September 1986 LIDS–P–1611
(revised Jan. 1989)

Event–Based Estimation of Interacting Markov Chains with Applications to Electrocardiogram Analysis*

Peter C. Doerschuk**

Robert R. Tenney†

Alan S. Willsky**

Abstract

In this paper we examine the problem of estimating the state of a distributed finite–state Markov process consisting of several interacting finite–state systems each of whose transition probabilities are influenced by the states of the other processes. The observations on which the estimation procedure is based are continuous signals containing signatures indicative of the occurrence of particular events in the various finite–state systems. The problem of electrocardiogram analysis serves both as the primary motivation for this investigation and as the source of a case study we describe in the paper. The principal focus of the paper is on the development of an approach that overcomes the combinatorial explosion of truly optimal estimation algorithms. We accomplish this by constructing a systematic design methodology in which the resulting estimator consists of several interacting estimators, each focusing on a particular subprocess. Important questions that we address concern the way in which these estimators interact and the method each estimator uses to account in its own model for the influence of other subprocesses.

*The research described in this paper was supported in part by the Air Force Office of Scientific Research under Grant AFOSR–82–0258. The first author was supported by fellowships from the Fannie and John Hertz Foundation and the M.D.–Ph.D. Program at Harvard University (funded in part by Public Health Service, National Research Award 2T 32 GM07753–06 from the National Institute of General Medical Science).

**Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 02139.

†Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 02139 and Alphatech, Inc., 2 Burlington Executive Center, 111 Middlesex Tpke., Burlington, MA, 01803.

1. Introduction

In our companion paper [3] we developed a methodology for modeling electrocardiograms (ECG's) that could be used as the basis for ECG signal processing/analysis algorithms. We refer to [3] for the motivation and review of past investigations that lead us to the spatial, temporal, and hierarchical decompositions that are featured in our methodology. Here we will only introduce the implications of these features for signal processing.

Our focus is on cardiac rhythms and therefore the focus of interest in this paper is on the estimation of cardiac events as captured in the evolution of the interacting finite–state processes that for the upper levels of the cardiac models developed in [3]. In Sections 1 and 2 of [3] we provided a discussion of the potential advantages in using these models as the basis for designing signal processing algorithms.

However, while truly optimal estimation based on these models would achieve these advantages, the computational load associated with optimal processing is prohibitively large. Thus the major issue is the development of feasible, suboptimal estimation algorithms. In this paper we investigate the development of such algorithms that take advantage of two important features of this class of estimation problems. First, the estimation of event sequences in the upper level model is essentially a decoding problem (i.e. the ECG is an encoding of the discrete cardiac events we wish to estimate). Consequently we make repeated use of an efficient technique for optimal estimation of finite–state processes first developed for coding applications, namely the Viterbi algorithm [4]. Second, since our models are distributed, we can consider the design of distributed estimators, consisting of interacting algorithms each focused on the job of estimating the state of a particular subprocess. Such estimation structures offer the attractive possibility of implementation in a distributed processor, thereby allowing significant improvements in throughput rates.

The design of such estimators also raises a number of important questions independent of the ECG application. In particular, since the several subprocesses of our upper level model interact strongly, it is not possible to estimate the state of a subprocess without accounting for the influence on it of other subprocesses. Consequently it is necessary to include a (hopefully aggregated) model of other subprocesses that captures the dynamics of the interactions these subprocesses have with the particular subprocess being estimated. Also, it is necessary for the estimators of interacting subprocesses to interact themselves (e.g. estimators of atrial and ventricular activity most certainly have information worth sharing!). The interaction between estimators implies that each estimator needs an aggregated model of the dynamics and the uncertainties in the other estimators in order to interpret the infromation it receives from the other estimators. In addition, since each estimator is using the same raw data but is interested in only some of the events in the data, it may be necessary to provide information to each estimator concerning estimated times of occurrence of other events in the ECG data (e.g. an atrial estimator may need estimates of R–wave locations from the ventricular estimator in order to assist it in locating the much smaller P–waves). Also, as one might expect, there may very well be a need for some iteration in this process so that a high level of performance and consistency among the estimators is achieved.

While electrocardiogram analysis has provided the motivation and examples for our work, there are a variety of other applications in which similar estimation problems arise. In particular, consider interconnected power systems which are made up of strongly interacting components subject to events (such as generator trips and line faults) that can precipitate events in other parts of the system. An extremely important problem is the design of distributed monitoring systems, and a critical aspect of this problem is determining how to structure the interaction among local monitoring systems in order to produce a consistent and accurate overall estimate of system status. Similar issues also arise in military contexts in distributed battle management and assessment. Our analysis begins in the next section with a case study for the ECG application which allows us to introduce the major questions that arise in designing distributed event estimation algorithms. In Section 3 we then extract from the case study a general, systematic design approach for distributed estimation of interacting processes. The paper concludes with Section 4 in which we discuss issues arising in the extension of our results and in particular in the design of a complete ECG rhythm tracking system.

2. An Estimation Example

The process (Figure 1) whose state is to be estimated models normal cardiac rhythm with occasional reentrant–mechanism premature ventricular contractions (PVC's; these result from a normal excitation of the ventricles in effect circling back on itself and causing additional ventricular contractions). Note several important features of the model:

(1) The model consists of two subprocesses, one (the SA–atrial submodel, denoted C0, with state x_0) representing the behavior of the upper chambers of the heart and the other (the AV–ventricular submodel, denoted C1, with state x_1) capturing the behavior of the atrial–ventricular connection and the lower chambers of the heart. The signatures modeled are the P–wave (corresponding to atrial depolarization), the R– and T–waves (corresponding to a normal ventricular depolarization–repolarization cycle) and the V–wave (corresponding to an aberrant reentrant PVC). The signatures are labeled P_i, R_i, T_i, and V_i respectively in the figure. The state transition probabilities (including inter–submodel interactions), the signature means and variances, and the zero–mean observation noise variances are also shown in the figure.

(2) The interactions between the submodels are infrequent but are extremely strong. In particular, the diagram shown for the SA–atrial submodel represents normal activity which occurs unless $x_1 = 13$ (initiation of a PVC) in the AV–ventricular submodel. When such an event occurs, it is possible for the electrical signal to propagate back to the upper chambers of the heart and in essence reset the timing of the heart's own pacemaker. This is captured by modifying the transition probabilities of x_0 so that with probability 1/2 x_0 is reset to state 25 when x_1= 13, and with probability 1/2 x_0 proceeds in a normal fashion. In the x_1 submodel the only transition probability affected by the value of x_0 is p^1_{01}. In particular, $x_1 = 0$ represents the resting state of the ventricles, which is a trapping state ($p^1_{01} = 0$) until the ventricles are excited ($p^1_{01} = 1$ for one time step) by an atrial contraction ($x_0 = 0$).

(3) The ECG measurements are available at a rate four times the clock rate of the x_0, x_1 processes. In order to allow signatures to start at any observation

sample, each signature appears four times with 0, 1, 2, or 3 leading zeros in the mean and covariance sequences. (The subscripts on the wave labels indicate the number of leading zeros).

(4) The initiation of reentrant PVC's is modeled by transitions out of states 12 and 21 in submodel C1. Occupancy of state 12 corresponds to the completion of a normal R, T–wave pair, and from this state there is a probability of 0.9 of returning to the resting state and a probability of 0.1 of entering state 13 corresponding to the initiation of a reentrant PVC. Note that there is a much higher probability (0.4) of initiating subsequent, consecutive reentrant PVC's (the 21–to–13 transition) which results in occasional occurrences of bursts of aberrant PVC's as are seen in episodes of ventricular tachycardia.

(5) The remaining states and transition probabilities model cardiac timing — propagation delays, recovery time following contraction, etc. The model does allow for some uncertainty in this timing behavior and therefore some variability in the heart rate (which with a Markov chain cycle time of 0.04 seconds is, on the average, 75 beats per minute). It is certainly possible to add even more variability, but for simplicity we have not done that here.

Figure 2 shows a plot of several typical segments of a realization of this model. (Recall the discussion of Section 4 in our companion paper [3] concerning the verisimilitude of the simulated ECG, especially the contrast between modeling for physiological accuracy and modeling for signal processing utility). Below the ECG tracing are several sets of annotations. The top row of annotations indicates the true times and types of waves that are present in the data (corresponding to the times at which transitions are made out of state 0 in submodel C0 (P–wave) and states 4 (R–wave), 7 (T–wave), and 13 (V–wave) of submodel C1). The remaining rows represent various annotations constructed during the estimation process, with the bottom row representing our final set of estimates.

A compact pictorial notation for interacting Markov chains is illustrated in Figure 3. Here the label C0 denotes the SA–atrial submodel and C1 the AV–ventriclar submodel shown in Figure 1. The arrows between C0 and C1 indicate that the state of each subprocess influences the transition behavior of the other. Also, the arrows labeled P, R, T, and V indicate the waveforms initiated by each subprocess. In addition, the variables $h_{01}(n)$ denote the sequence of interactions initiated by C0 and impinging on C1. That is $h_{01}(n)$ completely captures the influence C0 has on the transition probabilities of C1 for the transition $x_1(n) \rightarrow$

$x_1(n+1)$. Referring to Figure 1, we see that we can define $h_{01}(n)$ so that it takes on only two values

$$h_{01}(n) = \begin{cases} 0 & \text{if } x_0(n) = 0 \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

The only transition probability of C1 that is influenced by C0 is

$$p_{01}^{1} = \begin{cases} 1, & h_{01}(n) = 0 \\ 0, & h_{01}(n) = 1 \end{cases} \tag{2}$$

Similarly we can define the interactions $h_{10}(n)$ from C1 impinging on C0 as

$$h_{10}(n) = \begin{cases} 0 & \text{if } x_1(n) \neq 13 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

so that if $h_{10}(n) = 0$ the transition probabilities are as indicated in the figure, and if $h_{10}(n) = 1$ they are the average of these values and a probability 1 reset to state 25 from any other state. Note that there are far fewer values for these interaction variables than for the corresponding states. This fact is used in an essential way in constructing several aggregate models used in our estimation methodology.

