IX. SPEECH COMMUNICATION[*]

Academic and Research Staff

Prof. K. N. Stevens
Prof. M. Halle
Prof. W. L. Henke
Prof. A. V. Oppenheim
Dr. Mary C. Bateson
Dr. Margaret Bullowa
Dr. A. W. F. Huggins
Dr. D. H. Klatt
Dr. Paula Menyuk
Dr. T. H. Shriner
Dr. J. J. Wolf
Kay Atkinson-King

Graduate Students

R. E. Albright
M. Bilofsky
A. J. Goldberg
J. D. Oetting
J. S. Perkell
J. E. Richards
R. M. Sachs
F. F. Sellers
V. C. Shields, Jr.
V. W. Zue

A. CHILDREN'S PERCEPTION AND IDENTIFICATION OF STRESS CONTRASTS

Children are believed to perceive and produce stress and other prosodic features of language very early in life. Yet, we have little knowledge of what they actually know about rules of stress at various ages, or by what age this learning process has been completed. There has been very little research examining the acquisition of any of the stress rules in a language like English. This experiment was designed to determine children's competence at different ages with regard to one aspect of American English stress – that of the perception and identification of two types of phrases which, in the adult language, are minimal pairs with respect to stress. The first type is a compound noun which has a primary-secondary stress pattern, as seen in the example <u>gréenhouse</u>, a hothouse for flowers. The second type is a noun phrase comprising an adjective followed by a noun. This phrase has a secondary-primary stress pattern, as seen in the example <u>green hóuse</u>, a house painted green.[1]

1. Procedure

A total of 160 children in Kindergarten through Grade 8 in school, whose ages were from 5 to 13 years, were the subjects in this study. An additional 18 adults were tested in an identical manner for comparative purposes. All subjects were native speakers of English who were residents of the Boston area.

A stimulus tape was recorded of the experimenter producing 6 pairs of phrases that would be familiar to children and which could be pictured. These pairs were the following.

---

[*] This work was supported in part by the National Institutes of Health (Grant 5 RO1 NB04332-08), and in part by the U.S. Air Force Cambridge Research Laboratories, Office of Aerospace Research, under Contract F19628-69-C-0044.

Compound Nouns	Adjective Plus Noun
greenhouse	green house
redhead	red head
highchair	high chair
blackboard	black board
Red Sox	red sox
hot dog	hot dog

Each of the 6 pairs was presented 5 times making a total of 30 items, of which 15 were compound nouns and 15 were phrases with adjective plus noun. The 30 stimuli were arranged in 5 sets of 6 items in such a way that one member of each of the 6 contrasting pairs was presented once in each set. The order of phrases within each set was random.

All phrases to be tested were introduced to the children before the stimulus tape was played. The experimenter showed a picture of one item of the minimal pair, explained what it was, and then named it. This same procedure was repeated for the picture of the other member of the pair. The pictures and pronunciations of each pair were contrasted several times to maximize the differences. Children were given answer sheets containing sets of pictures of the minimal pairs. The order of these pictures matched the order of stimuli on the tape. Children then heard the stimulus tape and were asked to identify what they heard each time by marking the appropriate picture. For example, if the stimulus were hót dog, children would have to choose between a picture of a frankfurter on a bun and a dog panting in the hot sun. Each item was presented twice before children were asked to make a judgment.

Listeners were tested in small groups rather than individually, since an earlier pilot study had indicated that there were no differences in performance between subjects tested individually and subjects tested in small groups. Each group had approximately 10 children and one or more adults.

In order to test whether or not the experimental results reflected linguistic skill rather than just an ability to perform a mechanical task, a separate test was given to 51 of the youngest children who were in Kindergarten and Grade 1. The same stimulus tape was played to the children, but instead of making their responses by choosing between pictures on their answer sheets which depicted minimal pairs, they chose between pictures of maximally different items. Thus, for example, if the stimulus were hót dog, the children would have a choice of a frankfurter on a bun and another item such as a greenhouse. The results of this task comprehension test showed that only 2 kindergartners were not able to perform the task. These 2 children were subsequently eliminated from the study. The average percentage of correct responses for the remaining 49 subjects was 97% correct for Kindergartners and 100% correct for First Graders. On the basis of these results it is evident that children can perform the task.

2. Results

Results from the stress identification test indicate that the ability to correctly identify these minimal pairs increases with age. Figure IX-1 shows the percentage of correct responses plotted against grade and age. Measures of standard deviation are also given for each group. The grade groups on the abscissa are prefixed by "mid-" or "pre-" because some children were tested at the middle of the school year and some were tested at the beginning of the school year.[2] The ages given are average ages for the group of children in each grade. This figure indicates that Kindergartners are only producing 49% correct responses, while children in Grades 6, 7, and 8, as well as adults, produce responses approaching 100% correct. Between these extremes there is a consistent developmental trend with the percentage of correct responses increasing with age and grade.[3]

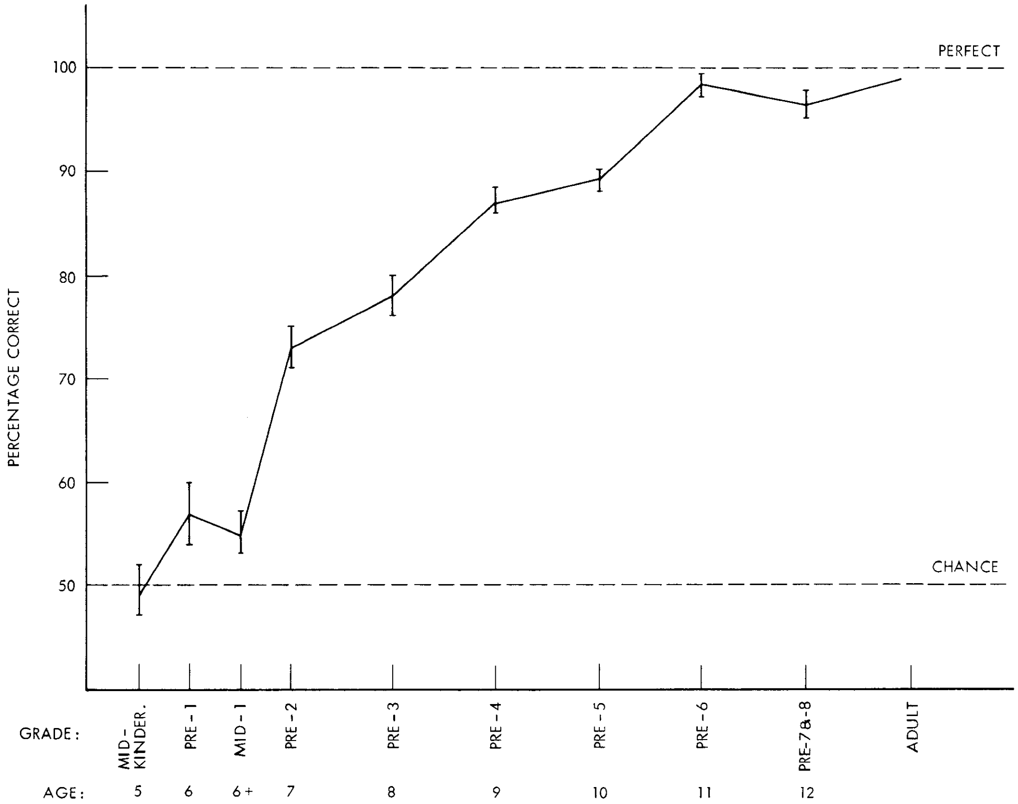


Fig. IX-1. Percentage of correct responses according to grade and age with standard deviation plotted for each point.

Chi-Square analysis indicates that the Kindergarten group is not scoring significantly more correct than incorrect responses; the Grade 1 group's number of correct responses is significantly greater than chance at the 0.05 level. Grade 2 and all higher grades produced significantly more responses than chance at the 0.001 level of significance. The greatest developmental change is between age 5, where the group does not perform significantly above chance, and age 7, where the group is producing 73% correct responses.

