Bret Battey
c/o Music, Technology, and Innovation
Faculty of Humanities
Clephan Building
De Montfort University
Leicester LE1 9BH
United Kingdom
bret@bathatmedia.com
www.BatHatMedia.com

Bézier Spline Modeling of Pitch-Continuous Melodic Expression and Ornamentation

In comparison to Western classical music, numerous other musical traditions place far greater emphasis on the expressive shaping of the continuum between scale steps. In Trevor Wishart's terms, "[T]he focus of attention . . . is heavily weighted towards aspects of the sound articulation" that "cannot be related directly to the [scalar and metric] lattice. Thus there is a complex articulation of portamento structures leading onto or away from the lattice pitches . . . and focusing on the evolution of timbre within individual events" (1996). In this article, these traditions are referred to as *pitch continuum traditions*. The scalar framework on which such music is suspended may be highly codified theoretically. But the detailed and highly subtle pitch curves, ornaments, and inflections of the tradition usually remain at most vaguely described. They are absorbed through oral transmission, and musical notation proves unequal in the task of representing them.

South Indian (Carnatic) and North Indian (Hindustani) classical musics are highly refined traditions of this type. The term *gamaka* is sometimes used in both traditions as a general term for ornaments, embellishments, and pitch slides and will be used in that sense here. M. R. Gautam notes that the word derives from the Sanskrit root *gam*, which means "to move." By extension, *gamaka* means a conveyor or one who guides, leading one through and illuminating the microtones and spaces between scale tones. "It is like incandescence to the lamp. Without it the *svara* [scale tone] would be like a vestureless, lifeless body" (Gautam 1981). Interpreting Seeger melographs of South Indian vocal performances, Robert Gjerdingen went so far as to suggest, "If we conceive of movement as a primary phenomenon, then notes and rhythms become secondary phenomena" (1988).

The above serves to introduce two areas of inquiry. The first is "composerly" in nature: how, with a computer, can one effectively and convincingly render expressive melodic forms inspired by pitch continuum traditions? The historical roots of computer music lie in a period of history where Western art music began to strongly challenge the primacy of the pitch lattice for organizing musical expression. Despite this, the digital nature of the computer and the biases built into most computer music tools—whether commercial in origin or otherwise—mean that it is still very difficult with such tools to render refined pitch-continuum melodic expression. The challenge lies not only in the poverty of tools available for shaping the pitch-continuum; it lies also in the need to shape pitch, amplitude, and spectrum together in a deeply complex relationship.

The second problem area is musicological. The question of how pitch curves and ornaments are shaped to support expression, how they function within melodic form, and how they differ in various sub-traditions are all areas worthy of inquiry. However, the subtle, highly varied, and non-discrete nature of *gamaka* renders them opaque to traditional forms of analysis. This is reflected in the fact that, though there is a sizeable canon of theoretical literature on Indian classical music, writings on the topic of *gamaka* are rare and highly subjective in the nature of their descriptions. Can the computer be used to help describe and analyze *gamaka*—or continuous expression curves in other pitch-continuum traditions—in a quantifiable fashion?

This article describes a set of techniques developed in response to these issues. In short, the approach entails extraction of the pitch curve, amplitude, and spectral centroid data from a re-

Computer Music Journal, 28:4, pp. 25–39, Winter 2004

corded performance. It then divides the performance into phrases, identifies critical inflection points in the expression curves of each phrase, and fits constrained Bézier spline curves to the data between those points. The outcome is a perceptually near-equivalent simplification of the expression envelopes and a perceptually relevant numerical description of the curves that constitute it. This information can be used for musicological analysis. Alternatively, by selecting, recombining, and editing segments of data, a composer can repurpose the information toward expressive computer rendering of custom melodic gestures and musical textures.

As this article will show, real complexities are involved in the task of fitting the constrained Bézier splines to the data to model a recorded performance. However, one intent of this article is to demonstrate that, analysis of recorded performances aside, Bézier splines themselves provide a highly appropriate curve vocabulary and are powerful, intuitive, and easy to implement for descriptions of continuous expression curves.

The techniques described in this article were integrated into a software prototype entitled Pitch Curve Analysis and Composition System (PICACS). The prototype was developed in Macintosh Common LISP using Bill Schottstaedt's Common LISP Music (CLM; ccrma-www.stanford.edu/software/clm) and Rick Taube's Common Music (CM; commonmusic.sourceforge.net).

The author engaged in the core research and development for PICACS in India while studying performance in the *khyal* Hindustani vocal tradition. Thus, the first iteration of PICACS was developed and tested around the demands of *khyal* vocal music. However, it has also been proved effective with diverse sources such as Carnatic vocals, Carnatic violin, Persian classical vocals, and Chinese *chin* (zither) performances. It also seems likely that the process would prove applicable for modeling aspects of Western classical, popular, and jazz performance practices.

All analysis and graphs in this article are based on recordings by the author of Dr. Vikas Kashalkar singing an *alap* of *raag yaman*, unless otherwise noted. An *alap* is a slow and "unmetered" portion of a performance in which the basic character of the *raag* is introduced and explored. Dr. Kashalkar listened to a drone on headphones while performing to avoid including the traditional drone in the recording itself.

Graphs and texts in the article describe note names of the Indian *sargam* scale step nomenclature *Sa Re Ga Ma Pa Dha Ni*, corresponding to the Western *Do Re Mi Fa Sol La Ti*. The symbol ">" indicates *tivra* (sharp), which may be applied to *Ma*. The octave number follows, with octave 1 being the octave of the drone. In Indian classical music, the frequency of *Sa* is not fixed; it varies by performer and performance. In the graphs, scale steps are determined by calculating the frequency ratios relative to *Sa* on the basis of a just intonation system described by Alain Daniélou (1980).

Data Extraction

PICACS utilizes autocorrelation for fundamental frequency estimation (Rabiner 1977), presuming the maximum peak in the autocorrelation function to represent the fundamental. However, the technique of some *khyal* vocalists strongly emphasizes upper harmonics, increasing the risk that a harmonic other than the fundamental could be extracted. Further, noise is often introduced into the vocal sound in *taans* (rapid, aggressive passages) in *khyal* style. Therefore, it was necessary to supplement the basic autocorrelation functions with features that attempt to preserve continuity in the pitch curve.

One conventional qualification is a minimum and maximum pitch value validity check. Subsequent autocorrelation measurements are then measured against a chosen ratio threshold defining continuity. If a measurement appears out of range or discontinuous, the autocorrelation of that frame will be executed again using a different window size or offsetting the window start by one or more samples, up to some given maximum number of window sizes and sample offsets. Quite often, a valid measurement will be returned before options are exhausted. If no valid measurement is returned, the fundamental is recorded as zero for the given window. Zero will also be recorded if the RMS amplitude of the window lies below a given threshold. After all measurements have been extracted, post-processing utilizes linear interpolation to bridge data gaps that are within the width of a given time window.

For amplitude, PICACS extracts a standard RMS function

$$r = \sqrt{\frac{\sum_{k=1}^{N} a_k^2}{N}} \quad (1)$$

where N is the number of samples in an analysis frame, and a_k is the linear amplitude of sample k.

The compositional intent of PICACS is not to imitate a performance, but rather to allow continuous aspects of expression to be abstracted, readily manipulated, and analyzed or re-purposed toward synthesis of a completely different timbral character. Therefore, detailed timbre modeling was not implemented. Instead, PICACS extracts the spectral centroid of the performance. The spectral centroid is an objective measurement that has been shown to have strong correlation to the subjective sense of brightness of a sound (Beauchamp 1982; Hajda 1996; McAdams et al. 1999; Jehan and Schoner 2001). In particular, an FFT-derived, fundamental-normalized calculation is used:

$$c = \frac{1}{f_0} \frac{\sum_{k=1}^{N} F_k A_k}{\sum_{k=1}^{N} A_k} \quad (2)$$

where N is the number of bins in the FFT analysis, F_k is the frequency of a given bin, A_k is the magnitude of the bin, and f_0 is the fundamental frequency at the start of the window. The core of the formula provides the spectral centroid in Hertz; the multiplication by $1/f_0$ provides a ratio of this value relative to the fundamental frequency. Thus, tones of the same brightness but of different fundamentals return the same value.

