A Flexible Analysis-Synthesis Method for Transients

Harvey Thornburg, Fabien Gouyon
CCRMA, Stanford University
harv23@ccrma.stanford.edu

Abstract

We propose a new transient model which naturally extends the sinusoidal model to nonstationary phenomena. An integrated online-offline segmentation approach is developed, followed by an explicitly time-variant sinusoidal model.

1 Introduction

The *sinusoidal model* is a convenient sound-representation in analysis-synthesis frameworks. The signal is modeled as a sum of sinusoids, with time-variant amplitudes and frequencies, plus a *residual* $r(n)$, or unmodeled part:

$$y(n) \quad = \quad \sum_{k=1}^{2p} A_{k,n} \cos(\textstyle\sum_{m<n} \omega_{k,m} + \phi_{k,n}) + r_n$$

Such a *parametric*, *time-frequency* representation is ideal for analysis-synthesis goals: the parametric nature of the model allows meaningful specification of *transformations* on sound material, while the "time-frequency" nature facilitates intuition in developing musically viable transformations.

Classical sinusoidal models; e.g. [MQ86] use STFT as front end. Implicit is the assumption of *local stationarity*. Estimation biases occur to the degree local stationarity is violated, causing resynthesis artifacts under transformation.

1.1 Definition of "transient"

We define *transient* as a temporal *region* or *event* where local stationarity "fails to hold" (made precise w.r.t. a specific *family* of model structures).

To retain compact models and good generalization performance in the context of musical signals, we specialize to the following types: *rapidly decaying modes*, *abrupt* (step) *changes* in modal structure, and *fast local variations* in modal parameters.

2 Segmentation

In segmentation, two interrelated goals are to detect the presence of abrupt changes in modal structure, and to estimate their locations. To motivate our integrated solution, we summarize features of optimal detection and estimation algorithms:

Detection: *Online* algorithms operate sequentially to detect a single change in model parameters. By the stopping time of the test, an "operating characteristic" (OC) tradeoff is optimized between miss and false alarm probabilities (P_M, P_{FA}) and detection delay (ΔT).

With small ΔT, the stopping time gives also a (biased) estimate of the segment point. To detect multiple changes we restart the test. When a group of changes lies between the true segment point and the stopping time, this "burst" goes undetected. Complexity is inherently $\mathcal{O}(N)$ in data size.

Estimation: By contrast, *offline* algorithms assume a fixed number of changes, and find maximum likelihood (ML) estimate of their joint locations. Current methods are $\mathcal{O}(N^2)$.

A feasible segmentation approach uses fast online algorithms to screen for changes, identifying *regions* where segment points can be found. Location estimates are refined by offline algorithms given these regions.

2.1 Online detection

The tradeoff $(P_M, P_{FA}, \Delta T)$ is optimized using a repeated sequential likelihood ratio test. Let θ_0, θ_1 be model parameters before and after change (respectively). Define:

$$X_n = \log \frac{f_{\theta_1}(y_n|\mathbf{y}_{1:n-1})}{f_{\theta_0}(y_n|\mathbf{y}_{1:n-1})}, \quad S_n = \sum_{k=1}^{n} X_k \qquad (1)$$

Before change, drift in S_n is given by $E_{\theta_0} X(n) = -K(\theta_0, \theta_1)$, or relative entropy rate between joint distributions $f_{\theta_0}(\mathbf{y}_{1:n}), f_{\theta_1}(\mathbf{y}_{1:n})$, (constant w.r.t. n if y_n is stationary). After change, the drift is $E_{\theta_1} X(n) = K(\theta_1, \theta_0)$ which ≥ 0 $(= 0$ only when $f_{\theta_0} = f_{\theta_1})$, so drift is negative before change, positive after change. Use of X_n reduces the problem to detecting a positive step-jump corrupted by noise. The jump height $J(\theta_0, \theta_1) = K(\theta_0, \theta_1) + K(\theta_1, \theta_0)$ is a symmetric quantity called *divergence* and (after normalization by $\mathrm{Var}(X_n)^{1/2}$) indicates detectability.

Choose two-sided thresholds $(-a, b)$; $a, b > 0$, and let $T = \inf\{n : S_n \in (-a, b)\}$. If $S_T = -a$, decide $\mathcal{H}_0 : \theta = \theta_0$ (no change), otherwise if $S_T = b$, decide $\mathcal{H}_1 : \theta = \theta_1$ (change). Then if X_n are iid, an optimal tradeoff between the error probabilities and T is achieved [Bas81].

