

UNIVERSITY GRANTS COMMISSION
Bahadur Shah Zafar Marg
New Delhi 110 002

1. **Project Report No.**
Final
2. **UGC Reference No.**
F.No. 41-666/2012 (SR) dated 23 Jul 2012
F.No. 41-666/2012 (SR) dated 23 Jun 2015
3. **Period of report**
From 01-07-2012 to 31-12-2015
4. **Title of the project**
Development of portable Doppler sodar system for wind farm siting applications
5. **(a) Name of the Principal Investigator**
Prof M Purnachandra Rao
(b) Department
Department of Systems Design
(c) University where work has progressed
Andhra University, Visakhapatnam 530 003
6. **Effective date of starting of the project**
01-07-2012
7. **Grant approved and expenditure incurred during the period of the report**
 - (a) Total amount approved**
Rs.11,95,000
 - (b) Total expenditure**
Rs.11,71,006
 - (c) Report of the work done**

i) Brief objective of the project

The objective is to design and develop a single-antenna sodar for total wind profile measurements. This requires the development of an antenna with light-weight acoustic transducers and enclosure. The measurements need to be calibrated against standard measurements and operate the system at two sites in Visakhapatnam.

ii) Work done so far and results achieved and publications, if any, resulting from the work

1. Kamala Kumari, A., Rao, A.B., Nagaraju, A., Rao, M.P., Vani, M., and Rao, K.T.: Development and experimental implementation of planar array antenna for monostatic Doppler sodar, *Global J. Res. Engg.* 13 (2013) 25-27.
2. Hareesh Babu, M., Bala Naga Bhushanamu, M. and Rao, M.P.: 2 Khz mono pulse acoustic signal generation for sodar application using PC sound card and Matlab graphical user interface, *Int. J. Res. Computer & Comm. Tech.* 2 (2013) 1522-1526.
3. Bala Naga Bhushanamu, M., Hareesh Babu, M. and Rao, M.P.: Design and Simulation of graphical user interface for sodar system using VC++, *Int. J. Innovative Res. Develop.* 2 (2013) 155-159.
4. Hareesh Babu, M., Bala Naga Bhushanamu, M., Raju, D.S.S.N., Benarji, B. and Rao, M.P.: 'A novel mathematical computing simulation methodology for acoustic phased array antenna of sodar system', *Int. J. Cur. Engg. Tech.* 2 (2014) 1064-1067.

iii) Has the progress been according to original plan of work and towards achieving the objective, if not, state reasons.

Yes, work is completed as planned.

iv) Please indicate the difficulties, if any, experienced in implementing the project

The antenna systems of the Doppler sodar were placed on the terrace of the three-storey building of the Department of the Systems Design in the university campus. Each antenna system had an acoustic enclosure, an array antenna with acoustic reflector, and a preamplifier unit. The measurements and calibration of the new Doppler sodar system were carried out since July 2014. Visakhapatnam was hit by an extremely severe cyclonic storm on 12th October 2014 with winds reported up to about 185 km per hour. This cyclone was named by the India Meteorological Department as 'Hudhud'. The extreme winds caused huge damage to the Visakhapatnam city and the region adjacent to it. The antenna systems of the Doppler sodar were damaged and some of the sub-units were swept away by the

winds. However, some restoration work was done and some useful results were obtained.

- v) **If project has not been completed, please indicate the approximate time by which it is likely to be completed. A summary of the work done for the period (annual basis) may please be sent to the Commission on a separate sheet.**

Project has been completed successfully.

- vi) **If project has been completed, please enclose a summary of the findings of the study. One bound copy of the final report of work done may also be sent to the University Grants Commission.**

A bound copy is being submitted to the Commission.

- vii) **Any other information which would help in evaluation of work done on the project. At the completion of the project, the first report should indicate the output, such as:**

(a) Manpower trained

5 research students are trained.

(b) Ph.D. awarded

Design and development of Doppler sodar system using pulse compression technique – A. Nagaraju, 2014

Development of an ultra-light acoustic array antenna system for portable Doppler sodar – K. Tarakeswara Rao, 2014

Development of a monostatic Doppler sodar system based on SDR concept – V. Vijaya Kumar, 2015

Design and development of multi-beam antenna for Doppler sodar applications – A. Kamala Kumari, 2015

(c) Publication of results

Kamala Kumari, A., Rao, A.B., Nagaraju, A., Rao, M.P., Vani, M., and Rao, K.T.: Development and experimental implementation of planar array antenna for monostatic Doppler sodar, *Global J. Res. Engg.* 13 (2013) 25-27.

Hareesh Babu, M., Bala Naga Bhushanamu, M. and Rao, M.P.: 2 Khz mono pulse acoustic signal generation for sodar application using PC sound card and Matlab graphical user interface, *Int. J. Res. Computer & Comm. Tech.* 2 (2013) 1522-1526.

Bala Naga Bhushanamu, M., Hareesh Babu, M. and Rao, M.P.: Design and Simulation of graphical user interface for sodar system using VC++, *Int. J. Innovative Res. Develop.* 2 (2013) 155-159.

Hareesh Babu, M., Bala Naga Bhushanamu, M., Raju, D.S.S.N., Benarji, B. and Rao, M.P.: 'A novel mathematical computing simulation methodology for acoustic phased array antenna of sodar system', *Int. J. Cur. Engg. Tech.* 2 (2014) 1064-1067.

(d) Other impact, if any

The work has created a good research facility in the university in the field of electronic instrumentation and lower atmosphere.

Signature of the Principal Investigator

Signature of the Head of the Department

Signature of the Principal

Signature of the Registrar

Description of the work done

A typical Doppler sodar system operating in monostatic mode that is capable of measuring the total wind vector consists of three antennas oriented in three different directions for sensing the three components of the wind. Some of the recent sodar systems have five antennas for cross checking the echo-signal quality. One antenna is pointed to zenith and the other two antennas are oriented to south (north) and east (west). The two antennas are usually tilted 20° from the vertical axis. The type of antennas range from a simple parabolic dish to more sophisticated planar surface array. The choice depends on a variety of parameters such as the site, height coverage required, etc. The antennas are usually having an acoustic absorbing cuff around each one of them to minimize ambient and environmental noise interfering with the atmospheric echo-signal. In fact, the acoustic enclosure is the bulkiest part of the total sodar system. In the proposed system, a phased array antenna that can produce the required three beams in three different directions will be used. This reduces the size and bulkiness of the antenna system, thus, making the system light-weight and portable.

The transmitter unit of the Doppler sodar system consists of a commercial available audio power amplifier of about 500 W output weighing about 10 kg. The transmitter unit of the proposed system in the project will have an in-house developed power amplifier incorporating state-of-the-art electronics. This also reduces the weight of the Doppler sodar system. The remaining electronic units of the receiver and other allied circuits will be housed in a single box weighing about a Kg. The data processing unit will be a laptop class of computer. The

entire signal generation for the transmission of three frequencies for the three different directions is done by the on-board sound card of the computer. The echo-signal, after pre-processing in the preamplifier and the receiver, will be given to the input of the sound card for digitization and implementation of the ramp action. All other data processing and display of the results will be carried out by the computer.

SODAR ANTENNAE

A more recent approach to design sodar antennas has been to use an array of many smaller elements, perhaps as few as 16 or as many as 100 or more, consisting of piezoelectric/neodymium tweeter drivers and horns. These elements have high efficiencies (up to 45% in the range of operating frequencies), are small in size and lightweight and are widely available. Because of their small size (<10 cm), optimal geometric arrangements with more elements per area are possible. This results in enhanced directivity at the operating frequencies of interest (2 kHz - 4 kHz) in the present work. It has been observed, with reference to earlier studies, that a smaller system made of 16-element array is not quite as 'good' as 36-element array system (having slightly lower heights and less reliable horizontal winds due to a broader beam, though it should be noted that the smaller can be deployed by a single person and is very valuable in studies in complex terrain or emergency situations).

Accordingly, an 8×8 acoustic array antenna was made in three designs with different radiators, namely Phillips model 1X9101, APT-165 model by Ahuja, and MA588 model by M.A. audio. The performance of each array antenna was evaluated. The measurements of each array antenna (forward conversion

efficiency, reverse conversion efficiency and directional response) were systematically conducted. Comparison between three array antennas operated at single 4.2 kHz frequency was made. Finally, the comparison between uniform equi-spaced 8×8 array antenna made with 40 radiators and 52 radiators was also made.

The experiments needed to evaluate the performance of the antennas were conducted inside an anechoic and a reverberation chamber of the Naval Science & Technological Laboratory (NSTL), a DRDO establishment. They consists the measurements of:

- (i) Axial transmit efficiency of the array
- (ii) Axial receive efficiency of the array
- (iii) Directional response of the array in three orientations.

The transmit (electrical to acoustic conversion) efficiency of the antenna has been measured by exciting the antenna at a constant input voltage ($1V_{rms}$) but at different frequency steps (1.9 kHz, 2 kHz, 2.1 kHz), while measuring the sound pressure levels generated on the acoustic axis of the antenna. In this case, a calibrated microphone (1 kHz - 5 kHz) was kept at a distance of 1 meter. The Bruel & Kjare 4189 model microphone was used, which gives a constant response within the range from 1 to 5 kHz.

The receive (acoustic to electrical conversation) efficiency of the antenna was also measured, using a single speaker kept at a distance of 1 meter (closer because of limitation of maximum voltage that could be fed to single element for generation of uniform acoustic pressure). The speaker was fed with varying voltages at different frequency steps (1.9 kHz, 2 kHz, and 2.1 kHz) in order to generate a constant acoustic pressure at the antenna aperture. The open circuit

rms voltage produced was recorded, corresponding to each frequency step, at the antenna terminals.

The directional response of the array was studied in four different orientations. Measurements were made for only three orientations because two of the orientations are symmetrical. In each orientation, the iron rod was moved to the diagonal of the antenna inside the anechoic chamber. The output of the microphone, which was positioned at a distance of 1 meter from the acoustic centre of the antenna on the axis, was measured by recording the generated sound pressure level.

8×8 Planar array with Philips tweeters

An 8×8 array was designed, fabricated and tested using commercially available Philips piezoelectric tweeters. A wooden square pane of length 22.5 cm covered with polyurethane foam was made as the base for embedding the tweeters. The foam was used for absorption of all the reflected acoustic energy. All the tweeters, each having 4Ω impedance, were equally placed on this wooden base. The six corner elements on each side of the array were later removed as part of the optimization of beam parameters such as beam width and side lobe suppression. Thus, a total of 24 elements were eliminated from the 8×8 array of 64 elements. The remaining 40 elements are connected in series and parallel fashion to each other so that a total impedance of 4Ω was achieved for the array. The array was operated at 4.6 kHz and the spacing between centers of adjacent elements was maintained at 3.7 cm, which is equal to half the wavelength. The array structure and outlay are shown in the Figure 1.

The Philips tweeter of model IX9101 was chosen for the array, and its technical specifications are as follows:

Diameter of each tweeter: 2.54 cm

Sensitivity: 97dB/W/m

Impedance: 4Ω

Maximum power of each element: 120 W

Maximum Sound dispersion: 90°

Range of operating frequencies of tweeter: 4 kHz - 21 kHz.

