Dept. for Speech, Music and Hearing
Quarterly Progress and Status Report

A parallel speech analyzing system

Carlson, R. and Granström, B. and Hunnicutt, S.

journal: STL-QPSR
volume: 26
number: 1
year: 1985
pages: 047-062

KTH Computer Science and Communication

http://www.speech.kth.se/qpsr

B. A PARALLEL SPEECH ANALYZING SYSTEM *

Rolf Carlson, Björn Granström, and Sheri Hunnicutt

Abstract

Inspired by the parallel nature of human speech perception we have set out a framework to formulate and explore speech analysis/recognition models. The speech is represented as a continuous flow of information in multiple channels. A flexible notation for describing interactions within and between channels is included. Graphic representations of the processing could be created online to facilitate evaluations and experimantation. The system has been used to formulate the lower levels of speech analysis including spectral transformations, lateral inhibition, temporal onset/offset effects, and a variety of phonetic cue-detectors. Interactions on higher levels such as lexical access still need to be worked out.

Introduction

The speech wave carries its information distributed in time, frequency, and amplitude. This paper concerns a program, RECSYS, that is especially made to combine this information into more complex patterns, cues, and to present the results in an efficient manner. The program is controlled by a higher-level notation enhancing readability.

The work is heavily inspired by the feature notation, which has proved to be a powerful tool to use for phonetic descriptions and phonological transformations. A broader classification has also been explored in lexical search (Huttenlocher & Zue, 1984). Rather than using features in our approach we will use continuous parameters that represent cues.

The goal with our approach is to explore the descriptive power of cues, and to use multiple cues to analyze, classify, and segment the speech wave. A successful model will help us to better understand the acoustic representation of speech and can also be implemented as part of a speech recognition system.

Klatt (1977) gives a review of the speech recognition systems for large vocabularies developed in the ARPA project. The HARPY system uses a network of nodes to represent the precompiled knowledge. The recognition result will be the best path trough this network. This approach has been expanded to a model of lexical access (Klatt, 1980).

---

* also presented at the French-Swedish Seminar, Grenoble, April 22-24, 1985.

Recently, alternative approaches in cognition and vision have explored models based on simple units that interact in parallel networks (Hinton & Anderson, 1981; Hinton, Sejnowski, & Ackley, 1984). The work on models of neural networks has had a strong influence on this approach (Rosenblatt, 1961). The result is represented as the total activity in the network. These methods have now also been explored in speech research (Elman & McClelland, 1983).

Parallelism is the second, and most important motivation for our current approach. Speech recognition systems cannot be based on simple decisions involving few parameters and little or no complementary supporting cues. This model should make it possible to use diverse analysis mechanisms which can be simple but should work in a coordinated structure.

Before the method is presented, we will discuss some further ideas underlying the approach. The first is that speech is continuous and hence should be treated as such. This means that the process should not be limited to a stationary analysis of an utterance. We will regard the process as a pipeline that has, as input, the speech wave or rather a representation of it, such as spectral patterns. The output will be the result of both the analysis and the input data. The history of each analysis or transformation is kept in the pipeline as a short term memory and can be used for later corrections.

A basic criterion of the model is that it should be straightforward to include results of current research on the function of the peripheral auditory system, see the Proceedings from a recent symposium on this topic (Carlson & Granström, 1982). Using conventional signal processing techniques we have earlier tried some of the proposed transformations in the context of a speech recognition system (Blomberg, Carlson, Elenius, & Granström, 1984). Several effects have been reported that could be useful for transforming the incoming data. Lateral inhibition and onset/offset effects can, for example, be included to emphasize important events in the speech wave. It should be possible to formalize these effects in a simple manner.

In the following, we will present the program RECSYS with the aid of a number of examples.

RECSYS

The computer program RECSYS consists of a pipeline in which a spectral representation is used as input and stored in a number of input channels. At each sample time, all data in each channel are moved like a delay line and new input is stored in the beginning of the line. It is thus possible to study not only one but a sequence of spectral representations.

In Fig. 1a, such a sequence of spectral sections is plotted. As usual, each spectrum has been slightly displaced which gives a better visual presentation but uses considerable space, which is a drawback. In Fig, 1b, the plots have been moved back to an orthogonal representation but the shading has been kept. This is done by a simple multi-

