Deep Learning Techniques in Tandem with Signal Processing Cues for Phonetic Segmentation for Text to Speech Synthesis in Indian Languages

Arun Baby, Jeena J Prakash, Rupak Vignesh and Hema A Murthy

Department of Computer Science and Engineering
Indian Institute of Technology Madras

arunbaby@cse.iitm.ac.in

Abstract

Automatic detection of phoneme boundaries is an important sub-task in building speech processing applications, especially text-to-speech synthesis (TTS) systems. The main drawback of the Gaussian mixture model - hidden Markov model (GMM-HMM) based forced-alignment is that the phoneme boundaries are not explicitly modeled. In an earlier work, we had proposed the use of signal processing cues in tandem with GMM-HMM based forced alignment for boundary correction for building Indian language TTS systems. In this paper, we capitalise on the ability of robust acoustic modeling techniques such as deep neural networks (DNN) and convolutional deep neural networks (CNN) for acoustic modeling. The GMM-HMM based forced alignment is replaced by DNN-HMM/CNN-HMM based forced alignment. Signal processing cues are used to correct the segment boundaries obtained using DNN-HMM/CNN-HMM segmentation. TTS systems built using these boundaries show a relative improvement in synthesis quality.

Index Terms: Deep Neural Networks, Convolutional Neural Networks, phonetic segmentation, signal processing cues

1. Introduction

Segmentation of speech into accurate time-aligned phonetic transcriptions plays a vital role in building robust speech systems, including statistical parametric speech synthesis (SPSS) systems, as the duration of HMM states is explicitly modeled and generated during synthesis [1]. The widely used HMM-based forced alignment is not ideal for speech synthesis [2, 3] as the location of the phoneme boundary is not used as a criterion for estimation of parameters, and often requires manual correction after the forced alignment. Manual labeling for a huge multi-lingual corpus is time-consuming and error-prone which warrants automatic procedures that are better than Viterbi force-aligned HMM segmentation. There have been many attempts at improving the accuracy of the HMM segmentation. In [4], a spectral transition measure is used to correct boundaries having abrupt spectral changes. In [3], the boundaries were iteratively moved forward or backward by one frame, depending upon the direction in which frame classification accuracy is increased.

Wherever hand-labeled data is available, for example, the TIMIT corpus [5], machine learning models have been trained to learn the boundaries [6]. For example, [7] uses support vector machine (SVM) and [8] uses a multi-layer perceptron to refine the HMM boundaries. The best-reported results on TIMIT database use a fusion of multiple acoustic front-ends (i.e. systems based on MFCCs, PLPs, RASTA-PLPs), on top of boundary correction models such as neural networks and single-state HMMs, thereby improving the segmentation accuracy to 96.7% within a tolerance of 20 ms [9]. However such hand-labeled data is not available for Indian languages.

Accurate phonetic segmentation becomes a problem when only the phoneme sequences are available and not their boundary locations. For syllable-timed languages, signal processing cues that are agnostic to the speaker can be used to get syllable boundaries [10]. Signal processing cues result in false alarms but seldom introduce deletions when the parameters are chosen such that the boundaries are overestimated. The phonetic transcription can be used in tandem with signal processing cues to eliminate insertions in such cases. Signal processing cues along with HMM-based alignment has been used for segmenting speech data in TTS systems for Indian languages, that are syllable-timed [11].

Conventionally, the posterior probability of how well an HMM state fits a frame is decided by a GMM [12]. With the recent success of deep neural networks (DNN) and convolutional deep neural networks (CNN) for automatic speech recognition (ASR), DNNs and CNNs have outperformed GMMs in acoustic modeling as they can handle highly non-linear relationships between the input and output [6]. Even though neural networks are widely used in speech recognition, they are not used for speech segmentation for TTS. This work is an attempt to exploit the discriminative power of deep networks in the context of phoneme segmentation.

The rest of the paper is organised as follows. Section 2 explains the role of signal processing cues in speech segmentation. The proposed system is described in Section 3. Section 4 discusses the experimental setup and results obtained. The work is concluded in Section 5.

