HISTORY OF COMPUTER MUSIC FROM MATHEWS TO "MAN IN THE MANGROVES"

James A. Moorer

May 2026

I was not in the first generation of computer music professionals, like Lejaren Hiller, Max Mathews, John Chowning, and many others. I do consider myself in the second generation, including Jim Beauchamp, Paul Lansky, and many, many others. I arrived at Stanford in 1968 to find an active computer music project in progress there with composers John Chowning, Loren Rush, and Leland Smith. One of the biggest barriers to making music with the computer was that an enormous amount of computation was needed to synthesize music. Chowning's discovery of FM synthesis was considered remarkable at the time because it could produce a range of musical sounds with relatively little computation. This notwithstanding, it was still a number of years before computer music could be made in real time, but that did not prevent composers from making music. Various compositional aids, such as Leland Smith's "Score" program allowed very detailed specification of the music in a way that was easily understood by composers. Note also that Smith's program allowed the use of random numbers for composition – to start notes or set parameters. Very soon an array of works had been produced that could not have been made in any other way. The precision and flexibility of computer synthesis was recognized very early as virtually the only way to go beyond a certain level of complexity and musical freedom, going beyond what unaided acoustical musical instruments could manage. The difficulty of programming complex works and building the instrument libraries assured that this way of making music would not replace trained composers or music schools.

I noticed early on that composers were having trouble navigating the complex numerical and signal processing questions that computer music implementations raised. For instance, the phenomenon of "aliasing" came as a bit of a surprise. As a trained engineer and mathematician, I immediately started trying to fill that need. I also started working with fellow graduate student John Grey who was working on psychoacoustics, which turned out to be crucial in understanding what we were hearing.

It soon became clear that making music by computer synthesis involves an extra step. In music school, we are taught to compose and orchestrate by learning the characteristics of the instruments of the orchestra. Some composers use extraordinary acoustic techniques, like *sul tasto* or *col legno* bowing. Some composers build new instruments. With pure computer music synthesis, we have to build our own orchestra. In the beginning, this was very difficult since we were starting with nothing but dreams. During the last 50 years, a great library of synthesized instruments has been made available. This notwithstanding, many times, as with the creation of "Man in the Mangroves", we still have to build the DSP framework for the sounds we wish to achieve. The library of sounds is extensive, but not exhaustive. We have many years of innovation ahead of us.

My friend and colleague, composer Pauline Oliveros (https://en.wikipedia.org/wiki/Pauline_Oliveros) once commented that "You have to build your own instrument". As an example, she was actively engaged in this with her microtonal accordion. I had to do this with "Man in the Mangroves" to bring the art and practice of speech synthesis for music up to date. This took about a year of my time.

We had the first Computer Music Conference in 1975 at the University of Michigan. We could all fit in one small classroom, but it was a wonderful opportunity to get together and discuss hardware, software, and music in general. In 1976, the Computer Music Conference was held at the University of Illinois. I was invited to give the keynote address – the first ICMC keynote. I recall quite clearly what I told the audience, and it still applies today, 50 years later.

For the most part, computers at the time were great, hulking devices, mostly used for research or business, such as spreadsheets and payroll. Computer graphics was limited to expensive specialized devices that were somewhat exclusive. We were living on the droppings of the military-industrial complex. More complete summaries may be found in [JAM 2000] and [JAM 2020]. We concluded rather quickly that for a long time to come, hardware acceleration would be the most direct way to speed up music synthesis, since the computer manufacturers had little or no reason to do so.

I met Peter Samson at MIT when I arrived in 1963. He had been programming the DEC PDP-1 to play 4-part Bach fugues. For sound output the top four bits of the

accumulator (there was only one) were wired to speakers. By toggling these bits on and off at regular rates, music could be made with these square waves. At Stanford, we contracted Peter and Systems Concepts, Inc. to build a large-scale dedicated audio synthesizer. This special-purpose device provided tremendous acceleration and was used at Stanford for a number of years, allowing composers speedy turn-around in their synthesis projects [LOY2013].

I was hired by George Lucas in 1980 with the goal of computerizing film sound production. While there, I build a large-scale digital signal processing (DSP) computer I called "The ASP". It was used in sound-effects production until about 1986. Probably the most memorable piece synthesized by the ASP was the THX logo theme ("Deepnote") which was shown in all THX theaters before the feature. I estimate it was performed more than a million times.

