Complex Signal Synthesis for Real-Time Simulations

Ray Hardison
Ixthos, Inc.

Introduction

Most DSP based applications process signals as inputs. This paper presents an application in which DSPs are used to generate complex real-world signals as outputs.

The Acoustic Signal Generation System (ASGS) was developed for the Naval Air Warfare Center Aircraft Division at Patuxent River, Maryland. Its function is to simulate the output of an 8-channel sonobuoy receiver as if the receiver is receiving data from the real world. These outputs drive the inputs of an Acoustic Signal Processor.

Application Overview

An Acoustic Signal Processor is a system that uses sonobuoy inputs to detect and localize submarines. It is designed to make use of the dynamic effects of submarine noise emission, reflectivity, and motion. The operator of the Acoustic Signal Processor uses combinations of different types of sonobuoys strategically placed to focus in on a target position. The effectiveness of the sonobuoys to detect the submarine will not only be determined by their position and the submarine dynamics, but also by the sound transmission characteristics of the water they are located in. For ASGS to provide a realistic simulation, all the dynamics of the real world environment need to be modeled.

The ASGS simulation consists of three major elements: sonobuoys, targets, and the ocean. These elements and their interrelationships are depicted in Figure 1.

1. Sonobuoys

A sonobuoy is a device that is dropped into the water from an aircraft and transmits the acoustic signals present in the water to the aircraft via a radio transmitter. Sonobuoys are available as omni-directional (LOFAR), directional (DIFAR), and directional active (DICASS) versions.

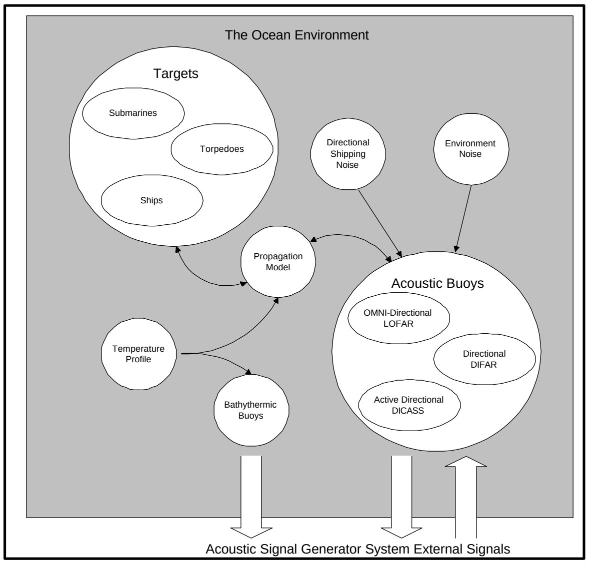

Figure 1. Application Environment

An omni-directional sonobuoy has a single acoustic band of output where directional sonobuoys have three signal components (omni, North/South, East/West) which are multiplexed together and sent to the aircraft on a single radio channel. Active sonobuoys' transmit signals into the water, called pings, which are used by the Acoustic Signal Processor for echo ranging.

ASGS simulates up to eight deployed sonobuoys. Buoys can be scuttled, re-deployed, or replaced at any time during a scenario.

2. Targets

The purpose of a sonobuoy is to detect targets. ASGS can simulate up to four targets present in a scenario, at one time. These targets can be any combination of three types: submarines, torpedoes, and surface ships. At any point during a scenario a target can be killed, initiated, launched (for a torpedo), and have its course, speed, or depth changed.

A sonobuoy can detect a target two ways, by its radiated noise emissions (passive detection), or by its sound reflection characteristics (active detection). ASGS models both with a large set of operator definable parameters.

A target's passive emissions are composed of narrow band and broad band components. Narrow band components consist of discrete tonals and harmonic families that can have target speed dependencies, and frequency instabilities. Broad band components form a noise signature. The magnitude of the passive emissions can change as a function of aspect to the target in addition to proximity.

