Text to Speech Synthesis Using Syllables as Functional Units of Speech

Reiner Wilhelms-Tricario*

SUMMARY: This is the description of an idea to build a syllable based speech synthesizer as part of a text to speech (TTS) system. The speech synthesizer is designed to emulate speech production with methods that are informed by what is known about the natural speech production processes, while aiming for high quality speech output. The goal is to build a detailed modular and partially hierarchical time-varying dynamic system that is controlled at the top by discrete multidimensional processes in an abstract space of feature variables. While not further investigating this, it is assumed that these can be derived from the input text. The features used to describe speech are closely related to speech articulation, rhythm and pausing, as well as other identifiable prosodic parameters. Together with phrase/rhythm and prosodic features, the syllable is used as atomic speech unit, and syllable features play a central role in the control of the synthesizer. For this, Fujimura’s C/D model provides a large part of the framework. The system is designed as a generative model of observable speech production processes that makes it possible to use system identification procedures, analysis-by-synthesis methods, and methods of machine learning. This data driven approach will be necessary in order to obtain the large number of parameters that are deemed necessary to specify with some accuracy the properties of observed speech produced by one or more speakers, so that the model can generalize to produce high quality speech from arbitrary text.

Key words: system identification, C/D model, syllables, Bayesian inference, Cubature Kalman Filter, RNN

1. Introduction

For at least more than a decade now it appears that the original dream of building truly natural speech synthesizers has completely faded away. Before that it was more or less seen as a feasible idea that one day one would be able to build “natural speech synthesis” based on exact computational models of the aerodynamics and acoustics of the vocal tract, by emulating the salient physiological properties of the articulators and human speech motor control in computational systems, and perhaps even combine these with models of cognition to synthesize speech. Nowadays, the best widely used speech synthesizers and text to speech systems (TTS) produce rather naturally sounding speech while they are predominantly based on an apparently very primitive way of modeling speech production, namely by concatenating snippets of recorded speech signals. A lot of improvements to avoid concatenation artifacts by smoothing and to improve unit selection have now been implemented in leading TTS systems. Hereby one of the main impacts comes from the simplest kind of intervention, namely drastically increasing the size of the database. At least for commercial speech synthesis this method has few competitors, even though recently parametric speech synthesizers may have a comeback. Hidden Markov model (HMM) based text to speech systems (HTS) and hybrids come increasingly closer in naturalness to concatenative speech synthesizers; they often provide much more flexibility to implement prosodic models, but they are still not widely accepted for commercial TTS systems, see e.g., Zen et al. (2009).

While research in speech synthesis and recognition was once mainly in the domain of linguistics and speech science, aeroacoustics, physiology and psychology, it became overwhelmingly a domain of computer science. There are of course some very good reasons for this. In particular, just as it happened in such areas as image and pattern recognition and classification, the classical methods of designing hand knitted feature systems to represent data and objects of the domain of study has been widely replaced by automatic methods of “feature discovery,” one of the bases of unsupervised machine learning. The central paradigm here is to build easily trainable generative models, for example, based on

* Vocoli/Haskins Laboratories (ヴォカリック/ハスキノン研究所)
stacks of reduced Boltzmann machines forming a so-called deep Boltzmann machine (DBM), or based on auto-encoders, see Bengio et al. (2013), Hinton (2013), Taylor and Hinton (2009).

Common to these methods is to find low dimensional representation empirically by sifting through large amounts of training data. The representation of high-dimensional data is learned in such a way that the information has to pass through a series of nonlinear re-mapping steps, structurally similar to artificial neural networks. Added noise and random dropouts during training may be used to introduce redundancy for improving robustness, and an important strategy is to pass the latent information through a bottleneck that enforces a low-dimensional representation. The systems are trained to optimally reconstruct the original training data from such drastically reduced representations, which implicitly requires that the salient information is not lost in the compression. One way to discover features is finding them in binary encodings in the narrowest bottleneck.

It is interesting to find that using such discovered features often results in big improvements, while it may be that no one knows - nor cares about - what these features “mean.” In particular, using automatically found binary features also allows to make use of very rapid nearest neighbor search algorithms, where a general property of the data structures used in the learning process is exploited, namely that pattern similarity measures can be approximated by Hamming distance measures in a binary feature space.

All this is not to say that all other methods of information representation are now replaced by systems distilled out of “big data.” To the contrary, it has often been found that even better systems for pattern recognition, both for speech recognition and for speech synthesis, may be designed as hybrid systems, in which hand-knitted features are combined with automatically discovered features to represent data objects and patterns, for example, by combining phonological features with other features that are automatically discovered from data.

As speech technology is increasingly leaning almost exclusively on computer science, it may get easily forgotten that a lot is actually known about how humans produce speech, and there is no good reason to simply ignore that knowledge. Thus, we seek a representation of speech that is informed by what is known about the organization of the speech articulatory system and motor control of speech, while still insisting on a data driven approach to find the fine details. Building of such models and finding their parameters will have to rely on large amounts of data that not only include the speech signal but also, as much as feasible, measurements of speech articulation that can be simultaneously recorded in non-invasive ways. These articulatory data are going to be mainly from video data, see e.g., Abe et al. (2012), Wilson and Erickson (2013), Wilson et al. (2012), and if possible by means of image processing of multi-camera recordings.