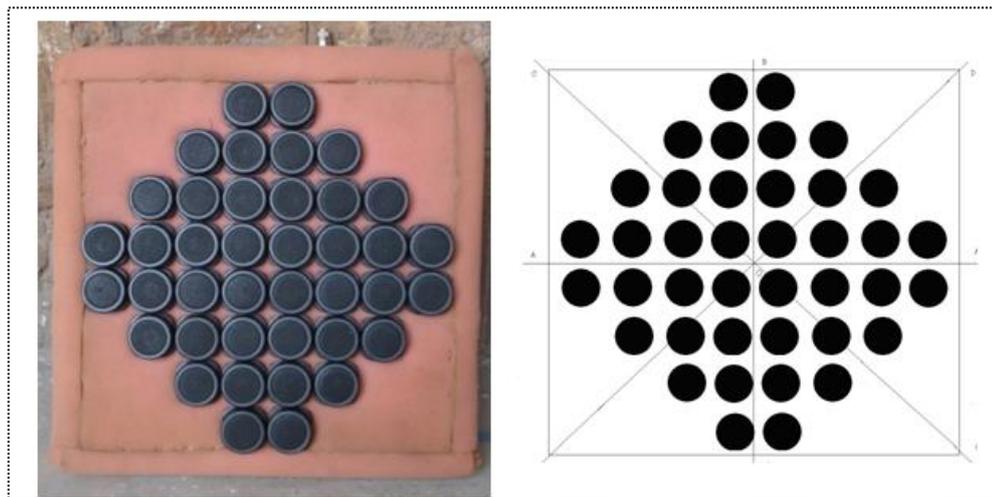


Figure 1 A photograph of the 8×8 array antenna (six elements removed at each corner) fabricated with Philips model IX9101 piezoelectric tweeters.

8×8 Planar array with Ahuja Tweeters

The second acoustic planar array antenna was designed with APT-165 piezoelectric tweeters. The array design was similar to the one described in the previous case (using Philips tweeters) but the dimensions of the wooden pane was increased to $84 \times 84 \text{ cm}^2$ due to larger size of the APT-165 piezoelectric tweeters. Accordingly, the array in this case has been designed for 2 kHz and the corresponding spacing between adjacent elements was fixed at 11 cm. The array antenna structure is shown in Figure 2.

The technical specification of Ahuja tweeters are as follows:

Model : APT-165

Manufacturer : Ahuja

Material : Piezoelectric

Diameter : 110mm

SPL : 93 dB at 2.83 V/m

Power: 300 W for 4 Ω systems and 150 W for 8 Ω systems

Frequencies: 1.8 kHz to 20 kHz

Impedance: 1000 Ω at 1 kHz

Sound dispersion: 45° in vertical or horizontal position

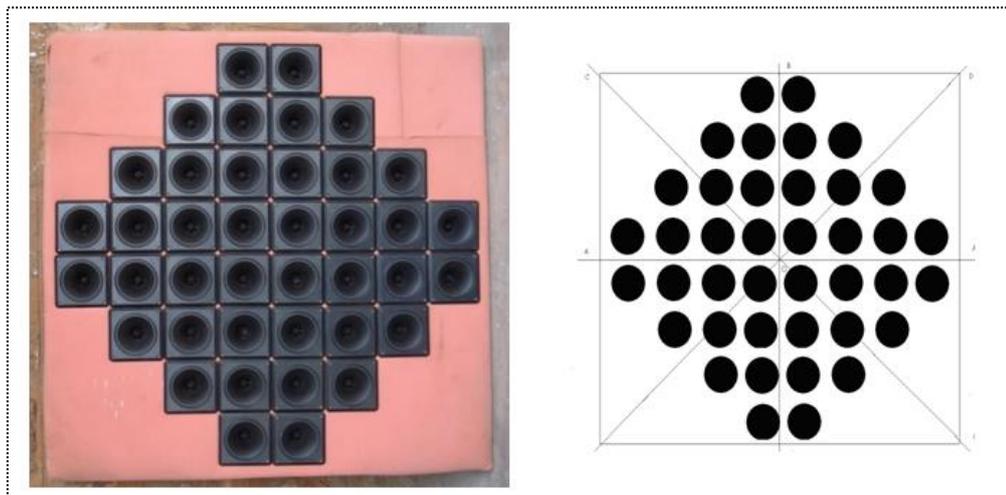


Figure 2 A photograph of the 8×8 array antenna (six elements removed at each corner) fabricated with Ahuja model APT-165 piezoelectric tweeters.

8×8 Planar array with MA588 Tweeters

Figure 3 shows the design of the third acoustic array, which was made with MA588 tweeters manufactured by M.A. Audio. This array was designed for 3.3 kHz. The dimensions of the array are 75×75 cm². The spacing between centers of adjacent elements is 5.1 cm. The features of MA588 tweeters are:

Neodymium Magnet

1" super dome tweeter

continuous input power	30 watts per driver
Peak input power	50 watts per driver
Sensitivity @ 1W/1M	90.2dB
Frequency response	2 kHz - 22 kHz

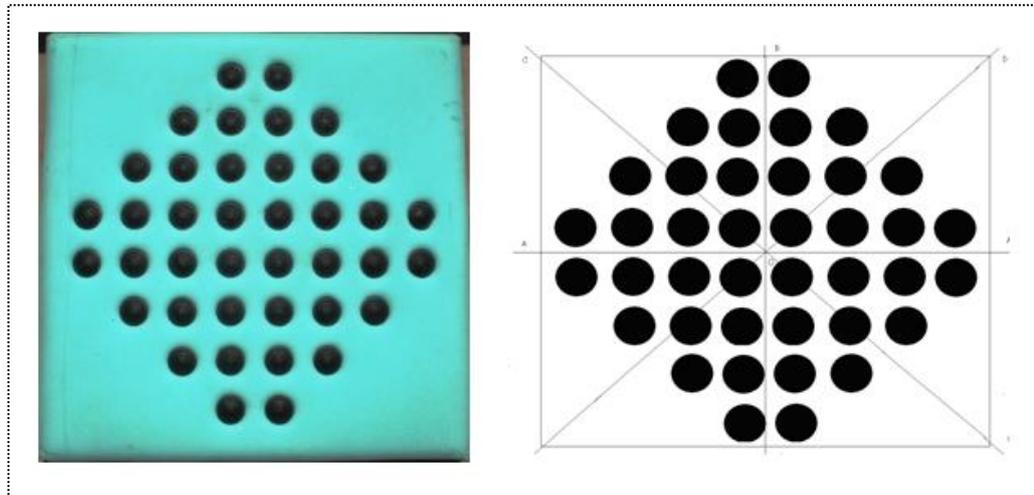


Figure 3 A photograph of the 8×8 array antenna (six elements removed at each corner) fabricated with MA audio model MA588 neodymium magnetic tweeters.

Series-Parallel feeding of array elements

The electrical impedance of each tweeter element is 4Ω . As mentioned earlier, the 40 elements were connected in the series-parallel combination, as shown in Figure 4. This design was simulated using CIRCUITMAKER simulator to obtain the impedance of the array antenna. In the simulations, each of the tweeters was assigned by its actual impedance value of 4Ω only. The results of the simulations gave the antenna design net total impedance exactly as that of a single tweeter at 4Ω . This configuration was used by assuming that it could successfully avoid overloading of the power amplifier, to be used in subsequent measurements, whose output impedance is also 4Ω .

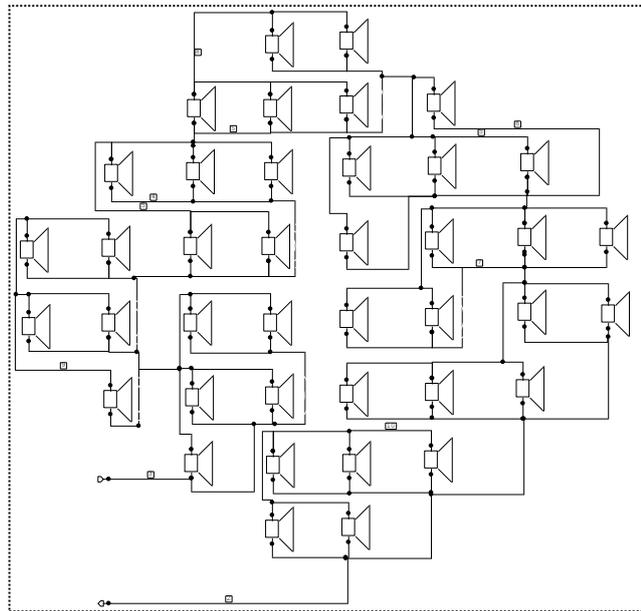


Figure 4 The configuration used to model the array antenna impedance using the CIRCUITMAKER software tools.

Experimental setup in anechoic chamber

The basic method of measuring the normalized power of any acoustic antenna is to measure its transmitting intensity by varying its elevation angle (θ) and keep the azimuth angle (Φ) constant.

These measurements should be performed in free field conditions. The anechoic chamber in the Naval Science and Technological Laboratories, a DRDO laboratory of the Ministry of Defense, Government of India, Visakhapatnam, was made available to us for making the measurements for the characterization of the array antennas. The procedure adapted in the experiment is described with the block diagram in Figure 5. An anechoic chamber is a closed room in which there are no echoes. In a well-designed acoustic anechoic chamber, the equipment under test will only receive signals which were emitted directly from the signal source, and not reflected from any other part of the chamber.

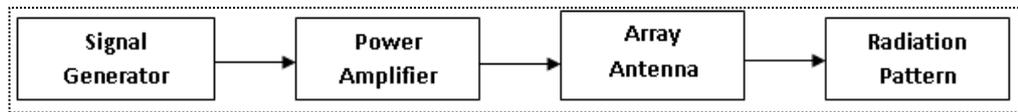


Figure 5 Block diagram of the experimental setup for the measurement and characterization of the array antennas.

The array was fed with an amplified voltage from a commercial Ahuja-make audio power amplifier. The power amplifier is driven by a frequency generator. A sensor receives the transmitted sound energy emitted by the array by placing the microphone (1 kHz - 5 kHz) at different angles of elevation from 0° to 180° at intervals of 5° . The experimental setup inside the anechoic chamber is shown in Figure 6.

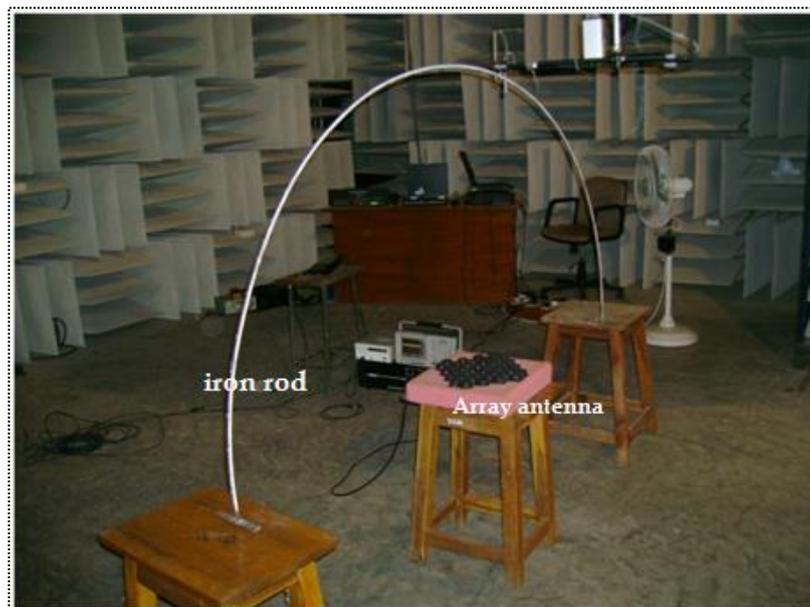


Figure 6 A view of the experimental setup inside the anechoic chamber in the Naval Science & Technological Laboratory (one of the DRDO network of laboratories), Visakhapatnam.

An iron rod of semi-circle shape with a diameter of 1 m was marked with elevation angles from 0° to 180° for this purpose. The sensor was placed manually at each angle and the relative intensity of array at that position was

measured using Type 2250 Investigator Sound Level Meter (SLM) connected to the microphone. It can be seen from Figure 6 that the antenna array was exactly positioned at the middle of the semi-circle (at 90°) so as to minimize or eliminate the errors in the measured pattern.