In the 80s and 90s, we saw the rise of interest in computer video and audio. The original Macintosh, circa 1984, had a built-in D/A converter. Note that hardware converters of that era were not built for audio – they were repurposed from military and scientific equipment. The introduction of the CD player in 1984 spurred interest in converters that were specially optimized for audio. By the 90's, every computer made came with some kind of graphics accelerator for video and an audio D/A converter. Some computers also featured A/D converters, often available as separate plug-in boards.

Fast-forward to the present, when 99% of modern processor chip area is optimized for video and audio streaming. The tail is wagging the dog. Rather than computer audio and video living off the advances in military and scientific technology, it is more that the revolution in military and scientific technology is being driven by the incredible advances in modern multi-media chip design. A self-piloting drone could not be made with the chips of 1980. Today, it uses chips designed for multi-media streaming. The irony is palpable.

"THE MAN IN THE MANGROVES COUNTS TO SLEEP"

In the early 2000's, I was performing at a local coffee house when I was approached by poet and FSU English professor Donna Decker about performing some of her poetry with musical accompaniment. This led to a series of

performances with Decker reading poems from her collection “Under the Influence of Paradise” while I added improvised accompaniment on tenor saxophone. This was reminiscent of the beat poets “poetry with jazz” from the 50’s, such as the work of Kenneth Patchen. I was particularly taken with one of these poems entitled “The Man in the Mangroves Counts to Sleep”. It is based on an interview she had with a man in a homeless encampment in a mangrove swamp in Key West, Florida. The man told her his story, noting that he had been a mathematician. As a trained mathematician, this resonated with me as well. As much as I enjoyed these performances, I felt that the poem needed a more forceful setting to drive home some of the important points. It touched on a number of aspects of the plight of the homeless, such as alcoholism, mental illness and health issues. I considered a number of musical settings – orchestra, string quartet, jazz trio, rock band and others. Each time I started on a setting, I got stuck. I finally decided that I had to appeal to the most flexible and most potentially powerful implementation possible – fully-synthesized computer music. This drew on much of my previous experience. I had extensive experience with DSP and speech synthesis with my early pieces at Stanford, plus I had had experience working with great modern composers, such as Luciano Berio, John Chowning, and others.

Since the poem is all first-person narrative, I needed a voice for the “man”. Donna suggested our friend, musician Frank Lindamood. He had the kind of husky voice that I thought was a good match for the imagined voice of the “man”. We had Frank record three performances of the poem. Since I needed them to be perfect, we sometimes had him repeat a line that I could later edit in so that all three performances used exactly the same words, no more and no less. I also recorded a reading of the first 100 prime numbers from my neighbor, William Heebink.

From my previous speech-synthesized piece “Lions are Growing”, I knew I needed exact timing of all the words down to the phoneme level. In 1976, I did this manually. The scope of this project was large enough that I felt I needed a lot of help from the computer. I wrote a python program that used the Penn Phonetics Laboratory Forced Aligner [P2FA] and the University of Sheffield Hidden Markov Model Toolkit (HTK) to perform an alignment of the recorded audio with the text of the poem. I formatted the result of this as an XML file. I did some post-processing on this to break the poem up into phrases of 3 or more words for easier handling. Note that some hand-corrections had to be made to the forced alignment

– P2FA/HTK does make mistakes – and to the segmentation into phrases. Given these data, I could easily get the exact timing of all the consonants and vowels, and could address the sounds down to the sample level (to the accuracy of P2FA).

Much of the following discussion may be found on my web page “The Making of Man” [JAM2002].

I had previously used linear prediction for “Lions are Growing” and had a pretty good idea how to do it. I felt it needed to be updated to more modern standards, such as 24-bit samples and 48 kHz sampling rate – that is, I needed higher quality speech synthesis than I had used before. Another problem was speed. In my 1979 Paper, “The Use of Linear Prediction in Computer Music Applications” [JAM1979], I noted that one minute of speech could be analyzed in “only” 45 minutes. Additionally, my analysis in 1975-79 used a maximum of 75 coefficients using Burg’s method [BURG1975]. I needed a much greater filter length for higher quality.

Note that there are a number of other options for speech analysis and synthesis. They all have advantages and disadvantages. I settled on linear prediction mostly because I was used to working with it. I could have used some kind of Fourier decomposition, but the results of the analysis are not as easy to modify for musical purposes. Linear prediction corresponds directly to the mechanism of production of speech, and thus is easily modifiable with (mostly) predictable results.