Our approach to state estimation for such a process involves the design of a set of interacting estimators, each of which focuses on estimation for a particular subprocess. Also, the existence of the interactions among subprocesses may require some iteration. For the present example our estimator can be viewed as consisting of three passes:

(1) Derive a preliminary estimate of ventricular activity (submodel C1).

(2) Based on the observed ECG and the estimates from pass 1, compute an estimate of atrial activity (submodel C0).

(3) Refine the ventricular estimate based on the observed ECG and the estimates of atrial activity from pass 2.

The results from (2) and (3) form the final estimate. This approach parallels the heuristic approach humans take in first identifying high signal–to–noise ratio (SNR) events (R– and V–waves), then using these estimates to assist in locating low SNR events (P–waves), and finally making adjustments to ensure accuracy and consistency. While we describe these three steps as separate passes through the data, it is straightforward to construct a pipelined structure in which the three steps proceed at the same time.

We now turn to a detailed examination of each of these three passes. Because the first pass focuses on submodel C1, it is natural to include an exact copy of this submodel in the estimator's model. However, it is also necessary to model the interactions impinging on the C1 submodel, i.e. $h_{01}(n)$. Possibilities range from the exact model of C0 depicted in Figure 1 to no model. We use the simplest possible aggregate model for submodel C0 with which we can still capture the full range of interactions with submodel C1, specifically we use a two–state model, corresponding to the two possible values of $h_{01}(n)$. In addition, we allow submodel C1 to reset the state of our two–state aggregate model, again reflecting behavior seen in the full model. In the full discussion of our approach to estimation, this type of aggregate model is referred to as an "S0–submodel". Details for this example are given in Figure 4.

There are several further points to make about this first pass. First, because the P–wave has a small amplitude in comparision to the R– and V–waves, which are the waves of primary concern for this pass, it is unlikely to be confused with an R– or V–wave. Therefore, though it is straightforward to define a S0 subprocess that initiates P–waves, we have not done so. Second, one can imagine several methods for choosing p in submodel S0 — matching some statistic of the exact submodel C0 or viewing p as a design parameter to be chosen to optimize estimator performance. In [2] several general statistical methods (which can be easily automated) are described for choosing parameters to match particularly useful statistics. In Section 3 we describe the statistical method used to obtain the value for p indicated in the figure. Finally, with this parameter specified, we have a complete model, and the first step estimator is designed to produced a minimum probability–of–error state trajectory estimate for this model (i.e. estimates of the states of S0 and C1 as functions of time) based on the observed ECG. This computation and those in all of our estimators are performed using the

Viterbi algorithm [4] which efficiently and recursively computes the optimal smoothed state trajectory, i.e. the best state estimate at each time is based on information before and after that time.

The Viterbi algorithm requires the process to be Markovian, while signatures (as in this example) that last more than one Markov chain cycle make the process nonMarkovian. However, it is straightforward to Markovianize the process by state augmentation and because there are few transitions that initiate signatures, the required augmentation does not radically increase the size of the state space. Though straighforward, the details of this augmentation process are rather tedious and are omitted.

The results of this first pass estimator are illustrated in the second row of annotations in Figure 2, where we have indicated the estimated times of occurrence of R–, T–, and V–waves. For the most part these estimates are quite accurate, thanks to the high SNR of these waves, although there are infrequent false alarms in the estimates caused by extra–long P–P intervals in which case the estimator attempted to match a T–wave with an actual P–wave.

The second step in our overall estimation structure is to estimate the state in the SA–atrial submodel. Therefore, it is natural to include an exact copy of the SA–atrial submodel in the estimator's model. The only direct information from the ECG for this step is the low SNR P–wave. However, there is also a great deal of indirect information available through the causal relationship between P– and R– waves and V– and P–waves.

First consider interactions initiated by C0. That is, consider the causality between P– and R–waves the latter of which only occur when the SA–atrial submodel successfully excites the AV–ventricular submodel. The goal is to exploit the auxiliary information concerning R–wave occurrences determined in the first estimation pass. At the very least one could imagine using the state estimates for S0 from the first pass which are estimates of interactions impinging on the AV–ventricular submodel. Since the 0–state in this submodel corresponds to the 0–state in the original submodel C0 (and thus to attempts to excite submodel C1), the estimates of times at which S0 is in state 0 would be likely estimates of times at which estimates of times at which S0 is in state 0 would be likely estimates of times at which $h_{01}(n) = 0$. However, because of the highly aggregated nature of S0, some of these estimates may be somewhat suspect. However, when such an estimate is coupled together with a closely following estimated occurrence of an R–wave (corresponding to the estimate of the C1 subprocess occupying state 4), the S0 estimate is much more likely to correspond to a true occurrence of an attempt at ventricular excitation. Consequently the information we provide to pass 2 from pass 1, which we will refer to as estimated augmented interactions, consists of the sequence of estimates of the states of S0 and C1 produced in pass 1.

In order to use the estimated augmented interactions we must model the errors they contain. Note, however, that the errors of importance here are not only memoryless errors (which could be modeled by static misclassification probabilities) but also errors in timing (e.g. the estimated time of occurrence of an R–wave may be in error by one or two samples). Consequently, we need a dynamic model for the way in which estimated augmented interactions provide information about C0. This is accomplished, as illustrated in Figure 5, by modeling the estimated interactions, denoted by z_1, as the observed outputs of an additional submodel of a class we refer to as S1 submodels. This additional submodel receives interactions from C0, whose state we wish to estimate. In order to model the fact that the estimates in $z_1(n)$ may contain time shifts relative to the actual values of the interactions $h_{01}(n)$, we take as the state of the S1 submodel a vector of the most recent interaction values. To minimize the size of the S1 state space one clearly wishes to minimize the dimension of this vector. For this study we found a dimension of 2 to be adequate, so that the state of S1 at time n is $(h_{01}(n-1), h_{01}(n-2))$. By examining C0, we see that it is impossible for h_{01} to equal 0 at consecutive times. Thus, there are only three possible S1 states which we have coded as follows in Figure 5:

$$0 = (0, 1) \ , \ 1 = (1, 0) \ , \ 2 = (1, 1)$$

Since h_{01} is a deterministic function of the state of C0 it is straightforward to derive the way in which $x_0(n)$ affects the transition behavior of S1 (Figure 5).

As in all of our models, the observation $z_1(n)$ is associated with transitions in the S1 subprocess which correspond to 3–tuples $(h_{01}(n-1), h_{01}(n-2), h_{01}(n-3))$ of interactions. Our measurement model is then the set of conditional probabilities

$$\Pr(z_1(n) \mid h_{01}(n-1), h_{01}(n-2), h_{01}(n-3)) \qquad (4)$$

Since the Viterbi algorithm provides us with noncausal estimates, we are free to build some noncausality into this model. Consequently, we have chosen to take $z_1(n)$ as the pass 1 estimate at time n–2, which therefore provides an estimate of $h_{01}(n-2)$. Thus the model allows us to capture time shifts of ±1. The specification of (4) can be obtained by analysis of the performance of the first step estimator. We have estimated these quantities via simulation.

We now must consider the interactions $h_{10}(n)$ initiated by C1 and impinging on C0, i.e. the effect of V–wave occurrences on C0. There is a similarity here with the modeling of S0 in the first pass but in the present context we also have the estimates from pass 1 which tell us something about these interactions. Specifically, since we used the exact C1 submodel in pass 1, we can deduce estimates of h_{10} (see (3)). We take these estimates as our observation z_2 for pass 2 (without any augmentation as was done for z_1 since the first step estimator used an exact model for C1 and consequently should produce comparatively accurate estimates). Also, as with the S1 submodel, we need to model possible estimation timing errors, so again we take the state of S2 to be a set of the most recent interactions, in this case $(h_{10}(n), h_{10}(n-1))$.[1] In this example it is impossible for h_{10} to equal 1 at two consecutive times, and thus we can code the feasible S2 states as

$$0 = (0, 0) \quad , 1 = (0, 1) \quad , 2 = (1, 0)$$

[1] Note that there is some asymmetry in comparison with the S1 submodel where the state was lagged one step. This is a result of the fact that in the S1 submodel, $h_{01}(n)$ is a deterministic function of $x_0(n)$. Thus for the state $x_0(n)$ to correctly "influence" the next transition in S1, we needed to introduce the time delay in defining the S1 state. This is not needed in S2, since there is no such deterministic coupling.

In this example the C0 submodel transition probabilities are shown in Figure 1 for $x_{S2}(n) = 0$ or 1 and incorporate the 0.5 probability reset to state 25 when $x_{S2}(n) = 2$. The S2 model is illustrated in Figure 5. Note that as with S1, there is a parameter p to be chosen to specify the S2 transition probabilities. This parameter was also chosen to match statistics of the true h_{10} process using a general method described in the next section. Finally, the observation $z_2(n)$, which is the pass 1 estimate of $h_{10}(n-1)$, is modeled as resulting from S2–transitions. Thus again we must specify a distribution, namely

$$\Pr(z_2(n) \,|\, h_{10}(n), h_{10}(n-1), h_{10}(n-2))$$

which we have again done by simulation.

This completes the specification of the second pass model. Note the complete absence of R–, T–, and V–waves. For the pass 1 estimation algorithm we argued that it was reasonable to consider omitting P–waves from the model since (a) we were focusing most attention on submodel C1 and (b) the P–waves were of low amplitude. In pass 2, the first argument holds (here we are focusing on C0), but the latter does not. In the general procedure described in the next section, we allow for the possibility of taking such waves into account through so–called subtractor submodels. However, as the results in this section and in [2] indicate, for ECG–type models such as the one considered here, that is unnecessary. Intuitively such waves can be ignored in the pass 2 estimation algorithm because through $z_1(n)$ and $z_2(n)$ we are providing indications of the times at which these waves occur. Given then the coupling between these waves and the likely times of P–waves, captured in the original C0–C1 model and in our simplified pass 2 C0–S1–S2 model, the pass 2 estimator will not try to account for R–, T–, and V–waves by placing P–waves in their locations.

A second issue we have ignored is that of allowing the C0 submodel to influence the S2 submodel motivated by the fact that the C0 submodel does influence the C1 submodel. However, it is precisely this influence that is focused upon in the S1 submodel, while the S2 submodel focuses on that part of the C1 submodel, dealing with V–waves, which is unaffected by the C0 submodel. Consequently, while our general modeling methodology allows C0 to influence S2, it is not necessary to include this bit of complexity in the present context.