To use for change detection, restart the test by setting $S_T = 0$ upon "no change". $a \to 0$ gives optimality w.r.t. $(P_M, P_{FA}, \Delta T)$ [Bas81]. Equivalently, track the minimum $M_n = \inf\{S_k; k < n\}$ and stop on $T_b = \inf\{n : S_n > M_n + b\}$ to detect change.

The minimum-location $U = \inf\{n : M_n = M_T\}$ gives an alternate indication of the segment point. The best situation seems that of symmetric drifts, enforced by adding $\delta = [K(\theta_1, \theta_0) - K(\theta_0, \theta_1)]/2$ to X_n. However, (δ, b) should be tuned to achieve various points on the OC.

2.1.1 Divergence algorithm

In practice models θ_0, θ_1 must be *estimated.* The *divergence algorithm*, due to Basseville et al. [BB83] gives a practical sequential scheme. Model θ_0 is estimated using a growing window from the last change point; θ_1 is estimated using a sliding window of fixed size. The right boundaries of both windows coincide. Before the sliding window crosses the change point, estimates $\hat{\theta_0}, \hat{\theta_1}$ are based on the same underlying model θ_0 and should be close. The drift before change is negligible but the drift after change is $K(\theta_1, \theta_0)$. We seem to lose an amount $K(\theta_0, \theta_1)$ in the detectability. By a novel approach [BB83], the drift before change $-K(\theta_0, \theta_1)$ is estimated by conditional expectation on past data, e.g. $\int \log\left[(f_{\theta_1}(y_n|\mathbf{y}_{1:n-1}))/(f_{\theta_0}(y_n|\mathbf{y}_{1:n-1}))\right]$ $\times f_{\theta_0}(y_n|\mathbf{y}_{1:n-1})dy_n$ is subtracted from the increment X_n. Integrating w.r.t $f_{\theta_1}(\mathbf{y}_{1:n-1})$ obtains the additional drift-correction recovered after change.

A natural model structure for *spectral* changes is the Gaussian-AR model. i.e. $y_n + a_1 y_{n-1} + \ldots + a_p = e_n$, where e_n is Gaussian white noise; orders $(p \in [4, 8])$ seem adequate in practice. For the AR model, drift-corrected X_n is obtained in [BB83].

2.1.2 Operating characteristic tuning

If $\mathrm{Var}(X_n) < \infty$, by Donsker's invariance principle as $n \to -\infty$, the *process* S_n before or after change converges in law to Brownian motion. This allows tractable analysis of stopping times, running extrema, etc. Thereby, tuning formulae for (δ, b) given $(P_M, P_{FA}, \Delta T)$ are obtained (see [Bas81]). Particularly for short sliding windows ($<$ 64 samples) the Brownian approximation generally overestimates detection delay, and underestimates false alarm rate. Also in this case, sample estimators for $\mathrm{Var}(X_n)$ fail to converge: future work may involve other limit-process models for S_n, for instance α-stable diffusion processes.

2.2 Offline estimation

2.2.1 Dynamic programming

By independence assumption, the joint likelihood is a product distribution in the segments, thereby log-likelihood cost *adds* over the segments. Suppose we precompute an optimum cost matrix $L[i, j] = -\log f_\theta(\mathbf{y}_{i:j})$ for all $1 < i < j < N$. The matrix may be displayed on a network of *layers* of *nodes*; each layer corresponds to a segment point; each node within a layer to a candidate position. Then $L[i, j]$ is *path cost* between nodes in adjacent layers for signal positions (i, j). The segmentation problem becomes: find the shortest total path: this obtains a dynamic programming solution [Kay98]. The total path cost equals the (negative) log-likelihood to be minimized. The segment likelihood $L[i, j] = f_\theta \mathbf{y}_{i:j}$ is a product distribution in the conditionals: $f_\theta(y_k|\mathbf{y}_{i:k-1})$. For Gaussian AR models, $L[i, j]$ is filled by $\mathcal{O}(N^2)$ iterations of the RLS recursions for each $1 < i < j < N$.

