Foundations and Trends® in
sample
Vol. xx, No xx (2010) 1–147

DOI: xxxxxx

now
the essence of knowledge

Linear Predictive Coding and the Internet Protocol

A survey of LPC and a History of of Realtime Digital Speech on Packet Networks

Robert M. Gray[1]

[1] *Stanford University, Stanford, CA 94305, USA, rmgray@stanford.edu*

Abstract

In December 1974 the first realtime conversation on the ARPAnet took place between Culler-Harrison Incorporated in Goleta, California, and MIT Lincoln Laboratory in Lexington, Massachusetts. This was the first successful application of realtime digital speech communication over a packet network and an early milestone in the explosion of realtime signal processing of speech, audio, images, and video that we all take for granted today. It could be considered as the first voice over Internet Protocol (VoIP), except that the Internet Protocol (IP) had not yet been established. In fact, the interest in realtime signal processing had an indirect, but major, impact on the development of IP. This is the story of the development of linear predictive coded (LPC) speech and how it came to be used in the first successful packet speech experiments. Several related stories are recounted as well. The history is preceded by a tutorial on linear prediction methods which incorporates a variety of views to provide context for the stories.

Contents

Preface

The origins of this paper lie in a talk I gave at the Special Workshop in Maui (SWIM) in January 2004 titled *California Coding: Early LPC Speech in Santa Barbara, Marina del Rey, and Silicon Valley 1967–1982* and on the historical research I conducted in preparation for the talk, including oral histories I gathered in 2003 from several of the principal players. During subsequent years the presentation evolved as I gave similar talks as part of the IEEE Signal Processing Society Distinguished Lecture Program and elsewhere. Topics and stories were added and deleted and tuned to different audiences as I received questions, comments, and anecdotes from a variety of listeners, from vintage pioneers to novice students. Some of the material converged in a short paper I wrote for the *IEEE Signal Processing Magazine*[58], and the talk material largely converged in summer 2007 with a new title of "Packet speech on the ARPAnet: A history of early LPC speech and its accidental impact on the Internet Protocol." Since giving the original talk I have hoped to find the time to collect the material into a single document combining several variations of the talk along with more of the stories and details, and this manuscript is the result. As with the presentations, the historical details require an overview of linear prediction

and its many cousins. While there are several excellent treatments of linear prediction and its application to speech processing in book and survey form (see in particular the classic references by J. Makhoul [91] and by J. D. Markel and A.H. Gray Jr [104]), the historical prerequisites for this article provide a natural motivation for providing my own overview emphasizing certain key common points and differences among the many viewpoints and approaches. The first part of this paper is a technical survey of the fundamental ideas of linear prediction that are important for speech processing, but the development departs from traditional treatments and takes advantage of several shortcuts, simplifications, and unifications that come with years of hindsight. In particular, some of the key results are proved using short and simple techniques that are not as well known as they should be, and I use the occasion to comment on some of the common assumptions made when modeling random signals. The reader interested only in the history and already familiar with or uninterested in the technical details of linear prediction and speech may skip Part I entirely.

I have revisited all of the talks, my notes, the oral histories, my email archives, and the piles of documents collected along the way. I have also chatted and exchanged emails recently with several principal players in the story I had not previously talked with and read through relevant documents from that time that I had not seen before. I have browsed the Internet for new views of old events, but I have preferred my original sources and have tried to not dilute the primary source material used in the talks or acquired since. I have drawn on published oral histories of some of the participants whom I did not personally interview to fill in some gaps.

As I admitted with the title of the original SWIM presentation, the original talks emphasized the Pacific Rim side of the story, reflecting my bias towards telling parts of the story which I thought were less well known at the time. This was a proper focus for the story of the serendipitous connections between the University of Southern California's Information Sciences Institute (USC/ISI) of the University of Southern California (USC) in Marina del Rey and the Speech Comunications Research Laboratory (SCRL) in Santa Barbara that smoothed the cooperation of signal processors implementing the Itakura-Saito au-

tocorrelation method of speech coding with the network scientists designing a protocol to facilitate realtime signal processing on a rapidly growing and evolving network. During 2009 the article expanded to include more details about other partners in the project, especially MIT Lincoln Laboratory, Bolt, Beranek, and Newman) (now BBN), and Stanford Research Institute (SRI, now SRI International). It is a notable part of the story that many of the key participants were with smaller organizations and universities, and not with the most famous research universities and laboratories of the time. Many other institutions play important roles which are discussed, especially Culler-Harrison Inc. (CHI) on the West Coast and Bell Telephone Labs, and the National Security Agency (NSA) on the East Coast.

The central story is the one of the title, the combination of linear predictive coding with packet network protocols and the hardware of the time that led directly to the first successful understandable realtime digital speech on the ARPAnet, the packet network developed by the Advanced Research Projects Agency (ARPA),[1] and indirectly to the separation of the Internet Protocol (IP) and its descendents from the earlier Transmission Control Protocol (TCP). Many of the heroes of this story are listed in Figure 15.1 — the members of the Network Speech Compression (NSC) group formed by Robert (Bob) Kahn of ARPA and chaired by Danny Cohen of USC/ISI. Historical threads of the main story led to other significant technical developments, several of which are described in the history and epilogue.

In an admitted abuse of author's privilege, I have included a few technical and historical comments on topics related to the primary stories that are of lesser impact than the principal stories but are nonetheless personal favorites.

Related material of interest deemed inappropriate for print is provided as Web links, which may also be found at `http://ee.stanford.edu/~gray/lpcip.html`. My favorite item is an MPEG-1 video made in 1978 to illustrate the early packet speech

[1] The Advanced Research Projects Agency was first created in 1958. It was renamed as the Defense Advanced Research Projects Agency (DARPA) in March 1972, renamed ARPA in February 1993, and renamed DARPA again in March 1996. We will use ARPA in this history.

conference calls on the ARPAnet, It effectively demonstrates the speech quality and computing equipment of the time (as well as the clothes and hair styles). I recommend the video to all students of speech processing or computer networks with an interest in history. With time I hope to scan some of the rare documents of the time and post them as well.

The two parts can be read separately by those lacking the interest or patience for both. The first part is a largely self-contained overview of the fundamentals of linear prediction relevant to speech processing and other similar fields such as the analysis of geophysical and biomedical data. The second part is a technical history which can be read without the details of the first part; however, backwards references to the first part may be necessary for notational and vocabulary details. Writing history can be a constant work in progress, and I recognize that some of my interpretations may be questioned and some of the historical facts may be disputed or in error. I welcome comments and corrections that I will endeavor to include in future versions.

Those wishing a much shorter treatment of the primary story without the linear prediction material may prefer my note [58] or the Internet history coauthored by my first PhD student Barry Leiner and a list of Internet luminaries [81]. Additional Internet history resources can be found at `http://www.pcbargainhunter.com/articles/history-of-the-internet.html`

Dedicated to Mike McCammon, 1943–2008

Part I:

Linear Prediction and Speech

1

Prediction

Suppose that a data sequence $X = \{X_0, X_1, \ldots, X_{m-1}\}$ of m real numbers is observed and it is desired to guess the next value, X_m. What is the best possible guess? To proceed we clearly need to specify what we mean by "best." Given a definition, there are a variety of naturally related questions that can be asked.

Optimal 1-step prediction What is the optimal predictor of the form $\hat{X}_m = \pi(X_0, \ldots, X_{m-1})$? Here no constraints whatever are placed on the predictor[1] and it can be arbitrarily complex.

Optimal 1-step linear prediction Instead of placing no constraints on the predictor, we require it to be a linear combination of the observations[2]. Linear operations are generally simple to implement in software or hardware. What is the optimal linear predictor of the form $\hat{X}_m = -\sum_{l=1}^{m} a_l X_{m-l}$? For theoretical

[1] For the mathematically picky, it is required that the function p be a measurable function of its vector argument. Such an assumption is needed to apply calculus and probability theory.

[2] More generally we could consider affine predictions, but zero means will be assumed for simplicity throughout this document and hence the added generality would provide no extra benefit for the increased notational complexity.

purposes we will be interested in the asymptotic cases where m grows without bound or is assumed infinite, but the case of primary interest will be for finite m. Historically the most important specific value for speech has been $m = 10$ as a good balance of performance and complexity. For example, it results in a good representation of up to five formants of speech signals which is sufficient to cover the 0 to 4 kHz range.

Modeling/density estimation A superficially different issue is to observe a sequence of data and use it to construct a probabilistic model that explains the data. The question is what is the "best" model for X_m alone or given its predecessors based on the observed data? Usually by "model" is meant a probability distribution or, simply, distribution, but it can include structure as well. For example, the model might consist of a simple random process such as coin flips or memoryless Gaussian random variables driving a time-invariant linear filter. Intuitively, doing a good job of predicting the future of a random process given past observations should be equivalent in some sense to having a good model for the process.

Spectrum estimation What is the "best" estimate of the power spectral density or of its inverse Fourier transform, the autocorrelation, of a random process underlying the observed data? Power spectra are important in a variety of engineering applications, including the processing of speech (due to its perceptual importance), geophysical, and medical signals.

All of the above optimization problems are classic problems of mathematics, statistics, and engineering, and all typically require various additional assumptions on the nature of the signal being observed. Most commonly it is assumed that the sequence is produced by a *stationary* random process, one with a probabilistic description that does not change with time. It is also usual to assume that the process is also *ergodic*, which with stationarity assures the time averages or sample averages will converge to probabilistic averages or expectations. Neither assumption is necessary for many of the basic results, but both greatly simplify the discussion. The mathematical equipment exists for

significant generalizations, but often at the cost of increased complexity of notation. We here take a compromise approach and often (but not always) assume stationarity and ergodicity for convenience, but argue that most of the ideas and theory generalize provided it is assumed that sample averages converge.

It is occasionally said in the speech literature that speech is non-stationary, but very accurate and perfectly rigorous stationary and ergodic models can be constructed by assuming that with speech one has an infinite family of stationary and ergodic random processes well modeling the various local behaviors of speech (e.g., different vowels, unvoiced sounds, etc.) and a stochastic switch which randomly chooses short sequences from these processes as its output. Subject to suitable technical assumptions, the overall process (a *composite* or *mixture* process) will also be stationary and ergodic. Hence the process exhibits the overall behavior a stationary and ergodic process, but it also exhibits modes of short-term behavior described by distinct stationary and ergodic processes. For signal processing purposes, one can do better if one can capture accurately this local behavior. This is the goal of our predictors/estimators.

The application of primary interest here is the coding of speech into a digital sequence of relatively low bit rate for the purpose of communicating or recognizing the speech. A wide literature exists on all of these topics in a speech context and the topics are intimately related. See, e.g., J. Makhoul's classic survey [91] and J.D. Markel and A.H. Gray Jr's classic book [104], both of which appeared after the main story recounted here, but during an explosive growth period for digital speech processing and, more generally, of digital signal processing.

2

Optimal Prediction

2.1 The Problem and its Solution

Many of the problems and results are most easily described in vector/matrix form. For any positive integer m define the m-dimensional random column vector

$$X^m = (X_0, X_1, \ldots, X_{m-1})^t = \begin{pmatrix} X_0 \\ X_1 \\ \vdots \\ X_{m-1} \end{pmatrix},$$

where the superscript t denotes transpose. For notational simplicity, we assume throughout that the random vectors have zero mean.

Formally define a predictor $\hat{X}_m = \pi(X^m)$ of X_m given X^m as *optimal* if it minimizes the mean squared error (MSE) defined as the expected square of the prediction error $\epsilon_m = X_m - \hat{X}_m$:

$$E(\epsilon_m^2) = E[(X_m - \hat{X}_m)^2] = E[(X_m - \pi(X^m))^2] \tag{2.1}$$

over all possible predictors p. This is a classic problem in probability and its solution is well known. The optimal predictor of the random

variable X_m given the observations X^m is the conditional expectation

$$\hat{X}_m = \pi(X^m) = E[X_m|X^m], \tag{2.2}$$

and the minimum MSE over all predictors is the conditional variance

$$\begin{aligned} E[(X_m - \hat{X}_m)^2] &= \min_{\pi} E[(X_m - \pi(X^m))^2] \\ &= E[(X_m - E[X_m|X^m])^2] \\ &\stackrel{\Delta}{=} \sigma^2_{X_m|X^m}. \end{aligned} \tag{2.3}$$

The facts that the optimal predictor is a conditional expectation and the minimum MSE is the corresponding conditional variance are often invoked for pedagogical reasons to give the conditional expectation and the conditional variance practical interpretations.

There are of course a variety of notions other than squared error for measuring the distortion between an estimate and the true value, but we focus on squared error without apology for all of the usual reasons: it is simple, it is widely used, it is tractable for computation, and it is amenable to detailed analysis. In addition, the most important generalizations in many signal processing applications (such as perceptual coding) are input or output weighted quadratic distortion measures, where the weighting might be determined by perceptual considerations. The basic ideas usually remain the same, but the notational details can swamp a presentation.

2.2 Gaussian vectors

The rub, of course, to the easily described optimal solution is that in general the conditional expectation can be extremely difficult to evaluate. The most notable exception occurs when the vector X^{m+1} is Gaussian. To describe this special case, define the covariance (or covariance function) by

$$R_X(i,j) = E[(X_i - E(X_i))(X_j - E(X_j))].$$

for integers i, j. Since we assume that the means are zero ($E(X_i) = 0$ for all i), the covariance function reduces to the autocorrelation or, simply, correlation function

$$R_X(i,j) = E[X_i X_j].$$

Define the $m \times m$ correlation matrix

$$R^{(m)} = \{R_X(i,j);\ i,j = 0,1,\ldots,m-1\} = E[X^m(X^m)^t].$$

Autocorrelation matrices of real-valued random vectors are symmetric and nonnegative definite. We will assume that the matrices we encounter are strictly positive definite so that the determinant $\det R^{(m)}$ is strictly positive and hence the inverse ${R^{(m)}}^{-1}$ exists. Then the Gaussian probability density function (PDF) for X^m is given by

$$f_{X^m}(x^m) = \frac{e^{-\frac{1}{2}(x^m)^t {R^{(m)}}^{-1} x^m}}{\left((2\pi)^n \det R^{(m)}\right)^{1/2}}.$$

In the hope of minimizing objections to an apparent assumption of a Gaussian distribution for so nonGaussian a signal as speech, it should be emphasized that one of the key points made here is that Gaussian assumptions for analysis can yield important results even for nonGaussian problems. The assumption often provides a shortcut because either

(1) the Gaussian PDF provides an extreme or worst case for a particular problem, and hence also a useful bound, or
(2) the Gaussian PDF results in a simple derivation for a general result that does not actually depend on the details of the distribution, but only on specific parameters such as the correlation.

Insight is rarely lost because of such a trick, because the traditional derivations often involve detailed mechanics such as variational methods which provide little intuition (and are frequently incomplete or on mathematically shaky ground).

A straightforward but tedious exercise (see, e.g., [53], (4.54)–(4.61)) using the properties of Gaussian random vectors shows that the the conditional expectation and the conditional variance are given in the

Gaussian case by

$$\begin{aligned} E[X_m|X^m] &= (R_X(m,0), R_X(m,1), \ldots, R_X(m,m-1)) {R^{(m)}}^{-1} X^m \\ &\triangleq r_m^t {R^{(m)}}^{-1} X^m \qquad (2.4) \\ \sigma^2_{X_m|X^m} &= E\left[\left(X_m - r_m^t {R^{(m)}}^{-1} X^m\right)^2\right] = \frac{\det R^{(m+1)}}{\det R^{(m)}}. \qquad (2.5) \end{aligned}$$

Define the predictor

$$\hat{X}_m = r_m^t {R^{(m)}}^{-1} X^m \qquad (2.6)$$

and the prediction error or residual resulting from this predictor by

$$\begin{aligned} \epsilon_m &= X_m - \hat{X}_m = X_m - r_m^t {R^{(m)}}^{-1} X^m \\ &= b^t X^{m+1} = \sum_{i=0}^{m} b_i X_i, \qquad (2.7) \end{aligned}$$

where

$$\begin{aligned} b^t &= (-r_m^t {R^{(m)}}^{-1}, 1) \qquad (2.8) \\ r_m^t &= (R_X(m,0), R_X(m,1), \ldots, R_X(m,m-1)), \qquad (2.9) \end{aligned}$$

where the final equation above recalls (2.5). This leads to a convenient representation of the MSE as a quadratic form:

$$\begin{aligned} E[\epsilon_m^2] &= E\left[\left(\sum_{i=0}^{m} b_i X_i\right)^2\right] = \sum_{i=0}^{m}\sum_{j=0}^{m} b_i b_j E[X_i X_j] \\ &= \sum_{i=0}^{m}\sum_{j=0}^{m} b_i b_j R_X(i,j) = b^t R^{(m+1)} b. \qquad (2.10) \end{aligned}$$

This relation, (2.5), and the definition of b in (2.8) together imply that

$$b^t R^{(m+1)} b = \frac{\det R^{(m+1)}}{\det R^{(m)}}. \qquad (2.11)$$

Note that this relation *depends only on a valid correlation matrix* $R^{(m+1)}$ *having* $\det R^{(m)} > 0$ *and a* b *vector defined in terms of* $R^{(m+1)}$, it does not require Gaussianity! A Gaussian assumption is required to

equate this quantity to a conditional variance, but not for (2.11) to hold.

Three remarkable facts are apparent from the above discussion. In the special case of a Gaussian vector X^{m+1}:

(1) The optimal predictor and the resulting mean squared error are completely determined by the correlation matrix $R^{(m+1)}$.
(2) The optimal estimator of (2.6) is a *linear* function of the observations!
(3) The mean squared error resulting from the use of the predictor of (2.6) is

$$E[(X_m - \hat{X}_m)^2] = b^t R^{(m+1)} b = \frac{\det R^{(m+1)}}{\det R^{(m)}} \tag{2.12}$$

with b given by (2.8).

If X^{m+1} were not a Gaussian vector, but did have a correlation matrix $R^{(m+1)}$, then we could still define a linear predictor $\hat{X}_m$ as in (2.6), but we could no longer claim that it was the the conditional expectation and the optimal (unconstrained) predictor. The MSE will still be given in terms of the correlation by (2.12), but it will no longer in general be the conditional variance and hence the minimum possible.

In addition to providing a useful formula for the MSE, the linear predictor of (2.6) has some easily derived properties that hold whether or not X^{m+1} is Gaussian. While the properties of optimal *linear* prediction are developed in the next chapter, it is useful to derive a few properties of the specific linear predictor of (2.6) first.

Suppose now that $\hat{X}_m = r_m^t {R^{(m)}}^{-1} X^m$, but do not assume that X^{m+1} is Gaussian. Consider the cross correlation between the scalar prediction error or residual ϵ_m and the observation row vector $(X^m)^t$:

$$\begin{aligned} E[\epsilon_m (X^m)^t] &= E[(X_m - \hat{X}_m)(X^m)^t] \\ &= E[X_m (X^m)^t] - E\left[r_m^t {R^{(m)}}^{-1} X^m (X^m)^t\right] \\ &= r_m^t - r_m^t {R^{(m)}}^{-1} E\left[X^m (X^m)^t\right] \\ &= r_m^t - r_m^t {R^{(m)}}^{-1} R^{(m)} = 0. \end{aligned} \tag{2.13}$$

Equivalently, $E[\epsilon_m X_k] = 0$ for $k = 0, 1, \ldots, m-1$. Thus the error and the observation vector are *orthogonal* and therefore also

$$\begin{aligned} E[\epsilon_m \hat{X}_m] &= E[\epsilon_m r_m^t {R^{(m)}}^{-1} X^m] \\ &= r_m^t {R^{(m)}}^{-1} E[\epsilon_m X^m] = 0, \end{aligned}$$

As a result the MSE for the estimator $\hat{X}_m$ is

$$\begin{aligned} E[\epsilon_m^2] &= E[\epsilon_m (X_m - \hat{X}_m)] = E[\epsilon_m X_m] \\ &= E\left[(X_m - r_m^t {R^{(m)}}^{-1} X^m) X_m\right] \\ &= \sigma_{X_m}^2 - r_m^t {R^{(m)}}^{-1} E[X^m X_m] \\ &= \sigma_{X_m}^2 - r_m^t {R^{(m)}}^{-1} r_m. \end{aligned}$$

In summary, for the linear predictor $\hat{X}_m$ of (2.6), regardless of whether or not X^{m+1} is Gaussian, we have that

$$E[(X_m - \hat{X}_m)^2] = \sigma_{X_m}^2 - r_m^t {R^{(m)}}^{-1} r_m = \frac{\det R^{(m+1)}}{\det R^{(m)}}. \tag{2.14}$$

3

Linear Prediction

3.1 Optimal Linear Prediction

Now drop the assumption that the random vector is Gaussian, but restrict the predictor to be a linear function of the observations, specifically an estimate of the form

$$\hat{X}_m = -\sum_{l=1}^{m} a_l X_{m-l} = -\sum_{l=0}^{m-1} a_{m-l} X_l \tag{3.1}$$

for some set of set of coefficients (called *regression coefficients*) $a_k; k = 1, 2, \ldots, m$. The choice of sign is traditional and it makes things work out in a nice form. The predictor can be expressed in vector form as

$$\hat{X}_m \;=\; -\overline{a}_m^t X^m, \tag{3.2}$$

where

$$\overline{a}_m^t \;=\; (a_m, a_{m-1}, \ldots, a_1).$$

The optimal linear predictor will be given by the set of regression

coefficients that minimizes the MSE

$$\begin{aligned} E[\epsilon_m^2] &= E[(X_m - \hat{X}_m)^2] = E\left[\left(X_m + \sum_{l=0}^{m-1} a_{m-l}X_l\right)^2\right] \\ &= E\left[\left(\sum_{l=0}^{m} a_{m-l}X_l\right)^2\right] = \sum_{l=0}^{m}\sum_{i=0}^{m} a_{m-l}a_{m-i}E[X_lX_i] \\ &= \sum_{l=0}^{m}\sum_{i=0}^{m} a_{m-l}a_{m-i}R_X(l,i) \\ &= \sum_{l=0}^{m}\sum_{i=0}^{m} a_l a_i R_X(m-l, m-i). \end{aligned} \tag{3.3}$$

The MSE can be written as a quadratic form:

$$E[\epsilon_m^2] = \overline{a}^t R^{(m+1)} \overline{a}, \tag{3.4}$$

where

$$\overline{a}^t = (\overline{a}_m^t, 1) = (a_m, a_{m-1}, \ldots, a_1, a_0 \stackrel{\Delta}{=} 1).$$

We can now state precisely the linear prediction problem of order m, or LP(m) for short, as a constrained quadratic minimization:

$$\text{Find } \alpha_m = \min_{s \in \mathbb{R}^{m+1}: s_m = 1} s^t R^{(m+1)} s, \tag{3.5}$$

$$\overline{a} = \min_{s \in \mathbb{R}^{m+1}: s_m = 1} s^t R^{(m+1)} s, \tag{3.6}$$

where $\mathbb{R}^{m+1}$ denotes $(m+1)$-dimensional Euclidean space.

In most speech and other engineering-oriented texts, this minimization is accomplished by calculus: take the derivative with respect to all of the a_i for $i > 0$ and set them to zero to obtain a collection of m linear equations in m unknowns. Although rarely done, to be thorough it has to be shown that the resulting stationary point is indeed a minimum, either by looking at second derivatives or by showing convexity. An alternative derivation involves using the orthogonality principal of estimation, that the optimal linear estimate must produce an error that is orthogonal to the observations, which leads to the same set of equations. Given the groundwork we have laid, however, all this work is unnecessary.

Comparing (2.10) with the definition of α_m yields

$$b^t R^{(m+1)} b \geq \min_{s \in \mathbb{R}^{m+1}: s_m = 1} s^t R^{(m+1)} = \alpha_m \tag{3.7}$$

On the other hand, α_m depends on the underlying distribution only through the correlation matrix $R^{(m+1)}$, so it is the same whether or not the distribution is Gaussian. But if it is Gaussian, then the best linear estimator can not yield smaller MSE than the optimal unconstrained estimator, which implies that

$$b^t R^{(m+1)} b \leq \alpha_m. \tag{3.8}$$

Equations (3.7) and (3.8) together with (2.11) show that

$$\alpha_m = b^t R^{(m+1)} b = \sigma^2_{X_m} - r_m^t {R^{(m)}}^{-1} r_m = \frac{\det R^{(m+1)}}{\det R^{(m)}} \tag{3.9}$$

and the optimizing s in (3.5) is $s = \overline{a} = b$ with b given by (2.8) and r_m by (2.9). This implies that the optimal linear predictor in vector form of (3.2) is given by

$$\overline{a}_m^t \quad = \quad -r_m^t {R^{(m)}}^{-1}. \tag{3.10}$$

This provides a proof of the global optimality of the linear estimator defined by (2.6), and the only assumption required is that $\det R^{(m)} > 0$. The result also yields several useful byproducts. First, exactly as in (2.13),

$$E[\epsilon_m X_k] = 0; k = 0, 1, \ldots, m-1, \tag{3.11}$$

the *orthogonality* condition of optimal linear estimators. One can start by proving that the solution to the LP(m) problem must yield an error satisfying these conditions, and then derive (3.10) from this fact. Thus (3.10) and (2.13) are equivalent.

Next, observe that (3.10) yields

$$\overline{a}_m^t R^{(m)} = -r_m^t \tag{3.12}$$

or

$$\begin{aligned} -\sum_{k=0}^{m-1} a_{m-k} R_X(k,i) &= -\sum_{k=1}^{m} a_k R_X(m-k,i) \\ &= R_X(m,i); i = 0, 1, \ldots, m-1, \end{aligned}$$

which we can rewrite as

$$\sum_{k=0}^{m} a_{m-k} R_X(k,i) = \sum_{k=0}^{m} a_k R_X(m-k,i) = 0; i = 0,1,\ldots,m-1. \quad (3.13)$$

An additional equation is obtained by expressing α_m using the orthogonality property as

$$\begin{aligned} \alpha_m &= E[\epsilon_m(X_m - \hat{X}_m)] \\ &= E[\epsilon_m X_m] = E[(X_m + \sum_{i=1}^{m} a_{m-i} X_i) X_m] \\ &= R_X(m,m) + \sum_{i=0}^{m-1} a_{m-i} R_X(m,i) \end{aligned}$$

or

$$\alpha_m = \sum_{i=0}^{m} a_{m-i} R_X(m,i) = \sum_{i=0}^{m} a_i R_X(m,m-i). \quad (3.14)$$

The point is that the solution to the LP(m) problem implies $m+1$ linear equations (3.13)–(3.14) for the $(m+1)$ variables α_m and $a_1, a_2, \ldots, a_m$. These equations are known as the Yule-Walker or discrete-time Wiener-Hoph equations. As will be seen, these equations simplify if the vector is assumed to be produced by a stationary random process. Conversely, given these equations we could solve them to find α_m and $a_1, a_2, \ldots, a_m$. Thus solving these equations is equivalent to solving the LP(m) problem.

3.2 Unknown Statistics

What if the correlation matrix $R^{(m+1)}$ is not known, but a long sequence of actual data $X_0, X_1, \ldots, X_{n-1}$ is observed which can be used to estimate $R^{(m+1)}$? Typically it is fair to assume that $n >> m$, but this assumption can be dubious and have an impact on the algorithms. Intuitively, if we want to estimate the $(m+1)$-th order correlation matrix, we need to have a data sequence that is much longer so we can observe the behavior of $(m+1)$-dimensional sample vectors.

A natural way to estimate the correlation values is to look at a time average or sample average autocorrelation of the data. *If* the underlying random process is stationary and ergodic, these time averages will

converge with probability one to the correlations defined as expectations. Before formulating these estimates, a few points relating to the underlying math and to practice are appropriate. First, using sample averages to estimate underlying probabilistic expectations is most easily justified in the stationary and ergodic case, but neither stationarity nor ergodicity are required for this approach to be useful. The key point is that if sample averages converge, which they will if the process is *asymptotically mean stationary* [49, 51], then they converge to the probabilistic average of an ergodic component of a stationary process called the *stationary mean* of the process. The stationary mean averages out most real-world nonstationarities, including initial conditions and block effects. The basic results stay the same, but the details get more complicated. To stay simple, we focus on the stationary and ergodic case with the caveat that most of the conclusions remain useful even when these properties are lacking.

There are two natural choices for estimating correlations based on sample averages. One choice is to take advantage of the stationarity and estimate the correlation $R_X(k,j) = R_X(k-j)$ as a function only of the lag and to estimate the correlation function and matrix as

$$\hat{R}_X(k) = \frac{1}{n}\sum_{l=m}^{n-1} X_l X_{l-|k|} \tag{3.15}$$

$$\hat{R}^{(m+1)} = \{\hat{R}_X(i-j);\ i,j = 0,1,\ldots,m\}. \tag{3.16}$$

The law of large numbers would suggest a normalization of $1/(n-m)$ and this choice would yield an unbiased estimator, but the biased estimator with a normalization of $1/n$ will yield a smaller error variance and hence is preferred in practice. This modification is further justified for two reasons. First, the normalization makes no difference in the solution to the LP problem. Second, the law of large numbers will still hold because the finite m will make no difference in the limit.