1.1 How Complex is Speech?

The articulatory structures used for speech production are usually thought to be subdivided into a few – more or less tightly coupled – dynamic systems that are controlled and coordinated by an accordingly structured control system. The articulatory system may appear to have too much complexity, especially if one considers the state space of soft tissue articulators like the tongue or lips modeled as continua, or the potentially very high-dimensional state space of the control system itself, which can be seen as a vast biological neural network processing motor and sensory signals.

Building computational models of speech production processes plays an important role for investigating and understanding speech and speech motor control. In order to gain knowledge about speech production such models have to be built with great care about parsimony, so that the models can be general enough to explain common speech phenomena. It is a quite different goal, and not so much guided by parsimony and Occam’s razor, if one wants to build speech synthesizers producing high quality synthetic speech that is nearly indistinguishable from real human speech, because it requires to represent many more very speaker specific features, and it is unlikely to achieve if we insist on building only systems that attempt to faithfully represent real physiological and physical processes: Many aspects of these can not be observed or measured.

There has to be a middle way though: Instead of completely giving up to build faithful models of human speech production, as is done in the design of most commercial speech synthesis methods, it should be possible to combine models of real speech production with artificial generative models where the emphasis is on accurately representing measured data, if necessary in disrespect of the actual nature of the processes that generated the data. In short, the motto is roughly this: Build physical or physics based models where possible and validate them with observed data to build a skeletal overall model, then use “fuzzy models” for the rest, to be able to represent the fine details of individual
speaker's voices.

Complex systems, under certain conditions, may not be so complex after all. If the components of a complex physical, biological or ecological dynamic system are strongly coupled, it is often possible to find rather low-dimensional approximate state space representations, which are much simpler than what the apparent complexity and large number of components suggest.

For example, for a dynamic system like the human glottis, which may be modeled with some accuracy by finite element procedures, it turns out that the kinematics and dynamics of its movements can be well described by merely three overlapping modes of oscillation. Also, the movements of the tongue during speech production are extremely constrained by its surrounding structures, and tongue movements during speech production make use of only a small part of the overall range of motor function of the multi-muscle system.

More generally, complex systems may reveal their actual simplicity once the system is at least partially understood: For example, in ecological networks, one often finds the establishment of leading species networks (forming the part of the network with highest centrality). In that case the population dynamics of a few key species can be used to predict most other partial processes.

Furthermore, if a complex dynamic system is sufficiently constrained and tightly coupled, it is often possible to discover empirically its main degrees of freedom by analysing sufficiently large amounts of observed data from measurements about the process of the system under study, even if a physically/biologically faithful realistic model does not exist.

1.2 Speech Motor Control

Speech is the outcome of complex and skilled coordination of the multi-muscle system that controls the vocal tract structures. Like any advanced motor skill it requires years of development and training to learn the smooth and accurate execution of speech motor control. It may not be entirely clear how the information about this skill is represented in the brain, but we will adopt the hypothesis that the speech production skill is in some sense modular and makes use of stored "coordinative structures" or synergies that represent frequently found speech gestures in some way.

Merely from a computational argument it is unlikely that the complex coordination for speech articulation is computed every time from scratch. Yet, it is just as unlikely that speech movements are stored as fixed programs and executed one by one - in concatenation - to create the entire movement trajectory for a phrase. The existence of co-articulation and the great variability of how an individual speaker pronounces speech portions (syllables, words and phrases) depending on an intended prosody, speaks for a representation that has both invariant aspects (ergodicity) and a great deal of flexibility in using moment to moment control, which is postulated to be the output of the motor planning.

Thus, general ideas about "motor-sensory integration" and "coordination structures," are realized in the model as coupled partial dynamic systems, subdividing the entire system in a way that is informed by the organization of real speech articulators. Some of these partial models can be implemented as physics based models or at least inspired by the natural processes themselves. Other partial dynamic system components, where this is impossible or too hard, are implemented as recurrent neural networks, see Figure 1.

Some of the parameters in these models can change
Figure 2  An illustration of “dynamic regimes”: The slices represent a manifold at different times in some parametric phase space. The course of a particular instantaneous state of the system is drawn as the curved trajectory. The phase portraits on the manifold are illustrated by the curved arrows on the planes; they determine the local behavior of the dynamic state; and they change slowly over time.

on several time scales, roughly corresponding to phrases, words, and syllables. This is to be distinguished from the (instantaneous) state of the model, which proceeds on the fastest time scale in the model, and which is subject to the more slowly changing dynamic regimes. The output of the model is computed based on the model’s state. It describes the moment-to-moment change of articulatory and acoustic parameters that can be observed in real speech production.

Figure 2 tries to give an intuitive depiction of “dynamic regimes”: The parameters of neural networks or actual physical models define a dynamic regime that is different for each of the shown slices in some phase space, as it gradually changes over time. The instantaneous state of the model here is a point on the manifold. It moves driven by the current dynamics or flow in the phase space: It follows the local flow direction, illustrated in the figure by the curved arrows in the plane. The overall dynamics (essentially the field of directions) determine how any point moves, from its initial conditions. It is specified by the slowly changing model parameters that change from syllable to syllable and within each syllable.

Ideas sketched out above about the behavior of tightly coupled and constrained dynamics of articulatory systems lead to the need for using directly measurable articulatory data in addition to speech signals for building this model. A similar idea is actually implied in the C/D model, where it is assumed that for any elemental gesture one crucial articulator can be named, and the observations of its movements are then used to sort out the remaining details of timing of the other articulators.