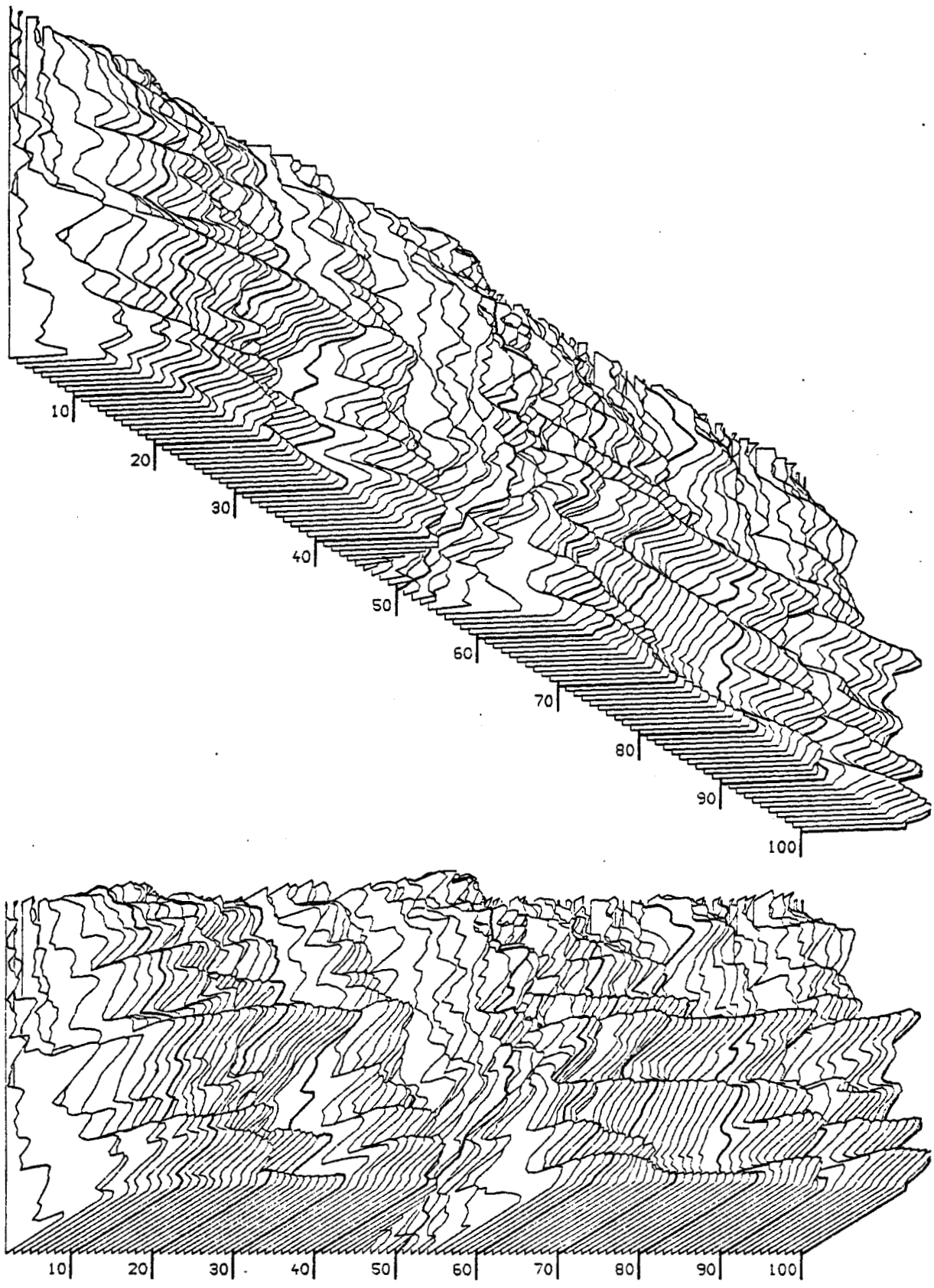


Fig. 1. Two representations of a sequence of spectral patterns.

plication of the x coordinate by a factor which makes the time axis horizontal. This seems to be a good representation. It keeps the "mountain structure," but takes a minimum of space. It can easily be aligned with other information. The angle of the amplitude axis can be changed by the user. This representation will be used in the following.

Units

Between each move in the pipeline a number of user-defined analyses takes place. We could regard each type of analysis as a "spider" with many legs standing on a matrix, the pipeline, with time and channels as parameters. Another illustration could be a cell that has connections to a certain number of elements in a matrix. Each connection can be activating or inhibiting. Such an analyzing structure is called a UNIT. The method is illustrated in Fig. 2. The mathematics in each unit can be simple, but groups of units form complex patterns. The user or model builder designs such a unit by writing a simple definition telling which elements in the matrix the unit is connected to, and the influence of the contents of the matrix elements on the unit's result. By this definition, a new active line in the matrix corresponding to the output of the unit is created and can be connected to other units. The result stored in this line is moved in synchrony with all data in the matrix. The program does a check on timing relations during compilation of the definitions and orders the calculations in such a way that the input to each unit is the current or delayed output from another. This is necessary since the program is implemented on a serial computer. The whole system is totally defined by the unit definitions and forms a parallell network.