Alternatively, one might have less faith in stationarity and estimate the correlation as a function of two time variables and not just their

difference:

$$\overline{R}_X(i,j) = \frac{1}{n}\sum_{l=m}^{n-1} X_{l-i}X_{l-j}; \tag{3.17}$$

$$\overline{R}^{(m+1)} = \{\overline{R}_X(i,j);\ i,j = 0,1,\ldots,m\}. \tag{3.18}$$

These sample-based estimates of the correlation can then be "plugged into" the solution to the LP(m) problem.

There are computationally efficient ways to solve the above equations or the equivalent LP(m) problem and hence to find the optimal linear predictor and the minimum mean squared error. In general one can use the Cholesky decomposition. If X_n is stationary, then $R^{(m+1)}$ is a Toeplitz matrix ($R_X(i,j) = R_X(j-i)$ and hence all entries on a common diagonal are equal, see, e.g., [60, 57]). In this case one can use the much simpler Levinson or Levinson-Durbin method. The $\overline{R}^{(m+1)}$ estimator is not Toeplitz and Cholesky is appropriate. In speech this is called the *covariance method* for historical reasons. The Toeplitz estimator $\hat{R}^{(m+1)}$ is amenable to the Levinson method, which in speech is called the *autocorrelation method*. The autocorrelation method has the additional extremely useful practical advantage that the linear filters it produces are stable, while the covariance method may produce unstable filters.

If the underlying process is stationary and ergodic, then as $n \to \infty$, then the matrices $\overline{R}^{(m+1)}$ and $\hat{R}^{(m+1)}$ will be asymptotically equal. For finite n, however, the two approaches can have differing behavior and both have their strong advocates. It is safe to say, however, that the autocorrelation method is significantly simpler, but the covariance seems to work better for small datasets where the sample averages are less trustworthy estimates of correlation.

3.3 Processes and Linear Filters

The next step is to assume that the random vectors are samples produced by a random process $\{X_n\}$, which means that we have a consistent family of joint probability density functions (PDFs) $f_{X^n}(x^n)$, where as before $X^n = (X_0, X_1, \ldots, X_{n-1})$, $n = 1, 2, \ldots$. There is a technical detail that crops up from time to time. The process

can be considered to be one-sided or unilateral by only allowing nonnegative integers n, in which case the process is often denoted $\{X_n; n = 0, 1, \ldots\}$, or we might consider a bilateral or two-sided process $\{X_n; n = \ldots, -1, 0, 1, \ldots\}$. In the latter case a complete description of the process is more involved since we need a consistent family of PDFs for the random vectors $(X_k, X_{k+1}, \ldots, X_{k+n-1})$ for all dimensions n and starting times k. If the process is assumed to be stationary, then the PDFs do not depend on the starting times and the one-sided process PDFs imply the two-sided PDFs. The annoying detail is that in the one-sided case a stationary process can be put through a linear time-invariant (LTI) filter and the output will not be stationary. For example, suppose that we have an LTI filter with unit sample response h_k which is nonzero only for $k = 0, 1, \ldots, m$. In the two-sided case the output is

$$Y_n = \sum_{k=0}^{m} h_k X_{n-k}, \text{ all } n$$

and a stationary input implies a stationary output. In the one-sided case, however, we have that

$$Y_n = \sum_{k=0}^{\min(n,m)} h_k X_{n-k} = \begin{cases} \sum_{k=0}^{n} h_k X_{n-k} & n = 0, 1, \ldots, m-1 \\ \sum_{k=0}^{m} h_k X_{n-k} & n \geq m \end{cases}.$$

This same behavior occurs if we use a two sided model, but initiate the model with $X_n = 0$ for negative n. Thus the output is *not* stationary, but it is asymptotically stationary in the sense that the problem lies in the initial conditions, and these wash out with time. Specifically, for $n \geq m$ the outputs of the two models are the same and are described by a stationary process. This detail crops up on occasion as one or the other of the two models may yield the simpler analysis, and asymptotically it makes no difference which is used. The difference in difficulty can sometimes be significant, so here we will pick the simpler model for the intended result. The same approach works even when the unit sample response is not finite, but converges in some suitable sense. In this case the output process will still be asymptotically stationary.

As stated earlier, If a process X_n is stationary, then the autocorrelation function simplifies to the form $R_X(k, j) = R_X(j - k)$ and the

autocorrelation function and the corresponding correlation matrices R^m are Toeplitz, and much of the theory of stationary (and asymptotically stationary) random processes derives from the theory of Toeplitz matrices (see, e.g., [60, 57]).

Having designed an optimal linear predictor for the mth sample given its m predecessors, we can apply the same operation at any time n to find a linear least squares prediction of X_n based on the m values preceding it:

$$\hat{X}_n = -\sum_{l=1}^{m} a_l X_{n-l} \tag{3.19}$$

(or the corresponding one-sided sum). This turns the previous static, "one-shot," result into a dynamical system, where at each time we predict what will happen next given what we have recently seen. The convolution of (3.19) allows us to view the prediction $\hat{X}_n$ as being produced by passing the input process X_n through a *prediction filter* described by a unit sample (Kronecker delta $\delta_k = 1$ if $k = 0$ and 1 otherwise) response

$$\pi_k = \begin{cases} 0 & k \le 0 \\ -a_k & k = 1, 2, \ldots, m \end{cases}$$

The previous formulation of the LP(m) problem yields the optimal a along with many properties, some of which simplify if the process is stationary. Summarizing several key points:

$$\hat{X}_m = -\overline{a}_m^t X^m = -(a_m \ldots, a_2, a_1) X^m = -\sum_{l=1}^{m} a_l X_{m-l}$$

where

$$\overline{a}_m^t = = (R_X(m), R_X(m-1), \ldots, R_X(1)) {R^{(m)}}^{-1} \tag{3.20}$$

$$\alpha_m = \overline{a}^t R^{(m+1)} \overline{a} = a^t R^{(m+1)} a = \frac{\det R^{(m+1)}}{\det R^{(m)}} \tag{3.21}$$

$$a = (a_0 \stackrel{\Delta}{=} 1, a_1, \ldots, a_m)^t. \tag{3.22}$$

Furthermore,

$$\alpha_m = R_X(0) + \sum_{k=1}^{m} a_k R_X(-k) \tag{3.23}$$

$$-\sum_{k=1}^{m} a_k R_X(i-k) = R_X(i); i = 1, \ldots, m. \tag{3.24}$$

As observed earlier, given the $m+1$ correlation values $(R_X(0) = \sigma_X^2, R_X(1), \ldots, R_X(m))$, we have $m+1$ linear and linearly independent equations in $m+1$ unknowns $\alpha_m, a_1, \ldots, a_m$ and hence we can solve the equations for the unknowns. Now observe that the converse is true, if we are given the $m+1$ parameters $\alpha_m, a_1, \ldots, a_m$, then we have $m+1$ linear equations in the $m+1$ unknowns $R_X(0), R_X(1), \ldots, R_X(m)$ (symmetry assuring that $R_X(k) = R_X(-k)$). Thus knowing a and α_m we can construct the correlations $R_X(0), R_X(1), \ldots, R_X(m)$ which satisfy (3.24-3.24) for the given a.

The predictor and the resulting linear least squares estimation error $E(\epsilon_n^2) = E[(X_n - \hat{X}_n)^2] = \alpha_m$ are the same for all n if the process is stationary. We can write the resulting error process as

$$\epsilon_n = X_n - \hat{X}_n = \sum_{k=0}^{m} a_k X_{n-k} \tag{3.25}$$

and hence the prediction error process can be viewed as the output of a linear time invariant filter with unit sample response a_k and input X_n. The process ϵ_n is also called the *residual* process. The linear filter described by a is called the *prediction error filter* or the *inverse filter* for the process. The prediction error filter a_k is related to the prediction filter π_k simply as

$$a_k = \delta_k - \pi_k = \begin{cases} 1 & k = 0 \\ -\pi_k & \text{otherwise} \end{cases}.$$

The system is depicted in Figure 3.1.

Fix $n = 0$, for example, and consider the behavior of the optimal linear predictor and the resulting mean squared error, α_m, for X_0 based on the previous m samples, $X_{-m}, \ldots, X_{-1}$ as a function of m. The α_m form a nonincreasing sequence in m since if $k > m$, the constraint set

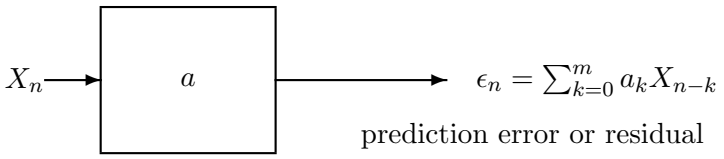


Fig. 3.1 Prediction Error Filter

over which the optimization is performed is larger for α_k then for α_m (the smaller m forces the higher order regression coefficients to be 0) and hence the minimization must yield a smaller or at least no larger value. Since the sequence is nonnegative and nonincreasing, it must have a limit which equals the infimum, say

$$\alpha_\infty = \inf_m \alpha_m = \lim_{m\to\infty} \alpha_m. \tag{3.26}$$

Limiting arguments can be used to show that α_∞ is also the linear least squares prediction error given all predictors of the form

$$\hat{X}_n = -\sum_{k=0}^{\infty} a_k X_{n-k},$$

that is,

$$\alpha_\infty = \inf_{a:a_0=1} E\left(\left[\sum_{k=0}^{\infty} a_k X_{n-k}\right]^2\right), \tag{3.27}$$

where the infimum is over all infinite sequences $a = (a_0, a_1, a_2, \cdots)$ for which $a_0 = 1$ (and for which the convolution sum makes sense). The infimum on the right has to be smaller than any of the α_m so that the result is proved by showing that the performance of any infinite order filter can be asymptotically achieved by a sequence of finite filters.

The trivial special case of $m = 0$ leads to a useful bound. In this case X_0 must be estimated based on no knowledge of its predecessors, that is, the predictor is simply a constant. It is well known that the mean squared error estimate of a random variable with no prior knowledge is the mean and that

$$\alpha_0 = \sigma_X^2. \tag{3.28}$$

Since α_m is nonincreasing, this yields the simple bounds

$$\sigma_X^2 \geq \alpha_m \geq \alpha_\infty \text{ all } m. \tag{3.29}$$

Thus the variance of the sample X_0 represents the *worst* possible mean squared error in a linear prediction system for a process X_n. A sufficient condition for this worst case to hold is that the process be uncorrelated and stationary. In this case the autocorrelation is given by

$$r_k(0) = \begin{cases} \sigma_X^2 & k = 0 \\ 0 & k \neq 0. \end{cases} \tag{3.30}$$

In this case $R^{(m)}$ becomes a diagonal matrix with diagonal elements σ_X^2, and hence for any m,

$$\alpha_m = \sigma_X^2. \tag{3.31}$$

Thus uncorrelated and stationary are sufficient condtitions for $\alpha_\infty = \alpha_m = \sigma_X^2$. We shall see that lack of correlation is both necessary and sufficient for this worst case where linear prediction offers no improvement over just guessing the mean.

3.4 Frequency Domain

Several of the previous ideas have useful expressions in the frequency domain, and these expressions lead to further properties. For the moment assume the usual stationary two-sided process model for simplicity.

Given an LTI filter discribed by a causal unit sample response $a_n; n = 0, 2, 3, \ldots$, the corresponding transfer function $A(f)$ is discrete-time Fourier transform of the unit sample response:

$$A(f) = \sum_{n=0}^{\infty} a_n e^{-i2\pi nf}.$$

A prediction error filter can be described either by the unit sample response a or by the system function A. If Π is the system function corresponding to a prediction filter with sample response π_k, then $A(f) = 1 - \Pi(f)$. Our focus will be on the prediction error filter rather

than the prediction filter, but occasionally it is useful to recall the connection.

If a stationary process X_n has an autocorrelation function $R_X(k)$, define the power spectral density as usual as the Fourier transform of the autocorrelation:

$$S_X(f) = \sum_{k=-\infty}^{\infty} R_X(k) e^{-j2\pi fk}. \tag{3.32}$$

The Fourier inversion formula yields the autocorrelation function in terms of the power spectral density,

$$R_X(k) = \int_{-1/2}^{1/2} S_X(f) e^{j2\pi fk} \frac{df}{2\pi}. \tag{3.33}$$

Using standard linear systems theory second order input/output analysis, the power spectral density of ϵ_n is

$$S_\epsilon(f) = S_X(f)|A(f)|^2 \tag{3.34}$$

and hence another way of writing the mean squared error is

$$E(\epsilon_n^2) = \int_{-1/2}^{1/2} S_\epsilon(f) df = \int_{-1/2}^{1/2} S_X(f)|A(f)|^2 df.$$

This formula provides a means of describing the LP(m) problem in the frequency domain:

$$\alpha_m = \min_{a:a_0=1} \int_{-1/2}^{1/2} S_X(f)|A(f)|^2 df, \tag{3.35}$$

where

$$A(f) = \sum_{n=0}^{m} a_n e^{-i2\pi nf}. \tag{3.36}$$

In the light of (3.27), we allow $m = \infty$. This minimization crops ups in mathematics, especially in the study of orthogonal polynomials (see, e.g., Grenander and Szego [60]).

A remarkable fact is that the asymptotic optimal one-step linear prediction mean squared error can be expressed in terms of the power

spectral density of a process by

$$\begin{aligned}\alpha_\infty &= \lim_{m\to\infty} \alpha_m = \lim_{m\to\infty} \frac{\det R^{(m+1)}}{\det R^{(m)}} \\ &= \lim_{m\to\infty} \left(\det R^{(m)}\right)^{1/m} = e^{\int_{-1/2}^{1/2} \ln S_X(f) df}. \end{aligned} \tag{3.37}$$

See, e.g., [57] for a proof using the asymptotic properties of Toeplitz matrices. This fact provides a variation on (3.29). Since the logarithm is concave, it follows from Jensen's inequality that

$$\sigma_X^2 = \int_{-1/2}^{1/2} S_X(f) df \geq e^{\int_{-1/2}^{1/2} \ln S_X(f) df} = \alpha_\infty, \tag{3.38}$$

with equality if and only if $S_X(f) = \sigma_X^2$ for all f. Thus if $\sigma_X^2 = \alpha_\infty$, then the power spectral density is a constant and hence the process is uncorrelated. Coupled with (3.31), this implies that uncorrelation of a random process is a necessary and sufficient condition for the worst case of maximal one-step prediction error.

A variation on this theme gives a far more important result. Suppose now that a process X_n is put through a prediction error filter A to form a prediction error process $\epsilon_n^{(A)}$ which will have power spectral density $S_{\epsilon^{(A)}}(f) = |A(f)|^2 S_X(f)|$. Further suppose that we now apply the LP(m) problem to the prediction error process, that is, we wish to solve the linear filter optimization

$$\alpha_\infty(A) = \inf_{b:b_0=1} \int_{-1/2}^{1/2} S_{\epsilon^{(A)}}(f)|B(f)|^2 df \tag{3.39}$$

where

$$B(f) = \sum_{n=0}^{\infty} b_n e^{-i2\pi nf}. \tag{3.40}$$

Applying the argument of (3.38) to the prediction error ϵ_n process instead of to the original process X_n we have that

$$\int_{-1/2}^{1/2} S_{\epsilon^{(A)}}(f) df \geq \alpha_\infty(A) = e^{\int_{-1/2}^{1/2} \ln S_{\epsilon^{(A)}}(f) df}, \tag{3.41}$$

with equality if and only if

$$S_{\epsilon^{(A)}}(f) = \sigma_{\epsilon^{(A)}}^2.$$

Since the convolution of two unit sample responses with leading coefficients 1 will be another unit sample response with leading coefficient 1, we have that

$$\begin{aligned}
\int_{-1/2}^{1/2} S_{\epsilon^{(A)}}(f)df &\geq e^{\int_{-1/2}^{1/2} \ln S_{\epsilon^{(A)}}(f)df} \\
&= \alpha_\infty(A) \\
&= \inf_{b:b_0=1} \int_{-1/2}^{1/2} S_X(f)|A(f)B(f)|^2 df \\
&\geq \inf_{a:a_0=1} \inf_{b:b_0=1} \int_{-1/2}^{1/2} S_X(f)|A(f)B(f)|^2 df \\
&\geq \inf_{c:c_0=1} \int_{-1/2}^{1/2} S_X(f)|C(f)|^2 df \\
&= \alpha_\infty = e^{\int_{-1/2}^{1/2} \ln S_X(f)df}. \qquad (3.42)
\end{aligned}$$

The point of all this is that if you have a linear predictor A for X_n which hits the optimal one-step linear prediction mean squared error, i.e., if

$$E[\epsilon_0^{(A)^2}] = \int_{-1/2}^{1/2} S_{\epsilon^{(A)}}(f)df = \alpha_\infty = e^{\int_{-1/2}^{1/2} \ln S_X(f)df}, \qquad (3.43)$$

then all the middle terms in (3.42) are equal and hence (3.41) holds with equality with both sides equal to α_∞. Simply stated, a necessary condition for a linear filter to solve the LP(m) problem for all filter orders is that the resulting prediction error process be uncorrelated or *white* with variance equal to α_∞. Plugging into (3.42) shows that the condition is also sufficient.

Thus the goal of linear prediction can be thought of as finding a filter that produces as white a prediction error process as possible with the correct variance. This may not be possible for finite m, but a common intuition is that one should make the prediction error *as white as possible* given the constraints.

As a final comment, we have also effectively shown that a Gaussian process is an extreme case in the context of one-step prediction: if you know the optimal linear predictor, no better predictor can be found

even if you allow the more general class of possibly nonlinear predictors. This makes the Gaussian case a best or worse case, depending on your point of view. It is the worst case in the sense that knowing the complete distribution in addition to the autocorrelation does not help, it is the best case in the sense that the simple linear predictor is unbeatable.

4

Autoregressive Modeling

4.1 Linear Prediction and Autoregressive Models

If a prediction error filter described by a or A has a stable inverse, then the error sequence can be inverted to recover the original process, as depicted in Figure 4.1. The figure assumes steady-state conditions,

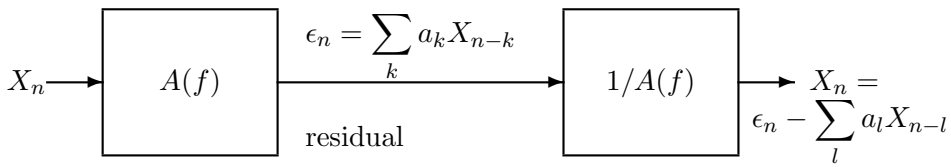


Fig. 4.1 Prediction Error Filter and Inverse

a two-sided stationary process. In practice it may be necessary to consider one-sided models as discussed earlier and hence also the initial conditions of the filter. As earlier discussed, in the one-sided case the output process in general is not stationary and hence either asymptotically stationary conditions must be assumed or time-domain techniques would be needed.

If we emphasize the right half of the figure and invoke the fact that $A = 1 - \Pi$ we can use simple linear systems theory to obtain Figure 4.2.

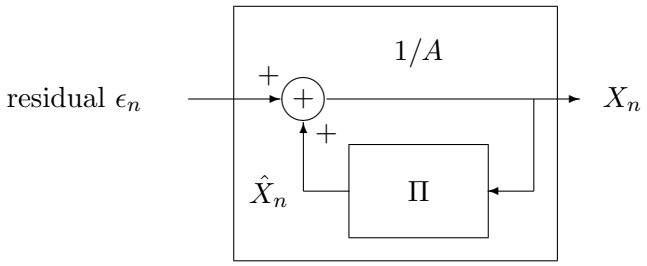


Fig. 4.2 Autoregressive Model

The figures suggests a possible *model* for the random process X_n as the result of linearly filtering another, possibly simpler, process. As we have seen, if a prediction filter is approximately optimum, then the prediction error or residual process should be approximately white. A process resulting from a casual LTI of the form $1/A$ driven by a white input process is called an *autoregressive process*. (We will be more precise shortly.) If A has order m, e.g., if it is the optimal mth order one-step prediction error filter for X_n, then the process is called an mth order autoregressive process. As shall be seen, these processes play a fundamental role in speech processing applications (as they do in many other signal processing applications). If the inverse of the prediction error filter driven by an approximately white process gives the exact process, this suggests that a good model might result by driving the same filter with a truly white process. In speech terms, this is an intuitive argument supporting a synthesis method — find the LP parameters for a segment of speech, then send a digitized version to the receiver. The receiver uses a local white noise source to drive the inverse prediction error filter, which should have a spectrum close to, and hence sound like, the original. So far this is all hand-waving, but we shall see several means of constructing a more solid argument.

To formalize the definition of an autoregresive process, a process

X_n is said to be an mth order autoregressive (or AR(m)) process if it can be written in the form

$$X_n = -\sum_{k=1}^{m} a_k X_{n-k} + W_n, \tag{4.1}$$

where W_n is an uncorrelated stationary process with variance, say, σ_W^2. This definition is sufficient to determine the second order properties of the process (autocorrelation and power spectral density), but in order to completely describe a process we need to be able to describe the joint PDFs of all orders. This is usually accomplished by strengthening the definition from uncorrelated to IID W_n and describing a stationary marginal density $f_{W_n} = f_W$. In this case (4.1) describes the conditional PDF

$$\begin{aligned} f_{X_n|X^n}(x_n|x^n) &= f_{X_n|X_{n-m},\ldots,X_{n-1}}(x_n|x_{n-m},\ldots,x_{n-1}) \\ &= f_W(\sum_{k=0}^{m} a_k x_{n-k}), \text{ all } n. \end{aligned} \tag{4.2}$$

To determine the complete family of joint PDFs to specify the process we either need to specify an m-dimensional marginal PDF $f_{X^m}(x^m)$ consistent with the stationary distributions, or we need to specify initial conditions, e.g., set $X_n = 0$ for $n \leq 0$. The first approach yields a stationary process, the second yields an asymptotically stationary process since the initial conditions eventually wipe out. Both approaches have their uses. In either case, the formula for the conditional PDF implies that the process is an m-th order Markov process. If W_n is an IID Gaussian process and one assumes a stationary PDF for X^m or sets initial conditions, then the process X_n is also Gaussian.

We have not yet made any rigorous connection between linear prediction and modeling, we have simply pointed out that linear prediction suggests a connection with autoregressive models. There are, however, several simple and important implications of the autoregressive structure for linear prediction. The remainder of this chapter collects several of these properties.

4.2 Linear prediction of AR(m) processes

We begin with some linear algebra to describe the process for the zero initial condition case. This representation yields significantly simpler proofs for several important results than does the usual stationary process representation. Define the autoregressive process by

$$X_n = \begin{cases} -\sum_{k=1}^{m} a_k X_{n-k} + W_n & n = 0, 1, 2, \ldots \\ 0 & n < 0. \end{cases} \tag{4.3}$$

and define the $n \times n$ Toeplitz lower triangular triangular matrix A_n by

$$A_n = \begin{bmatrix} 1 & & & & & \\ a_1 & 1 & & & & 0 \\ \vdots & a_1 & 1 & & & \\ a_m & & \ddots & \ddots & & \\ & \ddots & & & & \\ 0 & & a_m & \ldots & a_1 & 1 \end{bmatrix} \tag{4.4}$$

so that (4.3) can be expressed concisely as $A_n X^n = W^n$. We have with a little more linear algebra that

$$\begin{aligned} R_W^{(n)} &= E[W^n W^{nt}] = \sigma_W^2 I \\ &= E[A_n X^n X^{nt} {A_n}^t] = A_n E[X^n X^{nt}] {A_n}^t \\ &= A_n R_X^{(n)} A_n^t \end{aligned}$$

so that

$$R_X^{(n)} = \frac{1}{\sigma_W^2} A_n^{-1} {A_n^t}^{-1} = \frac{1}{\sigma_W^2} A_n^{-1} {A_n^{-1}}^t \tag{4.5}$$

and hence

$${R_X^{(n)}}^{-1} = \sigma_w^2 A_n^t A_n. \tag{4.6}$$

We can also express this by

$${R_X^{(n)}}^{-1} = \sigma_W^2 T_n$$

where $T_n = A_n^t A_n = \{t_{k,j}; k, j = 0, 1, \ldots, n-1\}$ with

$$t_{k,j} = \sum_{l=0}^{\min(k,j)} a_{l-k} a_{l-j}. \tag{4.7}$$

Before reaping the benefits of this representation, recall the discussion of stationary vs. nonstationary models. It is clear from (4.5) that $R_X^{(n)}$ is not a Toeplitz matrix and that hence X_n is not a stationary random process. This might cause worry that the frequency domain approach and results would not be applicable in this case. Fortunately, however, there is an easy solution. From (4.6)-(4.7), the *inverse* correlation matrix ${R_X^{(n)}}^{-1}$ is almost Toeplitz, all of its entries except in the upper left $m \times m$ square depend only on the difference of the row and column entry. As a result, it turns out that the matrix ${R_X^{(n)}}^{-1}$ is asymptotically Toeplitz as n grows, and hence so is the autocorrelation $R_X^{(n)}$. In particular, the behavior of the process essentially the same (as n grows) as if it were stationary with autocorrelation function [57]

$$R_X(k) = \frac{1}{\sigma_W^2} \int_{1/2}^{1/2} \frac{1}{|A(f)|^2} e^{j2\pi fk}, \tag{4.8}$$

that is, it behaves like a process with power spectral density

$$S_X(f) = \sigma_W^2 / |A(f)|. \tag{4.9}$$

Fixing the order m of the AR model we have for any positive integer $n \geq m$

$$\alpha_n = \frac{\det R^{(n+1)}}{\det R^{(n)}},$$

but

$$\begin{aligned} \det R^{(n)} &= \frac{1}{\det {R^{(n)}}^{-1}} = \frac{\sigma_W^{2n}}{\det A_n^t A_n} \\ &= \frac{\sigma_W^{2n}}{\det A_n \times \det A_n} = \sigma_W^{2n} \end{aligned}$$

since the determinant of a triangular matrix with 1s on the diagonal is 1. Thus for an AR(m) process initiated with zero initial conditions we have that

$$\alpha_n = \sigma_W^2, n = 1, 2, \ldots \tag{4.10}$$

Thus we have immediately that the minimum MSE achievable for any linear predictor of order greater than m is the same as that achievable

by a linear predictor of order m; that is, $\alpha_n = \alpha_m$ for all $n \geq m$! This means in turn that for an AR(m) process,

$$\alpha_\infty = \lim_{n\to\infty} \alpha_n = \alpha_m. \tag{4.11}$$

This simple result does not quite hold if instead the autoregressive process is initiated with a stationary distribution on m-tuples rather than the 0 initial conditions (and hence the resulting autoregressive process is stationary). In particular, in this case (4.6) does not hold exactly, but it does hold everywhere in the matrix except for the upper left hand $m\times m$ submatrix, which has diminishing effect as n grows. The theory of Toeplitz matrices ([60, 57]) implies that the two matrices are asymptotically equivalent and have the same determinants and hence for either the 0 initial condition or stationary random initial conditions,

$$\alpha_\infty = \lim_{n\to\infty} \alpha_n = \alpha_m = \sigma_W^2. \tag{4.12}$$

Furthermore, even in the nonstationary case one can meaningfully define the spectral density of the autoregressive source as the Fourier transform of the limiting autocorrelation function (which is the same as that resulting from the random initial conditions) as

$$S_X(f) = \frac{\sigma_W^2}{|A(f)|2} \tag{4.13}$$

and, in particular,

$$\alpha_\infty = e^{\int_{-1/2}^{1/2} S_X(f)|A(f)|^2}. \tag{4.14}$$

4.3 Correlation matching

Now suppose that we do not know that X_n is an autoregressive process, but we know (or estimate) $m+1$ values of its autocorrelation, $R_X(k); k = 0, 1, \ldots, m$. As suggested above, we can use the solution of the LP(m) problem to provide a *model* for the process as an mth order autoregressive process by constructing an autoregressive process, say Y_n as in (4.1) using the same a and a variance $\sigma_W^2 = \alpha_m$. In speech processing, a model might be used by a receiver to *synthesize* the speech, for example, by simulating a Gaussian autoregressive process with specified values of the autocorrelation. As we have seen, the

correlation of the autoregressive process Y_n will satisfy the Yule-Walker equations with parameters $a, \sigma_W^2 = \alpha_m$, but these parameters in turn satisfy the Yule-Walker equations in terms of the correlation R_X. Thus the two correlations must be the same up to lag m:

$$R_Y(k) = R_X(k); k = 0, 1, 2, \ldots. \tag{4.15}$$

This is called the *correlation matching* property [16, 104] of an AR(m) model fit to m correlation values: the model will have correlations equal to the given values for all lags up to m. For larger lags the correlations of the process X_n and its AR(m) model will in general differ (unless, of course, X_n is itself an AR(m) model).

Correlation matching provides one justification for picking a particular AR(m) model given observed data from the source. If one estimates the correlations up to lag m, then the AR(m) model provided by the LP(m) parameters agrees with the known correlation values and extends these values to provide an autocorrelation for all values. In other words, the given m values yield a prediction error filter A and a minimum MSE α_m. From (4.13) the corresponding AR(m) model will have power spectral density equal to

$$S_Y(f) = \frac{\alpha_m}{|A(f)|^2}$$

and hence autocorrelation function equal to the inverse Fourier transform

$$R_Y(k) = \int_{-1/2}^{1/2} \frac{\alpha_m}{|A(f)|^2} e^{j2\pi fk}$$

with

$$R_Y(k) = R_X(k), k = 0, 1, \ldots, m.$$

This is called the *autoregressive extension* of the known correlation values.