We might ask whether a child acquires the ability to identify these contrasts all at once or whether he seems to develop this skill gradually. The results shown in Fig. IX-2 shed some light on this question. This figure shows the percentage of children who individually scored a total number of correct responses significant above chance as determined by chi-square analysis. Two levels of significance are shown. The solid line indicates the proportion of children who individually scored over 70% correct responses (better than chance at $p < 0.05$), and the dashed line indicates the proportion who scored over 85% correct responses (better than chance at $p < 0.001$). In Grades 2 through 5 we observe a considerable variability between the percentage of children whose individual scores reached the two levels of significance. This variability

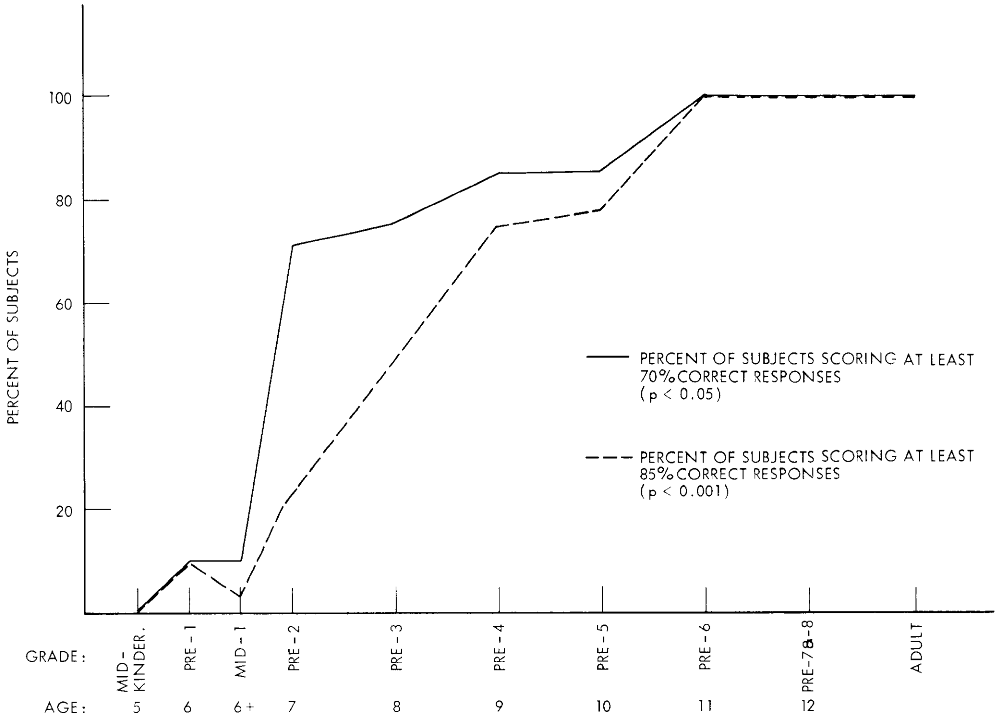


Fig. IX-2. Percentage of subjects with individual scores better than chance at two significance levels. The solid line indicates the proportion of children who individually scored over 70% correct responses (better than chance at $p < 0.05$), and the dashed line indicates the proportion who scored over 85% correct responses (better than chance at $p < 0.001$).

suggests a gradual learning process. For example, in Grade 2, where the greatest difference occurs, approximately three-fourths of the children in the group had a score of at least 70% correct responses ($p < 0.05$), but only one-fourth of these children with scores of over 70% correct had a score as high as 85% or more correct responses ($p < 0.001$). Hence, one-half of the Grade 2 group performed at a level high enough to indicate that they had some proficiency with the identification skill, because they could correctly identify 70% of the stimuli, but they had not mastered it completely, because they could not correctly identify as high as 85% of the stimuli. Thus the gap between these two levels of significance as shown in Fig. IX-2 represents the percentage of children who have begun to acquire the skill of identifying these phrases, but who are not yet able to respond correctly with a high degree of consistency.

Figure IX-2 also shows that, although the general picture is one of increasing ability with age, there is some individual variation in Grades 1 through 5. Thus, a few children as early as 6 years old performed almost as well as adults, and a few children as late as 10 years old did as poorly as 5-year olds. In Kindergarten, however, there were no children who scored significantly better than chance, and in Grades 6 and above all

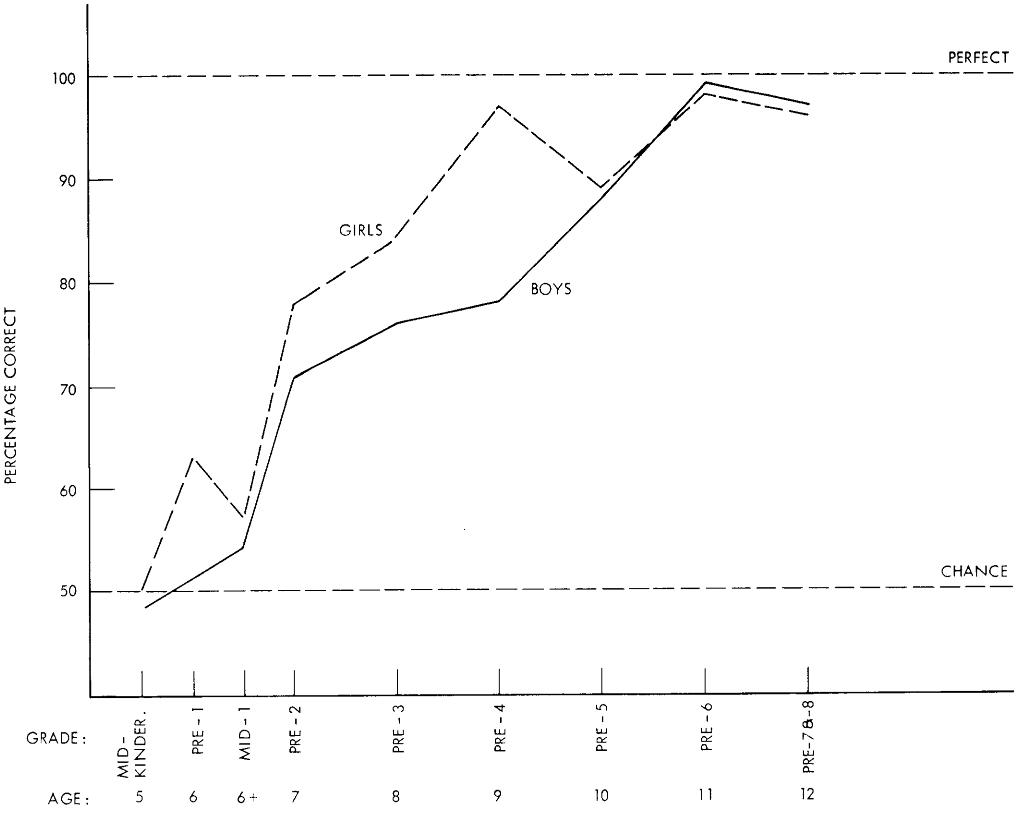


Fig. IX-3. Percentage of correct responses according to grade and sex.

subjects scored better than chance at the 0.001 level.

The general results also show sex differences at certain ages. Figure IX-3 shows the average percentage of correct items plotted against age and grade according to sex. There is a consistent trend for girls to score better than boys in grades below Grade 6. These differences are significantly in favor of the girls for Grades Pre-1, 3, and 4. Girls produce significantly more correct than incorrect scores in Grade 1, but boys do not do this until Grade 2. Also, girls approach the 100% level of correct responses earlier in the age range than boys.

3. Discussion

The results of this experiment suggest that the developmental trend in the ability to identify these noun phrases correctly and consistently is mainly due to an increasing knowledge of the rules of stress dealing with the distinction between these phrases, and/or an ability as listeners to use these rules. These stress rules are learned mainly as a function of age; and, although an awareness of what we call stress appears to be acquired very early in life, these particular rules are not learned until relatively late in the language acquisition process. Girls, generally, acquire proficiency with these rules at slightly earlier ages than do boys, and there is also some individual variation, but in general, the picture is a developmental one. A child at age 5 will not yet have acquired these rules; a child at age 11 will have acquired them, and the older a child is between these age extremes, the more likely it is that he will have learned the rules and will be able to use them consistently.[4]

The experimental data also suggest that the ability to use the rules consistently is acquired gradually. A child who can disambiguate the minimal pairs used in the experiment 70% of the time must have internalized the rule. Since this child still has 30% incorrect responses, however, he is not yet applying the rule consistently. This suggests that at this stage the rule is not an obligatory rule, but an optional one. An additional longitudinal study with individual children would be helpful to confirm this conclusion.

Since prosodic rules of various kinds (e.g., intonation of declaratives vs questions) apparently play a role in earlier periods of language acquisition, we might ask why the stress rules used to differentiate compound nouns from noun phrases are mastered relatively late in the language development process, and why they sometimes seem to be optional rules in the early stages of their being learned. The main reason which might be suggested is that the phonetic patterns which differentiate these phrases may not always be realized in the acoustic signal. This is true even of the adult model when the rules are clearly possessed. Adults may not always make the distinction in production because of various performance factors, and because the semantic content of the sentence usually carries enough information to make comprehension possible without the help of this acoustic distinction. Or, perhaps more importantly, the lack of phonetic differentiation between these phrases may result from interference by constrastive or emphatic stress shift. For example, although the stress normally would fall on the noun in the noun phrase *green hóuse*, describing a house that has been painted green, in the contrastive sentence, "It was a gréen house, not a brówn house," the contrastive stress rule would create a shift of stress to the adjective, making the noun phrase acoustically closer to the compound noun *gréenhouse*. Factors like the commonly occurring contrastive stress rule, therefore, might confuse the child and make the learning of the compound noun and noun phrase stress rules more difficult, or might make the rules appear optional. This would help to explain why children master these stress rules relatively late in the language acquitision process.