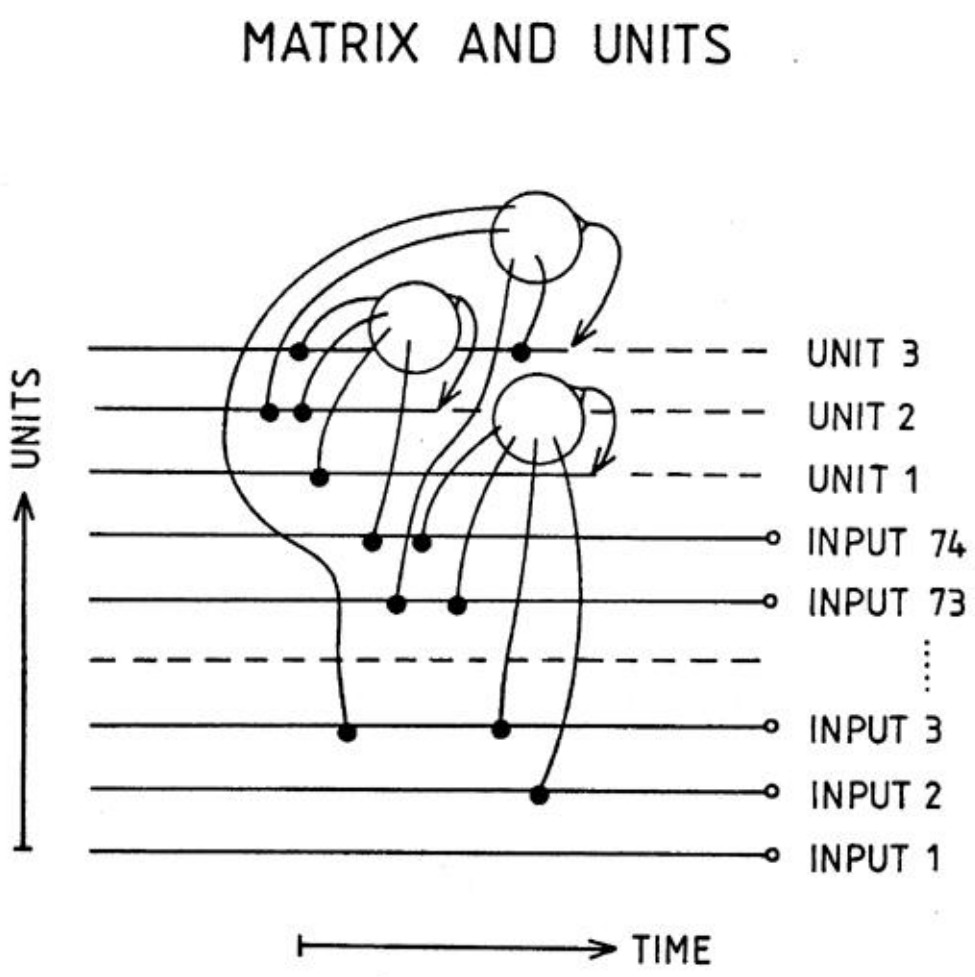
Each unit can be connected to all other positions in the matrix and detailed acoustic information can be combined with gross feature analysis. This will result in a system that has no explicit levels unless the user so wishes and expresses it by unit connections.

A unit can be very specific or have a general function. The earliest ones might create a new spectral representation. Another kind of unit could measure the spectral balance or movements of energy in the spectrum. As the system becomes more complex, several units can measure cues such as voice onset time or degree of aspiration.

A unit, at a higher level, can be a word-ending unit that stimulates a parts-of-speech unit to activate lexical entries also represented by units. Each such lexical unit can stimulate possible new lexical units. Whether the system can be used to represent knowledge at these high levels must be tested.

A network of units

In the following, a first attempt to use the proposed framework will be described. The examples given are only used to illustrate the method. We do not claim that the units chosen are optimal or even well thought out. Like the system as such, the notation is not fully developed. Therefore, it will not be explained in detail in our paper. Only



DEFINITIONS:

UNIT1=INPUT2+INPUT3+INPUT73+INPUT74

UNIT2=INPUT74+UNIT1+UNIT2+UNIT3

UNIT3=INPUT3+INPUT73+UNIT2+UNIT3

Fig. 2. Illustration of three UNITS connected to the elements in the Matrix.

some general remarks should be made. The calculation is performed in a serial order. At the starting point the RESULT is zero. It is changed like performing an evaluation on a simple pocket calculator. Some special functions have been used:

ADD<x,y> means RESULT=x+y

SUB<x,y> means RESULT=y-x

MIN<x,y> puts RESULT to the lowest value of x and y

MAX<x,y> puts RESULT to the highest value of x and y

.OR.(x) compares the RESULT and changes the RESULT to x if it is lower than x

.AND.(x) compares the RESULT and changes the RESULT to x if it is higher than x

We will, in the following, describe the system by going through a number of definitions, their functions, and their results.

Spectral shaping - change of the frequency scale

The input in our examples are 74 sample FFT spectra. The hamming window is 20 ms and the time interval between each spectrum is 10 ms. The sampling frequency is 16 kHz. Each 10 ms a new spectra is read into the matrix and stored as a column in the matrix. The preceding spectra are moved one step in the delay line. The input units are called INP0 to INP73 corresponding to the first 74 lines in the matrix. The first task will be to transform the input into a bark-like representation. This is easily done by the definitions in Table 1. The first definition tells the system that the value in the INP2 unit, at time 0, is taken and stored in the line in the matrix corresponding to unit B1. All text within double quotes ("....") is regarded as comments and the definition of the unit ends with a semicolon (;). We have now created 17 new units according to the definitions in Table 1; these result in a transformed spectral representation. Fig. 3a gives an example of the input spectrum and Fig. 3b shows the transformed spectrum of the first part of the Swedish sentence: " På utflykten grillade barnen glatt korven de fått med hemifrån."

