Dynamical Complexity Measurement with Random Projection: A Metric Optimised for Realtime Signal Processing

Chris Kiefer
Experimental Music Technologies Lab, University of Sussex
c.kiefer@sussex.ac.uk

ABSTRACT

There are many metrics available for observing the dynamical complexity of a signal, with multiple applications in computer music. Previous work demonstrated that the Effort To Compress metric could be used to modulate the behaviour of a feedback instrument, however the algorithm is challenging to run in realtime. This research explores the many metrics available, and evaluates a selection of them in their suitability for realtime signal processing in musical instruments. A new metric is proposed and evaluated: Random Projection Complexity (RPC). The results show that RPC has comparable performance to other complexity metrics, and is suited to realtime applications. The results also demonstrate viability of further complexity metrics for interactive computer music systems.

1. INTRODUCTION

Dynamical complexity metrics can reveal valuable information about time series, giving indications about structure, organisation and information content [1]. They can show the amount of regularity, disorder or randomness, in turn giving insights into the nature of the process(es) that produced the time series. Complexity metrics potentially have useful applications in instrument design, machine listening, sound processing, and sensor data processing but are relatively underused in computer music; they are not a common component of popular computer music toolkits, DAW plugins or DSP libraries. While they are already popular for signal processing in physics, neuroscience and physiological analysis, this practice has not yet transferred to computer music.

One example of the use of realtime dynamical complexity measurement in music is the *CoFlo* system [2]. It was developed to address challenges with the behaviour of feedback instruments, where the instruments move into saturation too easily due to a build up of a dominant frequency. *CoFlo* modulates the gain of the feedback instrument so that it tends not to go into saturation, instead widening out the zone before saturation, where the instrument feels most lively. The algorithm does this using a dynamical complexity metric, Effort To Compress (ETC) [3], which can detect when the system is moving towards saturation, and make gain adjustments to reduce it. It is able to detect saturation because the sound of a dominant feedback frequency has a simple structure, compared to the more complex sound of the instrument in an unsaturated state.

While *CoFlo* was shown to work effectively, there are challenges around using ETC for realtime signal processing. The ETC algorithm is computationally expensive, and needs plenty of processing headroom due to jitter in the time is takes to complete calculations. The algorithm can use an entire CPU to analyse data at a relatively small window size, rendering it inflexible in use, and precluding its applications in embedded computing within musical instruments. The *CoFlo* study did not test ETC against other algorithms, raising the question of whether other dynamical complexity metrics might also be effective in its place, and what the different qualities of these metrics might be for realtime signal processing?

A new dynamical complexity metric is proposed: Random Projection Complexity (RPC). RPC is compared to other metrics in two studies; one looking at RPC's basic ability for measuring dynamical complexity, and another comparison study using simulated audio feedback systems, similar to *CoFlo*. Finally, the realtime performance of RPC is compared to other metrics.

2. MEASURING DYNAMICAL COMPLEXITY

We should begin by asking what complexity is, and what it means in terms of sound? While there is no strict definition of complexity [1], Lloyd [4] suggests that it relates to how hard something might be to describe or create, and how organised it is. Dynamical complexity relates how to complexity of a system changes over time. There are multiple methods available for measuring this; Lau et. al review twenty different metrics, suggesting they can be divided into those that quantify either predictability or regularity in time series [5]. All of these metrics make an intuitive link from an observable measurement towards some aspect of complex dynamics.

This lack of a strict definition of complexity and diversity of measures could be seen as problematic, however different complexity metrics can be seen as ways of measuring different aspects (and possibly interacting dimensions [1]) of a system's behaviour, and one can find empirical evidence of dynamical complexity measurements correlating with the behaviour of realworld phenomena. They are used across multiple sciences as indicators of, among many other examples, levels of consciousness [6], cardiac health [7], stock market inefficiency [8], or ecosystem health (via soundscape analysis) [9].