Note that in our model we consider z_1 and z_2 to be independent measurements, which is clearly erroneous since they are both determined by the pass 1 estimation process. One can certainly construct a more complex model involving a joint distribution of z_1, z_2 given the combined information in the most recent transitions of S1 and S2, but this was not found to be necessary (since again z_1 and z_2 focus on different portions of the overall model).

In summary, the second pass of our procedure consists of the minimum probability–of–error estimation of the state trajectory of the model given in Figure 5 given the ECG measurement and the derived measurements z_1 and z_2 from the first pass. The results for this example are given in the third row of annotations in Figure 2 showing the times at which P–waves were estimated to have occurred. Comparing this to the top row of annotations we see that performance is quite good. Note that the erroneous R,T–wave pairs from pass 1 near 136.6 and 138.3 seconds did not lead to an erroneous P–waves in pass 2, thanks to our modeling of z_1 which incorporated the possibility of such false alarms. Note also the occurrence of P–wave timing errors (as illustrated near 80.2 and 99.9 seconds) all of which underestimate the P–R interval. Finally, note that it is possible in our model (and in the heart) for P– and V–waves to occur nearly simultaneously or for V–waves to preempt an already occurring P–wave from initiating a normal R–wave. Having knowledge of this, the pass 2 estimator will attempt to insert P–waves when the timing seems likely even though the presence of V–waves may obscure the P–wave. An example of correct estimates of this type can be found near 99 seconds. A false alarm can be seen near 82.6 seconds, and a missed detection near 83.3 seconds. While the value of such estimates is suspect (and not of particular consequence) they do provide rather graphic examples of the use our estimator makes of the timing and control information embedded in our models.

The third pass of the estimation process, whose purpose is to provide improved and consistent estimates of ventricular activity, is based on a model, illustrated in Figure 6, with structure analogous to that of pass 2 (with the roles of submodels C0 and C1 interchanged). We omit the details of the construction, as they are exactly analogous to those in pass 2. The estimator is again a minimum probability–of–error estimator using the ECG and the derived measurements z_1, z_2.

The result of applying this estimator is illustrated in the fourth row of annotations in Figure 2. The final, overall estimate (row 5) consists of the C0–state estimate of pass 2 (row 3) and the C1–state estimate of pass 3 (row 4). Comparing the top and bottom rows we see that the estimator has performed quite well. Disregarding the initial heartbeat (which was missed in pass 3 because of the specific way in which we implemented the initialization of the latter passes of our algorithm) all R–, T–, and V–waves were detected and located with no false alarms. Note that while there had been several false R, T–wave estimates in pass 1, these have been completely eliminated in pass 3, in which we have the benefit of using estimates of C0–behavior in order to enforce consistent overall estimation.

The estimation of P–wave occurrences is also quite good. Quantifying this performance, however, is an interesting question itself, since one is clearly not just interested in estimation errors at points in time but also in timing errors at points in the estimated event cycle — i.e. an estimation error of one time sample in locating a P–wave should not be thought of as a missed detection but rather as a timing error. Much more on the issue of performance measures for event–oriented estimation problems can be found in [2]. This example does, however, indicate the main ideas. In examining the results of the full simulation we find that there are only two isolated false positive P–wave indications and one isolated false negative (neglecting the initial heartbeat), where by "isolated" we mean that there is no nearby P–wave in the true or estimated state trajectories. Given that there are 230 heartbeats in this simulation, these correspond to a false positive rate of .009 and a false negative rate of .004. There are also 23 other paired false positives and negatives, where we have used the criterion of associating estimated and actual P–wave locations only if the waveforms at these locations overlap. This corresponds to a paired error rate of 0.10. Note that in our model, every R–wave must be preceded by a P–wave, and thus this pairing is to be expected. It is worth noting that in each of these paired errors, the estimated P–wave location was closer to the R–wave than the true R–wave, indicating a bias that may be removable (and is most likely due to the pass 2 estimator correlating the P–wave with the initial portion of the R–wave).

In [2] we consider a variety of other models. For example, we have examined models with transient AV block, i.e. models in which not every attempt at ventricular excitation leads to an R–wave even if the ventricles are apparently in the resting state. Because of the additional freedom in the model, one would expect some drop in performance. However the drop is extremely small for estimators based on the principles outlined in this section and formalized in the next.

3. A General Design Methodology

The example of Section 2 illustrates the major elements of a general estimator design methodology for distributed Markov chains which is described in this section. Specifically, consider the estimation of an interconnection of subprocesses, denoted C_0, C_1,...,C_N, with states x_0, x_1,...,x_N, given measurements of signals containing signatures corresponding to particular state transitions in these subprocesses. Let $h_{ij}(n)$ denote the interaction initiated by C_i and impinging on C_j at time n. This interaction is a deterministic function of $x_i(n)$, and the transition probabilities of C_j are deterministic functions of $\{h_{ij}(n)\,|\,i\neq j\}$. The assumption is that the set of possible transition probabilities for each C_j (and thus the set of possible values of $\{h_{ij}(n)\,|\,i\neq j\}$) is quite small.

Our overall estimator consists of an interconnection of local estimators (LE's), each of which focuses on the estimation of one of the subprocesses. Because of the existence of interactions with and events in the observed data due to other subprocesses, each LE not only must take these effects into account in its model but also must communicate with the other LE's.

During the initial pass through the data the LE's have no previous information to communicate and the LE for a specific submodel C_j will in general need:

(a) A complete model of the subprocess C_j on which it is focused.

(b) A model of the interactions impinging on C_j.

(c) A model of the waveforms generated by the other submodels.

The model referred to in (b) is called an S0 submodel, and a major objective is to make it as simple as possible in order to keep the LE as simple as possible.[2] We have taken the states of the S0 submodel to be in one–to–one correspondence with the possible values of the N–tuple $\{h_{ij}(n) | i=j\}$. In order to set the transition probabilities for the S0 submodel our primary approach has been to match these one–step transition probabilities to the actual steady–state versions within the original process, that is, to

$$\lim_{n\to\infty} \Pr(\{h_{ij}(n) | i{\neq}j\} \,|\, \{h_{ij}(n-1) | i{\neq}j\}, \{h_{ji}(n-1) = h_{ji} | i{\neq}j\}) \tag{5}$$

Unlike $\{x_i(n) | i{\neq}j\}$ conditioned on $\{h_{ji}(n) | i{\neq}j\}$, the highly aggregated $\{h_{ij}(n) | i{\neq}j\}$ conditioned on $\{h_{ji}(n) | i{\neq}j)$ is typically not a Markov chain and therefore the limit in (5) is not a trivial computation, though it is straightforward once the ergodic probabilities for $\{x_i(n) | i{\neq}j\}$ have been computed. Typically for models with infrequent changes in interactions, most of the transition probabilities specified in (5) are 0 or 1, and there are only a few parameters (such as p in Figure 4) for which this computation is necessary.[3]

[2] There are two distinct ways in which one can perform this modeling step and several that follow. In particular, in this section we describe the construction of a single S0 submodel capturing the interactions impinging on C_j from *all* other subprocesses. In [2] an analogous approach is described for constructing separate S0 submodels for the interactions initiated by *each* of the other subprocesses.

[3] Indeed for *all* of the cases considered in [2], the model was exactly as in Figure 4 (with different values of p), since in all of our cases there have been only two interaction values, one of which could not occur at consecutive times.

Note that we have included conditioning on $\{h_{ji}(n-1) | i{\neq}j\}$, which reflects the influence C_j has on the other subprocesses. This results in the transition probabilities of S0 being influenced by the state of C_j. Again we typically expect this influence to manifest itself as a small number of possible values for a small subset of the transition probabilies (e.g. in our case study only the parameter p in Figure 4 is influenced, and it only takes on two values).

Finally, note that there are cases in which the matching of the steady–state statistic (5) may be inappropriate since it, in essence, assumes that the transition probabilities of $\{x_i(n) | i{\neq}j\}$ do not change very frequently (so that steady–state is actually achieved) That is, (5) assumes that the interactions $h_{ji}(n)$ are constant so that the time variations observed in the actual $x_j(n)$ process must not lead to frequent changes in the interactions $h_{ji}(n)$. We refer the reader to [2] for examples violating this assumption and in which we must set the S0 transition probabilities in a different manner. Note that this assumption is in fact violated in our case study. In particular, while it is certainly true that h_{10}=0 for long periods of time, h_{10}=1 cannot possibly occur at any <u>two</u> consecutive times. In this case, since h_{10} = 1 corresponds to a reset of C0 to state 25, and since all states in C0 other than 0 correspond to h_{01} = 1, it is reasonable to reset the state of S0 to 1 whenever x_1 = 13. This is what is specified in Figure 1 <u>and</u> what we would calculate from (5). Thus (5) is often useful even if the assumption on which it is based is violated.

The model referred to in (c), denoted S3, is one of the <u>subtractor submodels</u>, referred to in the previous section. It is incorporated in order to keep the LE from interpreting waveforms generated by other submodels as coming from C_j. Our desire is to present the LE with observations containing only those signatures generated by C_j. Since this is not possible, we equip the LE with a mechanism for estimating when other signatures have occured so that it can in effect subtract out their effects. In general, one can construct a separate S3 submodel for each signature not initiated by C_j. While it is possible to couple these subprocesses with the C_j and S0 submodels, we have obtained good results with the simpler structure shown in Figure 7, in which each S3 submodel is a completely autonomous, aggregated process that produces interarrival statistics for the wave of interest identical to those produced by the exact model. Let $\tau_{SS}(n)$ denote the time between the n^{th} and $(n+1)^{st}$ occurrence of the signature S in the original process. Then we choose the two parameters p and q to match the probability that signatures occur at successive times and the mean time between successive signatures. That is

$$p = 1 - \lim_{n \to \infty} \Pr[\tau_{SS}(n) = 1] \tag{6}$$

$$\frac{p}{q} + 1 = \lim_{n \to \infty} E[\tau_{SS}(n)] \tag{7}$$

Again the statistics in (6),(7) can be calculated from the ergodic probabilities of the full model. In most cases $\Pr[\tau_{SS}(n) = 1] = 0$, so that

$$q = \frac{1}{\lim_{n \to \infty} E[\tau_{SS}(n)] - 1} \tag{8}$$

Therefore, in our general methodology we construct each initial LE model using C_j, S0, and S3 components as illustrated in Figure 8 and compute the initial pass minimum probability–of–error estimates for each LE. We are then in a position to consider a refinement pass, in which each LE reprocesses the data, together with information provided from the initial passes of the LE's.