Types and spectral shaping in time

The next step in our exemple is to shape this transformed spectrum, emphasize the onsets, and reduce the spectral level in the valleys. We will now introduce a new function of the system. It is possible to create a generalized unit without a connected line in the matrix. This is called a TYPE since it defines a type of action. The TYPE is distinguished from a unit by the preceding $ sign and has one element in

```
" 188 HZ" B1 =(INP2(0));
" 294 HZ" B2 =(INP3(0));
" 375 HZ" B3 =(INP4(0));
" 468 HZ" B4 =(INP5(0));
" 562 HZ" B5 =(INP6(0)+INP7(0))/2;
" 750 HZ" B6 =(INP8(0)+INP9(0))/2;
" 937 HZ" B7 =(INP10(0)+INP11(0))/2;
"1125 HZ" B8 =(INP12(0)+INP13(0))/2;
"1312 HZ" B9 =(INP14(0)+INP15(0)+INP16(0))/3;
"1594 HZ" B10=(INP17(0)+INP18(0)+INP19(0))/3;
"1875 HZ" B11=(INP20(0)+INP21(0)+INP22(0)+INP23(0))/4;
"2251 HZ" B12=(INP24(0)+INP25(0)+INP26(0)+INP27(0))/4;
"2625 HZ" B13=(INP28(0)+INP29(0)+INP30(0)
              +INP31(0)+INP32(0))/5;
"3094 HZ" B14=(INP33(0)+INP34(0)+M(INP35(0)
              +INP36(0)+INP37(0)+INP38(0))/6;
"3656 HZ" B15=(INP39(0)+INP40(0)+INP41(0)
              +INP42(0)+INP43(0)+INP44(0))/6;
"4219 HZ" B16=(INP45(0)+INP46(0)+INP47(0)
              +INP48(0)+INP49(0)+INP50(0)+INP51(0))/7;
"4884 HZ" B17=(INP52(0)+INP53(0)
              +INP54(0)+INP55(0)+INP56(0)
              +INP57(0)+INP58(0)+INP59(0)
              +INP60(0)+INP61(0)+INP62(0)
              +INP63(0)+INP64(0))/13;
"6000 HZ "
```

TABLE 1. Transformation into bark-like bands.

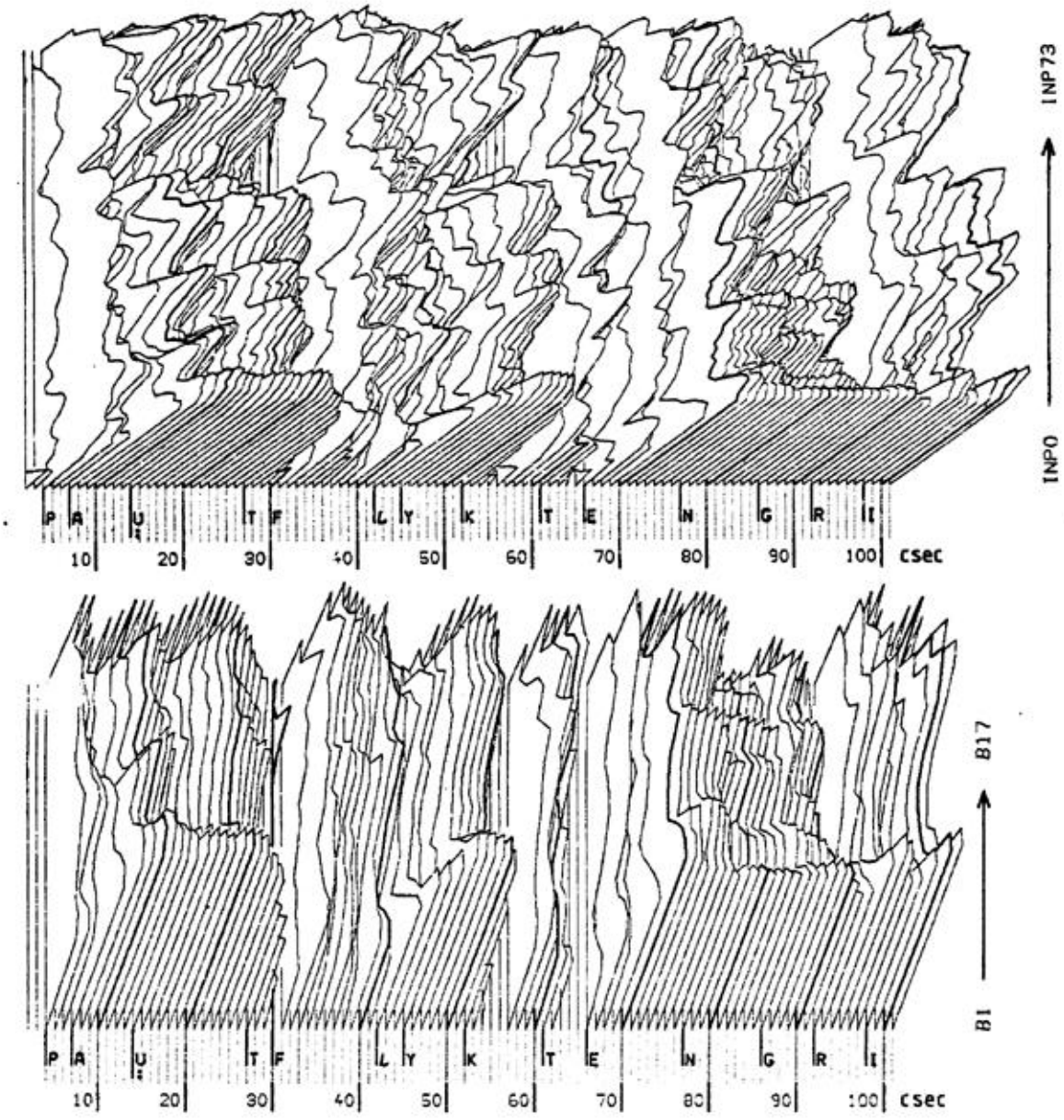


Fig. 3. Transformation into Bark-like bands. See Table 1.