In computer music, spectral entropy, a measure of frequency domain complexity, is used in machine listening (e.g. [10]), but there is little example of other metrics in use, especially in realtime interactive systems. Looking to their application in other fields, complexity metrics are broadly used for high-level description of system behaviour; this could have many applications in computer music, and as we have already seen, this approach has been effective in monitoring the state of feedback instruments with *CoFlo*. This leads back to the questions raised earlier; ETC was effective in observing a state of saturation in a feedback system, but also problematic to use in an interactive system because of computational demands. A new metric, Random Projection Complexity, which addresses these realtime challenges, will now be evaluated.

3. RANDOM PROJECTION COMPLEXITY

RPC is a form of compression-complexity metric (CCM). This family of metrics equates the compressibility of data to its complexity. Intuitively, simple data is easily compressible. Data with more complex and varied structures will be more challenging to compress, and at the highest end of the scale, noise will be very difficult to compress because it lacks regular patterns. ETC and Lempel-Ziv (LZ) metrics both use lossless compression algorithms to quantify complexity [3]. RPC uses random projection (RP), a technique that can be used for lossy compression, to measure complexity with improved realtime performance compared to ETC and LZ, at the cost of some random variance in the measurements. RP is founded on the Johnson-Lindenstrauss lemma [11], which states that points in a high-dimensional space $\mathbb{R}^h$ are likely to retain a similar distance from each other when projected into a lower dimensional space $\mathbb{R}^l$. With RP, data is projected into a lower-dimensional space using a matrix P with randomly generated coefficients. When coefficients are drawn from a Gaussian distribution, point-to-point distance is more likely to be preserved. The lower-dimensional projection is a compressed representation of the high-dimensional data, which can be approximately restored using an inversion of the projection matrix. RP is used in a range of applications including nearest neighbour search of audio features [12] and classification [13].

The RPC algorithm [14] takes five inputs: a time series w of length n, a window size h which is also the size of the higher dimensional space to project from, the size of the lower dimensional space l, a hop size α and histogram resolution β. The algorithm works in two stages; the first is **projection**. A Gaussian projection matrix $P^{w\times l}$ is generated, and a sliding window is moved across w. The results of projections from these windows into the lower dimensional space $\mathbb{R}^h \rightarrow \mathbb{R}^l$ are collected. The number of hops λ of the sliding window is calculated in equation 1. In equation 2, the projections are collected into matrix $Q^{l\times\lambda}$. $w_{a:b}$ indicates a slice of vector w for indexes in $[a, b)$.

$$\lambda = floor((n - w)/\alpha) + 1 \qquad (1)$$

$$Q = [Pw_{0:w}, Pw_{\alpha:w+\alpha} \dots Pw_{\lambda\alpha:w+\lambda\alpha}] \qquad (2)$$

The second stage in the calculation is **histogramming**. The rows in Q are rescaled individually between 0 and 1 (equation 3), and then the points on each column are collected into a multidimensional histogram H, with β bins in each dimension.

$$Q_i = (Q_i - min(Q_i))/max(Q_i) \qquad (3)$$

The value of RPC is calculated as the number of nonzero bins in H (equation 4).

$$RPC = count(H > 0) \qquad (4)$$

The RPC metric works on the intuition that time series with simple structure will have repeating patterns that project onto the same areas of the histogram, thereby occupying less of its area. Time series with more complex and varied structure will project onto a larger area of the histogram. This is similar to the intuition behind other CCMs described earlier. An example of this is shown with projections of the time series in figure 1; a sinewave with simple structure, a simulated cardiac ECG signal (generated with Neurokit [15]), and white noise.

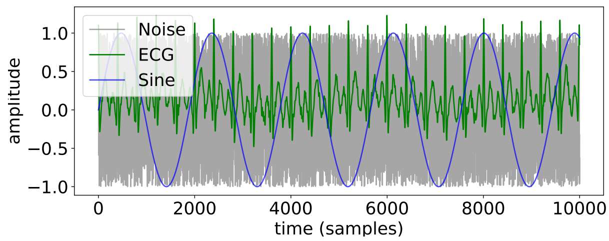

Figure 1. Time series for the demonstration of random projection: sine, simulated ECG, white noise

Figure 2 shows the nonzero bins in histograms of these waveforms, with parameters $h = 16, l = 2, \beta = 20$. Table 1 shows the value of RPC, calculated from these histograms. The ordering of the results is in line with other CCMs: noise has the highest score, and the ECG waveform, with its more varied structure, has a higher score than the sine wave.

