Performer Adaptive Scores: An Introduction

Robert J. Frank, D.M.A.
Division of Music, Meadows School of the Arts
Southern Methodist University
Dallas, TX 75275
robfrank@smu.edu

Abstract

This paper presents a new software tool written in the Max/MSP environment for the creation and performance of interactive computer music: the Performer Adaptive Score. A Performer Adaptive Score (PAS) is an interactive form of music for a solo instrument or voice and real-time computer that bridges the gap between fixed – or "pre-composed" – music and improvisation. It is a flexible system that allows composers to create works that retain the musical content and overall shape they desire, yet allows each individual performer to intuitively control the flow and substructure within these constraints with little to no technical effort or knowledge. This is accomplished through a matrix of pre-composed cells containing musical gestures with coordinated real-time audio effects processing. The performer's interpretations are read through the audio input and processed through pitch-tracking, amplitude measurement, and timing algorithms. These measurements are used by the program to determine cell ordering, dynamic and tempo markings, and cell timing. Music notation is displayed on-screen and all audio effects and display functions are automated in real-time.

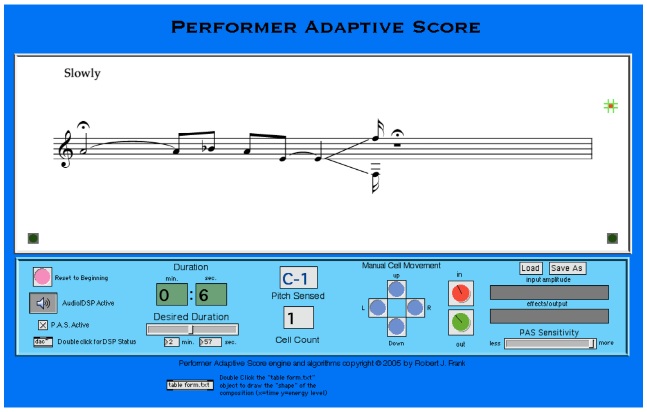
1 Introduction

Nearly every composer has had the experience of "tinkering" with a composition written for a fine performer in response to something the performer did in performance or rehearsal. Other times, performers or composers may have wished the composition was a bit longer or shorter to better fit a program or some other requirement. Creating sections or entire compositions with improvisation can achieve these results, but many, if not most, classically trained musicians have little experience or desire to improvise. This also distances the composer's intent from the final performance. Another solution to this problem is use of an "Open Form" (made popular in the compositions of Earl Brown, Lucas Foss, and Larry Austin among others.) These works allow performers to restructure a composition without true improvisation, but require conscious – rather than intuitive – choices from the performers and also do not allow for precise application of real-time audio processing effects linked to specific musical gestures.

In most of the existing research in computer music, the role of the computer has been primarily to analyze audio of pre-existing compositions in terms of phrase, pitch, harmony, tonality, etc.; to create new musical material (composition) primarily in aural form; or to create more expressive musical aural renderings of notated/MIDI works. My own prior research has focused on creating algorithmic pre-performance, compositional tools and real-time interactive works. With each successive work, I found myself moving toward a model in which the composer retained authorship of the musical content while allowing for customization of the composition to the unique talents of each individual performer (via brief improvisational sections, algorithmic accompaniment based on pitch recognition, etc.) However, the final limiting factor was the paper score, which always remained in a fixed form. If a truly adaptive work was to be developed, the score itself had to be dynamic as well. Unlike score following algorithms, this type of dynamic score needed to react to performance nuances and make compositional decisions in keeping with the composer's wishes for the work.

This led to my concept of the Performer Adaptive Score (PAS) and the development of the PAS Engine, a Max/MSP patch written by the author to create and realize a PAS composition. The PAS Engine was designed to be open and customizable enough to allow composers maximum flexibility when creating a PAS, while requiring minimal to no programming expertise on their part. The user interface was designed with the performer in mind, so that operation and use in rehearsal and performance requires only a microphone, stereo output, intuitive, on-screen sliders and knobs, and a simple drawing tool that uses proven intuitive concepts of formal representation. All pitch-tracking and audio analysis is accomplished via the audio input, so no MIDI devices are needed.

In keeping with my goals, design choices were made regarding the structure of the PAS Engine. By using external data files for the score and audio effects assignments, composers are able to create Performer Adaptive Scores without any programming using only the free, runtime version of Max/MSP. More advanced users, however, may create audio effects units and easily add them to the engine using the full version of Max/MSP. Within this limitation, every attempt was made to make the internal structure of each PAS Composition as flexible as possible, allowing this tool to be useful to as wide of a composer base as possible.