2. Acoustic cues for boundary correction

For syllable-timed languages, minimum phase group delay (GD) based processing of short-term energy (STE) is used for obtaining syllable boundaries [13]. Although GD based segmentation gives accurate syllable boundaries, it introduces a number of spurious boundaries for syllables starting or ending with a fricative or nasal, and syllables that start with a semivowel or affricate. The number of boundaries given by GD segmentation is decided by an empirical parameter, window scale factor (WSF) which determines the size of the lifter [1]. An appropriate value of WSF is chosen such that syllable boundaries given by HMMs are only approximated to boundaries of high confidence [11].

Additionally, spectral flux is used to address the issues of inaccurate syllable boundaries in the context of fricatives, affricates, nasals and semivowels. A modified version of spectral

[1]

$$Size\ of\ the\ lifter\ = \frac{Length\ of\ STE\ function}{WSF} \quad (1)$$

flux called sub-band spectral flux (SBSF) is used as a cue for boundary correction [14, 15].

The correction of the boundary between two syllables, syllable 1 and syllable 2, is hence performed on the basis of end phone of syllable 1 and start phone of syllable 2. The following correction rules are applied for obtaining accurate syllable boundaries:

Rule 1: The boundary between syllable 1 and syllable 2 is corrected using STE if syllable 1 does not end with a fricative or nasal and syllable 2 does not begin with a fricative, affricate, nasal or semi-vowel.

Rule 2: The boundary between syllable 1 and syllable 2 is corrected using SBSF if either the end phone of syllable 1 or the start phone of syllable 2 is a fricative or an affricate, but not both.

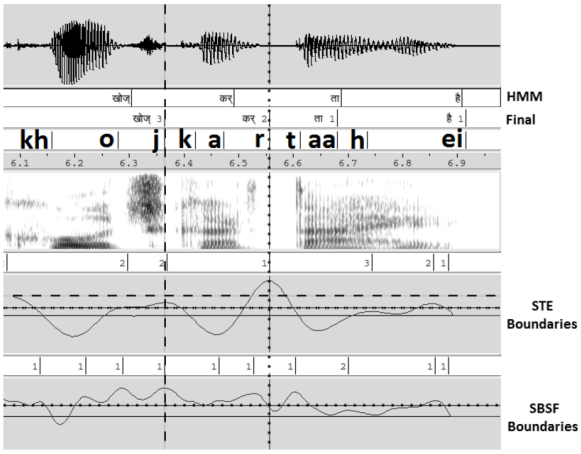

Figure 1: *Boundary correction using STE and SBSF.*

Figure 1 [2] shows our earlier efforts on boundary correction of the GMM-HMM (referred to as GMM-BC) system [14]. The panel HMM shows the initial syllable level alignment using GMM-HMM flat start. GD processing of STE, and SBSF are used as signal processing cues to correct the initial syllable boundaries. The panel Final shows the corrected syllable boundaries obtained using STE and SBSF. The boundary of the syllable *k-a-r* is corrected using STE using Rule 1 and boundary of the syllable *kh-o-j* is corrected using SBSF using Rule 2.

3. Proposed system

Neural networks are not used for speech segmentation in the TTS framework for Indian languages even though they are widely used in speech recognition. In this work, GMMs in HMM-GMM framework for phoneme segmentation in TTS systems are replaced by DNN and CNN for better phoneme segmentation. Acoustic models are built by training the neural networks with the GMM-HMM monophone alignment (also known as HMM-based phone alignment) as the initial alignment. The DNN-HMM/CNN-HMM are then trained iteratively to get accurate final phone boundaries. This is shown in Block II of Figure 2. The number of iterations is set to 8 empirically as the phone boundaries do not change much afterward.