5

Maximum Likelihood

Suppose that it is known (or assumed) that a process X_n is an order m Gaussian autoregressive process as in (4.1), but we do not know a or σ^2. Say we observe a sequence of samples $X^n = X_0, \ldots, X_{n-1} = x^n$. The *maximum likelihood estimate* of a, σ^2 given $X^n = x^n$ is defined as the values of a and σ^2 that maximizes the conditional probability density function of X^n given a and σ^2, $f_{X^n;a,\sigma^2}(x)$. To keep things simple, assume that the process is initialized so that (4.3) holds. Defining the triangular matrices A_n as in (4.4), then from (4.6) the Gaussian probability density function given the a, σ^2 is

$$f_{X^n;a,\sigma^2}(x^n) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} e^{-\frac{1}{2\sigma^2} x^{n^t} A_n^t A_n x^n}. \tag{5.1}$$

Since log is a monotonic function, maximizing (5.1) is equivalent to minimizing the negative of the logarithm, removing constant additive and multiplicative terms, and dividing by n; that is, minimize

$$\log \sigma^2 + \frac{1}{n\sigma^2} x^{n^t} A_n^t A_n x^n.$$

Application of the $\log r \leq r - 1$ inequality immediately yields that

$$\log \sigma^2 + \frac{1}{n\sigma^2} x^{n^t} A_n^t A_n x^n \geq \log \frac{x^{n^t} A_n^t A_n x^n}{n}, \tag{5.2}$$

with equality if

$$\sigma^2 = \frac{x^{n^t} A_n^t A_n x^n}{n}. \tag{5.3}$$

So the minimization is accomplished in two steps: first find A_n to minimize $x^{n^t} A_n^t A_n x^n / n$, and then equate σ^2 to the resulting minimum, i.e., use (5.3). Writing out the column vector

$$y = A_n x^n = \begin{pmatrix} x_0 \\ a_1 x_0 + x_1 \\ a_2 x_0 + a_1 x_1 + x_2 \\ \vdots \\ \sum_{i=0}^m a_i x_{m-i} \\ \sum_{i=0}^m a_i x_{m+1-i} \\ \vdots \\ \sum_{i=0}^m a_i x_{k-i} \\ \vdots \\ \sum_{i=0}^m a_i x_{n-1-i} \end{pmatrix}$$

it is clear that all of the terms except for the first $m-1$ terms have the same form, that is, $\sum_{i=0}^m a_i x_{k-i}$ for $k = m, \ldots, n-1$. Thus for large n the initial terms will form a negligable contribution and

$$\begin{aligned}
\frac{x^{n^t} A_n^t A_n x^n}{n} &= \frac{1}{n} \sum_{k=0}^{m-1} \left(\sum_{i=0}^k a_i x_{k-i} \right)^2 + \frac{1}{n} \sum_{k=m}^{n-1} \left(\sum_{i=0}^m a_i x_{k-i} \right)^2 \\
&\approx \frac{1}{n} \sum_{k=m}^{n-1} \sum_{i=0}^m \sum_{j=0}^m a_i a_j x_{k-i} x_{k-j} \\
&= \frac{1}{n} \sum_{k=m}^{n-1} \sum_{i=0}^m \sum_{j=0}^m a_i a_j x_{k-i} x_{k-j} \\
&= \sum_{i=0}^m \sum_{j=0}^m a_i a_j \frac{1}{n} \sum_{k=m}^{n-1} x_{k-i} x_{k-j} \\
&= \sum_{i=0}^m \sum_{j=0}^m a_i a_j \overline{R}_X(i,j) \\
&= a^t \overline{R}_{m+1} a.
\end{aligned}$$

Thus maximizing the likelihood is equivalent to solving the LP(m) problem for the sample estimate correlation matrix $\overline{R}_{m+1}$ and then setting $\sigma^2 = \alpha_m$, the MSE resulting from the LP(m) problem. Since n is assumed large, we can also make the approximation $\overline{R}_{m+1} \approx \hat{R}_{m+1}$ and use the autocorrelation method for a solution.

The conclusion is that maximizing the likelihood $f_{X^n;a,\sigma^2}(x^n)$ is approximately equivalent to solving $\text{argmin}_{a:a_0=1}\, a^t \hat{R}^{(m+1)} a$, that is, to solving the LP(m) problem with the plug-in stationary sample average estimate of the autocorrelation! Here the LP(m) method produces a model or density estimate, an mth order Gaussian autoregressive process fit to the data.

An important aspect of this approach is that it makes the assumption that the observed data is in fact Gaussian, an assumption that is certainly questionable if the data is sampled speech. On the other hand, the algorithm can be applied to any signal, including speech, and it will produce a Gaussian model. In the speech case, this model could be used to synthesize speech.

6

Maximum Entropy

Maximum entropy was introduced as an inference principle based on prior knowledge or observed data by E.T. Jaynes [76]. Although it has had its controversy, it has led to mathematical generalizations (minimum relative entropy or Kullback-Leibler divergence) that have yielded both interesting mathematics and applications. See, in particular, the classic book by Kullback [80] and the articles by Shore and Johnson [118], Lev-Ari et al. [82], and Jones [77] and the references therein for a description of the theory and several applications including speech. We do not dwell here on the motivation for the method past a brief comment, only on its mechanics and its relevance to the topic at hand. The comment is that the various typesthe differential entropy rate of a random process can be interpreted as a measure of how random a process is — the larger the entropy rate, the more random the process is. Thus choosing a model based on known constraints (in our case, on the correlation) is, in a sense, choosing the most random possible model consistent with the constraints.

Suppose that we have an estimate $\hat{R}^{(m+1)}$ of correlations to lag m of a stationary random process X_n. Thus, in particular, $\hat{R}^{(m+1)}$ is a Toeplitz matrix. What mth order Markov random process maximizes

the Shannon differential entropy rate defined by

$$h(X) = \lim_{n\to\infty} \frac{1}{n} h(X^n)$$

where

$$h(X^n) = -\int f_{X^n}(x^n) \log f_{X^n}(x^n)\, dx^n \,?$$

Unlike the maximum likelihood method, this approach makes no *assumptions* regarding the form of the underlying probability density functions. In particular, no Gaussian assumptions are made.

As do Cover and Thomas [36], accomplish the maximization in two steps. We take additional shortcuts, however. In particular, we do not use Lagrangian multipliers or variational methods to show that the Gaussian PDF maximizes the differential entropy, and we do not draw as heavily on information theory to show that the AR(m) process maximizes the differential entropy for a Gaussian process with $m+1$ correlation constraints.

First, suppose that X^n and Y^n are two zero mean random vectors with identical correlation matrices $R^{(n)}$ and that Y^n is Gaussian. We shall show that

$$h(X^n) \le h(Y^n) = \frac{1}{2} \log(2\pi e \det R^{(n)}).$$

Begin by writing

$$\begin{aligned} h(Y^n) &= -\int dy^n f_{Y^n}(y^n) \log f_{Y^n}(y^n) \\ &= -\int dy^n \frac{e^{-\frac{1}{2} y^{n^t} R^{(n)^{-1}} y^n}}{2\pi^{n/2} \det R^{(n)^{1/2}}} \log \left(\frac{e^{-\frac{1}{2} y^{n^t} R^{(n)^{-1}} y^n}}{(2\pi)^{n/2} R^{(n)^{1/2}}} \right) \\ &= \log \left(2\pi^{n/2} \det R^{(n)^{1/2}} \right) + \\ &\quad \int dy^n \frac{e^{-\frac{1}{2} y^{n^t} R^{(n)^{-1}} y^n}}{(2\pi)^{n/2} \det R^{(n)^{1/2}}} \frac{1}{2} y^{n^t} R^{(n)^{-1}} y^n. \end{aligned} \tag{6.1}$$

From linear algebra, for an n-dimensional vector x and an $n \times n$ matrix R,

$$x^t R x = \mathrm{Tr}(R x x^t). \tag{6.2}$$

Thus the final term in (6.1) can be written as

$$\begin{aligned} E\left[\frac{1}{2}Y^{nt}{R^{(n)}}^{-1}Y^n\right] &= \frac{1}{2}\,\mathrm{Tr}\left({R^{(n)}}^{-1}E\left[Y^nY^{nt}\right]\right) \\ &= \frac{1}{2}\,\mathrm{Tr}\left({R^{(n)}}^{-1}R^{(n)}\right) = \frac{n}{2}. \end{aligned} \tag{6.3}$$

This proves that

$$h(Y^n) = \frac{n}{2}\log\left(2\pi e\left(\det R^{(n)}\right)^{1/n}\right). \tag{6.4}$$

Next, the quantity

$$H(f_{X^n}\|f_{Y^n}) = \int dx^n f_{X^n}(x^n)\log\frac{f_{X^n}(x^n)}{f_{Y^n}(x^n)}$$

is known as the *relative entropy* (or *Kullback-Leibler divergence* or *cross entropy*) of the two PDFs. It follows immediately from Jensen's inequality (or the fact that $\log r \le r-1$) that

$$\int dx^n f_{X^n}(x^n)\log\frac{f_{X^n}(x^n)}{f_{Y^n}(x^n)} \ge 0 \tag{6.5}$$

with equality if and only if the two PDFs are the same (except possibly for a set of Lebesgue measure zero). This result is called the *divergence inequality*. Applying the inequality to our case where X^n and Y^n have equal correlation matrices and Y^n is Gaussian, we have that

$$\begin{aligned} 0 &\le -\int dx^n f_{X^n}(x^n)\log f_{Y^n}(x^n) - h(X^n) \\ &= -\int dx^n f_{X^n}(x^n)\log\left(\frac{e^{-\frac{1}{2}{y^n}^t{R^{(n)}}^{-1}y^n}}{2\pi^{n/2}\det {R^{(n)}}^{1/2}}\right) - h(X^n) \\ &= \log\left((2\pi)^{n/2}\det {R^{(n)}}^{1/2}\right) - \frac{1}{2}E[Y^{nt}{R^{(n)}}^{-1}Y^n] - h(X^n). \end{aligned}$$

By the same argument used in (6.3), the expectation of the quadratic form is just $n/2$ (the argument only depends on the common correlation, not whether or not the distribution with respect to which the expectation is taken is Gaussian). Thus

$$0 \le \frac{n}{2}\log\left(2\pi e\left(\det R^{(n)}\right)^{1/n}\right) - h(X^n) = h(Y^n) - h(X^n), \tag{6.6}$$

so that $h(Y^n) \geq h(X^n)$ as claimed. This shows that if two processes have the same autocorrelation function and one is Gaussian, than the Gaussian process has the larger entropy.

The second part of the proof focuses on on Gaussian processes and shows that if we constrain only $m+1$ values of the autocorrelation function, then the Gaussian process with those values having the largest differential entropy rate is the Gaussian AR(m) process with those values. This coupled with the first part implies that of all processes with these $m+1$ correlation values, the Gaussian AR(m) has the largest entropy rate.

Suppose now that X_n is a stationary Gaussian process with autocorrelation matrices $R_X^{(n)}$ and that Y_n is a stationary Gaussian AR(m) with autocorrelation matrices $R_Y^{(n)}$ process satisfying (4.15), that is, $R_X^{(m)} = R_Y^{(m)}$.

From (6.4), (3.37) the differential entropy rate of X_n must satisfy

$$\begin{aligned} h(X) &= \lim_{n\to\infty} \frac{h(X^n)}{n} = \lim_{n\to\infty} \frac{1}{2} \log \left(2\pi e \left(\det R_X^{(n)} \right)^{1/n} \right) \\ &= \lim_{n\to\infty} \frac{1}{2} \log \left(2\pi e \frac{\det R_X^{(n+1)}}{\det R_X^{(n)}} \right) \frac{1}{2} \log(2\pi e \alpha_\infty) \end{aligned}$$

and the entropy rate of the AR(m) process with the same autocorrelation function up to lag m must satisfy

$$\begin{aligned} h(Y) &= \lim_{n\to\infty} \frac{h(Y^n)}{n} = \lim_{n\to\infty} \frac{1}{2} \log \left(2\pi e \left(\det R_Y^{(n)} \right)^{1/n} \right) \\ &= \lim_{n\to\infty} \frac{1}{2} \log \left(2\pi e \frac{\det R_Y^{(n+1)}}{\det R_Y^{(n)}} \right) = \frac{1}{2} \log(2\pi e \alpha_m) \end{aligned}$$

and hence from from (4.11),

$$h(X) \leq h(Y). \tag{6.7}$$

This upper bound can be achieved by choosing X_n to be the AR(m) process chosen from the LP parameters.

7

Minimum Distance and Spectral Flattening

Suppose we have a process $\{X_n\}$ with autocorrelation R or power spectral density S (or estimates of these quantities). Suppose also that we have a measure of the "distance" $d(R, R_Y)$ or $d(S, S_Y)$ between this original correlation or spectrum and another correlation or spectrum chosen from a class of models such as the class $\mathcal{A}_m$ of all mth order autoregressive processes. The word "distance" is in quotes in this paragraph because it need not be a distance or metric in the mathematical sense, that is, a nonnegative symmetric function that satisfies a triangle inequality such as the Euclidean distance. Here we will use it as a general term quantifying the quality of representing one correlation or spectrum by another. Since it represents a distortion or loss function, we require that it be nonnegative.

Given such a distance there is a natural method for selecting the "best" model available in a class of models for a given correlation: a minimum distance or nearest-neighbor rule, find the model in the class with the smallest distance to the given correlation or spectrum.

An example is the Itakura-Saito distance defined by

$$\begin{aligned} d(S, S_Y) &= \int_{-1/2}^{1/2} \left(\frac{S(f)}{S_Y(f)} - \ln \frac{S(f)}{S_Y(f)} - 1 \right) \\ &= \frac{a^t R^{(m+1)} a}{\sigma_Y^2} - \ln \frac{\alpha_m}{\sigma_Y^2} - 1. \end{aligned} \tag{7.1}$$

This distance was proposed by Itakura and Saito as their "error matching measure" [70, 72, 52], but it also arises in other contexts. It is a special case of Kullback's *minimum discrimination information* for density/parameter estimation [80, 52, 55] and it appears in the early information theory literature as a relative entropy rate between Gaussian processes [113]). So once again the properties of Gaussian processes enter into the analysis of nonGaussian signals. A similar formula arises as the asymptotic likelihood function of a Gaussian process in Merhav and Ephraim [108] (equations (19)–(23)).

The $\ln r \leq r - 1$ inequality implies that the distance is nonnegative and choosing an AR(m) model using the LP(m) formulation minimizes the Itakura-Saito distance resulting in a value of 0.

The distance can be used in another way. Recall that one interpretation of the prediction error filter A was that it should be chosen to "whiten" the prediction error process as much as possible. The degree of success at whitening can be measured by the Itakura-Saito distortion from the truly white prediction error process (resulting from the optimal one-step predictor of infinite order) and the prediction error spectrum from an mth order prediction error filter A, suggesting that we choose A to minimize a "spectral flatness measure" [46]

$$d(S|A|^2, \alpha_\infty) = \int_{-1/2}^{1/2} \left(\frac{S(f)|A(f)|^2}{S_{\alpha_\infty}} - \ln \frac{S(f)|A(f)|^2}{S_{\alpha_\infty}} - 1 \right).$$

This again results in the LP(m) solution.

8

Linear Predictive Coding

Linear prediction methods have been shown to provide a means of fitting an autoregressive model to observed data in the form of correlation estimates, as summarized in Figure 8.1. The figure emphasizes the fact

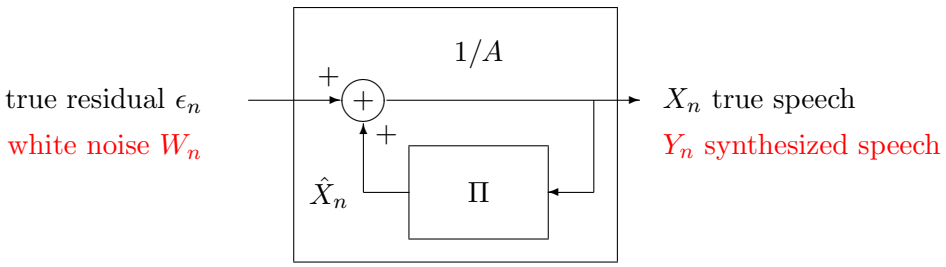


Fig. 8.1 Reconstruction and Modeling

that if the true prediction error or residual is used to drive the inverse of the prediction error filter, then the output is the original waveform. The modeling or synthesis enters when the true residual is replaced by a locally generated white noise, which will produce an output with a variety of properties:

- the output correlation matches the known correlations up to lag m, the order of the model,
- if the observed process were truly Gaussian AR(m), then the model would be a maximum likelihood estimate of the regression coefficients given the observed correlations to lag m,
- the model is the maximum differential entropy rate model with the given constraints,
- the model is the minimum Itakura-Saito distance fit to the observed correlation values by an AR(m) model, and
- the model yields the best spectral flatness of any AR(m) model.

A fundamentally important question that has not yet been mentioned is whether or not the models produced from real speech in this way will in turn synthesize speech that sounds like the original and is understandable. First, the model so far is simplistic and incomplete, not all of speech is well modeled using white noise as an input to autoregressive filters. This does produce good models for unvoiced sounds such as whispered speech, but voiced sounds are better modeled by driving AR filters by periodic pulse trains, suggesting a "switched" model as in Figure 8.2. Producing such a model requires additional complicated

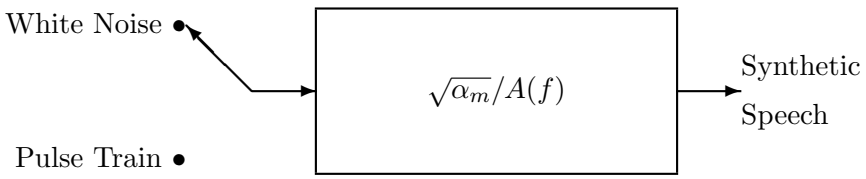


Fig. 8.2 Switched Model

techniques of voicing and pitch extraction. We here keep things simple and do not treat this important aspect of speech synthesis but focus on the single noise-driven model.

So far no coding has been involved since purely continuous or analog computation has been assumed. To communicate a model across a digital link such as a digital network or storage device requires that the model be converted into a digital or discrete representation. This is where quantization (or in the language of Shannon information theory, source coding) comes in. For each segment of time, perhaps 100 ms or a variable length depending on other signal processing, LP analysis is performed on the speech data to produce a model and the parameters describing the model, α_m and a or some equivalent set, are quantized either as a collection of scalars or as a single vector. Quantization is a separate topic with its own literature, and historically the early techniques used for speech coding involved simple quantization, either the implicit quantization in using digital computers (especially array processors) to do the computation or explicit uniform scalar quantization to reduce the bit rate to more manageable values.

Combining the LP analysis (or one of its many forms) with quantization yields the system that has come to be called linear predictive coding (LPC). LPC belongs to the traditional class of voice coders or *vocoders* (for "voice coder"), systems that do not attempt to produce a waveform matching the original in a squared error sense, but instead tried to produce synthesized speech with the correct spectral properties. We shall also look briefly at hybrids that take a middle ground between having a true residual and a locally generated "fake" residual at the decoder using parameters defined by the encoder. While LPC aims at very low bit rate transmission by sending only parameters describing the model, one can combine the model with an *approximation* of the true residual excitation in order to synthesize the reproduction speech. This hybrid produces a waveform coder that uses extra bits in transmission to provide improved quality by exciting the reconstruction filter with an approximation to the true residual. An added advantage to such a system is that it obviates one of the most difficult parts of LPC, detecting and extracting pitch. The hybrid approach also leads naturally to multirate speech coding systems — the LPC bits alone provide a low bit rate system when such is required, and a medium bit rate system results when adding residual bits to the model coder. The emphasis here is on basic LPC, but several approaches to residual

coding will be described as natural extensions and part of the history.

In Part II these techniques are put in the historical context of the development of LP techniques and their use in early packet speech communication.

Part II:

History: LPC and IP

9

1966: On-Line Signal Processing and Statistical Speech Coding

Glen Culler and the
On-Line (Culler-Fried) System

Several threads of the story originate in 1966 with the groundwork for the applications of linear prediction ideas to speech and for realtime signal processing on computer networks — two subjects of strong interest to Glen Culler, then a professor of electrical engineering at the University of California at Santa Barbara (UCSB). Culler's background was in mathematics and his interests included what would later be called computer science, especially the graphical display of signals, signal processing as an educational tool, and the use of networked computers. During 1966 he formally introduced

his On-Line System (OLS or Culler-Fried system), a system combining keyboards and oscilloscopes connected to a central computer (an IBM 360) to allow realtime signal processing at individual student terminals. The system gave students the opportunity to compute and visualize discrete Fourier transforms (DFTs) on real signals such as sampled speech. Culler is also remembered for building fast and effective computer systems. This combination of ideas — realtime signal processing and computer networks — is the common theme in this history. Two graduate students who helped develop the system were Mike McCammon and Dave Retz.

In 1966 Fumitada Itakura was a PhD student at Nagoya University. He had been interested for some time in statistics, mathematics, and signal processing, including developing a maximum likelihood approach to character recognition. He had begun to work on speech recognition through the guidance of Professor Teruo Fukumura of Nagoya University and Shuzo Saito at NTT. In speech processing he found fertile ground for his interests in stochastic processes, estimation, detection, and recognition. He was particularly interested in applying Hájek's ideas of autoregressive modeling [61] to speech as a means for recognizing vowels. The work was first published in a 1966 report of the NTT Electrical Communication Laboratory[115].

The report provided an approach to automatic phoneme discrimination using a statistical approach to speech processing wherein short segments of speech were modeled using Gaussian autoregressive processes. The basic idea was to form a maximum likelihood selection of the underlying probabilistic model based on observed speech data, where the model was described by regression or linear prediction (LP) coefficients. These coefficients characterize the optimal linear predictor of a data sample given a finite collection of earlier values. The coefficients, combined with voicing and pitch information, were communicated to a receiver to permit local synthesis of an approximation to the original speech. The ML approach specifically included solving the LP(m) problem, and the authors observed that this corresponded in the spectral domain to minimizing what Itakura and Saito called the "spectral matching measure," a spectral distance or distortion measure which

now bears their name [70, 72, 52]. Figure 9.1 from this seminal paper depicts the LP parameters being extracted using the autocorrelation method and transmitted to a decoder with voicing information. The decoder synthesized the reproduction speech from locally generated noise or a pulse train driving an autoregressive filter. The work had actually been presented a year earlier as a confidential internal NTT report. The ideas began to spread with the later publication in 1968–9 of two papers by Itakura and Saito: [70, 71], the first paper written in English. Itakura received his PhD in 1972 with his thesis titled *Speech Analysis and Synthesis based on a Statistical Method.*

The original work did not receive the attention it deserved in the U.S.A., partially because it was written in Japanese. It was read and appreciated, however, by Hisashi Wakita of SCRL in Santa Barbara. SCRL was a small speech research institute founded in 1966 by Gordon E. Peterson, who had spent his career at Bell Laboratories and the University of Michigan working on acoustic analysis and speech. He had moved to Santa Barbara because of failing health and founded SCRL in a beautiful location on a hilltop overlooking the ocean, city, and university. On his death in 1967 the leadership of SCRL fell to his former student, June Shoup.

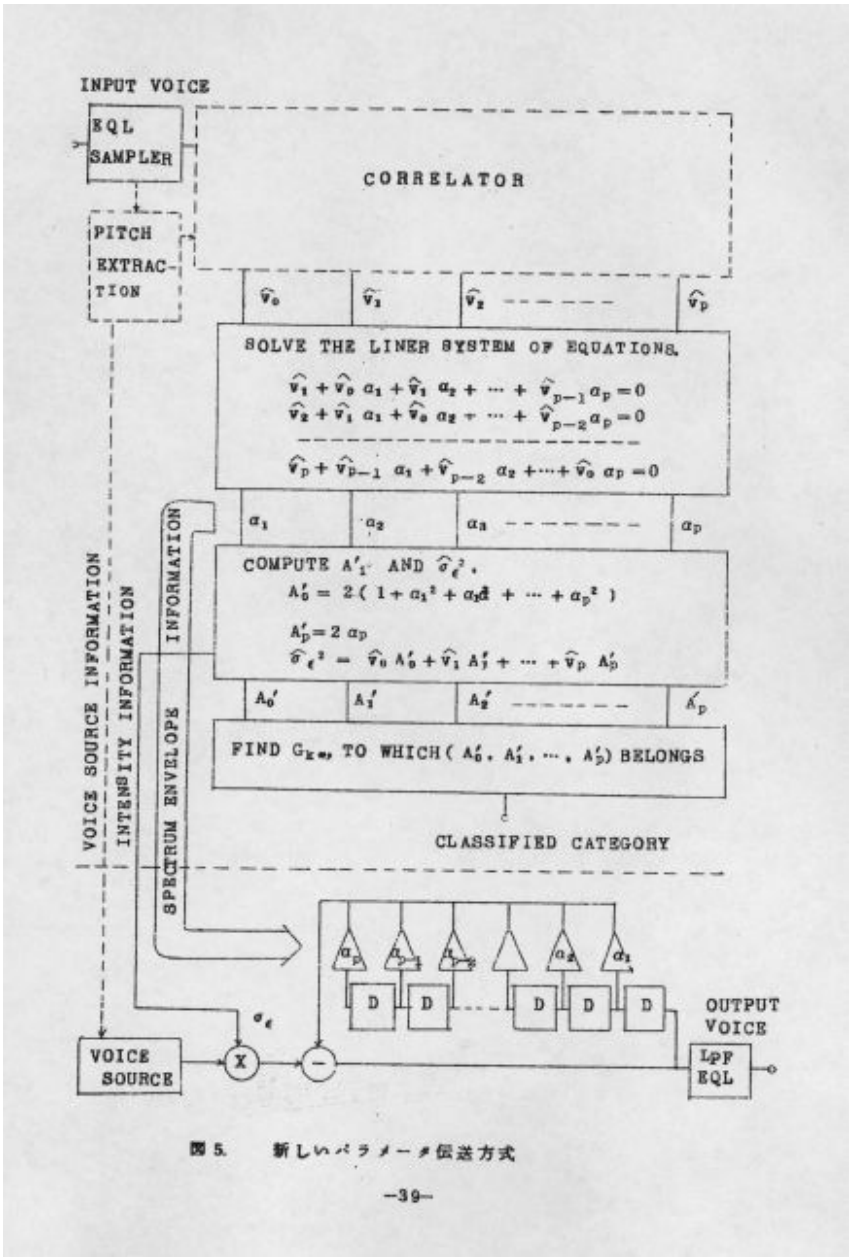


Fig. 9.1 The ML Speech Coder, from [115]

10

1967: Maximum Entropy and APC

Ed Jaynes and John Burg

John Parker Burg had been aware of geophysics and seismology since the time when, as a boy, he had followed his father around looking for oil by analyzing reflected underground signals from explosions. From his early days at Texas Instruments to his Master's degree from MIT he had been interested in deconvolution, unwrapping the effects of filters on signals in order to construct models from data. He had long championed the idea of multichannel Wiener filters and had become convinced of the importance of focusing on prediction error filters rather than the prediction filters alone, both to construct stable feedback models (autoregressive models) for seismic signals and to extend autocorrelations known for only a few lags to all possible lags. His intuition in the mid 1960s suggested several attributes he thought important for a spectral estimation or autocorrelation exten-

sion algorithm:

(1) The focus should be on the prediction error filter. If one could find a prediction error filter that whitened the signal, then the reciprocal magnitude error squared of the filter Fourier transform must be a good approximation of the original power spectral density one is trying to estimate. In other words, knowing the prediction error filter is equivalent to knowing the power spectral density of the signal.
(2) One should not assume a priori a model such as an autoregressive model; instead one should establish a variational principal for selecting a model that is consistent with the data. In his application that meant finding a model that agreed with measured sample autocorrelations up to a finite lag, say m. In other words, Burg's approach was based on correlation matching and his philosophy was to use the fewest possible assumptions on the underlying signal.
(3) The variational principal should choose the most random model possible that satisfies the constraints. Early in Burg's development his idea had been to assume an mth order Markov model if the constraints were on the first m autocorrelation values. Later he based his derivation on the Shannon differential entropy rate based on the interpretation that larger Shannon entropy corresponds to greater randomness. He formulated the problem as that of finding the process with the maximum differential entropy rate given m correlation constraints; and this problem was solved by the mth order autoregressive Gaussian random process solving the LP(m) equations for the given correlations. The principal of maximum entropy had been proposed and popularized earlier by Jaynes [76], but Burg extended the idea to differential entropy rate and continuous random processes.

Burg presented his maximum entropy approach [16] at the October 1967 meeting of the Society of Exploration Geophysists, where he won the award for best presentation of the conference. A noted information theorist, David Sakrison of the University of California at Berkeley

complained about the lack of theoretical support for the maximum entropy approach, claiming that it was essentially ad hoc, but Burg largely won over the audience with his arguments cited above. The algorithm performed significantly better than earlier algorithms for the same application. As a result, it quickly caught on.

Burg's derivation used variational methods that have remained the most popular approach to maximum entropy problems. The signals of choice were geophysical time series and not speech, but the resulting algorithm was essentially that of the Itakura and Saito autocorrelation method. The theoretical connections ran deeper than realized at the time. With the benefit of hindsight, the minimum Itakura-Saito distance can be viewed as a minimum discrimination information in the Kullback sense [80]; that is, the Itakura-Saito distortion is the relative entropy rate between two Gaussian processes, which in turn is the minimum possible relative entropy rate between any process with a given set of correlation values and a Gaussian AR(m) model [55]. Thus picking the best model in an Itakura-Saito distance manner is equivalent to finding the best model for the given correlation values in a Kullback minimum discrimination information sense.