the matrix as argument. This element specifies a center position of the type and is a reference point in the matrix. The elements in the type definition are referred to relative to this reference, by the notation M (line, time). Another way to describe the type is to regard it as a grid that could be put on top of the matrix, the position of the grid being defined by the reference point. As can be seen in Table 2, this type could be called or referred to by several units and the type defines the behavior of the whole group of units. The calling unit has, as argument, the center position that should be used by the type. This argument is added to the predefined positions of the type in the matrix. Each time a type is called, its definition is used as if it were part of the calling unit.

The first line in the table defines the type $INHIB. It takes the value from the matrix at the unit which is one step higher in frequency than the reference point at time 0. The type compares this value with the value one step below the reference point and keeps the highest value. The result is subtracted from the value at the reference point multiplied by 5. The result is then divided by 4 and added to the output from the type $INHIBT, which is described later in the text. Thus $INHIB compares the level in the two surrounding frequencies and uses the highest one to reduce the current level at the reference point. We have then created a function that enhances the high levels and reduces the low levels.

The following type, $INHIBT, has the function of enhancing onsets and offsets in time. It takes the difference in levels in the unit at the current time and the unit one step in time before. The negative difference is reduced by 10 and limited to a value between -30 and 0 by the .OR. and .AND. functions. The result is stored as an offset effect; a similar process will get a value for onset effects. The combined result is gained by summing the onset and offset effects.

The units BH1 to BH17 in Table 2 use these two defined types to create a new spectral representation which is shown in Fig. 4. Table 2 also includes some simple filtering often used in speech analysis. Note that the input to these filters is taken from the transformed spectral representation. The output can also be seen in Fig. 4.

Changes in levels

In this section, we will present a number of units that have the function of measuring changes in levels. If ordinary filtering is used, such as the units in Table 2, we get distortion effects when formants change frequency. This is especially pronounced when a formant is close to a band limit. If the task is to measure general changes, we want to eliminate these effects.

The two types, $DWN and $UP, presented in Table 3 are especially made to disregard formant changes. The $DWN type measures the level drop as the difference between the level of a unit at a particular time slot and the maximum level of that unit and its two adjacent neighbors one time slot later. If a formant changes frequency and reduces the