The late acquisition of these stress rules raises interesting questions about the developmental relationship between rules of stress and rules of syntax. There is a close connection between these two aspects of language, but we do not know whether knowledge of stress is prerequisite to or an aid in the acquisition of syntax or vice versa; perhaps both are acquired simultaneously. It is generally agreed that children have syntactic competence with compound nouns and noun phrases by age 5 or earlier. Thus, the experimental results which show that children do not master the stress rules that distinguish these phrases until much later suggest that the acquisition of the syntactic rules is prior to the acquisition of the stress rules. This indicates that the stress rules cannot be an aid to the acquisition of the syntax of these phrases. This type of priority may not exist with other prosodic and syntactic aspects of language, but it does appear to exist in the case of noun compounds and noun phrases.

I would like to thank Mr. Peter M. Close, Director of the M.I.T. Day Camp, and Miss Norene Casey, Director of the Bartlett School, for providing subjects for this study, and also Dr. A. W. F. Huggins for advice concerning the statistical analysis.

Kay Atkinson-King

Footnotes

1. Several linguists have discussed this contrast between compound nouns and noun phrases, and recently the rules involved have been formalized in Noam Chomsky and Morris Halle, *The Sound Pattern of English* (Harper and Row, New York, 1968).

2. Grades Pre-7 and Pre-8 were combined because they were both smaller than the other grade groups, and showed no significant differences in performance.

3. There are apparently two minor exceptions to this upward trend in performance. These are the inversions that occurred between the two groups of Grade 1 and among Grade 6 and Grades 7 and 8. The difference between the two groups of Grade 1 is not significant. It is due primarily to the exceptional performance of one girl in Pre-1, which was a smaller group than Mid-grade 1. Pre-1 had only 10 subjects, whereas Mid-grade 1 had 30 subjects. The slight dip occurring at Grades 7 and 8 is also not significant, and is probably due to the small number of subjects in Grades 6 through 8.

4. This experiment, of course, reveals nothing about children's use of stress rules in producing these minimal pairs. Further experiments are now being conducted to study the production aspect of these phrases.

B. AUTOMATIC ANALYSIS-BY-SYNTHESIS OF VOWELS

Recent work in characterizing speech samples for the recognition of the speaker[1] has indicated a need for a rapid, reliable facility for analyzing vowel spectra in terms of formant frequencies. Toward this end, a program for manual and automatic analysis-by-synthesis of vowels (VABS), has been implemented on the PDP-9 computer facility of the Speech Communication Group. This program is an extension of the analysis-by-synthesis techniques originally developed on the TX-0 computer.[2-4]

In an analysis-by-synthesis procedure, a spectrum based on hypothesized formant locations is compared with the spectrum of the vowel to be analyzed, and the hypothesized parameters are varied until the two spectra match. The process of varying these parameters may be done by the experimenter in an interaction with the analysis-by-synthesis program or by a strategy subprogram.

The analysis-by-synthesis program is designed to be used with a spectrum analyzer and fundamental frequency extractor. It is specifically written for the equipment shown in Fig. IX-4, but in principle it could be modified to function with any functionally equivalent configuration. At present, spectrum analysis is performed by a 36-channel filter bank covering 150-7025 Hz, and fundamental frequency is estimated by lowpass-filtering the speech above the first harmonic and measuring the intervals between zero crossings.

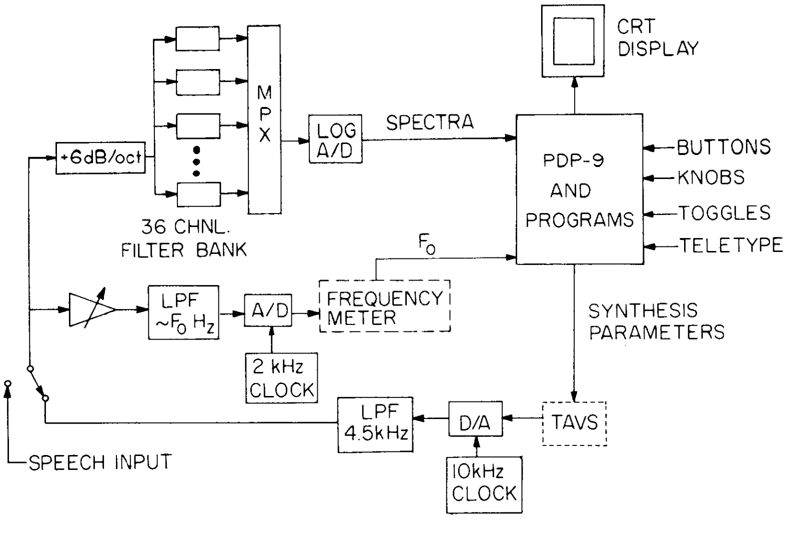


Fig. IX-4. The computer facility as configured for spectrum analysis and vowel analysis-by-synthesis.

The synthesis phase of the vowel analysis-by-synthesis is accomplished by generating a short segment of synthetic vowel and analyzing it with the 36-channel spectrum analyzer. This synthesis is represented by the lower branch in Fig. IX-4. The vowel generation is performed by a five formant cascade 10-kHz sampled data vowel synthesis program which was adapted from TASS, a speech synthesizer written by W. L. Henke.[5] The duration of the synthetic vowel need only be long enough that the filter band outputs reach steady-state values; 30 ms is sufficient for this.

The method of spectrum synthesis is the essential difference between this and the former analysis-by-synthesis implementation. In the original TX-0 version, the synthetic spectrum was computed in two stages. First the log magnitude of the Fourier transform of the vowel was computed at 100-Hz intervals, corresponding to the harmonics for an assumed fundamental frequency of 100 Hz. Then the response of the filter bank for such a vowel spectrum was computed. In the present version, this laborious calculation is eliminated by having the spectrum analysis performed by the same hardware that analyzes the speech input. This saving and the greater speed of the PDP-9 computer result in a much faster analysis. The analyzing filters below 3 kHz have bandwidths comparable to the spacing of the harmonics in normal speech. Hence the output of each filter is affected by the positions of the harmonics within its passband, as well as by the spectral envelope of the speech signal. In the present version, the synthetic spectrum can be derived from a signal with the same fundamental frequency as that measured in the original speech, rather than with an assumed fixed value. The synthetic spectrum is thus a more faithful replica of the spectrum it is intended to match. The main limitation on the accuracy of the synthesis lies in the quality of the glottal source approximation.

In the synthesizer, the glottal excitation is produced by exciting a digital glottal shaping filter with a train of impulses. The glottal filter used in the present version has complex conjugate poles at an equivalent s-plane position of $\sigma = 2\pi(125)\ \sec^{-1}$ and $\omega = \pm 2\pi(200)\ \sec^{-1}$.[6] The resulting pulse shape and spectrum envelope are shown in Fig. IX-5. Optionally, the poles may be moved along a line of constant Q through the point given above. This changes the time scale of the pulse shown in Fig. IX-5a and the break frequency of the spectrum in Fig. IX-5b, but not the shape of either. The synthesis parameter GBF (glottal break frequency) is the value of $\omega/2\pi$. This parameter was originally intended to compensate for interspeaker variations in glottal source spectrum, but it has not been proved appropriate. Consequently, a fixed value of GBF = 200 has been used.

1. Description of Program VABS

The analysis-by-synthesis procedure is under the control of up to three nested subroutines, depending on the degree of automation desired. The lowest level subroutine, VABSM, provides the most basic analysis-by-synthesis service. It generates the vowel waveform from a table of parameters given by the calling program, outputs it through a digital-to-analog converter, performs the spectrum analysis, and compares the