The LE for subprocess C_j will in general need the following elements in its model for a refinement pass:

(1) A complete model of C_j.

(2) A model of the information provided by the previous pass concerning interactions initiated by C_j.

(3) A model of the information provided by the previous pass concerning interactions impinging on C_j.

(4) A model of the information provided by the previous pass concerning times of occurrence of waveforms generated by the other subprocesses.

Elements (2) and (3) together correspond to (b) in the initial pass. They are split here because (i) it simplifies modeling the information and (ii) the information referred to in (2) and (3) typically comes from different sources or is of very different accuracy or structure since each LE has an accurate model of its own subprocess but only highly aggregated models of the others.

As discussed in the previous section, the models referred to in (2) and (3), denoted S1 and S2 submodels, respectively, must capture the timing and estimation uncertainties from the previous pass. Each accomplishes this by taking as its state space a moving window of the most recent interactions. In particular, the state of the S1 submodel consists of a window of the most recent values of the N–tuple $\{h_{ji} | i \neq j\}$ while the state of the S2 submodel is a window of the most recent values of the N–tuple[4] $\{h_{ij} | i \neq j\}$. An objective in designing these models is to keep the window lengths K_1 and K_2 small in order to minimize state space size. This desire is balanced by the need to model estimation timing errors (since the maximum such symmetric error that can be modeled corresponds to 1/2 the window length). In our work we have always taken this window length equal to 2.

The S1 dynamics are essentially a shift register memory, since each $h_{ji}(n)$ is a deterministic function of $x_j(n)$ and since the full C_j model is used by the LE. Specifically, given $x_j(n)$, the transition

$$\begin{gathered} x_{S1}(n) = \{h_{ji}(m) | i \neq j,\ m=n-K_2,\ldots,n-1\} \\ \downarrow \\ x_{S1}(n+1) = \{h_{ji}(m) | i \neq j,\ m=n-K_2+1,\ldots,n\} \end{gathered}$$

is deterministic, that is, for each present state there is one next state (whose identity depends on $x_j(n)$) that S1 will occupy with probability 1.

[4] Recall from the previous section that there is some asymmetry in the windows here, with the window for S1 stopping at time n–1, and the window for S2 stopping at time n.

The dynamics of the S2 submodel are not deterministic. As in the S0 submodel, we choose the S2 transition probabilities to match those in the original process. In particular, we choose these to equal

$$\lim_{n\to\infty} \Pr(\{h_{ij}(m)\,|\,i\neq j,\ m=n-K_2+2,\ldots,n+1\}\,|\,\{h_{ij}(m)\,|\,i\neq j, m=n-K_2+1,..,n\},\{h_{ji}(n)\,|\,i\neq j\}) \tag{9}$$

By including the conditioning on $\{h_{ji}(n)\,|\,i\neq j\}$ we can capture the interactions initiated by C_j and impinging on the other subprocesses (and therefore, in the LE model, on S2). However, as discussed in the previous section, the effects of these interactions are the primary concern of the S1 submodel, and thus it is worth seeking and typically possible to find a far simpler model. In fact throughout our work we have been able to completely eliminate the influence of C_j on S2 (which then operates autonomously, generating the interactions that impinge on C_j). This can be done by using (9) with $\{h_{ji}(n)\,|\,i\neq j\}$ set equal to the values that represent the most usual interaction[5] or by computing the average of (9) over the possible values of $\{h_{ji}(n)\,|\,i\neq j\}$ using their ergodic probabilities. We have used the latter of these two methods.

Consider next the modeling of the "measurements" provided by the previous data pass. With respect to S1, we have in general the following sources of information concerning the interactions initiated by C_j:

(i) The previous state estimate of C_j from its associated LE. From this we can directly compute an estimate of $\{h_{ji}(n)\,|\,i\neq j\}$.

(ii) The augmented interactions from each of the other LE's. These consist of the estimate of the interaction impinging on the C_i submodel associated with each LE (obtained from the aggregated S0 submodel used by the LE) and the corresponding C_i–state estimate.

Together this information forms a measurement, which we denote $z_1(n)$, and we model the information contained in $z_1(n)$ by

[5] In our ECG examples this corresponds to no attempt at interprocess excitation, as such electrical excitations occur over relatively short time periods (usually a single time sample).

$$\Pr(z_1(n)\,|\,\{h_{ji}(m)\,|\,i\neq j,\ m=n-K_2-1,\ldots,n-1\}) \qquad (10)$$

As discussed in the previous section, we have the flexibility of introducing some noncausality in order to model positive and negative timing errors. That is, we take $z_1(n)$ to be the previous pass estimates indicated in (i) and (ii) evaluated at time $n-1-K_2/2$. Finally, while it is possible to devise analytical methods to obtain approximations for (10), we have found it easier to evaluate these distributions by simulation.

For S2, we have the following sources of information concerning interactions impinging on C_j:

(i) The augmented estimated interaction provided by the previous pass of the LE for C_j.

(ii) The estimated state of each C_i provided by the associated LE. From these we can directly compute estimates of each $h_{ij}(n)$.

This information forms the measurement $z_2(n)$, which is modeled via

$$\Pr(z_2(n)\,|\,\{h_{ij}(n)\,|\,i\neq j,\ i=n-K_2,\ldots,n\}) \qquad (11)$$

Again we introduce some noncausality by taking $z_2(n)$ to be the previous pass information evaluated at $n-K_2/2$, and we determine (11) by simulation.

Finally, consider modeling the information available from the previous pass concerning waveforms generated by other submodels. Each such waveform is modeled by a second type of subtractor model denoted S4 which is similar in structure and principle to S2 submodels. Consider an S4 submodel corresponding to a particular waveform generated by submodel C_i. The measurement $z_4(n)$ provided by the previous pass LE for C_i is a sequence of binary annotations — 0 if the LE estimates that the particular C_i waveform was not initiated at that time sample and 1 if the estimate is that the waveform was generated. The state of the S4 submodel is a window of the most recent true values of these binary annotations. As with S2, the transition rates of this model are chosen to match the corresponding transition rates of sequences of binary annotations in the full model. If the counterpart to (9) is used, the S4 model will in general be influenced by C_j. Again, as in the case of S2, we have typically simplified this model so that S4 is autonomous, by averaging out the C_j–dependence using the ergodic distribution for $x_j(n)$.

The output of the S4 chain is a sequence of occurrences of the waveform being modeled. Such outputs occur at all S4 transitions to states with a 1 as the most recent annotation. The auxiliary observation $z_4(n)$ is again modeled via a probability distribution conditioned on the most recent S4 transition. We have determined distributions of this type via simulation.

The structure of the models on which each LE refinement pass is based is depicted in Figure 9. In principle one can envision making several refinement passes, with the final estimate consisting of the collection of C_j–state estimates from the final passes of the corresponding LE's. The primary purpose of the refinement passes is to improve the accuracy and consistency of this set of estimates. In particular, if one implemented a single, optimal estimator for the full process, one would know for certain that all transitions present in the final state estimate would be consistent (i.e. have nonzero probability in the full process). When one uses a collection of distributed, simpler LE's, there is no such guarantee, but the coordination made possbile by refinement passes makes the occurrence of inconsistent estimates extremely unlikely.

In the example of Section 2, the first refinement pass (pass 2) is crucial because it is the first pass to focus on sumodel C0. The second refinement pass (pass 3) is less important, though it does correct several false positive errors made by pass 1.

The complete procedure we have described requires the implementation of a full set of LE's for the initial pass (based on models as in Figure 8) and subsequent refinement passes (each based on its own model as in Figure 9). As in our example in Section 2, it is typically possible to simplify this design considerably. First of all, for each LE it is often not necessary in the initial pass to include subtractor submodels S3 for waveforms of low SNR compared to the waveforms generated by the submodel corresponding to the LE. Also, as we showed, it may not be necessary to include any S4 submodels, since the information provided through S1 and S2 submodels essentially provides timing information that allows the LE to avoid intervals in which these interfering signatures may appear. In [2] we present comparative results with and without S3 and S4 submodels that support these simplifications.

It is also typically possible to eliminate many of the LE's from each pass. For example, in the initial pass, one typically would implement LE's only for submodels generating the higher SNR signatures (such as R–waves), since the performance of initial pass LE's for other submodels with only low SNR signatures (or no signatures, as is the case for some rhythm models described in [2],[3]) will generally be unsatisfactory. Also, in order to achieve consistency, we do not need to refine all LE's in subsequent passes. In particular we typically can implement an alternating iterative structure much as in the example in which we initially estimate the C_j with high SNR signatures, then use these estimates to assist in estimating only the remaining C_i during the next pass; these estimates can then be used in turn during the following pass in the re–estimation of the C_j from the initial pass in order to improve the accuracy and consistency of the C_j–state estimates. Note that in addition to eliminating entire passes of LE's, such a structure reduces the quantity of z_1 and z_2 measurements to be processed by the remaining LE's. In fact, the full set of such information described previously has some redundancy, reflecting the fact that perhaps not all of this intermediate processing is needed. The structure described above simplifies the design by removing these redundant sources of information. In [2] we present results favorably comparing reduced designs of this type to estimators incorporating more or all of the LE's at each stage.

4. Conclusions

In this paper we have presented a methodology for the distributed estimation of interconnected finite–state processes given the observation of signals containing waveforms initiated by events in the various processes. The motivation for our work is the problem of automated ECG analysis, but the methods we have developed are of potential use in a variety of other applications such as the monitoring of distributed power networks.

The approach we have developed highlights the major issues that must be addressed in designing distributed estimators, namely the aggregated modeling of the interactions between other portions of the overall process and the particular subprocess being estimated and the dynamic modeling of the information provided by other estimators as part of the process of producing coordinated, consistent estimates of all the subprocesses. We have presented systematic procedures for constructing these models that can in fact be used as the basis for a completely automated estimator design procedure [2].

