
by

Dr. Miles A. Marks.

Microcomputer Music Research Unit
Schools of Studies in Computing
University of Bradford
Bradford
Yorkshire
BD7 1DP
England

Abstract.
An additive synthesis system requires to make complex decisions using artificial intelligence methods to optimize the use of available resources and achieve best possible sound quality.

Introduction.
The allocation of resources in a real time additive synthesis system requires many decisions to be taken rapidly. The techniques used apply equally to any additive synthesis system, but particularly to polyphonic instrument simulation. As generators are relatively expensive it is essential that they are manipulated wisely to create high quality sounds. This paper considers three areas of resource allocation optimization: generator allocation, a sound priority system, and waveform allocation. The resource allocation algorithms described within this paper have been implemented using the Bradford Musical Instrument Simulator (BMIS) (References 1 and 3).

Generator Allocation.
When a key on an instrument is played the task of the computer system controlling generator allocation is to obtain a generator, or generators, to reproduce the sound. The BMIS system synthesizes waveforms and stores them in waveform memory when a particular sound is selected. The waveform may consist of a single sinewave or a number of sinewaves added together to form a complex waveform. Generators are able to reference any of the stored waveforms to reproduce the required sound. Allocating generators is a trivial task when enough free generators are available. However, when all generators within a system are allocated, some method for synthesizing the new sound must be found. The most commonly implemented method is polyphonic allocation.
This approach segregates all the generators available into groups such that when a note is played a group of generators is allocated to that sound. Once all groups are in use and a new note is played then one of two possible choices has to be made. Either the new note is ignored, which is the normal case. Or, the new note sounds, but a previously played note is removed in its entirety even though it may still be sounding.

Polyphonic allocation suffers from a number of disadvantages:
1. The system is restricted to a particular number of notes rather than generators - all generators may not be producing sound.
2. A fixed number of generators is allocated to each sound over the entire keyboard range. When fewer generators are required at certain points of the keyboard this results in generator wastage.
3. As sounds decay and generators become silent no attempt is made to re-use silent generators individually, which theoretically are free, until the entire group is finished with.

Experiments indicate that an additive synthesis system employing polyphonic allocation, with these limitations, will produce lower quality sounds when all generators are allocated, than a system using sophisticated artificial intelligence methods for the allocation of resources.

The algorithms employed in the BMIS ensure that the usage of generators is highly optimized. Firstly generators are used individually and only when a generator will produce sound is it allocated (this overcomes disadvantages 1 & 2 above). When a generator has completed its function of sound production it is automatically returned to a free pool of generators (overcoming disadvantage 3 above). Finally, when all generators have been allocated and another generator is required for sound production a 'single' sounding generator is 'stolen' rather than a group therefore ensuring full polyphony. The methods employed to acquire this generator ensure that the delay to pick it up is not appreciably longer than the delay required to pick up a free generator, and that the generator removed is not producing sounds whose disappearance can be readily detected. This ensures that however many notes are sounding, the new note will always sound, without ready detection of notes or generators being removed.

A Sound Priority System

The need for a sound priority system only becomes apparent when the music generating system is using all its resources and further sounds are to be produced. A multi-priority system defines each sound component with

379

ICMC Proceedings 1988
a particular priority. The allocation procedure is controlled by queuing played notes and subsequently allocating generators in priority order starting with the highest first. In particular, the queuing ensures that when a group of notes is played together the allocation is in sound priority order, not in note order. This minimizes resources being allocated to one sound and then immediately being re-allocated to another.

When resources are in limited supply some components may sound and others may not. The components which don't sound need to be insignificant compared with the total sound the system is producing at that instance. (i.e. When many generators are sounding certain parts of the sound will have a higher importance than others. This may be because the higher importance sounds have larger amplitudes. If only one sound can be created the one of lower importance will be ignored.) Once all resources have been used then they may be reallocated under certain conditions. Firstly, lower priority sounds should never remove sounds of higher priority. Secondly, for particular priority levels it may be desirable, perhaps for speed reasons, not to attempt allocating resources to them at all.

Consider a sound priority system in its simplest form having two levels of importance, level 1 and 2. Level 1 sounds have absolute priority and any sounds specified in this category are created regardless of other system resources or restraints. Level 2 sounds, on the other hand, only use resources if they are free (i.e. not already allocated), otherwise if no resources are available then this portion of the sound is ignored. This ensures that the more important components of the sound (the louder parts) will always be reproduced while the lesser important components will be produced if resources are available. It can be seen that as the number of priorities increase the processor power required to handle these complex algorithms will also increase. Therefore, the number of priority levels will be dependent on the processor power available within a particular system.

Certain instrument simulations, particularly those with inharmonic partials, can use a significant amount of generators per note. A method called note limiting provides a means of attaching priorities to notes as they are being played. The priorities are dependent on keyboard activity. To achieve this the note data is delayed for a period before attaching a priority to each note ready for generator allocation, using the priority allocation scheme mentioned earlier. A note limiting procedure using this method has been incorporated into the BMIS system. For example, when a
chord is played on the tubular bells simulation, and limited resources are available, then only carefully selected notes from that chord will sound. Whereas the sound priority method provides an importance for each sound, the number of notes played together being of no importance, the note limiting facility takes into account the number of notes being played together. This ensures that allocation and immediate de-allocation of resources is minimized, as this is highly inefficient and degrades response time in a system considerably.

The approach of using sound priority techniques, allows the instrument creator to specify the components of the sound in hierarchical order. The resources can then be allocated during playing to achieve the most accurate and fullest quality of sound reproduction possible with the resources available at that time.

Waveform Allocation.

Although memory chips have increased in storage capacity and their cost has fallen, it is highly desirable to be able to use and manipulate waveform memory easily and to minimize the total amount needed in a system.

An instrument which has the capability of switching between different sounds must either store all the waveforms that may be required in waveform memory or some procedure must be implemented for generating waveforms into waveform memory when a different sound is selected. A system with the capability of pre-synthesizing complex instrument waveforms will normally store its harmonic data in computer memory. When a particular instrument is selected waveforms are generated into a waveform memory for subsequent use when notes are played. The BMIS system has a waveform generator facility available. To minimize the total amount of waveform memory required the software needs to allocate and de-allocate waveform memory, in real time, to the currently selected sounds. Appropriate waveforms are then generated into the allocated memory by highly efficient software which controls the hardware waveform generator.

If insufficient waveform memory is available to hold the waveforms generated then these waveforms must be compressed into a smaller number of samples. This results in a detectable loss of quality in the sound. An algorithm is available in BMIS for this purpose.

future Work.

Further improvements in the entire area of resource allocation in additive synthesis systems could be directed towards more complex evaluation of the importance of sounds allocated to generators that are to be "stolen", for example, a priority for removal
Could be attached to each generator in use, which is dependent on the existing generator sounding. The priority could take into account pitch as well as amplitude. For example, if all the generators sounding are of high pitch except one which is of low pitch, it may be unacceptable to remove this low pitched generator which could occur by using an amplitude only algorithm for removal.

Acknowledgements

I thank Dr. Peter Comerford for his advice and assistance with these resource allocation techniques and Miss Lucy Kitching for her valuable comments and encouragement.

References