independent of phrase length, as the matrices are block diagonal with a constant bandwidth.

The algorithm enforces continuity at minor phrase boundaries, but phrase boundaries explicitly break pre-planning. It does not seem desirable to allow tags at the beginning of phrase 2 to effect the pitch near the end of phrase 1. We were unable to find examples of such behavior in real speech data. People seem to end a phrase, without considering what the pitch will be at the beginning of the next phrase, then make any necessary pitch shifts during the pause between phrases or at the beginning of the following phrase.

[1] Shih, Chilin, and Kochanski, G. P., *Chinese tone modeling with Stem-ML*, ICSLP 2000 (Beijing, China, 2000).

[2] Grønnum, Nina 1992, *The Groundworks of Danish Intonation – An Introduction.* Museum Tusculanum Press, Copenhagen.

[3] Winkworth, A. L., Davis, P. J., Adams, R. D., Ellis, E. 1995, "Breathing patterns during spontaneous speech," *J. Speech and Hearing Research* **38(1),** 124-144.

[4] Winkworth, A. L., Davis, P. J., Ellis, E., Adams, R. D. 1994, "Variability and consistency in speech breathing during reading – lung-volumes, speech intensity, and linguistic factors," *J. Speech and Hearing Research*, **37(3)**, 535-556.

[5] McFarland D. H., Smith, A. 1992, "Effects of vocal task and respiratory phase on prephonatory chest-wall movements," *J. Speech and Hearing Research*, **35(5)**, 971-982.

[6] Ohala, J. and P. Ladefoged (1970). "Further investigation of pitch regulation in speech," *UCLA Working Papers on Phonetics*, 14, 12-24.

[7] Stevens, K. N., *Acoustic Phonetics*, MIT Press, 1998, ISBN 0-262-19404-X, pp. 40-48 and references therein.

[8] $\alpha = \cos^2(type \cdot \pi / 2) / N$, $\beta = \sin^2(type \cdot \pi / 2) - \alpha * N$, where *N* is the of points in the tag.