Performer Adaptive Score engine and algorithms copyright © 2005 by Robert J. Frank

Double Click the "table form.txt" object to draw the "shape" of the composition (x=time y=energy level)

Figure 1: The Performer Adaptive Score User Interface

A Performer Adaptive Score allows a composer to create a matrix of cells comprised of musical gestures of any type from traditional to graphic notation. Each of these cells is displayed one at a time on the computer's display in an ordering determined according to algorithms including both the composer's settings and a performer's interpretation in real-time. By measuring a performer's pitch, amplitude, and tempo and comparing these results to the composer's pre-set median values for these parameters, the PAS Engine works its way around the matrix according to pre-programmed parameters and an overall shape of the work drawn by either the composer or the performer. The body of the matrix may contain up to 99 cells and may be arranged in any number of rows and columns within that constraint. Composers may also include up to 50 cells as an introduction and/or up to 50 cells as a coda to help create a consistent beginning and/or ending to a work for a maximum of 199 cells total. Once the introductory cells are played in sequence, the PAS Engine proceeds to a specified starting cell within the matrix. From that point until the intended duration (minus the approximate duration of the coda) it moves within the matrix according to the performer's interpretations and the composer's settings. At the appropriate time, the PAS Engine proceeds to the coda cells and proceeds sequentially. Example 2 shows one possible configuration, consisting of a 3 cell Introduction, 10 by 3 matrix beginning with cell #15 and 2 cell Coda.

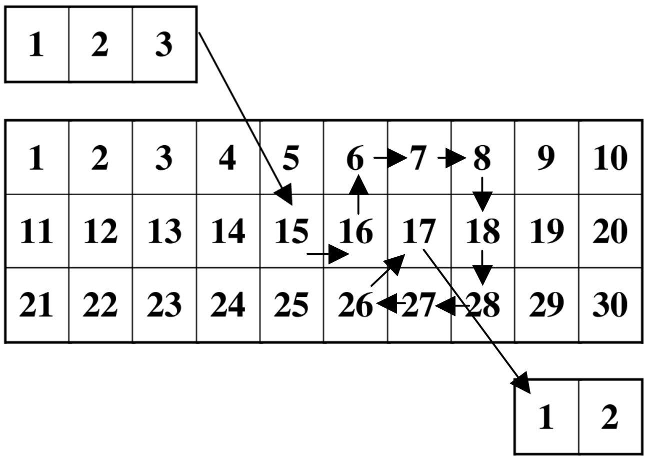


Figure 2: Matrix with Introduction and Coda

2 PAS Engine Control Algorithms

2.1 Overview

The PAS Engine makes its decisions based on some basic principles of how music flows. When music gets louder or faster, the listener may perceive a sense of increasing energy, activation, or arousal (McMullen, 1996; Radocy & Boyle, 2003). Conversely, as music slows and/or softens, there may be a sense of decreasing energy (Bartlett, 1996; Radocy & Boyle, 2003). The PAS Engine measures the average performance amplitude and the duration until a specified moment (a triggering point) in a cell and compares this to baseline values specified by the composer. It then measures the deviation from the intended overall effect and makes decisions about cell movement. It also uses this data to provide the performer with on-screen dynamic and tempo markings to help guide him or her in making interpretive decisions that reflects the composer's intent while allowing a certain level of artistic freedom.

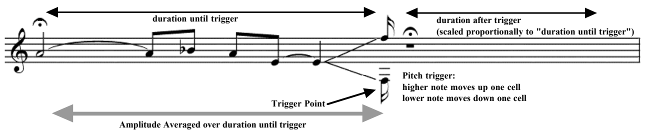

duration until trigger
duration after trigger
(scaled proportionally to "duration until trigger")
Pitch trigger:
higher note moves up one cell
lower note moves down one cell
Trigger Point
Amplitude Averaged over duration until trigger

Figure 3: Measures of pacing and dynamic levels of a musical gesture

The decision of which cell will follow each gesture is based on the above process and specific trigger(s) selected by the composer for each cell. For instance, in the above example (see Figure 3) the pitch triggers could be set so that if the performer chose the upper note the following cell would be up one row in the matrix, and if he or she chose the lower note it would be one row lower. If the duration until the trigger point was significantly less than the composer's pre-set value, the PAS Engine would move one cell to the right. Longer durations (indicating a slower pacing) would result in a movement one cell to the left. If desired, movement could be based on the relative average amplitude the performer played. The composer individually specifies the type of trigger for each cell and the resulting movement within the matrix that results for each trigger. Multiple triggerings can be allowed to provide movement based on a cumulative total of movement from multiple triggers. Standard triggers are:

- Trigger #1: At or below a given pitch
- Trigger #2: At or above a given pitch
- Trigger #3: Above a given amplitude
- Trigger #4: On a given pitch

Once triggered, the PAS Engine maintains a balanced phrase for each cell by proportionally scaling the remaining time after trigger to reflect the relative time it took to get to that point. Red and green indicator lights on the display signal to the performer when triggers are activated and when the next cell is about to be displayed.

