ELSEVIER

Speech Communication 39 (2003) 311–352

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Prosody modeling with soft templates

Greg Kochanski *, Chilin Shih

Bell Laboratories, Lucent Technologies, 600 Mountain Avenue, Murray Hill, NJ 07974, USA

Received 30 November 2000; received in revised form 4 February 2002; accepted 18 February 2002

Abstract

This paper describes a novel prosody generation model. We intend it to broadly support many linguistic theories and multiple languages, for the model imposes no restriction on accent categories and shapes. This capability is crucial to the next generation of text-to-speech systems that will need to synthesize intonation variations for different speech acts, emotions, and styles of speech. The system supports mark-up tags that are mathematically defined and generate f_0 deterministically. Underlying the tags is an articulatory model of accent interaction which balances physiological and communication constraints. We specify the model by way of an algorithm for calculating the pitch, and by way of examples. The model allows localized, linguistically reasonable tags, and is suitable for a data-driven fitting process.

Keywords: Intonation; Prosody; Tone; Modeling; Dynamics; Physiology; Algorithm; Computer language; Speech; Pitch; XML; Mark-up language; Communication; Text-to-speech

1. Introduction

The demands of interactive approaches to TTS require more freedom to express prosody than current systems allow. Most current TTS systems, including the Bell Labs TTS system, were designed to operate on text with little or no "mark-up" information beyond the text. The prosody subsystem was therefore designed conservatively, because of the intrinsic limitations of how reliably prosodic information could be deduced from the text. If some prosodic feature could not be reliably deduced, it was found better to produce a neutral prosody than the wrong one.

The next generation of TTS applications will not have this limitation, because many applications will be conducting a dialog, and will have state information corresponding to goals and intentions. The application may be "intending" to convey that a set of words is a single proper noun, that a word is especially important, or that a word needs confirmation. This state information needs to be expressed prosodically, so one should think of speech synthesis more in the context of a concept-to-speech system than a text-to-speech system. Similarly, there are applications where the simulation of emotions, subtle meanings in speech acts, and stylistic variations are desirable. This prosodic information can be supplied to the TTS

* Corresponding author.
E-mail address: gpk@bell-labs.com (G. Kochanski).

0167-6393/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0167-6393(02)00047-X

system by adding mark-up tags to the text. With marked text, the TTS system does not need to deduce as much, so it need not be designed conservatively.

The mark-up system is most useful if it is flexible enough to support any intonation event that a user or a future dialogue system might want to express. A pertinent question is then how to design a pitch generation system that will support linguistic models that are not yet defined.

In this paper, we introduce a prosody tagging and generation system **S**oft **TEM**plate **M**ark-up **L**anguage (Stem-ML). This system combines mark-up tags and pitch generation in one, therefore allowing future users and dialogue systems to control intonation events without the concern of writing a pitch generation component for the TTS system. We define a set of tags that serve the dual function of marking the text and pitch generation. The user can use these tags to describe linguistic events, and the tags automatically provide pitch generation support. It is thus most important to allow the model we define to represent any possible prosody. [1] A second goal is to mark it in a way that is compatible with standard linguistic assumptions: that accents are localized and associated with stress groups, words or syllables. A final goal is for this model to make use of information that is predictable from text, such as word accents, tones, and prosodic boundaries; this will allow us to minimize the number of tags that need to be added to text. Ultimately, we see this model becoming an "assembly language" where tags and their parameter settings would be produced by automated tools.

From a research point of view, it is important to have a model that bridges the gap from linguistic theories to the objective reality of a glottal oscillator with a time-varying frequency. The model needs to be general enough so that it can provide a quantitative representation of many different theories of intonation, and can therefore be used to compare theories.

1.1. Literature review

Most TTS systems divide the task of intonation generation into two components, a linguistic modeling component and a pitch generation component (Sproat, 1998). The linguistic modeling component is carried out as part of the text analysis, where the input text stream is processed and intonation events are deduced from the text and from high-level tags that contain non-deducible information about prosodic intent. The intonation events are then coded in abstract representations. Examples of the linguistic modeling component include ToBI (Silverman et al., 1992), Tilt (Taylor, 1998), INSINT (Hirst et al., 2000), among others. Lexical tone languages such as Chinese and Vietnamese conveniently provide some of this information from the lexicon.

The pitch generation component is the decoding process where f_0 contours are generated from the linguistic representations. Traditionally, the pitch generation component is designed to support a specific abstract representation and is implemented after the representation is known. For example, given ToBI labeling, one may write a rule set to describe the f_0 shapes and their pitch values (Anderson et al., 1984), or to use machine learning techniques to train the target values, including linear regression models (Black and Hunt, 1996), CART tree models (Dusterhoff et al., 1999) and dynamical system models (Ross and Ostendorf, 1999). These pitch generation models are the decoders of ToBI, and will not support concepts that are not represented in ToBI. It should be obvious that phenomena that are not coded in the linguistic modeling component cannot receive support from the pitch generation component.

In the remainder of this section, we review the literature in the area of intonation modeling, finding the common ground where multiple models might be interfaced to a common pitch generation component.

[1] We use the term "prosody" broadly, meaning a time series of speech information that's not predictable from a reasonable window (e.g., word sized or sentence sized) applied to the phoneme sequence. This could include pitch, amplitude, and gestures. The tag set also applies to tone shapes in tone languages, so we bring them under the umbrella term "prosody."

Table 1
Intonation Schools classified by the way they describe prosody

	Under-specified	→	→	→	Fully specified
Single component	INTSINT	ToBI Xu		Tilt, IPO	Olive, Machine learning
Two components	Grønnum			Fujisaki	
Multiple components					Van Santen

The primary goal of intonation research is to model natural f_0 contours of speech, preferably in relation to a transcription and a description of the prosodic intent of the speaker. The starting point of intonation research is the time series of f_0. But the interpretation of the f_0 information diverges widely among intonation schools. Table 1 represents a view of how one can classify the various intonation schools. The shape of an accent may be fully specified (i.e., defined without gaps) or under-specified (defined by disconnected regions or isolated points). Along another dimension, f_0 values at any given time may be treated as a single component or as the combination of multiple components.

INTSINT (Hirst et al., 2000) is an under-specified intonation system that defines an accent by a single point. Fitting quadratic spline curves through these points generates surface f_0.

The most widely used under-specified accent shape is represented by the ToBI school (Beckman and Ayers, 1997; Silverman et al., 1992), which developed from earlier works such as Pierrehumbert (1980), Liberman and Pierrehumbert (1984), and Pierrehumbert and Beckman (1988). Each accent is represented by no more than two points, which specify abstractly the relative contrast of high (H) and low (L). One goal of the ToBI system is to specify a minimal set of categorical labels for intonation. These labels are usually interpreted as phonological distinctions between accent types.

Xu et al. (1999) represents Chinese tones with under-specified static or dynamic targets. The surface f_0 contours are generated with a model that approaches these targets asymptotically within the domain of a syllable.

Tilt (Taylor, 2000; Taylor, 1998) allows more samples than ToBI near the peak of an accent and leaves the other regions unspecified, hence its status half way to a fully specified system. Tilt considers all accent types to be continuous variations of a single class. Surface variations are accounted for by changes in the continuous parameters. IPO (de Pijper, 1983) prepares a piecewise-linear approximation to the pitch contour. They then associate the slope and height of these lines with various types of accents.

Olive (1975) described a very early fully specified system, following work by Levitt and Rabiner (1970). His model stored the surface pitch vs. time contour as a function of the grammatical structure of the sentence. The contour was then approximated by polynomial splines attached to words, to allow for duration variations.

Several works using machine learning techniques generate densely sampled f_0 values, including Chen et al. (1992) and Malfrère et al. (1998). We classify these works as fully specified systems even though in some cases the concept of accent may not be clear. Ross and Ostendorf (1999) described an interesting machine learning system where a discrete learning system would predict vectors attached to phonemes and syllables, and these vectors would in turn drive a (learned) dynamical system to predict f_0.

The advantage of using an under-specified accent shape is that it allows sufficient distance between specified accent targets to allow a smooth f_0 transition, typically by way of interpolation. The drawback is that it ignores changes of shape between specified targets. On the other hand, a system with fully specified accents leaves little room to resolve conflicting targets. A simple concatenation of fully specified accents will result in a pitch curve with unnatural jumps at the concatenation joints. Many systems, such as Fujisaki (1983, 1988), use filters to smooth out abrupt changes in f_0. Alternatively, van Santen and Möbius (1997, 2000) require each accent to begin and end at zero to ensure smooth connections between accents.

Turning to the f_0 dimension of Table 1, many intonation schools treat surface intonation contours as the superposition of a phrase component and an accent component. Grønnum (1992) and Fujisaki (1983, 1988) are representatives of this view.

A well-defined model that fully specifies accent shape and uses multiple components is van Santen's (van Santen and Möbius, 1997, 2000; van Santen et al., 1998), where accents are represented by densely populated points, providing a mechanism to describe highly complex accent shapes in detail. We characterize van Santen's system as having multiple components, because in addition to the phrase component, each accent in the phrase also adds a phrase-length component that contributes to the surface f_0 contour.

The advantage of multiple components is that it provides a mechanism to separate individual accents from long-term effects. However, if one allows multiple components, then one necessarily faces the problem that there is no unique solution in the decomposition of a single f_0 time series into multiple components. Any such decomposition depends on a model of the speech process, and is only as good as the underlying model. In contrast, Liberman and Pierrehumbert (1984) explicitly reject the notion of a phrase curve and represent intonation contours as a single component. The advantage of representing f_0 information as a single component is that the representation of accent heights will then be transparent, which lends itself to convenient automatic labeling.

Stem-ML provides a well-defined mapping from tags to f_0 contours, replacing the pitch generation algorithm of TTS. Accent shapes are templates, represented by the **stress** tag (Sections 3.4 and 4.4), which can be over-specified (tags overlap in time), fully specified or under-specified. We allow a complex phrase curve to be described by the **step** and **slope** tags (Sections 3.2, 3.3, 4.1 and 4.2), but f_0 can also be represented without one. Each tag places constraints on the pitch calculation, and the resulting pitch contour is a compromise between two groups of constraints: physiological constraints that require the pitch trajectory to be smooth, and communication constraints that bring the surface pitch contour close to the tag specification (see mathematical description in Section 2). The templates bend to meet requirements from neighboring accents or the phrase curve, therefore we call them "soft" templates. Conflicts between accent target specification are resolved in a way that depends on strengths (Sections 3.4 and 4.5). Strong tags dominate the resulting pitch contour, while weak tags accommodate to strong neighbors.

Typically, there are many ways to represent a given prosody with Stem-ML, and one can write a Stem-ML description that is similar to many models in the existing literature. While one may need a non-trivial algorithm to translate from other tagging systems into Stem-ML tags, Stem-ML can provide a representation close enough for translation to be possible. For example, it can approximate van Santen's model with overlapping long **stress** tags, one tag per accent, along with a simple phrase curve. ToBI can be approximated with **stress** tags, each with two points in their *shape*, and no phrase curve.

An alternative classification of intonation systems is Ladd's (1996) distinction between overlay and linear sequence models. Again, we can build models in both classes. Overlay models build f_0 curves by superposing f_0 features of different sizes, for instance sentence, phrase, word, and syllable scopes. Stem-ML models of that class can be built using phrase curves and/or superposing **stress** tags of different scopes. On the other hand, linear sequence models are naturally described as a sequence of **stress** tags, one per tone or accent.

1.2. Concepts

The physical modeling in Stem-ML was inspired by tone languages such as Mandarin. 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, tone shapes can be far from ideal. Syllables in weak positions can even display inverted tone shapes as speakers prepare for the next strong syllable (Shih and Sproat, 1992; Xu, 1993). Stem-ML explains the changes in tone shapes in terms of interactions with nearby syllables (Kochanski and Shih, 2000; Shih and Kochanski, 2000). This indicates that prosody is pre-planned, and we suggest that the planning is done to minimize physiological effort given the communicative demands of speech.

Stem-ML assumes that humans are capable of pre-planning of pitch contours inside a phrase. [2] The final pitch curve depends on tags in both the forward and reverse directions inside a phrase. This provides a natural way of expressing interactions between neighboring accents and tones. Pre-planning of other aspects of speech has been shown, such as inspired lung volume (Winkworth et al., 1994, 1995; McFarland and Smith, 1992; Whalen and Kinsella-Shaw, 1997) and pitch as a function of sentence length (Shih, 2000). Experiment does not yet afford good evidence for the limitations or the maximum range of pre-planning. Indeed, the range may well be strongly variable. Practiced, prepared speech may have no clear limits to planning, while speech under heavy cognitive load may barely be planned to the end of a word. Stem-ML **phrase** tags (Sections 3.5 and 4.8) are the mechanism for specifying the limit of pre-planning. The Stem-ML model is causal between phrases, since the pitch at a given time depends only on the tags in the current and past phrases. However, the model is acausal inside a phrase since we assume a phrase is planned as a unit, so the pitch can be influenced by any linguistic event in the phrase.

Commonly, people seem to end a phrase without considering what the pitch should 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. In fact, this behavior is the definition of our phrases: planning stops at phrase boundaries. Thus, one places phrase boundaries at locations where the past pitch is independent of future linguistic features. In our experience, sentence boundaries and long pauses seem to imply Stem-ML phrase boundaries, but proper choice of phrase boundaries may well depend on the language being spoken.

Stem-ML makes one physically motivated assumption. It 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, and that the mapping between muscle activation and perceived prosody is not strongly non-linear. Thus there are smooth and predictable connections between neighboring accents, because muscles simply cannot discontinuously change position. The muscles that control the larynx cannot respond faster than 100 ms (Stevens, 1998, pp. 40–48 and references therein; Xu and Sun, 2000), a time that is only slightly shorter than a typical syllable, so we expect the intonation of neighboring syllables to interact. This interaction should be important in all languages. Our goal is natural-sounding speech, and a careful introduction of physiological constraints on the models can help text-to-speech systems sound more like a real human.

Öhman (1967) and Fujisaki (1983) were instrumental in incorporating physiological constraints in pitch generation. Xu et al. (1999) is a more recent work providing a quantitative model for Chinese tones. Some related work in articulatory modeling includes Browman and Goldstein (1990), Keating (1990), Moon and Lindblom (1994), and is reviewed in Perrier et al. (1996) and commentaries in Abry et al. (1998). The C/D model (Fujimura, 2000) is also noteworthy.

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 needed to produce precise prosody. Since energy costs increase with muscle velocities and accelerations, slow and smooth motions are less costly. Thus, on weak positions, the speaker tends to minimize effort by smoothly preparing for the next strong tone/accent, and largely ignoring the ideal shape of the weak syllable. Intermediate strengths yield intermediate results. This aspect of the model also builds upon Ohala (1992) who described speech as a compromise between effort and communication clarity, but used the concept only qualitatively.

[2] In this paper, a "phrase" is defined to be the interval between two Stem-ML **phrase** tags. Normally, a Stem-ML phrase would be associated with an utterance, intonational phrase, or breath group, but the precise association could vary from language to language or from theory to theory.

This same model can apply to other gestures related to language, so long as there is a direct relationship between muscle positions and the perceived gesture, and the relationship is not excessively non-linear. While pitch is generally believed to be the most important component of prosody, it has been known since the 1950s (Fry, 1955, 1958; Bolinger, 1958; Lieberman, 1960; Hadding-Koch, 1961) that amplitude is also an important component. Recent literature (Maekawa, 1998; Kehoe et al., 1995; Sluijter and van Heuven, 1996; Pollock et al., 1990; Sluijter et al., 1997; Turk and Sawusch, 1996; Erickson, 1998 and references therein) also provide support for amplitude, spectral tilt and jaw movement as important components of prosody. We believe that this model can apply to at least some of these motions.

A single Stem-ML tag can produce a correlated ensemble of changes in a variety of acoustic parameters. For instance, an accent could include both a rise in pitch and a bump in amplitude. Furthermore, the tag set can apply to facial features. The assumptions of direct relationship and no strong non-linearity are clearly true for facial expressions, as the muscle motions are directly visible.

In the case of the fundamental frequency of speech, one can define a signal we refer to as f_0^*, which should show smooth and continuous behavior. In voiced segments, f_0^* is the observed pitch with segmental effects removed, where we consider segmental effects to include all correlations of f_0 with the phoneme sequence. An example of using f_0^* to model intonation can be found in Black and Hunt (1996), where they use a smoothing technique to reduce the amplitude of segmental effects associated with consonants. For their algorithm, they report a 9.9 Hz RMS difference between f_0 and f_0^*, which can be taken as a rough estimate of the size of segmental effects.

Without segmental effects, the factors that influence the pitch are the vocal fold tension (Ohala and Ladefoged, 1970) and subglottal pressure (Monsen et al., 1978). The vocal fold tension and subglottal pressure are both smoothly changing functions of time, controlled by nerve impulses, Newtonian mechanics, and the viscoelasticity of tissue. The overall relationship between muscle activation and pitch is smooth, nearly linear, and the effects of the different muscles can probably be combined into a single parameter. For instance, even though low tones may be generated by activation of the sternohyoid muscle (Gårding et al., 1970), and high tones by activation of the cricothyroid (Atkinson, 1978; Simada and Hirose, 1978), as long as the dynamic response of the two sets of muscles are similar, the difference in the two responses should map nicely to f_0, because the difference corresponds to the extension of the vocal folds.