The reverse conversion efficiency of the array antenna was measured in the reverberation chamber of the Naval Science & Technological Laboratory, Visakhapatnam. During the experiment, the acoustic source and the array were kept at a standard distance of 1 m. The received power of the array was measured when a certain frequency from the source was transmitted. The source was fed by a frequency generator through an audio power amplifier, as shown in Figure 7.

The transmit efficiency response has been measured by positioning the microphone at a point one meter from the acoustic centre of the array, along its axis. Measurements are carried out at different positions of the microphone and its corresponding sound pressure level reading at three different frequency steps at 1.9 kHz, 2 kHz, and 2.9 kHz. For all the three frequencies, the first column of the table indicates elevation angle, the second column indicates SPL meter reading in dBs and the third column indicates normalized power. As expected, the SPL meter reading in dB has been the highest for all the three frequencies at 90° . The normalized power was deduced by subtracting the SPL meter reading of the highest intensity from the SPL meter reading at each elevation angle.

The variations of normalized power as a function of elevation angle were further drawn to demonstrate the trends of radiation patterns of Ahuja array

antenna with 40 radiators for three different frequency steps of 1.9 kHz, 2 kHz and 2.1 kHz.



Figure 7 Photograph showing the experimental setup inside the reverberation chamber in the Naval Science & Technological Laboratory, Visakhapatnam.

Figure 8 shows the measured radiation pattern of the antenna fabricated with Ahuja APT-165 tweeter elements at three frequencies. It is observed that the radiation pattern at 2 kHz produced a relatively narrow beam of 16° width. The side lobe level is also at about -12 dB. The radiation pattern at the other two frequencies, namely 2.1 and 2.2 kHz, found to be much broader at more than 20° . The side lobe levels are also high at -6 dB. This is a good result as higher level side lobes are unwanted in antenna engineering.

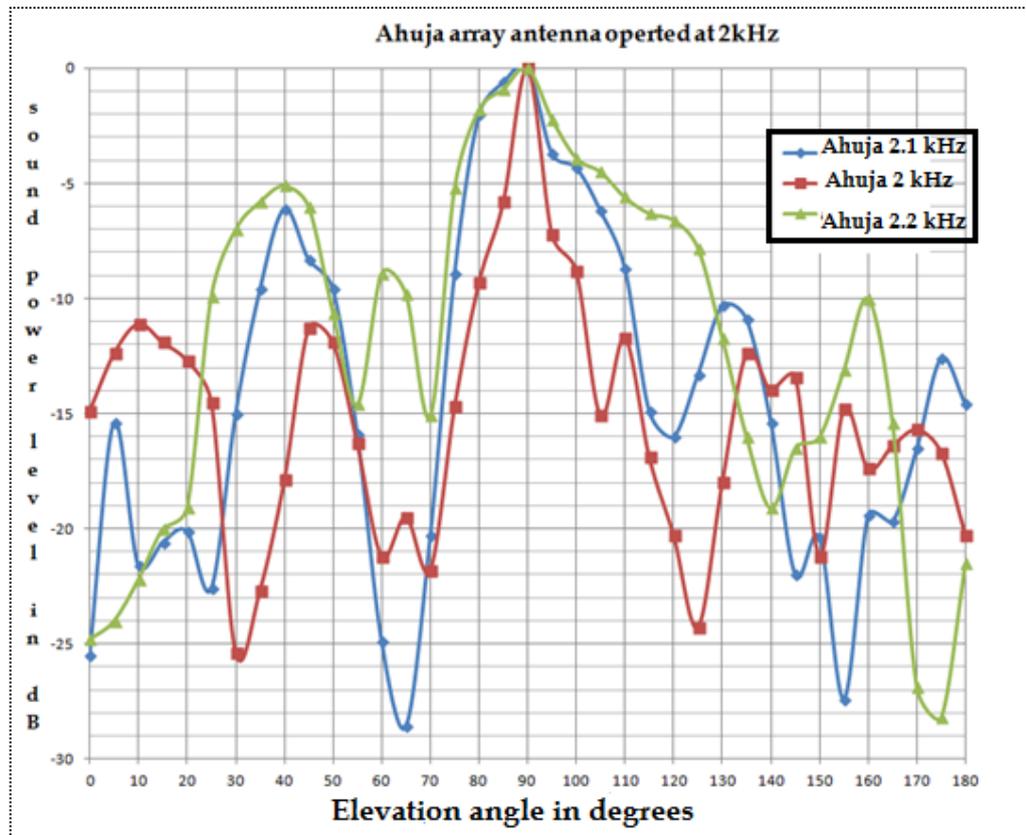


Figure 8 The measured radiation pattern of the array antenna made with APT-165 piezoelectric tweeters without six elements at each corner ($6 \times 4 = 24$).

The Figure 9 shows the radiation pattern of Philips array antenna with 40 radiators in three different frequency steps: 4.5 kHz, 4.6 kHz and 4.7 kHz. The pattern at 4.5 kHz has a beamwidth of 16° and the side lobe level of -19 dB. The pattern at 4.6 kHz has a beam width of 14° and the side lobe level is at -18 dB. At 4.7 kHz pattern, the beamwidth is 16° and the side lobe level is -12 dB.

The Figure 10 shows the radiation pattern of MA588 array antenna with 40 radiators at three different frequencies: 3.2 kHz, 3.3 kHz and 3.4 kHz. It is observed that the radiation pattern at 3.3 kHz shows a narrow beam with a width of 15° and a side lobe at -16 dB. This is considered as one of the best outputs. The beamwidth at 3.2 kHz is observed as 20° with the first side lobe at

- 14 dB. The radiation pattern at 3.4 kHz is measured as 18° with the first side lobe at -16 dB.

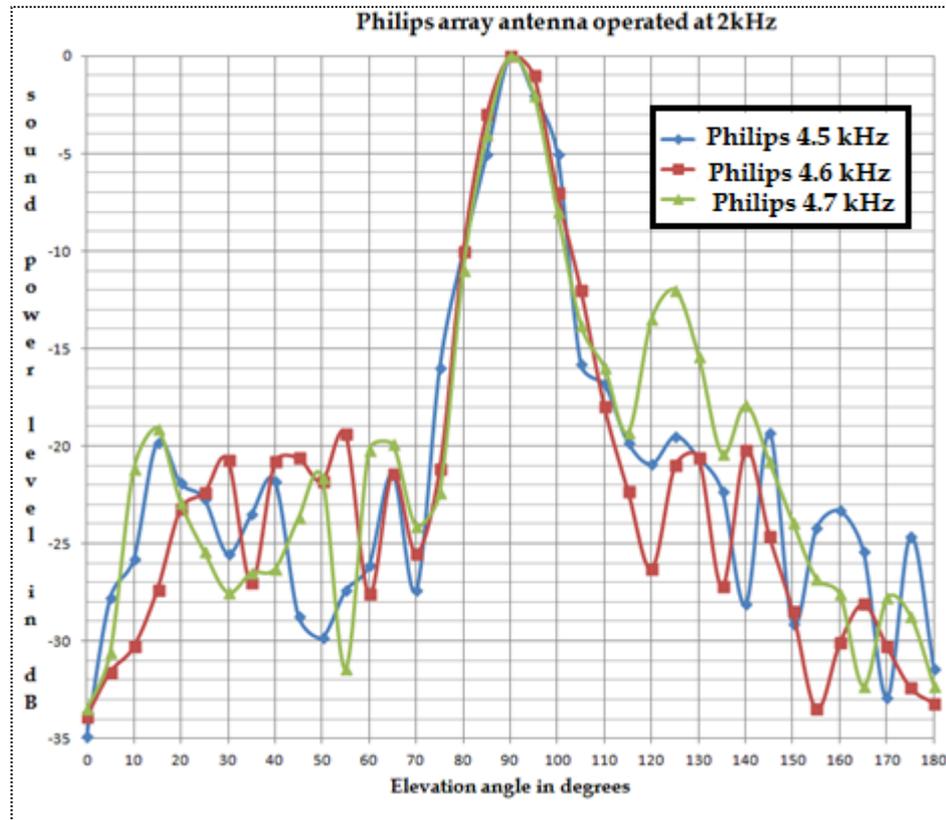


Figure 9 The measured radiation patter of the array antenna made with Philips model IX9101 piezoelectric tweeters without elements numbering six at each corner.

As per the specifications mentioned earlier for the tweeter elements used to fabricate all the three antennae, array made with Ahuja tweeters works in the frequency range between 1.8 kHz and 20 kHz. Similarly, the array with MA audio works between 2 kHz and 20 kHz, while the other one made with Philips tweeters works between 4 kHz and 21 kHz. We have made an attempt to operate all these three antennae at a specific frequency within these ranges and compare their performances. Accordingly, these three antennae were operated

at a single frequency of 4.2 kHz. The normalized polar radiation patterns of all the three antennae at 4.2 kHz are shown in Figure 11.

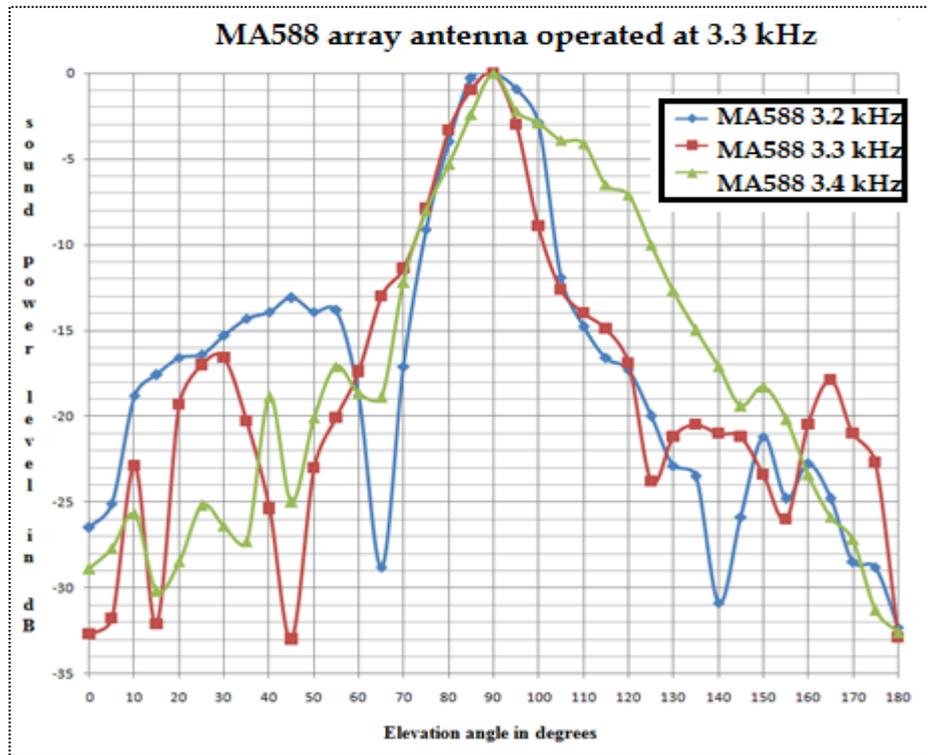


Figure 10 The measured radiation patter of the array antenna made with MA audio MA588 neodymium magnetic tweeters without elements numbering six at each corner.