By structuring the matrix so that there is a logical progression of range, dynamic level, activity, and flexibility from one side to the other, and by combining this with trigger movement settings that fit this structure, performers will be able to "press ahead" to more active cells, or "pull back" to less active materials, soar to higher passages or drop down to lower registers easily and intuitively. The preceding example might use pitch triggers that move up for higher notes, down for lower, left for softer and right for louder interpretations. When the composer chooses cells that reflect this movement, the result is a work in which the performer controls a specific ordering and flow of musical events that remain within the composer's desired intent.

2.2 Form Control and The PAS Value

The heart of the PAS Engine is its ability to evaluate a performer's interpretation of a given cell and make intelligent, musical decisions about which cell to display next, while also helping to guide the performer along a desired formal structure. This requires factoring together a number of parameters including the previously discussed triggers, pacing, dynamic level, and the overall shape a composer or performer may desire for a work.

An intuitive manner of expressing the shape of a work is in terms of an X-Y graph, with time being the X-axis and the perceived emotional response to the music's "energy level" being the Y-axis (Madsen, 1997). The energy level may be a dynamic level, pace at which a gesture is performed, or combination of both these factors. The PAS Engine uses a simple graphic tool to allow the composer or performer to draw the desired shape of the composition (See Figure 4.) Thinking graphically, it is easy to see where the peaks of a composition may be and where these peaks occur within the overall duration of the composition. Max/MSP has an object that allows graphic editing of a data set called *table*. Double clicking the *table* object opens a graphic tool that allows this means of manipulating the 127 values in the file **form.txt**. The X-axis is scaled over the desired duration of the work (less the Introduction and Coda durations, if present) and an energy level between 1 and 127 (default for this object) is derived for each moment of the work. Beginning and ending values in this table are used to determine dynamic and tempo levels for the Introduction and Coda sections.

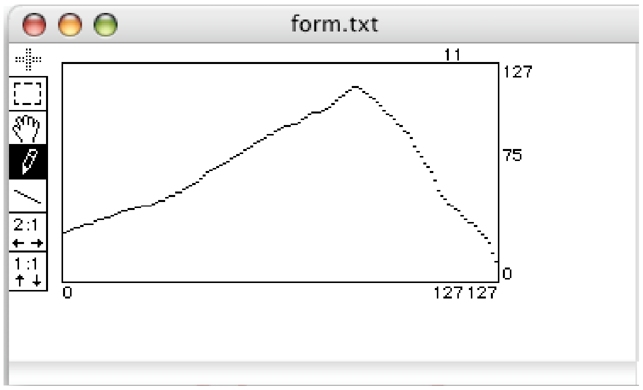


Figure 4: Drawing the "Energy Level" of a composition in the PAS Engine

One challenge to assigning a numeric value to an intuitive "energy level" for any given passage within a live performance is in determining the mutual influence of amplitude and relative tempo upon each other. By measuring both factors and scaling them in a user-definable manner, the PAS Engine is able to more humanly recognize when a performer is "pressing ahead" – which, for example, may come in the form of a slightly lower dynamic level but a much faster pace. Pace is determined by a *line* object moving from 200-0 over a duration of twice the pre-determined "intended duration" for the cell. The *line* object begins on the first sensed sound until the trigger is sensed, when it sends the *line* object's value (correlating to a percentage of the intended duration) to the analysis algorithms. (See Figure 5)

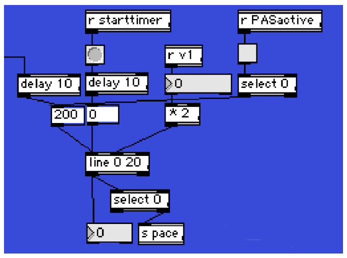


Figure 5: PAS Timing and Pacing Algorithm

The average amplitude and the measured duration until trigger are combined in an algorithm that weights and balances the congruence of the real-time performance to the baseline values assigned in the data for each cell and results in a "PAS Value" between 1 and 200. Values lower than 100 indicate the performance was below the intended energy level for that gesture, and values above 100 occur when dynamics and/or pacing are above intended values.

The PAS Value algorithm (see Figure 6) averages the ratio between intended verses actual average amplitude (expressed as a percentage) and the ratio between intended verses actual duration until trigger (expressed as a percentage from the above described *line* object.) One half of the difference between the actual percentage and 100% is multiplied by the PAS Sensitivity setting and added to the actual value before averaging, thus minimizing or exaggerating their variance proportionally.

$$\textbf{PAS Value} = \frac{(P+((100-P)/2)*S) + (A+((100-A)/2)*S)}{2}$$

P = Pace: 0 to 200 (%) of target
A = Average Amplitude: 0 to 200 (%) of target
S = Sensitivity: -.99 to +1.5

Figure 6: PAS Value Algorithm

At a Sensitivity (S) value of 0, no scaling takes place. As S increases above zero, the amount of variance from the centerline value of 100 is reduced up to 75%. S values below zero result in negative values which when summed with the actual variance results in a variance up to 150% of the actual amount. The result of this formula allows performers to adjust the reaction of the PAS Engine to best fit their own style and interpretation. Higher sensitivity settings on the performer's display (converted into negative S values in the PAS Pacing Movement algorithm) result in exaggerated variance levels and hence, exaggerated reactions by the PAS Engine, as shown in Figure 7.