It merits noting that Burg cared only about finding the prediction error filter and not about reconstructing the prediction error itself or creating an approximation to the original signal. Unlike the speech community, he did not require pitch detection or estimation.

Bishnu S. Atal joined Bell Telephone Laboratories in 1961 following a three-year term as a lecturer in acoustics at the Indian Institute of Science in Bangalore. His work focused on acoustics and speech. He credits [7] his initial interest in predictive coding to his 1965 discovery in an information theory seminar of an early paper by Peter Elias on predictive coding [41]. At the time he was a PhD student at the Brooklyn Polytechnic Institute. Elias' paper argued that using prediction to remove the redundancy in a random process retained the essential information while the resulting prediction error process could be more efficiently used for coding. The idea easily applied to coding discrete processes in a lossless fashion, since then knowledge of the residual errors permitted perfect reconstruction of the original source. For continuous or analog processes, however, the idea was more complex as

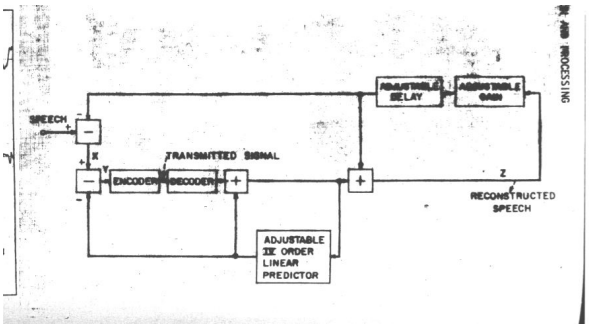
coding the residuals effectively does not immediately imply that the reconstructed process have the same fidelity unless the receiver has the same residuals and knows the predictor perfectly. When the predictor changes with time, feedback can cause problems with the reconstruction. Earlier in 1952 Cutler [39] introduced the idea for quantization with a fixed prediction, differential pulse coded modulation (DPCM) and its generalization, predictive quantization. In this case the predictor was known to the receiver and did not need to be transmitted. If the predictor adapts, however, then either the description of the predictor needs to be communicated or it needs to be embedded in the bits describing the error signal.

Working with Manfred R. Shroeder at Bell, Atal had the idea of using locally generated LP coefficients to predict the speech based on sample autocorrelation estimates using the covariance method. A predictor was designed based on the actual speech samples, and then applied to the reproduced speech samples to produce a residual process — the error between the speech sample and its prediction based on past reconstructed (coded) speech samples. The residual was then quantized and transmitted. At the decoder the digitized residual was added to the prediction of the corresponding sample based on past reconstructed samples, effectively passing the coded residual through the LP prediction error filter. The system was viewed as an adaptive predictive coding (APC) technique, but in the light of Figure 8.1 the system can also be viewed as a form of residual-excited linear predictive technique in that an approximation of the true residual is used to drive the inverse prediction error filter $1/A$ instead of using a locally generated white noise source.

APC was a waveform coder and, unlike the Itakura and Saito and Burg approaches, did not perform any explicit modeling. The only explicit quantization or coding was performed on the residual and not on the LP parameters. Intuitively it is reasonable to expect that doing a good job of quantizing the residual should yield a good overall signal quality when the residual drives the reconstruction filter. However, there is no guarantee that the best low rate match of residuals will produce the best overall signal match from the original speech to the final reproduction, the quantization is effectively being done open-loop.

To form a complete speech coding scheme the LP parameters also had to be digitized and transmitted as they changed, unlike the case in a fixed predictive quantization scheme. The APC work was reported in November [3] at the 1967 Speech Conference held in Cambridge, Massachusetts, and elaborated the following year [4, 5]. It was the first application of linear predictive methods to speech coding to appear in the English literature, but it did not consider an LP-based vocoder (later named LPC) as introduced earlier by Itakura and Saito. The work used the covariance method and a fully developed version was published in 1970 in the legendary research journal of Bell Labs, the *Bell Systems Technical Journal* [8].

In Atal's audience in Cambridge 1967 was digital signal processing pioneer Charles M. Rader of MIT Lincoln Laboratory in Lexington, Massachusetts. He was so excited about the ideas presented that he wrote a Lab Technical memorandum [121] describing the basic ideas, which was accompanied with hand drawn figures and the introductory remarks



APC of Atal and Schroeder

> *The paper was brief and difficult to understand, and the purpose of this memorandum is to explain its contents, based upon my efforts to conjure meaning out of the paper.*

The distribution list of the memorandum included Joseph Tierney, who would play a prominent role in the packet speech story as the primary Lincoln Laboratory participant. Also included were several prominent names in the signal processing community — Ben Gold, Al Oppenheim, and Tom Stockham — and the current (March 2010) leader of the Lincoln speech group, Cliff Weinstein. Although the paper stirred up

quite a bit of discussion, it had a limited impact on work at Lincoln and MIT until 1973. There had been significant channel vocoder and secure speech work at Lincoln Lab, primarily targeted at the then current modem speed of 2400 bps, and linear prediction was applied to reduce bit rate in a homomorphic speech coder by Weinstein and Oppenheim in 1971 [131], but intense activity on linear predictive coding would come later.

APC had the key component of finding the linear prediction coefficients, but it used the coefficients to adapt a predictive quantizer and not to produce an explicit speech model. In this original form, the bit rate had to be at least as large as the sampling rate and the effects of quantization of the LP parameters on the system were not considered. Implicit in the system, however, was the idea of a *residual-excited LP (RELP)* code because the decoder used a discrete approximation of the residuals to drive the reconstruction filter instead of a periodic pulse train or white noise sequence as in pure LPC. The name RELP (and several related names) did not appear until years later because the APC system was not viewed in this way. The bit rate was relatively high in comparison to LPC since at least one bit per sample was required for the residual (unless one downsampled) and an unspecified amount for the LP parameters. The promise was that the added bits could produce a higher quality reproduction by more faithfully producing the residuals and avoiding the tricky problems of pitch detection and estimation required by LPC. Later in 1975 several researchers returned to the idea of coding residuals to produce medium bit rate systems with better quality than the low rate LPC systems.

11

1968: SCRL, the Burg Algorithm, IMPs, and CHI

In 1968 John Markel was a graduate student at Arizona State, working on digital filter design on the side at Motorola. He also wrote tutorial articles on z-transforms in trade magazines, earning $65 a page. He would later note that as a PhD writing research articles for IEEE publication the tables would be turned — he would be paying the publisher about the same amount per page through page charges.

Markel dropped a required language course in French for the PhD program and moved to UCSB, where the Electrical Engineering Department agreed that the language requirement could be satisfied by his skill in Fortran. Late in the year he began work at SCRL to support himself and began his search for a PhD dissertation research topic. His skills in digital filter design and z-transforms proved useful in the speech processing work.

In his initial search for dissertation material, Markel was attracted to the work of Glen Culler, who had developed an approach to speech coding called the "Gaussian wave function." In Culler's view, the wave function, an ancestor of modern wavelets in general and the Gabor wavelet in particular, formed a natural basis for speech processing. With an elegant theory and the computational power of his on-line signal processing system [37], Culler was an attractive possibility to an ambitious PhD student interested in digital signal processing and computation. Markel approached Culler with the idea of doing dissertation research on wave functions, but he was summarily rejected.

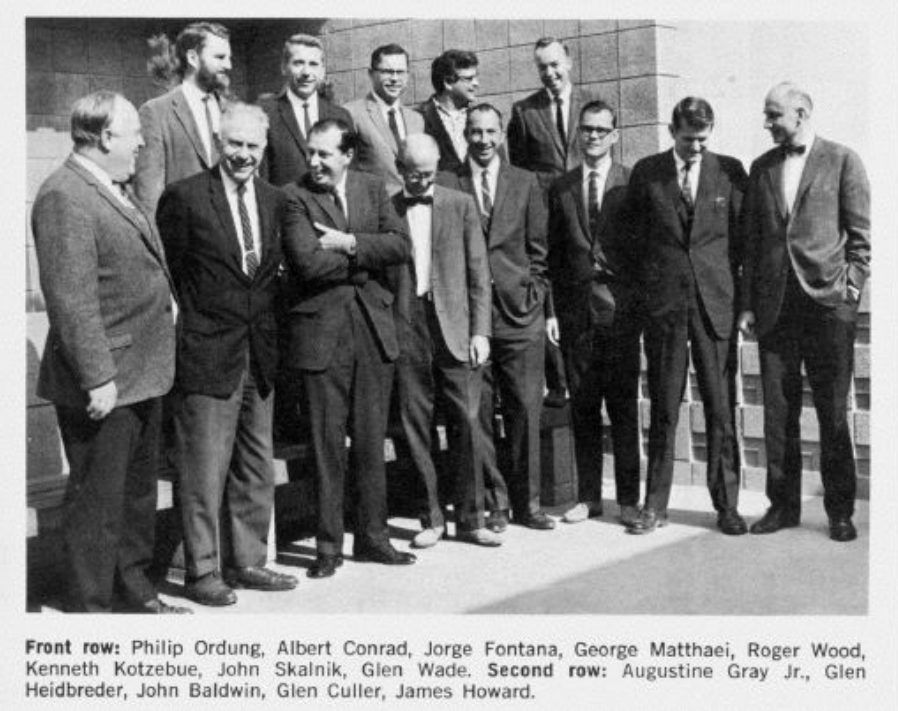
Markel persisted on his own to look into wave function methods,

and sought help on the theoretical aspects from UCSB Professor Augustine (Steen) H. Gray, who had been recommended as a genius at finding closed form solutions to difficult mathematical formulas. Markel put together a dissertation committee including Gray and Roger Wood and began work on implementing a speech coding system that analyzed speech to extract the parameters required to describe the wave functions, and then synthesized the speech from digitized versions of the parameters. The idea was a variation on the classic approach to the operation of a vocoder — a speech waveform is observed and an algorithm applied to construct a model of the speech. The parameters of the model are then digitized and transmitted to a receiver where the model is synthesized to produce a reproduction of the speech. The key difference with the wave function approach was the fundamental signals used and the resulting algorithms required to estimate the underlying parameters. The work progressed rapidly; Markel applied his extensive Fortran and z-transform experience to the development of algorithms for the necessary parameter estimation and synthesis. His PhD project had little overlap, however, with his SCRL work other than reinforcing his growing knowledge of and experience with speech.

While getting up to speed on speech, Markel read James Flanagan's classic *Speech Analysis, Synthesis, and Perception* [42] and set a goal to someday write and publish a book in the same series with the same publisher. As will be seen, he did just that.

In August 1968 John Burg presented "A new analysis technique for time series data" at a NATO Advanced Study Institute [17] , a technique now famous as the Burg algorithm. The method computed reflection coefficients from original data using a forward-backward algorithm. The method was later dubbed by Makhoul [92, 54] the "covariance lattice" approach in speech.

During 1968 Glen Culler, along with such ARPAnet pioneers as Elmer Shapiro, Leonard (Len) Kleinrock, and Lawrence (Larry) Roberts, contributed to the description of the ARPAnet and its Interface Message Processor (IMP) developed by Roberts' Network Working Group of ARPA's Information Processing Techniques Office (IPTO). This description formed the basis for ARPA's *Request for Quotation (RFQ)* to fund a contractor to design and develop the IMP. Culler was



Front row: Philip Ordung, Albert Conrad, Jorge Fontana, George Matthaei, Roger Wood, Kenneth Kotzebue, John Skalnik, Glen Wade. **Second row:** Augustine Gray Jr., Glen Heidbreder, John Baldwin, Glen Culler, James Howard.

Fig. 11.1 UCSB Faculty in 1968, including Steen Gray and Glen Culler

a member of the group and is said to have written one of the early drafts of the specification. The IMP was to serve as the basic "node" of the ARPAnet and was the forerunner of the Internet router. The key idea had been suggested by Wesley Clark following a meeting discussing possible means of network architecture. His idea was to have no common computer hardware requirements for membership in the network, but instead to concentrate on a common interface, the IMP, which could connect to any machine using suitable software for communicating between the local computers and the interface to the network. This freed the network designers from worrying about the many different computers on the network [112]. It was decided, based on an estimated budget, to build 19 IMPS, but financial realities scaled the initial order back to 4. BBN received a contract from ARPA to build and deploy 4 IMPs in January 1969 [116]. The first protocol used on

the fledgling ARPAnet was the Network Control Program (NCP), also called the Network Control Protocol.

Culler cofounded Culler-Harrison Inc (which later became CHI Systems, it will be referred to here as CHI) in Goleta, California next door to the University of California, Santa Barbara, to build some of the earliest array processors. The CHI designs were later acquired, improved, developed, and commercialized by Floating Point Systems (FPS). The FPS array processors eventually replaced the Signal Processing Systems SPS-41 array processors purchased for the ARPA packet speech project. CHI was the first of many organizations that trace their origins back to Culler's Computer Research Lab at UCSB. Many of these are depicted in the "family tree" drawn later in honor of Culler. The

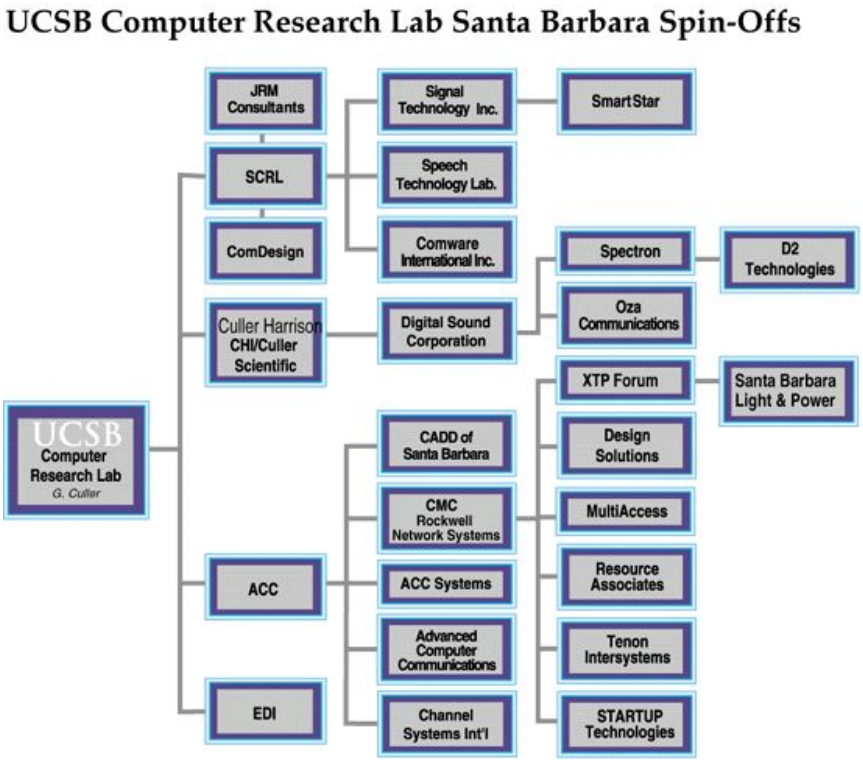


Fig. 11.2 Startups descended from Culler's Lab

tree shows the branches of both speech processing and computer net-

working efforts. The reader should be warned that not all of the leaves acknowledge this geneology.

12

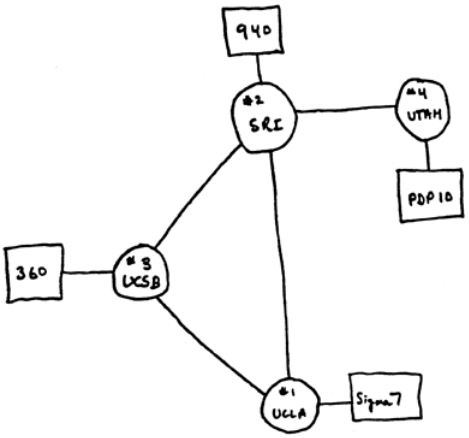
1969: SCRL, PARCOR, LPC, and ARPAnet

Early in 1969 Markel learned of Itakura and Saito's work from Hisashi Wakita, who had worked with Saito. He was highly impressed by the work and realized that his experience implementing the analysis, parameter extraction, digitization, and synthesis using the wave functions could be useful in developing practicable implementations for the Itakura and Saito autocorrelation method. He suggested to Wakita the possibility of seeking funding to support such work and how timely it would be since few in the US were yet aware of Itakura's work. He invited Steen Gray to join him as a consultant, and Gray joined him out of interest in the project. (SCRL did not pay consulting rates, only ordinary salaries with no benefits.) Gray struggled with the Japanese version of the early Itakura and Saito work, but as seen in Figure 9.1, the figures (and equations) were in English.

Itakura and Saito [71] introduced the partial correlation (PARCOR) variation on the autocorrelation method [71], which finds the partial correlation [59] coeffients. The algorithm is similar to the Burg algorithm, but it is based on classic statistical ideas and has lower complexity. Itakura and Saito's PARCOR paper [71] laid the groundwork for several papers during the next few years refining the approach and

considering related issues of digital filtering and quantization [73, 74].

In November B.S. Atal of Bell Labs presented at the Annual Meeting of the Acoustical Society of America [6] a vocoder based on solving the LP(m) problem and sending a digitized version of the system parameters (along with the pitch information) to a decoder. He named the system a linear predictive coder (LPC), and the name stuck. The key difference from the Itakura and Saito system was the use of the covariance method to solve the LP problem instead of the autocorrelation method. The abstract was published in 1970, and the full paper with Hanauer in 1971 [9].



ARPAnet in 1969

Thanks to to the efforts and influence of Culler, UCSB became the third node (IMP) on the ARPAnet (joining #1 UCLA, #2 SRI, and #4 University of Utah). SCRL and CHI were later connected to the UCSB IMP by dedicated phone lines. The artist of the map and other hand drawings depicting the early growth of the network is a matter of some mystery. Some have attributed it to ARPAnet legend (and then UCLA student) Jon Postel, but there are other candidates and the issue is not settled. No two computers on the net were the same; they included a Sigma-7, an SDS-940, an IBM-360, and a DEC-PDP-10. This was a key goal — to be able to communicate effectively among disparate computers. This was in stark contrast to later efforts in some commercial computer networks to use only proprietary protocols and to prevent, rather than enhance, compatibility. With the ARPAnet the goal was to make access as easy and standard as possible, regardless of the specific equipment choices of the users. It was this development

of compatible equipment-independent standards that would eventually lead to the Internet. Grad student Dave Retz again assisted Culler, this time with the necessary software development for connecting the IMP to the UCSB IBM-360. Retz then temporarily changed gears and started working on packet speech, spending time at SCRL.

13

1970–1971: Early LPC Hardware and SUR

John Makhoul in 1974

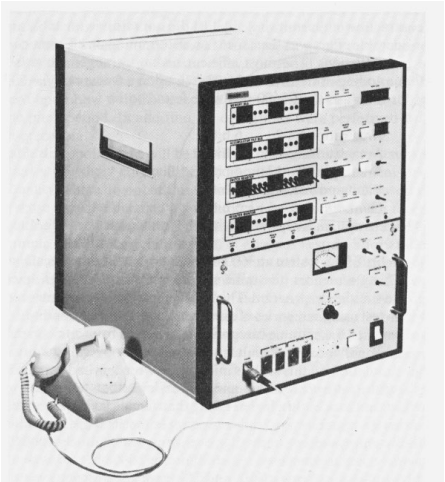
In 1970 John Makhoul joined BBN after receiving his PhD at nearby MIT working on speech recognition with Ken Stevens. He was hired in anticipation of an imminent ARPA program on Speech Understanding Research (SUR), but the program had not yet been announced. So as his initial assignment he was in his own words

> *given the job of rewriting an accounting program for the time sharing system on the PDP-10 in Fortran. The existing program had been written by Danny Bobrow in LISP and took two hours*

> *to run for each weekly accounting run. So, I rewrote the program in Fortran and embellished it; the result was a program that ran in one minute! Then, in June 1971, Bill Woods and I wrote the proposal to ARPA to fund work under the SUR program. The program lasted five years.*

In 1970 CHI developed the MP32A Signal Processor for sonar signal processing.

Realtime 16 bit LPC using the Cholesky/covariance method was first accomplished in hardware by Philco-Ford in Pennsylvania in 1971. Four were sold (to the U.S. Navy and the NSA), they weighed 250 lbs each and used a PFSP signal processing computer [104]. The LONGBRAKE II Final Report was published in 1974 [134]. Little trace remains of the system, but contemporaries described the group as not having their own speech gurus, choosing instead to implement available algorithms developed at other facilities.



LONGBREAK II

In 1971 ARPA initiated the Speech Understanding Research Program (SUR) with a grand goal of developing algorithms for understanding continuous speech. This was a separate project with its own history, but it funded speech work at BBN and SRI among others, and several alumni of the SUR program would later be involved in the packet speech work. At BBN, John Makhoul began thinking about suitable signal processing for speech recognition and reading papers on linear prediction, principally the 1971 paper by Atal and Hanauer [9], a monograph by John Markel [97], and earlier publications from Geophysics by Robinson and colleagues [114]. As a result of his research, Makhoul dis-

covered certain spectral matching properties of linear prediction, which he sketched in a 4-page paper [88] for the 1972 Conference on Speech Communication and Processing, a precursor to the International Conference on Acoustics, Speech, and Signal Processing (ICASSP). When Makhoul handed a copy of the paper to his boss, Danny Bobrow, Bobrow skimmed it and then asked him to write another version of the paper that people like him who were not electrical engineers would be able to understand. Makhoul recalls

> *As I started writing, I realized that there was a lot that I did not understand, so I embarked on an intensive research effort that culminated in the publication in Aug. 1972 of a 237-page BBN report on linear prediction and the spectral analysis of speech [89]. BBN printed 250 copies of the report and distributed it to key researchers in speech processing. When I next saw Atal at a technical meeting, he commented that the report was very "exhaustive," so I added "and exhausting!"Security people at BBN at one point complained to my management that they did not understand why I was at BBN during so many evenings and weekends!*

Tom Stockham, who had moved from MIT to the University of Utah and who was a member of the *IEEE Proceedings* editorial board, so admired the report that he invited Makhoul to write a tutorial review paper on linear prediction. The resulting classic paper [91] was published in the *Proceedings* in April 1975.

14

1972: Early Efforts towards Packet Speech

In 1972, CHI developed the AP120, one of the first array processors that would power the digital signal processing revolution of the decade. The Floating Point Systems FPS AP 120B, perhaps the most famous of the array processors, was a descendent of the CHI AP120.

The USC Information Sciences Institute (USC/ISI) was founded in Marina del Rey, California. The organization would play a key role in this story as well as in the development of the ARPAnet in general.

Robert (Bob) Kahn moved from BBN and leadership in the IMP development to ARPA, where he eventually replaced Larry Roberts as the head of IPTO in the Washington, DC, area. Kahn joined with Jim Forgie of Lincoln Laboratory and Dave Walden of BBN (who had helped develop the IMP) to initiate the first efforts towards demonstrating packet speech on the ARPAnet. They simulated the transmission of 64 Kbps PCM speech packets to understand how one might eventually fit packet speech onto a network with a limited bit rate. This work illustrated that realtime communication of signals as well as other types of information and data were being considered from the earliest days of packet networks, but the initial conclusion was that major changes in packet handling and serious data compression of the speech waveform

were needed. The ARPAnet was never fast enough to communicate the bits required for uncompressed speech in real time, even without the additional problems of delay, processing, errors inherent in networks, and sharing the network with other data. Kahn is generally credited as being both the initiator and the guiding hand of realtime signal transmission on packet networks, in particular the transmission of realtime speech recounted here and of the realtime video to come. Forgie would play a major role in the packet speech work, primarily on the computer side developing protocols for realtime signal transmission within and among networks. He was a 1951 MIT EE graduate who had spent his career at Lincoln Lab, primarily working on computer hardware and operating systems. His computer work had been funded by ARPA, so he was familiar with the agency and vice versa.

Danny Cohen was working at Harvard at this time on realtime visual flight simulation across a network, a problem requiring enormously sophisticated computer processing fast enough to provide almost instant impact. The computer power of MIT was connected to the console at Harvard, so the goal was to do the enormous number crunching in an invisible way across the network connection. Bob Kahn suggested to Danny that similar ideas might work for realtime speech communication over the developing ARPAnet and described his project at USC/ISI in Marina del Rey. Danny was intrigued and, perhaps as a result, moved to USC/ISI the following year. Danny had been familiar with computer networks and graphics, but had not been directly involved with voice or speech processing at the time. Indirectly, however, he had. Earlier in 1967 Danny had worked as a contractor for MIT Lincoln Lab, helping to connect Larry Roberts' DEC-338 in his ARPA/IPTO office (3D-169 in the Pentagon) to the powerful TX2 computer at Lincoln. The connection used point-to-point telephone lines (hence it was not networking). Carma Forgie also programmed for Roberts. The TX2 was a transistor-based computer, one of the most powerful computers available at the time. It had been designed by Wesley Clark and had one of the first graphical user interfaces, designed by Ivan Sutherland [45]. Both Clark and the TX2 reappear later in the story at key junctures. At Lincoln Danny often met Ben Gold — along with Charles Rader one of the founders of the field of digital

Carma Forgie at the TX2 at the MIT Lincoln Lab TX2 console (Reprinted with permission of MIT Lincoln Laboratory, Lexington, Massachusetts)

signal processing. As a contractor, Danny had no badge, and Ben often escorted him to the Lincoln exit. The pair were familiar enough to the guard that Danny found he could escort himself, saying goodbye to a virtual Ben in the stairway as he entered the corridor alone to arrive at the security station.

Markel, Gray, and Wakita began to publish a series of papers, which continued to appear over the next few years, on the implementation of the Itakura-Saito autocorrelation method along with contributions to the underlying theory [98, 130, 100, 99, 101, 102, 103]. Early versions of the papers were published as SCRL reports and presented at IEEE International Conferences on Acoustics, Speech, and Signal Processing (ICASSPs) and meetings of the Acoustical Society of America. In a foreshadowing of the modern open source movement, they provided Fortran code for LPC and the associated signal processing algorithms.

At the University of Utah, Tom Stockham's PhD student Steven Boll demonstrated the advantages of pitch-synchronous LPC over time-synchronous or frame-synchronous LPC and presented his work at a meeting at NSA. The presentation indicated early activity on LPC at the University of Utah. The technique itself eventually spread to many LPC systems, allegedly including the later NSA speech coding box, STU II, but the preference for pitch-synchronous processing was not universally held. John Makhoul writes

> *In the early 1970s, it was a given part of the speech folklore that one must use both pitch-synchronous analysis and pitch-synchronous synthesis. The fear was that, if the parameters of the synthesizer were updated time-synchronously (say every 10 ms), then a buzz at 100Hz would be heard. What we showed at BBN was that one can perform both time-synchronous analysis AND time-synchronous synthesis without any deleterious effect on the synthesized speech. We demonstrated this fact for the first time at one of the NSC meetings. (I will mention here that one of the prominent members of the Committee thought that he heard a buzz in the synthesized speech, but no one else did and, in fact, there was no buzz upon closer listening. It is amazing what one can make himself hear if he wants to.) Since then, time-synchronous analysis and synthesis have been the common thing to do. This simplified greatly both analysis and synthesis since pitch-synchronous analysis and synthesis are so cumbersome and error-prone. This item was included in BBN Report 2976 in 1974, but we did not make a big deal out of it in the open literature.*

At SRI, Earl Craighill, Dave Ellis, and colleagues developed a powerful time series capture, processing, and display program called TIMSER that ran on the PDP-10 computer. With funding from the Defense Communications Engineering Center (DCEC) in Reston, Virginia, and ARPA, Craighill and Ellis, along with Tom Magill, worked on speech vocoders and waveform coders based on Markel's LPC anal-

ysis/synthesis code, and they investigated excititation issues for LPC vocoders. In particular, they developed the theory and implementations of a variation on LPC called DELCO. The method [86, 87] extended a traditional data compression method [40] of reducing the data rate by not sending data when new parameters differed from previous parameters (or silence) by an amount less than some threshold. The technique was applied to LP parameters using spectral distance measures and to packet systems with packet header overhead and separate packet transmission. Unfortunately, the increased packet overhead swamped the additional compression gained by the technique, so the technique did not become part of the mainstream NSC work. The method did result, however, in an active exchange among NSC participants Craighill, Makhoul, Boll, and Markel. Some of the ideas that cropped up in the study would later lead to useful improvements in the adopted LPC algorithms. Another outcome of this work was a mixture algorithm for pulse/white noise excitation of the LPC synthesis filter.