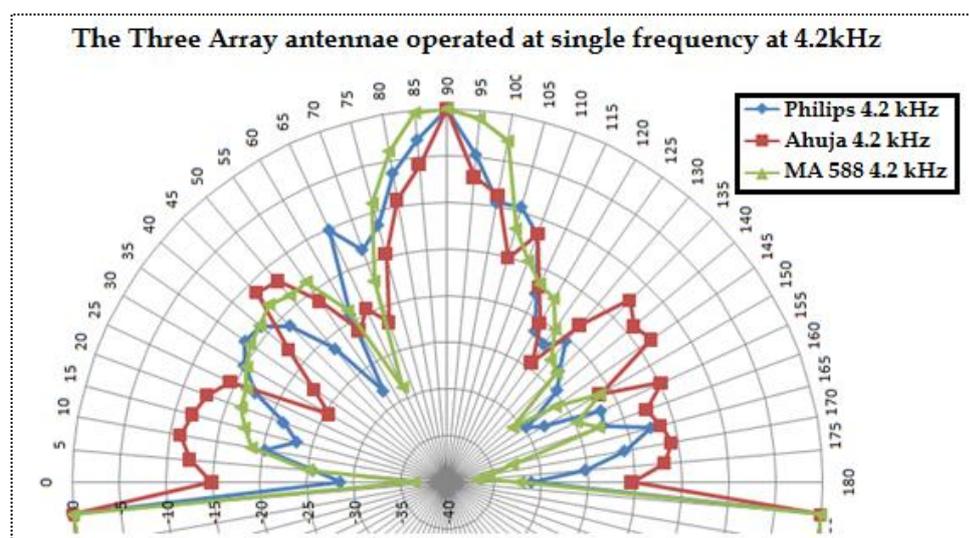


Figure 11 The polar plot showing the radiation pattern of the array antennas made with the three different tweeters (Philips, Ahuja, and MA audio) at 4.2 kHz.

It can be inferred from Figure 11 that the Ahuja and Philips antennae provide narrow beamwidth, while the side lobe suppression is more effective for the antenna with MA588 tweeters. Therefore, efforts were made further only on MA588 antenna design to improve its performance.

Since there exists a general impression that the performance of the antenna increases when there are more number of elements in the design. Thus, in order to verify whether such a proposition holds good or not for our antenna design, we have designed our MA588 antenna by varying the number of elements. Two different designs with 40 and 52 elements were made, as shown in Figure 12, and compared their performances at the frequency of 3.3 kHz. The distance between any two adjacent radiators is maintained at 0.5λ .

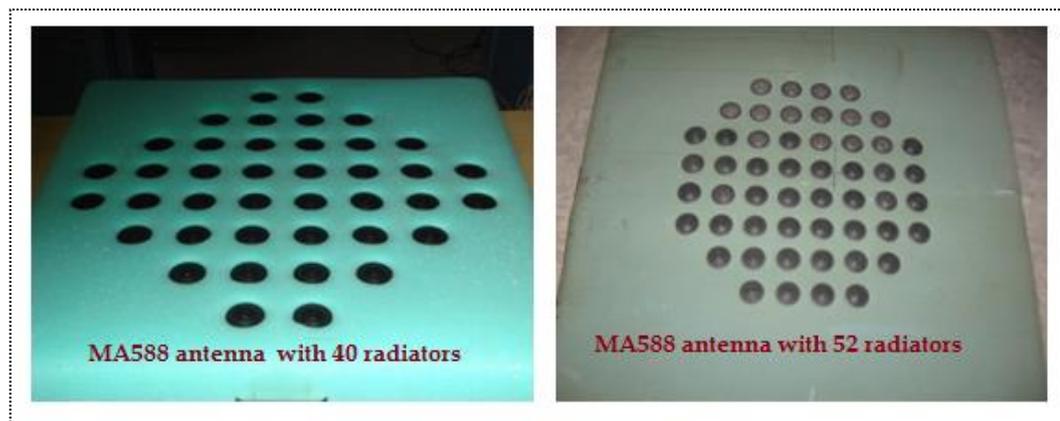


Figure 12 Photographs showing two array antennas made with 40 elements (left) and 52 elements (right) using MA588 tweeters.

Figure 13 shows the radiation patterns of 40 and 52 radiators made by MA588 operated at 3.3 kHz frequency. The 40 radiators array antenna had beamwidth of 12° and the side lobe level -10 dB. At the same time, the array consisting of 52 elements provided a beamwidth of 14° with a side lobe level at -16 dB.

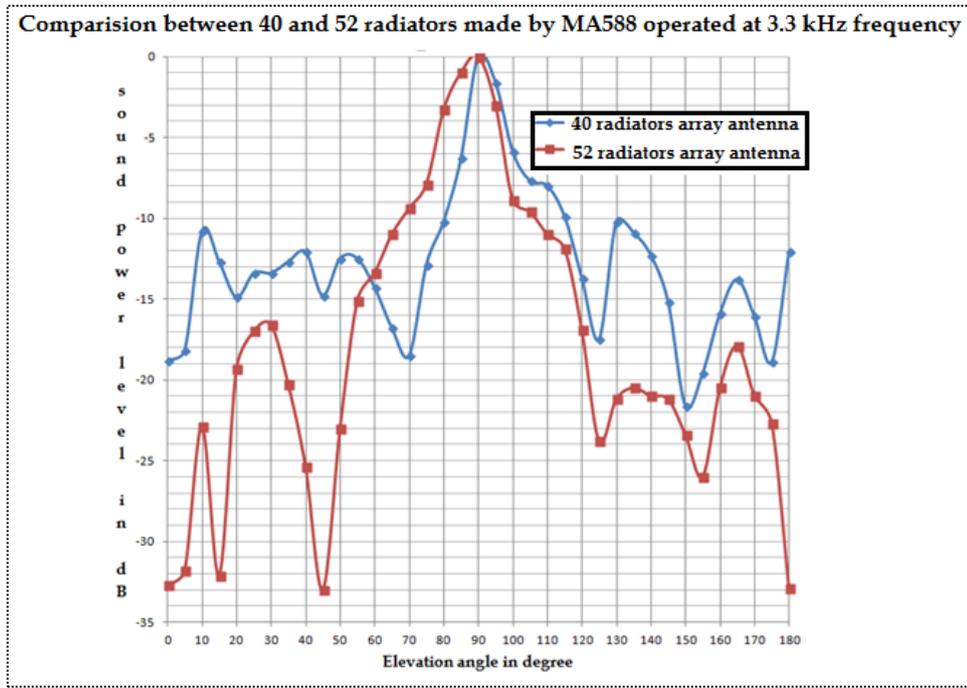


Figure 13 Radiation patterns of the array antennas made with 40 and 52 tweeters (MA588 neodymium).

The directional response of the array was measured in three different orientations, as shown in Figure 14. Since the orientations AA¹ and CC¹ are symmetrical to each other, they are considered as one and BB¹ and DD¹ are the other two orientations.

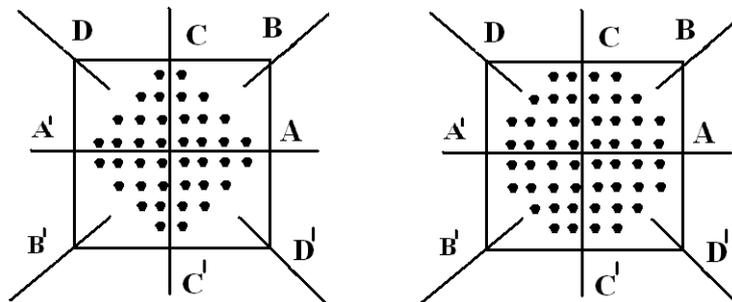


Figure 14 Diagram showing the orientations in which the directional response measurements were carried out on the two antennas with 40 and 52 elements of MA588 model.

The Figure 15 shows the polar radiation pattern of MA588 array antenna with 52 radiators operated at 3.3 kHz. The radiation patterns for the orientations AA' and BB' showed more or less similar with side lobes at much lower levels, thus indicating larger directivity for these orientations. On the other hand, the DD' orientation resulted in a strong gating lobe in between the elevations angles 15-45°, but still appears to have shown better directivity.

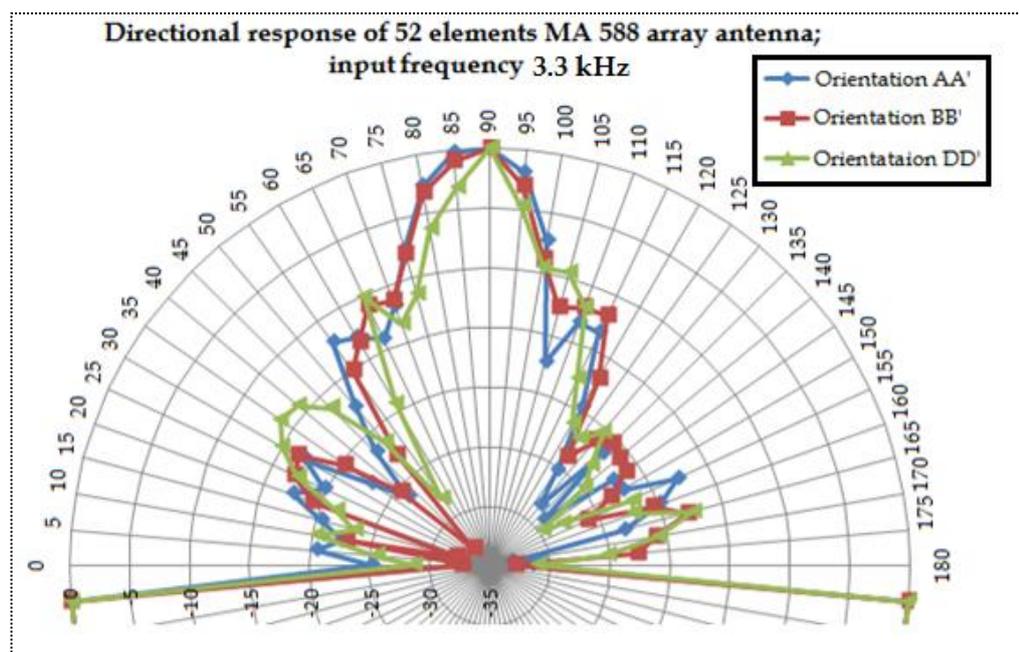


Figure 15 Radiation patterns of the 52-element array antenna measured in the three principle orientations at 3.3 kHz (AA', BB' and DD').

Figure 16 shows the polar radiation pattern of the array antenna fabricated with MA588 tweeters numbering 40 operated at 3.3 kHz. In this case, the radiation patterns for all the orientations, AA', BB' and DD', showed relatively strong side lobes leading to lesser degree of directivity; thus justifying the principle of increased degree of directivity with increased number of radiation elements in the array.

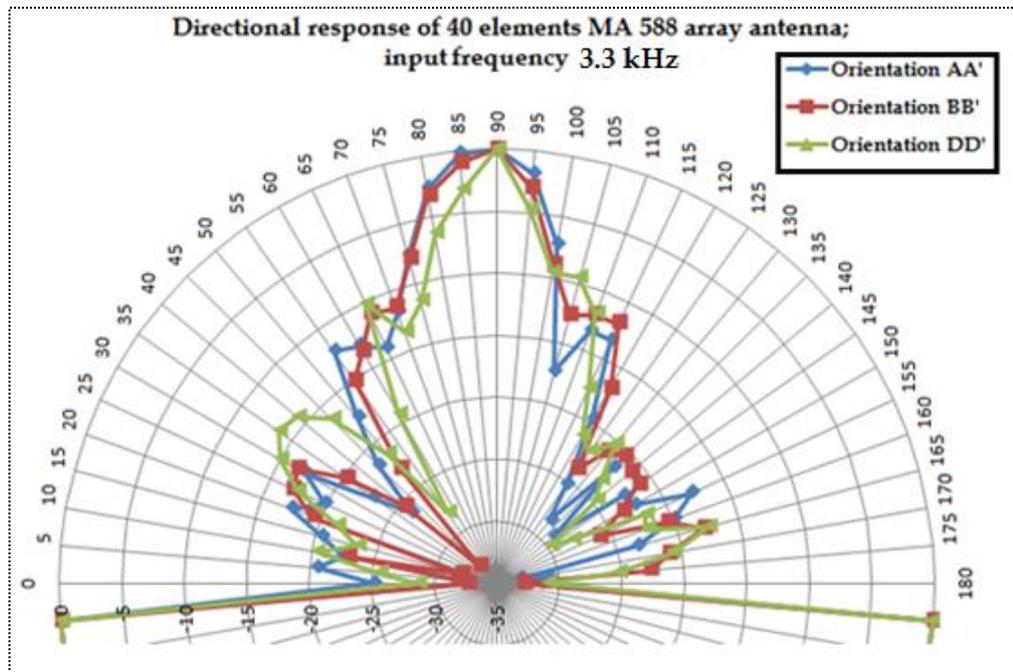


Figure 16 Radiation patterns of the 40-element array antenna measured in the three principle orientations at 3.3 kHz (AA', BB' and DD').