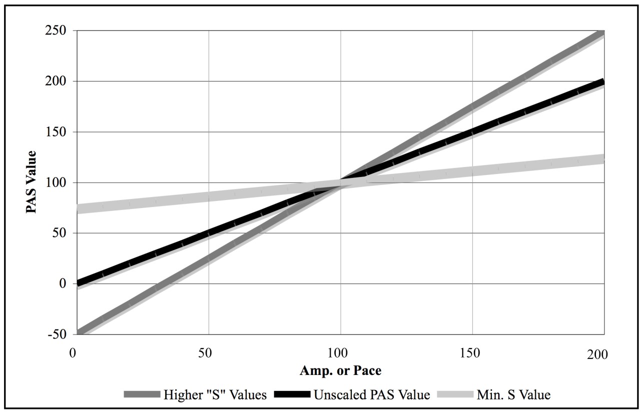


Figure 7: Effects of Sensitivity on PAS Value

Performers may adjust the level of sensitivity the PAS Engine uses to recognize these changes via a simple slider (0-250) on the main screen, which is subtracted from 150 to allow a more intuitive "left= lower/right=higher" response. Lower slider settings require (or allow, depending on how you see it) the performer to make more drastic and dramatic variances from the median values for each cell to elicit a response from the PAS Engine, while higher slider settings will cause greater variance in the PAS Value, resulting in a more sensitive response from the PAS Engine. By scaling the reactions before averaging, the PAS Engine can give greater or less weight to the parameter that is farther from 100 and result in greater or lesser variance in the PAS Value due to that parameter.

PAS Values of varying more than 25 from the centerline value of 100 are then sent to a cell movement trigger algorithm that selects subsequent cells based on composer-determined settings. For example, a PAS Value below 75 could be set to result in a movement left one cell (to cells of lower activity) and PAS Values above 125 could be set to result in a movement right one cell (to cells of higher activity). Composers may set the lower/higher movement response independently to any of four movements: up, down, left, or right to reflect the structure of their matrix of cells.

Once a PAS value has been determined, it is used to help determine the dynamic or tempo marking displayed to the performer. The ratios of the actual to intended amplitude and actual to intended pace are first scaled by the desired energy level as determined by the value returned from the *table* object for the file **form.txt**. These scaled values are then sent through a filter which blocks sending changes for which there is not a corresponding change in the PAS Value. Values making it through the filter send general JPG file formatted markings to the display (i.e. "slower", "faster", "softer", "louder", etc.) In this manner, the program does not attempt to micro-manage a performer's interpretation. If he or she is slightly louder than intended, but their pacing is proportionally slower, the lower PAS Value will take this into account and no corrective markings will be displayed. However, if both the dynamic *and* PAS Values are high, a corrective marking ("softer" or "much softer" depending on the amount) will be displayed. The same process is applied independently to tempo/pacing. The goal of this process is a more comprehensive evaluation of a performer's interpretation, with corrective markings displayed only when overall energy level factors match the individual parameters. Without this combination of factors, the results of early prototypes of the PAS Engine yielded only simplistic, surface-level feedback rather than the more comprehensive, musical feedback one would expect of a live, human listener.

3 Creating a Performer Adaptive Score

3.1 Matrix Construction Strategies

As noted earlier, the performer's changes in interpretation, dynamics, register, and tempo all are analyzed in real-time and used to select subsequent cells to display. Setting the size and number of rows and columns will have a significant effect on the resulting work. For example:

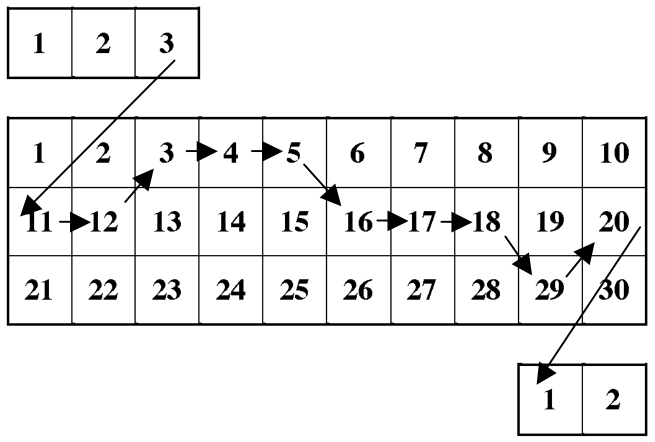


Figure 8: Linear structure for a matrix

The above matrix could be constructed to use pitch triggers to move up a row for higher passages, down a row for lower passages and a constant left-to-right movement at the end of each cell. Cell's musical material could be created as simply as transposing the musical materials in cells 11-20 up an octave for cells 1-10 and down an octave for cells 21-30. By giving the performer a choice of pitches at the trigger point in the cell, the performer has the option of performing the cell in whatever range they feel they sound best. The PAS Engine recognizes this and shifts the music for them automatically in the following cell. Should the performer feel the need to drop to a lower register due to fatigue or some other reason the PAS Engine would sense and accommodate this choice.