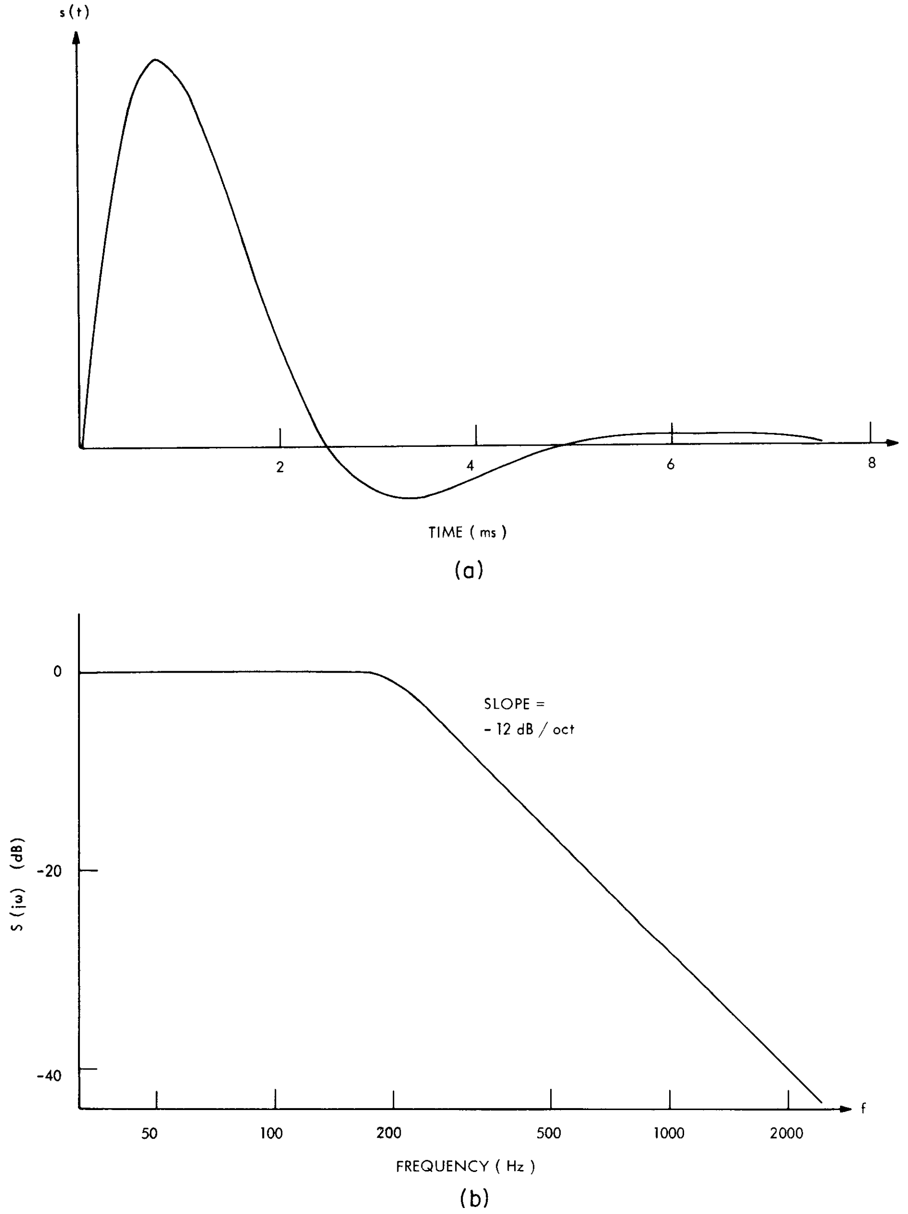


Fig. IX-5. Glottal source realization in the synthesizer. (a) Impulse response of the glottal shaping filter. (b) Glottal source spectrum envelope. The poles of this filter are located at $-2\pi(125) \pm j2\pi(200)\ \text{sec}^{-1}$.

spectrum of the synthetic vowel with a spectrum specified by the calling program. This subroutine is called by the higher level subroutines, and it can also be called directly by a user who wishes to vary the synthesis parameters manually. For instance, through programmed use of pushbuttons and knobs, any parameter can be varied by turning a knob. When running continuously, a manual analysis-by-synthesis program can produce and display approximately 4 iterations per second, so the user can observe the effect of each parameter variation directly on the cathode-ray tube display. The synthesis parameters to be specified are fundamental frequency (the measured value is normally used), the frequencies and bandwidths of the first four formants, and the glottal source break frequency.

The next higher level subroutine, VABSA, performs automatic analysis-by-synthesis, given a vowel spectrum to be matched and a parameter table of an initial approximation to the vowel. The strategy of varying the synthesis parameters was taken almost directly from Paul.[3,7] The parameters that are varied are the first four formant frequencies and the first three bandwidths. VABSA contains a flag that, if set by the calling program, will cause the subroutine to vary GBF every 4th iteration for lowest error. As we have stressed this has not been proved useful.

The error measure which the analysis-by-synthesis strategy attempts to minimize is the sum of squared differences, on a filter-by-filter basis, of the two spectra over the frequency range of interest. This range is a parameter given to the subroutine, typically the range 0-3175 Hz. The spectra are normalized for zero mean error before the squared error is calculated. The squared error as a function of the formant frequencies has local minima, corresponding to formants being placed in incorrect peaks of the spectrum. The initial approximation to the vowel to be analyzed need only position the formant frequencies close enough to their true positions that the strategy algorithms will not move a formant into an incorrect peak. The bandwidth values are not critical; they may be safely set to average values.

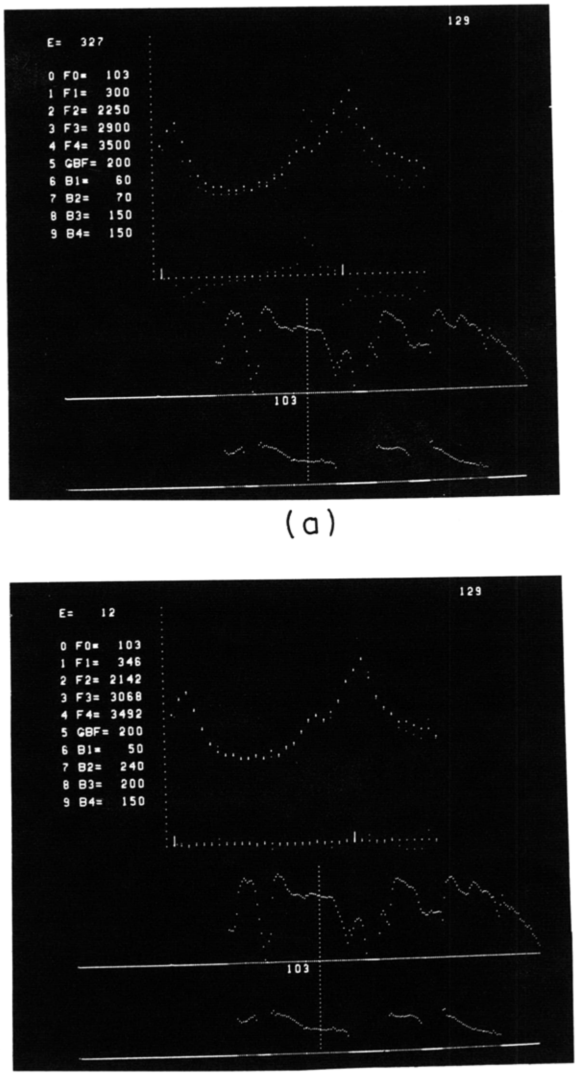
The program iterates the procedure of calculating new parameter values, synthesizing, and comparing with the spectrum to be matched until one of three conditions is satisfied: (a) the strategy algorithms produce no change in all parameters; (b) the iteration number is 15 and the best match has a squared error of less than 2 dB^2 per filter in the specified range, or (c) the iteration number is 20. The program performs approximately 5 iterations per second. After the analysis is completed, the parameter table in the calling program is overwritten with the new values of the parameters.

The third subroutine entrance, VABSI, arrives at an initial approximation and then effectively calls VABSA. VABSI requires a separate subprogram comprising stored spectra and values of the first three formants associated with them. The squared error between the given spectrum and each stored spectrum is computed, and the values of F_1, F_2, and F_3 associated with the best match are used, together with fixed values of

F_4, GBF, and BW_1-BW_4, as the initial approximation. A data subprogram containing one example of each of 12 vowels from the data on adult males from Peterson and Barney[8] and Stevens and House,[9] plus three additional versions of the vowel /i/, has given good initial guesses in the analysis of vowels from adult males. The initial value of F_3 in /i/ seems to be fairly critical, since it is often close to F_2 and F_4 and not visible as a separate peak. If vowels spoken by females or children were to be analyzed, the data subprogram could be changed or augmented.

2. Example of the Use of VABS

VABS is currently being used in connection with SPADE, a general-purpose speech spectral analysis program available to users of our computer. Figure IX-6 shows two display photographs illustrating two stages in the use of VABS. The cursor is positioned



(a)

(b)

Fig. IX-6.

SPADE display photographs showing two stages in the analysis-by-synthesis of an example of the vowel /i/. (a) After the initial approximation (b) Final result of the analysis. The two graphs in the lower halves represent low-frequency energy and fundamental frequency as functions of time. The vertical cursor shows the point in the utterance corresponding to the spectrum shown above. The points on the vertical axis of the spectrum represent 2-dB steps in amplitude; each horizontal point represents one of the 36 filter outputs. Two spectra are shown superimposed. The natural vowel spectrum is displayed with slightly more intense (larger) points. The difference between the spectra is shown on the abscissa.

in the vowel /i/. Figure IX-6a shows the results after the initial approximation phase of VABSI. The spectrum to be analyzed is displayed with intense points, and the synthetic spectrum is superimposed with less intense points. The difference between the two is displayed on the abscissa. The two markers on the abscissa delimit the frequency range over which the match is made; in this case, 0-3175 Hz. The squared error (E = 327) and the values of the synthesis parameters (all in Hz) are displayed at the left of the spectrum.

Figure IX-6b shows the results after the strategy algorithms of VABSA have performed their task. The two spectra are virtually identical over the frequency range of the match.