Having said this, it is not without problems. The first is order of the filter. In some preliminary experiments, I kept raising the order until I stopped hearing any change in the resulting synthesis. This was an order of 250 coefficients with Frank’s voice. None of the previous analysis techniques I had programmed to date would do that many coefficients. First, they started producing unstable filters, then the calculation of the coefficients would go unstable and produce NAN (“Not a Number”). At Stanford, I did some work on this exact question with Martin Morf and Daniel Lee, both graduate students in Stanford Electrical Engineering. They had come up with an analysis scheme called RLSN (“Recursive Least-Squares Normalized” - [LMF1981]). This scheme has astonishing stability. Every order up to about 750 has been stable and has produced stable filters. Additionally, the recursive nature of the calculation produces a full set coefficients *on every sample*. This makes sample-accurate time addressing very simple. The downside is that the

amount of data is large enough that *it cannot be computed offline and stored on the hard drive even today.* The files would be too large, and the time to call in a segment of the data is significant. It turned out to be both simpler and very nearly as fast to recompute the coefficients as needed. That was another benefit of the speed of the recursive form of estimation.

Other problems ensued with modifying the synthesis. I wanted independent control of timing, pitch, spectral shape and driving function. The first problem was that when I made the speech longer, it tended to sound like the speaker was drunk. In "Man", this might have been appropriate in certain places, but I needed to control the effect and eliminate it when needed. I found that one solution was to limit the amount of time stretch during the consonants. The P2FA analysis gave me the locations of the vowels and consonants so it was straightforward to limit the stretch just on the consonants.

A bit of listening to the synthesized sound led to a quest to "tame the sibilants". Even though there is less energy in the sibilants, they seemed to have a disproportionate impression of loudness. I found, again, that moderating the amount of stretch during sibilants helped quite a bit, but it was not enough. I ended up lowing the volume during any kind of sibilants. Again, it was trivial to find from the P2FA "map" where the sibilants were and drop the gain accordingly in a smooth manner. This sometimes caused a particular sibilant to become inaudible, but it seemed like a reasonable tradeoff.

Also, Frank, our narrator, had a tendency to "whistle" the /*s*/ in, for instance, "Counts to Sleep". This was a bit more tricky. I resolved it by applying a digital "de-esser" that I had set to exactly the frequency of Frank's whistle. Note that this is a common problem in the audio industry. It becomes much, much more annoying when the speech is stretched out to, say, double its length.

I had done some experiments at Stanford with modifying speech to make the speaker "sing" the utterance. This involves replacing the original pitch with a pitch that can change at syllable boundaries. I also added vibrato in the 3-5 Hz range. During legato transitions (all voiced), I had to limit the rate of pitch change to sound more natural. I downloaded a public-domain MIDI transcription of 190 Bach Chorales. I used four of these in "Man". The piece begins with chorale 20, "Ein Feste Burg Ist Unser Gott" with the lyrics replaced by the reading of prime

numbers. Note that in the last movement, the piece ends with the same chorale and more prime numbers, picking up in the 40's.

The driving function itself needed some study. Rather than use a pure pulse train, I found that rolling off the pulses and rolling in white noise in the 750-1500 Hz region made the smoothest speech without the "buzz" often present in synthetic speech. It also helped to randomize the phases of the components of the pulse train.

To help combat "narrative fatigue", I needed to keep changing the timbre of the speech. In many places, I could use the speech with the original timbre, but often I changed it either for special effects, or to emphasize some emotional content, or just to break up the monotony of a long narrative. This was suggested to me by composer and mentor Loren Rush. I used two principal modifications – spectral and driving function.

Note that it is virtually impossible to directly modify the filter computed by RLSN. The large number of coefficients makes most direct modification numerically unstable. The way I chose was to perform a sample-rate conversion, but then resample the result to produce a sound that can be moved up or down over a wide range. RLSN will happily track the spectrum at each point. We may then resample the resulting coefficients to correct to the original timing. This can then be treated like any other synthesis. Note that it requires that we replace the driving function. Otherwise the pitch would rise and fall with the spectrum. The end result is complete, seamless control of a "spectral warping" that can smoothly move the vocal resonances up and down with independent control of the pitch and timing. This is a very powerful tool for producing a wide variety of timbres, all based on human speech, and more or less comprehensible. More so if you know what is being said.

The simplest form of driving function modification is to use white noise. This produces a whispering sound. I also tried adding together several "normal" driving functions. Above about 3 components, it produced a rough, "barking" sound. More than 3 sounded like driving with white noise. One special case was to use an additional driving function at a much higher frequency – more than an octave higher – and not a harmonic of the "main" driving function. This produced an interesting sound that I used several times in the piece.