Detailed physiological models for f_0 are described in (Titze, 1993a) and references therein. Also see the discussion of the "Cover model" in (Titze, 1993b) for an example of how activity of the thyroarytenoid and cricothyroid muscles combine. Similar calculations involving the lung pressure also show a smooth dependence that is not strongly non-linear.

We are thus able to use a phenomenological model of the vocal fold oscillation, rather than a detailed model. Since the vocal fold tension seems to be the most important contribution, one can consider f_0^* to be an approximate measure of the vocal fold tension. We make quite weak assumptions about the behavior of the laryngeal oscillator: merely that f_0^* is a smooth function of a control parameter that has dynamics like a muscle. We do not need to associate the control parameter with any particular muscle. Since all the control parameters are smooth, we know that the frequency of the glottal oscillator must also be smooth except possibly at a few discontinuous jumps [3] (Herzel, 1995; Berry et al., 1996), such as register transitions.

Segmental effects can be approximated as perturbations on the glottal oscillator caused by changes in the environment in which it operates. While segmental effects are beyond the scope of this paper, they can be included in the model, also see Sections 1.4 and 2.6.

[3] These jumps are occasional discontinuous transitions from one mode of oscillation to another, such as modal speech to falsetto, or period doubling during glottalization. For a review of the properties of non-linear oscillators, see Pipes (1970), Moon (1987), and Ogorzałek and Maciej (1997).

Because Stem-ML is defined in physiological terms that are common to all humanity, and because we do not associate Stem-ML tags with particular language features, it has the possibility of being a language-independent description of prosody.

Stem-ML allows the existence of both phrase curves and local accents. The two concepts are distinguished by their scope. Local accents (i.e., **stress** tags) control the shape or value of f_0^* over the scope of the accent, which might be a syllable, word or stress group. Far from their center, they have little effect. The phrase curve, on the other hand, has no assumption of locality, and may be appropriate for pitch changes on scopes larger than a word.

While Stem-ML allows a description of pitch in terms of localized accents riding on a phrase curve, it does not enforce it. The system places minimal restrictions on the number of tags, the scope of tags, the location of tags, or parameter values. [4] We intend it to be theoretically neutral and language independent, so it can be used as a quantitative tool for comparing theories of prosody. As a consequence of this, a complete application that uses Stem-ML (such as a TTS system) will require a language-specific layer that defines which Stem-ML tags are associated with which linguistic events (Sections 1.4 and 5.2).

One can show that Stem-ML can represent any prosody by placing a short **stress** tag at each measured datum. As long as the tags' *strengths* are non-zero, there are then a set of equations relating the *shape* attributes to f_0 which are linear in the *shape* attributes, and can be shown to be non-singular. Then, the Fundamental Theorem of Linear Algebra shows that there is a set of *shape* attributes that will exactly reproduce the data. An equivalent proof can be constructed using one **step** tag per datum. Both proofs become straightforward if the *strengths* are large and the *smooth* parameter is small, in which case f_0^* simply follows the *shape* attribute (or the *to* attribute for **step** tags). Thus, Stem-ML is language independent, at least in the sense that it can represent the prosody of any language.

1.3. Justification

We justify the introduction of a prosody generation model on several grounds:

- It is capable of accurately reproducing any pitch trajectory in a compact, robust manner.
- It is language independent. We have used it to model languages with syllable-scope tones (e.g., Mandarin Chinese), word-scope accents (e.g., English), and we expect it can be used for languages where accents are attached to phrase boundaries.
- It is capable of representing reasonable prosodies for intimate mixtures of multiple languages. English names in the midst of a Mandarin speech stream can be tagged with English tags, and will come out with English accents. Having such linguistic flexibility for European systems is also obviously desirable. As a consequence, it can be used as a general, multi-language pitch generation component.
- It is reasonably theory neutral. For instance, Stem-ML tags can be mapped onto existing systems such as ToBI. Consequently, it should be possible to quantitatively compare different intonation systems and decide which are more successful in describing speech data.
- Stem-ML automatically meets physiological smoothness constraints on f_0^*.
- It models pre-planning of speech and interactions between neighboring accents.
- Stem-ML can represent long-range correlation in the pitch trajectory by its accent interaction rules and by optional use of phrase curves.
- It is suitable for machine fitting.

[4] For instance, you should not assume that there must be one **stress** tag per accent. The best representation may differ from language to language. Stem-ML allows you to use **stress** tags for each syllable, each word, or in arbitrary locations with arbitrary scopes. As another example, **step** tags need not be associated with phrases or sentences; they could be used to mark syllable-by-syllable prosody.

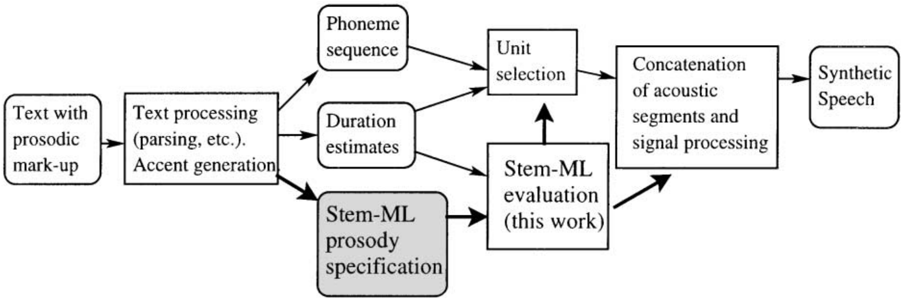

Fig. 1. A generic text-to-speech system, showing where Stem-ML modeling might be used.

1.4. Where does it fit in a TTS system?

When used in a TTS system (e.g. Fig. 1), this model interprets a tag set (Stem-ML, level 1) in the middle of the prosody subsystem. Input text contains a broader set of Stem-ML (level 2), not yet defined, that controls prosody through linguistic definitions. For example, some of these higher-level tags might approximate the ToBI mark-up scheme (Pierrehumbert, 1980; Beckman and Ayers, 1997; Silverman et al., 1992). The input text might alternatively comprise other languages that provide a high-level description of the prosody of a text stream, such as SSML (Taylor and Isard, 1997) and SABLE (Sproat et al., 1998). These languages are broad descriptions of prosodic intent while Stem-ML is a detailed description of pitch movement. In general, Stem-ML and these languages are complementary, and could work in tandem in one system.

The prosody subsystem contains two or three components:

- A linguistic modeling component to convert Stem-ML level 2 tags into level 1 tags. This component contains models for discourse and phrasal intonation, including microprosody of domains such as lists, movie titles, proper names, and numbers. It will model questions, mark new and important words in the discourse, and model requests for confirmation. This component also uses a lexicon to mark accent positions. Its output is a structure in memory that corresponds to text marked with Stem-ML level 1 tags.
- A pitch generation component that takes the Stem-ML level 1 tagged text and produces a time series of pitch values.
- A segmental effects component that calculates how f_0 depends on the phoneme sequence (Section 2.6). At the current state of the art, this component is optional, as segmental effects do not seem to have a major influence on the intelligibility of TTS systems, despite the fact that segmental effects can be perceptible and can help humans to recognize phonemes (Hillenbrand and Houde, 1996; Haggard et al., 1970; Hombert, 1978; Massaro and Cohen, 1976).

This document focuses on the pitch generation component, and defines all Stem-ML level 1 tags.

1.5. Outline of the algorithm

Stem-ML serves the dual function of being a prosody mark-up language and a pitch generation system. From the user's point of view, the system is a collection of tags. These tags can be used to describe prosodic events such as phrase curve, accents, properties of accents, and how different components combine to create

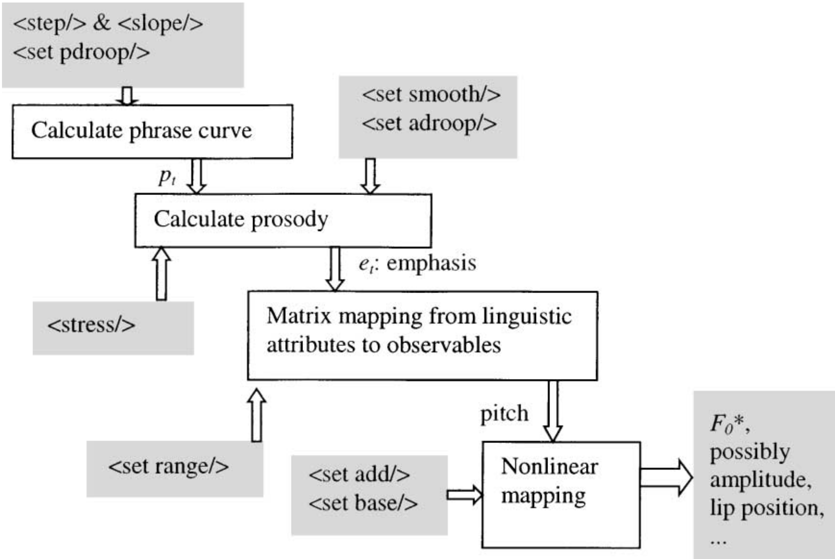

Fig. 2. A block diagram showing the Stem-ML algorithm. The white boxes show the steps of the algorithm. The gray boxes show input data and results.

the surface pitch contours. Internally each tag is defined mathematically with parameter settings describing variations.

Fig. 2 is the block diagram of the Stem-ML algorithm. The steps are

- calculate the phrase curve,
- calculate the prosody, relative to the phrase curve,
- map from an abstract description of prosody to observable quantities.

The gray boxes show the tags that influence each step. For example, `<step/>` and `<slope/>` are two types of tags that can be used to define phrase curves, and the `<stress/>` tags allow users to specify tone or accent templates.

Each tag puts a set of constraints on the prosody. A set of built-in constraints enforce smoothness and continuity of f_0^*. The algorithm accumulates constraints, then calculates the prosody that best meets the constraints. Each tag can have a different strength, and the strengths control how the system compromises between any conflicting constraints. One can look at the model as an implementation of elastic templates that compromise with their neighbors. We will describe the mathematical basis in Section 2, which will be followed by detailed description of the tags (Section 3). Examples showing tag usage and surface pitch variations corresponding to the parameter settings are given in Section 4.

2. Mathematical basis

We calculate the prosody by building a set of linear equations involving the pitch at every instant, then solving that set of equations. The equations can be divided up into several groups, depending on their origin. The first group of equations expresses the overall smoothness and continuity of the pitch curve. Each tag adds another group to describe its constraints on the pitch curve. When the set of equations cannot all be satisfied exactly (which is the common case), Stem-ML returns a pitch curve that compromises among the constraint equations.

Technically, the algorithm implements a regularized fit to soft constraints, by way of a least-mean-square solution of the constraint equations. It calculates one phrase at a time, and enforces continuity at phrase boundaries. The algorithm proceeds in four stages:

- First, it accumulates constraints on the phrase curve, then the resulting set of linear equations is solved to yield the phrase curve which best matches the constraints. The constraints come from **step** and **slope** tags.
- Second, the system accumulates constraints on the pitch trajectory, and solves for the optimal pitch at each time. These constraints come from **stress** tags and the phrase curve.
- Third, we map from a linguistic representation of prosody into the observables.
- Finally, we apply non-linear transformations to match human perception.

Note that points on both the phrase curve and the pitch trajectory can be vectors, controlling several observable components of prosody, like f_0^* and amplitude.

2.1. Phrase curve calculation

The first group of equations in the phrase curve calculation constrains the curve to be continuous. There is one equation for each time t, that relates each point to its neighbor: $p_{t+1} - p_t = slope_t \cdot \Delta t$, where p_t is the phrase curve, $slope_t$ is the *rate* attribute of the nearest preceding **slope** tag (Sections 3.3 and 4.2), and Δt is the interval between prosody calculations (typically 10 ms). Often, the slope is zero, and then these equations can be interpreted as requiring each point to be close to its neighbor, which implies continuity. All these equations have a fixed strength: $s_{[\text{continuity}]} = 0.01/\Delta t$ (Δt is measured in s). This group of equations has the side effect of enabling automatic interpolation between **step** tags (see Fig. 7).

Each **step** tag (Sections 3.2 and 4.1) adds a group of two equations to the set of constraints: $p_t = to$ and $p_{t+w} - p_{t-w} = by$, where $w = 1 + \lfloor smooth/2\Delta t \rfloor$ (rounding down) is half of the smoothing width (Sections 3.1 and 4.8), t is the position of the tag, and by and to are the tag's attributes. These equations allow you to specify the value of the phrase curve (via the *to* attribute) and/or to place steps in the phrase curve (with the *by* attribute). **Step** tags can be used to draw an arbitrary phrase curve. Each of these equations has a strength (defined below). The strength controls how closely the solution matches the tag. In the common case, where tags are widely spaced, any $strength \gg 1$ will cause the tag to be followed accurately.

Finally, when *pdroop* (Sections 3.1 and 4.3) is non-zero, we add one equation at each point that pulls the phrase curve down toward zero: $p_t = 0$. The droop equations typically have a very small strength individually: $s_{[\text{droop}]} = pdroop\Delta t$, but they act together to eventually bring the phrase curve down. *Pdroop* might be used to implement declination.

Overall, there are n unknowns (one p_t at each time point), and there is one droop equation for each, along with $n - 1$ continuity equations, and with two equations per **step** tag. There are more equations than unknowns, so the system is over-determined and we must find the solution that comes closest to matching all the constraints. We use a least-squares solution to implement the compromise.

The equations can be written in matrix form as $s \cdot a \cdot p = s \cdot b$, where s is the m by m diagonal matrix of strengths, a (a is m by n) contains the coefficients of the p_t in the equations, and b (which is m by 1) contains the right-hand sides of the equations (the constants). P is a (m by 1) column vector. M is the number of equations.

We transform the equations into normal form for solution, $a^t \cdot s^2 \cdot a \cdot p = a^t \cdot s^2 \cdot b$, because the left-hand side then contains a band-diagonal matrix ($a^t \cdot s^2 \cdot a$), with narrow bandwidth (superscript t denotes a matrix

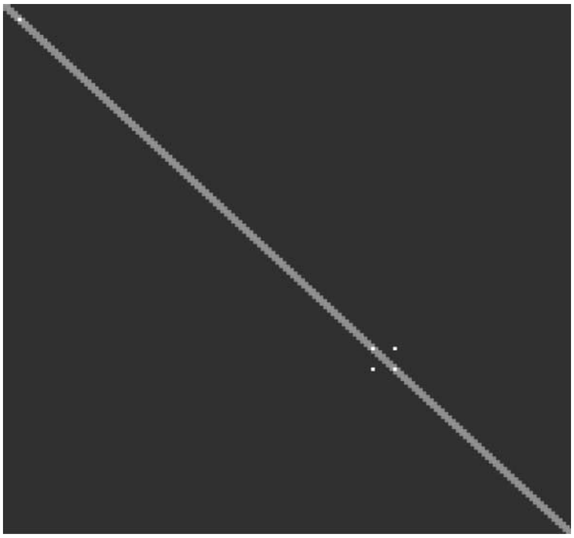

Fig. 3. Magnitude of the elements of $a^t s^2 a$ for the example shown in Fig. 11 (curve #2). Brightness increases with the magnitude of each matrix element; black is zero. Elements near the main diagonal (upper L to lower R) correspond to equations that relates nearby points on the phrase curve, and in general, the (i, j)th element corresponds to an equation that relates the ith and jth points on the phrase curve.

transpose). That bandwidth is no larger than w, which is typically much smaller than n or m. The narrow bandwidth is important because the cost of solving the equations scales as $w^2 n$ for the band-diagonal case, rather than n^3 for the general case. In our application, that scaling reduces the computational costs by a factor of 1000, and assures us that the number of CPU cycles per second of speech will be constant.

Fig. 3 shows the magnitude of the elements of $a^t \cdot s^2 \cdot a$ in an example calculation of a phrase curve (Fig. 11). The band-diagonal form is clearly seen. The bright spot on the diagonal in the upper left corner comes from an initial **step to** tag, and the four bright points near the middle of the image come from a **step by** tag at $t = 1$ s. The diagonal stripe comes from the continuity equations, which relate each point to its neighbors.

Example: Assume a sampling interval of $\Delta t = 0.01$ s, *smooth* $= 0.04$ s, *pdroop* $= 1$, and tags
`<slope rate=1 pos=0s/>`,
`<step to=0.3 strength=2 pos=0s/>`,
`<step by=0.5 pos=0.04 strength=0.7/>`.
One then gets the following set of equations:

1: $p0 = 0.3$; $s1 = 2$ # **step to**
2: $p6 - p2 = 0.5$; $s2 = 0.7$ # **step by**
3: $p1 - p0 = 0.01$; $s3 = 1$ # **slope**
4: $p2 - p1 = 0.01$; $s4 = 1$ # **slope**
5: $p3 - p2 = 0.01$; $s5 = 1$ # **slope**
6: $p4 - p3 = 0.01$; $s6 = 1$ # **slope**
...
11: $p0 = 0$; $s11 = 0.01$ # *pdroop*
12: $p1 = 0$; $s12 = 0.01$ # *pdroop*
13: $p2 = 0$; $s13 = 0.01$ # *pdroop*
...