3. Discussion

This vowel analysis-by-synthesis program is now being used to analyze the vowels of 21 adult male speakers. Analyses for some speakers result in significantly higher error scores than for others, thereby indicating that the synthesis model is not as good for their vowels. These discrepancies are probably due to differences in the glottal source spectrum. For the majority of speakers, however, the error scores for the vowels examined thus far, /i/, /ɪ/, /a/, and /æ/, are at least comparable to those reported earlier for only 3 speakers.[4]

J. J. Wolf

References

1. J. J. Wolf, "Choice of Speaker Recognition Parameters," Quarterly Progress Report No. 97, Research Laboratory of Electronics, M. I. T., April 15, 1970, pp. 125-133.
2. C. G. Bell, H. Fujisaki, J. M. Heinz, K. N. Stevens, and A. S. House, "Reduction of Speech Spectra by Analysis-by-Synthesis Techniques," J. Acoust. Soc. Am. 33, 1725-1736 (1961).
3. A. P. Paul, "An Automatic Spectrum Matching Program," Quarterly Progress Report No. 66, Research Laboratory of Electronics, M. I. T., July 15, 1962, pp. 275-278.
4. A. S. House and A. P. Paul, "Analysis of Error Scores Derived from an Automatic Vowel Reduction Scheme," Quarterly Progress Report No. 69, Research Laboratory of Electronics, M. I. T., April 15, 1963, pp. 161-163.
5. W. L. Henke, "TASS – Another Terminal Analog Speech Synthesis System," Quarterly Progress Report No. 95, Research Laboratory of Electronics, M. I. T., October 15, 1969, pp. 73-81.
6. L. R. Rabiner, "Speech Synthesis by Rule: An Acoustic Domain Approach," Ph. D. Thesis, Department of Electrical Engineering, M. I. T., June 1967, pp. 58-60.
7. A. P. Paul, "Strategy for the Automatic Matching of Speech Spectra," S. M. Thesis, Department of Electrical Engineering, M. I. T., 1961.
8. G. E. Peterson and H. L. Barney, "Control Methods Used in a Study of the Vowels," J. Acoust. Soc. Am. 24, 175-184 (1952).
9. K. N. Stevens and A. S. House, "Perturbation of Vowel Articulations by Consonantal Context: An Acoustical Study," J. Speech Hearing Res. 6, 111-123 (1963).

C. AN INTERACTIVE COMPUTER GRAPHICS AND AUDIO SYSTEM SANS LARGE BUDGETS AND GREAT FUSS

We shall report on the present status of certain aspects of the Speech Communication Group's computer system, and a few specific applications of that system. A second objective is to show by these examples what can be easily realized, with a small amount of judicious system design, in the area of interactive computer graphics on a quite modest system.

The notion is still quite prevalent that rather extensive hardware complemented by much systems software is a prerequisite for interactive computer graphics. The machine costs implied are generally in the vicinity of several hundred thousand dollars or greater. To this, then, may be added several man-years of graphics-oriented system programming in addition to the standard software system. Often whole new graphics-oriented languages are called for which entail the additional development of the language processors. Our point here is that such discussions should not discourage the small-system user from thinking almost in as large terms as his brethren who have many boxes of expensive hardware and a staff of system programmers.

The system under discussion comprises integrated hardware and system software combined with a body of programming techniques.

The hardware of the system contains an 18-bit word central processor (a PDP-9 computer, which is considered to be a fairly small machine) with 24K core memory, a small disk, and small magnetic tapes (DEC tapes) for auxiliary storage. Graphical input is provided for by a tablet. Graphical output is provided by a cathode-ray tube display monitor driven by an in-house designed display processor with a parts cost of approximately $3000.[1] There are also extensive real-time audio input/output facilities.[2]

The general software of the system has a file system and input/output monitor, editors, a relocating loader, an on-line debugger, a macro assembler, and various problem-oriented languages. FORTRAN IV is the language of all of the examples reported here. Other high-level languages available are SNOBOL (a text string processing language), FOCAL (a "desk calculator" type of interpreter), and a form of DYNAMO (which will be discussed in this report).

The FORTRAN IV language is used primarily because of its availability, both on this specific small machine and many others. It is not inherently well adapted to interactive graphics programming, having poor syntax and limited semantics, but since it is a procedure-oriented language, its semantics can be effectively extended by procedure calls on an integrated system library. Of major importance for the applications shown here are library packages for the creation and manipulation of display output; for the accessing of interactive inputs such as on the tablet, knobs, and switches; and for the input/output of real-time audio. A major conceptual expansion is machinery for the creation and processing of "plexes" (linked data structures and lists). Primitive procedures provide for the accessing of such data structures from a FORTRAN language environment, and a free-storage package furnishes the necessary storage accounting functions which allow completely arbitrary types of data structures to be built and manipulated.

1. Sample Applications

Figures IX-7, IX-8, and IX-9 illustrate some present applications and demonstrate the capabilities of the graphics system. They are photographs made from the on-line CRT display.

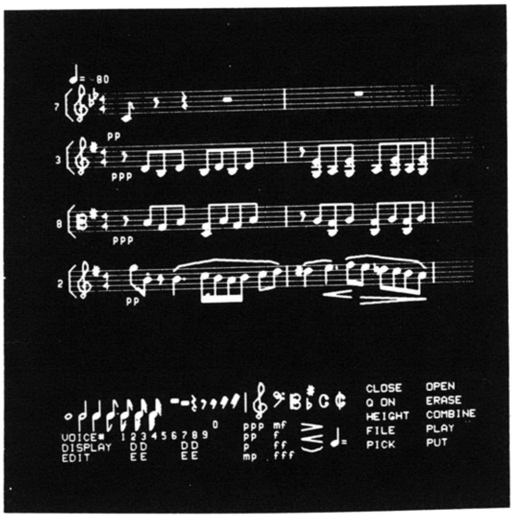


Fig. IX-7.

Excerpt from an educational motion picture film showing electromagnetic radiation from a dipole. The advantage of producing computer-generated animated films on-line is one of programming time. Many functions, such as adding title to scenes, selecting parameters, and determining sequences, can be executed during the actual filming. The result of this is that the production of simple types of computer-generated films becomes very quick and inexpensive. A three-minute film might thus have a total cost of only one man-hour for designing the script and writing the computer program, one man-hour for filming on the computer, and $10 for the film.

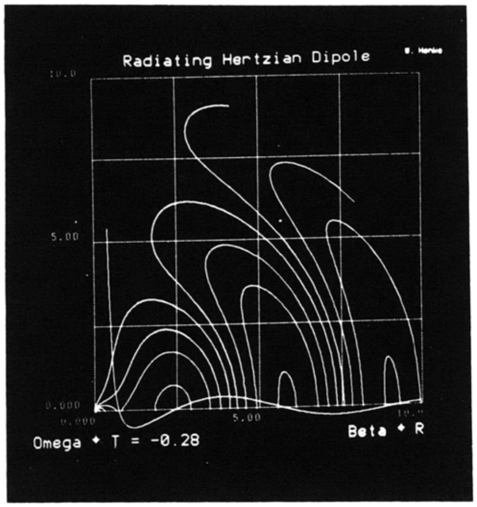
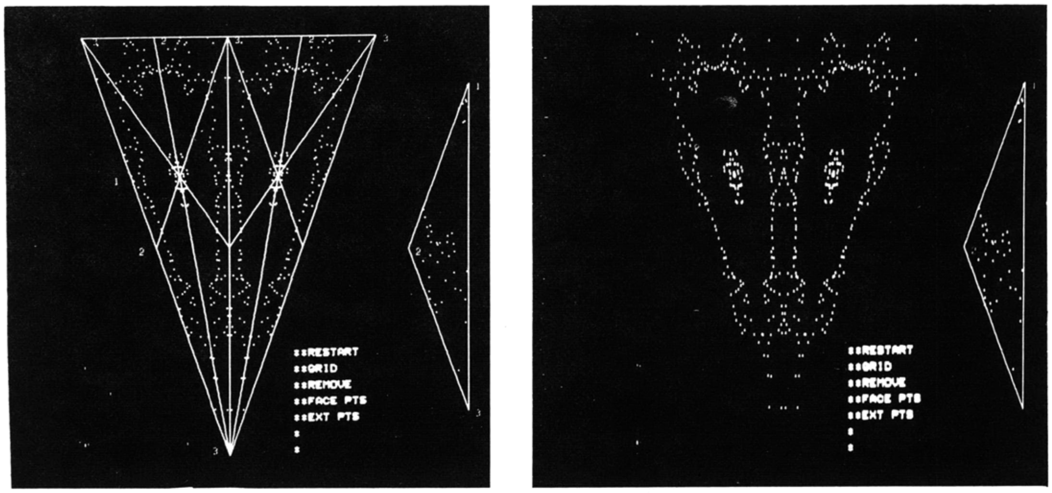


Fig. IX-8.

This computer display of a segment of the orchestral score for Mahler's Fourth Symphony was generated by using an experimental musical score editor. The upper part of the picture shows voices currently in the display (here, voices 2, 3, 7, and 8). The lower part of the display contains a "command menu" which is used in conjunction with the tablet to request the execution of editing functions and to pick out items (e.g., notes) to be placed on the staves. Associated synthesizers, initially implemented in software, input such scores and output acoustical renditions of the same. This work is part of a project to develop an interactive computer facility for the creation of electronic music. (This "traditional score" editor was a work of Walter Bilofsky.)