A final kind of timbre modification was to use what we call cross-synthesis. We can take a natural sound, such as a vacuum cleaner or a gurgling brook and run it through the vocal tract filter. This produces a wide range of strange sounds that are generally not comprehensible unless you know exactly what is being said. I use this in one place in the piece to make a "gurgling narrator" as a background sound.

My mentor, composer Loren Rush, also suggested locating points of emphasis – typically single words of phrases that are so important they need to get a spotlight. He suggested using a pause. That is, raise the emphasis by volume and timbre and go to dead silence after the end of the key phrase. This is what led to the 6-movement structure of the piece. I would build the tension in the narrative up to the key phrase, then pause. I started with 2-second pauses then later edited them to various lengths. This proved very effective in taking advantage of the natural grouping of the phrases and emphasizing the structure of the poem.

THE COMPOSITIONAL PROCESS

The above work (except for the pauses) was done before I started serious composition of the piece. I started by "blocking out" the piece by synthesizing the entire poem with a time stretch of 1.7 and raising the pitches by a perfect fifth. I then added accompaniment in several forms:

1. Chorales. As noted, these are generally Bach chorales, but also included two "tetrachord" chorales using words from the poem, and one chorale on "my country 'tis of thee".
2. "Mumbles". These consist of several highly modified voices slowly reciting words of the poem. This is used as background sound, perhaps reflecting the mumbling of the other residents of the homeless encampment.
3. Chants. Four or more voices in precise rhythm chanting words of the poem, sometimes on random pitches, sometimes on harmonics of the pedal tone.
4. Fantasies. Some phrases trigger sweeps of sound up or down (*e.g.* "Valley of Trash"). Some trigger wild gyrations of pitch. In one place, on the word "Lice", I make an effect like 50's science-fiction movies used. In others, like "my elliptical face", I make a mathematical joke (the phrase is repeated with pitch in phase quadrature – sine and cosine – which would make perfect ellipses).

I also needed a tonal basis for the piece, even though quite a bit of it is atonal. For a number of reasons, I chose 37.5 Hz as the fundamental pitch of the piece – the pedal tone, or the "God" tone. All the music, except the chorales, are based on harmonics (integral multiples) of this pitch. The piece starts with a chorale of prime numbers, then a "rain forest" of descending arpeggios on harmonics of 37.5 Hz. Each arpeggio is saying "The Man in the Mangrove " – exactly six syllables, on 6 consecutive descending harmonics. The combination of a number of these voices sets the tonal center of the entire piece to follow. In some ways, this is a nod to the perfection of numbers, such as the harmonic series, that contrasts with the chaos of life in the homeless camp. Note that when the rhythm is stable, the tempo is 60 beats per minute. The beat length of 1 per second is the time analog of the harmonic series in pitch. All the notes are multiples or integer-based fractions of 1 per second. Perfect, just-tempered harmony in pitch and in time representing a perfect clockwork idealized world. The poem has a spiritual aspect as well, signified by the appearance of the "angel". Reference is made later to the "music of the spheres". There are numerous references to the moon. The poem ends in a prayer.

This brings me to the main themes of the poem, and thus of the piece itself: order and chaos. Order is reflected by the pure perfection of mathematics. Chaos by life, especially life in the homeless encampment. The poem is a first-person narrative from the man. It starts by setting the scene of the homeless camp in the first movement. This movement ends with the "angel", which brings him to the final closing statement of the movement.

Movement 2 takes us back in time to the beginning of his story. Against the wailing voices in the sky, the man tells the story of his journey, how he was no longer able to live in society, in academics (chant on "numbers are forgiving"), or even the military. It seems he is not quite sure how he happened to end up in the homeless camp.

Movement 3 is a fantasy of a perfect clockwork world, where everything is in perfect rhythm and perfect harmony. It represents a vision of divine perfection which comes crashing back down into the daily routine, but it has left a knowledge of the divine, of the music of the spheres. Even his daily routine with its rhythm echos the perfection of a clockwork world.

Movement 4 represents the man's acceptance and accommodation to his situation. He no longer complains about his plight as long has he has his drink and his numbers. He accepts that this camp is the only place he could be.

Movement 5 is his last consideration of his routine and his acceptance, and the comfort of the familiarity of the Mangrove swamp. The images of the reflection of the black sky against the black water forms the backdrop of his existence and his persistent faith in his numbers.

Movement 6. While drifting to sleep, we drift into a fantasy and a final prayer and an ultimate transcendence.