Identification of Critical Points

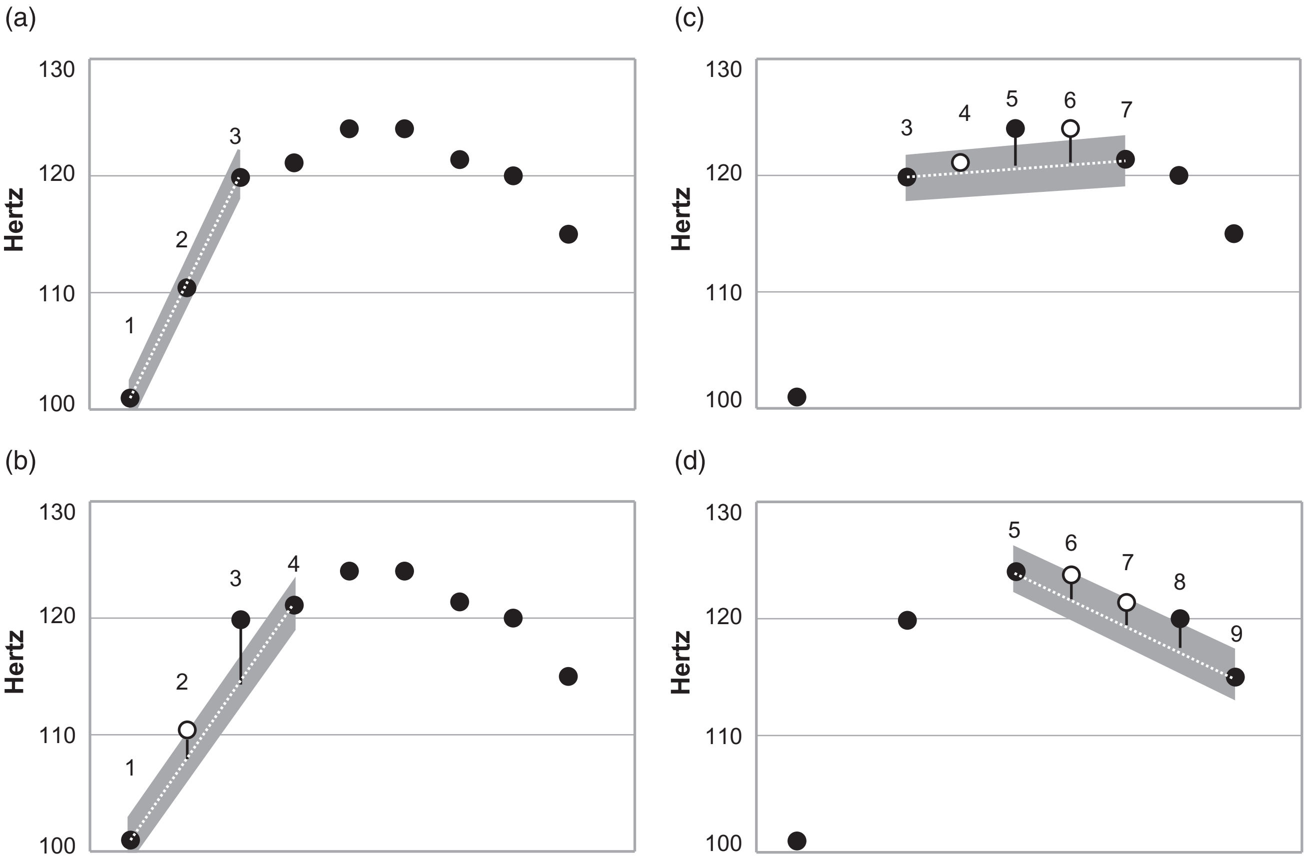
In the context of *khyal*, we can define a phrase as a musical unit sung with a single breath and exhibiting a continuous pitch curve throughout. In PICACS, the extracted pitch envelope is analyzed to segment the performance into such phrases based on identifying spans of contiguous non-zero pitch values. Any such spans shorter than a given time threshold are assumed to be anomalies and are discarded. PICACS uses a default time threshold of 0.5 sec.

For each phrase, it is then necessary to identify critical inflection points in the pitch envelope. These will include not only obvious maxima and minima, but also knees and points where periods of linear change are entered or exited.

First, all points on the curve where the slope changes between negative, zero, and positive are identified. These points and the endpoints together form a set of candidates from which the critical points will be selected. We will notate this set of candidates as $D = \{(t_i, f_i)\}_{i=1}^{k}$, where t_i and f_i are the time and frequency values of a point, and k is the total number of candidates.

Applying a *JND-line filter* thins the set of candidates based on just noticeable difference (JND) measurements for pitch perception. The intent is to discard candidate points that provide only imperceptible detailing in the pitch envelope. The following description of the details of the process will be most easily understood with reference to Figure 1. We start by accepting D_1 (the first point in the envelope) as the first critical point. For any critical point D_j, we test candidate points subsequent to a critical point D_j by projecting a line L between D_j and D_{j+d}, where d is initially 2. We test the candidate point D_{j+d-1} by measuring the ratio between that point and the frequency value of L at time t_{j+d-1}. If the ratio is within the pitch JND range appropriate for L at that point, the point is discarded, d is incremented, and the above process is repeated. Otherwise, the point in the set $\{D_{j+1} \ldots D_{j+d-1}\}$ that is the maximal ratio distance from L is accepted as a critical point and now becomes D_j, d is reset to 2, and the above process is repeated. If two or more points are maximal and equal, the earlier one with respect to time is chosen. By default, PICACS uses a simple JND measurement of $\pm 3\%$ at 100 Hz linearly interpolated to $\pm 0.5\%$ at 2000 Hz (Roederer 1979). The final point in the pitch curve is always included in the resulting set of critical points.

Figure 1. Selected steps in a hypothetical candidate point-filtering process. (a) Point 2 lies within a pitch-JND band projected from points 1 to 3 (as indicated by the gray line), so (b) point 2 is discarded (indicated by the open dot), and the JND band is projected from point 1 to point 4. Point 3 now lies outside the JND band, so (c) point 3 becomes the new test base point. JND bands are projected from point 3 to points 5 and 6, but only when projected to point 7 do the intervening points (5 and 6) lie outside the JND band. Point 5 lies farther from the JND band than does 6, so point 4 is discarded, and (d) point 5



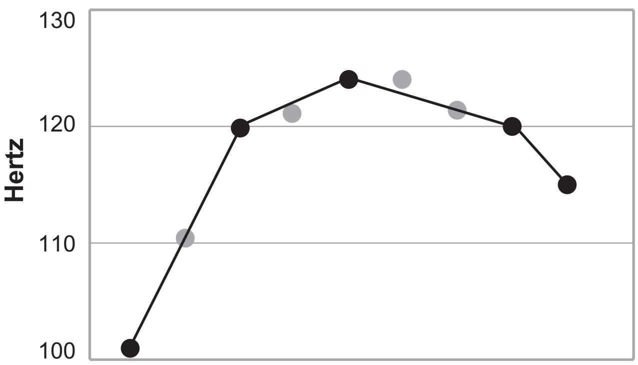
It can be argued that such JND measurements—developed for an "average" (and presumably Western) listener and highly variable depending on context, intensity, and duration—are not appropriate for the given task. The rationale for their use in PICACS is that the given JND curve is intended to provide only a reasonable starting point for analysis. It can be scaled on a phrase basis if audible details are being obscured or if extraneous details are being selected.