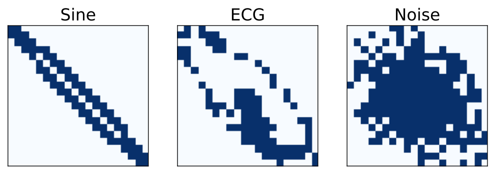

Figure 2. Sine, ECG and noise time series projected into two-dimensional histograms

	Sine	ECG	Noise
RPC	69	91	212

Table 1. RPC of time series

4. STUDY 1: COMPARISON WITH OTHER COMPLEXITY METRICS

This study compares RPC with other complexity metrics. A comparison to all other popular metrics is beyond scope, so a selection of metrics were chosen as follows:

ETC, LZ Two metrics using lossless compression, from the family of CCMs.

SE Shannon Entropy [16], a popular and widely used entropy-based metric

MSE Multiscale Entropy [17], a metric that addresses some limitations of SE

SFD Sevcik Fractal Dimension [18], a metric optimised for time series analysis

The intention of this study is not to question whether RPC is significantly different to these other metrics, but to question its efficacy in providing a similar quality of results.

4.1 Method

Following from Nagaraj and Balasubramanian's comparison of CCMs [3], the logistic map (equation 5) was chosen as a means for comparison. The logistic map produces time series that vary from periodic to chaotic behaviour as the bifurcation parameter r varies between 3.5 and 4. The Lyapunov exponent [19] λ is an excellent baseline measure of the complexity of the logistic map, because its calculation can be directly derived from its equation.

$$x_{n+1} = rx_n(1 - x_n) \tag{5}$$

Logistic maps were produced, with length 400, and with r in range $[3.5, 4]$ at intervals of 0.0005. The 6 metrics were calculated for each map, along with Lypunov exponents. SE, ETC and LZ require symbolic input, so for these metrics the time series were discretised using 256 equally spaced bins, to give a clear representation of the original signal. RPC was manually tuned to obtain a representative result, with $h = 16, l = 4, \alpha = 4, \beta = 6$. MSE requires a dimensionality parameter, this was calculated using the correlation dimension method.

4.2 Results

Metric	RPC	ETC	LZ	MSE	SE	SFD
PCC	0.892	0.910	0.911	0.759	0.891	-0.749

Table 2. Correlation coefficients for complexity metrics with λ

The results for each metric were normalised and plotted in figure 3. The plot shows that RPC offers similar results to ETC, LZ and SE. This result is confirmed by table 2, which shows how these results correlate with λ, using Pearson's correlation coefficient (PCC).

5. STUDY 2: SATURATION IN FEEDBACK SYSTEMS

Following from the *CoFlo* study, where ETC was found to be a good indicator of saturated states in feedback instruments, are RPC and the other metrics tested in section 4 good alternatives? And how do their qualities differ in this task? Study 2 explores these questions, using software feedback systems.

5.1 Method

The *CoFlo* study was a qualitative investigation of complexity analysis on feedback cellos and a halldorophone. In order to be able to effectively compare metrics in a similar task, a more controllable feedback instrument would be advantageous; for this reason, a software feedback system was used. This system was programmed in SuperCollider. It is inspired by the feedback cello, using a set of physically modelled bowed strings (using digital waveguides) which are passed through a physically modelled soundboard. The output signal is fed back through the soundboard along with the signals from the strings. This patch has a similar behaviour to the feedback cello in that, when left undisturbed, the feedback will slowly dominate the system, moving it into a stage of saturation, with a single dominant frequency. ETC could successfully detect this state of saturation, but can these other metrics achieve this too?