A target's reflection characteristic is composed of reflective losses at different positions along the structure. These reflections are all combined with the appropriate time delays and aspect ordering, to provide a complex reflection signature to the Acoustic Signal Processor. The reflection level is also aspect and proximal dependent.

3. The Ocean Model

Target emissions and reflections must pass through the ocean environment before being received at the sonobuoy. Sound propagation in the sea is influenced by physical characteristics of the water and its boundaries. ASGS models the propagation influences of the sea by permitting the operator to set parameters which are used to calculate the attenuation and interference effects that are to be applied to each target's signal. The resulting losses are frequency dependent since the ocean has a low pass characteristic. Under certain conditions, multi-path effects may also be present and must also be modelled.

The ocean also has a noise component that can be related to a number of different factors such as sea-state, distant shipping, and aquatic life. The magnitude and spectral shape of noise can mask a target's noise making it more difficult to detect. ASGS supports the simulation of environmental noises.

Design Considerations

To generate the complex signatures of the targets and environmental noises, it was realized a number of DSPs would be needed. Each target can have up to 394 frequency components in addition to 24 – 1/3 octave band noise components. Since frequency components have dynamic variants such as speed dependency, instabilities, and Doppler effects, every output sample needs to be calculated and summed together. Noise for each band needs to be separately derived, filtered, and added to the frequency components. The noise in each band needs to be from a different source to prevent adjacent bands from experiencing coherent additive effects in the overlapping filter roll-off regions.

ASGS supports a 50 square nautical mile gaming area. (The gaming area is the region where targets and sonobuoys can all reside and interact.) Propagation delay for sound in the ocean is roughly 1 second per nautical mile. This means that a target's sound emissions received by different sonobuoys at different distances from the target will have originated at the target at different times. ASGS must maintain the time relationships and target emission history so that signals received at different buoys are coherent for a target.

Active buoys presented a real-time response requirement for ASGS. Within 1 millisecond of receiving an external trigger from the Acoustic Signal Processor, ASGS must blank the sonobuoys output and raise the TTL output signal for the buoy for the period of the ping. The time of the ping initiation is used to calculate; the propagation times from the sonobuoy to the target, and the target reflection back to the sonobuoy. These timings must be accurate to 1 millisecond to obtain an accurate localization. In the period of time required for the ping to reach the target and return to the sonobuoy, ASGS must determine the target/sonobuoy geometry (distance, target aspect to sonobuoy, sonobuoy aspect to target) Doppler shift effects, reflection envelope, and arrival time. ASGS must simulate the reception of ping artifacts to all the sonobuoys in a scene capable of receiving them. These artifacts are bottom bounce (reverberation), direct ping reception, and target reflections.

It became apparent that hardware and software solutions would have to be combined to obtain a real-time full featured simulation. Initial investigations into the computational requirements of one target's signal generation showed computational requirements that far exceeded any current DSP product. (19 million sine operations per second would need to be performed for just the frequency components. 394 elements x 48000 samples per second)

Approach

To address the design considerations, a significant amount of algorithmic modeling was performed. Several methods were developed which lowered computational requirements at the expense of increasing memory requirements. Others were strategic in nature, reducing the number of times per second specific values were recalculated. The following address the most significant strategies employed.

1. Sine lookup tables

To reduce the number of sine operations that would be necessary to generate a target's frequency components, sine lookup tables were developed. Each harmonic family has a 2048 element table that is sized to one cycle of the fundamental frequency. Each family member present is summed into the table with its relative phase and magnitude. When a sample of output data is calculated, a lookup algorithm selects the table position to extract and performs a linear interpolation of the value. The same logic is used on a single sine lookup table for discrete frequencies. For discrete frequencies the computational reductions are minimal. But for harmonic families the computation reductions are significant.