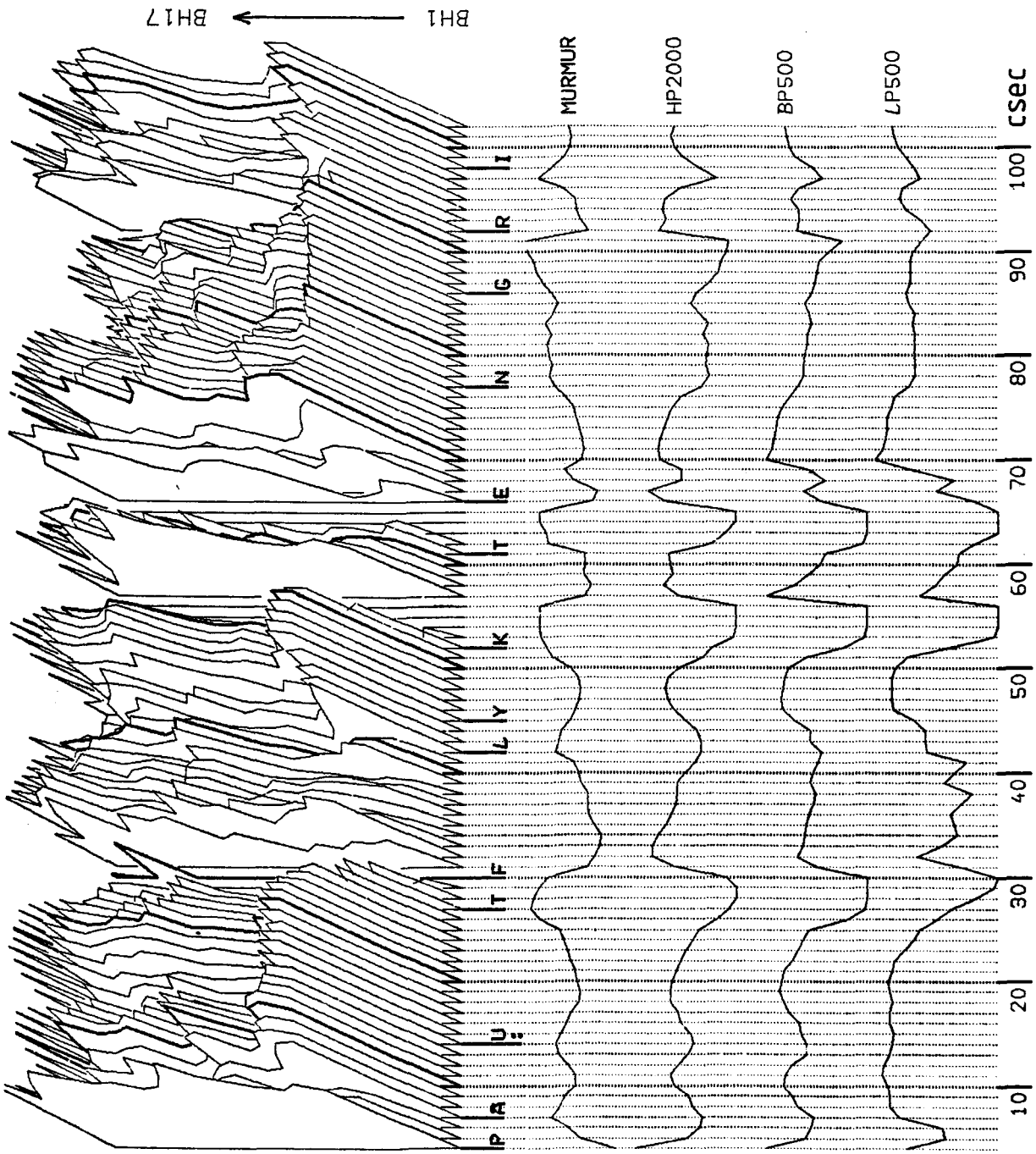


Fig. 4. Inhibition and some common filtering. See Table 2.

```
$INHIB=ADD<SUB<(M(1,0).OR.(M(-1,0))),(M(0,0)*5)>/4
              , $INHIBT(M(0,0))>;

$INHIBT=ADD<(M(0,0)-M(0,-1))+10.OR.(-30).AND.(0)
        ,(M(0,0)-M(0,-1))-10.OR.(0).AND.(30)> ;

BH1 =ADD<(B1(0)*5-B2(0))/4, $INHIBT(B1(0))>;
BH2 =$INHIB(B2(0));
BH3 =$INHIB(B3(0));     BH4=$INHIB(B4(0));
BH5 =$INHIB(B5(0));     BH6=$INHIB(B6(0));
BH7 =$INHIB(B7(0));     BH8=$INHIB(B8(0));
BH9 =$INHIB(B9(0));     BH10=$INHIB(B10(0));
BH11=$INHIB(B11(0));    BH12=$INHIB(B12(0));
BH13=$INHIB(B13(0));    BH14=$INHIB(B14(0));
BH15=$INHIB(B15(0));    BH16=$INHIB(B16(0));
BH17=ADD<(B17(0)*5-B16(0))/4, $INHIBT(B17(0))>;

"LOW PASS 0-500 HZ"      LP500=(BH1(0)+BH2(0)+BH3(0)+BH4(0))/4;

"BAND PASS 500-2000 HZ"       BP500=(BH5(0)+BH6(0)+BH7(0)+BH8(0)
                              +BH9(0)+BH10(0)+BH11(0)+BH12(0))/8;

"HIGH PASS 2000-6000 HZ" HP2000=(BH13(0)+BH14(0)
                         +BH15(0)+BH16(0)+BH17(0))/5;

"HIGH PASS 500-6000 HZ"  HP500=(BH5(0)+BH6(0)+BH7(0)+BH8(0)
                         +BH9(0)+BH10(0)+BH11(0)+BH12(0)+BH13(0)+
                         BH14(0)+BH15(0)+BH16(0)+BH17(0))/13;

" SIMPLE CUES "          UNVOIC=(HP500(0)-LP500(0)) ;

                         VOIC=(LP500(0)-HP2000(0)) ;

                         MURMUR=(LP500(0)-BP500(0)-2*HP2000(0));
```

TABLE 2. Shaping of spectrum with inhibition and time sharpening.

```
" MEASURE LEVEL DROP BUT DISREGARD FORMANT CHANGE "
$DWN=SUB<((M(0,0)),(M(-1,1)).OR.(M(0,1)).OR.(M(1,1))>.AND.(0);

FDWN2=$DWN(BH2(0));      FDWN3=$DWN(BH3(0));
                        ...........
FDWN14=$DWN(BH14(0));    FDWN15=$DWN(BH15(0));
" MEAN LEVEL DROP "
FDWN=" (FDWN2(0)+FDWN3(0)+FDWN4(0)+" (FDWN5(0)
                +FDWN6(0)+FDWN7(0)+FDWN8(0)+FDWN9(0)
                +FDWN10(0)+FDWN11(0)+FDWN12(0)+FDWN13(0)
                +FDWN14(0)+FDWN15(0))/11;
" MAXIMAL LEVEL DROP "
FDWNM(-4))=(FDWN2(0))).AND.(FDWN3(0)).AND.(FDWN4(0)).AND.(FDWN5(0))
 .AND.(FDWN6(0)).AND.(FDWN7(0)).AND.(FDWN8(0)).AND.(FDWN9(0))
 .AND.(FDWN10(0)).AND.(FDWN11(0)).AND.(FDWN12(0)).AND.(FDWN13(0))+5>;

" MEASURE LEVEL RISE BUT COMPENSATE FOR FORMANT CHANGE "
$UP=SUB<(M(-1,-1)).OR.(M(0,-1)).OR.(M(1,-1)),(M(0,0))>.OR.(0);

FUP2=$UP(BH2(0));        FUP3=$UP(BH3(0));
                        ..........
FUP14=$UP(BH14(0));      FUP15=$UP(BH15(0));

" MEAN LEVEL RISE "
FUP=(FUP5(0)+FUP6(0)+FUP7(0)+FUP8(0)+FUP9(0)
                +FUP10(0)+FUP11(0)+FUP12(0)+FUP13(0)
                +FUP14(0)+FUP15(0))/11;
" MAXIMAL LEVEL RISE "
FUPM=(FUP2(0)).OR.(FUP3(0)).OR.(FUP4(0)).OR.(FUP5(0))
 .OR.(FUP6(0)).OR.(FUP7(0)).OR.(FUP8(0)).OR.(FUP9(0))
 .OR.(FUP10(0)).OR.(FUP11(0)).OR.(FUP12(0)).OR.(FUP13(0))-5 ;
```