Fifteen recordings of the feedback system were collected, with the feedback gain manually tuned in each case to that the system moved from an initial unsaturated state into complete saturation over the course of roughly 15 seconds. In each recording, the frequencies of the strings were altered, to create a varied dataset for analysis. The first five recordings were at single frequencies, at 60Hz, 120Hz, 240Hz, 480Hz and 960Hz. The next 5 were major and minor chords at different pitches. The final five had 6 strings at randomly chosen pitches between MIDI notes 0 and 70, resulting in inharmonic pitch combinations. These pitches are noted in the accompanying source code [20], along with the SuperCollider code.

The recordings were manually annotated by the author, to mark three salient points:

P0 the moment when feedback becomes audible

P1 the point at which the saturation begins to sound louder than the original sound

P2 when the saturation dominates the system and overrides all other sound.

Automatic gain control would need to activate somewhere between P0 and P1, and reduce the system gain to prevent

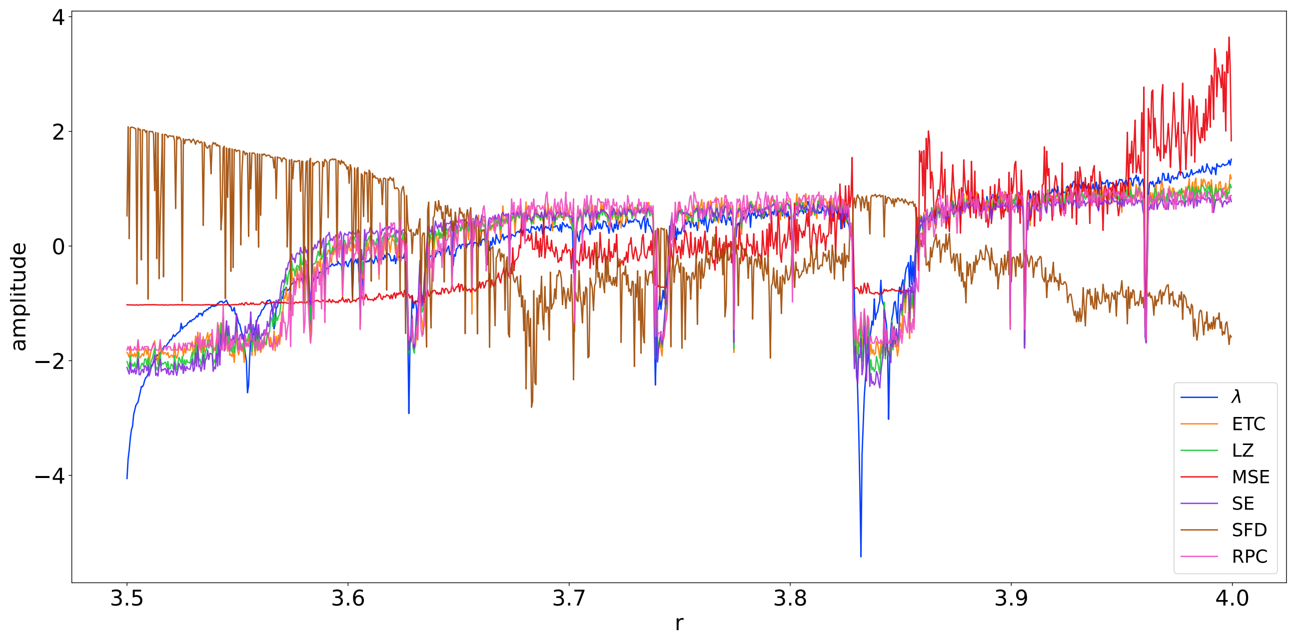

Figure 3. Comparison of complexity metrics measuring the logistic map at varying values of r

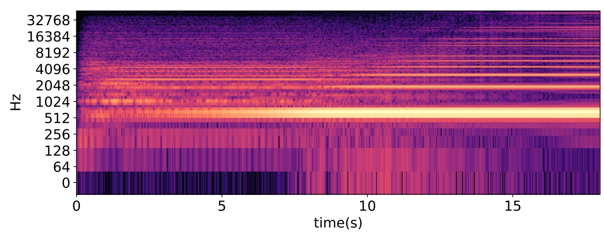

Figure 4. Example spectrogram for a feedback system recording, showing the system moving into saturation

saturation. It is acknowledged that this annotation from a single researcher is subjective, and the results are considered in the light of this. Figure 4 shows an example of a recording, where the system moves slowly into saturation with a dominant frequency around 512Hz.