A comparison study of the conversion efficiency of the three antennae fabricated with 40 elements, each with Philips, Ahuja and MA588, is made. As pointed earlier, the three antennae were tested at three different frequencies. The spacing is kept at 0.5λ for the three antennas. It is found that the antenna made with MA588 elements provided good conversion efficiency. A further examination of array antenna made with 40 and 52 elements of MA588 was carried out. It is found that 52-element array has better conversion efficiency and directional characteristics.

Figure 17 shows a few photographs taken during the visit of our Vice-Chancellor Prof G S N Raju, who himself is an expert in antenna design, on 30th January 2014. The Principal Investigator, Prof M Purnachandra Rao is explaining about the new antennae.



Figure 17 The three antennas are kept on display during the visit of our Vice-Chancellor Prof G S N Raju on 30th January 2014.

All the configurations of the array antennae presented up to now have a common problem. They fail to work during rain. In fact, the tweeter elements are not designed for all-weather operation. Another configuration of the type reflector-array is designed and tested with 5×5 tweeter elements. The operating frequencies are limited between 1,900 and 2,100 Hz. The basic design structure is shown in Figure 18. The antenna and the reflector are placed at, respectively, 70° and 55° with respect to the ground plane. This design ensures that the sound waves transmitted by the antenna are projected on to the reflector. The arrangement of the reflector is made in such a way that the sound waves travel straight upwards into the atmosphere after reflection. The echo-signals from the atmosphere first hit the reflector and then on to the array antenna. This arrangement protects the piezoelectric tweeters from rain and dust.

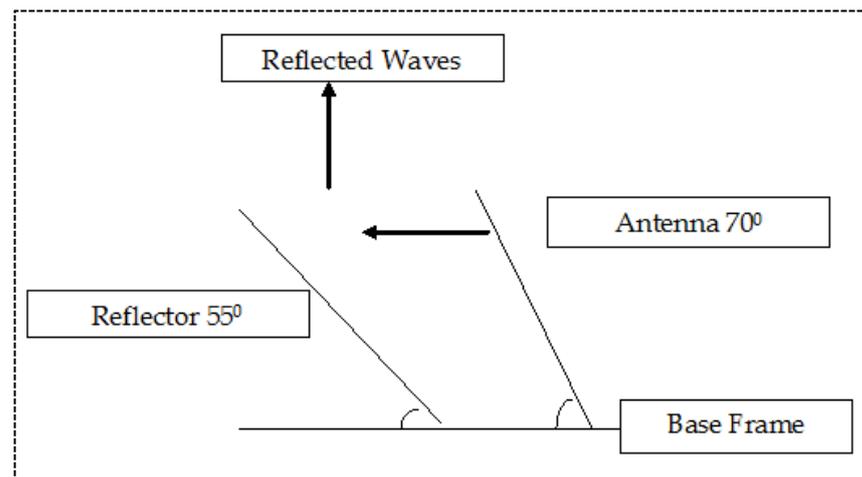


Figure 18 Reflector-array antenna structure with respect to base frame.

The reflector-array antenna is bulkier but it is recessed into a rain shield. This design is perhaps a little less susceptible to rain impact noise. Another advantage of this type antenna is suitable for beam-forming applications like electronics beam steering.

The 5×5 array is fabricated with Ahuja APT-165 piezoelectric tweeters. All the tweeters are equi-spaced on a wooden base of 25×25 cm². One element at each corner of the array is removed as part of the optimization of beam parameters such as beam width and side lobe suppression. Thus, a total of 4 elements are removed from the 5×5 array of 25 elements. The remaining 21 elements are connected in series and parallel fashion to each other so that a total impedance of 8Ω is achieved for the array (Figure 19).

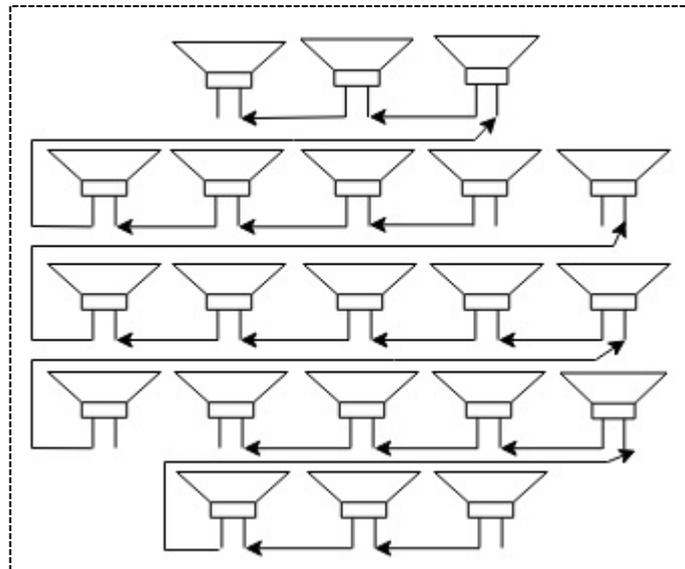


Figure 19 The series-parallel electrical connection of the 5×5 reflector-array antenna fabricated with APT-165 piezoelectric tweeters.

The array is designed for operation at 2 kHz and the corresponding spacing between adjacent elements is fixed at 11 cm. The geometry and the assembled array structure are shown in Figure 20.

The experimental arrangement utilized for the measurement of radiation parameters of the array antenna is shown in Figure 21. The antenna assembly is placed in the middle of the semi-circle iron rod to avoid errors caused by misplacement. The reverse conversion efficiency of the array antenna is

measured in the reverberation chamber. Figure 22 shows the experimental arrangement for the said measurements.

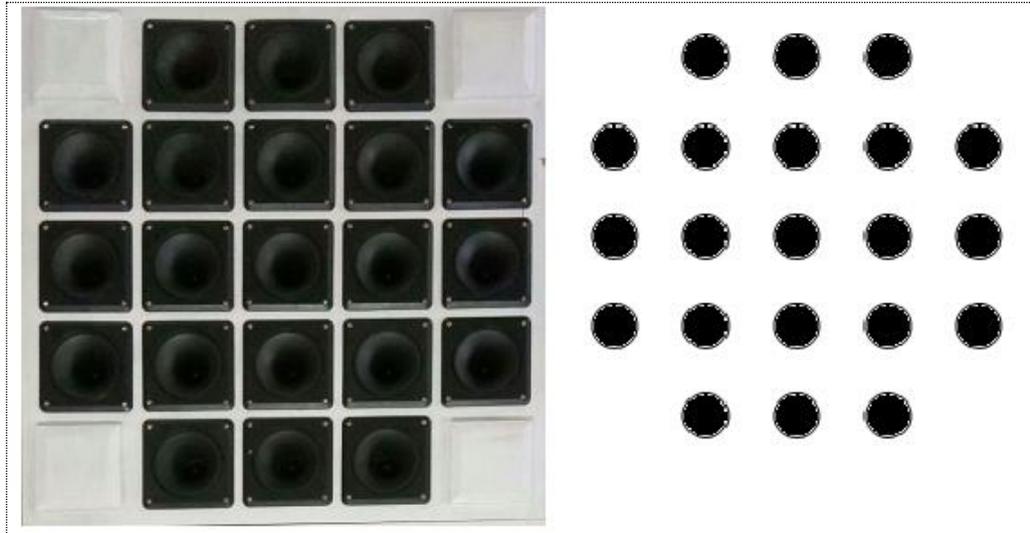


Figure 20 The assembled array antenna with APT-165 piezoelectric tweeter elements (left) and its physical layout (right).

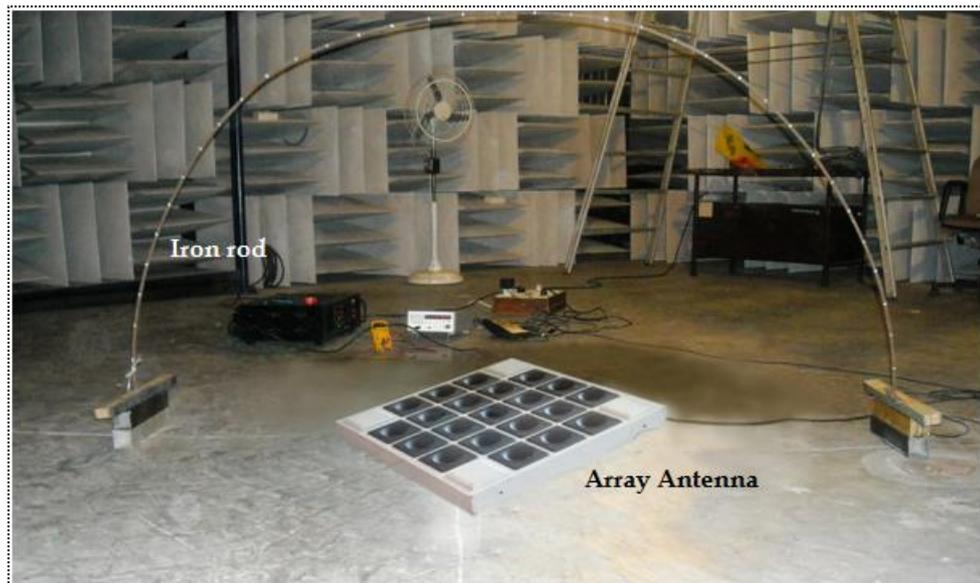


Figure 21 Picture showing the experimental setup for the measurement of the radiation parameters in the anechoic chamber.



Figure 22 Picture showing the experimental arrangement used in the reverberation chamber for the measurement of reverse conversion efficiency of the array.

Although the array antenna is designed for operation at 2 kHz, measurements to determine the radiation pattern are conducted at 1.9 kHz, 2 kHz, and 2.1 kHz. The variations of normalized power, as a function of elevation angle, are shown in Figure 23. The 1.9 kHz pattern has the beam width of 17° and the side lobe level at -6 dB.

On the other hand, the radiation pattern at 2 kHz is marked by a relatively narrow beam width of 10° and the side lobe level at -15 dB. This is the most desirable combination for sodar application. The radiation pattern at 2.1 kHz has a beam width of 20° and side lobe level of - 5 dB. Figure 24 shows the radiation pattern of the array plotted on a polar graph.