(a) (b)

Fig. IX-9. M. V. Cerrillo has proposed that facial images can be constructed by specifying points in a base triangle and then transforming these points to each of several triangles arranged in an image grid. By implementing the transformational algorithms on a computer, an almost instantaneous artist's transformational sketch pad can be realized to easily experiment with such proposals. (a) Shows a base triangle, an image grid, and sketched-in points; in (b) the image grid has been removed to give a better view of the image constructed from this set of points. (Project of James F. Smith.)

2. DYNAMO

One particular project, a DYNAMO implementation, will be described in greater detail in Section IX-D. Here we shall give a brief overview of that project.

DYNAMO is a continuous system simulator originally oriented toward industrial management activities. It has been implemented on several large machines during the last eight years.

The modeling of the interaction of various activities can be represented by a block diagram. In a usual DYNAMO implementation, the user specifies his model in the "DYNAMO language" which is a unit text record-oriented notation. In the present implementation, the model is specified graphically by using a tablet and display. We expect that most users will find the graphical notation more intuitive and therefore much easier to learn than the usual text-oriented notation.

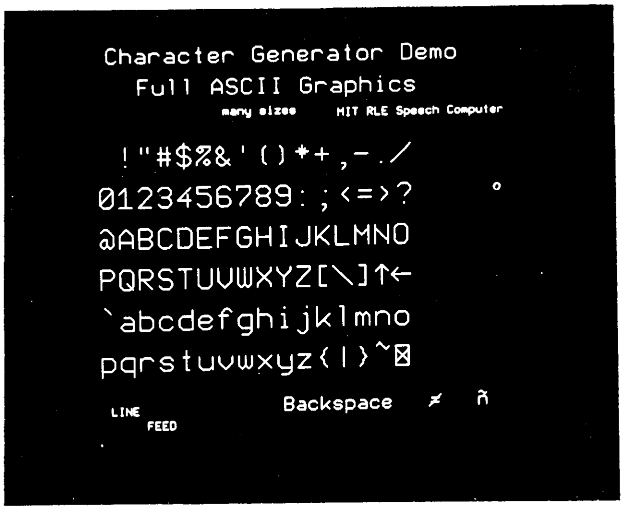
Since DYNAMO is a simulator, the ability to quickly modify the model on the basis of observations is a desirable feature that distinguishes this on-line interactive implementation from the usual off-line usage. In addition to its interactive graphical nature, another very significant economic aspect of this implementation is that it was done on a fairly small system, and in less than two man-months of time by a person who had negligible programming experience when he began.

Since DYNAMO is basically a continuous system simulator, the internal functions can easily be adapted to the simulation of other continuous systems, e.g., electric circuits, and we expect to exploit this property in the future. We also expect to use the general graphical block diagram layout procedures for the specification of interconnection diagrams for other applications, a prime example being that of music and speech synthesizer configurations. Such synthesizers can then be quickly realized by using digital signal-processing programs for experimental implementation.

3. Programming Techniques and the System Library

We feel that an important aspect of the system is what might be called a style of program design or body of programming technique. By encouraging users to think in terms of modeling their problem rather than in terms of "bit shuffling" on a computer, we believe that a large reduction in programming time on most projects can be achieved. Such a culture is embodied in the design of our system – particularly the system library which, although small, we believe provides services organized in such a way as to help the user think about his problem rather than harass him with computer-oriented details.

In order to give the flavor of programming when using the system, we include two excerpts from the documentation for the system library. Figures IX-7 through IX-10 have shown the graphical aspects of the system, and so these excerpts pertain to other aspects: a free-storage management package oriented toward data structure design, and an audio input/output package used by digital signal-processing programs.



```
Character Generator Demo
  Full ASCII Graphics
        many sizes    MIT RLE Speech Computer

 !"#$%&'()*+,-./
0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]↑←
`abcdefghijklmno
pqrstuvwxyz{|}~⊠
                Backspace   ≠   ñ
LINE
    FEED
```

Fig. IX-10. Demonstration of the capabilities of the stroke-type character generator subsystem of the display processor.[3]

IDENTIFICATION: FREE

FREE STORAGE PACKAGE, FOR DYNAMIC STORAGE ALLOCATION.
W. HENKE, OCT. 69

THE FREE STORAGE PACKAGE PROVIDES FOR THE CREATION AND PROCESSING OF "PLEXES" (LINKED DATA STRUCTURES AND LISTS) BY FURNISHING A STORAGE MANAGEMENT SERVICE. PRIMITIVE FUNCTIONS, SEE THE "RVLV" PACKAGE, PROVIDE FOR THE ACCESSING OF SUCH DATA STRUCTURES FROM A FORTRAN LANGUAGE ENVIRONMENT.

EXPLICIT CALLS ARE PROVIDED FOR REQUESTING BLOCKS OF FREE STORAGE AND FOR RETURNING BLOCKS TO FREE STORAGE. WHENEVER A BLOCK OF SPACE IS RETURNED IT IS COALESCED WITH ANY OTHER CONTIGUOUS BLOCKS CURRENTLY IN FREE STORAGE SO AS TO MINIMIZE THE SHATTERING PROBLEM.

THE CORE AREA USED FOR FREE STORAGE IS THAT LEFT BY THE LOADER AFTER ALL PROGRAM UNITS HAVE BEEN LOADED AND ALL NAMED COMMON BLOCKS HAVE BEEN ALLOCATED. IT IS THE SAME AREA THAT THE LOADER REGARDS AS BLANK OR UNNAMED COMMON. THUS ANY CORE LOAD WHICH USES THE FREE STORAGE PACKAGE SHOULD NOT USE BLANK COMMON, BUT INSTEAD USE NAMED COMMON BLOCKS WHEN THE COMMON FACILITY IS DESIRED.

MOST VALUED FUNCTIONS OF THIS PACKAGE ARE CONCEPTUALLY OF TYPE "POINTER", AND SHOULD BE DECLARED OF TYPE "INTEGER" IN A FORTRAN ENVIRONMENT. ALL SIZES ARE OF TYPE INTEGER.

CALL FRINIT -- INITIALIZE FREE PACKAGE.
MUST BE CALLED ONCE BEFORE ANY OTHER FREE PACKAGE CALLS, AND MAY BE CALLED AGAIN TO REINITIALIZE, I.E., EFFECTIVELY RECLAIM ALL ORIGINAL FREE STORAGE FOR A FRESH START.

FREEP = FREZF(SIZE) F4 -- "FREE ZEROED", GETS A ZEROED BLOCK OF 'SIZE' WORDS, VALUE IS A PTR (POINTER) TO THE BLOCK.

NEWFP = FRECF(SIZE, COPYP) F4 -- "FREE COPIED INTO", GETS A BLOCK OF 'SIZE' WORDS, COPIES BLOCK OF COPYP INTO NEW BLOCK, AND RETURNS PTR TO NEW BLOCK.

CALL FRETF(SIZE, RETPTR) F4 -- "FREE RETURN", RETURNS TO FREE STORAGE THE BLOCK OF 'SIZE' WORDS POINTED TO BY 'RETPTR'.

NEWOBP = FRECOB(OB) F4 -- COPY THE GIVEN OB INTO A NEW BLOCK FROM FREE STORAGE, VALUE IS PTR TO COPY OF OB. AN OB IS A BLOCK OF WORDS HEADED BY ITS WORD COUNT. WORD COUNT DOESN'T INCLUDE ITSELF, SO BLK SIZE IS WDCNT+1.

CALL FRETOB(OB) F4 -- RETURNS AN OB TO FREE STORAGE.

SIZE = FREL() F4 -- RETURNS THE SIZE OF THE LARGEST AVAILABLE REMAINING BLOCK OF FREE STORAGE.

HIS MANUAL REAL-TIME AUDIO I/O, SS PACKAGE 5/70 PAGE 1

IDENTIFICATION: SSDWAS

SS PACKAGE, "SAMPLED SIGNAL TO/FROM REAL-TIME SIGNAL STORE"
W. HENKE JAN 70

THIS PACKAGE PROVIDES A FACILITY TO DO REAL-TIME ANALOG SIGNAL (TYPICALLY AUDIO) INPUT/OUTPUT FROM A SPECIAL DIGITAL "REAL-TIME" SIGNAL STORE. THE DIGITAL REPRESENTATION OF THE ANALOG SIGNAL CONSISTS OF A SEQUENCE OF DIGITIZED SAMPLES UNIFORMLY SPACED IN TIME AT THE SAMPLING RATE.

ONE ENTRY -- SSPRD (SS PLAY, RECORD, AND DISPLAY) -- PROVIDES THE REAL-TIME FUNCTIONS OF PLAYING (OUTPUTTING AS AUDIO), RECORDING (INPUTTING AS AUDIO), AND DISPLAYING (WAVEFORMS AND SPECTRA) TO/FROM THIS SIGNAL STORE.

OTHER ENTRIES -- SSOUT AND SSIN -- PROVIDE THE USER A MEANS TO ASSEMBLE A SIGNAL IN THE SIGNAL STORE FOR SUBSEQUENT OUTPUT (PLAYING) OR TO ACCESS THE SIGNAL STORE TO READ A PREVIOUSLY INPUTTED (RECORDED) SIGNAL.