Each recording was analysed with the set of metrics. RPC was configured in four variations (see table 3), to reveal how it might be affected by different parameterisations. The configurations were chosen to cover a range of projection dimensions and histogram resolutions.

Configuration	h	l	α	β
RPC0	32	1	16	100
RPC1	16	4	8	10
RPC2	64	7	32	3
RPC3	64	7	64	3

Table 3. Configurations of RPC metric used in study 2

The metrics were run on overlapping windows of data, based on the approach that a realtime complexity analysis system like CoFlo takes. The windows were 1000 samples wide, with a 500 sample hop size. This resulted in a time series of complexity measurements from each metric, for each of the 15 recordings of the feedback system.

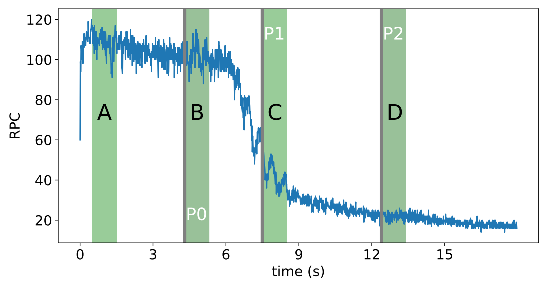

Figure 5. An example of the regions of interest and annotation points in a time series of complexity measurements of a feedback system moving into saturation (using RPC1)

To assess whether the metrics could differentiate between unsaturated and saturated sound, one second windows were identified in each time series, from regions of interest (ROIs): **[A]** a window from the beginning of the sample, representing unsaturated sound, and starting from 0.5s to give the feedback system some time to settle; windows **[B, C** and **D]** follow annotation points P1-3. Distances between the data in these ROIs were measured, to give an indication of difference in the measurements of saturated and unsaturated states. Dynamic Time Warping, which has been demonstrated as an effective metric for similarity between time series [21], was used to measure distance. Four distance measurements were made between the ROIs: AB, as a measure of being able to detect the early onset of saturation; AC, to identify when saturation becomes dominant; BC indicates the difference between early onset and dominant saturation, and CD is the difference between dominant and complete saturation. Figure 5 demonstrates an example of these annotations and ROIs, using an output from metric RPC1. Due to differences in scale between metrics, all time series were standardised to unit variance and zero mean before distances were calculated.

5.2 Results

Figure 6 shows the results. A higher distance indicates that metric would be better at differentiating between the ROIs. All metrics score well for CD. Interestingly, MSE and SFD, which were outperformed by other metrics in study 1, both perform very well, including good results for the distance AB, which identifies the early onset of feedback. RPC0 offers intruiging results; it shows low scores with little variance, although the mean is higher than LZ and SE. This lower quality could be due to the configuration of projection into a single dimension ($l = 1$). RPC1, RPC2 and RPC3 show higher average scores for tests AB, BC and AC then LZ and SE, with RPC1 and RPC2 also scoring higher than ETC. RPC3 was configured identically to RPC2 except for having a larger hop size. In this case, the effect was to reduce the quality of the measurements. It is acknowledged that the subjective annotations may create some variance in these results, but given that all metrics were calculated on the same data, the results show that RPC can be configured to perform at a similar level to other standard complexity metrics.