Figure 9 reflects a more adaptive, flexible structure. For example: let us say that the cells were composed in such a way that more energetic, higher, louder passages were used in the cells in the upper right corner of the matrix, gradually moving to lower, calmer, softer passages in the lower left corner. If the PAS Engine was set to move right for higher PAS Values and Left for lower PAS Values, and the pitch triggers were set for upward movement for higher pitch choices and downward movement for lower pitch choices, this composition would respond quite intuitively to performer responses. It could result in a wide variety of results depending on the performer's choices and the shape of the contour in the file **form.txt**.

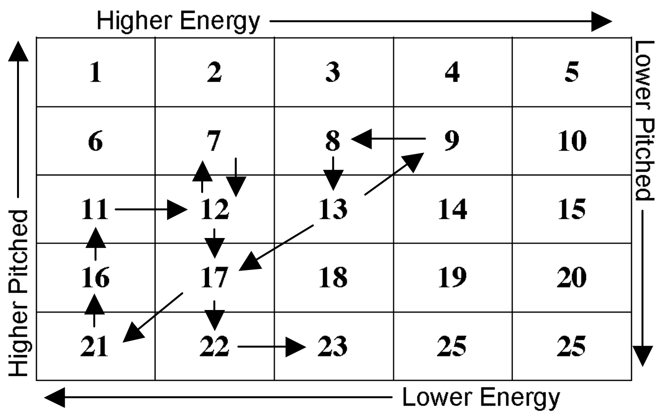


Figure 9: "Free-Walk" structure for a matrix with a low pitch-choice performance

A low pitch-choice performance may result in many combinations of cells 16, 17, 21 and 22 and no use at all of cells 4, 5, 9, and 10 for a low energy curve like the one in Figure 10. Lower sensitivity settings would result in more repetitions of cells, while higher settings would result in less repetition.

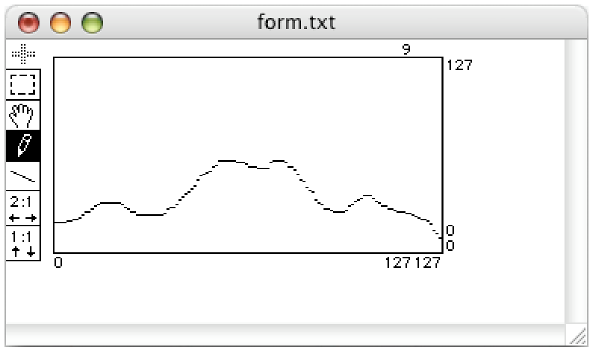


Figure 10: Low energy curve structure

On the other hand, the same matrix with a wildly varying **form.txt** curve (See Figure 11) would likely result in rapid movement throughout the matrix with little immediate repetition of cells.

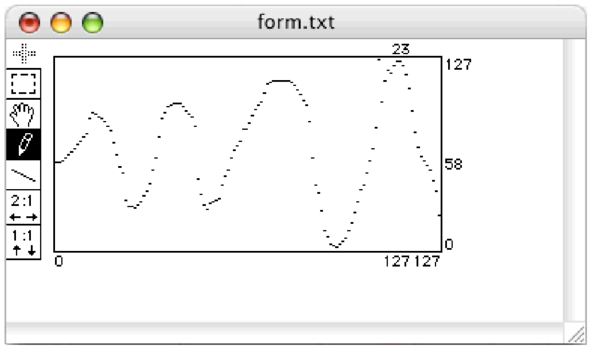


Figure 11: Varying energy curve structure

Once the types of movement within matrix structure have been clearly and logically defined, the composer may then compose cells that enhance or in some other manner utilize these characteristics. It may be helpful to think of composing adjacent cells to flow one into another or perhaps to choose only contrasting material in adjacent cells as a compositional strategy. The possibilities are endless.

3.2 Cell Creation

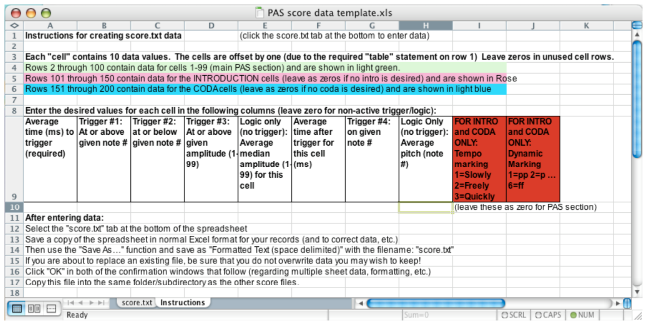
Each cell requires a trigger point to serve as a point of reference for determining the PAS value and for advancing the music to the subsequent cell automatically. Thinking about what type of trigger and at what point it will occur during the compositional process will be very helpful when creating the data files that control the score. Pitches are programmed using MIDI note numbers (with "Middle C" = 60). Composing cells that yield pleasing results at a variety of tempi and dynamic levels will most likely result in a more satisfying final realization.