The matrix a is then

$$a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ & & . & . & . & & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & . & . & . & & & & \end{bmatrix},$$

where each row corresponds to the left-hand side of one of the equations above. Each column corresponds to a time value. The right-hand side of the equations above goes into the b matrix,

$$b = \begin{bmatrix} 0.3 \\ 0.5 \\ 0.01 \\ 0.01 \\ 0.01 \\ 0.01 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}.$$

Each row, again, corresponds to one of the equations above. The diagonal elements of the strength matrix are

$$s_{i,i} = [2 \quad 0.7 \quad 1 \quad 1 \quad 1 \quad 1 \quad \ldots \quad 0.01 \quad 0.01 \quad 0.01 \quad \ldots],$$

where each entry corresponds to one equation.

In between phrases, the pitch must also be continuous. We enforce the physiological requirement of continuity between phrases by beginning the calculation of phrase 2 a little early, so that it overlaps the end of phrase 1, then taking values of the phrase curve and prosody which are known from the end of phrase 1 and substituting them into the beginning of phrase 2. This technique enforces a strictly causal relationship between phrases so that later phrases smoothly follow from earlier phrases, yet tags in the later phrases cannot affect the results of earlier phrases.

2.2. Pitch trajectory calculation

The next step is to calculate the prosody, e_t, based on the phrase curve and **stress** tags (Sections 3.4 and 4.4). In a simple text-to-speech system that only predicts pitch, the prosody is essentially the pitch trajectory. It contains all the peaks and valleys, and may differ from the pitch only by a simple scaling. We follow the same procedure as we did for the phrase curve (Section 2.1), though we end up solving a different set of equations. As before, a group of continuity equations apply at each point: $e_{t+1} - e_t = 0$, with a fixed strength $s_{[\text{continuity}]} = 0.01/\Delta t$. An additional group then expresses smoothness: $-e_{t+1} + 2e_t - e_{t-1} = 0$, each with a strength

$$s_{[\text{smooth}]} = \frac{\pi}{2} \cdot \frac{smooth}{\Delta t} \cdot \frac{0.01}{\Delta t}$$

(see Sections 3.1 and 4.6). The smoothness equations imply that there are no sharp corners in the pitch trajectory. Mathematically, they ensure that the second derivative stays small, which comes from the physical constraint that the muscles used to implement prosody all have a non-zero mass, therefore they must be smoothly accelerated and cannot respond jerkily.

As before, there is also a group of N droop equations, $e_t = p_t$, with strength $s_{[\text{droop}]} = adroop \cdot \Delta t$ (see Sections 3.1 and 4.7). These equations pull the pitch trajectory toward the phrase curve, much like *pdroop* pulls the phrase curve toward zero. This group can be interpreted as stating that **stress** tags have local effects, and that to some degree, the pitch will tend to follow the phrase curve, at least on time scales longer than $1/adroop$.

Next, each **stress** tag adds a group of equations: one equation that constrains its mean pitch relative to the phrase curve, and a set of equations that locally constrain the shape of the pitch trajectory. To derive these equations, the *shape* attribute of the **stress** tag is first linearly interpolated to form a dense array of target values. An accent defined by $shape = t_0x_0, t_1x_1, t_2x_2, \ldots, t_jx_j$ is interpolated to X_k, $X_{k+1}, X_{k+2}, \ldots, X_J$, where $k = t_0/\Delta t$ is the index of the first point of the accent's shape, and $J = t_j/\Delta t$ the index of the end of the accent. [5] We then define the accent template to be $Y_t = X_t + p_t$: the sum of the shape and the phrase curve. The equation that constrains the accent's mean pitch is then $\sum_{i=k}^{J} e_i = \sum_{i=k}^{J} Y_i$, with a strength $s_{[\text{pos}]} = strength \cdot \sin(type \cdot \frac{\pi}{2})$. As *type* increases from zero, one can see that the strength of this equation also increases from zero (which means that the accent doesn't care about its mean pitch), to *strength* when $type = 1$. See Sections 4.4.1, 4.4.2 and 4.5 for descriptions of *strength* and *type*.

There is also one equation for each point in the accent (i.e., from k to J). These equations define the shape of the accent: $e_i - \bar{e} = Y_i - \overline{Y}$, where $\bar{e} = \sum_{i=k}^{J} e_i/(J - k + 1)$ is the average value of the pitch trajectory over the accent, and $\overline{Y} = \sum_{i=k}^{J} Y_i/(J - k + 1)$ is the average pitch target of the accent. Subtracting the average values prevents these equations from constraining whether the accent sits above or below the phrase curve; the intent is to constrain just the shape. Each of these equations has strength

$$s_{[shape]} = strength \cdot \cos\left(type \cdot \frac{\pi}{2}\right)(j + 1)/(J - k + 1).$$

One then builds the a and b matrices and solves them, exactly analogously to the phrase curve. The bandwidth of these matrices is generally somewhat larger, as accents can be wider than the smoothing width, but one still sees a 100× speedup for the band-diagonal calculation relative to the general solution.

Fig. 4 shows the magnitude of the elements of the $a^t s^2 a$ matrix in an example calculation of e_t. Points near the diagonal show the coupling of prosody at nearby times; points further off the diagonal show longer-range interactions. The boxes correspond to the scope of each **stress** tag. The upper left box corresponds to the first, strongest **stress** tag: it is brightest, indicating that it has the largest strength and provides the tightest constraint the prosodic trajectory. The central band is wider than in Fig. 2, because the smoothness equations have been added to the set.

2.3. Optimization representation vs. constraint equations

The constraint equations can be cast into an equivalent optimization problem with an interesting interpretation. One can prove, by a rearrangement of the normal equations, that the equation $E = (a \cdot e - b)^t \cdot s^2 \cdot (a \cdot e - b)$ gives a minimum value of E for the same e that solves the constraint equations. So, finding e by minimizing E is equivalent to solving the constraint equations, but it is easier to interpret.

[5] Accents that extend outside a phrase are truncated at the phrase boundary.

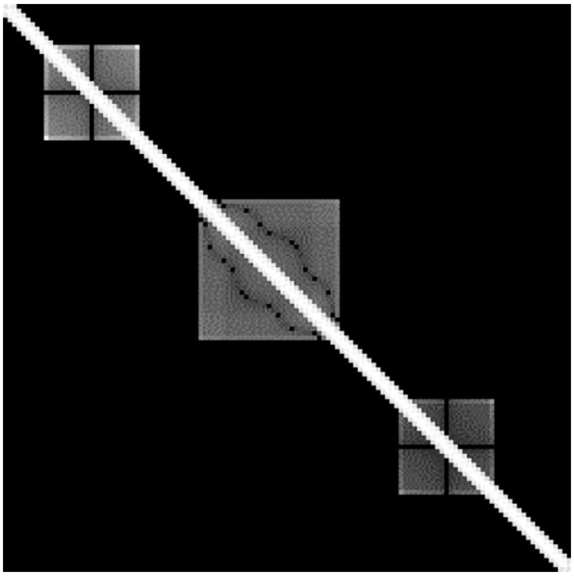

Fig. 4. The magnitude of elements of the $a^t s^2 a$ matrix for calculation of one of the pitch curve in Fig. 23, with the medial falling tone having a *strength* $= 3$. Black is zero. The central white band corresponds to the continuity, smoothness, and droop equations, while the three gray boxes correspond to the equations that define the shape and positions of the three accents.

We can break up equation for E, above, by selecting groups of rows of a and b. These rows correspond to sets of constraint equations, and E will be a sum over its fragments. The most interesting and suggestive way to break E is to separate out the continuity, smoothness, and droop equations into one group (we shall call it *effort*), and leave the constraint equations that come from tags in another (which we shall call *error*). Then, one can identify $E = effort + error$.

Qualitatively, the *effort* term behaves like the physiological effort: it is zero if the muscles are stationary in a neutral position, and increases as muscular motions become faster and stronger. Likewise, the *error* term behaves like a communication error rate: it is minimal if the prosody exactly matches the ideal target, and increases as the prosody deviates from the ideal. As the prosody deviates from the ideal, one expects the listener to have an increasingly large chance of misidentifying the accent or tone shape.

For tags with large strength, the *error* term increases steeply as the pitch deviates more from the target. The optimal solution will then have relatively small deviations. For weak tags, on the other hand, the *error* term is unimportant: it's OK for the pitch to deviate from the target, so long as the generated pitch is smooth and requires little effort to produce.

It seems reasonable that, while speaking, humans should attempt to minimize something like E. Certainly, when we speak, we wish to be understood, so we have to consider the error rate in the overall speech communication channel (speaker $\Rightarrow$ environment $\Rightarrow$ listener). Likewise, much of what we do is done smoothly, with minimum muscular energy expenditure (as displayed by the popularity of chairs and automobiles), so minimizing effort in speech is also a plausible goal. We suggest that this form of the model may provide some insight into the mental processes involved in speech generation.

2.4. Mapping linguistic concepts into observables

At this point, we have a time-varying prosody, which can correspond to the tension or extension in a group of muscles. The rest of the algorithm approximates the mapping of this hard-to-observe prosody into

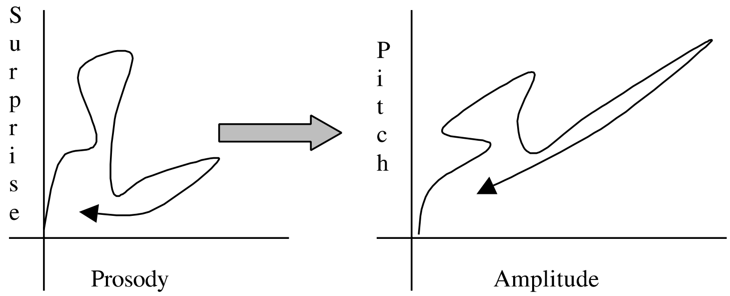

Fig. 5. Schematic example of mapping from linguistic coordinates to observables. The figure shows the time course of "surprise" and "prodosy" of a hypothetical utterance, and the corresponding outputs ("pitch" and "amplitude"). The matrix multiplication used in Stem-ML allows for cross-correlations between variables.

acoustic observables like f_0 and amplitude. In a simple implementation, the rest of the algorithm might approximate the oscillation frequency of the vocal folds as a function of muscle tensions.

From here, we assume that there are statistical correlations between the time-varying prosody we predict, e_t, and observable features in the speech signal. Since e_t is, in general, a vector, we simply multiply it by the matrix of cross-correlations, M. M is derived from **set** *range* (Section 3.1).

This matrix-mapping step can also be used to include correlations between acoustic variables that are known from physiological experiments. For instance, f_0 has been shown to increase with subglottal pressure at a rate of roughly 5 Hz/cm H_20 (Ladefoged, 1962; Ohala and Hirano, 1967; Lieberman et al., 1969). If Stem-ML is being used to model the amplitude of speech or other characteristic that is roughly equivalent to subglottal pressure, its correlation with f_0 can be included simply by setting the appropriate off-diagonal matrix element, as shown in Fig. 5.

2.5. Non-linear transformation and add setting

The relationship between pitch (measured as frequency) and the perceptual strength of an accent is not necessarily linear. Nor is there a linear relationship between neural signals or muscle tensions and pitch (see Fujisaki, 1988; Titze, 1993a). Consequently, any model of the pitch generation process needs to include the possibility of a non-linear mapping between the intended effort or attempted prominence and the final acoustic output.

To implement a controllable, generic non-linearity, the results from the previous stage, $e_t \cdot M$, are operated on by the function $f(x) = base \cdot (1 + \gamma x)^{1/add}$, where $\gamma = (1 + range/base)^{add} - 1$. This is an ad-hoc function that can smoothly describe linear behavior ($add = 1$), exponential ($add \to 0$), or behaviors in between. Always, $f(0) = base$ and $f(1) = base + range$. Each observable can have a different non-linearity, controlled by the appropriate component of the **set** *add* tag (Section 3.1).

Fig. 6 shows the effect of varying *add* values. It plots $f(x)$ vs. x, with the *add* parameter covering the range of normal use with values of 0.0, 0.5, 1.0 and 2.0.

2.6. Calculating segmental effects

We do not attempt to model segmental effects with Stem-ML tags. Segmental effects are caused by phoneme-dependent muscle control, changes of acoustic impedance, and changes in air pressure across the glottis as the articulators move to make different speech sounds. The cause of these effects is largely separate from the intentional control of f_0^*, and the two should be accounted for by separate mechanisms.

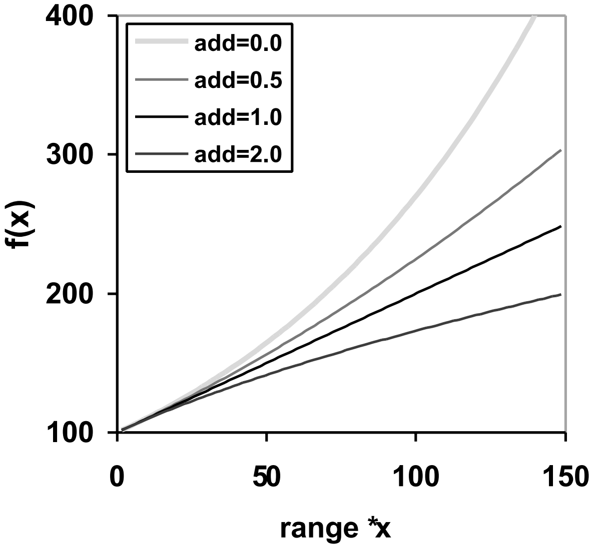

Fig. 6. Example traces of $f(x)$ with $base = 100$, for various values of *add*.

However, they could be included as reasonable extensions to the overall system. On one extreme, if one wanted to calculate segmental effects from a physical model of the larynx, (e.g., Titze, 1988 or 1989), one would need to supply the laryngeal model with values of subglottal pressure, effective vocal fold stiffness and possibly pre-phonatory glottal width. To the extent that the cricothyroid muscle is used for both voicing and f_0 control (Löfqvist et al., 1989), it could be included too. Stem-ML models could be built for each to approximate these quantities, since each quantity should have similar smooth dynamics. Approximations to the flow resistance and aerodynamic quantities of the upper vocal tract could then be based on the current phoneme, and the detailed physical model of the larynx could be evaluated to yield f_0. Essentially, such a detailed model would replace the ad-hoc mapping and non-linearity described in Sections 2.4 and 2.5.

On the other extreme, segmental effects derived from a machine learning system could be simply added onto f_0^* after the non-linear mapping. The machine learning system could be trained to predict the difference between Stem-ML's smooth f_0^* result and actual data for f_0 as a function of phoneme and neighboring phonemes.

Finally, in large-database TTS systems, the segmental effects may come automatically from the acoustic data. If acoustic units are selected on the basis of predicted f_0, and then are played without f_0 modification, units will carry their original segmental effects. It is plausible that the original segmental effects will be approximately correct and perceptually reasonable in their final context.

3. Stem-ML tags

We now turn to the definition of Stem-ML tags. These are low-level tags (level 1) that can be used to describe intonation contours. These tags may be used to define a higher-level language (level 2) that corresponds to language specific or situation specific events.

Stem-ML level 1 tags fall into four categories:

1. setting parameters,
2. defining the pitch curve,

3. marking accents,
4. marking boundaries.

3.1. Tags: set

Set accepts the following attributes (see Section 2 above for mathematical definitions):

- *max* = `value`: sets the maximum frequency (in Hz) that the voice (or the TTS system) should be allowed to produce. One value per phrase. Default = 550.
- *min* = `value`: sets the minimum frequency (in Hz) that the voice or TTS system should be allowed to produce. One value per phrase. Default = 40.
- *smooth* = `value`: sets the smoothing time of the pitch curve, in seconds (see Sections 2.2 and 4.6). This is also used to set the width of a pitch step (see Section 2.1). The same value of *smooth* is used for an entire phrase. Default = 0.06.
- *base* = `value`: sets the speaker's baseline, in Hz. The baseline sets the frequency in the absence of any tags. *Pdroop* causes f_0^* to droop toward the baseline. Typically 100 Hz for males, 200 Hz for females. This has a single value during a phrase. Default = 150.
- *range* = `mvalue`: [6] sets the speaker's pitch range, in Hz. All changes and most settings are measured as fractions of the speaker's range. Typically 150 Hz for males, 250 Hz for females. This has a single value during a phrase. Default = 200.
- *pdroop* = `value`: sets the phrase curve's droop rate toward the *base* frequency (see Sections 2.1 and 4.3). In units of fractional droop per second. Useful values range from 0 to 2. Default = 0.25. This has a single value during a phrase.
- *adroop* = `value`: sets the pitch trajectory's droop rate toward the phrase curve (see Sections 2.2 and 4.7). In units of fractional droop per second. Useful values range from 0 to 10. Default = 3. This has a single value per phrase.
- *add* = `value`: sets the non-linearity in the mapping between the pitch trajectory and f_0^*. *Add* = 1 is a linear mapping, where an accent will give the same f_0^* shift if it is riding on a high-pitch region or a low-pitch region. *Add* = 0 implies addition of $\log(f_0)$, so small accents will make a larger change to f_0^* (measured in Hz) when riding on a high phrase curve. *Add* > 1 gives a slower-than-linear mapping. Default = 0.5. See Sections 2.5 and 4.10.
- *jitter* = `value`: sets the RMS magnitude of the pitch jitter, in units of fractions of the speaker's range. One value per phrase. Default = 0. See Section 4.9.
- *jittercut* = `value`: sets the time scale of the pitch jitter, in units of seconds. The pitch jitter is correlated $(1/f)$ noise on intervals smaller than *jittercut*, and is uncorrelated (white) on intervals longer than *jittercut*. Large values of *jittercut* imply longer, smoother variations in pitch; small values imply short, choppy pitch changes. Set once per phrase. Default = 1. See Section 4.9.