In order to illustrate the various elements of our design process we have presented a case study corresponding to the tracking of a particular cardiac rhythm using synthetic data. The results presented indicate the potential of this design method. Two major issues remain to be considered, however, before a complete ECG rhythm analysis system can be constructed. In particular, while our distributed design yields estimators with far more modest computational demands than the corresponding optimal estimator, several steps can be taken to simplify these computations even more. First, as mentioned previously, it is possible to construct pipelined versions of our multi–pass estimators in which all passes are performed at the same time rather than in sequence. This achieves a several–fold increase in processing throughput. Also, the nature of the models arising in ECG analysis offer another possibility for simplification. Specifically, these finite–state processes typically display multiple time scale behavior (as actual signature–initiating events occur at a far lower rate than the sampling rate needed to capture interprocess timing). Consequently, it may be possible to use results on hierarchical aggregation of processes with several time scales [1] to construct more efficient estimators that not only display the spatial but also the temporal decomposition of these processes.

Finally, it is important to realize that the problem of rhythm tracking addressed here is only a first step in a rhythm diagnosis system. Specifically in such a system one wishes to identify the underlying distributed process model from a set of such models representing different cardiac rhythms. As in standard system identification problems, the computation of the likelihoods for a set of models can be performed efficiently using the estimates produced by estimators based on each of the models (e.g. see [5] for an application of this idea to ECG rhythm analysis based on R–wave location data only). In [2] we describe an approach to constructing such likelihoods based on the outputs of a set of estimators of the type described in this paper, but work remains to be performed to test this method and to develop efficient implementations.

Acknowledgement

We are grateful to Prof. R.G. Mark for the opportunity to use the M.I.T. Biomedical Engineering Center's computational facility.

References

[1] M. Coderch, A.S. Willsky, S.S. Sastry, and D.A. Castanon, "Hierarchical Aggregation of Singularly Perturbed Finite State Markov Processes," Stochastics, Vol. 8, pp. 259–289, 1983.

[2] P.C. Doerschuk, "A Markov Chain Approach to Electrocardiogram Modeling and Analysis," Ph.D. thesis, M.I.T. Dept. of Elec. Eng. and Comp. Sci.; also M.I.T. Lab. for Inf. and Dec. Sys. Rept. LIDS–TH–1452, April 1985.

[3] P.C. Doerschuk, R.R. Tenney, and A.S. Willsky, "Modeling Electrocardiograms Using Interacting Markov Chains," this Journal.

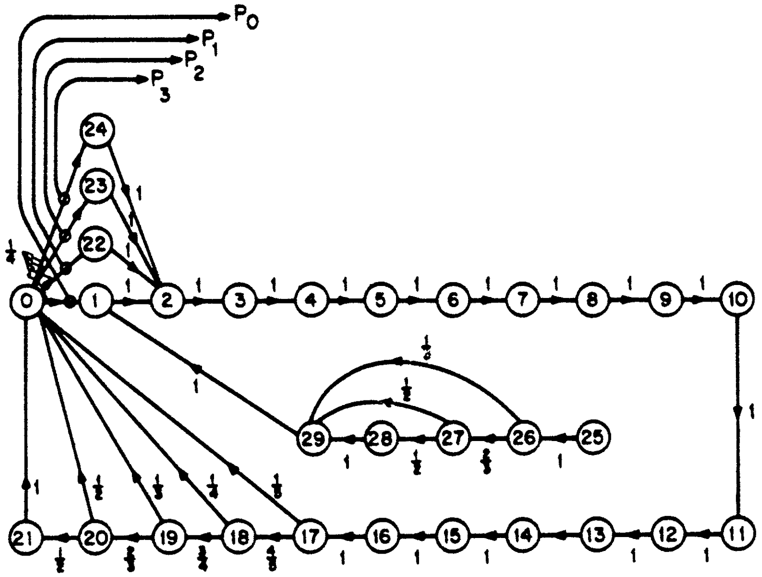
[4] G.D. Forney Jr., "The Viterbi Algorithm", Proc. IEEE, Vol. 61, No. 3, pp. 268–278, March, 1973.

[5] D.E. Gustafson, A.S. Willsky, J.–Y. Wang, M.C. Lancaster, and J.H. Triebwasser, "ECG/VCG Rhythm Diagnosis Using Statistical Signal Analysis, Part I: Identification of Persistent Rhythms, Part II: Identification of Transient Rhythms", IEEE Trans. on Biomed. Eng., Vol. BME–25, No. 4, pp. 344–361, July, 1978.

Table of Figures

SA-Atria: state x_0

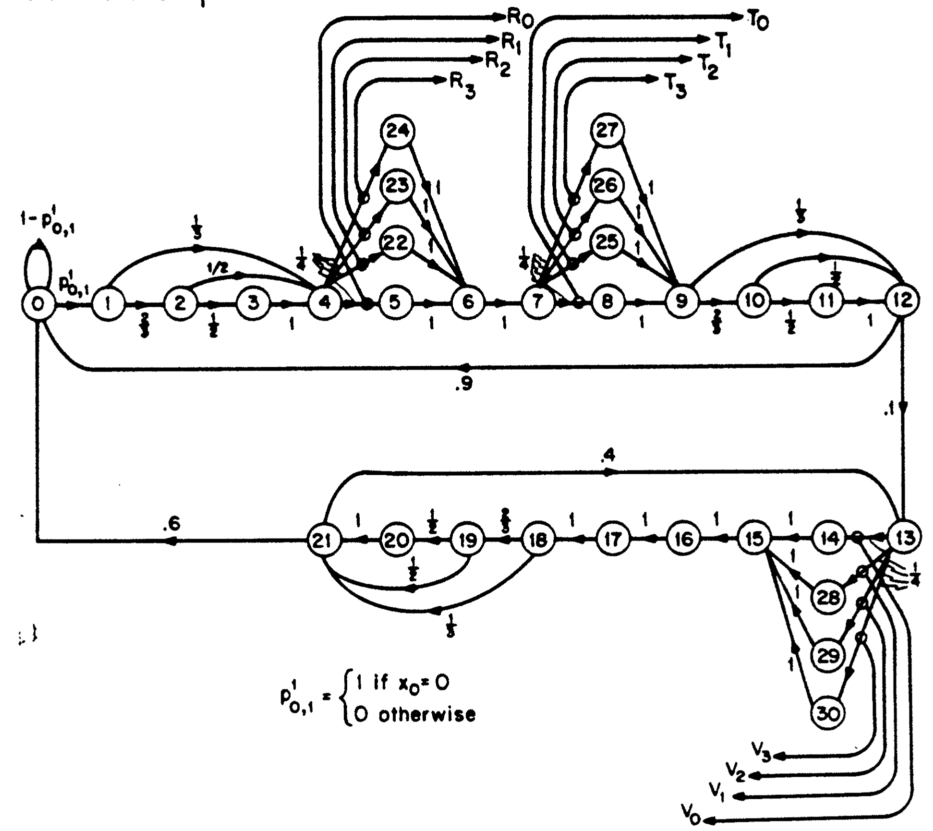
if $x_1 \neq 13$:



Call this tpm P^0

if $x_1 = 13$: then the tpm is $\frac{1}{2}(P^0 + Q^0)$ where $Q^0 = \{q^0_{i,j}\}$ and $q^0_{i,j} = \begin{cases} 1 & j = 25 \\ 0 & \text{otherwise} \end{cases}$

AV-Ventricles: state x_1



$p^1_{0,1} = \begin{cases} 1 & \text{if } x_0 = 0 \\ 0 & \text{otherwise} \end{cases}$

(a)

Fig 1(a)