In the case of long held pitches or long linear changes in pitch, the given algorithm can result in a set of critical points that, when a spline is fitted between them, will on average be perceptibly sharp or flat or will slew into such a state. Therefore, held or linear segments are identified by a slope heuristic based on the deviation in cents of the line segment over its length. If the slope of the segment is below a given range, a straight line is fit (in a least squares sense) against the pitch curve data while maintaining the two end time points. We can begin with a standard line formulation $f = st + o$, where t and f are time and frequency values in our envelope, s is the slope, and o is an offset. A conventional solution to the fitting task is to translate the line equation into a system of equations of the form $\mathbf{Ax} \cong \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} \mathbf{t_0} & \mathbf{1} \\ \mathbf{t_1} & \mathbf{1} \\ \vdots & \vdots \\ \mathbf{t_m} & \mathbf{1} \end{bmatrix}, \; \mathbf{x} = \begin{bmatrix} \mathbf{s} \\ \mathbf{o} \end{bmatrix}, \; \mathbf{b} = \begin{bmatrix} \mathbf{f_0} \\ \mathbf{f_1} \\ \vdots \\ \mathbf{f_m} \end{bmatrix} \tag{3}$$

That is, the **A** matrix contains a column of the known time points and column of ones, the **b** matrix contains a column of the known frequency values, and **x** represents the unknown slope and offset. PICACS uses QR decomposition (Johnson et al. 2002) to solve the best-fit solution for **x**. Given the *is the new test base point. JND bands are projected from point 5 to points 6, 7, and 8, but only when projected to point 9 does an intervening point (8) lie outside the band. Thus, points 6 and 7 are discarded and point 8 is retained. (e) The last point (9) is retained as a matter of course.*

(e)



resulting slope and offset, a line can be projected between the starting and end times of the problematic segment. The end frequencies of the fitted line replace the previous frequencies of the pair of critical points. If two contiguous segments are so adjusted, the frequency of their common critical point is determined by averaging the two calculated frequencies. Figure 2 depicts critical points of a phrase before and after the adjustment of held segments.

Finally, an additional filter is applied to remove any spurious critical points that are within a small time window of their predecessor. In PICACS, this value defaults to 0.05 sec.

Curve Fitting

Cubic functions of the form $f(x) = ax^3 + bx^2 + cx + d$ were considered first for defining the curves between critical points. Such functions provide an advantage in that they are relatively simple computationally, and routines for constrained fitting of such functions to data are readily available. However, the four coefficients required to describe the curve would provide little or no intuitive relationship to aesthetic character or intent. That is, they would prove very challenging to interpret from a musical perspective, and it would be difficult to know how to alter the coefficients to achieve particular expressive purposes. Sinusoidal functions were also considered, but again, the difficulty was finding a means to control the functions in aesthetically appreciable terms.

After viewing a variety of pitch curves extracted from performances, it seemed possible that Bézier splines could also provide an appropriate shape vocabulary for the task of modeling expression curves. As will be demonstrated, they provide the additional advantage of simple and intuitive means of control readily related to aesthetic intent. This latter attribute has led to their widespread adoption for shape control in computer-aided design programs and graphic-design applications, as well as for velocity control in keyframe-based interfaces in animation and video editing environments. In the case of PICACS, the simplicity of the shape control method for Bézier splines should also facilitate statistical analysis owing to the small set of numbers required to define the curve shape.

Bézier Splines

For the purposes of modeling a pitch curve, a third-order Bézier spline (a parametric cubic curve) can be expressed in the following form (Foley et al. 1990):

$$Q(u) = B_0(u)P_0 + B_1(u)P_1 + B_2(u)P_2 + B_3(u)P_3 \quad (4)$$

where $0 \le u \le 1$, points P_n are time and frequency points (T_n, F_n), and B_n are the Bernstein polynomials shown in Table 1.

Therefore, to determine the point $Q(u)$, which occurs at time $t(u)$ and frequency $f(u)$, two cubic polynomials must be calculated:

$$Q(u) = \begin{pmatrix} t(u) \\ f(u) \end{pmatrix} = \begin{pmatrix} B_0(u)T_0 + B_1(u)T_1 + B_2(u)T_2 + B_3(u)T_3 \\ B_0(u)F_0 + B_1(u)F_1 + B_2(u)F_2 + B_3(u)F_3 \end{pmatrix} \quad (5)$$

In general, points P_n are referred to as the control points of the curve; they define the position and shape of the curve (see Figure 3). More specifically, we will refer to points P_0 and P_3 as the endpoints of the curve and points P_1 and P_2 as the inner control points. The slope between an inner control point and its respective endpoint determines the slope of the tangent at that endpoint. The inner control points also determine the relative weighting of the curve toward their respective endpoints.

Figure 2. Adjusting critical points on extended linear segments. (a) Critical points prior to adjusting held segments. Bold lines indicate segments where the initial critical point selection is likely to result in audibly out-of-tune results. (b) Critical points after adjustment. Frequency values of problematic points have been altered by least-squares fitting of the indicated line segments to the pitch curve.