Arguments given to the **set** tag are remembered until the TTS channel is closed, even across phrase boundaries.

---

[6] Generally, an `mvalue` can contain a matrix (see Appendix A.1). By default, however, it is interpreted as a single floating point number that controls the pitch range (i.e., by default, you specify the 'eF' component). We define *range* as a matrix to cleanly express correlations among various aspects of prosody. For example, pitch and amplitude are often correlated, and likewise the mouth tends to be open wider for high amplitude speech. These correlations are expressed as off-diagonal elements in the matrix. Use of a matrix here also gives the user the ability to write tags in terms of more linguistic concepts like 'emphasis' or 'suspicion', and letting the system map to observables like 'f_0^*', 'amplitude' and 'mouth opening'. See Maekawa (1998).

3.2. Tags: step

The **step** tag takes several arguments, and operates on the phrase curve (see Sections 2.1 and 4.1):

- *by* = `value`. Steps are specified as a fraction of the speaker's range. The step in the phrase curve will appear as a smoothed step in the pitch output. The default value is zero.
- *to* = `value`. Force the phrase curve to have a certain frequency at the tag's position, specified as a fraction of the speaker's range. The default value is zero.
- *strength* = `value`. Controls how the step interacts with its neighbors. The default value is 1.
- *type* = `value`. Controls whether the target value or the size of a step is the strongest constraint. If it is important that the phrase curve should reach a particular value, then set *type* = 1. Alternatively, if the size of the step is critical, then set *type* = 0. Intermediate values let one control both the mean pitch and shape. If *by* and *to* are both specified, *type* defaults to 0.5; if just *by* is specified, *type* defaults to 0; if just *to* is specified, *type* defaults to 1. These defaults allows the **step** tag to behave sensibly for the inputs `<step to="0.3"/>` and `<step by="0.4"/>`, along with a more fully specified tag like `<step to="0.3" by="0.4" strength="1.3" type="0.4"/>`.

For convenience, we call `<step to=X/>` (i.e., *type* = 1) a **step to** tag, and `<step by=Y/>` (i.e., *type* = 0) a **step by** tag, though the Stem-ML interpreter treats them as endpoints of a contiuum.

3.3. Tags: slope

The **slope** tag takes one argument, and operates on the phrase curve (see Sections 2.1 and 4.2):
- *rate* = `value` "%"?: sets a rate of increase (or decrease) for the phrase curve. It is measured as a fraction of the speaker's range per second. If the "%" mark is present, it is measured as the fraction of range per length of the phrase. Common values are between −1 and 1. Default = 0.

3.4. Tags: stress

The **stress** tag defines the prosody relative to the phrase curve (see Sections 2.2 and 4.4). Think of **stress** tags as elastic objects, welded together. 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 with the hard-wired requirement that the pitch curve must be smooth. Their behavior will become clearer when we give examples in Section 4.4. **Stress** tags accept the following attributes:

- *shape*. This specifies the ideal shape of the accent curve. This is the shape in the absence of compromises with other **stress** tags and constraints. (See Appendix A.1 for syntax.)
- *strength* = `value`. Corresponds to the linguistic strength of the accent. Accents with zero strength have no effect on pitch. Accents with strengths much bigger than 1 will be followed accurately, unless they have strong neighbors. Useful values are between 0 and 10. Default is 1.
- *type* = `value`. 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 it 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 to varying degrees. Default is 0.5.

3.5. Tags: phrase

The **phrase** tag inserts a phrase boundary. Normally, this is used to mark a phrase or breath group. No pre-planning occurs across a **phrase** tag; the prosody before it is entirely independent of whatever tags appear after it (see Section 4.8).

4. Effect of the tags

In this section we will go through the Stem-ML tags one at a time, showing their effects and how they interact. Where appropriate, we will give examples of how they can be used to model real speech data.

In all the following examples, natural f_0 contours are plotted on the y-axis as a function of time with the symbol "*". Pitch curves generated by Stem-ML tags are plotted with solid lines, and phrase curves are plotted with dashed lines. The Stem-ML tags used to generate the pitch contour are given after the examples.

In the following examples that match real data, we use symbolic representations of Stem-ML tags, following a convention resembling INSINT (Hirst et al., 2000) for convenience and clarity. However, the similarity to INSINT is superficial, especially for **stress** tags.

Accent templates (**stress** tags) are represented by Greek letters while Chinese tones in later examples are represented by numerals in outline font. Subscripts indicate their strength values. All accent templates in these examples are aligned with the center of the accented syllable or tone. Their shapes are given in the small graphs. **Phrase** tags and **stress** tags are listed on separate lines. **Slope** tags are represented as "➶", **step to** tags as "↕", step by as "⌈", and **phrase** tags by "⊣". In addition, global parameters (i.e., attributes of the **set** tag) are given in the first line. Unless noted, **slope** tags and **phrase** tags are placed between words.

4.1. Step tags

The simplest tag, and one that is a good example for how tags interact in Stem-ML is the **step** tag with the *to* attribute (known here as **step to**). This tag places a constraint on the phrase curve, requesting that the phrase curve must have a certain value at the tag's position. If a phrase contains just a single **step to** tag, the phrase curve is set to the specified value, both before and after the tag. If you now add a second **step** tag, you will see the pitch compromise in between. Each tag fixes the pitch at its location (and on the side away from its neighbor), but in between, the algorithm produces a smooth interpolation.

Fig. 7 shows three examples of using **step to** tags. The example includes a small amount of *pdroop* to allow the cases to be distinguished. Absent *pdroop*, cases 1 and 2 give the same result.

The other form of the **step** tag, with the *by* attribute (**step by**), produces a bonafide step in the phrase curve. It makes a change in the pitch, but doesn't force either side to be any particular value.

`<step by=X strength="10"/>` simply means that the pitch after the tag should be higher by `X` than the pitch before. Normally, you'd fix the pitch on one end of the phrase with a **step to** tag.

Fig. 8 is an illustration of **step by** tags. No compromising is necessary in this example, as none of the constraints imposed on the pitch curve conflict.

More complex variants of the **step** tag are possible, when both the *to* and *by* attributes are specified. These allow you to express intermediate cases, where both the absolute position and the step size are important. The *type* attribute controls whether the target position (*to*, when $type \approx 1$) or the step size (*by*, when $type \approx 0$) is more important. These complex cases are analogous to the **stress** tag, Section 3.4.

Fig. 9 is an example showing a complex phrase curve that is approximated with **step to** and **step by** tags. This is a French sentence *Elle t'a rien donné, ta mére*? "She didn't give you anything, your mother?", with a

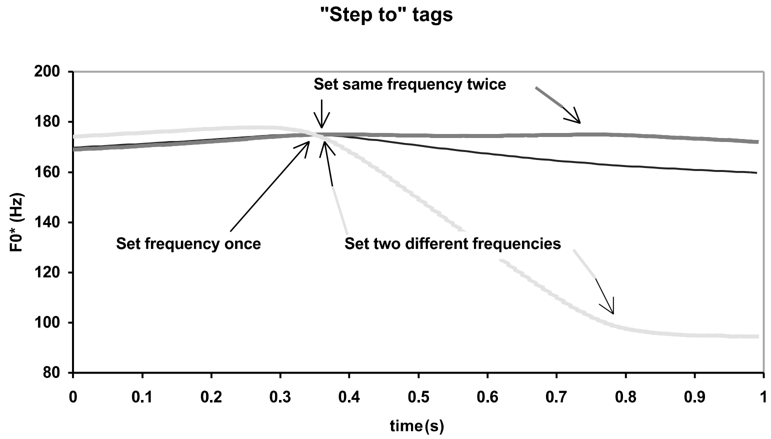

Fig. 7. Effects of the **step to** tag. The three lines are generated by `1: one tag: <step strength=10 to=0.5/> -or- ↕0.5`, `2: two tags setting the same frequency: ↕0.5 ↕0.5 -or- <step strength=10 to=0.5/>···<step strength=10 to=0.5/>`, and `3: two tags setting different frequencies: ↕0.5 ↕0 <step strength=10 to=0.5/>···<step strength=10 to=0/>`.

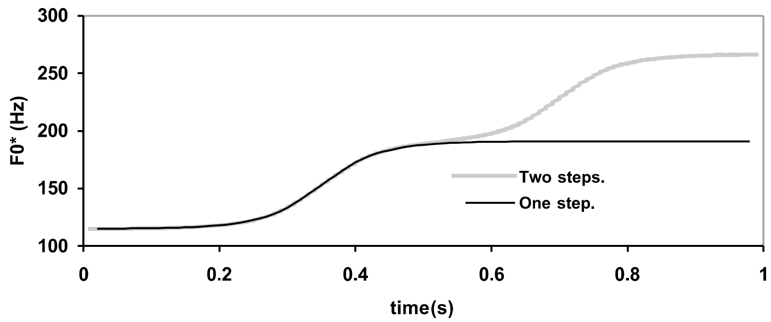

Fig. 8. Illustration of **step by** tags. Curves are generated by these tags: `Gray: <step to="0.1"/>···<step by="0.3"/>···<step by="0.3"/> -or- ↕0.5 ⌈0.3 ⌈0.3`. `Black: <step to="0.1"/>···<step by="0.3"/> -or- ↕0.1 ⌈0.3`.

dramatic incredulous rising intonation on the word *donné* starting at 99 cs, followed by a right dislocated, *ta mère* "your mother", which is another rising intonation catching the momentum of the previous rising slope, riding high near the top of the speaker's pitch range. The **step by** tag at 110 cs raises the phrase curve and supports the second rising accent in the high end of the speaker's pitch range. Alternatively, the step up at *donné* might also be represented by a pair of **step to** tags. We used an early rising accent template for the first word, *elle*, a peak accent for the word *rien*, and identical late rising accents on *t'a*, *donné*, and *mère*. The accent templates of this example are shown in Fig. 10. These templates, as well as other natural speech examples, are manually fit to the data.

Segmental effects cause discrepancies between natural and Stem-ML generated f_0 in some regions of Fig. 9. For instance, we see the raising effect of the phone *t* starting at 57 cs, and the lowering effect of phones *r*, *d*, and the final *r* starting at 70, 85 and 148 cs, respectively. The final drop in f_0 (at 150 cs) is perceptually unimportant, because it co-occurs with low amplitude. The accent is perceived as a rising one, so we use a rising template to model the f_0 curve.

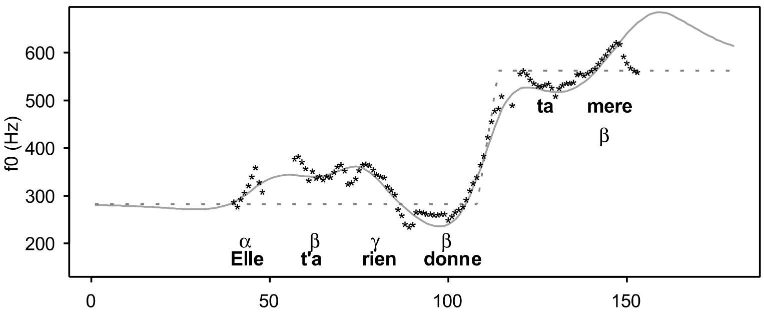

Fig. 9. The **step by** tag: raised pitch range in French incredulous question with right dislocation. See the text for the tags that generated the model in solid line.

```
Global parameters:
tag=set; add=1; smooth=0.05; base=200; pdroop=0; adroop=10; range=410;

Accent templates:
```

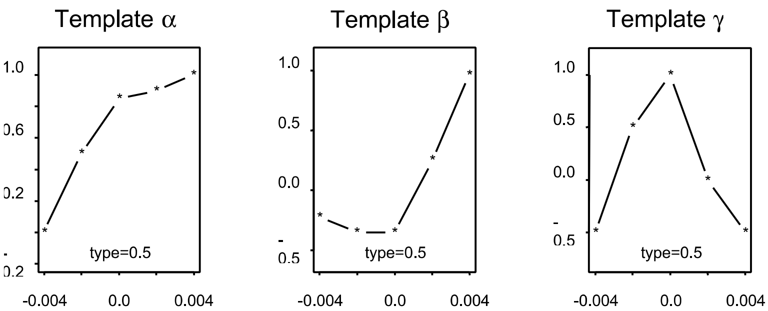

Fig. 10. Accent templates used to generate the model in Fig. 9.

```
Prosodic code: The second step tag is placed in the center
     of "ne".
Elle t'a rien donne ta mere?
```

$\updownarrow_{0.2}$ $\lceil_{0.8}$ $\dashv$

$\alpha_{0.5}$ $\beta_{0.4}$ $\gamma_{0.4}$ $\beta_{1.1}$ $\beta_{0.7}$

4.2. Slope tag

The next tag that is relevant for phrase curves is the **slope** tag. **Slope** makes the phrase curve tilt up or down to the left (forward in time) of the tag (see Section 2.1). **Slope** tags replace the current value of the *slope* attribute, so that after the sequence `<slope rate = 1/>`⋯`<slope rate = 0/>` the slope is zero. [7]

[7] Note that the strength is not specified. The **slope** tag changes the continuity equations, which always have a *strength* of 1.

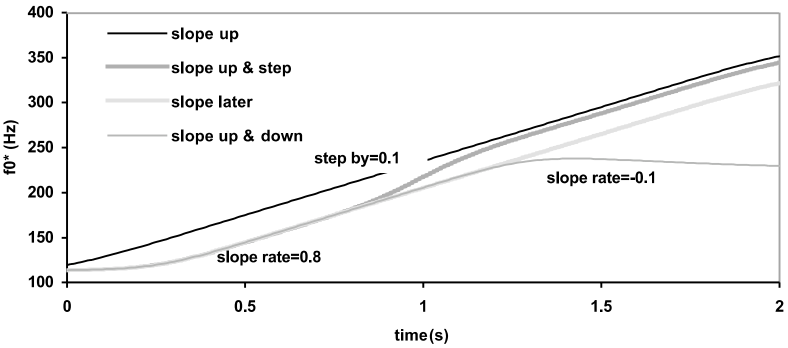

Fig. 11. Applications of the **slope** tag. The tags for each curve (from top to bottom at $t = 1.5$ s) are `1: <slope rate=0.8/>` -or- `↗0.8@t=0` `2: ···<slope rate=0.8/>···<step by=0.1/>` -or- `↗0.8 ⌈0.1` `3: ···<slope rate=0.8/>` -or- `↗0.8@t=0.3` `4: ···<slope rate=0.8/>···<set slope=−0.1/>` -or- `↗0.8 ↗−0.1`.

Fig. 11 shows some applications of the **slope** tag. We show four curves: a slope starting at the phrase boundary, one delayed 0.25 s, a slope up followed by a slope down, and a slope with a small step superposed. Again, no compromise is necessary.

Fig. 12 is an example of English coordinate structure: "(Several experts) said increased costs, and lowered chartering rates, ...". The parallelism in syntactic structure is echoed in the nearly parallel rising slopes in intonation. We implemented the rising intonation of the two coordinate phrases with slope rate of 0.13 and 0.15, placed at the first and the third vertical line, respectively. The accent templates are shown in Fig. 13. A low accent template is used on the unaccented words *said* and *and*, both showing up with low pitch. The accents of the rest of the sentence are uniformly rising, matching the use of the rising phrase curve.

A rising slope can also be expressed by a pair of **step to** tags defining the beginning and the end of the slope. For example, the following alternative expressions are roughly equivalent to the step and slope combination used above:

```
… said increased costs,
  ↕0.04                  ↕0.2
```

We note that the **slope** tag's *rate* and the *pdroop* attributes interact and it is possible to generate an unintuitive phrase curve, especially when *pdroop* is big (e.g., $\geqslant 1$).