Craighill's interest in speech processing dated from his PhD dissertation at Michigan State University with Bill Kilmer. Seeking a topic that combined his computer skills and implementation interests with his affinity for digital signal processing and linear filtering — especially Weiner-Hopf filtering — and with his preference for work not requiring a security clearance, Earl converged on the topic of speech recognition. He completed his dissertation, including consideration of speech signal representations, at SRI, which he joined in June 1968. Given his work in speech recognition, he naturally drifted towards the SRI portion of SUR. However the emphasis at SRI was on artificial intelligence and not on the signal representations and signal processing that interested him. Partially as a result, he moved from speech recognition into speech compression. Another reason for his move was his understanding of the limitations of the all-pole LPC methods for modeling certain types of sounds — such as nasals — which were better modeled using filters with zeroes. One way to handle such models was with more sophisticated excitation signals to drive the autoregressive filters. Craighill recalls:

> *When we started the excitation project, LPC analysis*

> *and the inverse filter concepts were firmly established. Our goal was to represent and code the error signal. One only needs to listen to an error signal out of the inverse LPC filter to realize that it's not at all white, rather very intelligible (sounds a little like pre-emphasized speech, the inverse filter is primarily a differencing calculation). I attributed this high degree of intelligibility to the fact that the LPC inverse filtering based on stationary signal and Gaussian processes (basically that second order statistics adequately describe the time series) assumptions didn't represent the transients which must have a considerable amount of speech information (intelligibility). The LPC synthesized speech is remarkably quite intelligible because of its good match to spectral peaks ...*

Dave Retz completed his UCSB PhD thesis on packet speech and changed projects to work on software to handle realtime signal acquisition, including speech and other signals.

In the autumn of 1972, Steve Crocker, the Program Manager of the ARPA SUR program, asked John Makhoul of BBN to work on speech compression and Vishu Viswanathan, a recent Yale PhD, was hired as the first full-time person on the effort. John became Vishu's mentor and collaborator in speech signal processing in general and speech coding in particular. Vishu's initial task at BBN was to develop and implement a complete LPC speech coder, including a reliable pitch extractor and efficient quantization of all LPC coder parameters. During the period he focused on improving the quality of LPC speech. He recently recalled

> *I still remember the issue of the effect of becoming too familiar with listening to LPC speech. In the beginning of every day, LPC speech sounded quite bad to me, with a lot of problems; as I kept listening to identify specific problems in an effort to fix them, the coded speech did not sound that bad after all! This clearly pointed out the need to get other people, especially those that are not directly involved in speech coder development, involved*

in listening tests, even for informal evaluation.

Even though BBN was involved heavily in ARPA networking activities, including building the ARPAnet and contributing to the design of network protocols, its involvement in the NSC program was only in the speech compression algorithmic part.

15

1973: USC/ISI and NSC

In 1973 Danny moved to USC/ISI and began work with Stephen (Steve) Casner, Randy Cole, and others and with SCRL on realtime operating systems. As Danny became more involved with the speech project, he learned of LPC from his SCRL colleagues as a promising candidate for speech coding.

Bob Kahn realized that the bandwidth available on the ARPAnet, then 50 Kbps, was insufficient for ordinary PCM at 64 Kbps and that serious compression was needed. Adaptive delta modulation (ADM) was considered, but it did not provide sufficient compression to fit into the 50Kbps leased lines on the ARPAnet along with the other data and the necessary overhead. At the time, few thought that packet networks in the foreseeable future would be able to support realtime speech or video. Again looking back, it is difficult to appreciate the then apparent difficulty.

To explore the possibilities of packet speech on the ARPAnet, Kahn formed the Network Secure Communications (NSC) program with Danny as its chair; Kahn became its *éminence grise*. [1]

[1] As several readers have asked me about this term, it merits a note as I can find no good synonym. It means someone with no official position (in this case on the NSC) who exerts

The goal was to pursue methods for digital speech on packet networks that would provide secure communications. The public use of the word "secure" was alleged to have made certain government agencies uneasy, so the acronym was soon redefined as "network speech compression." It was eventually referred to jokingly as the "network skiing club" or "national ski committee" because of a preference for winter meetings at the University of Utah in Salt Lake City, a half hour away from the ski slopes in Alta. As John Makhoul tells the story,

> *One of our NSC February meetings happened at the University of Utah in Salt Lake City, hosted by Tom Stockham and Steve Boll. The meeting was scheduled to end on Friday in order to allow attendees to spend the following weekend skiing in Alta or Snowbird. The attendees enjoyed this so much that we decided to hold the winter NSC meeting every year in Salt Lake City. In time, we started referring to NSC jokingly as the National Ski Committee.*
>
> *I recall after one of these meetings that I went skiing for the first time in my life in Alta. After taking a one hour ski lesson, Danny Cohen and Randy Cole accompanied me on the chair lift to the top of one of the slopes. They waited for me for a few minutes to make sure I was OK and then I suggested that they go off and leave me to negotiate the slope on my own, which they did. I will never forget the difficulty of that first run, but it got easier after that.*

As every ARPAnet node had different equipment and software, the early focus was on the interface and compatible standards.

The original members of the NSC were ARPA, USC/ISI, the University of Utah, BBN, MIT-Lincoln Laboratory, and SRI. They were soon

---

a powerful influence. Its origins lie with François Leclerc du Tremblay, who wore gray and had a powerful influence on the government of France through his service to and influence on Cardinal Richelieu, the official leader and *éminence rouge*. Bob Kahn participated in the NSC; he was not its official leader and was not involved in the speech compression details, but his guiding influence was crucial.

joined by SCRL and CHI . (The membership in November is shown in Figure 15.1.) While the list shows the initial primary representatives to the NSC from their institutions, the responsibilities and participation were often shared with and sometimes taken over by others. The following is a list of active participants other than ARPA (with apologies to any I have left out)

USC/ISI Danny Cohen, Steve Casner, Randy Cole, Paul Raveling
SCRL John Markel, Steen Gray, Hisashi Wakita
CHI Glen Culler, Mike McCammon, Dale Taylor, Gary Ball and a teen-aged programmer David Culler, Glen's son.
SRI Tom Magill, Earl Craighill, Dave Ellis, Don Alves, Don Cone, Andy Poggio, Darryl Rubin, Jim Mathis.
MIT Lincoln Lab Joe Tierney, Jim Forgie, Pete Blankenship, Doug Paul, Cliff Weinstein, Ted Bially
BBN John Makhoul, Vishu Viswanathan, Dave Walden
University of Utah Steven Boll, Tom Stockham

The presence of USC/ISI, SCRL, CHI, SRI, BBN, and Lincoln Lab was all natural given their activity and participation in the signal processing and networking aspects already described and their funding from ARPA and other government agencies. The presence of the University of Utah was a different story. Tom Stockham, one of the pioneers of digital audio and early digital image processing, had an existing ARPA contract primarily focused on digital audio and image processing. He also had a PhD student, Steven Boll, who had worked on LPC. So Stockham offered the Utah facilities to Bob Kahn for NSC and appointed Boll as representative. That left Tom more time for his focus on digital audio, including his famous work restoring the Caruso recordings, and laying the ground work for his new company, Soundstream. He also was diverted with his assignment to analyze the 18 1/2 minute gap in the Nixon tapes in early 1974. Boll recalls

> *Although Utah's contributions to the NSC efforts were minimal, these were exciting times for Utah. The big players (other than Tom) were in graphics. Dave Evans and Ivan Sutherland's influence nurtured many eventual*

ARPA Network Information Center
Stanford Research Institute
Menlo Park, California 95025

Marcia Keeney
SRI-ARC
November 14, 1973

Network Speech Compression Note #3
NIC 19946

NETWORK SPEECH COMPRESSION GROUP MEMBERSHIP LIST

Steve F. Boll
University of Utah
Computer Science Dept.
3160 Merrill Engineering Building
Salt Lake City, Utah 84112

SFB
(801) 581-8576
UTAH-10

Dan Cohen
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, California 90291

DC
(213) 822-1511
USC-ISI

Glenn J. Culler
Culler-Harrison, Inc.
150-A Aero Camino
Goleta, California 93017

GJC
(805) 968-1813
CHI2

Robert E. Kahn
Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

REK2
(202) 694-5921 or 694-5922
ARPA-TIP

D. T. (Tom) Magill
Stanford Research Institute
333 Ravenswood Avenue
Menlo Park, California 94025

DTM
(415) 326-6200 ext 2664
SRI

John Makhoul
Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Mass. 02138

JM
(617) 491-1850 ext 234
BBN-TENEX

John D. Markel
Speech Communications Research Lab, Inc.
800 Miramonte Drive
Santa Barbara, California 93109

JDM
(805) 965-3011
SCRL

Joseph Tierney
MIT Lincoln Laboratory
Lexington, Mass. 02173

JT3
(617) 862-5500 ext 277
IND

Fig. 15.1 NSC

> *stars including Warnock for Adobe, Catmull for Pixar, Jim Clark, Silicon Graphics, and Netscape, and Nolan Bushnell, Atari.*

and because the winter NSC meetings were usually held in Utah,

> *. . . in early and mid 1970's you could walk down the halls of the Merrill Engineering building and run into the Advisory Panel on White House Tapes: Bolt, Cooper, Flanagan, McKnight, and Stockham, sit down to a planning meeting with the pioneers of the Internet and LPC, or have a beer with future graphics multi-millionaire.*

At USC/ISI, Steve Casner and Randy Cole were the primary developers for NSC, with Randy doing the signal processing and Steve the PDP-11/45 work. Randy Cole had come to USC/ISI from the University of Utah, where he had been a graduate student of Tom Stockham. At Utah he had watched the installation of an IMP that would connect the University to the ARPAnet.

When the NSC was formed, SCRL was already actively involved in a variety of efforts in speech processing, including well-funded work on LPC speech coding for the military and NSA. For a small company they had remarkable computing resources, including two PDP 11/45s. At an SCRL reunion in 1999, John Markel recalled the arrival of an immense memory of 10 Megabyes in a box several feet square.

SRI would be active in the software development for the ARPAnet packet speech project, especially in the development of the LPC coder used in the 1975 and 1976 demonstrations to be described and was the primary developer of the SPS-41 synthesizer (including coefficient interpolation) used by both SRI and USC/ISI. Much of the SRI activity however was devoted to a second packet network, a wireless network based on packet radio — PRnet — which was initiated in 1973 by ARPA and involved SRI, BBN, Collins Radio, Network Analysis Corporation, and UCLA. Bob Kahn was the program manager of the PRnet program and also its chief architect. While the ARPAnet was based on landline point-to-point connections, PRnet was envisioned as

a mobile wireless extension to theARPAnet, an idea of particular interest to the U.S. military [109]. PRnet was intended to be a separate, adjunct network, but it was Kahn's intention that eventual network protocols should permit the smooth flow of information between the diverse networks as well as among those networks — what would later be called an *internet* of distinct packet networks, but at the time was called a *catenet* for "concatenated network."

The different networks had very different characteristics including bandwidth and bit rate, delay, throughput, cost, and packet loss rate, and hence each had a distinct collection of specific hardware and software issues which had to be resolved in a way that would permit interoperability. PRnet was a much smaller effort and remained experimental, while the ARPAnet served a growing community of researchers and users. The early emphasis in PRnet was on the reliable transmission of data from mobile terminals, but as with the ARPAnet, realtime signal transmission was also being considered from early on by Bob Kahn [79]. The PRnet was arguably the first distributed system deployed with embedded microprocessors and, in addition, also the first that employed direct sequence spread spectrum. It was surely the first to do both at the same time.

As with SCRL, the joint work of the NSC called for visits from SRI to USC/ISI. Earl Craighill recalls his travels:

> *I would arrive around 8 to 9 and begin working with the group programming the SPS-41 (Randy Cole was the main guy on this). We'd continue till 5 o'clock when that crew went home. The protocol programmers (Steve Casner was the lead) arrived about then and I started working with them. They went till 2 to 3 in the morning. I went to the motel, crashed for a couple of hours, then came back at 9 to do it all over again. After a few days of this, I needed a week to recover (but rarely got it)!*
> *During one of our test sessions between SRI and USC/ISI, I wanted to try the DELCO coding I had done. I sent the files to USC/ISI and we fired up the system. Randy Cole was talking on the other end. He*

> *had developed the capability of talking in perfectly natural sounding phone inverted speech! Instead of "repeat that sentence" he would say "peat rer that ten sen ce". This was done so well, that I was completely fooled. I thought sure that I had screwed up the indexing of the output speech. Further, Randy reported that I sounded fine. So I was doubly perplexed. How could it work in one direction and not the other? The ISI bunch was rolling on the floor since they could hear my frustration clearly even through the LPC vocoded channel.*

Tom Tremain of the NSA was an invited guest and attended NSC meetings regularly. Others also attended as guests on occasion, including Richard Wiggins and Bob Hewes of Texas Instruments (TI), Jack Raymond of Harris Electronics, George Kang and Dave Coulter of NRL, and Tom Quatieri and Al Oppenheim of MIT.

Other than the winter meeting, NSC meetings moved around the country, including Cambridge, Santa Barbara, Washington DC, Salt Lake City, and Dallas. Earl Craighill observed:

> *There were four different types of researchers in the NSC mix (some people were multiple types): signal processing mathematics-oriented, heuristic speech experimenters, realtime signal processing programmers, and network protocol coders. The NSC program started heavily in signal processing theory and speech quality measures (toll grade quality was the goal). As the algorithms firmed up, programming took over the meeting discussions.*

Steven Boll recalled that the meetings were often organized with speech the first day and networking the second. Meetings were often enhanced by social gatherings, including dinners in restaurants and in members' homes. Carma Forgie hosted a large gathering at her house in Arlington, Massachusetts, during an NSC meeting in Cambridge, with a crowd partaking in a buffet dinner while discussing technical and nontechnical subjects. She remembers Danny successfully dismantling an

intricate wooden puzzle, and then failing to put it back together after many efforts. When he departed, Danny thanked her for a fascinating evening. The intricate puzzle had been crafted by a Lincoln Lab engineer and Danny subsequently bought one for himself. Earl Craighill remembered an enjoyable dinner alone with the Forgies at their house during a meeting, as well as a not-so-enjoyable pre-meeting to an NSC meeting at the Forgies:

> *We were going over some details of NVP (maybe even the conferencing version). Darryl Rubin and I were in a rental car and we were to follow Danny from the rental lot. Danny was already a crazy Boston driver and I knew Darryl would have a hard time following him. As we were leaving the rental lot, Darryl noticed the rear view mirror was broken and he wanted to get a different car. When that was done, Danny was nowhere in sight. We tried to follow the directions we had, but Boston had no street signs (at that time, maybe they put some up since). We were hours late and everybody including Bob Kahn was not thrilled with us. Needless to say, we didn't score many technical points that night.*

John Makhoul vividly recalls two NSC meetings.

> *Before one of the NSC meetings, which was scheduled to be in Santa Barbara, I visited ISI. So, Danny Cohen asked me if I wanted to fly with him to Santa Barbara instead of drive. I was rather hesistant, because if Danny piloted an airplane the way he talked, I was in trouble. It was a beautiful, sunny day and I was the only passenger in that twin-engine small plane. What amazed me the most was that, when Danny would talk to ground control, I could not understand a single word (because of his heavy accent and my unfamiliarity with the domain), but I guess ground control must have understood every word, because they never asked him to repeat and we landed safely!*

One of the Utah meetings coincided with the big blizzard of 1978 in the Northeast when all ground traffic was halted and the airport was closed for a full week. So, Vishu and I informed Duane Adams, the ARPA program manager at the time, that we would not be able to attend the NSC meeting. After the meeting, we learned that Cliff Weinstein, who also lived in the Boston area, had actually made it to the meeting. Somehow, he had made his way to South Station in Boston, took the train to Philadelphia, and flew from there to Salt Lake City. So, Duane asked Vishu and me afterwards, if Cliff was able to attend, why couldn't you!

Participants in the program often traveled to the other institutions for joint work. Danny Cohen was ubiquitous and Jim Forgie and Earl Craighill recalled visits to USC/ISI and Jim to SCRL. Perhaps those at SCRL had the ideal situation — at their 1999 reunion several recalled the frequent travel between the hilltop paradise of SCRL in Santa Barbara and the waterfront paradise of USC/ISI in Marina del Rey. Looking back it seems an impossibly excellent working environment, working in wonderful locations with cutting edge technology and amazing toys on the forefront of the tsunami of the coming Internet.

John Markel became Vice President of SCRL.

In July, Ed Hofstetter at Lincoln Lab published a technical note titled "An introduction to the mathematics of linear predictive filtering as applied to speech analysis and synthesis" [63], and this time linear prediction caught on. The technical note cited Wakita, Makhoul, Markel, Gray, Atal, Hanauer, and Itakura along with the classic mathematical literature of Wiener [135] on prediction and Grenander and Szego [60] on prediction and Toeplitz forms. Visits from Lincoln to SCRL took place and Joe Tierney found the Wakita and Markel formulations along with their connections of the speech parameters to acoustic models to be clear and elegant. Tierney was selected by Lincoln Lab to serve as its representative on the NSC, in spite of his aversion to long meetings. Tierney had a strong background in speech coding, but he was primarily a hardware and signal processing expert.

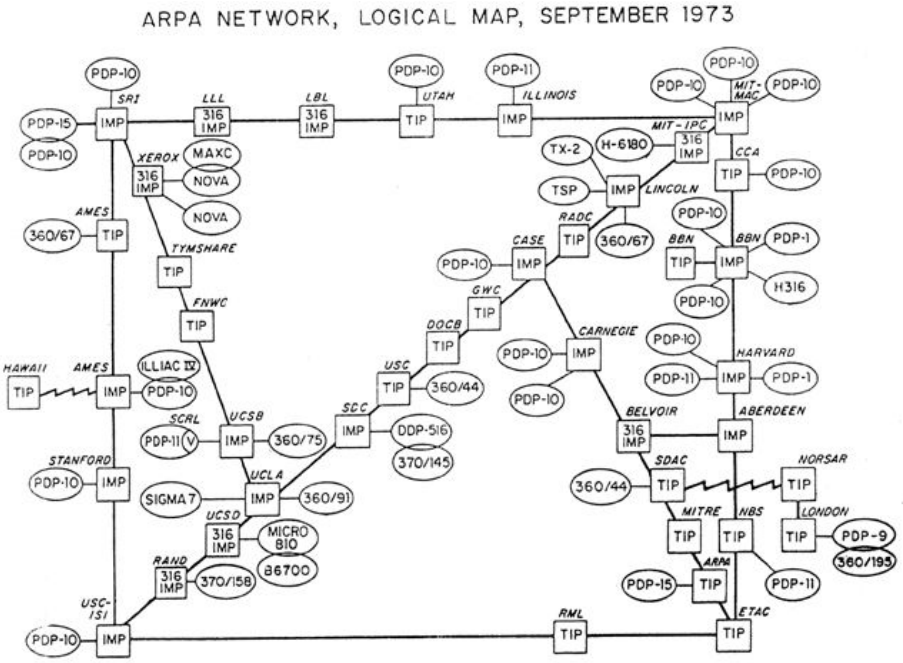
Jim Forgie provided the network knowledge for the Lincoln part of the group.

John Markel and Steen Gray
(Larry Pfeiffer is in the background)

During 1973 The USC/ISI and then the full NSC adopted the basic Markel/Gray software [100] implementation of the Itakura-Saito autocorrelation method as the vocoder technique for the network speech project. Thus LPC10, LPC with 10 coefficients, became the speech coding algorithm of choice. The primary hardware chosen by the group for the speech compression algorithms was a Digital Equipment Corporation (DEC) PDP-11/45 and the SPS-41 array processor for computing the autocorrelations, but both CHI and Lincoln Lab were exceptions and used their own systems: CHI using a CHI 32A with an early array processor (the AP-90 or AP-120), and Lincoln Lab using its TX2 computer with the Lincoln Fast Digital Processor (FDP). Software development was divided among USC/ISI, BBN, Lincoln Laboratory, and SRI. Remember that at this time DSP chips did not yet exist, and the digital signal processing algorithms were implemented in array processors. Lincoln Laboratory tradition holds that it was at about this time that the first realtime 2400 bps LPC on FDP was done by Ed Hofstetter using the Markel/Gray LPC formulation. Results were later published in 1975 and 1976 [64, 66].

At USC/ISI the Network Voice Protocol (NVP) was being developed by Danny Cohen and his colleagues in an attempt to provide a standard for realtime speech communication on packet networks. At roughly the same time the Transmission Control Protocol (TCP) was being developed by Bob Kahn and Vinton (Vint) Cerf to replace the original NCP as the basic network protocol. Cerf had been first a graduate student at UCLA, then an Assistant Professor at Stanford University, and later (in 1976) moved to ARPA. As originally envisioned and first published, the TCP incorporated a variety of network functions, including both the transportation of packets to the correct destination and the ensuring of data reliability and integrity. The NVP, however, used only the the ARPAnet message header of the NCP. Cohen realized that TCP as originally implemented was unsuitable for realtime communication because of packet and reliability constraints. He was likely unaware of the approach that Bob Kahn had originally intended for handling realtime communication with TCP, and later argued for the extraction from TCP of a simpler protocol without the constraints of the original TCP. This proposed secession would eventually result in the separation of the packet delivery portion of the original TCP into IP and the data integrity portion into a new TCP, resulting in the TCP/IP protocol. Kahn, who was aware of the implementation choices that had been made by the developers in the original implementation of TCP, intended that these limitations would be transitory as they were not mandated by the original design. In his view, this issue was best addressed by an implementation change that allowed the destination to control the process by which acknowledgments were sent back to the source. If acknowledgments were returned for realtime communications so as to insure the continuity of communications, rather than the reliable delivery of all packets, the source would keep sending on the assumption that prior packets were successfully delivered for the realtime data. Late arriving packets could also be saved by the destination, if desired, even if not played out.

For those who did not need the reliability of the new TCP, a new protocol, the User Datagram Protocol (UDP), would be developed later for nonsequenced realtime data for use in conjunction with IP.



The ARPAnet in 1973

Fumitada Itakura began a two-year visit with James Flanagan's Acoustic Research Group at Bell Laboratories.

The ARPAnet continued to grow, as shown in the figure.

John Burg left Texas Instruments to form his own company, Time and Space Processing (TSP) to design and produce seismic signal processing equipment. In a conversation with Bishnu Atal, Burg learned of LPC and the similarity in the algorithms he had developed for geophysical signals and Bell's recent work on linear prediction applications to speech coding. As a result, Burg decided to become involved in products for digital speech coding using his algorithms and take advantage of the amenability of digital signals to secure communications.

16

1974: TCP, NVP, and Success

In 1974 two seemingly incompatible major events in the history of the Internet occurred. The first and most famous was the publication of of the paper "A protocol for packet network communication" by Kahn and Cerf which describes what became the TCP/IP protocol [24] . The protocol design had begun the previous year with a grand goal of providing a protocol that would ensure the reliable transmission of packets through a diversity of networks, collectively forming an internet. The less well-known event, but key to this story, was the summer publication in an USC/ISI Technical Report on NVP by Danny Cohen and his colleagues in the NSC [67, 69, 29, 30, 31, 34]. The NVP spelled out the details of how realtime speech could be communicated on the ARPAnet, but it did not use the newly developed internetworking TCP as implemented, rather it used only the basic ARPAnet message headers. The basic goals were spelled out nicely in the preface to the protocol:

> *The major objective of ARPA's Network Secure Communications (NSC) project is to develop and demonstrate the feasibility of secure, high-quality, low-bandwidth, realtime, full-duplex (two-way) digital voice*

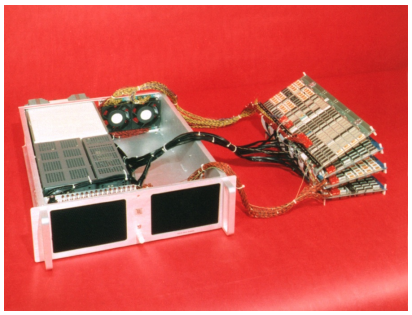
> *communications over packet-switched computer communications networks. This kind of communication is a very high priority military goal for all levels of command and control activities. ARPA's NSC project will supply digitized speech which can be secured by existing encryption devices. The major goal of this research is to demonstrate a digital high-quality, low-bandwidth, secure voice handling capability as part of the general military requirement for worldwide secure voice communication. The development at USC/ISI of the Network Voice Protocol described herein is an important part of the total effort.*

On January 28, 1974, Danny Cohen and Vint Cerf met in Palo Alto and began a long discussion regarding the handling of realtime applications competing with the necessity for transmitting reliable data. Danny described the difference between the two regimes as the dierence between milk and wine —

> *this analogy relates to the choice of what information to keep when a buffer overflows. With milk, you want to discard the oldest and keep the newest. With wine, you want to discard the newest and keep the oldest. Milk corresponds to the realtime packet voice data, because minimizing delay is critical for interactive speech. Wine corresponds to reliable byte-stream data (TCP) because none of the later data can be delivered until all the earlier data is reliably received and delivered.*

Most of the thought devoted to developing standard protocols for future networks had been towards the reliability issue, which typically involved the verification of data to determine when packets were lost or corrupted. However, techniques for reliability, particularly the need for retransmission of lost data, slowed transmission and could make realtime signal communication — such as speech, audio, and video —impractical. The initial protocol proposed was TCP, which was announced in the May 1974 *IEEE Transactions on Communications* [24].

The original TCP protocol as proposed in [24] focused on reliability and, although the paper was silent as to the process by which acknowledgements were to be sent, it was clear how the sender should interpret them. However, as actually implemented by Cerf's group at Stanford and his colleagues at BBN and UCLA, it was not friendly to realtime signals. Cohen realized this fact early and argued for separation of a simpler protocol more amenable to realtime processing from the original TCP, a proposal that would eventually lead to the extraction of IP and UDP from TCP and the formation of the new TCP/IP protocol suite, with TCP taking on a narrower meaning.



LDVT

During the year Lincoln Laboratory developed its Lincoln Digital Voice Terminal (LDVT), a computer dedicated to realtime implementation of speech compression algorithms, including LPC. A description of the system was published in 1975 by Blankenship [14]. CHI developed the AP-90B and the AP-120 array processors. The AP 120 was the ancestor of the subsequent Floating Point Systems FPS AP 120b.

In August 1974 the first successful (in a sense) realtime conversation took place between Randy Cole at USC/ISI and Jim Forgie at Lincoln using continuous variable slope delta modulation (CVSD) at 8 Kbps. Unfortunately the quality was so poor that neither Randy nor Jim could understand a word the other said, but the protocol was validated — and Randy reported he could recognize Jim's voice, verifying speaker identification if not speech communication! CSVD was used because it was relatively easy to implement and because the LPC implementation had not yet been completed on the SPS-41. Danny Cohen remembers the SPS-41 as being extraordinarily difficult to program. Danny quotes John Markel as saying it was impossible, and Randy quotes Danny as saying that programming the SPS-41 was "advanced logic design."

While the speech was incomprehensible, the experiment successfully validated the NVP [105, 106, 107].

Following several failed attempts in November, in December the first realtime two-way LPC packet speech communication took place over the ARPAnet at 3.5 kbs between CHI and MIT-Lincoln Laboratory [68, 38, 34, 133]. The experiment used the basic Markel and Gray implementations of the Itakura-Saito LPC algorithms [38, 101, 102] coupled with the NVP. CHI used an MP-32A signal processor plus an AP 120 (in some publications such as the formal NVP document, this is listed as an AP-90 array/arithmetic coprocessor), while Lincoln Laboratory used a TX2 and FDP. Unlike SCRL, CHI had the necessary hardware and connection data rate for the experiment. It was ironic that Glen Culler's company used the software largely developed by the potential PhD student he had rejected a few years earlier.

One could say that this December 1974 experiment was the first successful voice over IP (VoIP) experiment, but IP did not yet exist! It was in fact the realtime requirements — including those for realtime speech and video communication — that led to the separation of IP from TCP and the eventual formation of the TCP/IP protocol that is the foundation of the Internet. I find it quite amazing that the time from the creation of the NSC project to demonstrate the feasibility of realtime packet speech on the new ARPAnet and the successful demonstration was less than two years.

This is the climax of our principal story, but there were many derivative threads that merit telling. The NSC collaborative effort had a profound influence on several projects distinct from packet speech, including secure speech communication, improved speech compression, the development of DSP chips, and toys. Furthermore, packet speech would spread from the single land-based ARPAnet to satellite and packet radio networks and eventually to multiple networks and the Internet. The initial efforts at speech transmission over PRnet were not as encouraging as the successful realtime transmission of voice over the ARPAnet. SRI measured high packet loss rates of 40%! These were actually packets not received in time, so the reliability being built into TCP was no help. This number and the extreme size of TCP headers (360 bits) added fuel to fiery conversations between Danny Cohen and Vint Cerf

regarding the problems of realtime signal transmission over TCP. It also caused issues in the PR community. SRI showed that a major component of the rate loss was the processing delays in each PR node. The PR nodes all used National Semiconductor's IMP-16 processor chips, which were the first 16-bit chips made available commercially. The radio contractor, Collins, went back to the drawing board and designed a packet radio with better performance. The result would eventually be a dual-processor design that solved the delay problem (at least for moderate traffic levels). But it would not be until 1978–1979 that promising results with realtime packet voice on PRnet would be demonstrated and 1981 for the first successful demonstration of LPC speech on PRnet.

In June, Steen Gray and John Markel published their spectral flatness measure and showed the equivalence of flattening the spectrum and the autocorrelation method for solving the LP problem [46].

During 1974, the National Security Agency (NSA) completed development of its Secure Terminal Unit I (STU I) under the direction of Tom Tremain. Largely through Tom, the NSA had been an active supporter of the packet speech effort, both classified and unclassified, because digital speech was more amenable to secure communication than was analog speech. Analog speech "scramblers" were not believed to be sufficiently secure for military, diplomatic, and government applications. NSA had their own resources for the security part of the problem. From its inception, the NSA conducted its own internal work on speech and took advantage of the NSC research to develop its own speech box. The box was produced from 1977 through 1979 and used adaptive predictive coding (APC) with the Levinson algorithm. The boxes cost approximately $35K each. An excellent history of Tom and his role in the development of secure speech can be found in Campbell and Dean [22].