TABLE 3. Estimation of changes in levels.

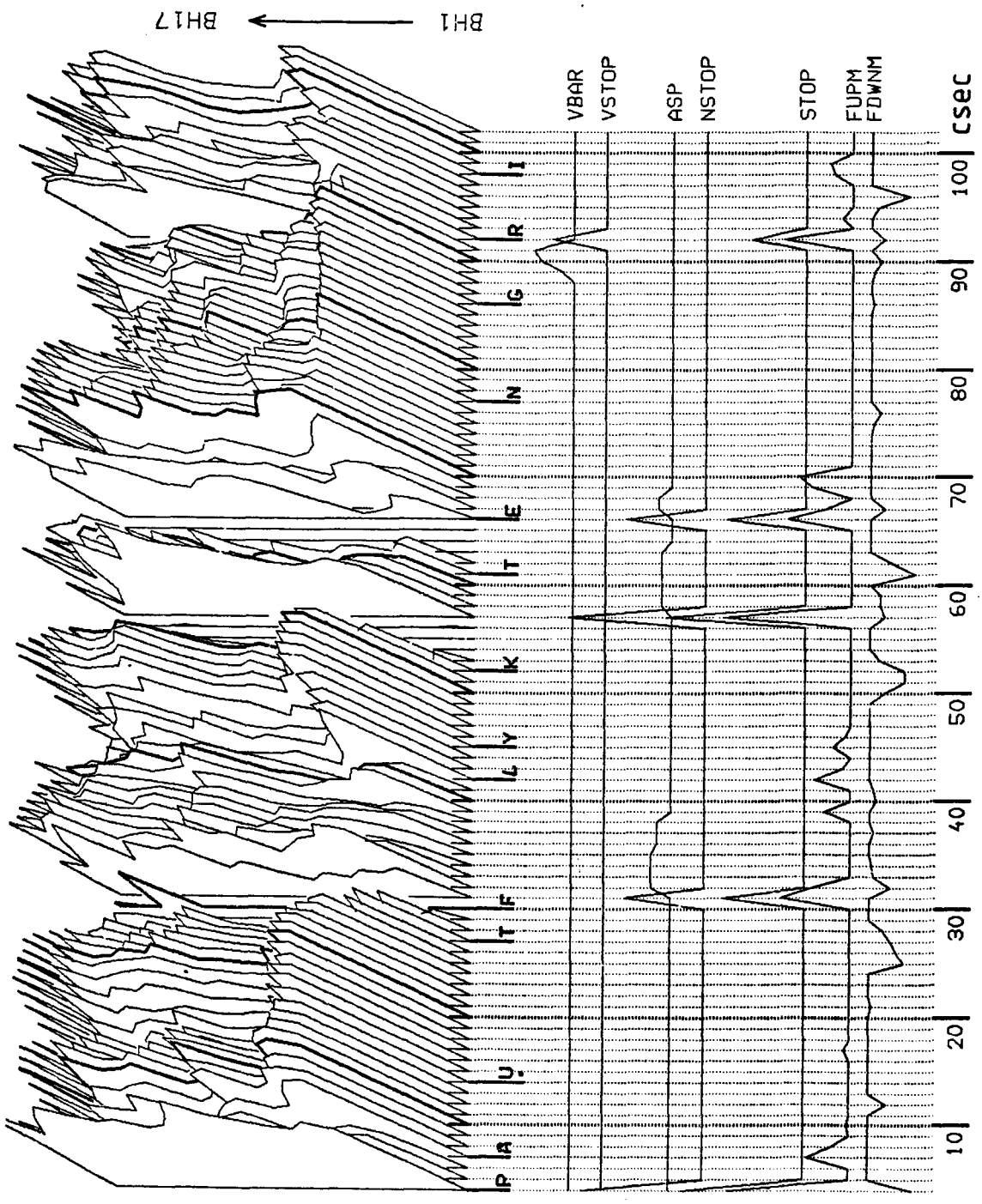


Fig. 5. Output from the STOP units. See Table 3 and Table 4.

level in one unit, the level will rise in the next unit and the output of $DWN will be kept close to 0. If the level changes more globally (in several units), the $DWN type will give a high negative response. The units FDWN1 to FDWN15 measure the level drops in all frequency bands. The FDWNM unit gives the maximal level drop and the unit FDWN gives the mean. The $UP type and units have a similar function but measure level rise rather than level drop.