4.3. Pdroop: phrase curve droopiness

Pdroop is a parameter that conveniently represents the systematic decrease in pitch that often occurs during a phrase. Common examples are the final phrase in a sentence, after emphasis, or the initial phrase in a paragraph. *Pdroop* operates on the phrase curve, pulling it down towards the base frequency. Points near **step to** tags will be relatively unaffected, especially if their strength is large, while points farther away will be pulled towards the base. The value of *pdroop* sets the exponential decay rate of the phrase curve, so that a step will decay away in $1/pdroop$ s. Thus, one can get a declining phrase curve by using a non-zero *pdroop* along with a positive **step to** at the beginning of a phrase (shown in Fig. 14). *Pdroop* also sets a limit to pre-planning in the phrase curve: a **step** or **slope** tag becomes largely irrelevant if it is farther than $1/pdroop$ s away. Note that *pdroop* pulls the phrase curve down just as much before a **step** tag as it does after, because we assume that the pitch trajectories are pre-planned.

Fig. 14 illustrates the effect of *pdroop*. The phrase curve is set high in the beginning, and is pulled down toward the base frequency.

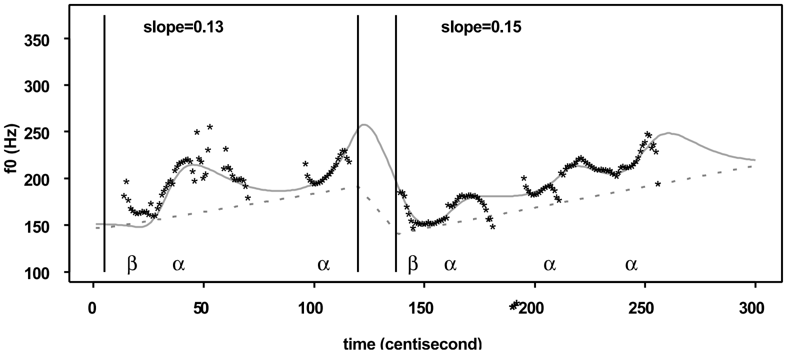

Fig. 12. The **slope** tag: rising slopes of English coordinate structure. See the text for the tags that generate the pitch curve in solid line.

```
Global parameters:
tag=set; add=1; smooth=0.06; base=135;  range=300; pdroop=0; adroop=6;

Accent templates:
```

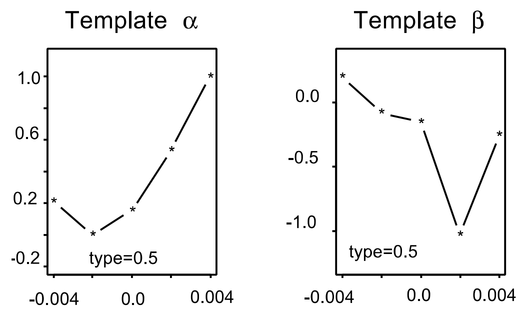

Fig. 13. Accent templates used to generate Fig. 12.

Prosodic code: A **slope** tag (↗) was placed in the beginning of each clause to generate the rising pitch movement. The pitch at the beginning of each clause is controled by a **step** tag(↕).

… said increased costs, and lowered chartering rates,

$\updownarrow_{0.04}$ ⊣ $\updownarrow_{0.01}$ ⊣

$\nearrow_{0.13}$ $\nearrow_{0.15}$

$\beta_{0.3}$ $\alpha_{0.5}$ $\alpha_{0.6}$ $\beta_{0.4}$ $\alpha_{0.3}$ $\alpha_{0.4}$ $\alpha_{0.4}$

Figs. 15 and 16 show Stem-ML fitting of two natural f_0 contours with varying declination slopes (Shih, 2000), which can be approximated with different settings of *pdroop*.

Fig. 15 is a Chinese sentence with a low tone (tone 3) at 69 cs, a rising tone (tone 2) at 84 cs, followed by 10 high-level tones (tone 1). The pitch level of the high-level tones gradually declines. We capture the

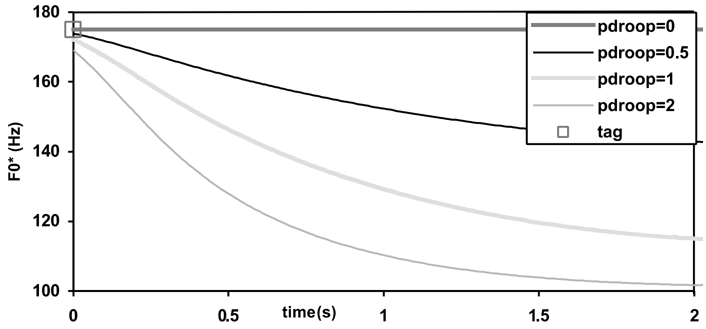

Fig. 14. The effect of *pdroop*. The phrase curve is set high at $t = 0$, and is pulled down toward the base frequency (100 Hz). The square marks the tag position. `<step to="0.5" strength="3"/> <set pdroop=various/>`.

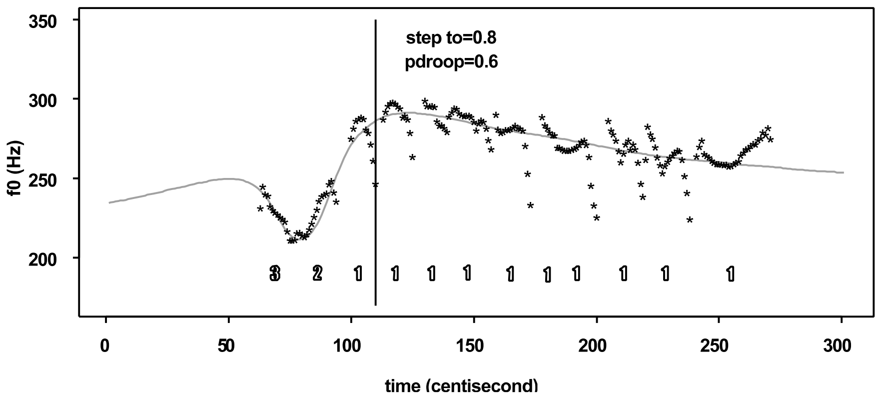

Fig. 15. The *pdroop* tag: gradual declination with *pdroop* = 0.6 for a series of Mandarin high-level tones. See the text for the tags that generate the pitch curve (—).

```
Global parameters:
tag=set; add=1;  smooth=0.06; base=175; range=120; adroop=2; pdroop=0.6;

Prosodic code: Numerals 1-4 represent Chinese tones.

Lao3 wang2 jin1 tian1 gang1 gang1 bang1 zhong1 yi1 zheng1 dong1 gua1
"Lao-Wang just help the doctor to steam winter melon today."
```

↕$_{0.5}$ ↕$_{0.8}$

$3_{0.6}$ $2_{0.9}$ $1_{0.7}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$

declination curve with a **step to** tag to 0.8 of the pitch range and a *pdroop* setting of 0.6. The vertical line in the plot marks the location of the **step to** tag.

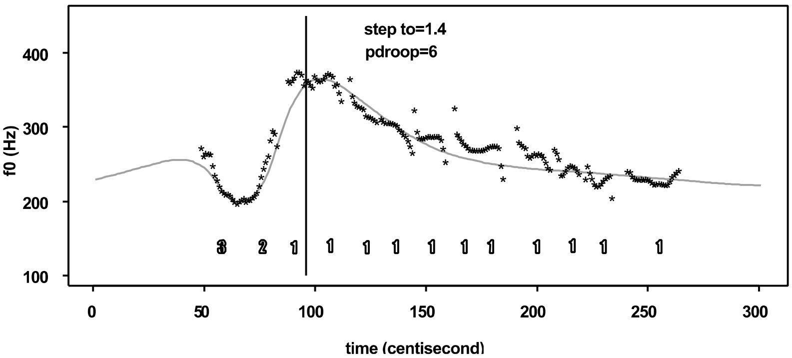

Fig. 16. The *pdroop* tag: steep declination with *pdroop* = 6 for a series of Mandarin high-level tones which follow an emphasized word. See the text for the tags that generate the pitch curve in solid line.

```
Global parameters:
tag=set; add=1;  smooth=0.06; base=190;  range=200; adroop=2;  pdroop=6;

Prosodic code:

Lao3 wang2 jin1 tian1 gang1 gang1 bang1 zhong1 yi1 zheng1 dong1 gua1 zhong1
"Lao-Wang just help the doctor to steam winter melon bowl today."
```

$\updownarrow_{0.5}$ $\updownarrow_{1.4}$ ⊣

$3_{0.6}$ $2_{0.9}$ $1_{0.7}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$

Fig. 16 has similar tonal composition as Fig. 15, but with 11 high-level tones. The high-level tones show a steep declination slope. This is captured with a **step to** to 1.4 of the pitch range and a *pdroop* setting of 6.

Most of the Stem-ML tags are kept constant between these two examples: The tonal templates and the strength specifications of all syllables are the same. The variations are accounted for by the difference in pitch range, the magnitude of the **step to** tags, and most importantly, the variation in the *pdroop* settings.

4.4. The stress tag

The **stress** tag allows you to accent words or syllables in a very general manner. You specify three things: the ideal 'Platonic' (Plato, 366 BCE) shape of the accent, which is the shape it would have without neighbors, and if spoken slowly. Second, you give the accent type. Finally, you specify the strength of the accent. Strong accents tend to keep their shape; weak accents tend to be dominated by their neighbors. Table 2 shows qualitatively how accents interact with their neighbors.

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 50 Hz". At the other extreme, *type* = 1, the shape doesn't matter, but the average pitch is important. An example might be "the pitch is 50 Hz above the phrase curve". Intermediate types are possible, and give you accents that define both a shape and a mean pitch.

Table 2
Summary of accent interactions

Accent interactions vs. *strength* and *type*	*Type* ≈ **0**	*Type* ≈ 0.5	*Type* ≈ 1
Strength ≫ neighbor's & *Strength* ≫ 1	The accent keeps its shape precisely. Neighbors will bend to accommodate it.	The accent's shape and mean pitch are precisely as specified. Neighbors must adjust.	The accent's average pitch is precisely controlled. Neighbors bend or shift to accommodate
Strength ≈ neighbor's	The shape will be a compromise with the neighboring accents. The neighbors will control average pitch.	The shape and mean pitch will be similar to the tag's specification, but both will compromise with the neighbors.	The average pitch will be a compromise with the neighboring accents. The neighbors will control the shape
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. Little smoothing is applied to the accent.		
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 is controlled by smoothness and continuity requirements.		

4.4.1. Compromises between stress tags—1

While it is normal to write a phrase curve without conflicting requirements that would cause the system to compromise, compromises abound when the pitch trajectory (prosody) is being calculated from **stress** tags. It is easy to find situations where the speaker wants to end one accent low, yet start the next one at a high pitch. Somehow, the accents need to be reshaped, or the pitch has to be adjusted. Stem-ML can do either.

In the following five figures (Figs. 17–21), we explore the interaction between two nearby accents/tones. The first is a level tone with a well-defined pitch. The second is a falling tone. We'll see in each figure how the pitch curves behave as we adjust the target pitch of the first tone. The first figure shows the response of 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 for the falling tone, until in the last figure, where its shape becomes unimportant (*type* = 1).

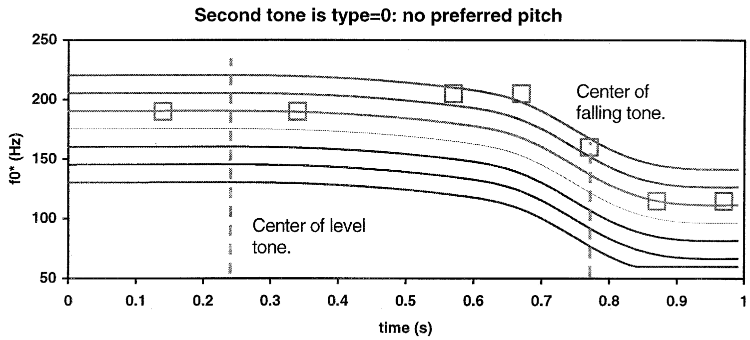

Fig. 17. A falling tone following a level tone. Note that the resulting pitch curves are parallel, because only the shape of the second tone is constrained. The lowest curve runs into the system's minimum frequency. The *shapes* of the **stress** tags are shown by the squares. `<stress strength="4" type="0.8" shape="−0.1s`***Y***`, 0.1s`***Y***`"/>···<stress strength="4" type="0" shape="−.2s.3,−.1s.3,0s0,.1s−.1,.2s−.1"/>`. We generate level tones at different heights by varying ***Y*** from −0.1 to 0.5.

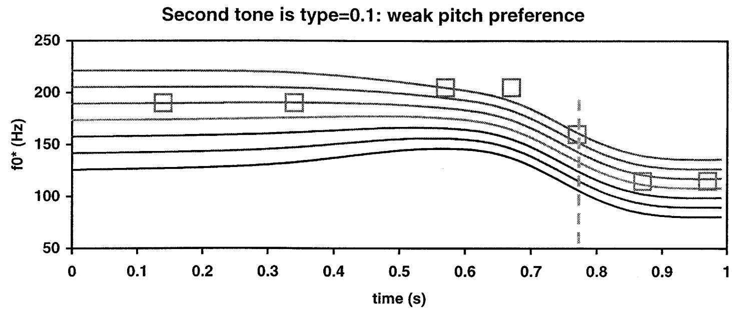

Fig. 18. A falling tone with a weak pitch preference following a level tone. The pitch curves start to bunch up on the falling tone, as its pitch preference begins to be felt. `<stress .../>···<stress type="0.1"...>`.

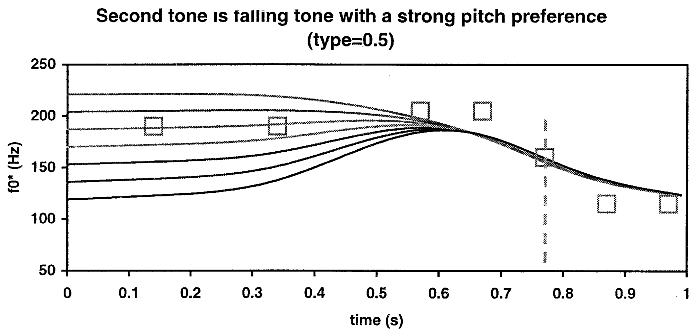

Fig. 19. The falling tone now has a strong pitch preference. It defines both its shape and pitch quite rigidly. Note that when the preceding level tone is low, the pitch now must increase in preparation for the second tone. `<stress.../>···<stress type="0.5".../>`.

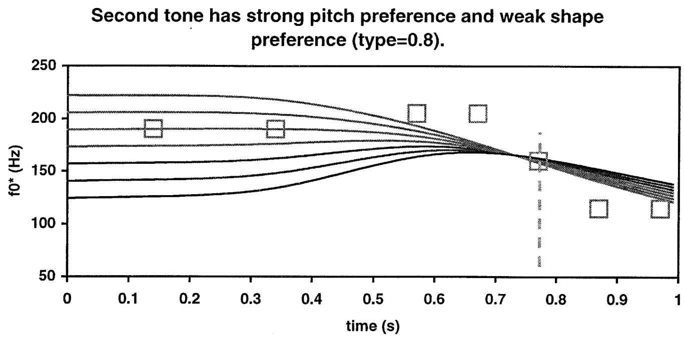

Fig. 20. With *type* = 0.8, the second tone is primarily defined by its pitch. The shape is now relatively unimportant, but the tag still manages to force the pitch to decline near its midpoint. When the first tone has a low pitch, the pitch curve now needs to rise strongly in between the two tones, so that the pitch will be right at the center of the second tone.

4.4.2. Compromises between stress tags—2

If we bring nearby accents together, we can get another example of compromises between tags. Note that Stem-ML is not an additive model: the result of putting two accents on top of each other is not the sum of the two accents. It corresponds to a single accent of the same shape and type, but twice the strength. From a

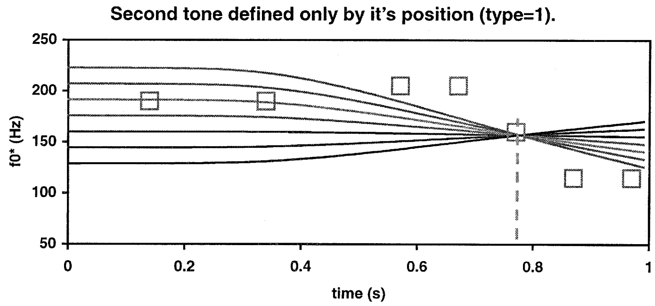

Fig. 21. In this last figure in the sequence, the second tone is defined completely by its pitch. The shape of the falling tone becomes irrelevant for *type* = 1.