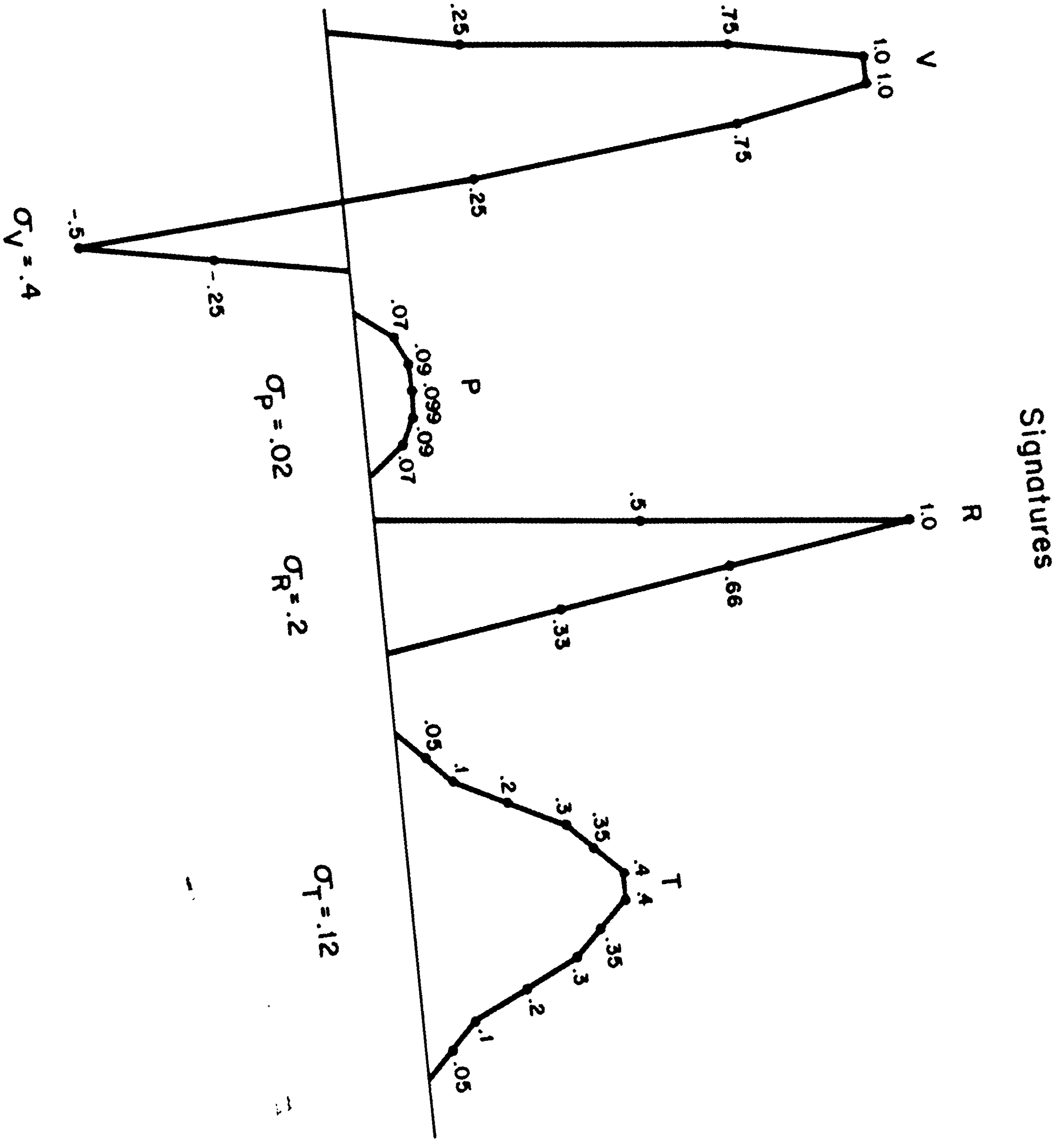

Signatures
V
.25
.75
1.0 1.0
.75
.25
-.5
-.25
$\sigma_V = .4$
P
.07
.09
.099
.09
.07
$\sigma_P = .02$
R
.5
1.0
.66
.33
$\sigma_R = .2$
T
.05
.1
.2
.3
.35
.4
.4
.35
.3
.2
.1
.05
$\sigma_T = .12$

Fig 1(b)

reentrant in mcmod5, ecg truth LE1P0 LE0P1 LE1P2 global

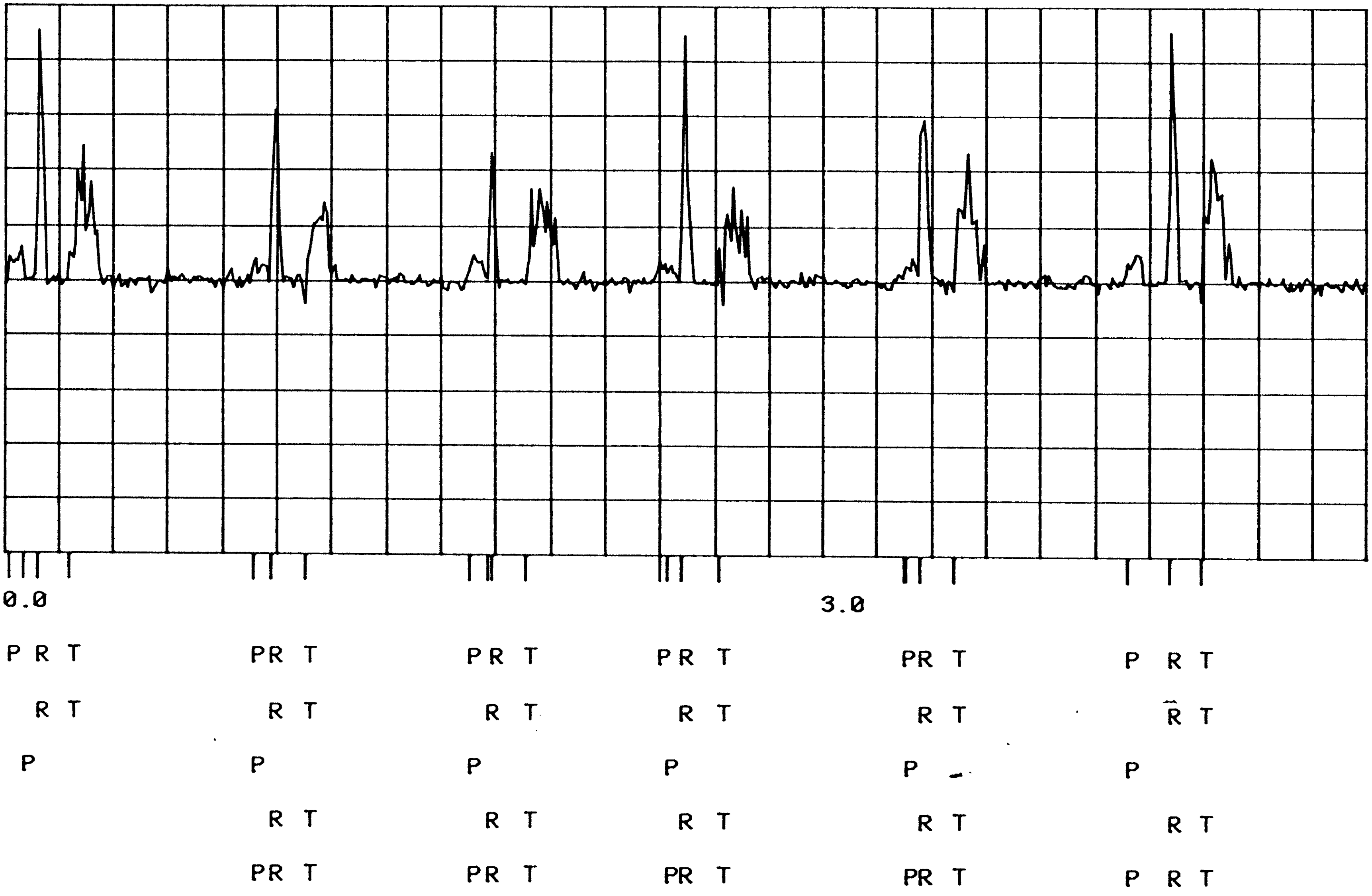


Fig 2 Panel 1

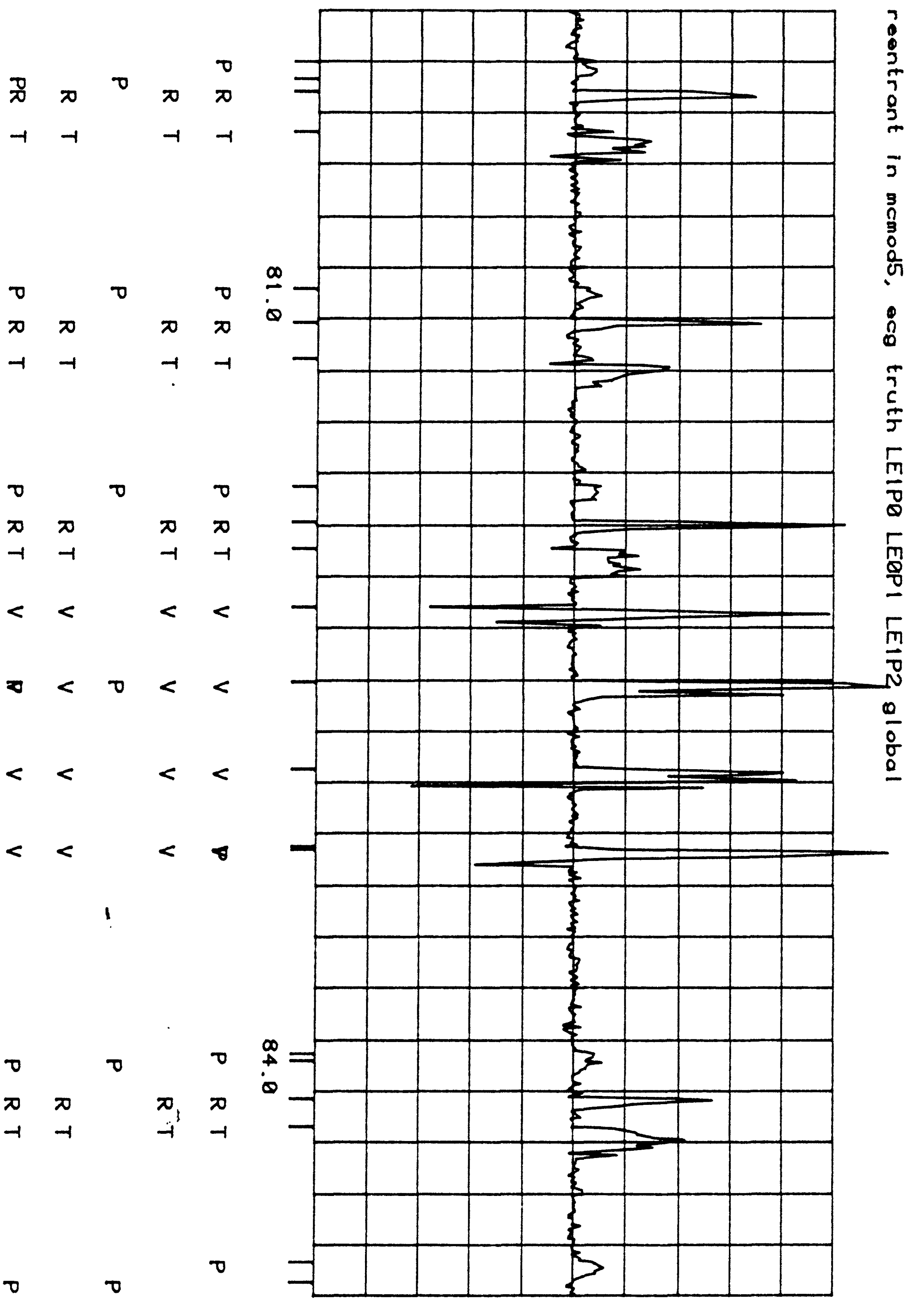

reentrant in mcmod5, ecg truth LE1P0 LE0P1 LE1P2 global
81.0
84.0
P R T
R T
P
PR T
V