Dave Retz finally left Santa Barbara in 1974 and moved to SRI, where he worked for Doug Engelbart in the Augmentation Research Center (ARC). During the next year he joined the PRnet program.

17

1975: PRnet, TSP, Markelisms, quantization, and residual/voice-excited LP

In a conversation over breakfast during an NSC meeting in Cambridge, Mass., Earl Craighill and Tom Magill of SRI convinced Bob Kahn of the importance of exploring packet speech traffic on the PRnet. Kahn accepted the suggestion, a contract was awarded, and Craighill assumed leadership of the project. Craighill recalls scribbling on napkins with assumptions of 5 ms per PRnet hop (transmission time of a typical speech packet at 100 kbps with a couple of msec processing time). Packet speech immediately became the dominant application on the PRnet. As Craighill describes it,

> *We created quite a demand on the PRnet project people. We were the only "users" of the PRnet with our speech experiments, but they weren't entirely happy because of our nontypical requirements. The entire PRnet software needed to be changed in each PR node whenever we did a speech experiment. Don Alves and Don Cone bore the brunt of this. Darryl Rubin and Jim Mathis were Vint's graduate students at SU when TCP was being implemented. I believe Jim gets the credit for the*

> *first implementation of TCP. Darryl and Jim both came to SRI, Jim responsible for most of the PRnet local software (Terminal Interface Units —TIUs, gateways, . . .). BBN and Collins did the other software — PRUs and Stations (main control nodes). Darryl worked on our NSC project, writting the initial version of the SIUs (Speech Interface Unit). He left SRI and eventually became Microsoft's main Internet architect. Andy Poggio started working on the NSC software, overhauling the SIUs, implementing ST, NVP, and NVCP.*

John Burg's company TSP began work on realtime speech coding hardware using the Burg algorithm. Charlie Davis managed the hardware and algorithm implementations. A strong motivation was the goal of fitting four 2400 bps speech channels into a 9600 bps modem, thereby lowering the cost of business telephones by cramming four channels into each line. They entered the field late, but they hoped to produce a better quality product at a lower price. It would turn out, however, that their hoped-for market did not materialize; their primary customers would be government agencies.

John Markel noted that many algorithms had been published for "optimal quantization" of LPC parameters and observed that any system is optimal according to some criterion. (Al Oppenheim is also credited with having discovered and promoted this fundamental observation.) I recall hearing other memorable Markelisms of the early seventies regarding speech research:

- *Never believe demonstrations based on the training data.*
- *Never trust the quality of the compressed speech if the original is played first.*

The latter aphorism had a variation, which Vishnu Viswanathan recalled as follows:

> *Around 1973, John [Makhoul] played an audio demo consisting of the following sequence: LPC residual signal, the corresponding original speech signal, and the*

> *same LPC residual signal repeated. On the first listen, the LPC residual sounded like noise, but after hearing the original speech, a replaying of the LPC residual sounded surprisingly intelligible. This demo provided two valuable lessons. First, the residual signal still had useful information left in it. In my subsequent work, I developed techniques in an effort to minimize the information in the residual signal. Second, when playing an audio demo to compare speech intelligibility, it is important to play the coded speech first, followed by the original speech (or a higher bit-rate reference) and not the other way around.*

In June Viswanathan and Makhoul of BBN published their work on quantization properties of various equivalent LPC parameter sets, quantifying the effects of quantization error on the reproduced speech spectrum in LPC systems [126]. The work provided guidance for choosing the best parameterization (the log area ratios (LAR) were the best at that time) and on optimally quantizing the chosen parameters. The work resulted in optimally chosen quantization tables that were shared by the NSC community and spawned further work on the theory, algorithms, and applications of optimal quantization in LPC systems (e.g., [48]).

In October, D. Chaffee and J. Omura of UCLA announced an algorithm for designing a codebook of LPC models based on rate distortion theory ideas for speech coding at under 1000 bps [27, 26]. This was the first successful effort at vector quantization (VQ) in LPC, directly choosing the entire model from a codebook of possible models rather than first solving the LP(m) problem and then individually quantizing the parameters.

Two variations on LPC aimed at producing a waveform coder rather than a vocoder appeared during 1975. Both can be viewed as variations on the original APC waveform coder of Atal and Schroeder [3] in that both used the LP parameters as part of a waveform coder. Now, however, the digitization of the LPC coefficients was explicitly considered and the bit rate was reduced by means of downsampling. The

reproduction required the original rate, however, so that some form of nonlinear processing was required to reconstruct the missing high frequency information. The two approaches differed in whether they used an LP model excited by a coded residual, or used the low pass voice information itself together with an LP model to reconstruct the high frequencies. The first is the most relevant for this history, but the second approach is mentioned because of its close similarity and use of LP methods in a speech coder.

Un and Magill of SRI developed a scheme that they dubbed *residual excited linear predictive (RELP)* coding [123, 124]. They extracted an LPC model and used the prediction error filter to produce residuals. The residuals were then downsampled, quantized, and transmitted along with the quantized LPC coefficients. The quantization technique used adaptive delta modulation and the decoder incorporated an ad hoc nonlinear spectral flattener to produce approximately white residuals over the original band. The quantized residuals were then used to drive the reconstructed inverse prediction error filter to produce the final reproduction.

C.J. Weinstein [132] of MIT Lincoln Laboratory developed a *voice-excited linear prediction (VELP)* by applying LP ideas to classic voice-excited vocoders. The VELP system produced coded LP parameters at the encoder, but did not produce LP residuals for quantization. Instead it downsampled a baseband version of the speech and quantized it for transmission. At the decoder, the LP parameters were used to "flatten" or whiten the nonlinearly processed reconstructed baseband signal to produce an excitation for the LP reconstruction filter. The output of this filter was used for the high frequency reconstruction and the baseband reconstruction for the remainder. An alternative use of LP ideas for voice-excited vocoders was published the same year by Atal and Stover [10].

The RELP and related approaches held promise for medium bit rate speech coding because they avoided the pitch detection and quantization difficulties of pure LPC, because the quantized LP parameters were used to find the residual, and because of a potentially better match of the reproduction filter excitation and the true residual. Furthermore, transmitting discretized LP parameters yielded an embedded

or multirate system: one could send at a low rate of, say, 2400 bps using pure LPC or, by transmitting an additional 4000 bps, generate a better excitation at the decoder (converting the system into a waveform coding system rather than a vocoder). RELP and similar systems shared the problem of APC in that the residual error quantization was accomplished open loop. That is, the residual (or voice) excitation was quantized on its own and not in the context of the effect of the resulting overall quantization error in the final reproduction produced by the residual when used to drive the reconstruction filter. Although intuitive, there is no guarantee that a good fit to a residual will yield a good overall fit from input signal to final reconstruction. Six years later a closed loop version would be described, which used a discrete set (or codebook) of possible residuals as did the RELP system, but which would fit the resulting reproduction to the input with minimum distortion. In other words, the sequences in a residual codebook are not themselves fit to speech residuals; instead they drive the inverse prediction error filter to produce the sequences that are fit to the input speech signal. This is the basic idea behind residual codebook excited linear predictive codes developed several years later that use LP ideas to provide excellent quality waveform coders.

Not all speech research was entirely serious. Vishu Viswanathan relates the following anecdote of more playful research at BBN by John Makhoul.

> *In January 1975, John prepared an entertaining audio tape that demonstrated the independent manipulation of the LPC parameters (LPC spectrum, time, and pitch) to perform voice morphing in interesting ways. The demo started with the following sentence spoken by Bert Sutherland, our department manager at the time: "System reliability has become an increasingly important issue." Using control of time, the sentence was first sped up and then slowed down. Then, by a combination of manipulating pitch, time, and sampling rate for playback (a higher playback sampling rate is equivalent to stretching the frequency axis), the same sentence*

appeared to have been spoken by a female! For the third part of the demo, Sutherland appeared to be singing that sentence, with music composed by John (which involved manipulating pitch and time). In the last part of the demo, Sutherland appeared to be singing the same sentence and tune, but now as a duet (with himself)! This demo tape was a big hit with visitors to BBN over the years; it was also played in one of the NSC meetings to the delight of the attendees.

18

1976: Packet Speech Conferencing, Speak & Spell

Danny Cohen in
Digital Voice Conferencing

In January of 1976 the first LPC conference over the ARPAnet based on LPC and NVP was successfully tested at 3.5 kbps with CHI, USC/ISI, SRI, and Lincoln Laboratory participating. In 1978 an amateur dramatization of a conference such as the 1976 conference was made in the video *Digital Voice Conferencing* at USC/ISI, and an MPEG-1 file can be viewed at http://ee.stanford.edu/~gray/dv.mpg. The transcription of the film to MPEG was done by Billy Brackenridge of Microsoft, Inc. The voice of Danny Cohen was dubbed because of concerns about the understandability of his accent following compression. A DEC PDP 11/45 can be seen in the background and the CSVD boxes appear in the film, which is described as "a hypothetical conference using features of two real systems."

The NVP was formally published in March with Danny Cohen as lead author. It speaks so well for itself and the cooperation and friendship among the participants that it merits quoting excerpts :

The Network Voice Protocol (NVP), implemented first in December 1973, and has been in use since then for local and transnet realtime voice communication over the ARPAnet at the following sites:

- *Information Sciences Institute, for LPC and CVSD, with a PDP-11/45 and an SPS-41.*
- *Lincoln Laboratory, for LPC and CVSD, with a TX2 and the Lincoln FDP, and with a PDP-11/45 and the LDVT.*
- *Culler-Harrison, Inc., for LPC, with the Culler-Harrison MP32A and AP-90.*
- *Stanford Research Institute, for LPC, with a PDP-11/40 and an SPS-41.*

The NVP's success in bridging among these different systems is due mainly to the cooperation of many people in the ARPA-NSC community, including Jim Forgie (Lincoln Laboratory), Mike McCammon (Culler-Harrison), Steve Casner (ISI) and Paul Raveling (ISI), who participated heavily in the definition of the control protocol; and John Markel (Speech Communications Research Laboratory), John Makhoul (Bolt Beranek & Newman, Inc.) and Randy Cole (ISI), who participated in the definition of the data protocol. Many other people have contributed to the NVP-based effort, in both software and hardware support.

The development, implementation, and successful testing of realtime packet speech communication and hence of VoIP has its origins in this group, which had been established and guided by Bob Kahn. While the specific nature of LPC was not essential to the successful development of realtime signal processing on the ARPAnet and eventually the ubiquitous presence of speech, audio, and video on the Internet, the SCRL open source implementations of the Itakura/Saito LPC algorithms were used in this initial demonstration.

In the same year a separate historical thread involving LPC began at Texas Instruments with the development of the Speak & Spell toy by Larry Brantingham, Paul Breedlove, Richard Wiggins, and Gene Frantz.

Prior to joining TI, Wiggins had worked on speech algorithms at MITRE in cooperation with Lincoln Laboratory. According to Vishu Viswanathan, during that time Wiggins contacted John Makhoul and Vishu to chat about linear prediction and LPC synthesis. Years later Wiggins told Vishu that TI had made liberal use of a 1974 BBN report [90] that gave extensive details about the BBN LPC coder. Wiggins also visited Itakura and Atal at Bell (Itakura had a visiting appointment at Bell), NSC, and George Kang at NRL. While at TI he visited SCRL and USC/ISI in the summer of 1977. Wiggins asked many questions and was quietly and successfully absorbing the many recent advances in speech processing.

Linear Prediction of Speech by J.D. Markel and A.H. Gray Jr was published by Springer-Verlag, fulfilling John Markel's goal set seven years earlier. The book became for many years the standard reference for linear prediction methods in speech. In the words of Joseph P. Campbell of MIT Lincoln Laboratory, who worked at the NSA during its many years of active participation and support for the development of speech compression and recognition,

> *it was considered basic reading, and code segments (translated from Fortran) from this book (e.g., STEP-UP and STEP-DOWN) are still running in coders that are in operational use today (e.g., FED-STD-1016 CELP [21]).*

The goal of publishing the book had been a priority for Markel and the book was finished in a flurry of writing, editing, and correcting. He experienced a serious letdown once the goal was accomplished until the next year when he made the decision to leave SCRL to form his own company. Musing on this years later he remarked that one of the best pieces of advice he could give to young researchers would be to set new goals as you near achieving old ones. Accomplishing a long-held dream may not provide the satisfaction you expect if you do not have another

goal lined up.

On the research side, Gray and Markel published a paper on distance measures for speech processing, formalizing and comparing the Itakura-Saito distance and many others that had been developed during the previous several years [47].

During the summer, initial tests were conducted communicating data from a mobile terminal on PRnet, using a van equipped and operated by SRI, to a terminal on the ARPAnet.

The SRI van

An implementation of TCP developed at SRI was used; these original experiments involved only data, not realtime signals [109]. Years later the van was donated by SRI to the Computer History Museum in Mountain View, California, where it can now be seen.

In 1975 the Atlantic Packet Satellite Network (Atlantic SATnet) began operation on the Intelsat IV satellite. The network, also under the direction of Bob Kahn, was a cooperative effort among the US (ARPA) and the UK initially; later Norway, Germany, and Italy were connected. Of particular note was the role played by Irwin Jacobs, of Linkabit and later Qualcomm fame, who was responsible for the overall network architecture [75]. Comsat, Linkabit and BBN all played key technological roles in implementing the network. Like PRnet, SATnet was a separate packet network, but the intention was to permit smooth communication through all of the networks. TCP had been designed for this purpose, allowing data originating in one network to flow through another to a destination in yet another. The Atlantic SATnet had very low bandwidth links (9600 baud), but more importantly, for speech traffic, very long delays—on the order of seconds. This made it quite unsuitable for speech traffic. In the words of Earl Craighill,

> *The main problem was in the channel access protocol. Basically, several round trip times (each round trip is*

> *about a quarter second) were needed to make a reservation so that a packet of data could be sent. This, incidently was a similar problem to one we encountered with the IMP-to-IMP protocol (NCP) in our initial ARPAnet experiments. I can believe that is why Jim Forgie et al. had trouble with their speech experiments over that network. I did considerable experiments with Paal Spilling over the Atlantic SATnet first on ARPA contracts, then on Navy contracts. We couldn't hold very good speech conversations because of the long delays. It was more of a push-to-talk system where you would talk in paragraphs rather than short sentences.*

One-way delays of a second or more were common, and two-second delays in the round trip made conversation difficult. The delays had multiple causes, but major contributors were reservation schemes requiring that all links be reserved before actual packets could be transmitted and the retransmission of lost packets.

In the development and optimization of a speech coding algorithm, there is a clear need for a quantitive measure of the quality of the speech coder output. To facilitate automated speech quality evaluation needed for speech coder development and optimization, BBN proposed a framework for objective speech quality measurement [93] and reported results from their work on objective speech quality evaluation of both non-realtime and realtime speech coders in subsequent ICASSP conferences. BBNs initial work in this area served as the focal point for subsequent investigations by a number of researchers, including Prof. Tom Barnwell of Georgia Tech, who did extensive work in this area and also published a book on objective speech quality evaluation [13]. The original framework introduced in 1976, along with perceptually sensitive distance measures developed by others, was incorporated in the now widely-used ITU Recommendation P.862 called PESQ (Perceptual Evaluation of Speech Quality).

19

1977: STI, STU, Packet Speech Patent, IP Separation, and MELP

In 1977 John Markel left SCRL to found Signal Technology Inc. (STI). He took with him several research grants for LPC and digital filtering, but the initial focus was on the first STI product — the Interactive Laboratory System (ILS), a realtime signal processing package based on SCRL software. The system had been inspired by Culler's On-line system, but Culler's system depended on special purpose hardware, while the SCRL/ILS package was entirely software, open source Fortran which could be modified by the users. The ILS was widely used in university and industrial speech research for many years until being largely supplanted by Unix software from Entropic Speech, Inc, a company founded by John Burg and John E. Shore in the late 1980s.

Development began for STU II, an LPC/APC system for secure speech developed by the NSA. Development took place from 1977–1980 and it was produced from 1982 through 1986 at a cost of about $13K/unit.

In April 1977, James Flanagan of Bell Laboratories, Inc. applied for a patent for "packet transmission of speech' and U.S. Patent 4,100,377 was subsequently granted in 1978 [34, 35]. The seeking and granting of this patent was perceived by many members of the NSC as being

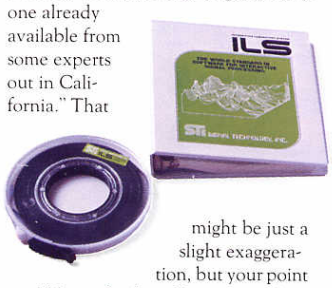
WHY YOU SHOULDN'T WRITE YOUR OWN SIGNAL PROCESSING SOFTWARE.

BY A.H. "STEEN" GRAY, Jr., Ph.D.
Vice President, Signal Technology, Inc.

I've been in this business long enough to know that some things never change. The "make or buy" quandary as it applies to software is a good example. "We've got some expensive programmers; let them earn their keep." How many times have you heard that when you've suggested buying a program package that seems perfectly tailored to your application?

HOW TO ANSWER

If your line of work is signal processing, the answer should be reasonably simple. Just say, "It would take us ten years and a bundle of money to come up with a package as good as the one already available from some experts out in California." That



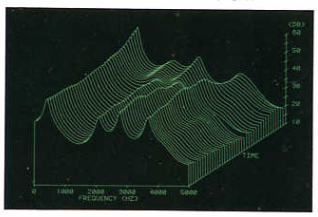
might be just a slight exaggeration, but your point would be well taken. You see, we do have the last word in interactive signal processing software. It's called ILS, and with over 200 installations worldwide, it is often referred to (and not just by us) as the "world standard."

WHAT ILS IS

ILS is a highly modular set of FORTRAN programs that make up a sophisticated interactive software system with standard file structures, documentation and ongoing support. To date, it is performing with excellent results in a wide variety of industries and technologies, including: **speech—noise and vibration—acoustics—biology—medicine—simulation—digital filtering—sonar—radar—seismic—** and some we aren't being told about.

Many of our users also find DACS, our Data Acquisition and Conversion Software, and APAS, our Array Processor Application Software, of great value in their applications.

WHAT COMES OUT



With any compatible computer system and the appropriate terminals, ILS will give you: **pattern analysis—digital filtering—signal editing—3-D displays—modeling—correlation—convolution—spectral density—signal displays—coherence—** and maybe even a picture of your Aunt Sally, if that's what you want.

THE POINT IS

The important thing is, ILS is available *now*. It has been proven in a multitude of applications around the world. And it can cost you a lot less in time and money to buy it from us rather than developing a comparable package yourself.

FREE DEMO

If you have a compatible graphics terminal and modem, we can arrange to give you an on-line demonstration of ILS. Simply call (toll free) and ask for our ILS marketing representative at (800) 235-5787. We also have a video tape that illustrates many of the features of ILS and its capabilities. We'll be happy to send it to you.

IF YOU'RE STILL WITH US

If you've read this far, you probably have some kind of interest in signal processing. I'd be delighted to send you a reprint of the series of three articles on digital filtering that I co-authored with John Markel. Write to me at the address below.

STI SIGNAL TECHNOLOGY, INC.

15 West De La Guerra Street, Santa Barbara, CA 93101
California (805) 963-1552 • **Outside California Call Toll Free (800) 235-5787**
TWX 910-334-3471—SIGNAL TEC SNC

Circle No. 29

Fig. 19.1 STI Advertisement (from 1981)

inconsistent with ARPA's open development process that led to the successful transmission of packet speech over the ARPAnet. Nonetheless, the patent office did grant this patent, and its term of applicability has long since expired; thus, its role in the development of packet speech was not significant over the long run. There were concerns that AT& T might use this patent to restrict the use of packet speech, but they never did.

TSP demonstrated its Model 100 LPC box at the Armed Forces Communication and Electronics Association annual show with a price of $12K/box. The product was considered by many at the time to have the best quality among speech coding boxes at 2400bps, especially in comparison with the STU II then available. Charlie Davis, who worked at TSP at the time, recalls other LPC-based speech boxes at the time produced by Telemux and by GTE-Sylvania, but I have not been able to find further information on these products. I have learned from Al Levesque that speech coder development was done at Sylvania Applied Research Lab (ARL) during the late 1960s as contract work for the NSA and that those involved included Levesque, Harry Manley, Joe deLellis, Harry Shaffer, John O'Brien, Calvin Howard, Richard Freudberg, Jim Stoddard, and Jay Luck. The work likely paralleled that of the NSC because of Tom Tremain's active involvement with NSC and his directing the speech program at NSA. Levesque recalls that the GTE work in the seventies did follow the Atal/covariance approach rather than the Itakura/autocorrelation approach. When ARL was closed in the mid 1970s, the speech work moved to GTE Government Systems, where it involved Tony Busa, Arnie Michelson, Aaron Goldberg, Al Arcese, and George Rosen. Work on speech coding for the NSA, including LPC, continued through the STU III.

Unlike the NSC public algorithms, the details of Burg's implementation were proprietary, and TSP folks recall turning down requests from the NSA for the implementation details. Various branches of the government began buying the TSP boxes and soon became TSP's best customer. Boxes went to the military, the NSA, and the White House. TSP made two versions, one of which was carefully shielded to avoid any external radiation — a property insisted upon by the intelligence community. TSP people recall James Bond-like deliveries of boxes to

men in black at airports. The product continued as viable product through the mid 1980s, when STU III provided better quality at lower cost.

In August Danny Cohen, Vint Cerf, and Jon Postel at USC/ISI discussed the need to handle realtime traffic – including speech, video, and military applications. They agreed that the best plan was to extract IP from TCP to provide a separate protocol suitable for realtime data, and they created the user datagram protocol (UDP) for nonsequenced realtime data.

During 1977–1978, BBN proposed and developed a mixed-source excitation model for low bit rate speech coders [94]. Instead of the traditional binary excitation model (pulse sequence for voiced sounds and random noise source for unvoiced sounds), the new model used both sources: a pulse sequence at low frequencies and random noise at high frequencies, with the cutoff point being a parameter that was extracted from the speech and transmitted. The mixed-source model minimizes the buzzy speech quality typical of the binary model and leads to more natural-sounding speech. The mixed-source model has been subsequently used by a number of researchers. All modern low bit rate speech coders, including the DoD 2.4 kb/s speech coding standards, MELP (mixed excitation linear prediction) and MELPe, use the mixed-source excitation model.

20

1978: IP, PRnet, and Speak & Spell

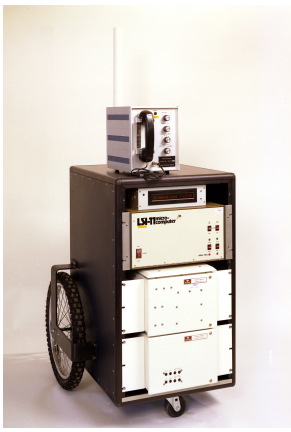
In January IP was officially separated from TCP in version 3 [25]. The TCP/IP suite stabilized with version 4, which is still in use today. It is ironic that the current popular view is that VoIP is novel — when in fact IP was specifically designated to handle realtime requirements such as packet speech and video. In a sense, VoIP existed before IP did and it contributed to the birth of IP. Interestingly, it is often claimed that TCP/IP was not designed for use on wireless networks when, in fact, it was designed specifically to accommodate the wireless PRnet. These are two of the lingering myths of the TCP/IP history.

Earl Craighill and his colleagues at SRI began a series of experiments on a specially configured Lab net (with coaxial cables rather than antennas) and created a specification for a special PRnet protocol suite (Cap5.6S) that worked well for speech traffic. They also developed some algorithms for changing the packet lengths to account for network changes (traffic loads, topology changes due to mobility). These experiments led to the design and implementation of a speech compression and packetization system that could be installed in the SRI van, which was equiped with packet radio, networking, and eventually speech hardware. Experiments began with voice communication between mobile

PRnet nodes and fixed ARPAnet nodes using the van. The experiments again confirmed that the original TCP, as implemented, was not appropriate for realtime traffic — like speech conversations — reinforcing the pressure to separate IP from TCP. Constraints on the number of transmited packets through multihop networks and the processing time required hampered progress during the initial work, because significant changes were needed to the network protocols. Through the next several years many van trips, data recordings, and analysis would lead to the final configuration demonstrated successfully in June 1982.

The mobile speech terminal developed by SRI is shown below left. On the right is a photo of Jan Edl taken inside the van intended to show that an ordinary telephone could be plugged into the PRnet equipment. (Photos courtesy of Don Nielson of SRI. See his excellent article on the SRI van [109].)

Earl Craighill in the SRI Van



During April and May LPC conferencing over the ARPAnet was demonstrated using variable frame rate (VFR) transmission (2–5 kbps) among CHI, USC/ISI, and Lincoln Laboratory using a variable-rate LPC algorithm developed by Viswanathan and his colleagues at BBN.

The basic ideas of the method had been described by Makhoul, Viswanath, Cosell, and Russell in 1974 [90], but it was not published until later in 1977 [127] (and subsequently summarized in a review article [128]). The key idea was to transmit coder parameters only when significant changes have occurred since previous transmission, an approach shared by the SRI DELCO algorithm.

Viswanathan provided the specification for a VFR-based LPC speech coding system for use by the NSC community, and this was used in all subsequent implementations and ARPA demonstrations. The specification involved separate VFR techniques for LPC coefficients, pitch, and gain and required a 3-bit frame header that indicated which parameter type(s) were being transmitted for a given frame.

In June a completely different thread of the story arrived on the scene when the TI Speak & Spell toy hit the market. It was the first consumer product incorporating LPC and the first single chip speech synthesizer. It was also one of the first DSP chips [136]. The device incorporated speech synthesis from stored LPC words and phrases using TMC 0281 one-chip LPC speech synthesizer. A version of BBN's VFR technique was used. Before the product announcement, Wiggins called Markel, Makhoul, Atal, and Gold to acknowledge their contributions to speech and to announce the Speak & Spell. Markel replied to Wiggins with a question — if he had been so helpful to the development of the device, where were his royalties? Wiggins told him not to worry, they would arrive soon. A week later Markel received his own Speak & Spell. Randy Cole realized why Wiggins had asked so many questions when a package with a Speak & Spell arrived out of the blue at USC/ISI. Randy later wrote

> *As soon as we found out about the Speak and Spell I asked Richard for the specs of the TI LPC, and some chips. After signing away my firstborn, I went to TI and came back with the specs and chips. It was pretty simple to change our LPC analysis code, which by then was running on the FPS AP-120B, to output the correct parameters. We first built a little box with a ROM and TI chip and took it to an NSC meeting. When you*

Speak & Spell Development Team: Gene Frantz, Richard Wiggins, Paul Breedlove, and Larry Brantingham
(Thanks to TI and Gene Frantz for permission to use the photo.)

> *pressed the button it said "Welcome to the NSC meeting." Later we built a version with a serial input port and hooked that to the PDP-11 for synthesizing LPC transmitted over the net.*

John Makhoul of BBN also received a complementary Speak & Spell.

I well remember buying my own Speak & Spell as soon as it came out, and my kids Tim and Lori using it constantly for weeks. Thanks to some help from Gene Frantz following the SWIM workshop, it is running again and occupies a prime location in my office bookshelves, displaying the autographs of Gene Frantz and Richard Wiggins.

A second sattelite packet network, Wideband SATnet (WBSAT-

net), with an improved bit rate of 3 Mbps was initiated in 1978 under the sponsorship of ARPA and the Defense Communications Agency (DCA) [62], partially in response to the speed and delay problems of the Atlantic SATnet. WBSATnet considered packet voice from the beginning, and in addition to increased speed it made an important change to the channel access protocol. A short packet (of, say, LPC coded speech) could make its own reservation and get transmitted in one satellite hop. Nodes were placed at SRI, Lincoln Lab, USC/ISI, and DCEC.

Late in the year, Gray, Buzo, Matsuyama, Gray, and Markel presented clustered LPC vector quantization (VQ) using the Itakura-Saito distance as a quality measure at conferences [56, 18]. Steen Gray gave a talk on "Speech compression and speech distortion measures" and played a tape of 800 bps vector quantized LPC speech at the 1978 Information Theory Workshop in Lake Tahoe. The tape had been finished just in time for the workshop, but unfortunately when played it sounded awful. A quick phone exchange with John Markel yielded the reason, it turns out no one had listened to the original data used to train the codebooks and the test data used to test them. Both were awful. Another lesson for speech researchers was clear — a faithful reproduction of lousy original speech will not impress anyone. The problem was fixed and the method refined over the next two years [56, 18, 19, 83, 20, 2], eventually evolving to produce good quality 800 bps systems [137, 138]. Its descendants still exist in some very low bit rate coding systems.

21

1979: Satellite Networks

Packet speech migrated to satellite networks in 1979 with ARPAnet/Atlantic SATnet conferencing using LPC10 at 2.4kbps among USC/ISI, Lincoln Laboratory, the Norwegian Defense Research Establishment (NDRE), and UCL (Cambridge). Jim Forgie participated in the demonstration, but recalled over two decades later a somewhat embarrassing demonstration before the fully successful calls. He was supposed to demonstrate the system at a meeting in Washington DC with a conversation between Washington and the UK, but prior to the demonstration it was found that the link only worked in one direction — he could hear the UK but they could not hear him. So before the demonstration a dialog was scripted and rehearsed, and the demonstration appeared to be successful.