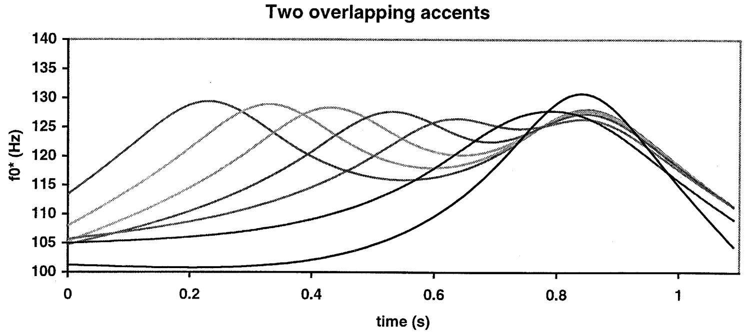

Fig. 22. Interaction of two accents. `<stress strength="4" shape="-.15s0,-.1s0,-.05s.1,0s. 3,.05s.1,.1s0,.15s0" type="0.5"/>···<stress strength="4" shape=(see above) type="0.5"/>`.

practical TTS point of view, the system avoids putting undesirable emphasis in between two nearby accents. Stem-ML can simulate the combination of two laryngeal gestures in (Munhall and Löfqvist, 1992) without the problem of a summation model. For the laryngeal opening gestures studied in that paper, simple summation of the two gestures predicts that the larynx will be open further as the two gestures overlap. On the contrary, they observe that the maximum opening is nearly constant, a natural result for a Stem-ML model. Fig. 22 shows the result of two identical accents as they are brought progressively closer together (one accent comes in from the right, the other is stationary at 0.83 s). The final, highest peak shows the two accents sitting on top of one another.

4.5. The strength of accents

In Stem-ML, all accents 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. Fig. 23 shows this effect by simulating three accents: a strong high tone, then a low 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.

In the next two examples, we show examples of tone interactions in actual speech data. Figs. 24 and 26 illustrate the variations in accent strength in Mandarin. The two examples are two renditions of the same

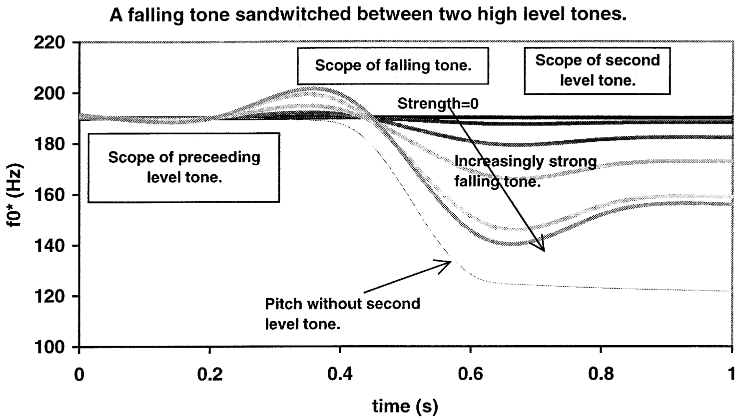

Fig. 23. The interactions between three accents as the strength of the middle one (a low-falling tone) is varied. In the black, top-most curve, the low-falling tone is unimportant with zero strength, and gradually assumes its ideal shape as its strength is increased from 0 to 4. Its neighbors are increasingly perturbed. `<stress strength="4" type="0.3" shape="-0.1s0.3,0.1s0.3"/>`··· `<stress strength=various type="0.5" shape="-.15s.2,-.1s.2,0s0,.1s-.2,.15s-.2"/>`···`<stress strength="2.5" type="0.3" shape="-0.1s0.3,0.1s0.3"/>`.

Chinese word *zang1 mao2-yi1* "dirty sweater", where the tonal combination is high level, rising, and high level. The rising tone of the middle syllable may be realized weakly, as in Fig. 24, or strongly, as in Fig. 26. The same templates are used for both examples, and are shown in Fig. 25.

The pitch discrepancies in the *zang* regions between natural and generated f_0 in both figures are consistent with the segmental effect of the phone *z*, an alveolar affricate, which raises f_0 during the beginning section of the vowel.

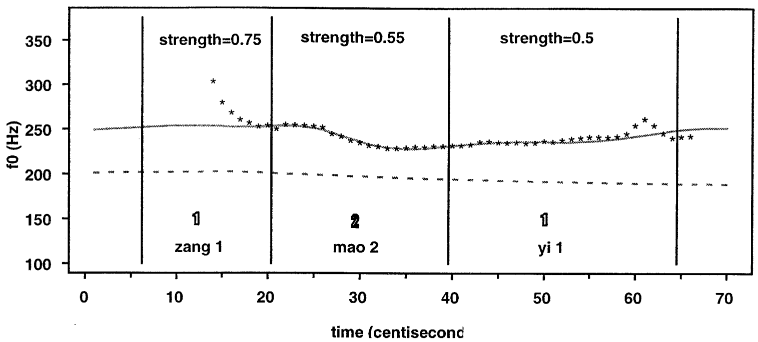

Fig. 24. Strength of accents: Mandarin example with a weak middle syllable. See the text for the tags that generate the pitch curve in solid line.

```
Global parameters:
tag=set; add=1; smooth=0.05; base=130;  range=250; pdroop=1; adroop=5;

Accent templates:
```

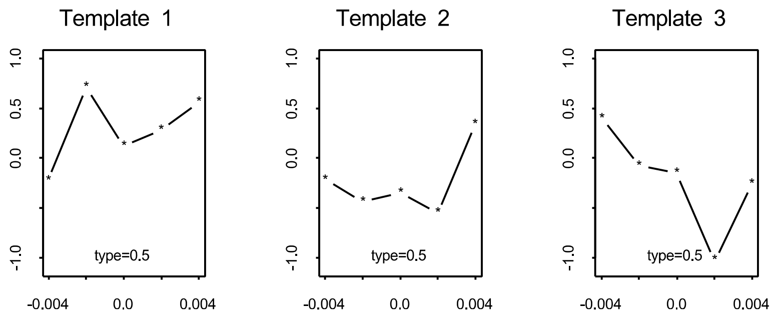

Fig. 25. Chinese tone templates used to generate Fig. 24 as well as the models in the following Chinese examples.

Prosodic code: Each syllable in the Chinese example has a tone template that is lexically determined. The templates are placed in the center of the syllable

zang1 mao2 yi1 *"dirty sweater"*

↕$_{0.3}$ ┤

1$_{0.75}$ 2$_{0.55}$ 1$_{0.5}$

The Stem-ML tags used to generate Fig. 26 are identical to the example above, except for the *strength* parameters of the syllables.

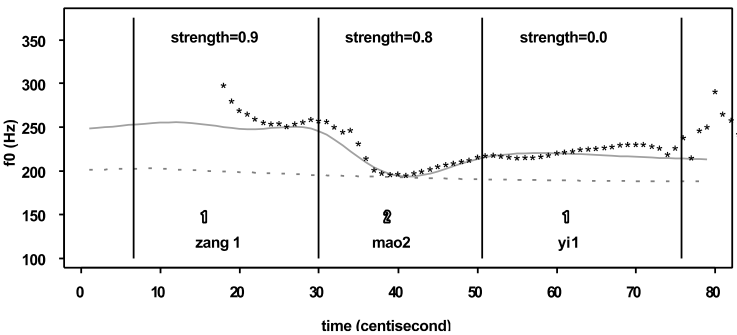

Fig. 26. Strength of accents: Mandarin example with a strong middle syllable. See the text for the tags that generate the pitch curve in solid line.

Global parameters:

tag=set; add=1; smooth=0.05; base=130; range=250; pdroop=1; adroop=5;

Prosodic Code:

zang1 mao2 yi1 *"dirty sweater"*

↕$_{0.3}$ ┤

1$_{0.9}$ 2$_{0.8}$ 1$_{0.0}$

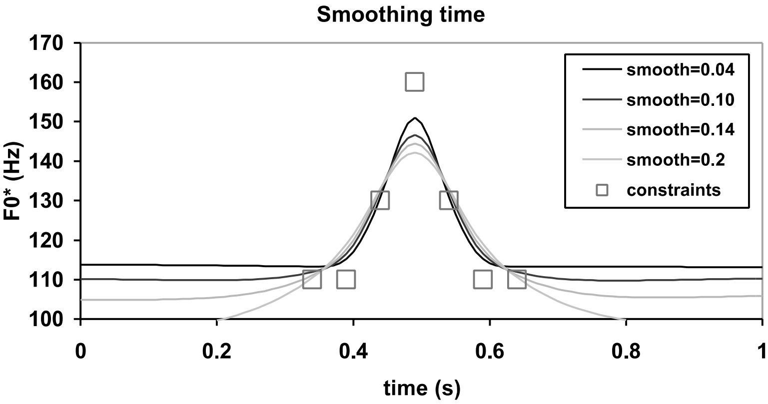

Fig. 27. An accent with different smoothing times (increasing downward at $t = 0.5$ s or upwards at $t = 0.3$ s). The open squares mark the specified shape of the accent. The curve with *smooth* = 0.2 is substantially over-smoothed, relative to the shape of the accent. `<set smooth=various/> <stress strength="4" shape="-.15s0,-.1s0,-.05s.1,0s.3,.05s.1,.1s0,.15s0" type=".5"/>`.

4.6. The smooth attribute: muscle response time

The final parameter critical for defining accents and their interactions is the *smooth* attribute, expressed in seconds. Normally, it should be set to the time it takes the speaker to change pitch (i.e., a voluntary pitch step in the middle of an extended vowel). Fig. 27 shows the effects of smoothing time on the same accent. The *smooth* attribute varies from 0.04 to 0.2.

4.7. Adroop: pitch trajectory droops toward the phrase curve

The *adroop* parameter is closely analogous to *pdroop*, except that *adroop* pulls the pitch trajectory toward the phrase curve. It allows you to limit the amount of pre-planning that Stem-ML assumes. Accents farther than $1/adroop$ seconds away from a given point will have little effect on the local pitch trajectory. [8] Fig. 28 illustrates the effect of the *adroop* attribute.

4.8. The phrase tag: limiting pre-planning

Phrase tags mark boundaries where pre-planning stops; they are normally placed at phrase boundaries. Stem-ML assumes that people are capable of planning their prosody a few syllables in advance of its actual production. This pre-planning lets the speaker smoothly compromise between difficult tone combinations and also helps him or her avoid running above or below their comfortable pitch range. **Phrase** tags allow you to control the scope of advance planning.

In Fig. 29, we see how the **phrase** boundary tag prevents changes in the falling tone from affecting the region before the **phrase** tag. The phrase boundary allows the section from 0 to 0.42 s to be controlled exclusively by the first tag. Without the **phrase** tag, the entire curve would depend on the shape and size of the falling tone. Fig. 19 shows a contrasting example where there is no **phrase** tag, thus the effects of the second tone are allowed to reach well backwards.

[8] Recall that Stem-ML also explicitly limits look ahead pre-planning to a single phrase, so setting *adroop* = 0 is usually little different from e.g., *adroop* = 0.3.

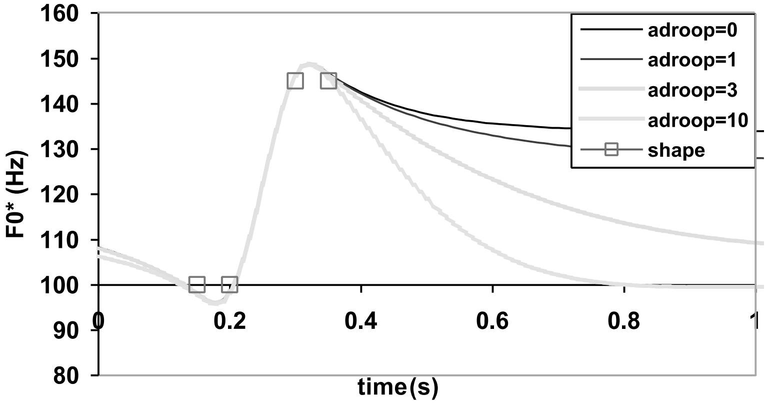

Fig. 28. Effect of the *adroop* tag. Here, the pitch curve is a constant 100 Hz. The squares show the accent's defined shape. `<set adroop=various/> <set smooth=".08"/> <step to="0" strength="3"/>···<stress shape="−.1s0,−.05s0, .05s.3,.1s.3" strength="3" type=".5"/>`.

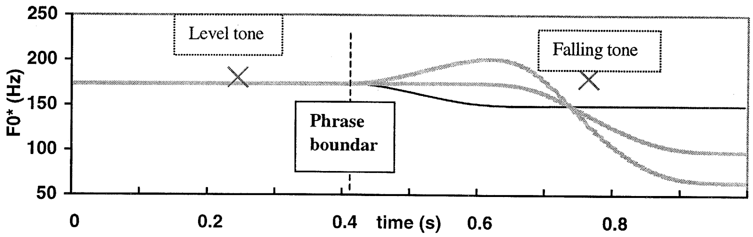

Fig. 29. Effect of a **phrase** tag. The **phrase** tag acts as a one-way wall, allowing tags before it to affect the future, but preventing future tags from affecting the past. This figure shows a level tone, a phrase boundary, followed by a tone of varying amplitude. The region before 0.42 s is completely unaffected by changes in the falling tone. `<stress strength="4" type="0.8" shape="−0.1s0.3,0.1s0.3"/> ···<phrase>···<stress strength="4" type="0.1" shape=various/>`.

4.9. Jitter and jittercut: random variation

People will not say the same sentence identically in separate trials. From a TTS point of view, the jitter and jittercut tags can be used to introduce some random variation into the pitch trajectory, so that repeated phrases will not sound mechanically identical. The random pitch curves are $1/f$ noise, with a high frequency cutoff set by the glottal musculature (i.e., the value of the *smooth* parameter is used), and a low frequency cutoff set by the *jittercut* parameter. Setting *jittercut* to the mean word length will give you random accents inside of words, but little variation on the scale of a phrase. On the other hand, setting *jittercut* to the phrase length will give you a random phrase curve, with relatively little variation inside words (see Fig. 30).

4.10. The add attribute

The most noticeable effect of the *add* setting is that it controls how the f_0 excursion of an accent changes, depending on the phrase curve. For small $add < 1$, a given **stress** tag will make a larger f_0 change if it rides on top of a high area of the phrase curve than in a low region. For $add = 1$, the size of an accent (measured as f_0, not perceptually), is independent of the value of the phrase curve.

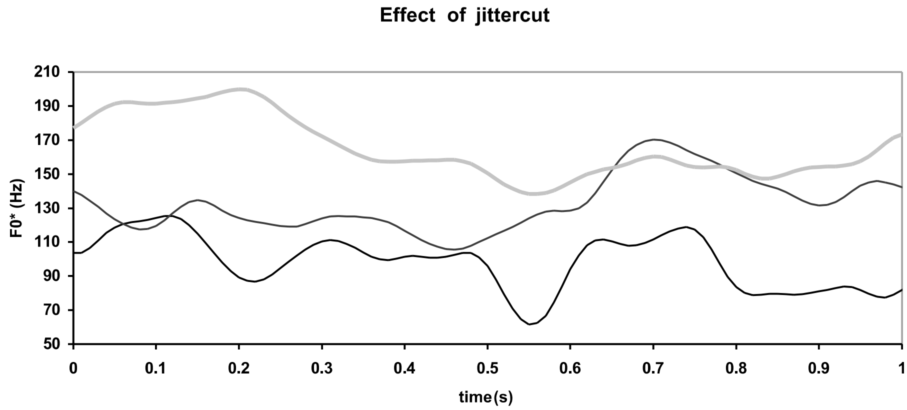

Fig. 30. Random pitch trajectories from *jittercut* = 0.1 s, 0.3 s, 1 s (from bottom to top). The curves are vertically shifted for display clarity. `<set jitter="0.1" jittercut=various/>`.

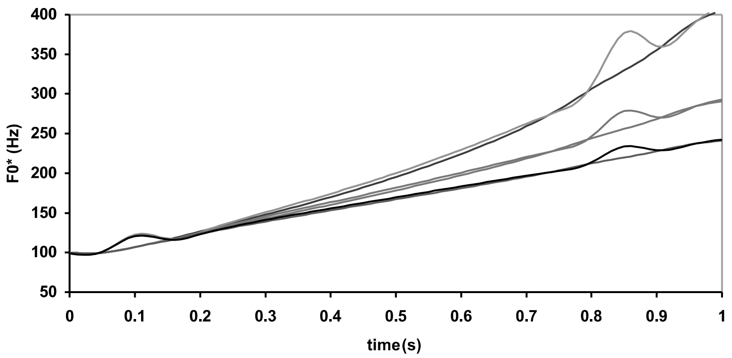

Fig. 31. Pitch trajectories with different values of *add* = 0 (top), 0.5 (middle), 1 (bottom). We show each value both with and without a pair of identical **stress** tags. `<set add=various/>···<slope rate="1"/>, with or without a pair of <stress strength="3" type="0.5" shape="−0.1s0,0.05s0,0s0.1,0.05s0,0.1s0"/> tags`.

This effect can be seen in Fig. 31, which shows three pairs of pitch trajectories, with different values of the *add* parameter. Each pair displays the effect of identical accents: one member of the pair has the accents on top of a phrase curve, the other member just shows the phrase curve. The top pair assumes $add = 0$, to give a logarithmic relationship between frequency and perceived pitch: when we command the system to provide a uniform slope in pitch, the frequency increases faster than linearly. As a consequence, small accents that ride on top of a high phrase curve give larger frequency excursions. The bottom pair assumes $add = 1$, so that $f(x) = x$, and the frequency increases linearly. In this case, the size of the accents is independent of their position on the phrase curve.