6. STUDY 3: PERFORMANCE BENCHMARKS

Study 2 demonstrates that all of the tested metrics were able to detect saturation in the simulated feedback system. Which of them are most suited to realtime use in musical instruments? It's clear from examination of the algorithm that MSE will struggle in a realtime system due to computational complexity, and it's known that ETC suffers form performance issues, but the other algorithms might work well. An algorithm that runs well in realtime might have low CPU demands, and low jitter, i.e. it takes a similar time to run on each window of data. It might also run with low memory overhead (this would be advantageous for running on microcontrollers) and might be able to take advantage of parallelisation.

6.1 Method

All algorithms were programmed in C++, in what the author considers to be optimal implementations. The test program used compiler optimisation at the *O3* level. Eigen library [22], known for high-performance, was used for all linear algebra functions. Source code is provided at [14]. MSE, due to expected low performance, was not tested. ETC was kept as a reference comparison to *CoFlo*. The same configurations for RPC as study 2 were used. The study measured runtime speed, but did not test memory use.

A recording of a jungle soundscape was chosen as the source data for analysis, selected for variation and complexity in the structure of the sound. For each trial, the metrics cycled through 500 sample windows of this data. All trails ran over 1000000 iterations, except for ETC which was significantly slower, and was run over 100000 iterations instead. The time taken for each analysis was recorded with a high resolution timer using the C++ *clock()* function. Tests were run on a Dell XPS 7590, with a 2.4GHz Intel i9 CPU and 32GB RAM, running Ubuntu 22.04.

6.2 Results

Table 4 shows the benchmark results, with the median time in milliseconds that the metrics took to analyse 500 sample windows of audio data. ETC, as predicted, is very slow in comparison to the RPC metrics and SFD. The fastest metrics are roughly $7300\times$ faster than ETC at this window size (although ETC's speed would improve for smaller windows), and would be approximately $5500\times$ faster than realtime for processing 44.1kHz audio. SFDs speed is due to its simplicity of calculation. RPC has a slightly more involved algorithm, but is able to take advantage of hardware optimisations for parallel vector processing that are present in many CPUs. LZ and SE are slightly slower, but still easily fast enough for realtime audio processing.

7. RPC IN USE

RPC was implemented as a SuperCollider UGen, and trialed in a *CoFlo* style scenario, to prevent the software feedback system from study 2 from going into saturation. The system was modified so that RPC analysed the audio in the feedback loop, using the same configuration as RPC1. When the output of the RPC analyser dropped below a threshold, the feedback gain was reduced proportionally to stabilise the loop. This approach worked successfully. Figure 7 shows a spectrogram example of a recording; the system does not approach saturation. At around 20s, a feedback frequency starts to emerge near 256Hz, but it is suppressed by gain modulation. The synth ran without issue in realtime.

8. CONCLUSIONS

Random Projection Complexity is a new metric in the family of complexity-compression metrics, that takes the approach of *lossy* compression, to gain speed at the cost of some random variance in the results. It was developed as a potential solution to the challenges of realtime performance for ETC in *CoFlo*, a system for modulating the behaviour of feedback instruments. Two studies demonstrate that RPC has comparable results to other existing complexity metrics, and that it can respond to saturating feedback in a software instrument, also demonstrated in practice in section 7. Performance benchmarking reveals that RPC can be tuned to run with excellent realtime performance. These factors position it as a convincing replacement for ETC in *CoFlo*.

The studies reveal some challenges with RPC. Firstly, due to the nature of random projection, there will be variance in how data appears in the low dimensional space, but it's not yet clear as to how much variance there is, and how this contributes towards the quality of the results. RPC has more parameters that the other metrics; ETC, LZ and