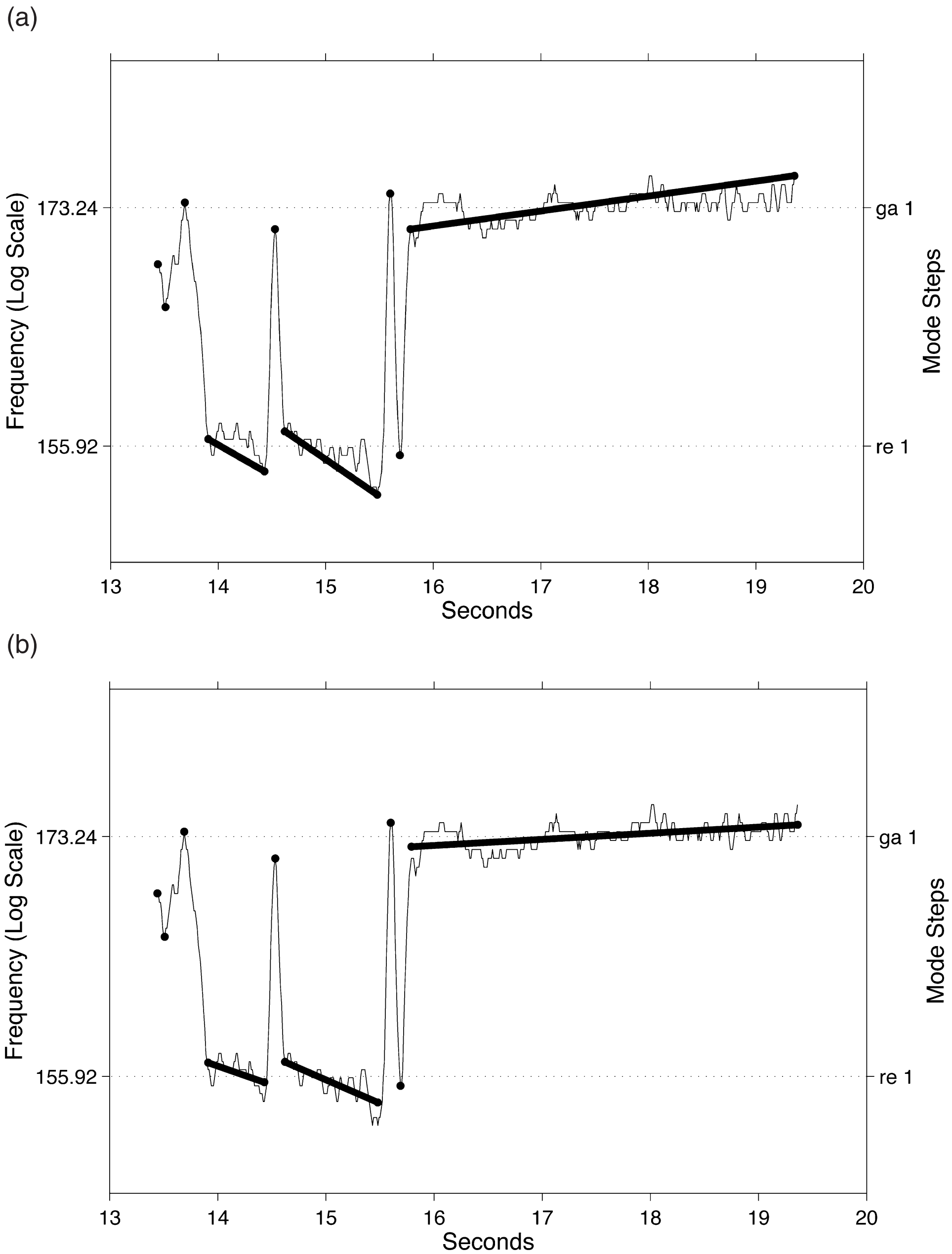


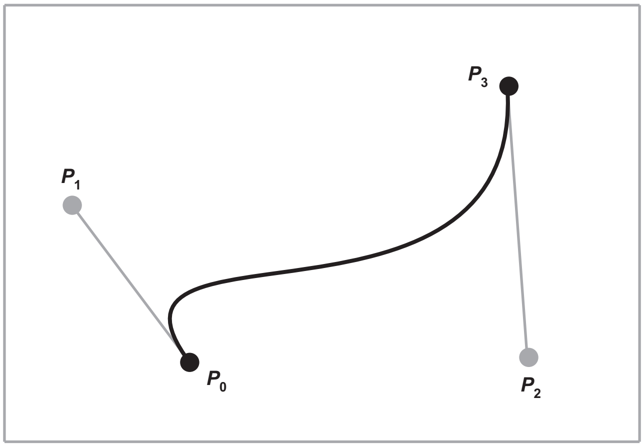
Table 1. Bernstein Polynomials Used in Equation 4

$B_0(u) = (1-u)^3$
$B_1(u) = 3u(1-u)^2$
$B_2(u) = 3u^2(1-u)$
$B_3(u) = u^3$

Constrained Formulation

Using conventional Bézier splines, we have an eight-dimensional definition of the curve shape: a pitch and a frequency point for each of four control points. If the dimensions of the definition can be

Figure 3. Third-order Bézier spline defined by four points.



reduced while retaining sufficient range of behaviors to match the problem domain, the simplification will be a gain from both the musical analysis and the compositional perspective. Furthermore, given that the time domain must always move forward, the model must be constrained to ensure that the curves are always monotonically increasing or decreasing. Monotonicity is also desirable for the purpose of pitch analysis, ensuring that the curve's endpoints are global maxima and minima. Finally, in the vast majority of cases, curves have a slope of zero when they join at the critical point. Therefore, the following constraints are applied: $F_1 = F_0$ and $F_2 = F_3$, $T_0 \leq T_1, T_2 \leq T_3$. That is, moving the T position of the inner control points within the bounding box defined by the endpoints is the only means for shaping the curve. It then proves convenient to define these two T positions as ratios of the distance between T_0 and T_3:

$$R_0 = \frac{T_1 - T_0}{T_3 - T_0},\ R_1 = \frac{T_3 - T_2}{T_3 - T_0},\ R = \{R_0, R_1\} \qquad (6)$$

Any curve in the model can now be described with two endpoints $\{T_0, F_0\}\{T_3, F_3\}$ and two scalars $\{R_0, R_1\}$.

Figure 4 depicts the range of curves supported by this approach.

Fitting Techniques

The simplicity of the curve for the end user is purchased with complexities involved in fitting a Bézier curve to ordered data. We wish to establish a least-squares fit of a constrained Bézier curve to the ordered set of time and frequency data points $\hat{C} = \{C_i\}_{i=0}^m\ \{(t_i, f_i)\}_{i=0}^m$ by choosing the best time positions for the two inner control points. This problem cannot be solved directly by standard linear least-squares methods.

Parameterization Estimation

Three approaches to fitting the spline are described below. A fundamental problem is that we cannot analytically determine the precise nodes $\hat{U} = \{u_0, \ldots, u_m\}$ appropriate to our data set $\hat{C} = \{(t_i, f_i)\}_{i=0}^m$. Thus, all three methods rely on an initial estimation of this parameterization. Classical techniques for such estimation are described by Gerald Farin (1993); of these, PICACS utilizes centripetal parameterization (Lee 1989), in which $u_0 = 0$ and

$$u_i = u_{i-1} + \frac{\|C_i - C_{i-1}\|_2^\alpha}{\sum_{j=1}^{m} \|C_j - C_{j-1}\|_2^\alpha},\ 1 \leq i \leq m,\ 0 \leq \alpha \leq 1 \qquad (7)$$

Here, $\|C_i - C_{i-1}\|_2$ indicates the 2-norm of the consecutive points C_i and C_{i-1}, that is, $\|C_i - C_{i-1}\|_2 = \sqrt{(t_i - t_{i-1})^2 + (f_i - f_{i-1})^2}$. In the PICACS implementation, α is always set to 0.5, although theoretically other values may provide tighter parameterization for some data sets. For $\alpha = 0.5$, the equation may be simplified to

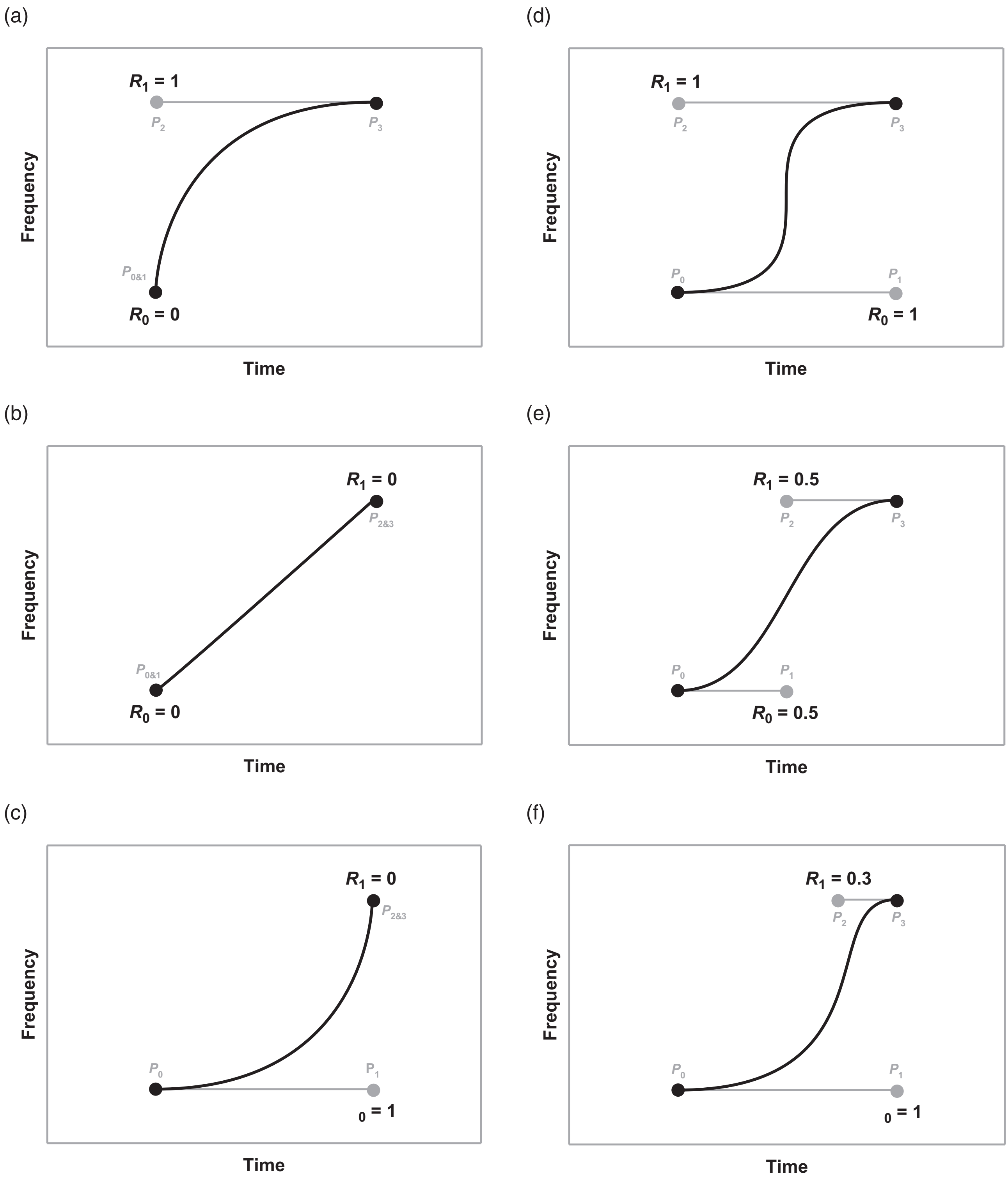
$$u_i = u_{i-1} + \frac{(t_i - t_{i-1})^2 + (f_i - f_{i-1})^2}{\sum_{j=1}^{m} (t_j - t_{j-1})^2 + (f_j - f_{j-1})^2} \qquad (8)$$