THUS SIGNAL GENERATION PROGRAMS CAN CALL THE SS PACKAGE WITH EACH SAMPLE AS IT IS GENERATED. THE SEQUENCE OF SAMPLES SO SPECIFIED IS ASSEMBLED INTO THE STORE. SSPRD() MAY BE SUBSEQUENTLY CALLED TO PLAY AND VISUALLY DISPLAY THE WHOLE SIGNAL WHICH HAS BEEN THUS CONSTRUCTED.

THE SECONDARY STORAGE DEVICE USED FOR THE SIGNAL STORE IS THAT ASSIGNED TO LOGICAL I/O UNIT #1. A HIGH DATA TRANSFER RATE IS NEEDED FOR REAL-TIME I/O FROM THE STORE, AND OF THE CURRENT HARDWARE ONLY THE DISK HAS THE NEEDED RATE. THUS A DISK HANDLER (DKA OR DKD) SHOULD BE ASSIGNED TO I/O SLOT # 1. DECTAPE MAY BE USED FOR SSOUT, AND ALSO FOR PLAYBACK USING VERY LOW SAMPLING RATES. AN ABSOLUTE ADDRESSED PORTION OF THE ASSIGNED SECONDARY STORE IS USED FOR SIGNAL STORAGE. ANY OTHER DATA (E.G., FILES) WHICH EXISTED ON THAT PORTION WILL BE OVERWRITTEN.

FOR SAMPLED SIGNALS THAT NEED NOT BE ACCESSED IN REAL-TIME, STORAGE AS FILES IN FORTRAN BINARY FORMAT IS SUGGESTED INSTEAD OF THE SINGLE SPECIAL PURPOSE REAL-TIME SIGNAL STORE.

THE SS PACKAGE GETS CORE BUFFER SPACE FROM THE FREE STORAGE PACKAGE, SO USERS SHOULD NOT USE "BLANK" COMMON IN ANY CORE LOAD WHICH USES THE SS PACKAGE. "NAMED" COMMON SHOULD BE USED INSTEAD. (THE SPACE REQUESTED IS CURRENTLY 6200 WORDS.)

CALL SSPRD F4 -- SS PACKAGE PLAY, RECORD, AND DISPLAY.

PLAYS AND RECORDS FROM/TO THE SIGNAL STORE, AND DISPLAYS WAVEFORMS AND SPECTRA OF THE SIGNAL. SEE THE DOCUMENTATION FOR "DWAS" [RLE QPR #95, P 69, + APPENDMENTS] FOR THE HARDWARE INPUT COMMANDS TO SSPRD. NOTE THAT AFTER SSPRD HAS BEEN CALLED IT RETAINS CONTROL UNTIL THE ESCAPE COMMAND (BUTTON 35) IS ISSUED, WHICH

CAUSES CONTROL TO BE RETURNED TO THE CALLING PROGRAM. THE SAMPLING RATE IS DETERMINED BY THE SETTING OF THE HARDWARE SWITCH "CLOCK2". AUDIO OUTPUT APPEARS AT THE "DA/CBO" JACK ON THE AUDIO PATCH PANEL, AND AUDIO INPUT SHOULD BE SUPPLIED TO THE "MPX4" JACK. FOR MOST APPLICATIONS THE ANALOG SIGNAL SHOULD BE LOW-PASS LIMITED AT ONE-HALF THE SAMPLING RATE (ANALOG FILTER JACKS ARE AVAILABLE ON THE PATCH PANEL).

SINGLE SAMPLE AT A TIME ACCESS TO THE SIGNAL STORE:

THESE ENTRIES TRANSFER SIGNAL SAMPLES TO/FROM THE USERS SIGNAL PROCESSING PROGRAM AND THE "REAL-TIME" SIGNAL STORE.

CALL SSOINI(ERRLBL,ERCODE) F4 -- SAMPLE OUT TO SIGNAL STORE INITIALIZE.

SSOINI MUST BE CALLED BEFORE USING USING SSOUT.

ERRLBL: A STATEMENT LABEL TO WHICH CONTROL WILL BE TRANSFERED IF ANY SUBSEQUENT CALLS TO THE SS PACKAGE ENCOUNTER ERRORS. [NOTE THAT ACTUAL ARGUMENTS OF TYPE LABEL CAN BE REALIZED IN FORTRAN USING THE "ASSIGN N TO LABELVAR" CONSTRUCT WHERE LABELVAR IS OF TYPE INTEGER.]

ERCODE: AN INTEGER VARIABLE WHICH WILL BE UPDATED WITH THE ERROR CODE WHEN AN ERROR OCCURS, I.E., JUST BEFORE PROGRAM CONTROL IS TRANSFERED TO ERRLBL.
ERROR CODES ARE:
0   NO ERROR
1   SIGNAL STORE FULL

CALL SSOUT(SAMPLE) F4 -- SINGLE SAMPLE OUTPUT TO SIGNAL STORE.

SAMPLE: IS APPENDED TO THE SEQUENCE OF SAMPLES BEING ASSEMBLED. IT IS AN INTEGER VARIABLE WHICH SHOULD BE SCALED SO THAT PEAK VALUES OF THE REPRESENTED SIGNAL ARE JUST WITHIN A RANGE OF + TO - 131,071 (2**17 - 1).

CALL SSFINI() F4 -- FINISH AN SSOUT RUN.

MUST BE CALLED TO TERMINATE ANY SSOUT OUTPUT, INCLUDING THOSE WHICH MUST BE TERMINATED DUE TO A FULL SIGNAL STORE.

CALL SSIINI(ERRLBL,ERCODE) F4 -- SAMPLE INPUT FROM SIGNAL STORE INITIALIZE.

HIS MANUAL   REAL-TIME AUDIO I/O, SS PACKAGE   5/70 PAGE 3

SSIINI MUST BE CALL BEFORE USING SSIN

ERRLBL: SAME FUNCTION AS FOR SSOUT.

ERCODE: SAME FUNCTION AS FOR SSOUT.
ERROR CODES ARE:
1   END OF SIGNAL IN SIGNAL STORE REACHED

SAMPLE = SSIN() F4 -- INPUT A SINGLE SAMPLE FROM SIGNAL STORE.

DECLARE SSIN OF TYPE INTEGER. SAMPLE IS THE NEXT SEQUENTIAL SAMPLE FROM THE SIGNAL STORE, SCALING THE SAME AS FOR SSOUT.

THE SIGNAL STORE CAN BE OPEN FOR BOTH READING AND WRITING SIMULTANEOUSLY. AS LONG AS READING (SSIN) IS KEPT AHEAD OF WRITING (SSOUT) THE NEW SIGNAL BEING WRITTEN WILL NEVER OVERWRITE THE OLD SIGNAL BEING READ.

ENTRY TO ALLOW USER TO ACCESS SS CORE BUFFER SPACE WHEN IT IS NOT BEING USED BY THE SS PACKAGE. SSOUT AND SSIN USE THIS BUFFER, IN ADDITION TO SSPRD, SO THE SPACE SHOULD NOT BE USED SIMULTANEOUSLY WITH ANY CALLS TO THE SS PACKAGE.

CALL SSBUF(SSBUFP,SIZE) F4

OUTPUT INTEGER ARGS:

SSBUFP: PTR TO THE CORE BUFFER ACQUIRED BY THE SS PACKAGE.

SIZE: BUFFER SIZE (WORD COUNT).

W. L. Henke

References

1. J. G. Fiasconaro, "A Computer-Controlled Graphical Display Processor," S. M. Thesis, M.I.T., 1970.
2. W. L. Henke, "Speech Computer Facility," Quarterly Progress Report No. 90, Research Laboratory of Electronics, M.I.T., July 15, 1968, pp. 217-219.
3. M. A. Bromberg, "A Stroke-type Character Generator for a Computer Display Processor," S. B. Thesis, M.I.T., 1970.

D. DYNAMO GRAPHIC SYSTEM

DYNAMO[1,2] is a simulation system defined several years ago to simulate the dynamics of industrial activities. Technically it is a continuous system simulation comprising integrators and several arithmetic functions. As a result, it is capable of being used for a large class of problems.

The program to enter the DYNAMO flow diagram and display the resulting system activity was written in the FORTRAN language. It has been run on the R.L.E. speech computer.

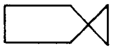
The DYNAMO flow diagram is entered by using the tablet and pen. An example of such a flow diagram is shown in Fig. IX-11. The rectangles are levels, the hexagonal blocks are auxiliary functions, and the functions shaped like  are rates. Figure IX-11 is an example of an elementary industrial dynamics situation. If the sales rate changes, the inventory changes. The company should then change its work force to react to the new sales rate. This is done via a hire-fire rate.

Figure IX-12 shows the type of output which the computer gives. In this case, 4 functions: work force, production, inventory, and hire-fire rate are plotted. The horizontal scale is in weeks. At week 10 (in this case) the sales rate changed from 200 to 250. Figure IX-13 shows the output when the sales rate is a positive ramp starting at week 10.