We can see how the *add* attribute can describe what is important in speech communication by showing three examples:

First, if perceptual effects are most important, and one's model of pitch generation assumes that the speaker adjusts accent sizes so that they sound "equivalent", it may be appropriate to compare a pitch change to the smallest detectable frequency change (DL). [9] These DL values increase with frequency, and

[9] This is the frequency difference limen (DL), loosely called the "just noticeable difference" (JND). It is measured by comparing the pitch of pairs of tone bursts. See Moore (1989, pp. 158ff).

Wier et al. (1977) have fit their frequency dependence as DL $\propto e^{(f^{1/2})}$, where f is the pitch. In our model here, such a dependence corresponds to some relationship between accent strength and frequency that is intermediate between linear and exponential, roughly, $add = 0.5$.

As a second example, if the speaker does not adapt him/herself for the listener's convenience, one could get values of $add > 1$. For instance, if muscle tensions are assumed to add, $f_0 \approx \text{tension}^{1/2}$ and $add \approx 2$.

As a final example, Fujisaki has used a logarithmic scale for f_0 contours, based on a model where muscle extensions are specified by neural control signals, combined with a vocal fold stiffness that increases exponentially with extension. Such behavior corresponds to $add = 0$ in our model.

5. Using Stem-ML to build a model of intonation

Stem-ML is designed to be flexible and theory neutral. A consequence of this design is that there are very few inherent constraints that restrict the usage and the combination of Stem-ML tags. The same pitch contour can often be approximated many different ways, using different sets of tags, some of which may well be linguistically unreasonable.

Stem-ML can be theory neutral because it is an over-complete representation of f_0. Because there are many ways to use Stem-ML to represent a given pitch curve, many different theories of prosody can be mapped onto Stem-ML. This means that one must define a language-specific layer on top of Stem-ML. For instance, one must decide whether or not to use a phrase curve, and decide whether accents are best associated with words or syllables, among other choices. If one does not restrict Stem-ML's flexibility, there

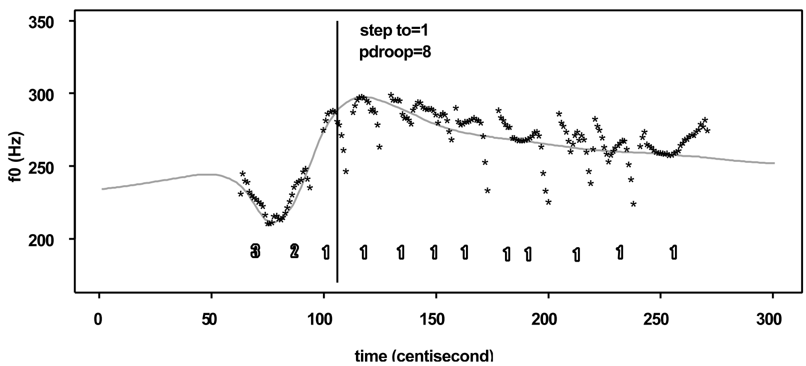

Fig. 32. Alternative tag set for Fig. 15. The large value of *pdroop* suggests a steep decline. The usage is problematic since the data clearly suggests a gradual declination slope.

```
Global parameters:
tag=set; add=0.5; smooth=0.06; base=225;  range=180; adroop=2; pdroop=8;

Prosodic code:

Lao3 wang2 jin1 tian1 gang1 gang1 bang1 zhong1 yi1 zheng1 dong1 gua1
"Lao-Wang just help the doctor to steam winter melon today."
```

$\updownarrow_{0.5}$ $\updownarrow_{1.0}$ ┤

$3_{0.5}$ $2_{0.5}$ $1_{0.5}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$ $1_{0.2}$

will be many equivalently good representations of any given utterance, and further analysis may become impractical.

5.1. Multiple interpretations of data

One must be careful if one uses automated methods to learn Stem-ML tags. To illustrate the potential pitfalls, we show in Fig. 32 how one of the earlier examples (Fig. 15) can be accounted for by a totally different combination of Stem-ML tags with a *pdroop* value of 8 s^{-1}, which could suggest a very steep declination rate. To avoid venturing too far into the wrong track, any model building has to be constrained to be consistent across a reasonable variety of data. Lessons learned from controlled experiments may help us to find the right model, especially if one can link parameter variations to experimental conditions. Evaluating results on a separate set of data helps to avoid over-fitting problems.

5.2. Language-specific constraints on Stem-ML

An example of a set of language-specific set of constraints on Stem-ML which was successfully used in automatic fitting of Mandarin (Kochanski et al., 2001; Kochanski and Shih, 2001), we use the following rules:

- Just five templates (tones 1–4 and a neutral tone) generate all surface tone shapes. The templates are stretched (in time) and scaled (in pitch) for each syllable.
- Pitch scaling of templates and Stem-ML *strength* are controlled by the same parameter. Thus, we assume that as syllables become stronger they are both articulated more carefully and expressed with a wider pitch range.
- Syllable strengths are derived from a word-strength and a metrical pattern for the word. Words with the same number of syllables share the same metrical pattern.
- Stem-ML **phrase** tags were placed at each pause of 150 ms or more.
- Phrase curves are straight line and shared.
- All utterances share the same Stem-ML *smooth*, *range*, and *base* parameters.

If Stem-ML is to be used for human labeling of speech, one must create labeling standards equivalent to the ToBI annotation rules (Beckman and Ayers, 1997). The standards must specify what tags (or combination of tags) can be used in what circumstances. If these standards are designed properly, they can eliminate ambiguity without seriously compromising Stem-ML's ability to represent the pitch contour. These rules or standards then become part of the complete language model that connects linguistic annotations to acoustic data.

5.3. Example of building a language model

As a concrete example of how one might model a language, we will describe a simple model of a small corpus of Mandarin Chinese words, similar to that described in (Kochanski and Shih, 2000).

The first step in building the language model is deciding how to represent the relevant linguistic features. In this case, there are relatively few options: Mandarin is known to be a tonal language, with tones associated with syllables. We choose to model tones with **stress** tags, associating one per syllable. There are four classes of **stress** tags, one for each tone.

In order to keep the model as simple as possible, we will assume that each **stress** tag is generated by stretching a corresponding template so the length of the template is proportional to the length of the syllable. [10] The assumption of four tone templates is crucial, as it allows a very compact representation of the language, since the tone shapes only have to be specified once, not once for each syllable. Tag stretching is defined by two parameters per tone class, one for the length of the tag relative to the syllable and one for an offset between the syllable center and the template center. The shape of the template is defined by five parameters per tone class. A more detailed description of shape seems unnecessary, based on an inspection of the data. We also allocate two parameters per tone class to scale and shift (in pitch) tone templates as a function of strength.

We put free parameters on the *add*, *smooth*, *base*, *adroop* and *pdroop* settings, for a total of five parameters. These are constant across all utterances, and characterize things like the speaker's mean f_0, typical declination rate and muscle response time.

In this example, we allow each utterance to have its own straight-line phrase curve, accounting for two parameters per utterance. The phrase curves are implemented with **step to** and **slope** tags. These phrase curves were intended to capture any systematic declination in the pitch.

Finally, each syllable has a parameter that sets the strength of the associated **stress** tag. In a larger database, these *strength* parameters would be the most numerous parameters, and also the most important, because they would be the only ones which could capture local prosodic effects. In this small database, the situation is less clear cut because there are about as many parameters that define the tone shapes (44) as parameters that set the strength of individual tones (38).

The data was obtained from a female native Mandarin speaker. [11] Utterances were isolated one and two syllable words, spoken in a laboratory setting. We estimated f_0 with the `get_f0` program of ESPS/Waves (Talkin and Lin, 1996), and manually checked for voicing errors and locations where f_0 might be estimated incorrectly. Next, we fit the model to the data by varying the model's parameters to minimize the RMS error between the data and the model, evaluated over voiced regions. We used the optimizer that was used in (Tyson et al., 1998).

In unvoiced regions, the data do not constrain f_0^*. This lack of glottal oscillation does not imply that $f_0^* = 0$, it merely means that the amplitude of oscillation is zero. Specifically, the vocal folds can be tensed and ready to vibrate, even in unvoiced regions. Unvoiced regions can be generated without changing vocal fold tension, by reducing the subglottal pressure, by pressing the folds together, by holding them wide apart, or by closing the upper vocal tract. When we fit models to data, we constrain the models only with the voiced regions, leaving f_0^* in the unvoiced regions free.

The resulting fit is shown in Fig. 33. The entire corpus is shown. The 3–3 combination is absent due to tone sandhi (Shih, 1986).

A discussion of the resulting parameter values is not really valuable, since the database is so small. Instead, we refer readers to Kochanski et al. (2001) and Kochanski and Shih (2001) for a detailed analysis of a larger corpus. However, we will note two effects that are characteristic of Stem-ML models:

- The average pitch of tones depends on their context. This occurs because the tones need to maintain their shape (at least approximately), and because they need to make smooth connections to their neighbors (because of muscle physiology). This effect can be seen in the average height of tone 4, especially comparing the isolated tone to the (4,1) pair. Likewise, tone 1 gets pushed down when preceded by a tone 4.
- Coarticulation effects can substantially distort tone shapes. Note, for instance, the compression of the pitch range of tone 2 in the (4,2) pair relative to isolated examples. Similarly, the "high-level" tone 1 can become significantly tilted in the (4,1) or (2,1) pairs.

[10] The template is stretched to cover the entire syllable, including unvoiced consonants.
[11] C. Shih, one of the authors. Data was recorded in 1997, well in advance of any work on this model.

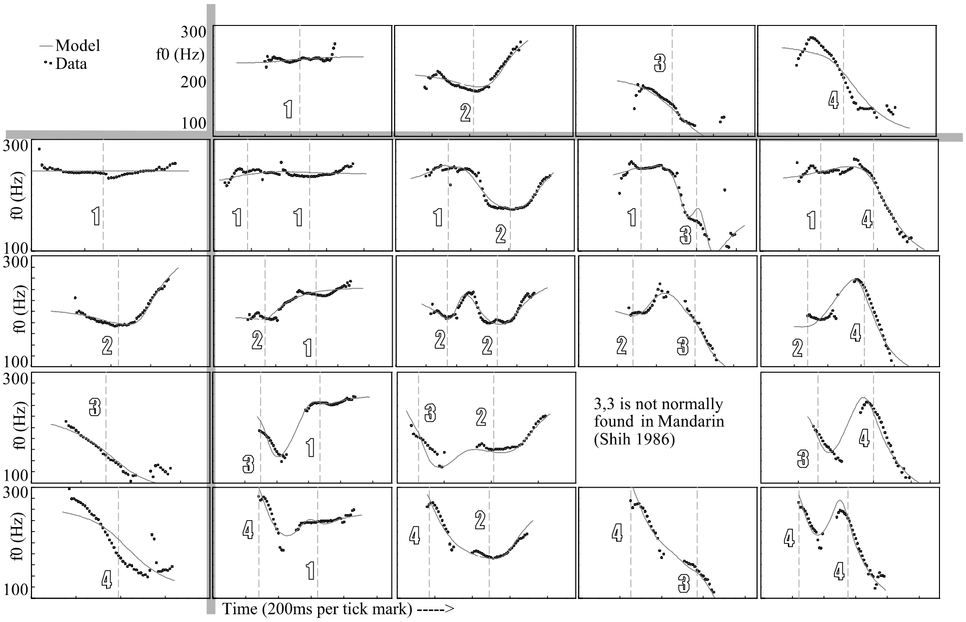

Fig. 33. Data vs. Stem-ML model for a small Mandarin corpus. Syllable centers are marked with vertical dashed lines, and the numbers in outline font identify the tones. The top row and leftmost column show isolated single syllables, while the remainder of the figure shows two syllable utterances. The modeled f_0 curves are all derived from the same four Stem-ML templates. Note that the model captures much of the coarticulation between tones: see for instance the change in tone 4's mean f_0 from an isolated tone to the 4,1 tone pair.

Acknowledgements

We thank two anonymous reviewers for extensive comments, and Cindy Pribble for editorial assistance.

Appendix A. Tag definitions

We will specify tags in XML format here. In this description, literal strings are quoted, then (following regular expression notation), '?' marks optional tokens, '*' marks zero or more occurrences of a token, and '+' marks one or more occurrences. Options are shown with '|', and parentheses and newlines are used for grouping. Tags are defined in the XML namespace http://prosody.multimedia.bell-labs.com. See http://www.w3.org/XML for information describing XML, including namespaces. Other information on Stem-ML may be found at http://prosody.multimedia.bell-labs.com.

A.1. Tag grammar

```
Tag = "<" tagname AttValue* "/>"
```
Example:
```
<set base = "200"/>
  # Set base frequency to 200 Hz.
```
Each tag is composed of two parts: a tag name, and a set of attribute-value pairs that control the details of what happens. All of the tags are 'point' tags, which are self-closing. We implement Stem-ML with point

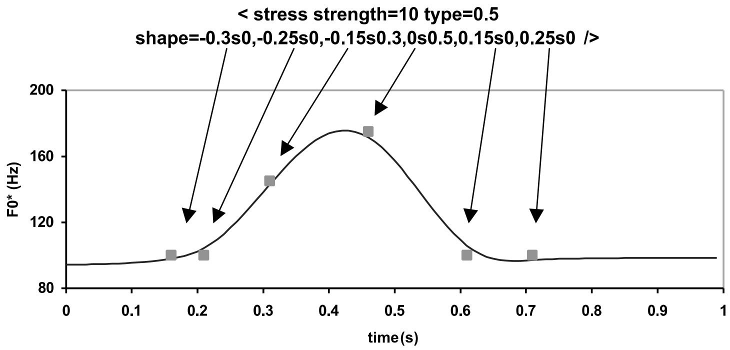

Fig. 34. Sample **stress** tag and resulting pitch trajectory.

tags to allow it to mix better with other mark-up information. Non-self-closing tags must be properly nested in XML, and it is not obvious that prosodic markup would nest well with syntactic or semantic markup.

`Tagname = "set" | "step" | "slope" | "stress" | "phrase"`

Lists of legal attributes can be found in Sections 3.1–3.5.

The *shape* attribute of the **stress** tag has a fairly complex syntax. You specify the shape of a template as a set of (time, pitch) points.

`Shape = shape_from_points,`

`Shape_from_points = (point ",")* point`

A *point* in the shape argument of the **stress** tag follows the syntax:

`point = float ( "s" | "m" | "p" | "y" | "w" ) value.`

It specifies a point on the accent curve as a (time, frequency) pair (frequency is expressed as a fraction of the speaker's range). Time is measured in seconds (s), milliseconds (m), phonemes (p), syllables (y), or words (w). One does not need to specify the accent curves too finely, as the resulting pitch curve will be smooth. Fig. 34 shows an example.

Stem-ML doesn't restrict itself to predicting f_0^*. Many values can be vector quantities, with components corresponding to amplitude, glottalization, face motions, or whatnot.

`value = float | ( float letter )+`

`mvalue = float | ( float letter letter )+`

The letter in a value tells you what component of prosody it is associated with, if you are controlling more than one component of prosody (e.g., f_0^* and eyebrow position). The two letters in an `mvalue` correspond to two indices in a matrix mapping from perceptual parameters (e.g., 'emphasis') to observable output values (e.g., 'f_0^*' or 'subglottal pressure') (see Section 2.4). A value or mvalue can be a single `float`, for a simple system that predicts one-component prosody, like pitch.

A.2. Tag grammar: motions

In most TTS implementations, the binary equivalent of Stem-ML tags are inserted, in the appropriate places, into a memory structure that describes the utterance. The tags are built and inserted by the linguistic modeling component of the TTS system, based on lexical properties and syntactic information. However, if Stem-ML is used on a serial data stream, it is convenient to place tags between words, and shift the accents into the correct position. Stem-ML allows that with the *move* attribute, which is legal as part of all tags.

`AttValue = position | other_attributes`

`position = "move" "|" motion+`

```
motion = (float | "b" | "c" | "e") ("r" | "w" | "y" | "p" | "m" | "s") | "*"
```

The system evaluates motions in a left-to-right order. The position is modeled as a cursor that starts at the beginning of the first phoneme following the tag. [12] You can specify motions in units of phrases (r), words (w), syllables (y), phonemes (p), milliseconds (m), or seconds (s). Phrases and words can be useful units if the tags are congregated at the beginnings of phrases.

- Motions specified in phrases skip over any pauses between phrases.
- Motions specified in words skip over any pauses between words.
- Moves specified in syllables treat a pause as 1 syllable.
- Motions specified in phonemes treat a pause as 1 phoneme.
- Using a 'b', 'c' or 'e' as a motion will move the cursor to the nearest beginning, center or end of a phrase, word, syllable or phoneme. The notation `move = "er"` is a convenient way to place a tag at the end of a phrase (e.g., for a boundary tone).
- Moves specified in seconds just move the cursor that number of seconds.
- The motion "*" (stressed) moves to the center of the next stressed syllable.
- If two tags are moved to the same position, the tags are evaluated in order of their appearance in the input text.