A Rough, Linear Least-Squares Solution

Given the estimated parameterization $\hat{U}$, a rough solution to the fitting problem can be established by fitting the curve entirely in terms of $t(u)$. That is, because there are no unknowns in the $f(u)$ equation in Equation 5, we can only solve the system by finding an appropriate $\{T_1, T_2\}$ in the $t(u)$ equation. Rearranging Equation 5 to place the known $\{T_0, T_3\}$ and unknown $\{T_1, T_2\}$ on opposite sides of the equation and incorporating the data points t_i yields

$$B_1(u_i)T_1 + B_2(u_i)T_2 = t_i - B_0(u_i)T_0 - B_3(u_i)T_3 \qquad (9)$$

Figure 4. Hypothetical pitch curves defined by third-order Bézier splines in which the four control points are bounded by frequency constraints $F_1 = F_0, F_2 = F_3$ and time constraints $T_0 \leq T_1, T_2 \leq T_3$. R is the set of the ratios defining the position of T_1 and T_2 between T_0 and T_3. (a) R = {0 1}; (b) R = {0 0}; (c) R = {1 0}; (d) R = {1 1}; (e) R = {0.5 0.5}; (f) R = {1 0.3}.



One can now form the system $\mathbf{Ax} \cong \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} \mathbf{B_1(u_0)} & \mathbf{B_2(u_0)} \\ \mathbf{B_1(u_1)} & \mathbf{B_2(u_1)} \\ \vdots & \vdots \\ \mathbf{B_1(u_m)} & \mathbf{B_2(u_m)} \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} \mathbf{T_1} \\ \mathbf{T_2} \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{t_0 - B_0(u_0)T_0 - B_3(u_0)T_3} \\ \mathbf{t_1 - B_0(u_1)T_0 - B_3(u_1)T_3} \\ \vdots \\ \mathbf{t_m - B_0(u_m)T_0 - B_3(u_m)T_3} \end{bmatrix} \quad (10)$$

This can be solved using standard linear least-squares techniques, such as QR decomposition. The resulting $\{T_1, T_2\}$ provides a reasonable estimate in many circumstances. By either implementing an upper- and lower-bound constraint into the solver or substituting straight lines in cases where an out-of-bounds solution is returned, it might prove a viable calculation for use in situations in which analysis must be executed quickly, such as in live performance. However, in some circumstances it can fail to track curves effectively, and a better solution is clearly required. The rough least-squares solution does, however, provide a useful starting point by providing an initial estimated solution to use in the two nonlinear approaches described below.

Nonlinear Least Squares with Alternating Minimizations

The first high-quality solution is a nonlinear approach for fitting based on methods described by Richard Bartels and David R. Warn (1994). They note that one can use a succession of alternating minimizations: one in which the parameterization $\hat{U}$ is optimized given a fixed set of control points, then the other in which the control points are optimized to provide a best fit given this $\hat{U}$. The alternating minimizations proceed until the system converges. The minimization of $\hat{U}$ is facilitated by the fact that each u_i can be minimized separately, with each instance being a simple univariate minimization.

This approach was tested in MATLAB using the constrained nonlinear solver FMINCON. The objective function accepted $\{T_1, T_2\}$ as the unknowns, and $\hat{U}$ and the remaining known control points were passed as givens. The set $\{T_1, T_2\}$ was constrained to $T_0 \leq T_1,\ T_2 \leq T_3$. Internally, the objective function first individually minimized each u_i on the interval $0 \leq u_i \leq 1$ using FMINBND, then rendered the Bézier spline, and finally returned the total least-squared residual $r(T_1, T_2) = \|C_i - Q(\hat{U}; T_1, T_2)\|_2^2$. The method proved capable of returning a very low residual but was also computationally expensive. It may be possible to create a more efficient solution through analytical determination of a gradient.

Note that, with this solution, to truly recreate the closest approximation curve to $\hat{C}$ after solving the system, one would have to use not only the $\{T_1, T_2\}$ returned by the solver, but also the final $\hat{U}$. In compositional application of PICACS, however, one will normally be rendering a Bézier spline by generating a $\hat{U}$ of equally spaced u_i and applying this and the given $\{T_1, T_2\}$ to Equation 5. Given that this equally spaced $\hat{U}$ would be different from that created by the solver to determine $\{T_1, T_2\}$, the resulting curve would, in all likelihood, not represent as tight a fit as one using the solver's final $\hat{U}$.

Nonlinear Least Squares with Linear Interpolation

This observation leads to a third approach to the fitting problem, in which we incorporate the generation of an equally spaced $\hat{U}$ directly into the solver to more closely fit the context in which the solution will be applied. We again use a nonlinear solver with an initial guess provided via Equation 10 and implement an objective function that takes a proposed $\{T_1, T_2\}$ and the remaining known control point data as givens. With this method, however, the objective function generates a $\hat{U}$ consisting of evenly spaced nodes and renders a Bézier spline $Q(\hat{U}; T_1, T_2)$. We now want to compare each spline value $f(u_i)$ to the original data. But, whereas in our original data each f_i corresponds to a point t_i, in the spline each $f(u_i)$ corresponds to an equally spaced u_i, each with its own $t(u_i)$. That is, we cannot analytically determine the value of the spline at a given t_i, nor can we determine the value of the data set corresponding to an arbitrary $t(u_i)$. Instead, to compare the original data and the spline, we must estimate the value of the original data at each point $t(u_i)$.

To do so, we establish $\bar{P}(\hat{C})$, a function consisting of piecewise linear segments formed between the points of data set $\hat{C}$. We can now estimate the least-squared residual between the original data and the rendered curve:

$$r(T_1, T_2) = \sum_{i=0}^{m} (\bar{P}(t(u_i)) - f(u_i))^2 \tag{11}$$

That is, we are using each $t(u_i)$ from the Bézier spline as a linear interpolation index into the original data set to calculate an approximate $f(t(u_i))$, allowing us to then estimate the distance between the spline value $f(u_i)$ and the data.

With this approach, it is also easy to reduce calculation time by reducing the density of nodes in $\hat{U}$. To the author's ear, audibly acceptable results can still be attained while thinning the nodes by as much as 25% on pitch curves containing 100 data points per second.