The sine lookup table's benefit has two major implementation costs. First the tables must be constructed prior to simulation start, and at that point a large number of sine operations must be performed. Second, the tables consume a significant amount of memory space in a DSP, and that's where they need to be located to get maximum performance. These tables were the main reason the Analog Devices 21060 was considered the prime candidate for the task since it has .5 Mbyte internal memory. Likewise the ability to perform all calculations and interpolations in 32 bit floating point minimized resolution impacts due to the 2048 table quantization

2. Cascaded noise filters

A target's noise signature can be specified as amplitudes for 24 - 1/3 octave frequency bands. It is constructed by summing the output of 24 second order Butterworth band pass filters each fed with random noise. Since the phase response of a filter used in this manner is not relevant, an Infinite Impulse Response (IIR) filter was used. Implemented as a cascaded bi-quad, the IIR can be performed very quickly on a DSP. To further reduce the computational time, lower frequency bands were filtered at lower sample rates then summed, interpolated, and summed with a higher frequency group. In all, 4 groupings occurred, with only the 4 highest frequency bands being filtered at the full sample rate

To reduce the amount of time required to calculate random numbers, one full vector of random numbers was calculated, which was used in an inverse order for every other band. This significantly reduced the number of random number calculations, while keeping adjacent bands incoherent.

3. Digital delay lines

To deal with the propagation time issues, a digital delay line was implemented. One was needed for each target and would need to be capable of holding at least 50 seconds of signal data. Extraction logic was developed to remove data from the delay line at each arrival time and apply the appropriate Doppler compression / expansion effects. Each targets delay line was implemented as a circular queue with out-of-range buoys receiving zero values. To implement the delay line structures required a large local shared memory area that had low overhead accessibility.

4. Time partitioning

ASGS runs a simulation in one-second steps. This means for each second, a new position is calculated for all targets and buoys, propagation losses and arrival times are determined, any operator or scripted events are acted upon, and the operator's display is updated. This works fine for the user's interface and the physical geometry calculations since in both cases things change in a relatively slow progression. But for the simulation signals creation, transitions need to be smooth between seconds, plus the accuracy of arrival times needs to be to the millisecond for positional determination.

To accommodate the different time resolutions, and provide a flexible environment for remote user's interfaces, the software design was partitioned into three pieces. The user's interface which receives updates once per second, a simulation control program which performs the one second environment calculations, and a signal synthesis section. The simulation control program runs one second ahead of the signal synthesis so that the synthesis section can have "work to" values that it can apply to the signals evenly over the one second periods giving smooth transitions.

In the signal synthesis section, modulators for long term and short term frequency instability, are calculated every millisecond versus once per sample. This reduces the computational time and partitions the signal data into 1 millisecond packets that get passed through a series of processes. The result being no one process has possession of more than a couple of packets at once.

Since a 48 kHz sample rate was only required at the last stage of signal generation (where the multiplexing of the signal occurs) a working sample rate of 24 kHz was used for target signals and directional elements.

Implementation

ASGS is implemented as three discrete sub-system elements: the Client Interface Software, the Simulation Control Program, and the Buoy Synthesizer System.

The Client Interface Software (CIS) is the graphical user's interface for simulation scenario creation, execution and control. It can be executed on a remote UNIX workstation or run as a task on the VME host computer. Motif style menus and edit windows are utilized so the operator can create or modify the definition files for targets, buoys, ocean models, and scenarios. A scenario can be totally script driven, with operator interaction possible at any point.

The Simulation Control Program runs on a VMIC VMEbus single board computer under the control of the CIS. Once the operator has selected the scenario elements, the CIS sends the information to the SCP. The SCP loads the DSPs with their programs and establishes communication. Then the SCP waits for a start command from the CIS. Once started the SCP executes a once per second simulation pass that calculates all target and buoy dynamics. It then updates the BSS and CIS.

The SCP can be interrupted during its one second cycle when a ping occurs. The SCP records the time, buoy, and parameters of the ping, determines when the ping will arrive at any targets or other buoys, and determines what effects the DSPs will need to generate for the simulation.