Acoustic cues give robust syllable boundaries for a subset of syllables as discussed in Section 2. Syllable boundary correction using signal processing cues (GD of STE, and SBSF)

[2]This Figure is reproduced from [14] with author's permission

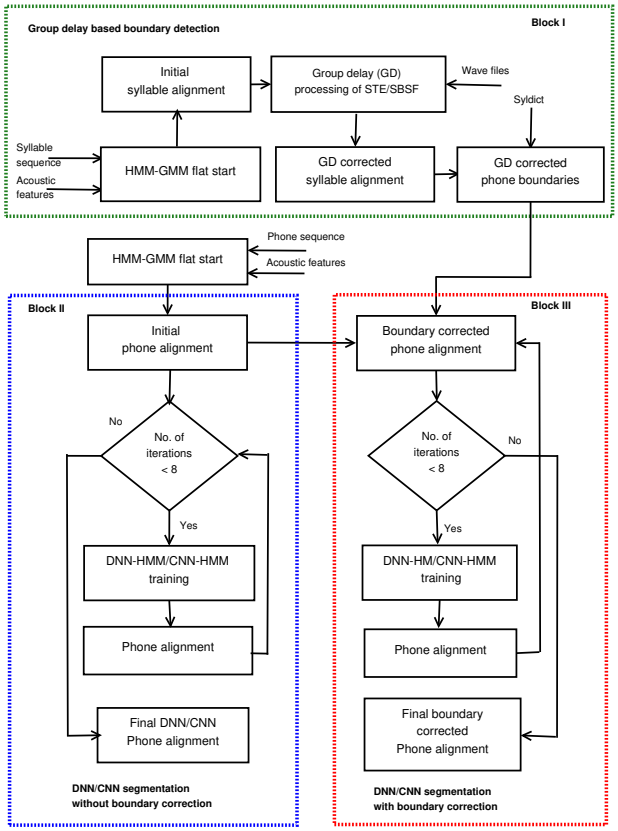

Figure 2: *Block diagram of proposed system*

after GMM-HMM flat start initialisation is shown in Block I of Figure 2. The boundaries of the last phone of each corrected syllables are marked as GD corrected phone boundaries. A syllable to phone dictionary (shown as syldict in Figure 2) is used to map from syllable to phoneme sequence. Most of the phone boundaries given by neural networks are better than GMM-BC alignment, which uses signal processing cues along with GMM-HMM based forced alignment. But they are not able to outperform with respect to some of the boundaries obtained using GMM-BC segmentation.

The proposed framework, where the boundaries obtained using DNNs/CNNs are further corrected using signal processing cues is shown in Block III of Figure 2. Similar to segmentation using deep networks, GMM-HMM monophone alignment is used as the initial phone alignment. These phone alignments are corrected, either forward or backward, using GD corrected phone boundaries. The boundary corrected phone alignments are then used for training neural networks. The alignments obtained after deep network training are again corrected using GD corrected phone boundaries and this process is repeated 8 times iteratively. After the 8^{th} iteration, phone alignment obtained from deep networks are corrected again using GD corrected phone boundaries as shown in Figure 2.

4. Experiments and Results

4.1. Datasets Used

The experiments are conducted on five Indian languages. A subset of Indic database [16] is used for the experiments. The details of the data sets used are given in Table 1. The utterances

are recorded by a single native speaker of the corresponding language in a noise-free studio environment at a sampling rate of 48KHz, 16 bits per sample. For grapheme to phoneme conversion of the native text, a unified parser for Indian languages is used [17].

Table 1: *Dataset used*

Language	**Gender**	**Duration** (in hrs)	**No. of utterances**	**No. of phones**
Hindi	Male	5.00	2192	58
Hindi	Female	5.00	2144	58
Bengali	Male	5.00	3093	52
Kannada	Male	3.43	1289	49
Kannada	Female	3.82	1229	48
Malayalam	Male	5.00	3063	52
Telugu	Male	4.24	2478	49

4.2. Segmentation

Segmentation of speech data is performed at phone level using the following methods:

- GMM-HMM with boundary correction based on signal processing cues
- DNN-HMM without any boundary correction
- CNN-HMM without any boundary correction
- DNN-HMM with boundary correction based on signal processing cues
- CNN-HMM with boundary correction based on signal processing cues

In the first approach, conventional GMM-HMM framework is used for phone level segmentation. 39-dimensional mel frequency cepstral coefficients (MFCC) features are used for training HMMs. Vowels, consonants, and pauses are modeled as 5-state 2-mixture, 3-state 2-mixture, and 1-state 2-mixture models respectively. In the next four methods, DNN-HMM/CNN-HMM is used for segmentation. For training the DNN and CNN, 40-dimensional filter bank features are used. The number of layers and the number of nodes used in DNN/CNN are detailed later.