Suppose a partial model and synthesizer could be built which merely - but quite accurately- predicts the movements of the jaw, ultimately from the text of the spoken utterances (this could be called a Text-to-Jaw synthesizer). The signals that it generates, jaw position and velocity, represent an observable partial process of speech production. Even though only some of the details of the acoustic properties of the speech signal are directly attributable to jaw movements, the model’s predicted jaw movement signals can be used to augment the input to an extended model, which may for instance predict lip parameters. Getting the timing of the jaw movements right is an important step to be able to make inference about the timing of the other articulators. Obviously, to realize such a model requires to find its parameters from data.

2. Model Description

Taking seriously the idea that speech is the result of the movements (in a generalized sense) of a complex dynamic process, building a speech production model means to design a computational dynamic system that is capable of reproducing time series of observable signals, which can be obtained by measurements from a human speaker.

To turn this into a text-to-speech system, the generative model, that is, the speech production model, needs to be in some way controlled by the input text that one wants to synthesize. Therefore it needs to be equipped with a control system that operates on features that describe speech and can be extracted automatically from the text to be synthesized. The additional requirement is that by design the processes in the domain of the feature description of speech have an overall correspondence with the dynamic processes in the generative model, which produce observable signals, namely speech signals and articulatory time series.

Even though we may postulate that in the natural system the processes that go on in the brain for speech production may also use some kind of feature description of speech, hardly any of this can be confirmed by observations. However, we may postulate and use a plausible abstract model for this, and that is where the C/D model comes in.
Figure 3 Model overview.

Figure 3 shows an overview of the overall design. For the synthesis of a complete phrase, words are divided into syllables, using a process that includes dictionary lookup of the syllables to obtain their feature description. Beyond syllable features, additional word-specific features may be added, such as lexical stress, and part-of-speech information, as well as other features depending on context. For each syllable, the (intrinsic) syllable features are obtained by table lookup, and in some cases modified by rule to obtain the syllable features in multisyllabic words, see also section 3.4. These features, both syllable specific and beyond the syllable level, enter the C/D model and also the box shown in Figure 3 to the left of the C/D model. This box implements a mapping from syllable features and context features to parameters that modify the dynamic behavior of the signal generator shown as a large box at the bottom. The output of the C/D model, labeled in the figure as “syllable timed actions,” are not the actual control signals but rather prototypical signals that operate on an abstract time or phase: The figure shows them as $U[s]$, whereby $s$ is a parameter that increases by 1 from the beginning of a syllable to its end. The mapping of this parameter onto real time is done in the box labeled “syllable duration & timing model,” which needs to be adapted to a particular speaker. This is illustrated in the figure by making the parameterization $t$ (for time) the output of a realtime clock.

The signal generator in the large box at the bottom, which is essentially the same as Figure 4 and further discussed in section 2.3, is driven by two inputs: the moment to moment controls signal $u(t)$, and the parameter control that changes the dynamics of the modules in the signal generator, either by changing physical meaningful parameters or by changing weights in neural networks. Within the signal generator, components of the control signal vector $u(t)$ are sent to different modules, which is indicated by the dashed arrows. Additional control signals within that box are generated by some of the modules and augment the control input of other dependent modules; these signals are the curved solid arrows.

In the implementation, wherever meaningful physically models can not be provided, the partial systems are realized more specifically by a general structure which is composed by a discrete recurrent neural network and a second neural network that is usually not recurrent, see Figure 1. These neural networks are updated every 5 ms or less. For simplicity, we assume they have invariant topology, that is, the size and connectivity of the layers are fixed. However, at least some of their weight and bias coefficients vary from syllable to syllable. These coefficients are represented by the parameters A and B, which are weight matrices and biases.

The purpose of the box labeled “Multi-delay” in Figure 1 is to copy the output of the output layer of the recurrent network back to the input layer with one or more delays. Here it is understood that the unity of time is the frame duration for the speech signal, typically 5 ms or less. This corresponds to a duration during which spectral coefficients can be assumed approximately constant. The input to the second network is the current output of the first network and, if necessary, may be augmented by delayed previous states and inputs. The second system with output $y$ provides thus a vector of spectral and other coefficients every 5 ms or
less. These signals drive a parametric speech synthesizer that generates the speech signal.

To make this model useful requires using system identification methods to find the parameters of the overall model and its components that have the potential to represent specific data, given the proper parameters. Since for parts of the model there is no attempt to make a replica of natural systems but only to mimic them, these components need to have a rich enough parametrization so that the observed time series of measurements are within the model's overall output domain.

Besides the requirement of taking into account important facts of the natural articulatory system, an important additional requirement on the design is that known methods of system identification and state estimation can be used to infer its parameters from data. Methods and open problems for this are discussed in section 3.

2.1 Role of the C/D Model

The C/D model provides a framework for the control of articulatory movements based on utterance and syllable features. Rather than specifying speech as a sequence of phonological feature bundles, like in the classical distinctive feature theory, the C/D model abolishes the phoneme description, and replaces it by specifying syllable features. Fujimura (2004) describes the difference as follows:

In the C/D model, the concatenating units are syllables instead of phonemes. This implies that there is no concatenation of consonants and vowels; a syllable is an integral unit, which is specified phonologically by syllable or syllable component features, instead of being composed by concatenating smaller segments.