PICACS implements this solution with a Nelder-Mead nonlinear solver, also known as a Simplex or Polytope solver. This is a robust solver for circumstances in which a gradient is not available (Gill, Murray, and Wright 1981). Because the Simplex solver itself does not support boundary constraints, those constraints are implemented in the objective function by returning a maximal residual when boundaries are violated. The Simplex solver is slow compared to more complex methods, but it has proved sufficiently fast for the problem domain, and it has the benefit of being relatively easy to program. Unusually long curves, such as a single pitch curve extending over several seconds, slow down computation considerably.

Contiguous Splines

The complete model of the pitch envelope consists of a series of contiguous Bézier splines in which the endpoint of one spline is the starting point of the next spline. In the PICACS implementation, these are referred to as *bezlists*. Each critical point is now a joint between splines. These joints—as well as the start of the first segment and the end of the last—are called *bezlist nodes*. (Note that this is a different use of the term ''node'' from that which describes the components of $\hat{U}$.)

Amplitude and Brightness Modeling

The same process of critical-point identification and curve fitting is applied to the amplitude and spectral centroid envelopes. The amplitude envelope is treated with a moving-average filter prior to analysis to reduce the noisiness of the signal, though this does diminish the range of the curve by scaling back extrema points. The default JND difference is 1 dB for amplitude analysis. JND measurements for spectral centroid do not exist, but in practice a JND threshold of 0.5 units has worked effectively for simplifying the often-complex envelope while maintaining expressive detail.

Rendering

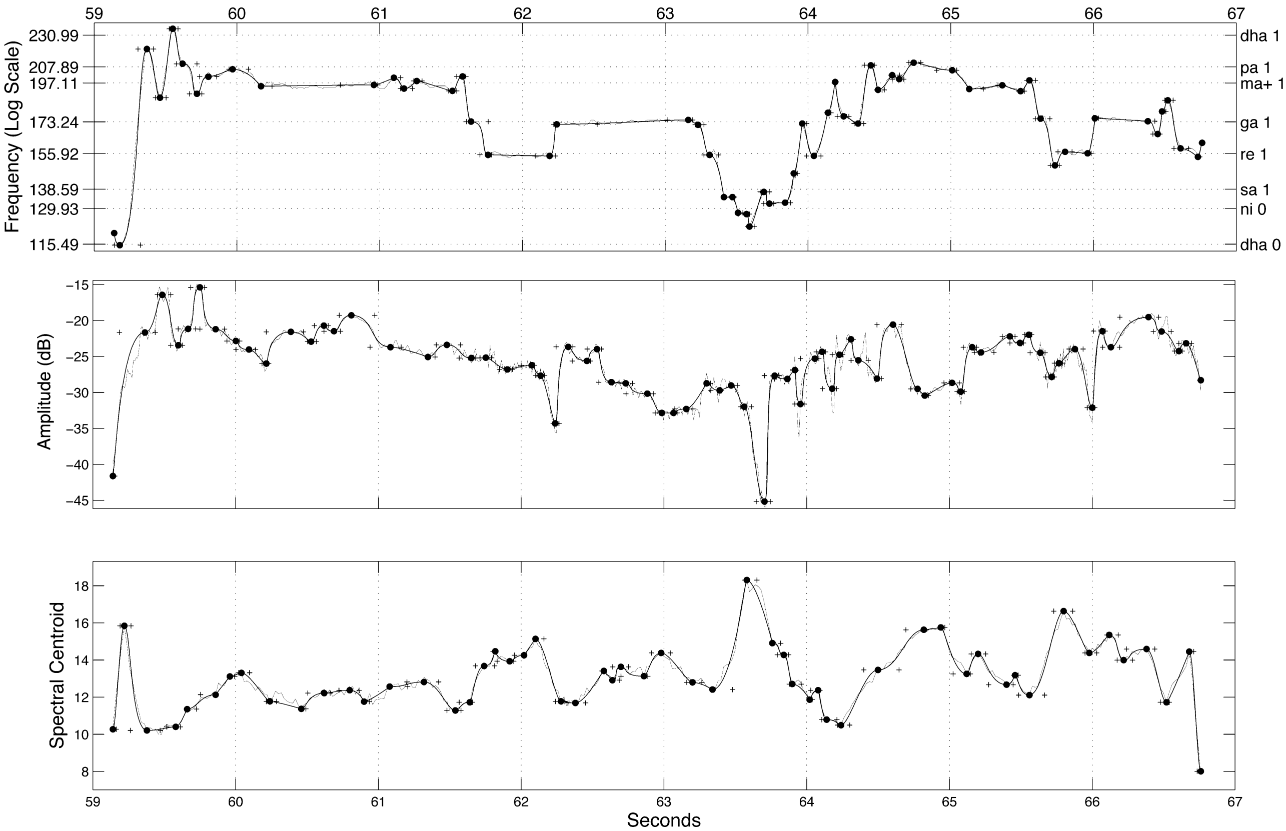
PICACS provides functions for rendering the pitch, amplitude, and spectral centroid bezlists into breakpoint envelopes for synthesis. The amplitude and spectral centroid envelopes are rescaled from a range between a user-specified floor and ceiling to a range from 0 to 1. The envelopes can be altered exponentially, offset, or scaled, after which they are cropped to remain within the 0 to 1 range. The data can then be rescaled to fit the parameter needs of a given synthesis instrument.

Demonstration

The DVD accompanying this issue contains tracks demonstrating the PICACS modeling process. Track 1 is a sung phrase from an *alap* in *Raag Yaman*. Figure 5 depicts the PICACS model of this phrase superimposed on the extracted expression curves.

Track 2 was synthesized directly from the extracted expression curves by means of an FM Violin instrument of three modulators and one carrier designed by Bill Schottstaedt and implemented in

Figure 5. Pitch, amplitude, and spectral centroid curves and Bézier models of a phrase from an alap *in* Raag Yaman*. Gray lines are the original data, and black lines are the Bézier model. Bezlist nodes appear as black dots. Plus signs ("+") indicate positions of inner control points.*