The Buoy Synthesizer System (BSS) consist of 3 Ixthos Quad SHARC VMEbus boards, one with 2 - 16Mbyte DRAM modules, and another with a analog I/O board, and a TTL I/O board. The processes are grouped into four discrete functions. They are; target signal generation, buoy reception, buoy effects generation, and buoy output generation. All DSPs' use the SHARC link ports for passing the synthesized data.

1. Target Signal Generation

The target signal generation process runs on each of the DSPs on one of the boards. This is also the board that has the 2 - 16Mbyte DRAM modules. Each process uses half of one module for the digital delay line. Every millisecond the process calculates all target passive emissions, and cavitation, noise burst or tone burst effects, sums them together, and places them into the delay line. Then the process will remove from the delay line up to eight sonobuoys' received data, based on arrival times provided by the SCP. This data gets sent every millisecond to the buoy reception process

2. Buoy Reception

The buoy reception process runs on two of the DSPs and each process receives data from two target signal generation processes. It applies propagation loss filters to the data, and computes the directional components of each target at each buoy. Each buoy's data gets summed, and then passed on to the buoy output process.

3. Buoy Effects Process

The buoy effects process runs on 4 DSPs. Each process handles two buoys and calculates the environmental and directional shipping noise, and any active ping effects occurring for each buoy. Ping effects are generated with information provided by the SCP and consist of; target reflections, reverberation, and direct pings received from other buoys. Directional weights are applied and they are summed into the buoys noise data.

4. Buoy Output Process

There are two buoy output processes that are computationally alike, yet they differ in their I/O handling responsibilities. One controls the analog I/O, which means all eight channels of output must be collected and formatted into a one millisecond buffer by the process. In addition it must receive eight channels of input data and evaluate it for the presence of ping request signals from the Acoustic Signal Processor. Flags are set for active requests, and cleared once a request has ended. These flags are sent to the other buoy output process that has access to the TTL outputs.

Both processes evaluate the flags for the buoys they are responsible for. When a valid request is received for a channel, the process issues a VME interrupt to alert the SCP that the request has occurred. Meanwhile, the process that issued the interrupt will "blank" the channel's output (zero level signal) for the duration of the ping.

All BSS processes keep a 1 millisecond counter which is advanced with each data transmittal. Since all processes feed into the buoy output process, all counters are synchronized to each other. The SCP looks into one of the BSS processes to keep its one second scenario clock synchronized one second ahead of the BSS. It performs this by using an Ixthos HostAPI function to read a processor's data memory location of the one millisecond counter.

Testing Approach

The development of ASGS required a significant amount of incremental testing, specifically in the BSS. Where the CIS and SCP could be easily validated by conventional methods, the BSS had to be validated by collecting seconds of data and using data analysis / visualization software to determine if time domain and frequency domain elements of the data were correct. It was in this area, that the delay line memory modules served a dual role. The first process to be developed and tested was the target signal generation process that conveniently generates a time history in the delay line. By extracting the data from the delay line into a file on a PC, analysis software could process and display spectral time histories. This method proved so useful in evaluating all the variations of the process capabilities, that a strategy was developed to position one of each processes in the BSS around the two memory modules and have them place their respective outputs/inputs into the memory for analysis. Then, incrementally, each stage of the data through the BSS was validated. A simple test of the analog I/O board to validate its ability to output in analog form what was expected from numerical data, quickly confirmed that the data flow's integrity would be maintained into the output stage.

Once all flows and functions had been verified, ASGS had its first connection established to an Acoustic Signal Processor. Within a two week period, ASGS was verified, validated, and delivered.

Ray Hardison is the Manager of Software Engineering at Ixthos Inc. in Leesburg Virginia. Prior to working at Ixthos he was a principal developer of the ASGS system for The CRT Corporation in Rockville Maryland. He has 20 years of experience in acoustic signal processing, ASW applications, and computer simulations.