As is done in the C/D model, syllable concatenation is here only understood in an abstract sense, not on the output signal level. The proposed synthesizer implements a concatenation process between syllables using interpolation to blend over syllable specific parameter vectors from syllable to syllable. The parameters usually are coefficients that determine the overall behavior of a dynamic system while usually not setting the initial state conditions. Therefore, changing these parameters amounts to changing the dynamic regime from syllable to syllable, while co-articulation effects are expected to occur automatically. Some of the model's sub-systems will be represented by recurrent neural networks. In that case, the parameters that change from syllable to syllable are the biases and weight coefficients of the neural network.

2.2 About the Control Signals u(t)

The C/D model provides a framework of actuators which produce signals that are more or less articulatory in nature. The output of the actuators is a multidimensional control signal, here named $u(t)$. However, in the original design of the C/D model the principled decision was made to keep phonological and phonetic aspects of speech production separated. To make the connection from the abstract feature representation of syllables to a phonetic implementation, the C/D model provides a framework based on the concept of elemental gestures. Before discussing this, here are two excerpts that make this more specific - from Fujimura and Williams (2008):

"Inherent consonantal gestures are defined as local time functions (impulse-response functions, IRFs) for elemental gestures that, in superposition, implement the set of consonantal features in the same syllable component (onset, codas, or syllable affix). Each syllable is identified only by its phonological features, without reference to phonemic segments or any temporal ordering of tautosyllabic features. A skeleton, comprising a time series of syllable pulses with intervening boundary pulses, represents the rhythmic organization of the implemented phonetic phrasal structure. Each of the pulses has its scalar magnitude computed according to the discourse context. The time-variant pattern of pulse amplitudes constitutes the phonetic metrical structure, viz., stress pattern, of the utterance. Lexical and phrasal pitch accents are implemented as tonal melody contours along with vocalic contours in the base function."

"Articulatory controls in multiple independent dimensions of phonological melody, manifesting vocalic and tonal feature specifications of syllables and boundaries as step responses to changes in phonetic status, are linked in time to syllable pulses. A computation of temporal distance between the syllable pulse and its subordinate pocs (p-fix, onset, codas, s-fix) pulses that mark the edges of the syllable core and optional extension components, makes it possible for a multidimensional actuator within the C/D model to evoke inherent consonantal gestures of phonological features for each syllable component. Assuming a table of stored impulse-response and step-response functions, the model temporally links all tautosyllabic gestures to the pertinent syllable pulse. The base function of an utterance comprises the skeleton, represented by the pulse train, and associated melody functions. Consonantal gestures are superimposed onto this base function, where they are temporally organized around each
syllable pulse."

"Elemental consonantal gestures are evoked as impulse-response functions, triggered by the pertinent syllable pulse, or, more directly, its subordinate (pos) pulse with the amplitude copied from the syllable pulse. Amplitudes of all gestures reflect the pertinent syllable magnitude. By superimposing these local fast time functions for consonantal gestures onto the slower vocalic gestures of the base function, which change from syllable to syllable, we generate control functions for articulatory dimensions to produce speech signals. The mapping in this last stage of organization from the control functions, interacting with each other, into speech movements and corresponding acoustic parameters, such as formant frequencies, is strongly nonlinear."

And the following is from Bonaventura and Fujimura (2007):

"With respect to the impulse response function, the C/D model assumes that each elemental consonantal gesture constitutes a fixed ballistic motion pattern, which, as a passive response, is evoked by a time-shifted replica (onset pulse, coda pulse, etc.) of the syllable pulse as the excitation. This conceptual model of the elemental gesture is not necessarily meant to be the exact modeling of the physical process, since the movement process may well include active processes based on localized feedback processes. The point is that, phenomenologically, this picture of the whole course of action for the demisyllable as autonomous and independent from other concomitant gestures helps us to understand and represent quantitatively what occurs in the extremely complex phenomena in terms of phonetically effective control variables. In the case of vocalic gestures, since the temporal change is the effect of basically a syllable-to-syllable slow change, there may well be a significant role of auditory feedback. The C/D model provides a simplified phenomenological description that suffices to capture basic properties of speech production principles and their effective representation. For computation of the phonetic implementation, all elemental gestures for consonants, as demisyllabic constituents (i.e. onsets, codas, or syllable affixes) of the syllable are stored in an impulse-response function (IRF) table, and only its amplitude and triggering time are assumed to be controlled under the government of the syllable pulse, ..."

In the C/D model the articulatory gestures are generated by specific motion patterns that are parametrized with very few parameters, while the transformation into actual speech movements and acoustic parameters is conceptually external to the C/D model: It is the job of the signal generator. This is exactly what the proposed extended model tries to achieve.

It should be noted that for building a synthesizer the concept of actuator models that produce elemental gestures stipulates a modular approach, and the point of this proposal is to design these modules so that the C/D model can be made specific for a given speaker who is speaking a particular language (while remaining in a more or less restricted domain of speech production, such as reading aloud documents or books). Obviously a lot of parameters are needed to do this. To start, we would need to find at least a model that can predict syllable durations and syllable strengths, which are strongly coupled in the C/D model and assumed to be known. Furthermore, we need tables and algorithms to compute the relative timing of these elemental gestures specific for the syllable type.

While certain easily observable articulatory parameters such as jaw opening are quite directly connected to the elemental gestures of the model, they are not the same thing, even though in some experimental works the elemental gestures and observed articulatory parameters were basically assumed to be the same thing.

2.3 Modularity

Apparently, observed articulatory data are the next best thing we can use to find parameters of the model. This task can be simplified by first making partial models, where possible, find their parameters from data, and then augment the system by coupling them together.