4.2.1. GMM-HMM with boundary correction (BC) based on signal processing cues (GMM-BC)

The boundaries obtained with HMM-based segmentation may not be accurate; group delay based processing of STE and SBSF is used in tandem with HMM-based forced alignment to obtain better syllable boundaries. The waveforms are then spliced at the syllable level and embedded re-estimation is performed by restricting to the syllable boundaries and monophone models are built. This process is detailed in [14] (chapter 4).

4.2.2. DNN-HMM

The DNN architecture used here has 6 layers. The steps involved in DNN-HMM segmentation is given in Algorithm 1. The initial alignment is given from GMM-HMM flat start initialisation, after which RBM pretraining is performed. DNN is trained using the initial alignment. 90% of the alignment is used for training and 10% is used for validation. The alignment obtained from DNN training is fed back iteratively 8 times as shown in Block II of Figure 2.

Algorithm 1 DNN-HMM segmentation

1. **Input Features:** 40 dimensional filter bank features are used as input for DNN. The features are spliced over 11 frames to add context information to DNN.
2. **RBM Pretraining (6 layers):**
   - A layer by layer training of RBM is performed.
   - The first layer of RBM is a Gaussian-Bernoulli layer and is trained with an initial learning rate of 0.01.
   - The rest of the layers are Bernoulli-Bernoulli layers and are trained with an initial learning rate of 0.4.
   - The momentum parameter is set to 0.9 and 20 epochs are used for training each layer.
3. **DNN Training (6 layers):**
   - The DNN weights are layer by layer initialized with the pre-trained RBM weights.
   - The DNN is trained using stochastic gradient descent using back propagation.
   - A mini-batch size of 256 is used for training.
   - After each epoch, the network is tested on the error-validation data to determine whether to accept or reject the model. If the model is rejected the learning rate is halved for the next epoch.

4.2.3. CNN-HMM

The steps involved in training the CNN is given in Algorithm 2. CNN training is done similar to DNN training as discussed in Section 4.2.2.

Algorithm 2 CNN-HMM segmentation

1. **Input Features:** 40 dimensional filter bank features with 3 pitch coefficients are given as input to the network. The features are spliced over 11 frames to add context information into CNN training.
2. **Convolutional layer:**
   - Two convolutional layers are used with 1024 nodes in each layer.
   - The convolutional window is of dimension 8.
   - A pooling window of size 3 and no overlap of pooling window is used in pooling layer.
   - The CNN layer used a feature map number of 256 and 128 for first and second convolutional layer respectively.
3. **Fully connected layer:**
   - 4 fully-connected layer with 1024 nodes in each hidden layer is used.
   - The fully connected layer are trained by first performing a pre-training and followed by iterative training using the features extracted through CNN layer.

4.2.4. *DNN-HMM with boundary correction (DNN-BC)*

The initial monophone alignment (GMM-HMM flat start initialisation) is modified with GD corrected phone boundaries. This corrected monophone alignment becomes the initial alignment for DNN training. DNN training is similar to that in Section 4.2.2 except that, after each iteration, the phone alignment is corrected as explained in Block III of Figure 2. After the 8^{th} iteration, the boundaries are again corrected using GD corrected boundaries to get the final phone alignment.

4.2.5. *CNN-HMM with boundary correction (CNN-BC)*

CNN training is performed similarly to Section 4.2.3. Boundary correction is performed similarly to DNN-HMM with boundary correction as explained in Section 4.2.4.