The arpeggios from the beginning come back a few more times, symbolizing order and bringing us back again to the tonal base of the piece with a bit of order. We end with reminders of the chaos around him. At the end, the piece dissolves in a huge sweep of sounds, all saying "the other side of zero" and ending in an exhale as the man drifts off.

When I started this piece, I knew the scope of it was so large that I would never be able to notate all the musical elements. I have infinite respect for those who make scores of great complexity. I needed help from the computer. I wrote a number of "compositional assistants" to partially automate the task of detailed composition. For instance, there were many places where I just needed a group of sounds where I knew the general contour of the desired sounds (pitches, durations) but was not concerned with the exact values. I called these "clusters" and wrote one routine for making the clusters. I would specify the start time and duration of the cluster, along with the number of musical elements to go into the cluster, and a number from zero to one specifying where the most dense part of the cluster should be (the centroid of the distribution, to be precise). In addition, I passed in a routine to generate the desired sound for each element of the cluster. The "rain forest" sequences were built this way. Each note was given its placement in the cluster as a single number, zero to one. This allowed the sound generating routine to generate the detailed parameters used to drive the speech synthesis. I would in no way call this "AI". I viewed the computer as a "composer's assistant" rather than an independent entity. Note that random numbers were used liberally in this process. The cluster generator did the placement entirely through random numbers, given the duration, the number of sounds (notes?) and the location of the centroid of the distribution.

There was another assistant that would take a MIDI data set along with the starting word/syllable of the text and produce a detailed synthesis plan that stretched and compressed the syllables to fit the timing and pitch of the MIDI notes. This involved a fair amount of work to get the singing synthesis to sound right. In some sense, this required programming many of the aspects that singers do absolutely naturally without thinking. For instance, long notes are always on vowels. Nobody makes an */s/* or */sh/* last, say, 3 seconds. Care must be taken to synthesize the consonants properly, while more freedom is available to adjust timing during the vowels (or silences).

As the complexity of the piece increased throughout the compositional process, it became harder and harder to adjust the gains properly. There are places in the piece where there are as many as 100 individual sounds occurring. One might think that we can just display a 100-channel music mixing desk and control everything manually. One problem with computer synthesis is that the orchestra itself is constantly changing. It is as if performers are constantly walking on and off the stage, some showing up for only a few seconds, never to return. The mixer would have had to be over 300 channels to logically group similar sounds together. My impression is that unless you have some organizing principle, like the film industry's division into music, dialog and effects, or the music industry placing group microphones and soloist microphones around the orchestra, there is no way to even remember what channel a particular sound is on. I have watched mixers in a big film mix (64 channels) going through the channels one by one to find some offending sound that popped out somewhere[1]. In "Man", I did not come up with any satisfactory solution to this issue, so I was forced to manually adjust everything through trail and error (mostly error). In [JAM2000], I suggested automating this process by tracking the perceptual audibility of each sound which could then be automated. The composer could then just specify what audibility was desired from each sound rather than an absolute gain in dB. The program could then use perceptual models to generate the individual gains that would satisfy the audibility requirements.

As noted above, random numbers were liberally used to determine compositional details. This turned out to be a mixed blessing. Yes, it did relieve the burden of

[1] This was called a "werf" by the engineers

deciding and writing down hundreds of numbers, but as usual, it comes with some added complexities. Anyone who has seen Mickey Mouse as “The Sorcerer’s Apprentice” will understand that there will be unintended consequences. The composer is often surprised at the results. One problem is that generally an odd consequence is not repeatable – it depended on the random numbers falling out in a certain way. This sometimes produces serendipitous results, but usually not. One run of “Man” featured a key word of a phrase ending up about 3 seconds late. This happened with the phrase “The Only Place”. Since it would not repeat the exact sequence of random numbers, I placed that one sound manually and made it permanent.

Another problem with random numbers is that the overall length of the piece and duration of each segment can be different. For instance, “Man” is about 13 minutes long. I found that different builds could vary in duration by as much as 90 seconds. This didn’t matter too much until we decided to make an animation to go along with the music. The first time I revised the piece after engaging the animators, they screamed that all their timings had changed. I partially fixed this by taking a snapshot of the random number generator before each movement so that I could rebuild one movement at a time and perfectly reproduce the others. This is, of course, a partial solution. I think the only definitive solution would be to use multiple random number generators – perhaps named and organized hierarchically – so that relatively fine control is possible. Or perhaps add some more “intelligence” so that the exact duration of each compositional event could be specified. The use of random numbers for composition needs more thought than I could give it here.