WBSATnet remained a more attractive option than Atlantic SATnet for speech. In addition to higher speed and smaller delays, another compelling reason for using the WBSATnet was the freedom indeveloping gateway protocols. Again quoting Earl Craighill,

> *ARPAnet and even the PRnet were primarily developing data protocols embodying the IP notion of stateless*

> *transmission. That is, each individual packet had all the information necessary for its transport across all networks in its header. So there was no state kept in the gateways. This principle is fundamental to the TCP/IP protocols and has contributed greatly to the open democratic nature of the Internet. As you know, the debate is still going over this issue (now called network "fairness") addressing the issue that ISPs can control traffic to favor their clients by preference routing in their gateways. Well, that is all beneficial for data traffic, but for our early experiments, it seemed a big waste to include the same information in each header of a "stream" of packets like a speech talkspurt. So, the ST protocol was proposed by Jim Forgie. It was basically a gateway level protocol that was intended to use UDP and provide a "stateful" alternative to IP for stream traffic. The key idea was that repeated header information would be kept in the gateways and a short header index was transmitted with each packet. This made the efficiency (ratio of packet information to overall packet length) better for speech, especially highly coded LPC speech.*

Jim Forgie wrote the aforementioned specification for a realtime transmission streaming internet protocol called ST [43] based on joint work with Danny Cohen and Estil Hoversten. Hoversten was then at Comsat, he later moved to Linkabit. The protocol was described as an extension to IP and a lower level protocol than NVP. In parallel, work was going on by Cohen and others at USC/ISI on NVP II, which would build a voice protocol on top of IP (or ST). SRI continued working on packet voice on PRnet, now using improved hardware, ST and IP, and the NVP in speech interface units.

The radical concept of keeping repeated header information in the gateways and transmitting a short header index with each packet was not likely to be in the ARPAnet and PRnet gateways, so instead it was implemented in WBSATnet gateways connecting WBSATnet to PRnet and to Lexnet, a local network developed at Lincoln Lab and provided

to USC/ISI and SRI. Lexnet was a 1 mbit/s coaxial cable net extending about 1000 ft It was intended to concentrate speech conversations for transport over the WBSATnet; hence it was effectively an interface with the gateway.

Lincoln Lab was the prime contractor on the WBSATnet, and speech was the only traffic. WBSATnet and ARPAnet were considered as the two long-haul available networks, and WBSATnet became the primary choice for speech traffic among Lincoln Lab, USC/ISI, and SRI. The ARPAnet was used for a number of conferencing experiments, but apparently not in conjunction with PRnet.

CHI produced LPCap, a dedicated speech compression system based on LPC for ground to air communications which was used by NASA Ames.

22

1981: NVP II and Residual Codebook Excitation

On April 1, 1981, a new Network Voice Protocol, NVP II, which formally built the protocol on top of IP (with an option of using UDP instead), was published by Cohen, Casner, and Forgie [33]. At this point the approach became voice over IP in fact as well as in spirit. The original version omitted Casner's name, but it was subsequently included at Danny Cohen's request. The protocol had been under active development along with the ST protocol since the late 1970s. According to Danny Cohn, ST was born after NVP-II was tried over IP and IP was found wanting over Satellite links.

The CHI-5 array processor for speech processing was developed.

LPC speech was demonstrated on PRnet.

Larry Stewart [119, 120, 50] in his Stanford University PhD research used codebooks searched by a Viterbi algorithm inside a prediction loop with the predictor provided by a minimum distortion selection from a collection of prediction error filters, thus exciting the LPC model by an excitation codebook and selecting the codeword by a closed-loop minimum distortion search. The codebooks were constrained to lie on the branches of a directed graph called a *trellis*. The trellis encoding structure was a low complexity vector quantizer which allowed the residual

error to be chosen from a codebook in a manner minimizing the overall distortion from the input waveform to the final reproduction (as shown in Figure 22.1, taken from a 1984 survey of vector quantization following after Stewart [119]). His simulations were run using the legendary Alto systems at Xerox PARC. Stewart used clustering ideas to gener-

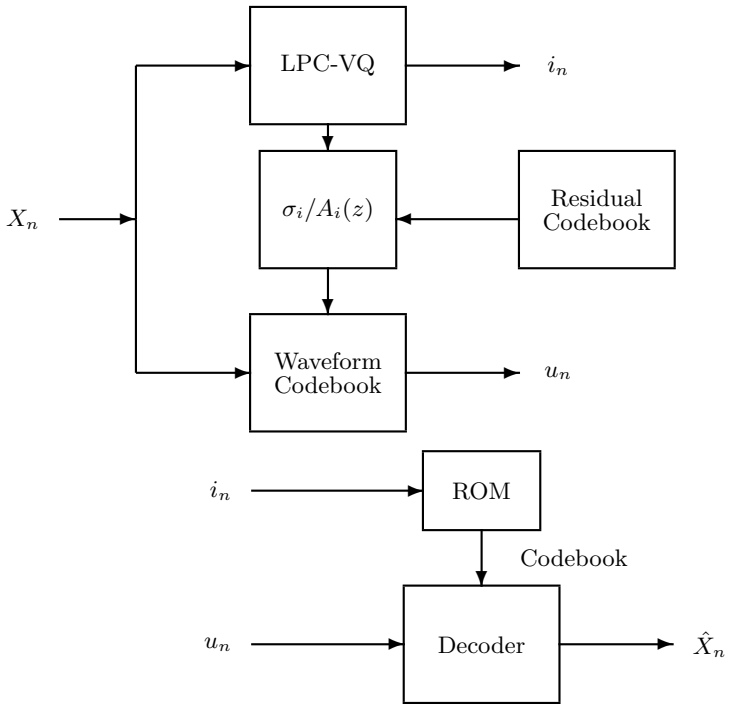


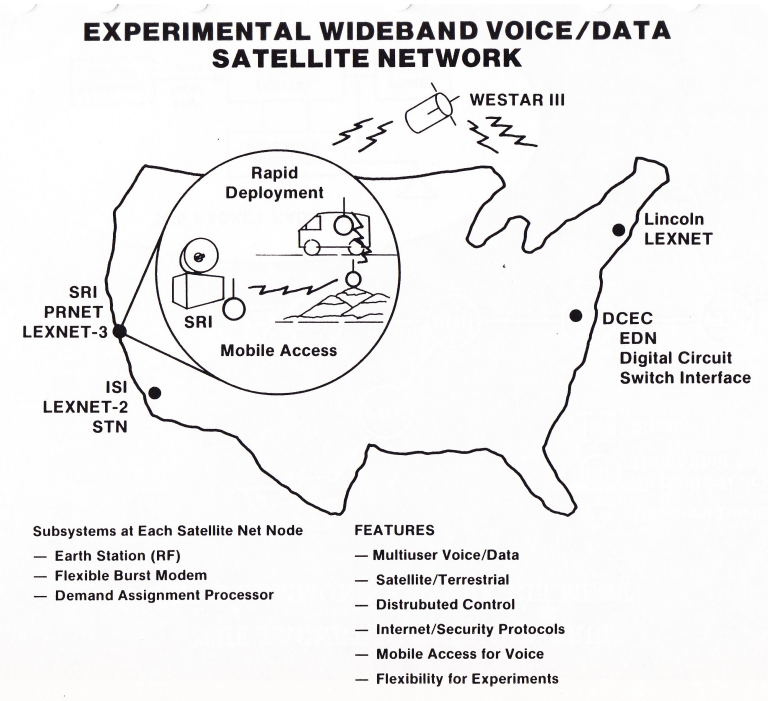
Fig. 22.1 Stewart's "Hybrid Code" (from [50])

ate both the codebook of LPC models and the codebook of residual excitation. He observed in his dissertation that the the reproduction codebooks for the original speech produced by the residual codebooks driving the reproduction filter were the same as those arising in RELP codes, the significant difference in quality arose from the way the codebooks were used. In the RELP codes the effective vector quantization was applied to the residuals, a minimum distortion rule was used to choose a reproduction residual vector to fit a true residual vector, and the reproduction used to drive the reproduction filter to obtain the re-

production. In the hybrid code, however, the minimum distortion rule was used to match the original signal to the output of the reproduction filter driven by residual codewords. The latter system optimized the matching of final waveform to the original, not the matching of residual sequences. The decoder then used the residual codeword as an excitation for the LPC model filter.

23

1982: Voice through the Internet



Three Network Voice

True robust internetwork packet voice was finally convincingly validated in 1982 with demonstrations of LPC mobile PRnet to USC/ISI on April 23, to WBSATnet (at Lincoln Lab) on May 4, and WBSATnet (Lincoln Lab and USC/ISI) on June 3. During these demonstrations the SRI van was driving down Bayshore Freeway, and the packets were sent through PRnet, continued through the WBSATnet, and then through Lexnet to Lincoln Lab, to be decoded by equipment developed by G. C. O'Leary, P. E. Blankenship,, J. Tierney, and J. A. Feldman [110, 111]. As with the original 1974 ARPAnet experiments, all three sites used the LPC10 spec for voice encoding, but there were three different hardware and software implementations. This was the first "internet" realtime voice communication! During the two previous years SRI (primarily Craighill and Norwegian Paal Spilling had been working on packet speech on a local experimental network with a variety of protocols, coders, and equipment. The experiments added more evidence to the shortcomings of the original TCP for realtime transmission and to the potential of packet speech on PRnet. The background, atmosphere, and pace of the preparations for these demos is well described by Earl Craighill:

> *So, this led to actually taking our speech stuff to the real PRnet that was implemented in the Bay Area. This caused some consternation in the PRnet support crew, especially when we said we were changing the PRnet protocols. There was a strong culture (and rightly so) to not touch anything that was working. The Bay Area PRnet was a major demonstration facility for ARPA and they paraded many military personnel through its paces. The ride in the van where a general could read his Email back at the Pentagon was often described as a religious experience. But we managed to prevail and protocols were changed (requiring a drive up to the repeaters on Mission Ridge, Mount San Bruno, Mount Ummanum, and the Eichler behind Stanford). We did many runs in the van and finally standardized a route that went from SRI out Willow Road to 101, down to*

San Antonio Road, then back to SRI on Alma Expressway. We knew the places where mobile handoff would occur and collected a lot of data (individual packet headers showing routing info with time stamps) and did a lot of staring at massive printouts. We were told in January 1982 of the June 1982 presentation to high level (no one below a colonel) military and Government types. ARPA was going all out. We just had our wide-band earth terminal installed. LL was still coding ST and the spec was in flux. We also had just got the (blue) CHI processors with their LPC10 vocoder implementation. So we had many pieces to check out and get working together. We were somewhat behind everyone because we had to wait for their piece/final specs before we could do coding and integration into the SIUs. I found the following table [Editor's note: see Figure 23] which indicates the hectic pace right up to the June demo. I was back at LL for the June Demo. Jan Edl was in the Van. Andy Poggio was madly trying to debug the SIU code at SRI. All our tests the night before had failed. So, Duane Adams (The ARPA Program Manager) and I had a backup strategy of vugraphs only to present the SRI part. It was tense. I was in the back of the room on the phone to Andy while Duane was presenting. We were scheduled for the demo at 9:40. At 9:35 , Andy said he had it. I gave a thumbs up to Duane and he turned on the speaker in the conference room. There was never anything as sweet as Jan Edl's voice coming out of the speaker all the way from the BayShore Freeway! Talk about last minute!! Later, I gave Andy a three minute egg timer to use in preparing for future demos.

It turns out that the problem Andy was fighting wasn't a bug in his programs, but rather an incomplete implementation by LL of the ST spec in their Lexnet voice terminals. They had supplied these same terminals to USC/ISI, so up to this point, like systems were talking

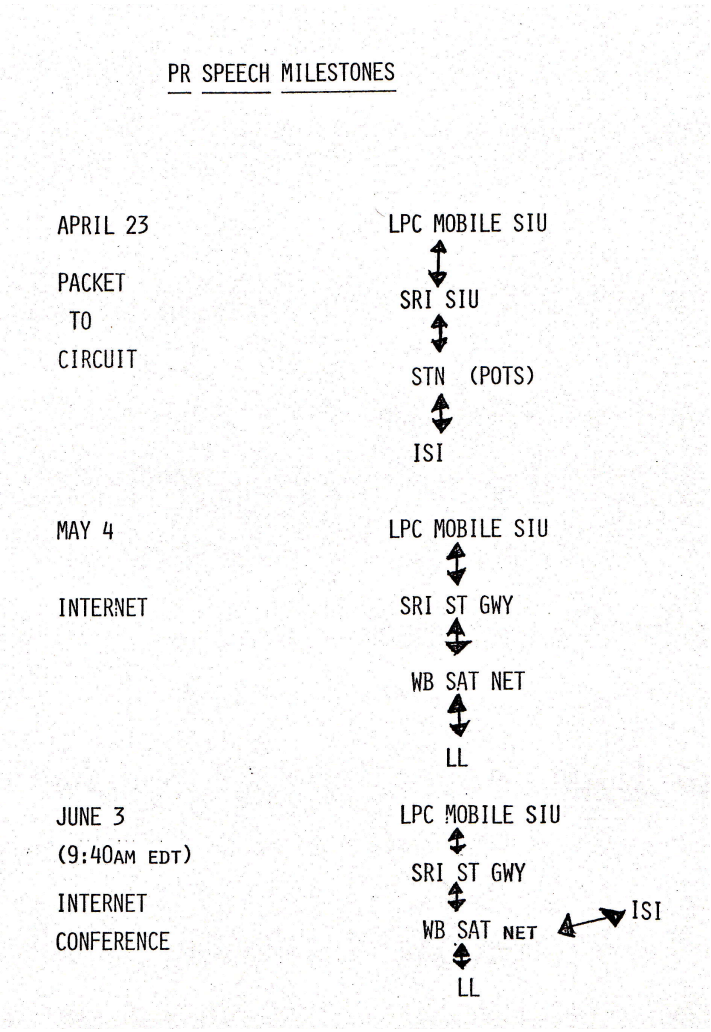


Fig. 23.1 From [1]

to like systems and no problems were noticed. Our SIU implementation was the first "test" of the different ST implementations. The spec was right, Andy had implemented it right, but we couldn't talk to the Lexnet terminals. Andy later (after the demo) uncovered the prob-

> *lem, an incomplete implementation of the initial signin protocol. That explained the strange behavior and was the reason why his work-around (see below) worked! The Lexnet terminals just threw away all the "extra" protocol. Just another day on the frontier!!*

Andy Poggio recalls

> *As the June demo demo time was approaching, I still didn't have a solution in which I had confidence. The only thing I knew for sure was that there was an interoperability problem. I could find no bugs in my code because, as it turns out, there weren't any. I couldn't change my code to make it compatible with the other implementation because I didn't know at the time just how the other version was mis-implemented. During our earlier, brief interoperability testing, I had noticed that the three party conference calls usually failed, but worked once in a while. In all of the very few successful trials, I had connected to the conference call last, as opposed to first or second. So as the deadline approached, I suggested that SRI join last, with a hope that this would work once again. We did, it worked, and the rest is history.*

Andy further reported the following research detour that took place after the successful June 1982 demonstration.

> *Late one night after the June demo, we were working with LPC coding. We had a recording of Willie Nelson handy (it must have been on cassette – no iPods or even CDs in those days) and thought we would pass it through the LPC encoder and then decoder to see how it sounded. As Willie sang, his LPCed voice was more or less intelligible. The amazing thing was that his accompanying guitar playing was entirely absent! There was no trace of the guitar, only Willie's voice came through.*

It was quite miraculous and at the time I had no idea such a thing was possible. When he stopped singing and just played guitar, what we heard was a very weird, almost human sounding guitar.

Lincoln Laboratory's single-card device using 3 NEC7720 chips on 18 sq.in. and 5.5 w was shown to the NSA Director. In combination with NSA's in-house development of a DSP-chip-based 2.4 kbps modem, this provided a convincing demonstration that secure desktop telephones were feasible, and led directly to the decision to go ahead with the STU-III development — which eventually displaced TSP and other LPC boxes.

Two other implementations of the Lincoln Lab device were developed, including the box in the photo, which can still be seen by visitors to Lincoln Lab (photo courtesy of Andreu Veà). The devices are described by Jerry O'Leary:

The device in the picture is one of at least 3 implementations of a compact LPC vocoder developed at Lincoln in the early 80's as part of our DARPA program. All the versions used a set of three chips mask-programmed for LPC using the Nippon 7720 DSP chip. At that time, three different chips were required to handle the complexity the full LPC algorithm. (I think the segments were something like analysis, synthesis, and pitch extraction.) The programming and implementations were done by Joel Feldman.
There were three versions:

(1) The first version was designed to work in the

Packet Voice Terminals and were used in the June 1982 Internet demonstration. The vocoder card passed frames of data to the terminal, which handled the call setup and interface to the network.

(2) The second version was a was implemented on a single card and could operate independently. It was a minimal implementation, designed to show how compact the design could. It generated a 2400 bps synchronous bit stream which could be connected to a standard modem and talk over an ordinary dial-up telephone line. This is the version, we are told, provided the proof of concept demonstration necessary for NSA's decision to proceed with the STU 3 program development. The STU 3 system was based on a more capable DSP chip (from TI?) which was developed a year or so later and which could implement the entire LPC algorithm in a single chip. The STU 3 system was finally decommissioned in December 2009.

(3) The system in the picture was developed as a general speech peripheral to a computer. The device was designed to look to the host computer like a standard terminal. It provided a number of control functions as well. Because of the extra overhead, the interface was actually a 4800 bps asynchronous interface. The intent was to provide researchers with a I/O device which could be used in the development of custom speech processing algorithms and applications. Fifty or sixty of the boxes were built by Adams-Russell under contract to Lincoln. Most of these were distributed to various contractors in the DARPA community.

24

Epilogue

The NSC

. . . it's hard to overstate the influence that the NSC work had on networking. . . . the NSC effort was the first real exploration into packet-switched media, and we all know the effect that's having on our lives 30 years later.
Randy Cole

. . . some of the early work on advanced network applications, in particular packet voice in the 1970s, made clear that in some cases packet losses should not be corrected by TCP, but should be left to the application to deal with. This led to a reorganization of the original TCP into two protocols, the simple IP which provided only for addressing and forwarding of individual packets, and the separate TCP, which was concerned with service features such as flow control and recovery from lost packets.
Barry Leiner et al. [81]

It is said that every silver lining has a cloud, and perhaps the downside of the NSC project was that after packet speech had been convincingly demonstrated on the ARPAnet and then the Internet, it was largely moribund for more than a decade while the Internet spread and access to computers grew (and the economy changed). But with the advent of the VoIP reincarnation of packet speech and the ubiquitous Internet, that all changed. Jim Forgie voiced a concern that I have heard often over time — that the improvement of hardware and bandwidth rendered compression obsolete, so that much of the original packet speech work proved unnecessary. Nonetheless, the initial development of realtime signaling on packet networks stands on its own as a stunning contribution, regardless of subsequent developments. It was the first demonstration of something that over thirty years later might seem obvious, but at the time few were prescient enough to predict (outside of science fiction, where notoriously wrong predictions were also made such as the third millennium use of slide rules, magnetic tape recorders, and photographic film requiring development!). I also do not think that increased bit rates and bandwidth obviated the need for compression. There is still interest in cramming ever more channels onto whatever bandwidth is available, and not all channels have enormous bandwidths (try using a satellite cell phone or Internet connection). In the words of Bob Kahn,

> *Wireless networks will likely have serious bandwidth limitations for the foreseeable future. So, even if the fiber optics world has terabits to spare and compression goes by the boards there, it will still be needed on the wireless networks. And those, in good internet style, will all be connected to everything else.*

Finally, LPC ideas remain in many modern standards, including the increasingly dominant voice over IP.

Realtime Protocols

The subsequent success of the TCP/IP protocol suite, including the transmission of realtime signals including speech, audio, and video has

been so astonishing that it is often taken for granted. Only we elders appreciate how hard the problem appeared to be three decades ago. The separation of IP from TCP allowed the creation of UDP on top of IP, then later RTP (Realtime Transport Protocol) was created on top of UDP, rather than on top of IP, primarily because operating systems provided unprivileged API access to UDP but not to IP directly. RTP evolved in part from NVP. It was developed by the Internet Engineering Task Force's Audio/Video Transport Working Group, which was chaired by Steve Casner from 1992 through 2003. In 1990 the International Telecommunications Union formally adopted packetized voice protocols (G 764). RTP is the dominant protocol for realtime speech and video traffic today and it is likely to grow even more dominant as mobile phone traffic moves increasingly to IP-based systems. RTP also lies at the heart of such popular video protocols as H.261 and H.263. It is often argued that http carries more voice, audio, and video traffic than any other protocol, but much of that is not realtime (and is TCP-based). For example, the enormous video and audio traffic of YouTube™ requires buffering and is not interactive. Modern variations of LPC and CELP dominate digital voice traffic, especially RPE-LPC (regular pulse excited LPC), CELP-VSELP (vector excited LPC), and ACELP (algebraic CELP). These all involve vector quantization of the excitation codebook for LP models within the autoregressive feedback loop, an idea dating back to 1981.

Vector Quantization

John Makhoul writes:

> *In parallel with the ARPA NSC program, and following it, we executed numerous projects in speech compression for different agencies and at all data rates, from 150 bits/s to 16 kbits/s. At the lower data rates, we made heavy use of vector quantization (VQ). In 1984, Ned Neuberg at the Department of Defense, provided funding for me to write a book or a long paper on "any topic you would like". At the time, I had intended to write a book on linear prediction, but I was very much intrigued*

> *by VQ and felt that it was not well understood. So, I went about researching VQ and trying to understand it better. The result was an invited paper (with Roucos and Gish) in the Proc. of IEEE in November 1985 [95]. Among other contributions, the paper presented a model of VQ that decomposed its workings into four components (linear dependency, nonlinear dependency, pdf shape, and dimensionality). Later on, Tom Lookabaugh, under Bob Gray's supervision, provided rigorous mathematical proofs for the different components of the VQ model [10].*

See also [85] for Lookabaugh's development of the ideas of Makhoul, Roucos, and Gish.

Standards and Boxes

Multipulse residual excitation codedbooks were reported in 1982 [11] and randomly generated codebooks in 1984 [12] and 1985 [117]. Perceptually weighted quadratic distortion measures were used to achieve better performance. In [117] the code structure combining residual codebooks with LP models was aptly named *codebook excited LP (CELP)* codes. Charlie Davis recalls that Codex Corporation was thinking along similar ideas of using excitation codebooks to drive LPC models using a minimum distortion match at about the same time. These systems provided very good speech quality at around 4800 bps. The addition of postfiltering by Chen and Gersho [28] to the CELP system in 1987 led to the Federal Standard 1016 CELP coder [21], that included some of the old old Markel/Gray software.

The development of STU III had begun in 1984 and several models went into production in 1987, selling for about $2K each. It was the dominant speech coder for many years and is still in use, although it is being replaced by MELP (multipulse excitation linear prediction) at 2400 bps and G729 CS ACELP at 8,000 bps. The MELP coder uses a five-band version of the mixed-source excitation model introduced by BBN in 1978. Through Tom Tremain, the NSA played a key role in the pioneering work on speech coding and packet speech during the

1970s and 80s. Tom Tremain died in 1993. Voice coding research and development at the NSA came to an end in 2003.

Where are they now?

Glen Culler received the 1999 National Medal of Technology from President Clinton for "pioneering innovations in multiple branches of computing, including early efforts in digital speech processing, invention of the first online system for interactive graphical mathematics computing, and pioneering work on the ARPAnet." In 2000 he received the Seymour Cray Award. Culler died in May 2003. CHI continued to produce signal processors and DSP chips for commercial or military applications. In 1985 CHI became Culler Scientific Systems and developed the Personal Supercomputer and the Culler computer until the appearance of the RISC machines reduced the market. Culler Scientific was subsequently acquired by SAXPY and then by Star Technologies.

In late 1985, Bob Kahn left ARPA to found the "Corporation for National Research Initiatives (CNRI)" where he, along with the CNRI staff, continue to work on fostering research and development efforts that promote the national information infrastructure. CNRI has played a central role in the development of gigabit networking in the US, led the Internet standards process for many years, and is pioneering the development of the "Digital Object Architecture" to manage information on the net. Bob continues to run CNRI and is still personally active in multiple national level research initiatives. Bob has received numerous awards for his work on

the Internet, including the 2004 ACM Turing Award, the Presidential Medal of Freedom, and the National Medal of Technology.

Dr. Cerf left ARPA in 1982 to join MCI, where he helped develop the first commercial electronic mail system known as MCIMail. In 1986, he became employee number two at CNRI, working with Bob Kahn on an early aspect of the Digital Object Architecture involving Knowledge Robots, mobile programs that are known simply as Knowbots. Vint was responsible for CNRI's early efforts in leading the Internet standards process, during which time he became the first President of the Internet Society. In 1994, Vint left CNRI to once again join MCI, where he stayed until 2005, at which time he became Google's "Internet Evangelist." Vint has received numerous awards for his work on the Internet, including the 2004 ACM Turing Award that he received jointly with Bob Kahn. He also received the Presidential Medal of Freedom and the National Medal of Technology.

Danny Cohen had a full career in computer networks and realtime signal processing in industry, startups, and on advisory boards. He remained at USC/ISI through 1993, when he left to co-found Myricom. He is currently a Sun Distinguished Engineer and a member of the National Academy of Engineering. He has been known to boast of his membership in the Flat Earth Society and his association with the Computer Science Department of Oceanview University, in Oceanview, Kansas.

Danny Cohen

Following his visit to Bell Labs, Itakura returned to NTT to continue research on speech, especially on line spectrum pair (LSP) analysis. I visited Dr. Itakura in 1980 and toured the speech research projects at NTT. Itakura told the story of how his contributions to speech processing had not been fully appreciated in the early days. It was not until the announcement of the TI Speak & Spell Toy that NTT management realized that a technology developed in house and generally considered a mathematical curiosity had been successfully implemented and turned into a product by Americans. Itakura was then given the

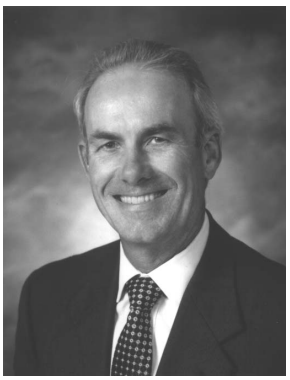
facilities and staff to work on all aspects of a variety of speech coding and recognition projects, ranging from the underlying theory through VLSI implementation.

He was promoted to Chief of the Speech and Acoustics Research Section in 1981. Among his many awards are the 2003 Purple Ribbon Medal from the Japanese Government, the 2005 IEEE Jack S. Kilby Signal Processing Medal, and the 2005 Asahi Prize, the most prestigious nongovernmental award in Japan. He used the financial portion of the Asahi award to fund the Itakura Prize Innovative Young Researcher Award of the Acoustical Society of Japan. In 1984 Itakura left to become a professor of communications theory and signal processing at Nagoya University.

John Burg sold TSP and later cofounded Entropic Processing with John E. Shore. In 1988 the company split into Entropic Speech and Entropic Geophysics to handle their distinct applications for speech and geophysical signal processing. In 1999 Burg and Shore sold Entropic to Microsoft. Burg retired, donating some of the funds gained from Entropic to found the Edwin T. Jaynes International Center for Bayesian Methods and Maximum Entropy at Boise State University in 2000.

Bishnu Atal had a long and successful career at Bell Labs, where he served as head of the Acoustics and Audio Communication Research Department and of the Speech Research Department. He was a Fellow of both Bell Laboratories and AT&T and also served Technical Director at the AT&T Shannon Laboratory. Among his numerous awards are membership in both the National Academy of Sciences and the National Academy of Engineering, the IEEE Morris N. Liebmann Memorial Field Award of the IEEE, and the Benjamin Franklin Medal in Electrical Engineering. He retired in 2002 to become an Affiliate Professor at the University of Washington. Bell Telephone Laboratories, Inc., was renamed AT&T Bell Laboratories and became a wholly

owned company of AT&T Technologies, the former Western Electric. During the 1990s, AT&T spun off most of Bell Labs, including Murray Hills, to Lucent Technologies. A few of the researchers stayed with AT&T and moved to the AT&T Research Laboratory. In 2006 Lucent merged with Alcatel. The Alcatel-Lucent Bell Laboratories in Murray Hill no longer has active speech research.



John Markel became involved in real estate while president of STI; this allowed him flexibility in adjusting the size of his company by expanding or contracting within his own buildings and leasing the rest. His management skills were put to the test when the death of a friend led him to step in to help and eventually manage a 50-unit subdivision outside of Aspen, Colorado. His career path took many turns as he moved farther from his technical past. In 2003 he sold the last of his developments and during the oral history proudly announced his new career as tennis bum.

John Makhoul led the Speech Processing Department at BBN, which was later merged with the Natural Language Processing (NLP) activity to form the Speech and Language Processing Department (later the Speech and Language Technologies Department). He managed that department for some time, but later gave up the management part and became Chief Scientist, Speech and Signal Processing. He also serves as Director of the Science Development Program and is an Adjunct Professor at Northeastern University. In 2006 he published a history of speech processing at BBN [96]. In 2009 he was awarded the James L. Flanagan Speech and Audio Processing Award and Medal by the IEEE for his "pioneering contributions to speech modeling."