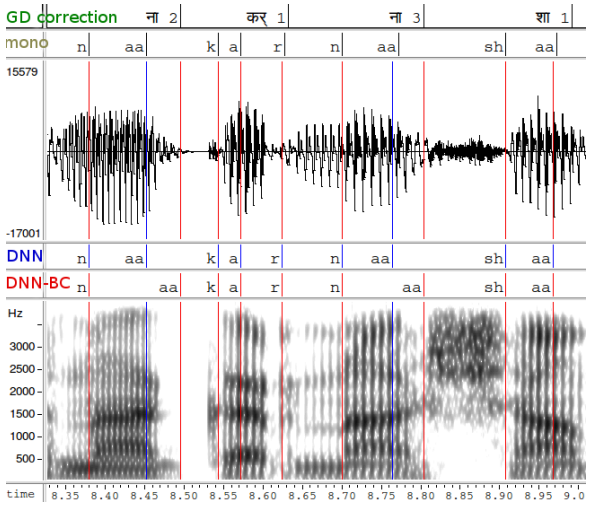

Figure 3: *Sample STE and SBSF correction with DNN*

Figure 3 shows a part of a Hindi speech utterance, with syllable boundaries obtained after GD based correction, initial monophone alignment with HMM flat start initialisation (shown as mono in figure), with phone boundaries obtained using DNN segmentation with boundary correction (shown as DNN-BC in the figure) and without boundary correction (shown as DNN in the figure). The numbers 1, 2 and 3 corresponds to: 1 for no boundary correction, 2 for STE based correction, and 3 for SBSF based correction. Phone boundaries obtained with DNN segmentation without boundary correction is shown in blue colour and that obtained after performing boundary correction is shown in red colour. It is observed that DNN with GD based correction, the boundary of vowel *aa* in the syllable *n-aa* is improved in two different contexts. In the first *n-aa*, boundary of *aa* is corrected using STE (Rule 1 of Section 2) and in the second *n-aa* it is corrected using SBSF (Rule 2 of Section 2).

4.3. Text-to-speech systems

HMM-based speech synthesis systems with STRAIGHT (HTS-STRAIGHT) [18] are built with the segmentation obtained using various approaches discussed in Section 4.2. The built systems are evaluated by subjective measures by conducting two listening tests- degradation mean opinion score (DMOS) and word error rate (WER). The DMOS and WER tests are performed by 6-20 participants across the various languages.

For DMOS evaluation, 30 different sentences synthesized using each of the methods discussed in Section 4.2 is played randomly along with originally recorded utterances. The participants are allowed to listen to the speech utterances only once. Participants are asked to rate all the sentences on a scale of 1-5, where 5 being the best and 1 refers to the worst quality. In the second test to calculate WER, participants are asked to listen to semantically unpredictable sentences (SUS) and transcribe them. Semantically unpredictable utterances are generated using each of the 5 methods (Section 4.2). WER is calculated based on the number of insertions, deletions, and substitutions in the transcription. The result of DMOS and WER tests is shown in Tables 2 and 3 respectively. Compared to the DNN/CNN systems, DMOS test shows an average relative improvement of 9.42% for DNN-BC/CNN-BC systems across the languages. These systems also show significant improvement of 14.8% over the GMM-BC system.

Table 2: *Degradation mean opinion scores*

Language	CNN	CNN-BC	DNN	DNN-BC	GMM-BC
Hindi-male	4.03	4.32	4.08	4.55	3.99
Hindi-female	3.35	3.70	3.36	3.51	3.17
Bengali-male	3.26	3.71	3.18	3.60	3.02
Kannada-male	3.64	3.72	3.42	3.44	3.40
Kannada-female	3.13	3.51	3.22	3.44	3.19
Malayalam-male	3.82	4.40	4.02	4.43	3.44
Telugu-male	3.50	4.08	3.67	3.92	3.48

Table 3: *Word error rates (%)*

Language	CNN	CNN-BC	DNN	DNN-BC	GMM-BC
Hindi-male	3.14	0.28	4.42	2.00	5.85
Hindi-female	7.50	2.50	6.00	1.00	8.75
Bengali-male	6.50	1.81	5.55	1.61	6.40
Kannada-male	4.33	2.00	3.33	2.00	5.66
Kannada-female	4.76	3.57	3.57	2.38	5.95
Malayalam-male	3.33	1.66	3.33	0.50	5.66
Telugu-male	8.18	1.59	7.46	2.69	9.90

5. Conclusions

Parametric speech synthesis systems also require accurate segmentation of the training data at phone level for training a good model. Acoustic modeling using DNNs has shown great promise in the context of ASR for many languages. Nevertheless, the phone boundaries are still inaccurate for speech synthesis systems. In this paper, an attempt is made to improve the boundaries obtained in a DNN-HMM/CNN-HMM system using signal processing cues. TTS systems are built using the obtained phoneme segments. Results of the listening test show that the quality is improved after using signal processing cues along with deep learning techniques. Sample test utterances used for the evaluation are available at the https://www.iitm.ac.in/donlab/is2017/seg.php.