In addition to the speech signal, synchronous time series of articulatory data need to be obtained, using only non-intrusive instrumentation. For example, a combination of microphones and two or more cameras that record at least at 200 frames per second, may give enough information to assess lip and jaw movements. It is possible to extract at least one parameter describing the jaw movements from video, for example by tracking painted markers, and with some more work it should also be possible to extract an approximation of the area of the mouth opening from a frontal view of a camera (relevant for radiation impedance). Further, we could find parameters describing lip spread, lip opening and, by using a combination of cameras, lip protrusion. This should be facilitated if we can succeed tracking several painted markers on the skin in three dimensions from the video data.

From a modularity approach, the next step is to build a partial model that can predict the observed movements of these parameters, from the abstract feature
representation of the recorded speech, syllable by syllable. If such a model is capable to regenerate measured articulatory data, it provides indirectly information about the entire process, which includes the production of all acoustic parameters. Making use of the modularity of the system, and first building partial models for observable articulatory data will make the parameter estimation easier.

It is most likely advantageous to put the model together in modules that can initially be partially trained independently from each other. For this we try to follow as much as possible the causality network that is suggested by the speech articulation system. This is expressed in Figure 4. For each of the partial models in the figure, which are realized either as physical models or as recurrent neural networks, it is possible to find observables in the measured data that can be used in the system identification procedures. Some of these models are highly stylized: For example, we assume that we have no direct observation of the tongue movements, but we can at least make a stylized model by using formant information. The purpose of the box labeled “virtual tongue dorsum” in Figure 4 is to compute, given the parameters from the CD model, the existing measured F1 and F2 data. Its parameters can be found by system identification applied to this reductionist model. Even if the model can only be seen as a stylized model and does not represent the physical reality of tongue movements, if it can be fitted to the data so that it regenerates F1–F2 trajectories, it can be used as a proxy to a “tongue dorsum” model in the overall model.

Figure 4 should make the order of building these partial models clearer. From the left: The pulmonary model is basically a model for phrasing: We will assume for the time being that the durations of breath pauses and other pauses between phrases is known during the model training. For the final synthesizer we may have to provide an additional heuristic model that makes decisions about the specification of breath groups, using information from the input text.

The pulmonary model is supposed to generate a set of signals that corresponds to the observed speech signal intensity (and can be physically measured as short time signal intensity or RMS). Since it is possible to measure the noise intensity in breath pauses, the pulmonary model should predict parameters of this signal. It will be necessary to specify a special set of additional speech units besides syllables, in order to represent speech phenomena like short and long pauses, breath pauses and the like. These observable speech phenomena are usually easy to measure and parameterize and should therefore be represented in the model.

The combined two boxes in Figure 4 that have to do with laryngeal control, F0 and voice quality parameters, have to be designed depending on the availability of measurements that allow to estimate its parameters: It may be possible, though more difficult especially for female speakers, to measure approximations of larynx height from video observations. If this can be done, we would like to model the dynamics of the process and use a larynx height parameter as input to the model that predicts F0 and possibly voice quality parameters, if these can be measured from the speech signal.

Part of the data acquisition of speech signals is also the extraction of parameters describing lip movements from video. The corresponding model near the center of Figure 4 will be used to re-generate these data. It uses as input, like all the other partial models, the control signals generated by the extended C/D model with the “action generator” described earlier in connection with Figure 3. In addition, it receives an input from the “jaw” model box: Jaw opening and velocity. During training we can initially use the measured jaw movement data, but use the model-generated jaw movements while tuning the lip model to predict the lip movements even if the jaw movements may not exactly match the data. The reason for this training strategy is that the finished synthesizer does not have the measured jaw movements, nor any other measurable articulatory data, so each partial dynamic system component that depends on other (usually earlier trained) components has to compensate as well as possible for the imprecisions of their parents, both during training and at run time.

The output of the lip movement model, just like the outputs of the other boxes, is used to augment the
control signal for the acoustic parameter model. This model depends on all other systems, and is trained last, using measured spectral parameters from the speech signal as training data. During its training the fixed parameters of the other partial models are no longer updated.

3. Methods and Open Problems

The model is designed to generate trajectories of speech parameters that correspond, at least in principle, to actual measurements on the speech process of individual human speakers. The model provides for each type of measurements observer functions or emission models, the states and parameters of which need to be found from data.

The input to the model are sequences of features that are extracted from text, which for the training phase is either provided as reading material or obtained by automatic and corrected transcription of the speaker's utterances. One could say that causality in this model is partially suspended: The production of the current syllable, current sound etc. may depend not only on the current and syllable but also on the next few and perhaps even the next sentence or paragraph. This is because both prosodic and syllable features are obtained from a larger window of analysis of the text. This dependency of current speech output on past and future events is coherent with how humans not only think ahead but also automatically use speech motor pre-planning with several hundred milliseconds look-ahead.

Since the process of the abstract discrete feature generation in the model is a construct and can not be observed, inference about which features are active at a given time from measured data is rather limited and is for the most part not attempted. It is left to the control model to provide the sequence of abstract features (for a given text), while finding the details of timing is a task of inference from data. It is thus attempted to find parameters related to the relative timing of articulatory events, and further to find parameters that determine the fine details of speech signal output.

3.1 Bayesian Networks and Timing

Figure 4 presents a simplified sketch of the dependencies within the speech production model. It also indicates the information flow, both from the abstract feature processes downstream and upstream from the data to partial models. The model is formulated as a generative model, and even though Bayesian inference for the model is intractable in general, it is often possible to use approximate Bayesian inference to find details of parts of the model from data. This is in particular viable if there are partial generative models that explain certain observable data independently from others. One example of this is the partial generative model of jaw movements, and another is a simple glottis model that can be fit to speech signals obtained by inverse filtering, see below.