2.2.2 GNC algorithm

Blake [Bla89] considers a related problem known as "weak string", where cost objective relates to likelihood of a piecewise AR(1). The cost is formulated as a sum of *quasiconvex* functions. If these functions were actually convex, overall cost would be convex. The idea of GNC (gradated nonconvexity) is: Start with related convex functions and morph them into quasiconvex counterparts, meanwhile running gradient descent. This reduces problems with local minima, obtaining an approximate solution in roughly $\mathcal{O}(N)$ time. The exact DP-solution can be used to assess the quality of the GNC-solution. Blake finds acceptable performance exceeding that of stochastic methods (e.g. simulated annealing). GNC awaits extension to the AR(p) case.

2.3 Online-offline integration

The diagram (Fig.1) summarizes our integrated solution, which breaks down by the following steps:

(1) Apply the online sequential test in *forward* scan. Keep both *stopping* and *min* times, i.e. a list of pairs $\{T_f, U_f\}$. Allow (via OC) high false alarm rate.

(2)Repeat (1) using *backward scan*, giving a list of pairs $\{T_b, U_b\}$ By the fundamental asymmetry in the divergence test, we get some independent information in the backward pass: it's unlikely to get the same false alarm in both passes.

(3) **Matching algorithm** Match forward and backward pairs. Unmatched pairs are eliminated as false alarms by thresholding a cost function. Currently *Length of convex hull* ($\{T_b, U_b, T_f, U_f\}$) and *Maximum difference*: $(\max|T_f - U_b|, |T_b - U_f|)$ are used as cost functions. An efficient sequential approach is used following an algorithm in [MQ86] for matching spectral peaks. A list of convex hulls of matched pairs is returned. Multiple segment points (*burst*) are indicated upon $T_b > T_f$. Subsequently, we pad both ends of each hull by $4p$ samples to ensure reliable model estimation for $L[i, j]$ computation.
(4) Run the offline algorithm on each region. Upon indication of burst, optimize over the number of changes as an order parameter, using Akaike criterion [MU86].

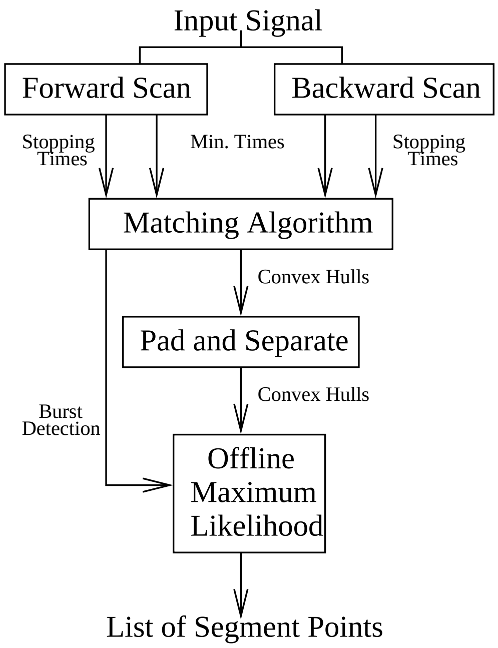

Figure 1: Segmentation: integrated approach

3 Transient modeling

To each region between identified segment-points we apply a separate transient model to account for rapidly time-varying and exponentially decaying modes. In the initial modeling stages (via order selection) a "stationarity test" may determine the salience of transient phenomena for that region: at which point, one may choose to switch to STFT-based methods, or continue with the rest of the transient model.

A system diagram is shown in Fig.2. Selected steps are detailed:

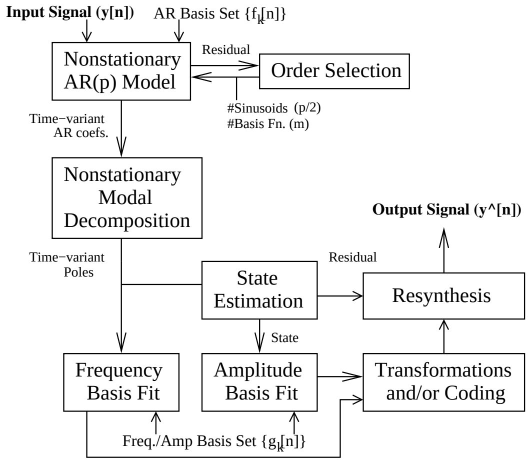

Figure 2: Transient modeling: system diagram

3.1 Fitting time-variant AR model

A method due to Grenier [Gre83] is: Given a time variant AR model; e.g. $y(n) + a_{1,n}y_{n-1} + \ldots + a_{p,n}y_{n-p} = e_n$, constrain the coefficients to a *basis*: for every $k = 1 \ldots p$, $a_{k,n+k} = \sum_{j=1}^{m} a_{kj} f_j(n)$. Basis $\{f_j\}$ should be structured, orthonormal; choosing f_1 constant recovers the stationary case upon $m = 1$; increasing m allows for greater nonstationarity. Grenier transforms the time-variant model into a stationary multichannel AR model; we identify using a covariance-form RLS technique.

3.2 Obtaining time-variant modes

Consider first the stationary case. Alternatively to the AR-polynomial form, we define the *modal realization* for y_n from e_n in state space: $x_{n+1} = Fx_n + Ge_n; y_n = Hx_n; F = \mathrm{diag}(p_1 \ldots p_p); G = [(\prod_{j\neq 1}(p_1 - p_j))^{-1} \ldots (\prod_{j\neq p}(p_p - p_j))^{-1}]^T$ is the *residue matrix*, and $H = [1 \ldots 1]^T$ sums over the states. Here, poles $\{p_k\}$ are the (assumed distinct) roots of $1 + a_1 z^{-1} + \ldots a_p z^{-p}$. The realization gives a parallel bank of one-pole sections, each producing a complex sinusoidal component. Real and imaginary parts of $\log x_{k,n}$ give respectively log-amplitude/phase per component; real and imaginary parts of $\log p_k$ give log-decay and frequency.

In the time-variant case, cascade poles may *differ* from the instantaneous roots of the AR-polynomial. Kamen [Kam88] obtains a recursion for the right cascade pole:

$$p_p(n) \quad = \quad -a_{1,n} - \sum_{k=2}^{p} a_{k,n} / (\prod_{j=2}^{k} p_{p,n+1-j}) \qquad (2)$$

Initialization $p_{p,0} \ldots p_{p,1-p}$ comes from roots of a prior (stationary) AR polynomial. Assuming p distinct roots, this gives by (eq.2) p distinct right pole trajectories.

Unfortunately, pole trajectories are not *attractive* (robust to errors in initial conditions). Most problematic are *low frequency modes*; small perturbations in AR coefficients may alter a complex pole pair into two real poles. Thereby the recursion (eq.2) is unable to produce complex poles. Slowly unstable, oscillatory errors result; we apply a second basis fit (separate for frequency/amplitude) to smooth these errors, as well as to recover a compact representation for the transient.

3.3 State estimation

Initial amplitudes and phases must be known to reconstruct sinusoidal trajectories. However, it is *insufficient* to estimate only the initial state. Errors in dynamics (time-variant poles) cause state error to drift. Whenever AR-modeling residual e_n is persistent, state noise gives another source of drift.

We use y_n to "close the loop" on these drifts. We estimate $\hat{x}_n$ to minimize a weighted error criterion for each n: $E_n = \sum_{k=1}^{N} W_{n,k}|y_k - H\Phi_{k,n}\hat{x}_n|^2$ (define $\Phi_{i,j}$ to satisfy $x_i = \Phi_{i,j}x_j$ in the absence of state noise) With a constant window ($W_{n,k} = 1$) drifts are merely "recentered" because dynamics impose a rigid constraint. However, using a double-exponential sliding window $W_{n,k} = \alpha^{|n-k|}$, we trade-off influence of nearby observations vs. influence of dynamics (indirectly via far-away observations) on estimation of the current state.

We note $\hat{x}_n$ can be obtained as the Fisher combination of a *forward* estimator $\hat{x}_{f,n}$ given $\mathbf{y}_{1:n}$, and strictly *backward* estimator $\hat{x}_{b,n}$ given $\mathbf{y}_{n+1:N}$: both estimators involving respectively, forward and backward one-sided exponential windows. By adapting exponentially-windowed RLS formulas [SK94], we obtain recursions:

Forward estimation Compute in arrays: $\hat{x}_{f,n}, P_{f,n}$, for $n = p \dots N-1$:

$$\begin{aligned} r_{n+1} &= 1 + \alpha^{-1} H F_n P_{f,n} F_n^* H^* \qquad (3)\\ k_{n+1} &= \alpha^{-1} F_n P_{f,n} F_n^* H^* / r_{n+1}\\ \hat{x}_{f,n+1} &= F_n \hat{x}_{f,n} + k_{n+1}(y_{n+1} - H F_n \hat{x}_{f,n})\\ P_{f,n+1} &= \alpha^{-1} F_n P_{f,n}(I - F_n^* H^* H F_n P_{f,n}/r_n) F_n^*\\ &\quad + \lambda G_n G_n^* \end{aligned}$$

Backward estimation Compute in arrays: $\hat{x}_{b,n}, P_{b,n}$, for $n = N-p \dots 1$:

$$\begin{aligned} r_{n-1} &= 1 + \alpha^{-1} H F_n^{-1} P_{b,n}^{+} F_n^{-*} H^* \qquad (4)\\ k_{n-1} &= \alpha^{-1} F_n^{-1} P_{b,n}^{+} F_n^{-*} H^* / r_{n-1}\\ \hat{x}_{b,n-1}^{+} &= F_n^{-1}\hat{x}_{b,n}^{+} + k_{n-1}(y_{n-1} - H F_n^{-1}\hat{x}_{b,n}^{+})\\ P_{b,n-1}^{+} &= \alpha^{-1} F_n^{-1} P_{b,n}^{+}(I - F_n^{-*} H^* H F_n^{-1} P_{b,n}^{+}/r_{n-1})\\ &\quad \times F_n^{-*} + \lambda F_n^{-1} G_n G_n^* F_n^{-*}\\ \hat{x}_{b,n-1} &= F_{n-1}^{-1}\hat{x}_{b,n-1}^{+}\\ P_{b,n-1} &= F_{n-1}^{-1}(P_{b,n-1}^{+} + \lambda G_{n-1} G_{n-1}^*) F_{n-1}^{-*} \end{aligned}$$

Terms involving λ, G correspond to adjustments for state noise, justified by considering the Kalman filter for the equivalent stochastic problem [SK94]. λ is an estimate of the residual variance $E[e^2(n)]$ for time-variant AR modeling.

Combining estimators For $n = p+1 \dots N-p$, compute:

$$\hat{x}_n = [P_{f,n}^{-1} + P_{b,n}^{-1}]^{-1}[P_{f,n}^{-1}\hat{x}_{f,n} + P_{b,n}^{-1}\hat{x}_{b,n}] \qquad (5)$$

For $n \le p$, use $\hat{x}_n = \hat{x}_{b,n}$; for $n > N-p$, use $\hat{x}_n = \hat{x}_{f,n}$.

4 Concluding remarks

Results (step-by-step examples of both segmentation and transient modeling algorithms) will be demonstrated during the conference. The most pressing issues for future work are: (1) improve OC-tuning in online segmentation by considering other limit-processes for S_n (eq.1); (2) extend Blake's GNC algorithm to AR(p) case for faster offline segmentation; (3) improve accuracy and stability in (eq.2), particularly for low-frequency modes.

References

[Bas81] M. Basseville. Edge detection using sequential methods for change in level–part ii: Sequential detection of change in mean. *IEEE Trans. ASSP*, 29:32–50, 1981.

[BB83] M. Basseville and A. Benveniste. Sequential detection of abrupt changes in spectral changes of digital signals. *IEEE Trans. Info. Theory*, 29:709–24, 1983.

[Bla89] A. Blake. Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE Trans. PAMI*, 11:2–12, 1989.

[Gre83] Y. Grenier. Time-dependent arma modeling of nonstationary signals. *IEEE Trans. ASSP*, 31:899–911, 1983.

[Kam88] E.W. Kamen. The poles and zeros of a linear time-varying system. *Lin. Alg. Appl.*, 98:263–89, 1988.

[Kay98] S. Kay. *Fundamentals of Statistical Signal Processing Vol. II: Detection Theory.* Prentice Hall, Englewood Cliffs, N.J., 1998.

[MQ86] R.J. McAulay and T.F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. ASSP*, 34:744–754, 1986.

[MU86] T. Matsuoka and T.J. Ulrych. Information theory measures with application to model identification. *IEEE Trans. ASSP*, 34:511–17, 1986.

[SK94] A.H. Sayed and T. Kailath. A state-space approach to adaptive rls filtering. *IEEE Sig. Proc.*, 11:18–60, 1994.