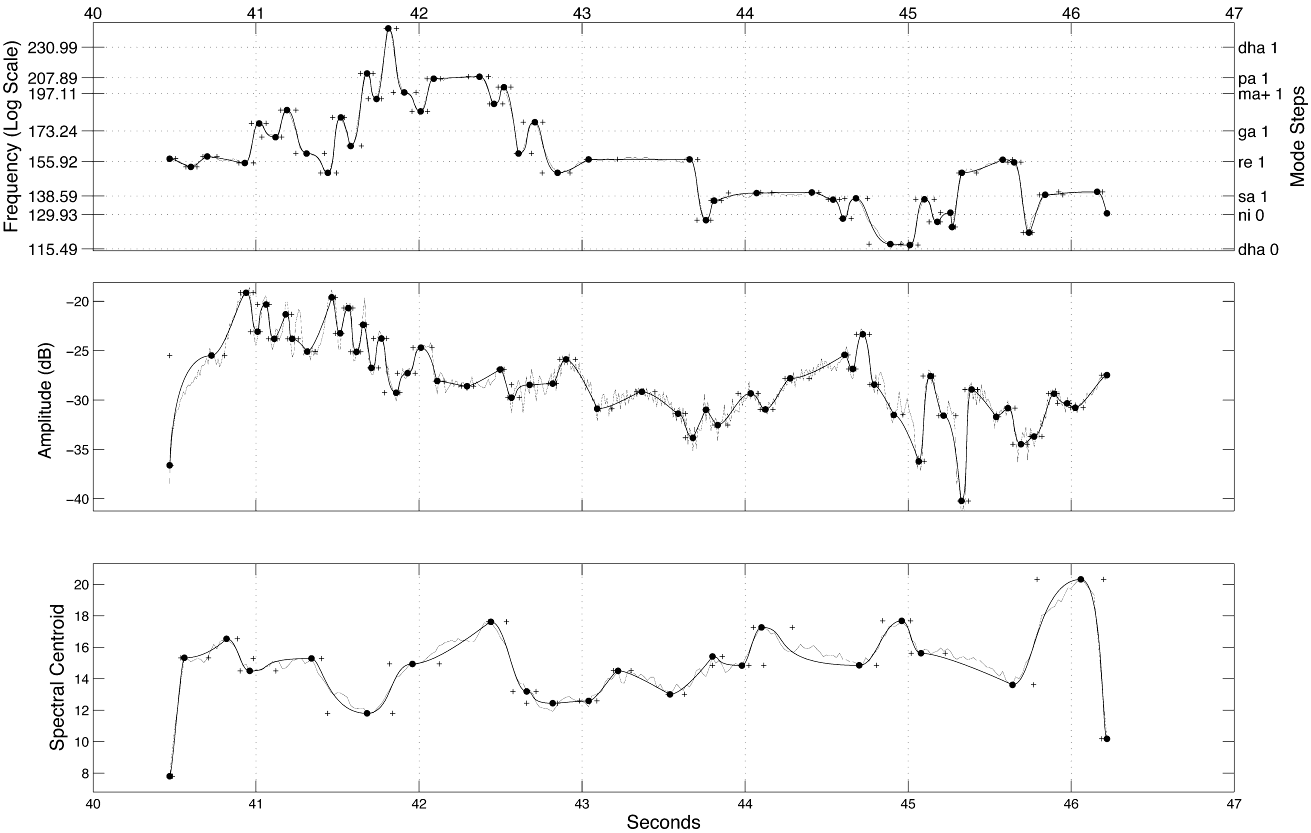


CLM. The spectral centroid scalar range was raised to the power of 0.25 and mapped to an FM-index range of 0 to 1 for the first two of the three FM carriers. Track 3 was synthesized with the same instrument, but from the Bézier-modeled data. Upon close listening, it seems that the model did not pick up all of the audible detail of the vibrato on first long-held note (*tivra Ma*). After re-analyzing the phrase with the JND range scaled by 0.65, Track 4 was rendered, providing a more satisfactory match of the vibrato.

In Track 5, the model is rendered with a richer sound design. It is worthwhile to describe this instrument, because it demonstrates useful strategies for utilizing PICACS data to solve synthesis problems. To provide the needed brightness variation, a sample of a plucked mountain dulcimer string was selected. The amplitude envelope of the sample was extracted, inverted, and applied to the sample such that the amplitude decay of the string was negated. The result was a sample of several seconds of constant amplitude but decaying high frequencies. This sample was played back utilizing GRANI, a granular synthesis instrument by Fernando Lopez-Lezcano and Juan Pampin implemented in CLM. It is important that a granular synthesis instrument continuously bend each individual grain to conform to the overall pitch envelope, or important details in curves generated by PICACS are likely to be obscured (though this can in itself be an interesting artifact). GRANI was altered to provide this ability.

Grain duration and density were controlled by mapping the desired ranges with an envelope expressing a normalized, absolute-value, first-derivative of the frequency envelope. This allows longer grains to be used during pitch plateaus where the risk of an overtly granular sound is high.

Figure 6. Pitch, amplitude, and spectral centroid curves and Bézier models of a second phrase from an alap *in* Raag Yaman.



Grain duration is shortened as the slope of the pitch envelope increases, thereby maintaining the clarity of detail in the envelope. At the same time, grain density increases to compensate for the shorter grain length. The spectral centroid scalar envelope was raised to the power of 0.75 to emphasize the brightness, and it was then mapped to the duration of the sample such that the grain time-pointer moved to the beginning of the sample for high brightness and toward the end for lower brightness.

The CLM CONTRAST unit generator was applied to the result, mapped by the spectral centroid envelope to synthesize additional high-frequency components; the result was amplitude-balanced against the original granulated sound. Finally, a Butterworth low-pass filter was applied to the sound, with the spectral centroid range mapped to a cutoff frequency range of 10 Hz to 10 kHz.

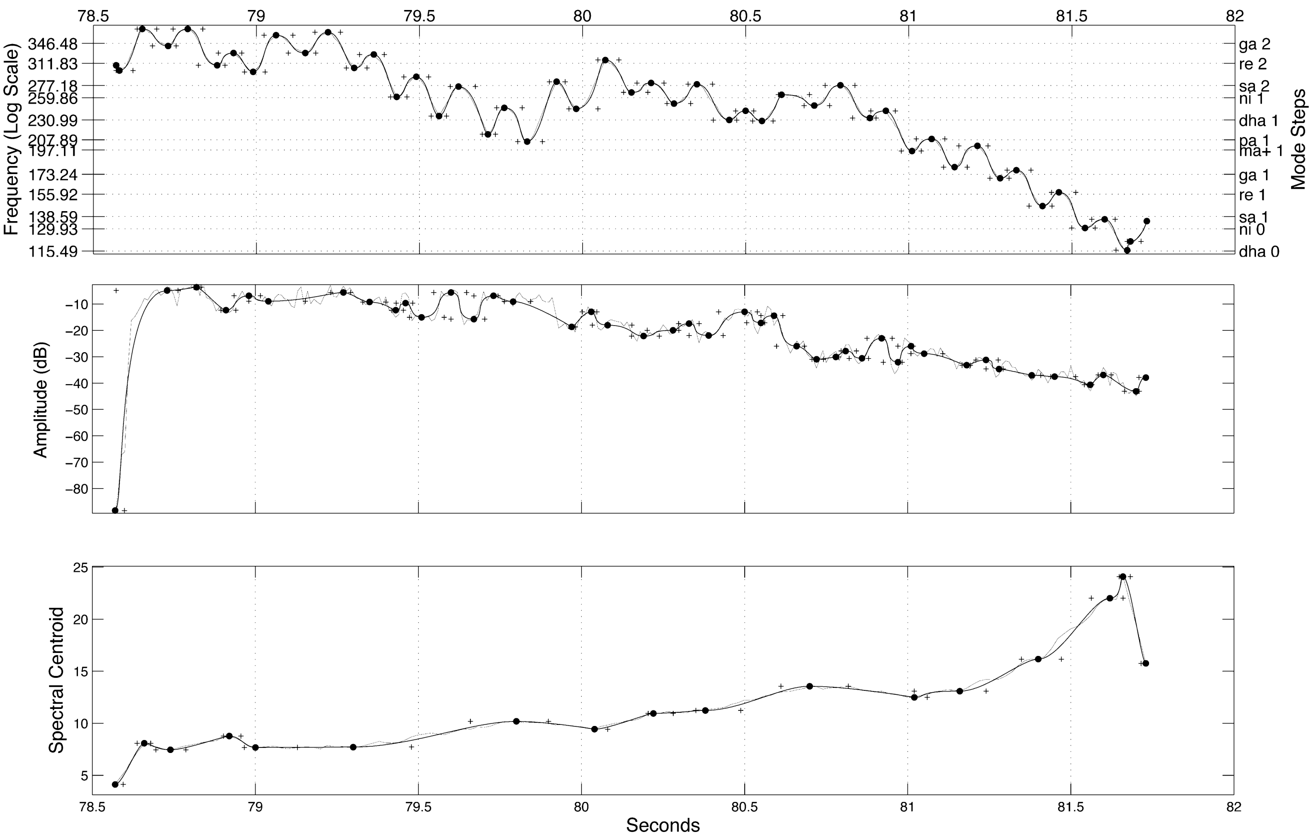
Track 6 presents the original sung phrase mixed with the modeled version. Track 7 is another sung phrase from an *alap* in *Raag Yaman*; Track 8 presents a synthesized rendition of the PICACS model. Figure 6 depicts the PICACS model.

Track 9 presents a vocal *taan* in *Raag Yaman*, and Track 10 presents a synthesized rendition of the PICACS model. Figure 7 depicts the PICACS model.