If the boxes in the modular design shown in Figure 4 are considered as nodes and the control links as edges, then the whole system forms a directed acyclical graph (DAG), while this is usually not true for what is inside the boxes. However, since each box has a clearly defined set of inputs and outputs, it is possible to describe the entire system as a dynamic Bayesian belief propagation network, and analyse the parameter estimation as solving a Bayesian inference problem.

The general strategy is to start with training subsystems that are “close to” observable data, for example, jaw movements, lip movements, but also the glottal waveform for voiced sounds, obtained from the speech signal by means of a simple physical model of the glottis. Once these systems are trained, they can remain fixed during the training of more detailed sub-systems, in particular the subsystem that generates acoustic parameters, which has several inputs from other submodels and shown in Figure 4.

3.2 Intractability Problem

Three kinds of parameters/data can be distinguished in building and describing the proposed dynamic models: (a) a model's instantaneous state as a point in some phase space, which can be generalized coordinates and moments or velocities describing the movements of subsystems based on physics, for example in a glottis model, see Appendix A.2, or other rapidly changing parameters together with their corresponding rates of change in arbitrary domains - for example for spectral coefficients; (b) time varying control signals that are computed outside of the dynamic system model, and (c) configurational parameters, which either remain fixed throughout or are changed over time by a process that is external to the dynamic system.

For example, if a version of the two-mass model of the glottis is used as a submodel, parameters of type (a) are the positions and velocities of the two masses and several flow variables; type (b) parameters are slowly varying control signals that determine the aduction of the glottis; and examples of parameters of type (c) are the width of the glottis, the length of the tracheal tube and the volume of the supra-glottal ventricular space.
The second example is a recurrent neural network. Parameters of state type (a) are the hidden states in the layers of the network; parameters of type (b) are the control input, and parameters of type (c) are the bias and weight coefficients that fully define the network. Also of type (c) are parameters in a system that controls changes of bias and weight coefficients in the network depending on syllable specific features.

Methods to estimate parameters from data, in particular of types (a) and (c), will be discussed further below. Probably the hardest problem is to find time varying control input signals of type (b). They usually cannot be observed, and trying to estimate them is an intractable problem, because the output of the model depends on its input only by means of the dynamic state, which itself needs to be estimated.

For example, while it may be possible (while not attempted here) to build a more or less simplified physical model for the tongue muscular system, finding the input to such a model, namely muscle activation signals, is almost completely intractable. Even if actual estimates could be made, they are next to impossible to validate by measurements, unless the commitment to only using non-intrusive measurements is dropped, which is not an option.

Instead of trying to solve this problem directly, the C/D model is used to compute prototypical control signals, which are utterance and syllable specific (for a language or dialect) but usually not speaker specific. These signals need to be further shaped by modules that are specifically adapted to individual speaker’s speech.

The advantage of this approach is that the estimation of dynamic states and the system identification for finding fixed parameters can be simplified if the control input to the dynamic systems can be assumed to be known. It is then a matter of changing the remaining parameters of the model so that it can represent measured data, regardless of whether or not the input signals are realistic or similar to actual signals in natural speech production.

The immediately apparent difficulty is that for training subsystems the timing has to be known, in particular, for some of the submodels their input has to be known which depends on the relative timing of articulators.

An alignment between the measured data stream and the timing of the hidden feature variables, including syllable features, is necessary for training. An approximate alignment can be obtained initially by methods that are also used in automatic alignment in HMM speech. While in the classic discrete Markov model for speech the timing is only indirectly specified by means of a matrix of transitional probability coefficients between Markov states, it is possible to build instead a semi-Markov model where the time of stay in one of the discrete states is explicitly modeled by a continuous probability distribution. The forward–backwards (Baum–Welch) algorithm for training the HMM’s parameters is then modified, which is exemplified in Zen et al. (2007) for the case of Gaussian distributions for durations. This amounts to using alternating steps of reestimating the parameters of the emission models for each Markov state, assuming the most likely durations of stay, and reestimating the duration models after finding the most likely points of state transitions given the updated emission models. A method of finding the most likely durations of stay for a sequence of Markov states, using instead more appropriate Gamma distributions for the duration models, is outlined in Appendix A.1.

### 3.3 Learning Parameters by System Identification

The proposed speech production model has to make specific how the feature description of speech translates into a multi-dimensional output of parameters that can be directly used to generate a speech signal. It is obvious that the model requires a large number of parameters that have to be found from data.

One role of the C/D model is to compute control parameters for modules in the model (the actuators). To make the distinction more specific, in this augmented model the control signals generated by the actuators in the C/D model are not directly functions of time; instead they are parameterized as functions of “syllable time,” and then translated by speaker specific models for syllable timing into actual time dependent control signals. This facilitates the parameter identification for the rest of the model, since we are now dealing with dynamic systems with known control input.

For the purpose of state estimation and system identification of parameters in the recurrent neural networks, but also for the physically inspired partial models, specific Bayesian filtering methods exist, and one will be applied here. In particular the most recent development of nonlinear Kalman filters, namely the Cubature Kalman filter, see Arasaratnam and Haykin (2010), Arasaratnam et al. (2010), and Arasaratnam (2013), provide very robust techniques for nonlinear state estimation and parameter identification. The use of these filters is exemplified in Appendix A.2 for a lumped dynamic model of the glottis.