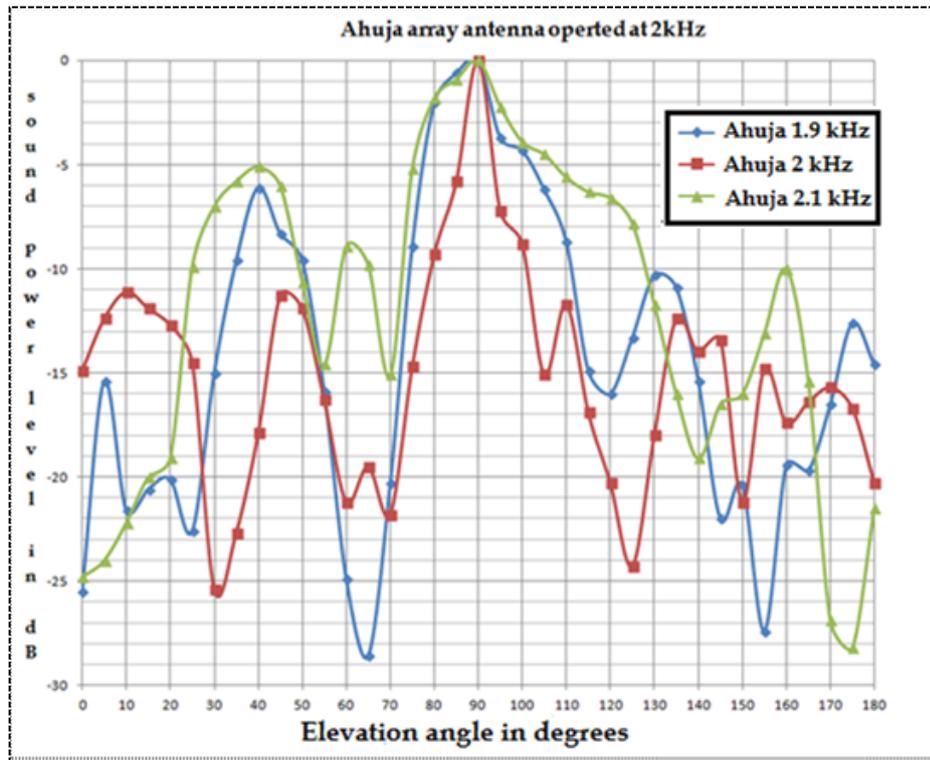


Figure 23 Measured radiation pattern of the 5×5 array antenna with APT-165 piezoelectric tweeters at 1.9 kHz, 2.0 kHz and 2.1 kHz.

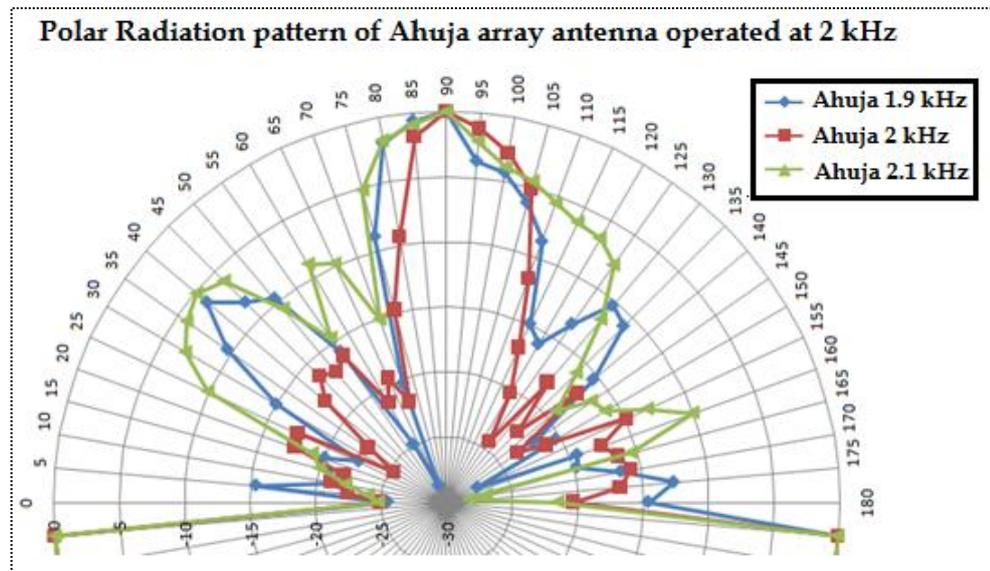


Figure 24 Polar plot showing the measured radiation pattern of 5×5 array antenna made with APT-165 piezoelectric tweeters at 1.9 kHz, 2.0 kHz, and 2.1 kHz.

The 5×5 array antenna is operated at 1.9 kHz, 2 kHz, and 2.1 kHz. The sound source is maintained in reverberation chamber at approximately 100 dB. The calculated reverse conversion efficiency values are shown in Table 1. The conversion efficiency is higher at 2.0 kHz which is the operating frequency.

Table 1 Reverse conversion efficiency of the 5×5 array antenna

Frequency	RMS voltage	Source intensity	Array intensity	Efficiency (%)
1.9 kHz	0.32 V	100 dB	14.08 dB	14.08
2.0 kHz	0.35 V	99 dB	14.86 dB	15.01
2.1 kHz	0.30 V	99 dB	13.52 dB	13.65

The reflector-array antenna assembly is shown Figure 25. A hylam sheet of 8 mm thickness is used as the reflector. The experimental setup for the measurement of radiation pattern and other parameters of the reflector-array antenna is shown in Figure 26.



Figure 25 Photograph of the reflector-array antenna assembly with hylam sheet of 8 mm thickness.

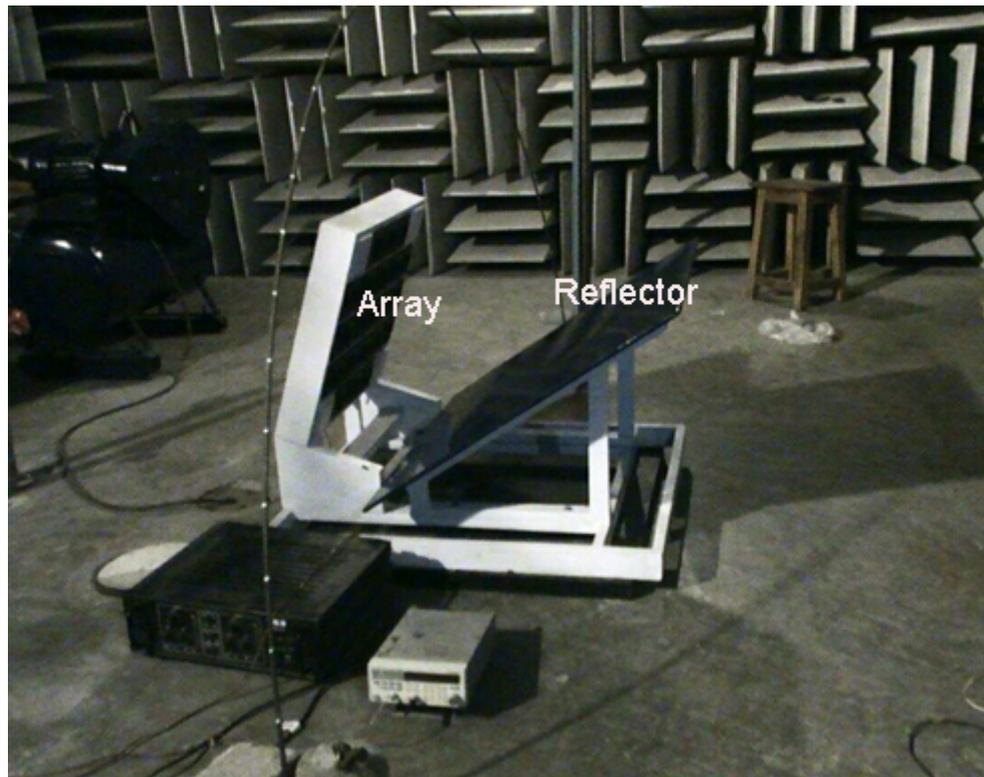


Figure 26 Picture showing the experimental setup for the measurement of radiation parameters of the reflector-array antenna.

The measured radiation pattern of the reflector-array antenna is shown in Figure 27. It is observed that the beam width remained almost static even with the newly introduced reflector configuration. The polar plot for the same observations is given in Figure 28.

The directional response of the reflector-array antenna at 2 kHz is shown in Figure 29. It is observed the beam width is about 15° and the side lobe level is at about -10 dB for BB' orientation. The directional response for the other two orientations is also similar.

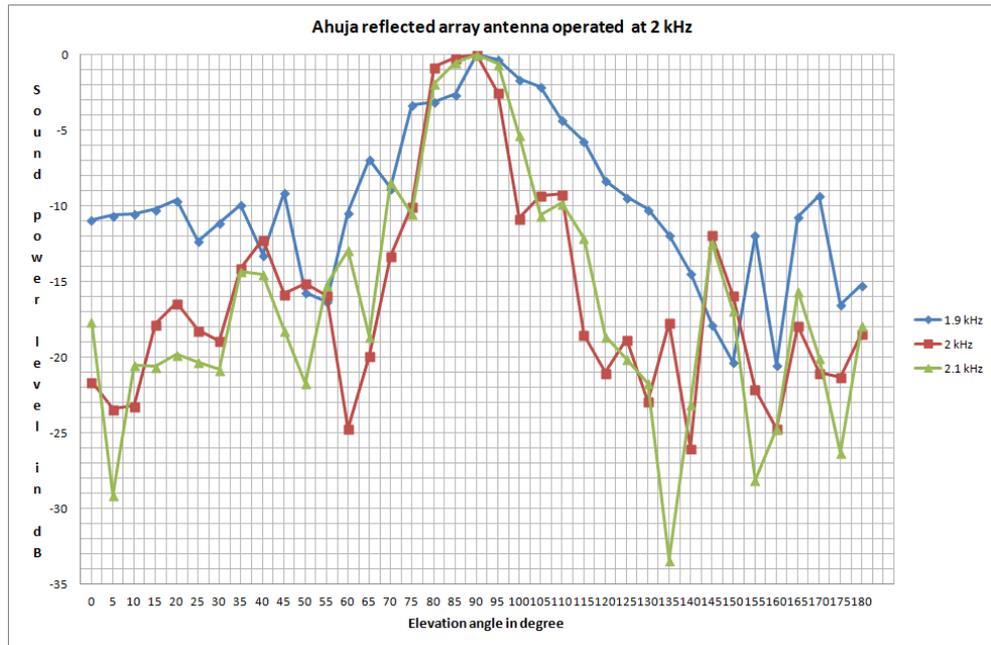


Figure 27 Measured radiation pattern of the reflector-array antenna with APT-165 piezoelectric tweeters at 1.9 kHz, 2.0 kHz and 2.1 kHz.

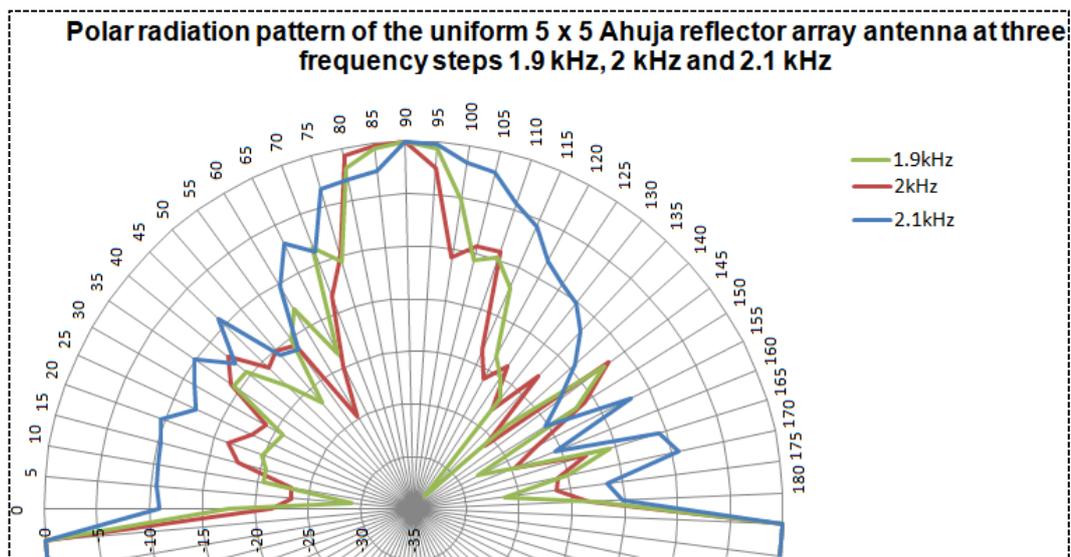


Figure 28 Polar plot of the measured radiation pattern of the reflector-array antenna at the three designated frequencies in the polar plot.