Cells displayed on screen may consist of anything the composer chooses (graphics, photos, traditional notation, etc.) within the given graphic size of 950 pixels wide and 200 pixels tall. A template for Finale® 2005 or later named **PAS TEMPLATE.MUS** is included to help create cells with traditional musical notation. Composers may add or remove staves, reduce or enlarge, or alter any format as desired as long as all music for that gesture/cell fits on the given page size. It is recommended that dynamic and tempo markings be left off, as the PAS subroutine determines and displays these based on the settings in the file **form.txt** and the performer's interpretations. Once completed, each cell can be exported as a graphic file using the graphic tool in Finale®, and converted to a 950 by 200 pixel JPEG file format using any standard graphic file editor.

3.3 Data File Creation

The PAS Engine uses several external data files to store settings and score operations, thus allowing composers to create PAS Scores without the need for any Max/MSP programming. The **settings.txt** and **form.txt** files are created by the PAS Engine and do not require manual editing. The file **matrix.txt** stores size and layout of the matrix, movement instructions for triggers, and reverb mode settings. Each cell accesses two data files: **score.txt** to determine movement triggers, durations, and amplitudes, and **fx.txt** to determine the audio processing effects to apply to that cell. All of these files are in formatted text, space delimited format for use with the Max "table" object. These files must be located in the same folder/subdirectory as the PAS Engine and music cell JPEG files.

The following Microsoft Excel® templates are included with the PAS Engine and have been configured so that values can be simply entered in the cells: **PAS matrix data template.xls**, **PAS score data template.xls**, and **PAS FX data template.xls**. Composers may also use the formula and fill functions in Excel® to speed repetitive data values or create gradually varying settings across a wide number of cells. Both templates contain full instructions selected via a tab on the bottom, which describe the step-by-step process to saving the file in the proper format (Figure 12).



PAS score data template.xls

1 **Instructions for creating score.txt data** (click the score.txt tab at the bottom to enter data)

3 **Each "cell" contains 10 data values. The cells are offset by one (due to the required "table" statement on row 1) Leave zeros in unused cell rows.**

4 Rows 2 through 100 contain data for cells 1-99 (main PAS section) and are shown in light green.

5 Rows 101 through 150 contain data for the INTRODUCTION cells (leave as zeros if no intro is desired) and are shown in Rose

6 Rows 151 through 200 contain data for the CODAcells (leave as zeros if no coda is desired) and are shown in light blue

8 **Enter the desired values for each cell in the following columns (leave zero for non-active trigger/logic):**

A	B	C	D	E	F	G	H	I	J
Average time (ms) to trigger (required)	**Trigger #1: At or above given note #**	**Trigger #2: at or below given note #**	**Trigger #3: At or above given amplitude (1-99)**	**Logic only (no trigger): Average median amplitude (1-99) for this cell**	**Average time after trigger for this cell (ms)**	**Trigger #4: on given note #**	**Logic Only (no trigger): Average pitch (note #)**	**FOR INTRO and CODA ONLY: Tempo marking 1=Slowly 2=Freely 3=Quickly**	**FOR INTRO and CODA ONLY: Dynamic Marking 1=pp 2=p ... 6=ff**
								(leave these as zero for PAS section)	

11 **After entering data:**

12 Select the "score.txt" tab at the bottom of the spreadsheet

13 Save a copy of the spreadsheet in normal Excel format for your records (and to correct data, etc.)

14 Then use the "Save As…" function and save as "Formatted Text (space delimited)" with the filename: "score.txt"

15 If you are about to replace an existing file, be sure that you do not overwrite data you may wish to keep!

16 Click "OK" in both of the confirmation windows that follow (regarding multiple sheet data, formatting, etc.)

17 Copy this file into the same folder/subdirectory as the other score files.

Figure 12: PAS score data template.xls

3.4 Programming Audio Effects for Each Cell

The standard FX units included with the PAS Engine are: Reverberation, Panning Delay, Pitch Drop, Warp, and Single Delay. The **PAS FX Testing Application** is included for composers to try out these various effects and to determine what variable settings best provide the desired result. This application also allows composers to see the average amplitude level for passages, experiment with reverb settings, and try various FX processes and settings simultaneously. Each unit consists of an audio input (mono), trigger input, two variable inputs, and two audio outputs (left and right). Composers fluent in Max/MSP programming may create their own audio processing units as *poly~* objects following these specifications (complete specifications are included in the reference manual.)