It is unlikely that the identification of all parameters for such a complex model can be obtained in one big
learning task for all parameters simultaneously, hence the modularity in the design. For instance, if we can obtain the parameters for a recurrent neural network structure that models the jaw movements and lip parameters, we can keep these fixed when the model is further extended: The signals produced by already fixed components can be used to augment the output of the C/D model to the other components of the model.

In the learning algorithm, switching or transitioning between different dynamic regimes is accomplished by assuming a weighted interpolation between dynamic system parameters in the transition from one syllable to the next (for example by using simple quadratic splines). The learning parameters need to be weighted accordingly: During the transition from one syllable to the next, the parameters of the most active syllable at a given time are modified in training with the largest learning parameter, and only based on training data that correspond to that syllable.

### 3.4 Speaker and Language Specific Models

In the proposed model it is assumed that there is a pronunciation dictionary in which the syllabification of almost any word can be looked up, otherwise robust fallback methods have to be built that allow to generate the syllabification from rule. As is the case for other text to speech synthesizers, the feature description of speech is not complete by providing the syllable features alone, but it is a very significant part.

In particular for syllable feature lookup already some work exists that can be used here. Figure 5 expresses the syllable feature organization of American English in a tree structure; this is based on work by J. C. Williams (personal communication), who created a database for pretty much any syllable that may appear in American English, where it should be noted that German languages are among those with the most complex syllable structure.

Obviously a similar task is ahead for other languages, preferably concentrating on frequently spoken languages with simpler syllable structure. Thereby it still must be assumed that the description of speech from text is not complete with the syllable feature description, and additional work needs to be done to find phrase and pause features.

**References**


Wilson, I. and D. Erickson (2013) “Effect of syllable on-
set, coda, and nucleus on degree of skin stretching over mandible.” *Proceedings of International Congress on Acoustics* 19.


(Received May 22, 2015, Accepted Sept. 4, 2015)

**Appendix**

**A.1 Maximal Likely Durations of States in a Markov Model**

In the classic hidden Markov model (HMM) the duration of staying in one of the states is often defined in terms of transitional probabilities, in which the probability of going to state $j$ if the Markov process is in state $i$ is represented by a transitional coefficient, e.g., $Pr(s_i = i|s_{i-1} = j) = a_{ij}$, which need to be estimated during training.

An alternative is to assume for each state a continuous probability distribution over the length of stay in that state. The appropriate type of probability density function for durations is the Gamma pdf, which is defined for a random variable $d$ as: $p(d; \kappa, \Theta) = \frac{d^{\kappa-1}e^{-d/\Theta}}{\Gamma(\kappa)}$ with a shape parameter $\kappa$ and a scaling parameter $\Theta$. The maximum (mode) of this distribution is at $d = \Theta(\kappa - 1)$.

In the simplest case it can be assumed that these durations are mutually independent and thus only depend on the state of the Markov process, making the transitional probability tables $a_{ij}$ superfluous. Otherwise, each $a_{ij}$ would need to be replaced by a couple $(\Theta_{ij}, \kappa_{ij})$, defining one duration probability for each pair of states. The simpler case is assumed in the following: The duration probability only depends on the current state.

Suppose now that for an utterance the sequence of $M$ states $s_i$ is known, and with it the coefficients $\kappa_i$ and $\Theta_i$ are known, and it is assumed further that the total duration is known as $T$. Then we ask the question: Which set of duration values $\{x_i, i = 1, \ldots, M\}$ maximizes the probability functions for the states $s_i$? This leads to maximizing the product of the probabilities, $\prod_{i=1}^{M} \frac{x_i^{\kappa_i-1}e^{-x_i/\Theta_i}}{\Gamma(\kappa_i)}$ under the constraint $\sum_{i=1}^{M} x_i = T$. Instead we use the logarithm of the distribution, making use of a Lagrange multiplier $\lambda$. Thus, one obtains a likelihood function that needs to be maximized:

$$l = \sum_{i=1}^{M} (\kappa_i - 1) \log x_i - x_i/\Theta_i - \kappa_i \log \Theta_i - \log \Gamma(\kappa_i)$$

$$+ \lambda(T - \sum_{i=1}^{M} x_i)$$

(1)

If it can be further enforced that all $\kappa_i > 1$ and $\Theta_i > 0$, this function has a unique maximum because of the convexity of the logarithm and the fact that any Gamma distribution goes to zero for very large values of $x_i$. Taking the derivative with respect to the unknown $x_i$ and $\lambda$ leads to $M + 1$ equations:

$$1 - \kappa_i + \frac{x_i}{\Theta_i} + \lambda x_i = 0 \quad \text{for } i = 1, \ldots, M$$

(2)

$$\sum_{j=1}^{M} x_j - T = 0$$

(3)

The system is nonlinear because of the products $x_i \lambda$, and it can be solved using a Newton method. If we denote current vector of durations at step $n$ as $x_n$, the following update equations have to be solved for the increments $\Delta x = x_{n+1} - x_n$ and $\Delta \lambda = \lambda_{n+1} - \lambda_n$:

$$\begin{pmatrix} A & x_n \\ 1^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -f_n \\ -g_n \end{pmatrix}$$

(4)