Fig 2 Panel 2

reentrant in mcmod5, ecg truth LE1P0 LE0P1 LE1P2 global

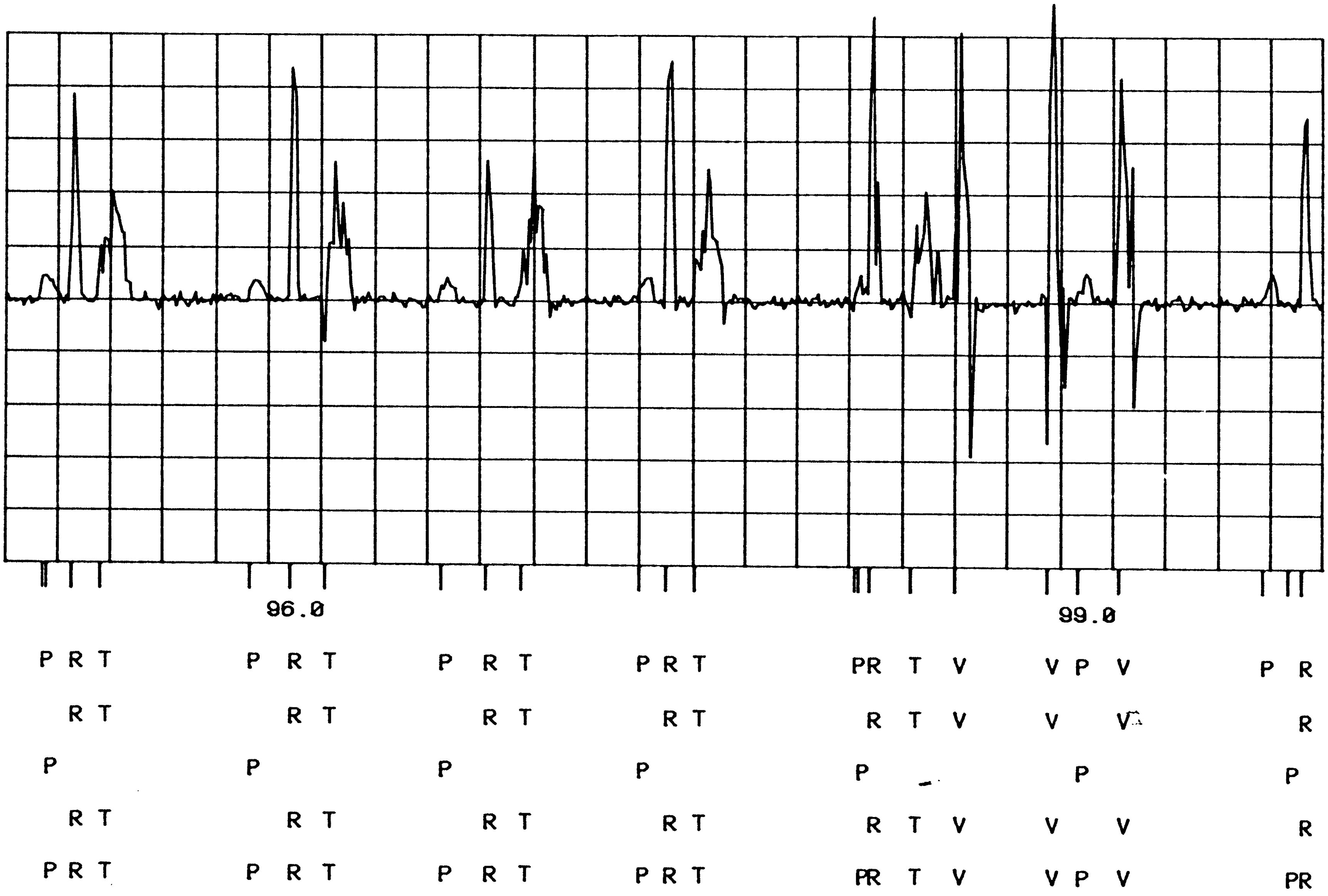


Fig 2 Panel 3

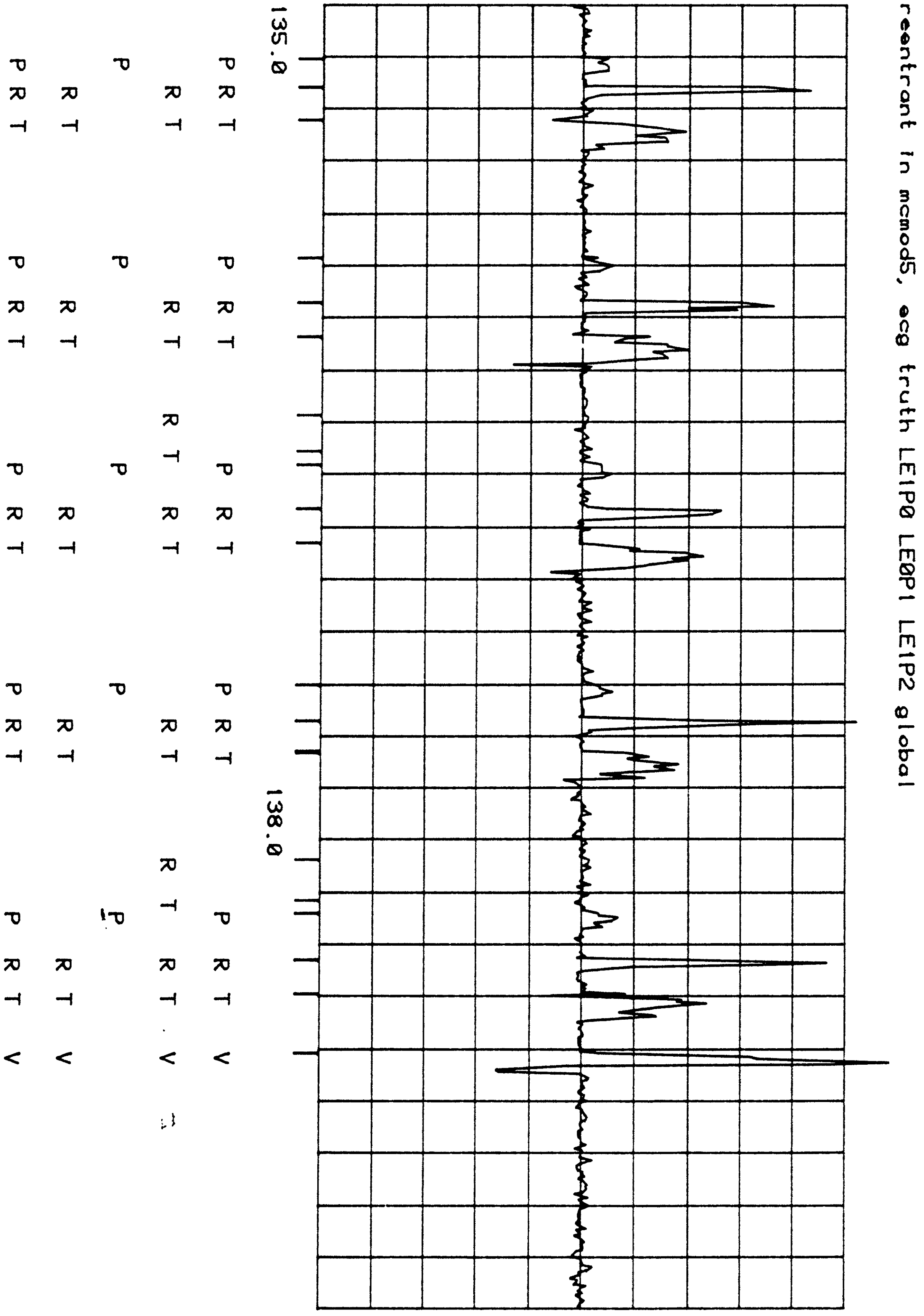

reentrant in mcmod5, ecg truth LE1P0 LE0P1 LE1P2 global
135.0
138.0
P R T P R T P R T P R T P R T V
R T R T R T R T R T R T R T R T V
P P P P P
R T R T R T R T R T V
P R T P R T P R T P R T P R T V

Fig 2 Panel 4

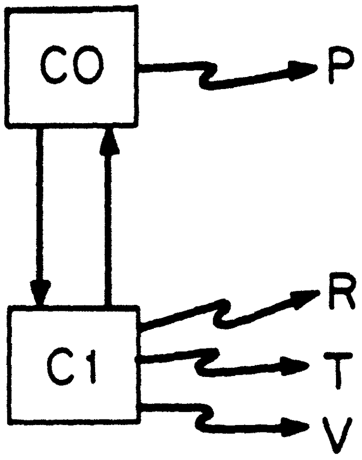
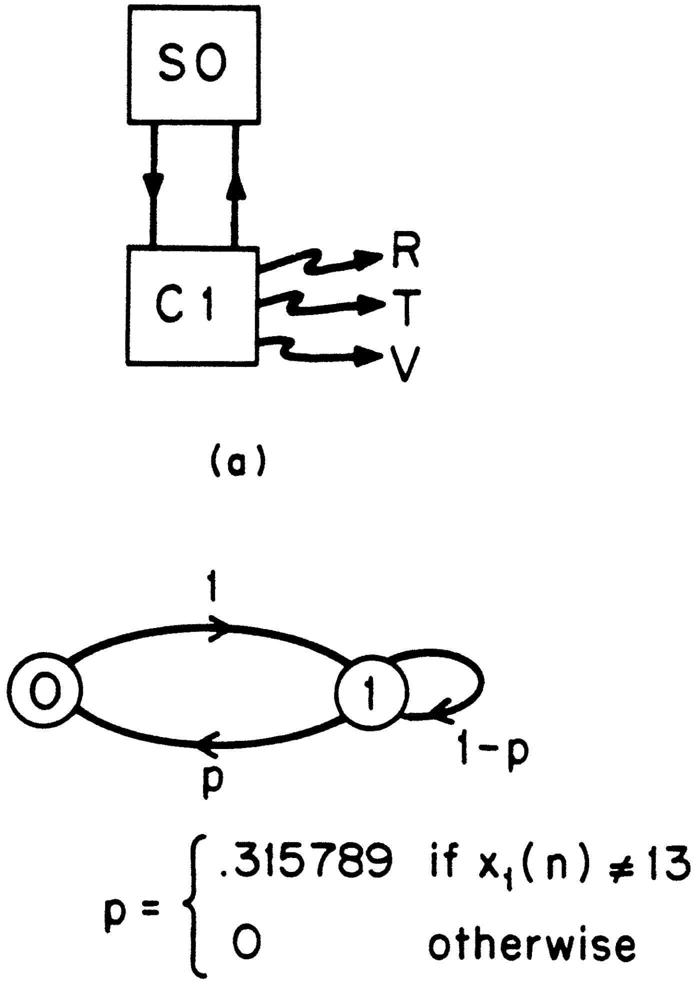