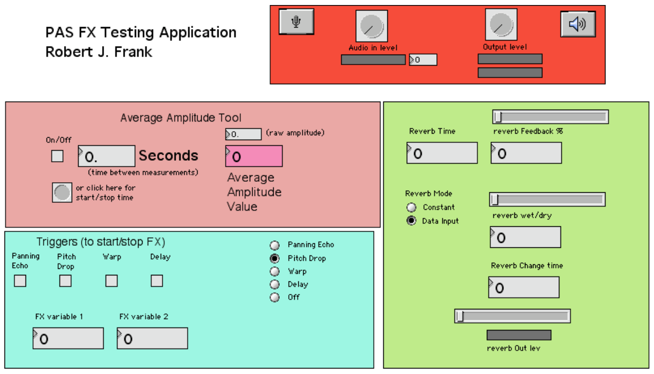
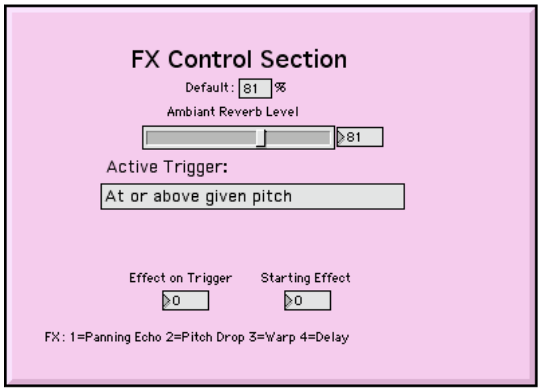


Figure 13: The PAS FX Testing Application

Within the PAS Engine, ambient reverb may be applied to the raw incoming sound to allow performers to customize the wet/dry level of their sound for any given performance space. This is set on the FX Control Section Panel in the

PAS Engine (located just below the main control panel.) Changes made to the slider will prompt the user to save settings when the program is exited. A variety of settings can be saved using the "Save Settings" button, allowing for different presets for different performance spaces. Either the composer or the performer may change this setting.


FX Control Section

Default: 81 %

Ambiant Reverb Level: 81

Active Trigger: At or above given pitch

Effect on Trigger: 0

Starting Effect: 0

FX: 1=Panning Echo 2=Pitch Drop 3=Warp 4=Delay

Figure 14: FX Control Section within the PAS Engine

Reverberation (Reverb) is applied consistently throughout a work, though the settings may be changed on a cell-by-cell basis. Each other effect may be applied at the beginning of a cell or at the trigger point. This means that up to three effects – reverb, starting FX, and trigger FX – may be sounding at the same time. The triggering point could also be used to cancel or change settings on the starting FX. It is not necessary to have a starting FX or trigger FX.

FX audio units included in this version of the PAS Engine include:

Reverb: The PAS Engine allows for the application of reverb throughout a cell with composer specified delay time (in milliseconds) and feedback amount (0-99). The PAS Engine also allows the composer to specify a time frame over which the reverb unit smoothly changes to the new settings. Reverb may also be set to a fixed level throughout a composition, in which case the PAS Engine will ignore all reverb change commands. Any changes made will also effect the ambient reverb.

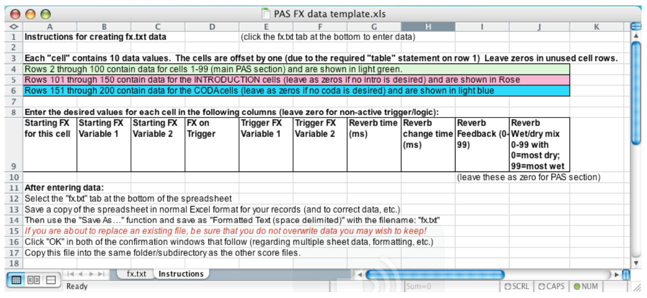
Panning Echo: This effect captures a specified amount of audio (variable #1 in milleseconds) immediately preceding the trigger point and applies a left-right panning delay to it with a feedback amount (0-99) as set by variable #2.

Pitch Drop: This effect produces a descending echo/ glissando effect on a continuous basis while active. Variable #1 controls the drop rate and variable #2 controls the depth of the effect.

Warp: This effect produces a flanger-based 'warping" of the audio input, bending the pitch and time of the echo of an event on a continuous basis. Higher range (variable #1) and amount (variable #2) values yield more dramatic effects.

Simple Delay: This effect delays the incoming audio by the amount (in milliseconds) specified in variable #1 and at the amplitude specified in variable #2. It is a mono effect, sent to both left and right channels.

Like the **score.txt** file, the **fx.txt** file is a large data file which has an Excel$^{®}$ template **PAS FX data template.xls** that greatly reduces the time and effort needed to enter data. This file contains data for each cell that specifies the audio effects processing information. As noted earlier, the **FX Testing Application** makes determining variable values easy and allows composers to audition the included effects.