The following description of the use of DYNAMO is meant to give the flavor of how it is used.

Figure IX-11 shows a DYNAMO flow diagram as drawn on the CRT face. Initially, the large blocks are drawn, then the lines having arrowheads are drawn, then the command to change display size is given. '050' is entered into the keyboard to make the new blocks one-half of the size of the original one. Then the smaller blocks are drawn. The rest of the lines are then put in. The text and initial values are also entered.

When an auxiliary (hexagonally shaped block) is drawn, a display such as is shown in Fig. IX-14 requests the type of auxiliary. The plus sign (+) on the picture denotes the position of the pen.

When a line is drawn, a display like Fig. IX-15 asks for the line type. The line type is important when the line enters a level, rate, or equation type of block. In such cases, the line type determines how the value of the block is determined. For example, with a level block such as is shown in Fig. IX-16, there are 4 lines entering the block. They are types 1, 2, 6, and 3.

Type	Name
1	+ Addition
2	- Addition
3	+ Multiply
4	- Multiply
5	+ Division
6	- Division

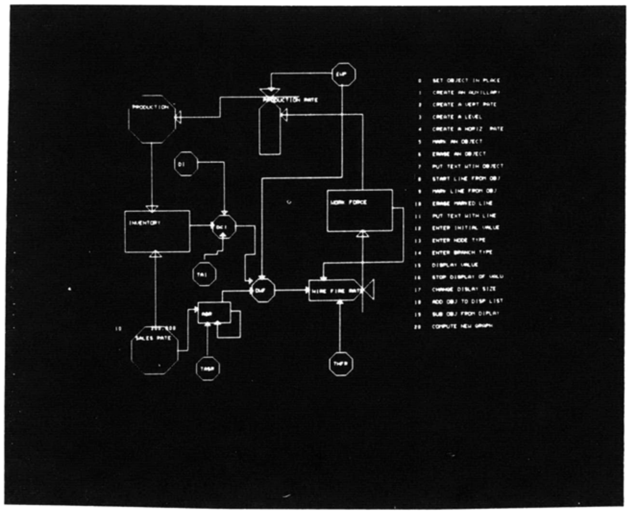


Fig. IX-11. Model of a sample system.

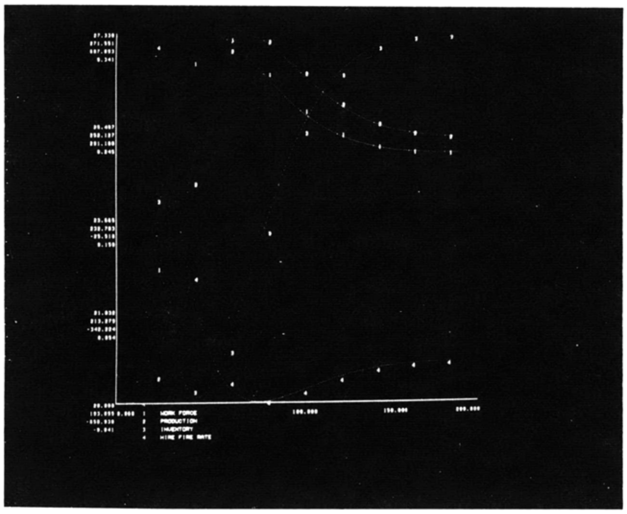


Fig. IX-12. Simulation output for sample system.

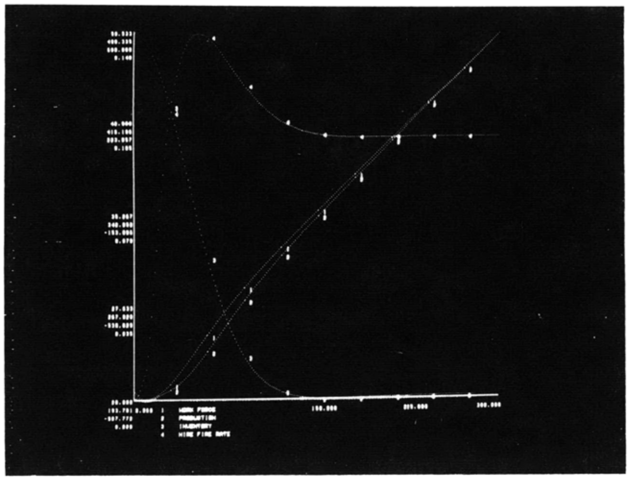


Fig. IX-13. Simulation output for sample system with a different sales rate.

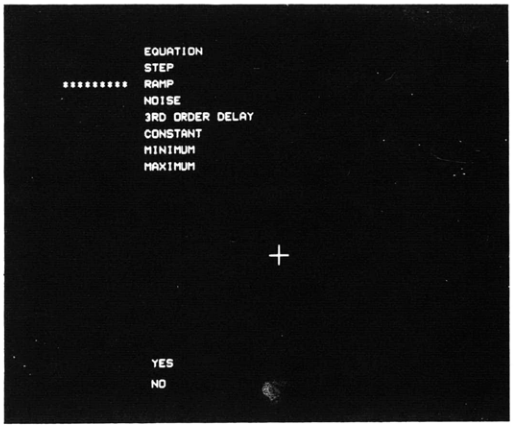


Fig. IX-14.

Model editor requesting type of auxiliary.

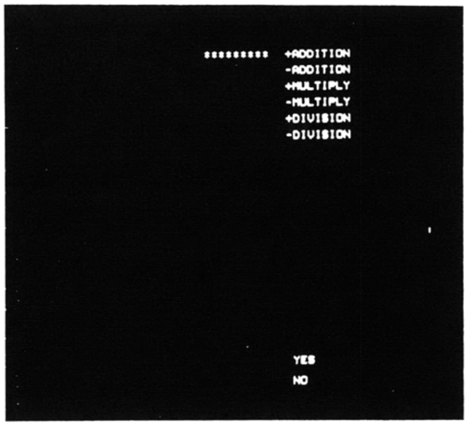


Fig. IX-15.

Model editor requesting type of input for a branch.

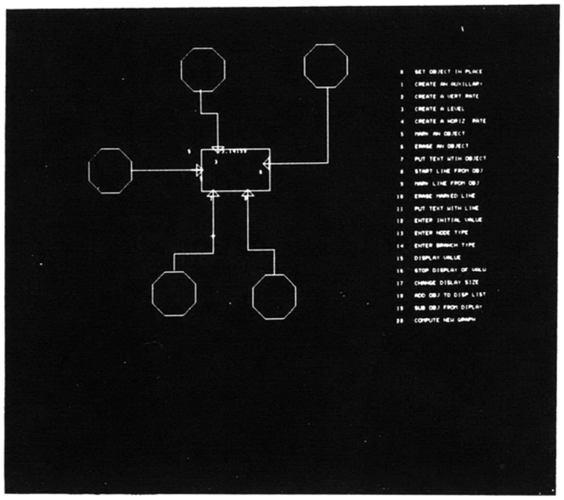


Fig. IX-16. Segment of a model displaying branch input types.

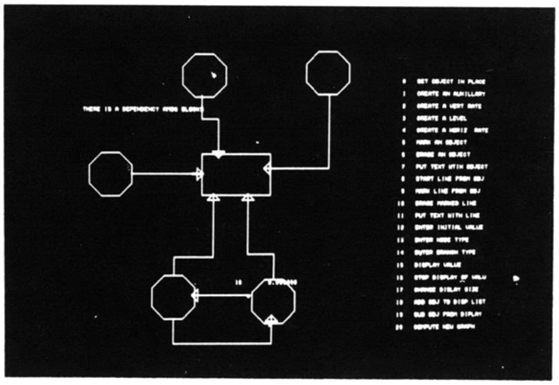

THERE IS A DEPENDENCY AROS BLOCK

Fig. IX-17. Display showing an inconsistent model with a warning to the user.

So the new value of the level will be

$$L.K = L.J + DT\left((A_1-A_2)*A_3\left(\frac{1}{A_4}\right)\right),$$

where

L.K is the new value of level

L.J is the old value of level

DT is the increment of integration

A_i is the value of Auxiliary which has an arrowhead of type i.

Figure IX-17 shows the result if there is a chain of auxiliaries that all feed each other. The computer is not able to compute a new value for the auxiliaries, so it outputs the command 'there is a dependency among blocks' after the request for computing a new graph is made.

Figure IX-16 and IX-17 show how the marked node has displayed with it the node type (at the upper left-hand side) and the initial value (upper right-hand side).

The graphical DYNAMO system provides the capability of quickly setting up a model of a continuous system and determining its response to an input. The quick response makes it easy to note problems in the model and change the system to behave as desired. Defining the system as a flow diagram rather than through a computer language provides a very easily comprehended overview of the system being simulated.

F. F. Sellers, Jr.

References

1. A. L. Pugh, DYNAMO User's Manual (The M.I.T. Press, Cambridge, Mass., 1963).
2. J. Forrester, Industrial Dynamics (The M.I.T. Press, Cambridge, Mass., 1961).