Negative moves are allowed, but the cursor cannot be moved out of the phrase. [13]

Example:

```
<step move = "*0.5y" by = "1"/>
  # Put a step in the pitch curve, with the steepest part of the step 0.5 syllable
  after the center of the first stressed syllable after the tag.
```

References

Abry, C., Badin, P., Boë, L.-J., Perrier, P., Schwartz, J.-L., 1998. Les Cahiers de l'I.C.P. Bulletin de la Communication Parlée, No. 4. Université Stendhal, Grenoble, France.

Anderson, M., Pierrehumbert, J., Liberman, M., 1984. Synthesis by rule of English intonation patterns. In: Proceedings of ICASSP 1, San Diego, CA, pp. 2.8.1–2.8.4.

Atkinson, J.E., 1978. Correlation analysis of the physiological factors controlling fundamental voice frequency. J. Acoust. Soc. Amer. 63, 211–222.

Beckman, M.E., Ayers, G., 1997. Guidelines for ToBI labeling (version 3, March 1997). Available from <http://www.ling.ohio-state.edu/phonetics/ToBI/ToBI.0.html>.

Berry, D.A., Herzel, H., Titze, I.R., Story, B.H., 1996. Bifurcations in excised Larynx Experiments. J. Voice 10, 129–138.

Black, A.W., Hunt, A.J., 1996. Generating F_0 contours from ToBI labels using linear regression. In: Proceedings of ICSLP 96, Philadelphia, PA, USA.

Bolinger, D.L., 1958. A theory of pitch accent in English. Word 14, 109–149.

Browman, C.P., Goldstein, L., 1990. Tiers in articulatory phonology with some implications for casual speech. In: Kingston, J., Beckman, M. (Eds.), Papers in Laboratory Phonology I: Between the Grammar and Physics of Speech. Cambridge University Press, Cambridge, pp. 341–376.

Chen, S.H., Hwang, S.H., Tsai, C.Y., 1992. A first study on neural net based generation of prosodic and spectral information for Mandarin text-to-speech. In: Proceedings of IEEE ICASSP, vol. 2, pp. 45–48.

de Pijper, J.R., 1983. Modelling British English Intonation. Foris Publications, Dordrecht-Holland, ISBN 90-6765-004-8.

[12] Silence doesn't count.

[13] The Bell Labs TTS system will actually allow you 100 ms of leeway outside a phrase. By definition, this 100 ms leeway also corresponds to 1 phoneme, 1 syllable, 1 word or 0.1 phrases.

Dusterhoff, K.E., Black, A.W., Taylor, P., 1999. Using decision trees within the Tilt intonation model to predict f_0 contours. In: Proceedings of the Sixth European Conference on Speech Communication and Technology (Eurospeech '99), 5–9 September 1999, Budapest, Hungary.
Erickson, D., 1998. Effects of contrastive emphasis on jaw opening. Phonetica 55, 147–169.
Fry, D.B., 1955. Duration and intensity as physical correlates of linguistic stress. J. Acoust. Soc. Amer. 30, 765–769.
Fry, D.B., 1958. Experiments in the perception of stress. Language and Speech 1, 126–152.
Fujimura, O., 2000. The C/D model and prosodic control of articulatory behavior. Phonetica 57, 128–138.
Fujisaki, H., 1983. Dynamic characteristics of voice fundamental frequency in speech and singing. In: MacNeilage, P.F. (Ed.), The Production of Speech. Springer-Verlag, Berlin, pp. 39–55.
Fujisaki, H., 1988. A note on the physiological and physical basis for the phrase and accent components in the voice fundamental frequency contour. In: Fujimura, O. (Ed.), Vocal Fold Physiology: Voice Production, Mechanisms and Functions. Raven, New York, NY, pp. 347–355.
Gårding, E., Fujimura, O., Hirose, H., 1970. Laryngeal control of Swedish word tones. Ann. Bull. Res. Inst. Logoped. Phoniatr. 27, 135–149.
Grønnum, N., 1992. The Groundworks of Danish Intonation – An Introduction. Museum Tusculanum Press, Copenhagen.
Hadding-Koch, K., 1961. Acoustico-Phonetic Studies in the Intonation of Southern Swedish. C.W.K. Gleerup, Lund, Sweden.
Haggard, M., Ambler, S., Callow, M., 1970. Pitch as a voicing cue. J. Acoust. Soc. Amer. 47, 613–617.
Herzel, H., 1995. Non-linear dynamics of voiced speech. In: Awrejcewicz, J. (Ed.), Nonlinear Dynamics: New Theoretical and Applied Results. Akademie Verlag, Berlin.
Hillenbrand, J.M., Houde, R.A., 1996. Role of F_0 of and amplitude in the perception of intervocalic glottal stops. J. Speech Hear. Res. 39, 1182–1190.
Hirst, D.J., Di Cristo, A., Espesser, R., 2000. Levels of representation and levels of analysis for the description of intonation systems. In: Horne, M. (Ed.), Prosody: Theory and Experiment Studies. Kluwer Academic Publishers, Dordrecht (Studies Presented to Gösta Bruce).
Hombert, J.-M., 1978. Consonant types, vowel quality and tone. In: Fromkin, C. (Ed.), Tone: A Linguistic Survey. Academic Press, New York.
Kehoe, M., Stoel-Gammon, C., Buder, E.H., 1995. Acoustic correlates of stress in young children's speech. J. Speech Hear. Res. 38, 338–350.
Keating, P.A., 1990. The window model of coarticulation: articulatory evidence. In: Kingston, Beckman, (Eds.), Papers in Laboratory Phonology I. Between the Grammar and Physics of Speech. Cambridge University Press, Cambridge, pp. 451–470.
Kochanski, G.P., Shih, C., 2000. Stem-ML: language-independent prosody description. In: Proceedings of the International Conference on Spoken Language Processing 2000, Beijing, China.
Kochanski, G.P., Shih, C., 2001. Automatic modelling of Chinese intonation in continuous speech. In: Eurospeech, Aalborg, Denmark, pp. 911–914.
Kochanski, G.P., Shih, C., Jing, H., 2001. Hierarchical structure and word strength prediction in Mandarin prosody. In: 4th ISCA Tutorial and Research Workshop on Speech Synthesis, Perthshire, Scotland.
Ladd, D.R., 1996. Intonational Phonology. Cambridge University Press, Cambridge, ISBN 0521-47575-9.
Ladefoged, P., 1962. Subglottal activity during speech. In: Proceedings of the 4th International Congress of Phonetic Sciences, pp. 247–265.
Levitt, H., Rabiner, L.R., 1970. Analysis of fundamental frequency coutours in speech. J. Acoust. Soc. Amer. 49 (2, part 2), 570.
Liberman, M.Y., Pierrehumbert, J.B., 1984. Intonational invariance under changes in pitch range and length. In: Aronoff, M., Oehrle, R. (Eds.), Language Sound Structure. MIT Press, Cambridge, MA, pp. 157–233.
Lieberman, P., 1960. Some acoustic correlates of word stress in American-English. J. Acoust. Soc. Amer. 32, 451–454.
Lieberman, P., Knudson, R., Mead, J., 1969. Determination of the rate of change of F_0 with respect to subglottal air pressure during sustained phonation. J. Acoust. Soc. Amer. 45, 1537–1543.
Löfqvist, A., Baer, T., McGarr, N.S., Story, R.S., 1989. The cricothyroid muscle in voicing control. J. Acoust. Soc. Amer. 85, 1314–1321.
Maekawa, K., 1998. Phonetic and phonological characteristics of paralinguistic information in spoken Japanese. In: Proceedings of the International Conference on Spoken Language Processing 98, Sydney, Australia, paper 997.
Malfrère, F., Dutoit, T., Mertens, P., 1998. Fully automatic prosody generator for text-to-speech. In: Proceedings of the International Conference on Spoken Language Processing 98, Sydney, Australia, paper 355.
Massaro, D.W., Cohen, M.M., 1976. The contribution of fundamental frequency and voice onset time to the /zi/–/si/ distinction. J. Acoust. Soc. Amer. 60, 704–717.
McFarland, D.H., Smith, A., 1992. Effects of vocal task and respiratory phase on prephonatory chest wall movements. J. Speech Hear. Res. 35 (5), 971–982.
Monsen, R.B., Engebretson, A.M., Vemula, N.R., 1978. Indirect assessment of the contribution of subglottal air pressure and vocal fold tension to changes in the fundamental frequency in English. J. Acoust. Soc. Amer. 64 (1), 65–80.

Moon, F.C., 1987. Chaotic Vibrations: An Introduction for Applied Scientists and Engineers. Wiley-Interscience, New York.
Moon, S.-J., Lindblom, B., 1994. Interaction between duration, context, and speaking style in English stressed vowels. J. Acoust. Soc. Amer. 90, 40–55.
Moore, B.C.J., 1989. An Introduction to the Psychology of Hearing. Academic Press, New York.
Munhall, K., Löfqvist, A., 1992. Gestural aggregation in speech: laryngeal gestures. J. Phonetics 20, 111–126.
Ogorzałek, M.J., 1997. Chaos and Complexity in Nonlinear Electronic Circuits. World Scientific, Singapore.
Ohala, J.J., 1992. The segment, primitive or derived. In: Docherty, G.J., Ladd, D.R. (Eds.), Papers in Laboratory Phonology II: Gesture, Segment, Prosody. Cambridge University Press, Cambridge, pp. 166–183, ISBN 0-521-40127-5.
Ohala, J., Hirano, M., 1967. Studies of pitch change in speech. UCLA, Working papers on phonetics, pp. 80–84.
Ohala, J., Ladefoged, P., 1970. Further investigation of pitch regulation in speech. UCLA, Working Papers on Phonetics, 14, pp. 12–24.
Öhman, S., 1967. Word and Sentence Intonation, A Quantitative Model. Department of Speech Communication, Royal Institute of Technology (KTH)., pp. 20–54.
Olive, J.P., 1975. Fundamental frequency rules for the synthesis of simple declarative English sentences. J. Acoust. Soc. Amer. 57, 476–482.
Perrier, P., Ostry, D.J., Laboissière, R., 1996. The equilibrium point hypothesis and its application to speech motor control. J. Speech Hear. Res. 39, 365–378.
Pierrehumbert, J., 1980. The phonology and phonetics of English intonation. Ph.D. dissertation, MIT.
Pierrehumbert, J.B., Beckman, M.E., 1988. Japanese Tone Structure. The MIT Press, Cambridge, MA.
Pipes, L.A., 1970. Applied Mechanics for Engineers and Physicists, third ed. McGraw Hill, New York.
Plato, 366 BCE. The Republic. Book 7.514–7.521, Athens.
Pollock, K.E., Brammer, D.M., Hageman, C.F., 1990. An acoustic analysis of young children's productions of word stress. J. Phonetics 21, 183–203.
Ross, K.N., Ostendorf, M., 1999. A dynamical system model for generating fundamental frequency for speech synthesis. IEEE Trans. Speech Audio Proces. 7 (3), 295–309.
Shih, C., 1986. The prosodic domain of tone sandhi in Chinese. Ph.D. dissertation, University of California, San Diego.
Shih, C., 2000. A declination model of Mandarin Chinese. In: Botinis, A. (Ed.), Intonation: Analysis Modeling and Technology. Kluwer Academic Publishers, Dordrecht, pp. 243–268.
Shih, C., Kochanski, G.P., 2000. Chinese tone modeling with Stem-ML. In: Proceedings of the International Conference on Spoken Language Processing 2000, Beijing, China.
Shih, C., Sproat, R., 1992. Variations of the Mandarin rising tone. In: Proceedings of the IRCS Workshop on Prosody in Natural Speech, Technical Report IRCS, University of Pennsylvania, pp. 193–200.
Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., Hirschberg, J., 1992. ToBI: A standard for labeling English prosody. In: Proceedings of the International Conference on Spoken Language Processing 92, vol. 2, pp. 867–870.
Simada, Z.B., Hirose, H., 1978. Physiological correlates of Japanese accent patterns. Annu. Bull. Res. Inst. Logoped. Phoniatr. 5, 41–49.
Sluijter, A.M.C., van Heuven, V.J., 1996. Spectral balance as an acoustic correlate of linguistic stress. J. Acoust. Soc. Amer. 100 (4), 2471–2485.
Sluijter, A.M.C., van Heuven, V.J., Pacilly, J.J.A., 1997. Spectral balance as a cue in the perception of linguistic stress. J. Acoust. Soc. Amer. 101 (1), 503–513.
Sproat, R. (Ed.), 1998. Multilingual Text-to-Speech Synthesis: The Bell Labs Approach. Kluwer Academic Publishers, Dordrecht.
Sproat, R., Hunt, A., Ostendorf, M., Taylor, P., Black, A., Lenzo, K., 1998. SABLE: A standard for TTS markup. In: Proceedings of the International Conference on Spoken Language Processing 98, Sydney, Australia, pp. 1719–1724.
Stevens, K., 1998. Acoustic Phonetics. The MIT Press, Cambridge, MA, ISBN 0-262-19404-X.
Talkin, D., Lin, D., 1996. Get f_0 online documentation, ESPS/Waves release 5.31. Entropic Research Laboratory, 1996. (Entropics was purchased by Microsoft in 2000.) Algorithm based on Talkin, D., A robust algorithm for pitch tracking. In: Kleijn, W.B., Paliwal, K.K. (Eds.), Speech Coding and Synthesis. Elsevier, New York.
Taylor, P.A., 1998. The Tilt intonation model. In: Proceedings of the International Conference on Spoken Language Processing 98, Sydney, Australia, paper 827.
Taylor, P.A., 2000. Analysis and synthesis of intonation using the Tilt model. J. Acoust. Soc. Amer. 107 (3), 1697–1714.
Taylor, P., Isard, A., 1997. SSML: A speech synthesis markup language. Speech Communication 21, 123–133.
Titze, I.R., 1988. The physics of small amplitude oscillation of the vocal folds. J. Acoust. Soc. Amer. 83 (4), 1536–1552.
Titze, I.R., 1989. On the relation between subglottal pressure and fundamental frequency in phonation. J. Acoust. Soc. Amer. 85 (2), 901–906.
Titze, I.R., 1993a. Principles of Voice Production. Prentice-Hall, Englewood Cliffs, NJ, pp. 36–42, 91–102, 192–213, ISBN 0-13-717893-X.

Titze, I.R., 1993b. Principles of Voice Production. Prentice-Hall, Englewood Cliffs, NJ, p. 205 (figure 8.7), p. 208 (figure 8.9), p. 210 (figure 8.11), p. 212 (figure 8.12).

Turk, A.E., Sawusch, J.R., 1996. The processing of duration and intensity cues to prominence. J. Acoust. Soc. Amer. 99 (6), 3782–3790.

Tyson, J.A., Kochanski, G., Dell'Antonio, I., 1998. A detailed mass map of CL0024+1654 from strong lensing. Astrophys. J. Lett. 498 (2), 107.

van Santen, J.P.H., Möbius, B., 1997. Modeling pitch accent curves. In: Intonation: Theory, Models, and Applications. Proceedings of ESCA Workshop, Athens, Greece, pp. 321–324.

van Santen, J.P.H., Möbius, B., 2000. A quantitative model of F_0 generation and alignment. In: Botinis, A. (Ed.), Intonation: Analysis Modeling and Technology. Kluwer Academic Publishers, Dordrecht.

van Santen, J.P.H., Shih, C., Möbius, B., 1998. Intonation. In: Sproat, R. (Ed.), Multilingual Text-to- Speech Synthesis: The Bell Labs Approach. Kluwer Academic Publishers, Dordrecht, pp. 141–190.

Whalen, D.H., Kinsella-Shaw, J.M., 1997. Exploring the relationship of inspiration duration to utterance duration. Phonetica 54, 138–152.

Wier, C.C., Jesteadt, W., Green, D.M., 1977. Frequency discrimination as a function of frequency and sensation level. J. Acoust. Soc. Amer. 61, 178–184.

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 Hear. Res. 37 (3), 535–556.

Winkworth, A.L., Davis, P.J., Adams, R.D., Ellis, E., 1995. Breathing patterns during spontaneous speech. J. Speech Hear. Res. 38 (1), 124–144.

Xu, C.X., Xu, Y., Luo, L.S., 1999. A pitch target approximation model for f_0 contours in Mandarin. In: Proceedings of the 14th International Congress of Phonetic Sciences, San Francisco, pp. 2359–2362.

Xu, Y., 1993. Contextual tonal variation in Mandarin Chinese. Ph.D. dissertation. The University of Connecticut.

Xu, Y., Sun, X.J., 2000. How fast can we really change pitch? Maximum speed of pitch change revisited. In: Proceedings of the International Conference on Spoken Language Processing 2000, Beijing, China.