6. Acknowledgements

The authors would like to thank the Department of Information Technology, Ministry of Communication and Technology, Government of India, for funding the project, Development of Text-to-Speech synthesis for Indian Languages Phase II, Ref. no. 11(7)/2011HCC(TDIL). The authors would also like to thank Dr. S Umesh, Sandeep Reddy Kothinti and Basil Abraham of Speech Processing lab, Department of Electrical Engineering, IIT Madras for helping with the kaldi-toolkit. The authors also thank Aswin Shanmugam for permitting to use Figure 1.

7. References

[1] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[2] A. Sethy and S. S. Narayanan, "Refined speech segmentation for concatenative speech synthesis." in *INTERSPEECH*, 2002.

[3] A. W. Black and J. Kominek, "Optimizing segment label boundaries for statistical speech synthesis." in *ICASSP*, 2009, pp. 3785–3788.

[4] Y. jun Kim and A. Conkie, "Automatic segmentation combining an hmm-based approach and spectral boundary correction," in *ICSLP*, 2002, pp. 145–148.

[5] V. Zue, S. Seneff, and J. Glass, "Speech database development at mit: Timit and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351 – 356, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0167639390900107

[6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[7] H.-Y. Lo and H.-M. Wang, "Phonetic boundary refinement using support vector machine," in *ICASSP*, vol. 4, 2007, pp. IV–933.

[8] K.-S. Lee, "Mlp-based phone boundary refining for a tts database," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 981–989, 2006.

[9] A. Stolcke, N. Ryant, V. Mitra, J. Yuan, W. Wang, and M. Liberman, "Highly accurate phonetic segmentation using boundary correction models and system fusion," in *ICASSP*, 2014, pp. 5552–5556.

[10] V. K. Prasad, T. Nagarajan, and H. A. Murthy, "Automatic segmentation of continuous speech using minimum phase group delay functions," *Speech Communication*, vol. 42, no. 3, pp. 429–446, 2004.

[11] S. A. Shanmugam and H. Murthy, "A hybrid approach to segmentation of speech using group delay processing and hmm based embedded reestimation." in *INTERSPEECH*, 2014, pp. 1648–1652.

[12] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[13] H. A. Murthy and B. Yegnanarayana, "Group delay functions and its applications in speech technology," *Sadhana*, vol. 36, no. 5, pp. 745–782, 2011.

[14] S. A. Shanmugam, "A hybrid approach to segmentation of speech using signal processing cues and Hidden Markov Models," M. S. Thesis, Department of Computer Science Engineering, IIT Madras, India, July 2015. [Online]. Available: http://lantana.tenet.res.in/thesis.php

[15] S. R. Vignesh, S. A. Shanmugam, and H. A. Murthy, "Significance of pseudo-syllables in building better acoustic models for indian english tts," in *ICASSP*, 2016, pp. 5620–5624.

[16] A. Baby, A. L. Thomas, N. L. Nishanthi, and T. Consortium, "Resources for Indian languages," in *CBBLR – Community-Based Building of Language Resources*. Brno, Czech Republic: Tribun EU, Sep 2016, pp. 37–43. [Online]. Available: https://www.iitm.ac.in/donlab/tts/index.php

[17] A. Baby, N. L. Nishanthi, A. L. Thomas, and H. A. Murthy, "A unified parser for developing Indian language text to speech synthesizers," in *International Conference on Text, Speech and Dialogue*, Sep 2016, pp. 514–521.

[18] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveign, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 34, pp. 187 – 207, 1999.