Stop cues

We have now created a base for the search for more phonetically oriented cues. Stop cues have been chosen as examples since they are simple but not trivial.

The STOP unit in Table 4 adds up information from several units. The maximal value in two level increases measured by FUPM (FUPM(0) and FUPM(1)) gives a positive contribution. A level drop at time 1 could be a cue for an end of stop explosion so the FDWNM is included in the definition (FDWNM(1)). The unit is inhibited if the current rise is 0. This is implemented with the MIN function. (MIN<....., (FUPM(0)*10)>.) Preceding high pass energy in HP2000 (HP2000(-1) and HP2000(-2)) is used to inhibit a positive response from the unit. Furthermore, the unit will not give a positive response if it already gave a positive response one step earlier (-STOP(-1)).

The stop is regarded as voiced (the VSTOP unit) if it has a murmur, but the output is inhibited if the explosion has a high level drop after it. The unvoiced stop (unit NSTOP) is simply the remaining component of the STOP unit response. The output from the units in Table 4 can be seen in Fig. 5. The units ASPS and ASP give response to aspiration and the VBAR to voice bar.

In this section we have tried to illustrate how different units can be combined in complex structures despite the simple behavior of each unit. Many cues can be used for a special purpose.

```
STOP =MIN<(FUPM(0).OR.(FUPM(1))-FDWNM(1)),(FUPM(0)*10)>
            -HP2000(-1)-HP2000(-2)-STOP(-1)  ;

VSTOP=(MURMUR(-1))*10 .AND.(STOP(0)) (2*FDWNM(1)) ;

NSTOP=(STOP(0)-VSTOP(0)*2) ;

ASPS=MIN<(NSTOP(-2)+NSTOP(-1)) , (-FDWNM(0))>;

ASP=MIN<(UNVOIC(0)-FDWNM(0)-FDWNM(-1)-FUPM(0)) ,
          (ASP(-1))>.OR.(ASPS(0));

VBAR1=(MURMUR(0)).AND.(VSTOP(1)) ;

VBAR2=(MURMUR(0)).AND.(VBAR1(1)).OR.(VBAR1(0));

VBAR=(MURMUR(0)).AND.(VBAR2(1)).OR.(VBAR2(0));
```

TABLE 4. Search for cues.

Conclusion

We have presented an analysis system called RECSYS and some of its features. Only to some degree does it fulfill our wishes. Some problems that we have not been able to solve include decision processes and spectral normalization. If we regard the system as a speech recognition system, it is not fully clear how a final result should be presented. When should we activate an output of the most likely utterance? Decisions have purposely been avoided in the system, letting each unit have a continuous output. The result is so far only a number of units that have different levels of activity.

Another problem is the time alignment. How should we adjust for speech tempo? One possibility would be to run the system at several levels and let the units decide what the content and timing should be for the next level. This has some definite drawbacks. The flexibility of being able to test all activities in the pipeline at each time and level is lost. We have no good answer to this problem.

A final problem, and perhaps at present the most serious one, is the notation and its complexity. Each definition is more or less a mathematical formula. It is very far from a common description of linguistic knowledge. This could to some degree be clarified by good unit names, but it might be difficult to get a good overview of the system. A more general notation would be preferred, but how should it look?

Despite the mentioned problems in the current approach, we want to argue that the program RECSYS can be useful for speech research. It will force us to formulate our current knowledge in rules or, in our case, in unit definitions that are testable.

References

Blomberg, M., Carlson, R., Elenius, K., & Granström, B. (1984): "Auditory models in isolated word recognition," Proc. IEEE Int.Conf. on Acoustics, Speech and Signal Processing, IEEE Catalog No. 84CH1945-5, 2, 17.9.2, San Diego.

Carlson, R. & Granström, B. (Eds.) (1982): The Representation of Speech in the Peripheral Auditory System, Elsevier/North Holland Biomedical Press, London.

Elman, J.L. & McClelland, J.L. (1983): "Exploiting lawful variability in the speech wave," Paper presented at the Symposium on Invariance and Variability in Speech Processes, MIT (to be published).

Hinton, G.E. & Anderson, J.A. (Eds.) (1981): Parallel Models of Associative Memory, N.J. Erlbaum.

Hinton, G.E., Sejnowski, T.J., & Ackley, D.H. (1984): "Boltzmann machines: constraint satisfaction networks that learn," Technical report CMU-CS-84-119, Carnegie-Mellon University.

Huttenlocher, D.P. & Zue, V.W. (1984): "A model of lexical access from partial phonetic information," Proc. IEEE Int.Conf. on Acoustics, Speech and Signal Processing, IEEE Catalog No. 84CH1945-5, 2, 26.4.1, San Diego.

Klatt, D.H. (1977): "A review of the ARPA speech understanding project," J.Acoust.Soc.Am. 62, pp. 1345-1366.

Klatt, D.H. (1980): "Speech perception: a model of acoustic-phonetic analysis and lexical access," in Cole (Ed.), Perception and Production of Fluent Speech, Hillside, N.J. Erlbaum, pp. 243-288.

Rosenblatt, F. (1961): Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan, Washington D.C.