I used computers made in 2009. They were state-of-the-art at the time. Even so, it took 5½ hours to build the entire piece and had a memory footprint of about 68 GB. Since I could rebuild one movement at a time independently, I often had several computers working simultaneously. In 2022, I added a computer using the AMD Ryzen 9800X3D as the processor. It reduced the build time to 90 minutes. It seems that every time someone tells me that Moore’s law has been reached, some new development, architectural or technological allows yet another big increase in compute speed. One has to wonder if there ever will be a hard limit.

The piece is so complex that it would have been impossible to produce it in 1980 or even in 2000. I was impressed in [JAM2000] that I could run 2 channels of concert-hall reverberation by direct convolution in real time. Today, I can run hundreds of channels of million-point convolutions in real time. I saw an improvement of about 18,000 from 1975 to 2000. I have seen another increase of "a few" hundred times more since then (over a million times 1975 speeds).

CONCLUSIONS

I will conclude by some observations and a suggestion.

I noted earlier that for computer music synthesis, we start by building our orchestra. Given the unlimited freedom that pure computer synthesis allows, this will probably always be an issue, although one might hope that at some point we reach a "critical mass" and relegate orchestra creation to a curiosity, perhaps like Harry Parch [https://en.wikipedia.org/wiki/Harry_Partch].

Nonetheless, despite some of the difficulties of creating pure computer music, over the last 50 years we have created an enormously powerful tool and instrument. Let me reveal here what I said in 1975 at the second ICMC: I advocated the creation of composer's assistants. I advocated building musical knowledge into the computer to aid with composition and realization. In my Heyser lecture for the AES in 2000, I again raised the idea of digital assistants and repeated the call in my follow-on paper in 2020. In "Man in the Mangroves", as noted above, I built an array of very simple "composer's assistants" to fill in details of composition where the exact placement of the musical elements was not as important as the overall contour of the musical event. These were simple enough that nobody would describe them as "intelligent". Surely we can do better. Perhaps neural nets will be the vehicle for musical intelligence, but what will we ask it to do? How do we train musical intelligence into a computer? Do we give it examples of "good" music and "bad" music? I am sure that this question of what musical intelligence is or can be will be a subject of much discussion and debate in the years to come. These questions are surely part of why the musically-intelligent assistants I suggested in 1975 have been slow in coming. There are now a number of devices available that perform some features that could be called "intelligent". I expect the number and sophistication of these devices will increase sharply over the next decade or two.

EPILOGUE

One remaining question in my mind is “what are we going to do with this powerful tool?” The concerts at this very conference present many answers to this question. I would like us all to consider the social aspects of our work. Beautiful technology does not always make beautiful music. While dealing with the intricacies of high-tech music synthesis, we must never lose track of the human aspects of what we do. Technology must always be developed for human ideals, hopes and desires. Let us work to use this great instrument to uplift the human condition and make this a better world – for us, for our children, for our children’s children – in our journey to the other side of zero.

REFERENCES

[P2FA] "The Penn Phonetics Laboratory Forced Aligner" https://babel.ling.upenn.edu/phonetics/old_website_2015/p2fa/index.html

[HTK] "The Hidden Markov Model Toolkit" https://spandh.dcs.shef.ac.uk/ed_arena/htk/index.html

[JAM1975] "Signal Processing Aspects of Computer Music" Proceedings of the IEEE, Volume 65, Number 8, August 1977, pp1108-1137

[JAM1979] "The Use of Linear Prediction in Computer Music Applications" https://www.jamminpower.org/PDF/JAM/Linear%20Prediction%201979.pdf

[JAM2000] "Audio in the New Millennium" AES Heyser lecture https://aes.org/community/technical-council/richard-c-heyser-memorial-lecture-series/memorial-lecture-at-108th-james-a-moorer-audio-in-the-new-millennium/

[JAM2002] The Making of "Man in the Mangrove" https://www.jamminpower.org/MAN_makingof.html

[JAM2020] "Audio in the New Millennium – Redux" AES Convention 149, Paper Number 10438

[BURG1975] "Maximum Entropy Spectral Analysis", PhD Dissertation, Stanford, 1975

[LMF1981] Lee, Morf and Friedlander "Recursive Least Squares Ladder Estimation Algorithms" IEEE Trans C&S V28, #6, June 1981, 467-481

[LOY2013] Loy "The Systems Concepts Digital Synthesizer: An Architectural Retrospective" Computer Music Journal V37, #3 September 2013, pp49-67