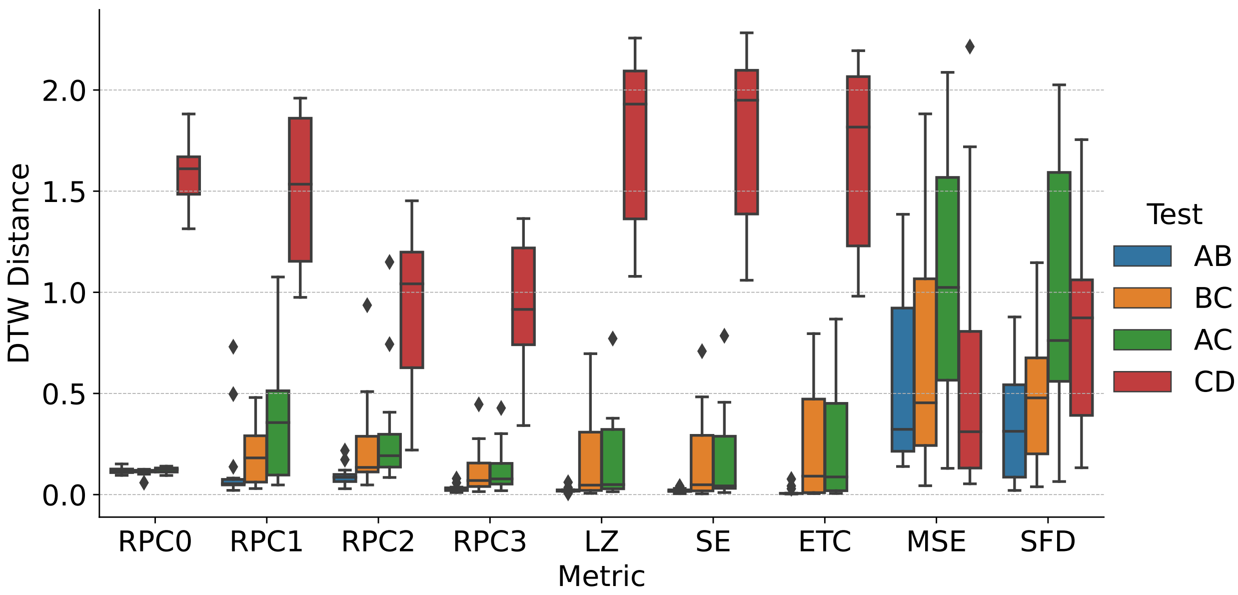

Figure 6. Study 2 results, showing difference in measurements of ROIs with varying states of saturating feedback

	RPC0	*RPC1*	*RPC2*	*RPC3*	*LZ*	*SFD*	*SE*	*ETC*
Median (ms)	**0.002**	0.008	0.004	**0.002**	0.068	**0.002**	0.015	14.742
Std	0.001977	0.001658	0.001195	0.000997	0.009068	0.000758	0.001841	0.409827

Table 4. Benchmark Results

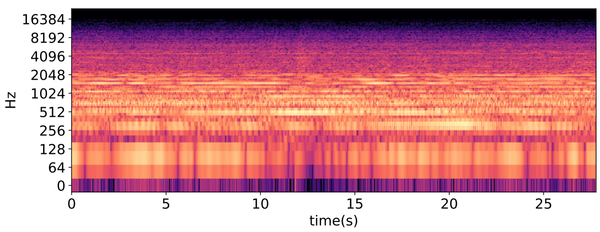

Figure 7. A spectrogram of a recording of a software feedback system, with RPC preventing the system moving into a saturated state

SE work with symbolic data and require a single choice of how the data is discretised; RPC in contrast has four parameters, which simultaneously affect both quality and performance. This means that some experimentation is required to tune RPC to a particular tasks, but also means that RPC can be flexible to balance quality and performance in different settings. The last challenge with RPC is to find a technique for normalisation of the output; the metric scales nonlinearly with the combination of h, l, and β.

The studies also offer insights into the behaviour of other complexity metrics that were tested. MSE worked fairly well with the logistic map and very well in the feedback detection task, however it's too computationally expensive for realtime applications. LZ and SE seemed to perform well across all studies, although LZ runs more slowly compared to the fastest metrics. SFD is intruiging; despite a negative result for the logistic map, it performed very well in study 3 and also exhibits excellent realtime performance.

This variance across the metrics helps to reinforce the idea that there is no *correct choice* of complexity measurement, the metric needs to be chosen to suit a particular context. These metrics offer the potential to give insights into the meta-behaviour of audio systems, and many of them are able to run in realtime, offering interesting potential in musical instrument design and computer music. RPC adds a new option to the selection of metrics available to computer musicians.