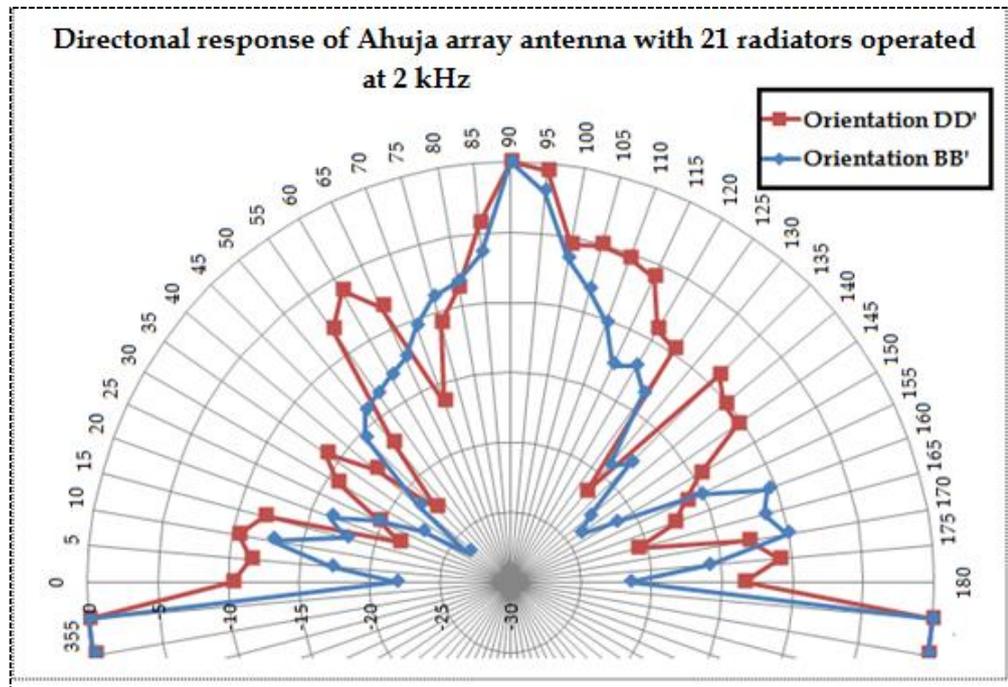


Figure 29 Measured directional response of the reflector-array antenna made with APT-165 piezoelectric tweeters.

In general, the environmental noise and other human-made noise interfere with the operation of the Doppler sodar system. The noise by strong wind is another important factor degrading the quality of the echo-signal. It is customary to have an acoustic enclosure around the antennae to take care of all the noise factors mentioned above. In the present experiment also, each reflector-array antenna is placed inside an acoustic enclosure made with water-proof plywood sheets (Figure 30). The enclosure has four sides tied with iron anglers at each of the four corners. Before assembling the enclosure as one unit, one side of each panel is painted with tar and then fixed with tar felt. A layer of glass wool of thickness of about 10 cm is placed on the tar felt. A generic polyethylene mesh is fixed with wooden bars above all these layers. This arrangement is carried out to avoid sagging due to rains and also aging (Figure 31).

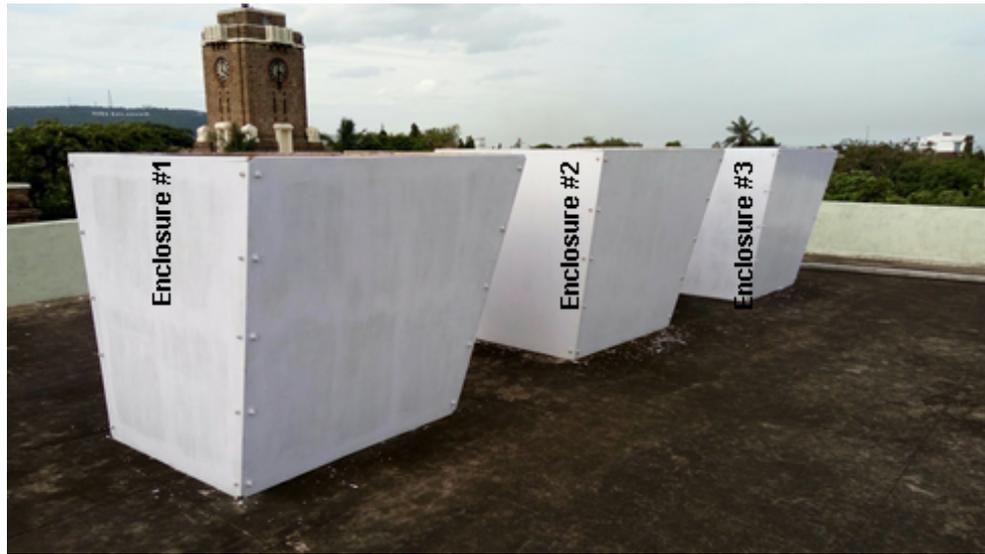


Figure 30 Picture showing the acoustic enclosure made with water-proof plywood sheets (JVD building tower clock in the background).



Figure 31 Picture showing the acoustic enclosure built with water-proof plywood sheets of 1.8 m height.

The reflector-array antenna placed inside the acoustic enclosure is shown in Figure 32. The measurements of the atmospheric boundary layer reported later in this report are made with this antenna.



Figure 32 The new reflector-array antenna inside the acoustic enclosure. The whole assembly is installed on the terrace of the building of the Department of Systems Design.

SODAR ELECTRONICS

A state-of-the-art electronics sub-unit system is developed and experimented for the tri-axial monostatic Doppler sodar system. All the analog and/or digital electronics circuitry that have been used for the generation of the three operating frequencies including a crystal oscillator, divide-by counters, wave-shaping circuits, gating circuits and the time control circuits, etc are dispensed with a suit of new software tools that utilize the on-board sound card resources in the present unit.

Transmitter

Rapid advancements in the microelectronics and associated developments in digital world including PCs and other class of computing technologies have changed the way the sodar transmitters looked and worked till a few years ago. The present-day sodar transmitters have less analog and/or digital circuitry and have more software tools. Until recently, DSP cards plugged into the bus of a PC-class computer mimicked the classical crystal oscillator, wave-shaper, gating and timing control circuitry of a typical Doppler sodar system. The block diagram of such a system is shown in Figure 33.

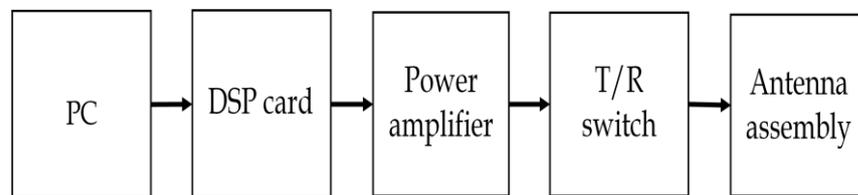


Figure 33 Block diagram of the transmitter unit in the Doppler sodar system.

It is hoped that the resources of sound cards are used to replace the so-called digital signal processor cards, thus, reducing the cost and complexity of the Doppler sodar systems. The transmitter-chain now consists of a multimedia PC or a laptop computer, the power amplifier, the T/R switch and the antenna assembly. The block diagram of the system is shown in Figure 34.

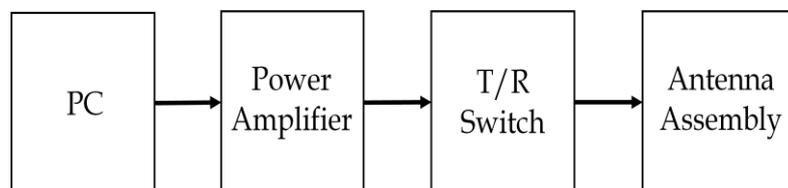


Figure 34 Simplified block diagram of the transmitter of the Doppler sodar system.

A suit of software tools are developed for the generation of the three transmitter signals using the on-board soundcard. It also provides the synchronization between the transmitter and receiver units. The software tools are developed in Microsoft Visual C++ programming language using multi-thread technology.

Direct digital synthesis: A variety of synthesizing techniques are in vogue. The selection is a tradeoff among the system parameters, system complexity, and cost. Digital methods use, in principle, a technique known as direct digital synthesis (DDS). The technique is sometimes referred to as numerically controlled oscillator.

DDS designs fall into six major categories: pulse output DDS, fractional divider or pulse snatching DDS, sine output DDS, triangular output DDS, phase interpolation DDS, and jitter injection DDS (Reinhardt *et al.*, 1986; Rahim *et al.*, 2006). DDS offers fast switching speed, high-resolution (the step size of the synthesizer), small size and low power, good economics, good reliability and good reproducibility of digital design. Since the signal is manipulated digitally, it is easy to modulate and achieve accuracies not attained by analog techniques. It is also convenient to interface with the computing machines that usually control the synthesizers.

In its basic form, a DDS consists of a precision reference clock, an address counter, a programmable read only memory (PROM), and a DAC (Figure 35). The digital amplitude information that corresponds to a complete cycle of the sine wave is stored in the PROM. The PROM acts as a sine wave lookup table. As the address counter steps through each memory location, the corresponding digital amplitude of the signal at each location drives a DAC which, in turn,

generates the analog output signal. A fundamental problem with this basic DDS system is that the output frequency can be changed only by changing the reference clock frequency or by reprogramming the PROM. Neither of these options supports high-speed output frequency hopping.

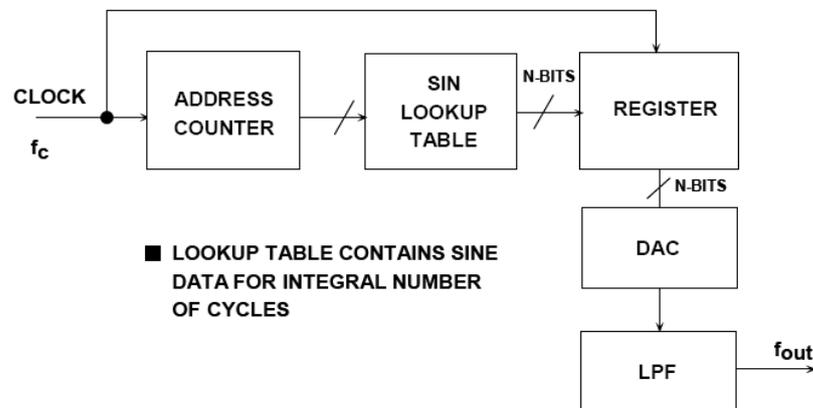


Figure 35 Simplified block diagram of a direct digital synthesizer.

With the introduction of phase accumulator function into the digital signal chain, this architecture becomes a numerically-controlled oscillator which made the DDS more flexible. This is achieved by replacing the address counter with an n-bit variable-modulus counter and phase register before the sine lookup table. The modified DDS architecture acts this as a 'phase wheel'. This is best understood by visualizing the sine wave oscillation as a vector rotating around a phase circle (Figure 36). Each designated point on the phase wheel corresponds to the equivalent point on a cycle of a sine waveform.

As the vector rotates around the wheel, visualize that a corresponding output sine wave is being generated. One revolution of the vector around the phase wheel, at a constant speed, results in one complete cycle of the output sine wave.

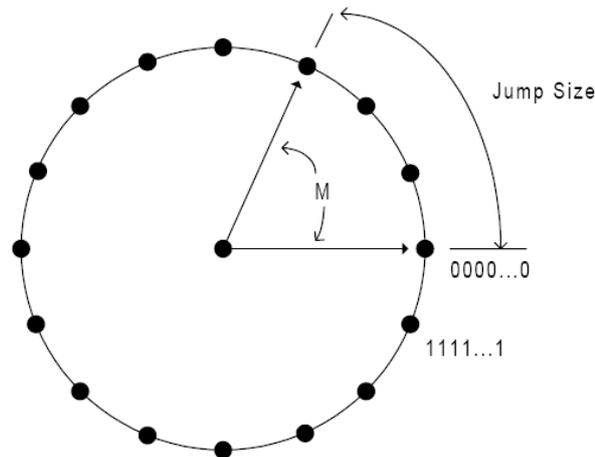


Figure 36 Diagram explaining the principle involved in the digital phase wheel.