Steve Casner's work at USC/ISI advanced from packet voice to packet video and multimedia conferencing over a period of 20 years, ending when he was recruited to a Silicon Valley startup to commercialize the technology. He also served as chairman of the Audio/Video Transport working group of the Internet Engineering Task Force for 11 years to develop the widely-used standard Realtime Transport Protocol

(RTP) for packet audio and video.

Randy Cole was hired by Richard Wiggins at TI to manage his lab, where he remains working on audio R&D.

In 1975 Tom Stockham left the University of Utah for Soundstream, a company he founded, which was the first commercial digital recording company. Tom later acquired early onset Alzheimer's disease and passed away in 2004. In addition to his fame as the father of digital audio, the recapture of Caruso's voice, and the analyzer of the Nixon tapes, Tom won numerous professional awards such as the IEEE Jack S. Kilby Signal Processing Medal and membership in the National Academy of Engineering, but his impact on audio engineering is best highlighted by his winning a Grammy, Emmy, and Oscar.

Steven Boll got a contract with ARPA to investigate noise suppression and left the coding world. In 1982, he left the University of Utah and went to work for ITT in San Diego, where he was a manager for a research group involved with word spotting, speaker and language recognition, and speaker verification. Boll retired in 2004.

Jim Forgie spent his career at Lincoln Lab working on networking, eventually moving from technical work into management and then retiring in 1995. He still lives in Arlington on Stoney Brook Rd, where he and Carma were hosts for the NSC discussions and parties.

Vishu Viswanathan left BBN in 1990 to work for Richard Wiggins at TI. He was elected Texas Instruments Fellow in 1996 and became a Lab Director in 2001, managing TIs R&D activities in speech and acoustic technologies, including R&D support to TIs Voice over IP business. He retired from TI in February 2009.

SCRL eventually left Santa Barbara and moved to USC, where June Shoup remained head and Hisashi Wakita continued as a research scientist. He commuted for a while until his research grant ran out. On the day he found out his grant was not being renewed, a small team from Panasonic came to his office to offer him a job working with the Panasonic Speech Technology Laboratory, a research lab in Santa Barbara that had formerly belonged to Matsushita and which employed several of former SCRL and a few former STI employees. Since Panasonic did not know he was going to be unemployed, they had offered him a full

lab to lure him away from SCRL. Wakita remained with Panasonic until his retirement.

Acknowledgements

My research work in speech during the 1970s and 1980s was partially supported by the National Science Foundation. Thanks to Huseyin Abut for the invitation to speak at the 2004 SWIM workshop, which motivated me to do the historical research and chat with many of the people involved. Thanks to J. D. Markel, A.H. "Steen" Gray Jr., John Burg, Earl Craighill, Charlie Davis, Danny Cohen, Steve Casner, Joe Tierney, and Jim Forgie for interviews and discussions, and to Steven Boll, Joseph P. Campbell, Vint Cerf, Randy Cole, David Culler, Rich Dean, Yariv Ephraim, Jim Forgie, Bob Kahn, Mike McCammon, Jim Murphy, Don Nielson, Gerald O'Leary Andy Poggio, Larry Rabiner, Vishu Viswanathan, Cliff Weinstein, and Richard Wiggins, for chats, critiques, suggestions, and emails. Thanks also to Leslie Cooney for her editorial suggestions and corrections.

References

[1] D.A. Adams, D. Cohen, J.W. Forgie, C.J. Weinstein, E.J. Craighill, J. Makhoul, V. Viswanathan, R. Schwartz, G. C. O'Leary, J. A. Feldman, G. Culler, E. Greenwood, and R. Brodersen, *Packet Speech Program Review Meeting*, MIT Lincoln Laboratory, June 3, 1982. 125

[2] J.-P. Adoul, J.L. Debray, and D. Dalle, "Spectral distance measure applied to the design of DPCM coders with L predictors," *Conf. Record 1980 IEEE ICASSP*, Denver, CO, pp. 512–515, April 1980. 115

[3] B.S. Atal and M. R. Schroeder, "Predictive coding of speech signals," *Proc. 1967 AFCRL/IEEE Conference on Speech Communication and Processing*, pp. 360–361, Cambridge, Mass, 6–8 November 1967. 59, 98

[4] B.S. Atal and M. R. Schroeder, "Predictive coding of speech signals," *Rep. 6th Int. Congr. Acoust.*, Y. Konasi, Ed., Tokyo Japan, Rep.C-5-5, August 1968, 59

[5] B.S. Atal and M. R. Schroeder, "Predictive coding of speech signals," *WESCON Tech. Papers*, Paper 8/2, 1968. 59

[6] B.S. Atal, "Speech analysis and synthesis by linear prediction of the speech wave," presented at the 78th Meeting of the Acoustical Society of America, San Diego, November 1969. Abstract in *J. Acoust. Soc. Am.*, Vol 47, p. 65, 1970. 67

[7] B.S. Atal, "The history of linear prediction," *IEEE Signal Processing Magazine*, pp. 154–161 , March 2006. 57

[8] B.S. Atal an M.R. Schroeder, "Adaptive predictive coding of speech signals," *Bell Sys. Tech. J.*, Vol. 49, No. 8, pp. 1973–1986, October 1970. 59

[9] B. S. Atal and S.J. Hanauer "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoustic Society of America*, Vol. 50, pp. 637–655, August 1971. 67, 70

[10] B.S. Atal and V. Stover, "Voice-excited predictive coding system for low bit-rate transmission of speech," *J. Acoust. Soc. Am.*, Vol. 57, Supplement 1, p. 535, Spring 1975. 99

[11] B.S. Atal and J.R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *Proc ICASSP/82*, May, 1982, pp. 614–617. 132

[12] B.S. Atal and M.R. Schroeder, "Stochastic coding of speech signals at very low rates," in *Proc. Int. Conf. Commun*, ICC'84, pp. 1610–1613, May 1984. 132

[13] T. P. Barnwell III, M. A. Clements, and S. R. Quackenbush, *Objective Measures for Speech Quality Testing*, Prentice Hall, Englewood Cliffs, New Jersey, February 1988. 106

[14] Peter E. Blankenship, "LDVT: High Performance Minicomputer For Real-Time Speech Processing," EASCON '75, pp. 214A-G, 1975. 93

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[16] John Parker Burg, "Maximum entropy spectral analysis," presented at the 37th Meeting of the Society of Exploration Geophysisists, Oklahoma City, Oklahoma, October 1967. 36, 56

[17] John Parker Burg, "A new analysis technique for time series data," presented at the NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics," Enschede, The Netherlands, Aug. 1968, reprinted in *Modern Spectrum Analysis*, D. G. Childers, ed., IEEE Press, New York, 1978. 62

[18] A. Buzo, R.M. Gray, A.H. Gray, Jr., and J.D. Markel, "Optimal Quantizations of Coefficient Vectors in LPC Speech," *1978 Joint Meeting of the Acoustical Society of America and the Acoustical Society of Japan*, Honolulu, HI, Dec. 1978. 115

[19] A. Buzo, A.H. Gray, Jr., R.M. Gray, and J.D. Markel, "A Two–Step Speech Compression System with Vector Quantizing," *Proceedings of the 1979 Int'l. Conf. on Acoustics, Speech and Signal Processing*, pp. 52–55, Wash. DC, 1979. 115

[20] A. Buzo, A.H. Gray, Jr., R.M. Gray, and J.D. Markel, "Speech coding based upon vector quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP–28, pp. 562–574, October 1980. 115

[21] J. P. Campbell, T. E. Tremain, and V. C. Welch, "The Federal Standard 1016 4800 bps CELP Voice Coder," *Digital Signal Processing*, Vol. 1, no. 3 (1991): 145 – 155. 104, 132

[22] J.P. Campbell, Jr., and Richard A. Dean, "A History of Secure Voice Coding: Insights Drawn from the Career of One of the Earliest Practitioners of the Art of Speech Coding," *Digital Signal Processing*, July 1993. An expanded pdf version is available at `http://www.nsa.gov/about/_files/cryptologic_heritage/publications/wwii/sigsaly_history.pdf`. 95

[23] Casner, S.L., Mader, E.R. and Cole, E.R., "Some Initial Measurements of ARPANET Packet Voice Transmission," *IEEE 1978 National Telecommunications Conference*, December 1978.

[24] Vinton Cerf, Robert Kahn, "A protocol for packet network intercommunication," *IEEE Trans. Commun.*, Vol. 22, 627–641, May 1974. 91, 92, 93

[25] "Specification of Internetwork Transmission Control Program TCP Version 3," Vinton G. Cerf, Advanced Research Projects Agency, and Jonathan B. Postel, Information Sciences Institute January 1978. 111

[26] D.L. Chaffee, *Applications of rate distortion theory to the bandwidth compression of speech signals*, PhD Dissertation, UCLA. 98

[27] D.L. Chaffee and J.K. Omura, "A very low rate voice compression system," *Abstracts of Papers, International Symposium on Information Theory*, p. 69, October, 1974. 98

[28] Chen and A. Gersho, "Real-Time Vector APC Speech Coding at 4800 bps with Adaptive Postfiltering," *Proc. ICASSP '87*, pp. 2185–2188. 132

[29] D. Cohen "Specifications for the Network Voice Protocol," USC/Information Sciences Institute, ISI/RR-75-39, March 1976. Also NSC Note 68, January 1976. 91

[30] D. Cohen, "RFC0741: Specifications for the Network Voice Protocol," 22 Nov 1977. Available at http://www.ietf.org/rfc/rfc741.txt. 91

[31] D. Cohen, "A voice message system," in R. P. Uhlig (ed.), Computer Message Systems, pp. 17-27, North-Holland, 1981. (Discusses ARPA voice project.) 91

[32] D. Cohen, "Packet communication of online speech," *Proceedings of the May 4-7, 1981, National Computer Conference*, Chicago, Illinois, pp. 169–176 1981.

[33] D. Cohen, S. Casner, and J. W. Forgie, "A Network Voice Protocol NVP-II," USC/ISI and Lincoln Laboratory Report, April 1, 1981. 119

[34] D. Cohen, Excerpts from "Packet Speech Program Review Meeting," Sponsored by DARPA, Hosted by MIT Lincoln Laboratory, "Network voice protocols," USC/ISI, pp. 40–59, 'Packetized Speech Overview," pp. 17–23, 3 June 1982. 91, 94, 107

[35] Danny Cohen, "Realtime Networking and Packet Voice," SIGCOM'99. 107

[36] T. M. Cover and J. A. Thomas, *Elements of Information Theory, Second Edition*, Wiley Interscience, Hoboken, New Jersey, 2006. 41

[37] Glen J. Culler, "An Attack on the Problems of Speech Analysis and Synthesis with the Power of an On-Line System," *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, Donald E. Walker, Lewis M. Norton (Eds.): Washington, DC, May 1969. pp. 41–48. 61

[38] Glen J. Culler, Michael McCammon, and J.F. McGill, "Realtime implementation of an LPC algorithm," *Culler/Harrison Inc. Quarterly Technical Report on Speech Signal Processing Research at CHI, during Nov. 1974–April 1975*, May 1975. 94

[39] C. C. Cutler, "Differential PCM," U. S. Patent 2 605 361, July 29, 1952. 58

[40] L. D. Davisson, "The theoretical analysis of data compression systems," *Proceedings of the IEEE*, Volume 56, Issue 2, Feb. 1968, pp. 176 – 186. 76

[41] P. Elias, "Predictive coding I," *IRE Trans. Inform. Theory*, Vol. 1, No. 1, pp. 16–24, Mar. 1955. 57

[42] J.L. Flanagan, *Speech Analysis, Synthesis, and Perception*, Springer, 1965. Second Ed. 1972. 62

[43] James W. Forgie, *ST - A Proposed Internet Stream Protocol*. IEN 119, M. I. T. Lincoln Laboratory, 7 September 1979. 117

[44] Gold, Bernard (invited paper), "Digital Speech Networks", Proc. IEEE Vol. 65, No. 12, Dec. 1977. (Discusses ARPA voice project.)

[45] Eva C. Freeman, Ed., *MIT Lincoln Laboratory: Technology in the National Interest*, Lexington, Mass.: MIT Lincoln Laboratory, 1995. 73

[46] A.H. Gray Jr and J. D. Markel, "A Spectral-Flatness Measure for Studying the Autocorrelation Method of Linear Prediction of Speech Analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 22, No. 3, June 1974, pp. 207–217. 45, 95

[47] A.H. Gray and J.D. Markel, "Distance Measures for Speech Processing," *IEEE Trans. on ASSP*, Vol. 24, no. 5, Oct. 1976. 105

[48] A. H. Gray, Jr., R. M. Gray and J. D. Markel, "Comparison of optimal quantizations of speech reflection coefficients," *IEEE Trans. on Acous., Speech & Signal Process.*, Vol. ASSP–25, pp. 9–23, Feb. 1977. 98

[49] R.M. Gray and J.C. Kieffer, "Asymptotically mean stationary measures," *Annals of Probability*, vol. 8, pp. 962–973, Oct. 1980. 19

[50] R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, Vol. 1, pp. 4–29, April 1984. 119, 120

[51] R.M. Gray, *Probability, Random Processes, and Ergodic Properties, Second Edition*, Springer, New York, 2009. First Edition published January 1988, corrected version available at http://ee.stanford.edu/∼gray/arp.html. 19

[52] R. M. Gray, A. Buzo, A. H. Gray, Jr., and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acous., Speech and Sign. Proc.*, Vol. ASSP–28, pp. 367–376, Aug. 1980. 45, 53

[53] R. M. Gray and L. D. Davisson, *Introduction to Statistical Signal Processing*, December 2004, Cambridge University Press, Cambridge, UK. Individual copies available for download at http://ee.stanford.edu/˜gray/sp.html. 11

[54] A.H. Gray, Jr., and D.Y. Wong, "The Burg algorithm for LPC Speech analysis/synthesis," *IEEE Trans on Acoustics, Speech, and Signal Processing,''* Vol. 28, No. 6, December 1980, pp. 609–615. 62

[55] R.M. Gray, A.H. Gray, Jr., G. Rebolledo, and J.E. Shore, "Rate distortion speech coding with a minimum discrimination information distortion measure," *IEEE Transactions on Information Theory*, vol. IT–27, no. 6, pp. 708–721, Nov. 1981. 45, 57

[56] R.M. Gray, A.Buzo, Y. Matsuyama, A.H. Gray, Jr. and J.D. Markel, "Source coding and speech compression," *Proceedings of the 1978 Int'l. Telemetering Conf., 24*, Los Angeles, CA, pp. 871–878, Nov. 1978. 115

[57] R.M. Gray, "Toeplitz and Circulant Matrices: a Review," *Foundations and Trends in Communications and Information Theory*, vol.2, no. 3, pp. 155-329, 2005. Originally published as Information Systems Laboratory Technical Report, Stanford University, 1971. Revised and reprinted numerous times and currently available at http://ee.stanford.edu/˜gray/toeplitz.pdf. 20, 22, 27, 34, 35

[58] Robert M. Gray, "The 1974 origins of VoIP," *IEEE Signal Processing Magazine*, Vol. 22, July 2005, pp. 87–90. 1, 4

[59] U. Grenander and M. Rosenblatt, *Statistical Analysis of Stationary Time Series*, John Wiley & Sons, NY, 1957. 66

[60] U. Grenander and G. Szegö, *Toeplitz Forms and Their Applications,* University of Calif. Press, Berkeley and Los Angeles, 1958. 20, 22, 26, 35, 87

[61] J. Hájek, "On linear statistical problems in stochastic processes," *Czechoslovak Math J.*, Vol. 12, pp. 404–444, 1962. 52

[62] H. Haggestad, *IEEE Communications Magazine*, "An overview of packet-switching communications," Vol. 22, No. 4, pp. 24–31, April 1984. 115

[63] E.M. Hofstetter, "An introduction to the mathematics of linear predictive filtering as applied to speech analysis and synthesis," *Lincoln Laboratory Technical Note* 1973-36, 12 July 1973. 87

[64] E.M. Hofstetter, P.E. Blankenship, M.L. Malpass, and S. Seneff, "Vocoder Implementations on the Lincoln Digital Voice Terminal," *Proc. of Eascon 1975*, Washington, D.C. (Sep.–Oct. 1975). 88

[65] Hofstetter et al., "Microprocessor Realization of a Linear Predictive Vocoder," *Lincoln Laboratory Technical Note, 1976-37* (Sep. 1976).

[66] E. M. Hofstetter, "Microprocessor Realization of a Linear Predictive Vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, No. 5, pp. 379-387, Oct. 1977. 88

[67] ISI/SR-74-2 Annual Technical report May 1973-June 1974. 91

[68] ISI/SR-75-3, Annual Technical report, May 1974-June 1975. 94

[69] ISI/SR-76-6, Annual Technical report, July 1975 – June 1976. 91

[70] F. Itakura and S. Saito, "Analysis synthesis telephony based upon the maximum likelihood method," Reports of 6th Int. Cong. Acoust., ed. by Y. Kohasi, Tokyo, C-5-5, C17–20, 1968. 45, 53

[71] F. Itakura and S. Saito, "Analysis synthesis telephony based on the partial autocorrelation coefficient," Acoust. Soc. of Japan Meeting, 1969. 53, 66

[72] F. Itakura and S. Saito, "A statistical method for estimation of speech spectral density and formant frequencies," *Electron. Commun. Jap.*, vol. 53-A, pp. 36–43, 1970. 45, 53

[73] F. Itakura and S. Saito, "Digital filtering techniques for speech analysis and synthesis," *Conf. Rec. 7th Int Congr Acoustics*, Paper 25C1, 1971. 67

[74] F. Itakura and S. Saito, "On the optimum quantization of feature parameters in the parcor speech synthesizer," *Conference Record, 1972 International Conference on Speech Communication and Processing*, Boston, MA, pp. 434–437, April 1972. 67

[75] I.M. Jacobs, R. Binder, and E.V. Hoversten, "General Purpose Satellite Networks," *Proceedings IEEE*, Vol 66, No. 11, pp. 1448-1467, November 1978. 105

[76] E.T. Jaynes, "Information Theory and Statistical Mechanics," *Physical Review*, Part 1: Vol. 106, No. 4, pp. 620–630, May 1957, Part 2: Vol. 108, No. 2 pp.171–190, October 1957. 40, 56

[77] L.K. Jones, "Approximation-theoretic derivation of logarithmic entropy priciples for inverse problems and unique extension of the maximum entropy method to incorporate prior knowledge," *SIAM J. Appl. Math.*, Vol. 49, No. 2, pp. 650–661, April 1989. 40

[78] R. E. Kahn, "The organization of computer resources into a packet radio network," *IEEE Trans. on communications*, Vol. 25, No. 1, pp. 169–178, January 1977.

[79] R.E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," *Proceedings of the IEEE*, Vol. 66, No. 11, pp. 1468–1496, November 1978. 84

[80] S. Kullback, *Information Theory and Statistics*, Dover New York, 1968. Reprint of 1959 edition published by Wiley. 40, 45, 57

[81] B.M. Leiner, V.S. Cerf, D.D. Clark, R.E. Kahn, L. Kleinrock, D.C. Lynch, J. Postel, L. C. Roberts, and S.S. Wolff, "The past and future history of the Internet," *Communications of the ACM*. Vol. 40, No. 2, 102–108, February 1997. See also `http://www.isoc.org/internet/history/brief.shtml` 4, 129

[82] H. Lev-Ari, S.R. Parker, and T. Kailath, "Multidimensional maximum-entropy covariance extension," *IEEE Transactions on Information Theory*, Vol. 35, No. 3, pp. 497–508, May 1989. 40

[83] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design,"*IEEE Trans. on Comm.*, Vol. COM–28, pp. 84–95, Jan. 1980. 115

[84] Lookabaugh:89 T. Lookabaugh, "Variable rate and adaptive frequency domain vector quantization of speech," PhD Dissertation, Stanford University, Stanford, CA, June 1989.

[85] T. Lookabaugh and R. M. Gray, "High–resolution quantization theory and the Vector Quantization Advantage," *IEEE Transactions on Information Theory*, Vol. 35, pp. 1020–1033, September 1989. 132

[86] D.T. Magill, "Adaptive speech compression for packet communication systems," *Conference Record of the IEEE National Telecommunications Conference*, pp. 29D-1 – 29D-5, 1973. 76

[87] D. T. Magill, E. J. Craighill, and D. W. Ellis "Speech Digitization by LPC Estimation Techniques," *SRI report on ARPA contract DA.H. C04-72-0009 covering the period 3 October 1972 through 31 March 1974*, 1974. 76

[88] J. Makhoul, "Aspects of linear prediction in the spectral analysis of speech," *Proceedings 1972 Conference on Speech Communication and Processing*, pp. 77-80, April 1972. 71

[89] J. Makhoul and J. Wolf, *Linear Prediction and the Spectral Analysis of Speech*, tech. report 2304, BBN, Aug. 1972. 71

[90] J. Makhoul, V. Viswanathan, L. Cosell, and W. Russell, *Natural communication with computers: Speech compression research at BBN*, Report No. 2976, Vol. II, Bolt Beranek and Newman, Cambridge, MA, Dec. 1974. 104, 113

[91] John Makhoul, "Linear prediction: a tutorial review," *Proceedings of the IEEE*, Vol. 63, No. 4, April 1975. 2, 8, 71

[92] J. Makhoul, "Stable and lattice methods for linear prediction," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 25, pp. 423–428, October 1977. 62

[93] J. Makhoul, R. Viswanathan, and W. Russell, "A framework for the objective evaluation of vocoder speech quality," *Proc. ICASSP 1976*, pp. 103–106, April 1976. 106

[94] J. Makhoul, R. Viswanathan, R. Schwartz, and A.W.F. Huggins, "A mixed-source model for speech compression and synthesis," *Journal of the Acoustical Society of America*, vol. 64, pp. 1577–1581, December 1978. 110

[95] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," invited paper, *Proc. IEEE*, vol. 73, no. 11, pp. 1551–1588, Nov. 1985. 132

[96] John Makhoul, "Speech Processing at BBN," *IEEE Annals of the History of Computing*, vol. 28, no. 1, pp. 32–45, January-March 2006. 136

[97] J. Markel, "Formant trajectory estimation from a linear least-squares inverse filter formulation," *SCRL Monograph No. 7*, Oct. 1971. 70

[98] J.D. Markel, "Digital inverse filtering–a new tool for formant trajectory estimation," *IEEE Trans. on Audio and Electro Acoustics*, pp. 129–13?, June 1972. 74

[99] J.D. Markel, A.H. Gray, and H.Wakita , *Linear Prediction of Speech — Theory and Practice*, SCRL Monograph No. 10, Speech Communications Research Laboratory, Santa Barbara, California, 1973. 74

[100] "Documentation for SCRL Linear Prediction Analysis/Synthesis Programs," Speech Communications Research Labs, Inc., Nov. 1973. 74, 88

[101] J.D. Markel and A.H. Gray, Jr., "On autocorrelation equations as applied to speech analysis," *IEEE Trans. on Audio and Electroacoustics*, Vol. 21, pp. 69–79, April 1973. 74, 94

[102] J.D. Markel and A.H. Gray, Jr., "A linear prediction vocoder simulation based upon the autocorrelation method," *IEEE Trans on Acoustics, Speech, and Signal Processing*, Vol. 22, pp. 124–134, April 1974. 74, 94

[103] J.D. Markel, NSC Note 47, Two NSC Group Proposals and a Status Report, SCRL, 11 November 1974. 74

[104] J.D. Markel and A.H. Gray, Jr, *Linear Prediction of Speech*, Springer-Verlag, 1976. 2, 8, 36, 70

[105] M. McCammon, Draft Summary of First CHI–LL LPC Attempt, 11/26/74. 94

[106] M. McCammon, Draft Experiment #2 CHI–LL LPC Voice Contact, 11/26/74. 94

[107] M. McCammon, Report on Third CHI–LL LPC Voice Experiment 26 November 10:00–1:00 AM PST. 94

[108] N. Merhav and Y. Ephraim, "Hidden Markov modeling using a dominant state sequence with application to speech recognition," *Computer, Speech, and Language*, vol. 5, no. 4, pp. 327–339, Oct. 1991. 45

[109] D. Nielson, "The SRI van and early packet speech," *Core 3.1*, a publication of the Computer History Museum, p. 7, February 2002. (Available at `http://www.computerhistory.org/core/backissues/pdf/core_3_1.pdf`.) 84, 105, 112

[110] G. C. O'Leary, "Local Access Facilities for Packet Voice," *Proc. 5th Int. Conf. Computer Communications*, Oct. 1980, pp. 281–286. 123

[111] G. C. O'Leary, P. E. Blankenship, J. Tierney, and J. A. Feldman, "A Modualar Approach to Packet Voice Terminal Design," *AFIPS Conf. Proc. NCC*, Vol. 50, May 1981. 123

[112] J.E. O'Neill, "The role of ARPA in the development of the ARPAnet, 1961–1972," *IEEE Annals of the History of Computing*, Vol. 17, No. 4, 1995. 63

[113] M. S. Pinsker, "Information and information stability of random variables and processes," Holden Day, San Francisco, 1964. 45

[114] E.A. Robinson, *Statistical Communication and Detection*, New York: Hafner Publ. Co., 1967. 70

[115] S. Saito and F. Itakura, "The theoretical consideration of statistically optimum methods for speech spectral density," Report No. 3107, Electrical Communication Laboratory, NTT, Tokyo, December 1966. 52, 54

[116] J. Salus, "Casting the Net: From ARPAnet to Internet and Beyond," Addison-Wesley, Reading, Mass., 1995. 63

[117] M.R. Schroeder and B.S. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," in *Proc. ICASSP'85*, pp. 937–940, May 1985. 132

[118] J. E. Shore and R. W. Johnson, "Properties of cross-entropy minimization," *IEEE Trans. Inform. Theory*, Vol. 27, pp. 472–482, July 1981. 40

[119] L. C. Stewart, *Trellis Data Compression*, Ph.D. Dissertation, Department of Electrical Engineering, Stanford University, June 1981. 119, 120

[120] L.C. Stewart, R.M. Gray, and Y. Linde, "The design of trellis waveform coders,"*IEEE Transactions on Communications*, Vol. COM–30, pp.702–710, April 1982. 119

[121] C.M. Rader, "Some notes on predictive coding of speech," MIT Lincoln Laboratory Technical Memorandum No. 62L-0101, Dec. 1967. 59

[122] P. Spilling and E. Craighill, "Digital Voice Communications in the Packet Radio Network," *International Conference on Communications (ICC '80)*, Seattle, WA, June 8-12, 1980.

[123] C.K. Un and D. Thomas Magill, "Residual excited linear prediction vocoder," *J. Acoust. Soc. Amer.*, Vol. 55, Supplement (A), Spring 1974. 99

[124] C.K. Un and D. Thomas Magill, "The residual-excited linear prediction vocoder with transmission rate below 9.6kbs," *IEEE Trans on Communications*, Vol 23, pp. 1466-1474. 99

[125] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM Journal on Matrix Analysis and Applications*, Vol. 19, 499-533, 1998.

[126] R. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, pp. 309–321, June 1975. 98

[127] R. Viswanathan, J. Makhoul, and R. Wicke, "The application of a functional perceptual model of speech to variable-rate LPC Systems," *Proc. ICASSP 1977*, pp. 219–222, May 1977. 113

[128] V.R. Viswanathan, J. Makhoul, R.M. Schwartz, and A.W.F. Huggins, "Variable frame tate transmission: A review of methodology and application to narrow-band LPC speech coding, *IEEE Transactions on Communications*, vol. 30, pp. 674–686, April 1982. 113

[129] V. Viswanathan, A. Higgins, and W. Russell, "Design of a robust baseband LPC coder for speech transmission over 9.6 kbit/s noisy channels," *IEEE Transactions on Communications*, vol. 30, pp. 663–673, April 1982.

[130] H. Wakita , "Estimation of the vocal tract shape by optimal inverse filtering and acoustic/articulatory conversion methods," *SCRL Monograph No. 9*, SCRL, Santa Barbara, CA 1972. 74

[131] C.J. Weinstein and A.V. Oppenheim, "Predictive Coding in a Homomorphic Vocoder," *IEEE Trans. on Audio and Electroacoustics*, September 1971, Vol. 19, No. 3, pp. 243–248. 60

[132] C.J. Weinstein, "A Linear Prediction Vocoder with Voice Excitation," EASCON Proceedings, Sept.–Oct. 1975. 99

[133] C.J. Weinstein and J.W. Forgie, "Experience with speech communication in packet networks," *IEEE Journal on Selected Areas in Communications*, Vol. 1, No. 6, December 1983. 94

[134] J. Welch, "LONGBRAKE II Final Report," Contract No. DAAB03-74-C-0098, Philco-Ford Corporation, Willow Grove, Pennsylvania, 1974. 70

[135] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series*, The Technology Press and J. Wiley and Sons, New York, 1957. 87

[136] Wiggins, R. and L. Brantingham, "Three-chip system synthesizes human speech," Electronics, Aug 31, 1978, 109-116. Uses TMC 0280, LPC-10, 600-2400 bps. 113

[137] D.Y. Wong, J.-H. Juang, and A.H. Gray, Jr., "An 800 b/s vector quantization LPC vocoder," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 30, pp. 770–780, 1982. 115

[138] D.Y. Wong and J.-H. Juang *Vector/matrix quantization for narrow-bandwidth digital speech compression*, Final Report summarizing contract No. F30602-81-C-0054, Signal Technology, Inc., July 1982. 115