Library and Editing Functions

The PICACS library functions allow a pitch Bézier segment or set of segments to be selected and stored—along with the associated amplitude and spectral centroid data and segments—for later retrieval. The editing functions facilitate the construction of melodies through manual creation of

Figure 7. Pitch, amplitude, and spectral centroid curves and Bézier models of a vocal taan.



Bézier segments, through direct copying of analysis data into the construction editor, and through the insertion and manipulation of library entries. Specific challenges arise in designing an approach to edit the data, but discussion of these issues and their solutions lies outside the scope of this article.

Demonstration

Track 11 on the accompanying DVD presents a short demonstration realized by the author in which he used PICACS to process and edit his own improvisational singing. This was not an attempt to synthesize Indian classical music or express a *raag*. Rather, it was a free-form etude that the author created to exercise the editing tools and to provide an incremental step towards abstracting and integrating his *khyal* experience with his own compositional style and melodic thinking. The sound was rendered in CLM utilizing a processing chain similar to that described for the previous sound examples. The sound source was a blown bassoon harmonic. The drone uses the same data as the melody but with all pitch nodes set to *Sa*. The result is a dynamic drone, as if a single instrument were playing the drone and melody.

Musicological Applications

Because it combines identification of critical points with a simple and intuitive two-dimensional designation of the curve between those points, Bézier modeling should provide an effective foundation for seeking quantitative generalizations about pitch curves and ornaments. It should be made clear, however, that PICACS only models localized phenomena and does not relate them to higher-level structure. Some musicological purposes may be achieved with this model alone. Others, such as stylistic generalization and attempts to relate ornamental shape to melodic function, would require that the PICACS model be integrated with some other higher-level modeling of musical structure.

In the case of Hindustani or Carnatic music, a researcher could compare the stylistic variations among different *gharanas* (musical lineages) performing the same composition. The role of curve shapes in creating certain affects, creating and releasing tension, or in emphasizing the specific characteristics of a *raag* could be more easily investigated. The treatment of microtonal intonation in the context of specific gestures could be considered. Psychoacoustic investigations into pitch-JND discrimination between cultures and in the context of pitch-continuum melodies could be pursued. Automatic transcription—a task entailing distinct challenges in pitch-continuum traditions—could be more easily addressed.

The musicological research could serve the compositional task by creating statistical models of gestures or ornaments. Functions could be created with the information such that the composer could request that the editing environment add an ornament to a certain note or to apply a textural treatment—such as a *taan*—to a range of notes. The program would then create the expression curves, randomizing parameters within the statistical range specified by the model.

Issues and Future Work

PICACS remains prototype software, running only on Mac OS X using OpenMCL, Common Music, Common Lisp Music, and Gnuplot. The author's current priority is using it compositionally, reworking, expanding, and refining the editing feature set based on specific needs surfacing in the compositional process. The interface is currently commandline based, but clearly a graphical interface would be ideal for viewing and optimizing analysis parameters and editing the data. Given a demonstration of significant compositional value from PICACS, the author may pursue development of a graphic implementation for public release.

PICACS and the Bézier-modeling process raise a number of questions regarding music perception, particularly regarding the level of just-noticeable detail in pitch, amplitude, and spectral centroid curves. Though the heuristic methods described in this article probably suffice for many purposes, it seems that they could also be used in developing psychoacoustic testing regarding human perception of detail in pitch-continuum musics. Such an analysis could perhaps fold back into the system, providing clues toward the creation of a less heuristic approach to critical-point identification.

Conclusions

Bézier spline modeling of continuous expression data provides a viable means for addressing the challenge of manipulating and rendering pitch-continuum melodic materials, whether for analytical or compositional purposes. Though conceived initially in the context of Hindustani classical music, the modeling process clearly has much broader value and could be applied to other musical traditions.

Bézier splines, which are in themselves easy to generate, highly lend themselves to the construction of expression curves in computer music applications, even in ground-up composing contexts. In that light, one would hope that Bézier splines would become as commonly implemented in computer music applications as they are in professional graphics and animation packages.

Acknowledgments

A Fulbright Fellowship, a program of the United States Department of State, made PICACS research and development in India possible. The United States Education Foundation in India (USEFI) administers the India Fulbright. The project was hosted by the Computer Science Department of the University of Pune, Maharashtra, Dr. Sharad Gore, Director.

The work would not have been possible without the generous and patient *khyal* vocal and theory instruction of Dr. Vikas Kashalkar and Dr. R. G. Chinchanikar. I extend special thanks to Dr. Jayant Kirtane of the University of Pune Computer Science Department for support, knowledge, critique, and enthusiasm. Thanks also to Dr. Donna Calhoun of the Applied Mathematics Department of the University of Washington for her time and assistance in addressing the challenges of Bézier curve fitting.

Thanks also to Dr. Ramesh Gangolli of the University of Washington, Dr. Hari Sahasrabuddhe of the Indian Institute of Technology, Mumbai, Dr. James Beauchamp of the University of Illinois, and Dr. Richard Karpen and Dr. Juan Pampin of the Center for Digital Arts and Experimental Media at the University of Washington.

A residency at the MacDowell Colony in Peterborough, New Hampshire, facilitated development and application of the compositional components of PICACS.

References

Bartels, R. H., and D. R. Warn. 1994. ''Experiments with Curvature-Continuous Patch-Boundary Fitting.'' *IEEE Computer Graphics and Applications* 14(5):64–73.

Beauchamp, J. W. 1982. ''Synthesis by Spectral Amplitude and 'Brightness' Matching of Analyzed Musical Sounds.'' *Journal of the Audio Engineering Society* 30(6):396–406.

Daniélou, A. 1980. *The Ragas of North Indian Music*. New Delhi: Munshiram Hanoharlal.

Farin, G. 1993. *Curves and Surfaces for Computer Aided Geometric Design*. San Diego: Academic Press.

Foley, J., et al. 1990. *Computer Graphics: Principles and Practice*, 2nd ed. New York: Addison-Wesley.

Gautam, M. R. 1981. ''Gamakas of Hindustani Music.'' *The Journal of the Music Academy of Madras* 52:202–209.

Gill, P. E., W. Murray, and M. H. Wright. 1981. *Practical Optimization*. New York: Academic Press.

Gjerdingen, R. O. 1988. ''Shape and Motion in the Microstructure of Song.'' *Music Perception* 6(1):35–64.

Hajda, J. 1996. ''A New Model for Segmenting the Envelope of Musical Signals: The Relative Salience of Steady State.'' Audio Engineering Society Preprint #4391, 101st Convention. New York: Audio Engineering Society.

Jehan, T., and B. Schoner. 2001. *An Audio-Driven, Spectral Analysis-Based, Perceptual Synthesis Engine.* Paper presented at the 110th Audio Engineering Society Convention, 12–15 May, Amsterdam.

Johnson, L. W., R. D. Riess, and J. T. Arnold. 2002. *Introduction to Linear Algebra*, 5th ed. New York: Addison-Wesley.

Lee, E. T. Y. 1989. ''Choosing Nodes in Parametric Curve Interpolation.'' *Computer Aided Design* 21(6):363–370.

McAdams, S., J. W. Beauchamp, and S. Meneguzzi. 1999. ''Discrimination of Musical Instrument Sounds Resynthesized with Simplified Spectrotemporal Parameters.'' *Journal of the Acoustical Society of America* 105(2):882–897.

Rabiner, L. R. 1977. ''On the Use of Autocorrelation Analysis for Pitch Detection.'' *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25(1):24–33.

Roederer, J. C. 1979. *Introduction to the Physics and Psychophysics of Music*, 2nd ed. New York: Springer-Verlag.

Wishart, T. 1996. *On Sonic Art*. Amsterdam: Harwood Academic Publishers.