PAS FX data template.xls

	A	B	C	D	E	F	G	H	I	J	K
1	**Instructions for creating fx.txt data**				(click the fx.txt tab at the bottom to enter data)						
2											
3	**Each "cell" contains 10 data values. The cells are offset by one (due to the required "table" statement on row 1) Leave zeros in unused cell rows.**										
4	Rows 2 through 100 contain data for cells 1-99 (main PAS section) and are shown in light green.										
5	Rows 101 through 150 contain data for the INTRODUCTION cells (leave as zeros if no intro is desired) and are shown in Rose										
6	Rows 151 through 200 contain data for the CODAcells (leave as zeros if no coda is desired) and are shown in light blue										
7											
8	**Enter the desired values for each cell in the following columns (leave zero for non-active trigger/logic):**										
9	**Starting FX for this cell**	**Starting FX Variable 1**	**Starting FX Variable 2**	**FX on Trigger**	**Trigger FX Variable 1**	**Trigger FX Variable 2**	**Reverb time (ms)**	**Reverb change time (ms)**	**Reverb Feedback (0-99)**	**Reverb Wet/dry mix 0-99 with 0=most dry; 99=most wet**	
10									(leave these as zero for PAS section)		
11	**After entering data:**										
12	Select the "fx.txt" tab at the bottom of the spreadsheet										
13	Save a copy of the spreadsheet in normal Excel format for your records (and to correct data, etc.)										
14	Then use the "Save As…" function and save as "Formatted Text (space delimited)" with the filename: "fx.txt"										
15	*If you are about to replace an existing file, be sure that you do not overwrite data you may wish to keep!*										
16	Click "OK" in both of the confirmation windows that follow (regarding multiple sheet data, formatting, etc.)										
17	Copy this file into the same folder/subdirectory as the other score files.										
18											

Figure 15: PAS FX data template.xls

Composers may also add title screens, copyright notices and other information on start up graphics displayed on the performer's interface. Full descriptions of all file formats and specifications are listed in the *Performer Adaptive Score Reference Manual.*

4 Current Limitations and Future Development Plans

As a first-generation prototype for dynamic, on-screen, interactive scores, and also as an entirely new genre of interactive compositions, there are inherent limitations to, and numerous future possibilities for, Performer Adaptive Scores. Currently, only a single, monophonic instrument or voice is supported and the pitch recognition ability of the *fiddle~* object is most accurate for only pitches above C3 (MIDI Note Number 48). Plans for future research and development include the exploration of supporting interactive, multiple performers via wireless networking of multiple computers, addition of record keeping and statistical decision making for more consistent performances, enhanced and expanded pitch tracking algorithms, and expansion of the library of FX processing modules. As works are created using this tool, it is hoped that new user feedback might serve to refine the user interface, score and FX structure, and ease of use for the PAS Engine. It is hoped that this tool may serve as a means for both students and experienced composers with little to no programming experience to create interactive compositions and for performers to overcome any technological fears and venture into the realm of real-time, interactive, computer music. The potential for use as a teaching tool, providing phrasing and interpretive feedback rather than new performance cell data, is also being considered as a secondary function for this tool.

5 Example: *Tic-Tac-Toe*

Tic-Tac-Toe, a short, simple composition, has been created to demonstrate the operation of the PAS environment. It consists of a one-cell Introduction, a 3-by-3 matrix and a one cell Coda. The matrix uses a free-walk structure, with vertical movement determined by higher (up) and lower (down) pitch choices and horizontal movement determined by the PAS Value (lower values move left one cell, higher values move right one cell).

6 Availability

The complete Performer Adaptive Score package, including all files listed in this paper, is available via a free download from the author via his web site (http://faculty.smu.edu/robfrank/PAS.htm).

7 Acknowledgements

The PAS Engine is written in Max/MSP® by Cycling74 software (http://www.cycling74.com) and uses the *fiddle~* object by Miller Puckette, MSP port by Ted Apel and David Zicarelli (http://crca.ucsd.edu/~tapel/software.html).

References

Bartlett, D. L. 1996. Physiological responses to music and sound stimuli. In D. A. Hodges (Ed.), *Handbook of Music Psychology* (2nd ed.), pp.343-386. San Antonio: IMR Press.

Madsen, C. K. 1997. Emotional response to music as measured by the two-dimensional *CRDI. Journal of Music Therapy* 34 (3), 187-199.

McMullen, P. T. 1996. The musical experience and affective/aesthetic responses: A theoretical framework for empirical research. In D. A. Hodges (Ed.), *Handbook of Music Psychology* (2nd ed.), pp.387-400. San Antonio: IMR Press.

Radocy, R. E., & Boyle, J. D. 2003. *Psychological Foundations of Musical Behavior (4th ed.).* Springfield, IL: Charles C. Thomas.

Rowe, Robert. *Machine Musicianship*. Cambridge, MA: MIT Press, 2001.