In the Jacobian matrix on the left, the matrix $A$ is a diagonal $M \times M$ matrix with the diagonal elements $\frac{1}{\Theta_i} + \lambda$ for $i = 1, \ldots, M$, and $I$ is a $M$-dimensional vector of all 1’s. The $M$-dimensional vector $f_n := J(x_n, \lambda_n)$ has components $1 - \kappa_i + \frac{x_n}{\Theta_i} + \lambda x_n \ln \Theta_i$ and $g_n = \sum_{j=1}^{M} x_{n,j} - T$. This results in

$$\Delta \lambda = g_n - 1^T A^{-1} f_n$$

$$\Delta x = -A^{-1} f_n - A^{-1} x_n \Delta \lambda$$

(5)

(6)

Since $A$ is diagonal it can then be seen that

$$\begin{pmatrix} A^{-1} f_n \end{pmatrix}_i = x_{n,i} + \frac{1 - \kappa_i}{\Theta_i}$$

$$\begin{pmatrix} A^{-1} x_n \end{pmatrix}_i = \frac{x_{n,i}}{\lambda_n + 1/\Theta_i}$$

(7)

(8)

and multiplying from left with $1^T$ means summation over the elements of these vectors. The update equation for $\lambda$ is:

$$\lambda_{n+1} = \lambda_n - \frac{T + \sum_{i=1}^{M} x_{n,i}}{\sum_{i=1}^{M} x_{n,i}}$$

$$+ \frac{1}{\sum_{i=1}^{M} \frac{1}{\lambda_{n+1}/\Theta_i}}$$

(9)
Instead of updating equation (6), new values for $x_{n+1}$ can also be obtained simply by solving each equation (2) as:

$$x_{n+1,j} = \frac{k_i - 1}{\lambda_{n+1} + 1/\Theta_i}$$

The iteration may be started with $\lambda_0 = 0$ and $x_{0,j} = \Theta_i(k_i - 1)$ which are the mode values for each of the Gamma pdf’s. In experiments this algorithm converged within 3 to 5 steps to high precision.

A.2 Estimation of the State of a Dynamic Glottis

Model from Speech

In this preliminary experiment, a version of the two-mass model of the glottis from Lucero and Koenig (2005) was combined with some lumped elements representing wall vibrations, losses and resonances in the trachea and the vestibular space as well as using an assumed constriction somewhere in the vocal tract above the glottis with cross section $A_v$ with viscous loss $F_v(A_v, U_v)$ (not shown in the figure). The small circuit diagram can be described with the following 7 differential equations, in which $A_m$ represents the minimal cross section in the glottis, and the other variables can be found from inspection of Figure 6:

$$\left( L_x + L_z \right) \dot{U}_x - L_x \dot{U}_y = P_t - R_i U_s - R_e (U_s - U_g + V_i)$$

$$L_x \dot{U}_y + (L_c + L_g) \dot{U}_g = P_t - R_i U_s - R_e (U_s + V_g) + 2 \frac{U_g^2}{2} \left( \frac{1}{\lambda_t^2} - \frac{1}{\lambda_e^2} \right)$$

$$L_x \dot{U}_g + L_g \dot{U}_l = R_e (U_s + V_g) - R_i (U_g + V_i)$$

$$L_e \dot{U}_g + L_g \dot{U}_e = R_e (U_s + V_g) - R_i (U_g + V_i)$$

$$L_e \dot{U}_e - L_g \dot{U}_g = (L_c + L_t) \dot{U}_l$$

The last three equations come from additional dynamic state variables for turning second order systems into first order systems. The above seven differential equations, together with four more first-order differential equations for the displacements and velocities of the coupled two-mass dynamic system of the half-glottis, see Lucero and Koenig (2005), constitute together an eleven-dimensional continuous state space model. The system of explicit ordinary differential equations is solved numerically with a fixed step third-order Runge–Kutta method.

The speech material, sentences recorded from a female speaker, was down-sampled to 16 kHz and inverse filtered after preemphasis with a 16th order prediction filter; and the residual signal was integrated with a leaky first order integrator, so that the resulting signal roughly compares to the rate of change of the glottal volume flow.

It was then assumed that this residual signal, using an empirically found scaling factor, represents the measurements corresponding to the observables of the 11-state glottis model, namely the differential of the estimated glottal flow, and the Cubature Kalman filter algorithm was used to estimate the eleven states of the model for every sampling point, while between sampling points the model’s state was propagated forward.
Text to Speech Synthesis Using Syllables as Functional Units of Speech

by the Runge–Kutta solver.

Figure 7 shows an example of the obtained estimated glottal flow and the estimated movements of the two masses in the glottis model.

The cubature Kalman filter computes the estimates of the state and covariance of the state by sampling its prior normal distribution at $2N$ points ($N$ is the number of dimensions) and thus evaluates numerically the expectation integrals. The time-discrete version of this algorithm is used because the state transition function is implemented here by numerically integrating the dynamic equations of motion for each time interval between measurements. Like for other Kalman filter methods, a weakness of the method is that the results depend to some extent on reasonable initial estimates for the variance of the measurement noise and the system noise. The work-around was to reestimate repeatedly the measurement noise from the actual error signal between model predictions and data, and to define the system noise variances of the states of the model as fixed small ratios of the variances of the states themselves.

Currently the model relies on simple heuristics using a voiced/unvoiced feature computed from the speech signal for computing the adduction and abduction, which becomes the rest position of the two masses representing the vocal folds, and further makes use of an approximation of FO to modulate the stiffness ratio of the vocal folds. The overall model may be improved by extending it to directly account for wall losses, if measurements from skin surface contact microphones can be incorporated.