9. REFERENCES

[1] M. Mitchell, *Complexity: A guided tour*. Oxford university press, 2009.

[2] C. Kiefer, D. Overholt, and A. Eldridge, "Shaping the behaviour of feedback instruments with complexity-controlled gain dynamics," in *20th International Conference on New Interfaces for Musical Expression*, 2020, pp. 343–348.

[3] N. Nagaraj, K. Balasubramanian, and S. Dey, "A new complexity measure for time series analysis and classification," *The European Physical Journal Special Topics*, vol. 222, no. 3-4, pp. 847–860, 2013.

[4] S. Lloyd, "Measures of complexity: a nonexhaustive list," *IEEE Control Systems Magazine*, vol. 21, no. 4, pp. 7–8, 2001.

[5] Z. J. Lau, T. Pham, S. A. Chen, and D. Makowski, "Brain entropy, fractal dimensions and predictability: A review of complexity measures for eeg in healthy and neuropsychiatric populations," 2021.

[6] M. Schartner, A. Seth, Q. Noirhomme, M. Boly, M.-A. Bruno, S. Laureys, and A. Barrett, "Complexity of multi-dimensional spontaneous eeg decreases during propofol induced general anaesthesia," *PloS one*, vol. 10, no. 8, p. e0133532, 2015.

[7] M. Costa, A. L. Goldberger, and C.-K. Peng, "Multiscale entropy analysis of biological signals," *Physical review E*, vol. 71, no. 2, p. 021906, 2005.

[8] L. Zunino, M. Zanin, B. M. Tabak, D. G. Pérez, and O. A. Rosso, "Complexity-entropy causality plane: A useful approach to quantify the stock market inefficiency," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 9, pp. 1891–1901, 2010.

[9] S. Siddagangaiah, C.-F. Chen, W.-C. Hu, and A. Farina, "The dynamical complexity of seasonal soundscapes is governed by fish chorusing," *Communications Earth & Environment*, vol. 3, no. 1, p. 109, 2022.

[10] N. Collins, "Noise music information retrieval," in *Noise in and as music*. University of Huddersfield Press Huddersfield, United Kingdom, 2013, pp. 79–96.

[11] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of johnson and lindenstrauss," *Random Structures & Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.

[12] M. Casey, C. Rhodes, and M. Slaney, "Analysis of minimum distances in high-dimensional musical spaces," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 1015–1028, 2008.

[13] T. I. Cannings and R. J. Samworth, "Random-projection ensemble classification," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 79, no. 4, pp. 959–1035, 2017.

[14] C. Kiefer, "libcccrt," https://github.com/chriskiefer/libcccrt, 2023.

[15] D. Makowski, T. Pham, Z. J. Lau, J. C. Brammer, F. Lespinasse, H. Pham, C. Schölzel, and S. H. A. Chen, "NeuroKit2: A python toolbox for neurophysiological signal processing," *Behavior Research Methods*, vol. 53, no. 4, pp. 1689–1696, feb 2021. [Online]. Available: https://doi.org/10.3758%2Fs13428-020-01516-y

[16] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[17] M. Costa, A. L. Goldberger, and C.-K. Peng, "Multiscale entropy analysis of complex physiologic time series," *Physical review letters*, vol. 89, no. 6, p. 068102, 2002.

[18] C. Sevcik, "A procedure to estimate the fractal dimension of waveforms," *arXiv preprint arXiv:1003.5266*, 2010.

[19] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica D: nonlinear phenomena*, vol. 16, no. 3, pp. 285–317, 1985.

[20] C. Kiefer, "Python notebooks," https://github.com/chriskiefer/RPCSMC, 2023.

[21] R. J. Kate, "Using dynamic time warping distances as features for improved time series classification," *Data Mining and Knowledge Discovery*, vol. 30, pp. 283–312, 2016.

[22] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.