Fig 3

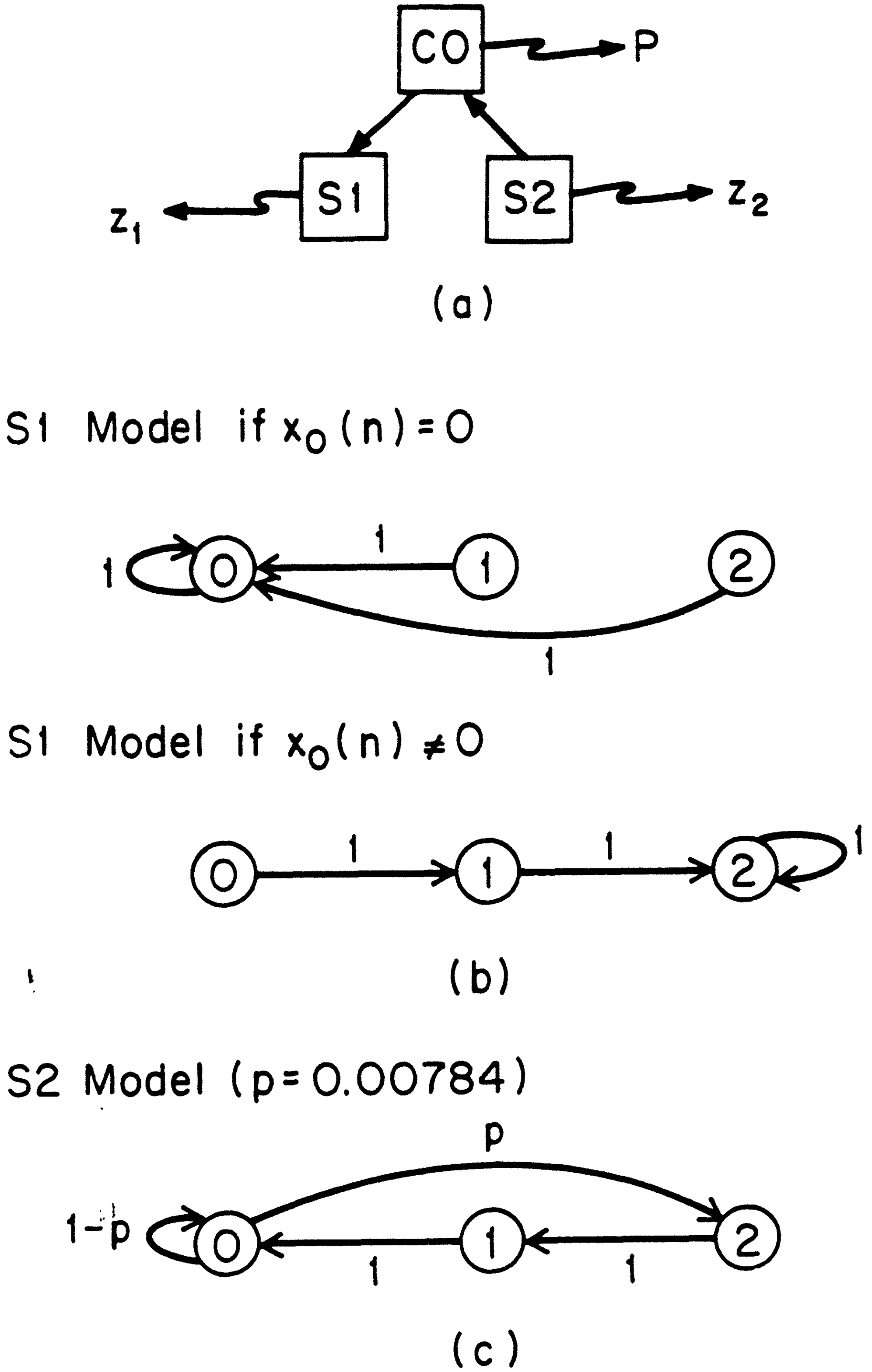


(a)

$$p = \begin{cases} .315789 & \text{if } x_1(n) \neq 13 \\ 0 & \text{otherwise} \end{cases}$$

(b)

Fig 4(a)(b)



(a)

S1 Model if $x_0(n)=0$

S1 Model if $x_0(n)\neq 0$

(b)

S2 Model ($p=0.00784$)

(c)

Fig 5 (a)(b)(c)

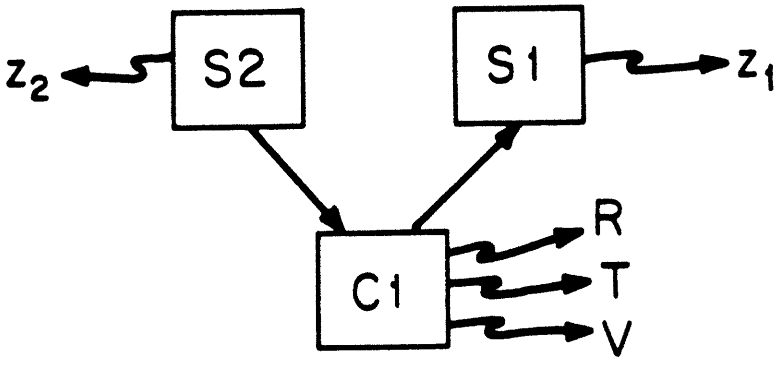


Fig 6

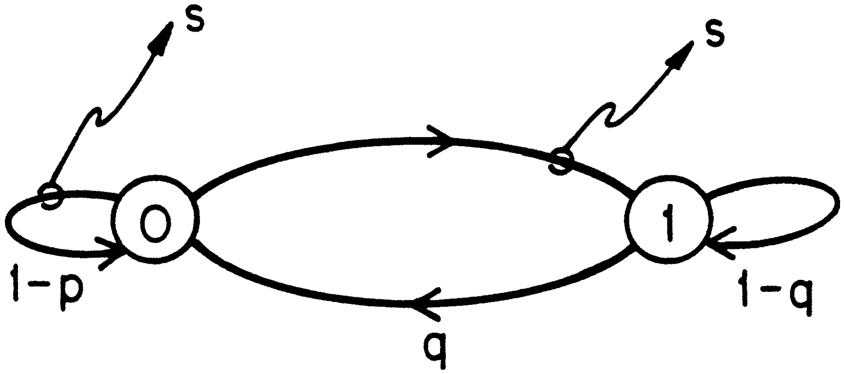


Fig 7

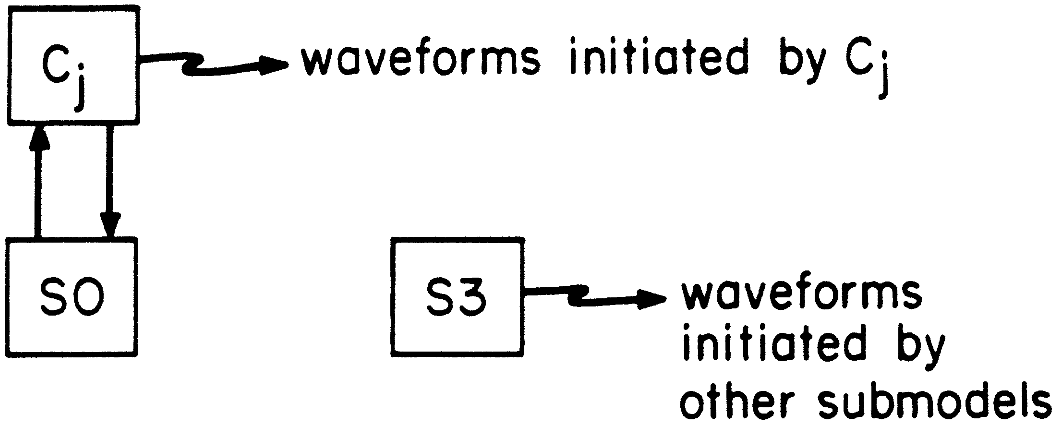

C_j
waveforms initiated by C_j
SO
S3
waveforms initiated by other submodels

Fig 8

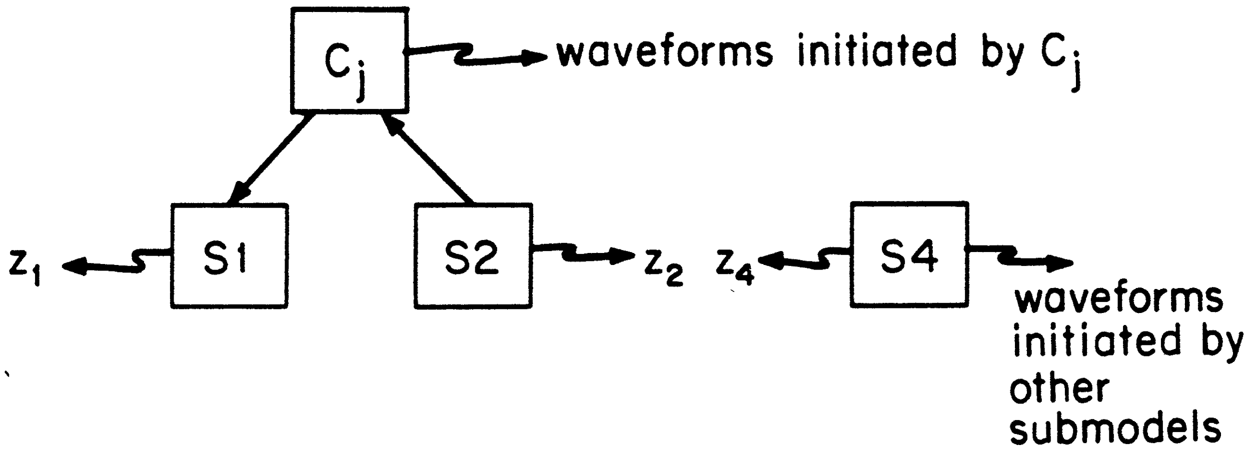

C_j
waveforms initiated by C_j
z_1
S1
S2
z_2
z_4
S4
waveforms initiated by other submodels

Fig 9