The phase accumulator provides the equivalent of the vector's linear rotation around the phase wheel (Figure 37). The contents of the phase accumulator correspond to the points on the cycle of the output sine wave. The number of discrete phase points placed in the phase wheel is determined by the resolution, n , of the phase accumulator. The output of the phase accumulator is linear and cannot be directly used to generate a sine wave except a ramp. A phase-to-amplitude lookup table is used to convert a truncated version of the phase accumulator's instantaneous output value into the sine wave amplitude information that is presented to the DAC. The signal flow through DDS architecture is depicted in the Figure 37.

Most DDS architectures exploit symmetrical nature of a sine wave and utilize mapping logic to synthesize a complete sine wave cycle from $\frac{1}{4}$ cycle of data from the phase accumulator. The phase-to-amplitude lookup table generates all the necessary data by reading forward-backward through the lookup table.

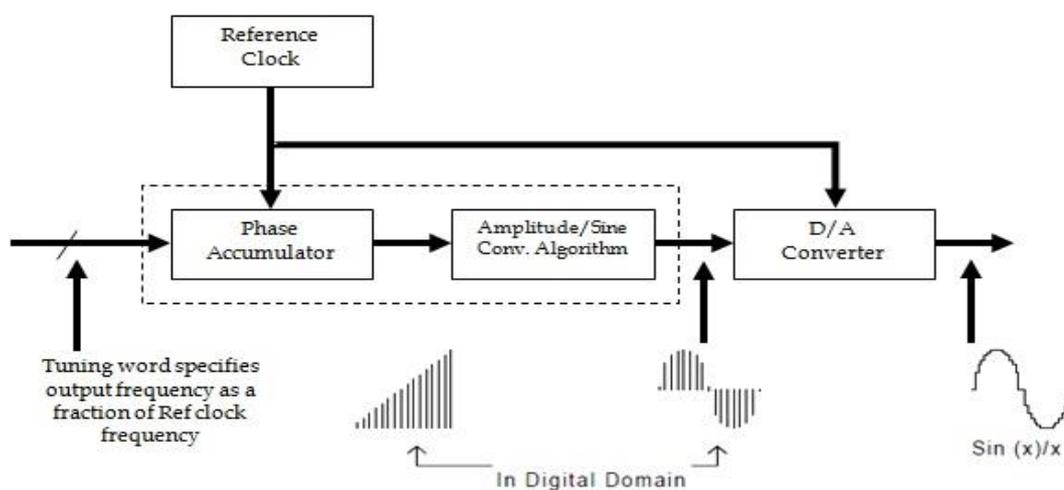


Figure 37 Diagram showing the signal flow through the DDS system.

The phase accumulator is actually a modulo-M counter that increments its stored number each time a clock pulse is sent. The magnitude of the increment is determined by a binary-coded input word M contained in a delta phase register that is summed with the overflow of the counter. The word in the delta phase register forms the phase step size between reference clock updates; it effectively sets how many points to skip around the phase wheel. The larger the jump size, the faster the phase accumulator overflows and completes its equivalents of a sine wave cycle. For $n = 32$ -bit phase accumulator, an M value of 0000.....0001 (one) would result in the phase accumulator overflowing after 2^{32} reference clock cycles (increments). If the M value is changed to 0111.....1111, the phase accumulator will overflow after only 2^1 clock cycles, or two reference clock cycles. This control of the jump size constitutes the frequency tuning resolution of the DDS. The changes to the value of M result in changes in the output frequency.

Pulse output DDS: The pulse output DDS is the simplest of the DDS categories (Reinhardt, 1993). An n -bit accumulator set up to add a frequency word once

every clock period (Figure 38). The accumulator will overflow, on the average, once every $2^n/K$ clock periods, so the average frequency of overflows will be $f_0 = F f_c$, where $F = K/2^n$. The frequency output of this synthesizer is merely the carry output of the accumulator for a pulse output or the most significant bit (MSB) of the accumulator for an approximate square wave output. The basic problem with this architecture is that it has very high levels of spurs and phase jitter.

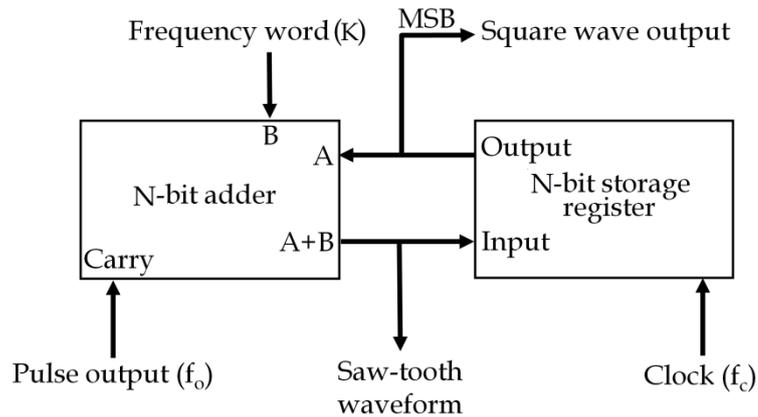


Figure 38 Block diagram of a typical pulse output DDS.

Fractional divider DDS: The fractional divider works similar to pulse output DDS (Figure 39). A programmable divide-by $n/n+1$ counter normally divides clock cycles by n . At each divide output, an N -bit accumulator is clocked to add the frequency word K to itself. Whenever there is a carry the next divide cycle is then set for divide by $n+1$. This produces an output whose average frequency is given by:

$$f_0 = \frac{f_c}{n + F}$$

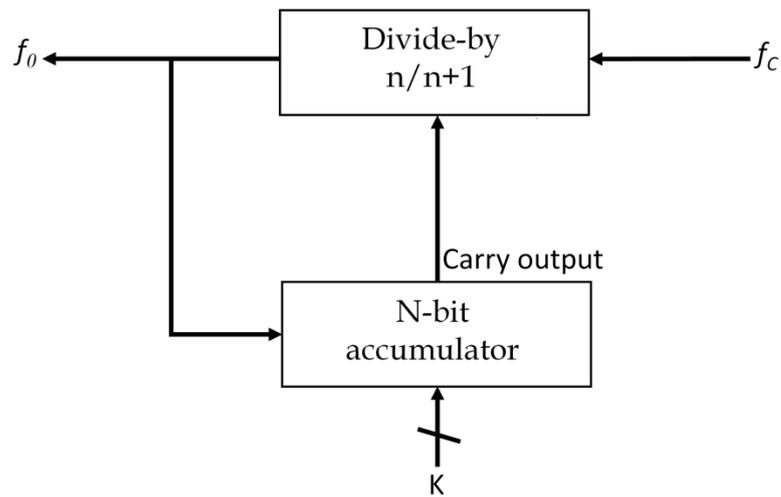


Figure 39 Block diagram of a typical fractional divider DDS.

The fractional divider DDS also produces a high degree of jitter like the pulse output DDS (Reinhardt *et al.*, 1989).

Sine output DDS: The sine output DDS produces a smoother, more sine-like signal by adding a sine look-up table and DAC to the pulse type DDS (Essenwanger and Reinhardt, 1998). The block diagram of the sine output DDS is shown in Figure 40. The sine look-up table computes to the resolution of the sine table. The output of the sine table is then sent to a DAC which outputs a voltage proportional to the sine. Result of this process is to produce a stepped sine wave output which has very low levels of spurs and phase jitter when low pass filtered.

A typical stepped output of a sine output DDS is shown in Figure 41. The levels of spurs and phase noise in this DDS are directly related to the accuracy and resolution of the sine table and DAC. Generating a high resolution sine value directly from a single table usually requires a prohibitively large ROM, so techniques have been developed to reduce the ROM requirements by computing the sine value from several lower resolution tables.

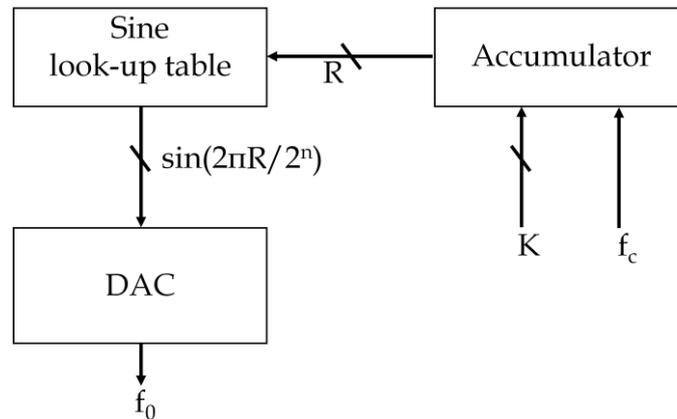


Figure 40 Block diagram of a typical sine output DDS.

Generally, DAC quantization is the limiting factor in sine output DDS. A pure sine wave at f_0 will be recovered when the output is lowpass filtered with a cutoff frequency below $f_c/2$ (Reinhardt, 1993).

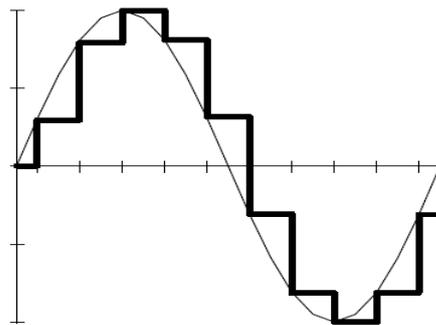


Figure 41 Output form from a typical sine output DDS.

Triangular output DDS: A triangle output DDS is another variation of a sine output DDS without a sine table (Reinhardt, 1985). The block diagram of atypical triangular output DDS is shown in Figure 42. The accumulator register value R of a pulse output DDS is used to drive a DAC directly after passing through a bit complement logic circuit. This triangle wave output has lower spurs than the outputs of a fractional divider or pulse output DDS.

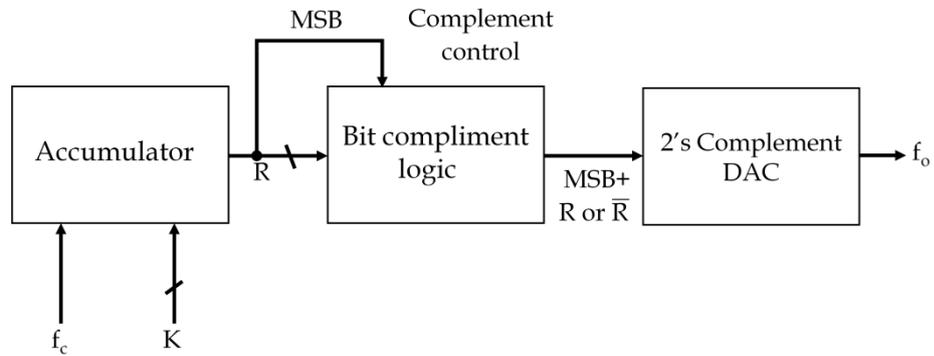


Figure 42 Block diagram of a typical triangle output DDS.

Phase interpolation DDS: The phase interpolation DDS utilizes the fact that, whenever an output transition occurs in a fractional divider, the accumulator register value R is proportional to the phase difference between the output transitions of the DDS and that of an ideal frequency generator. Thus, if R is used to phase shift or delay the output of a fractional divider DDS, lower phase jitter and spurs result (Reinhardt, 1993). As shown in Figure 43, the fractional divider is used in a phase-locked loop. This inverts the fractional division process, so the output is given by:

$$f_o = (n + F_o)f_c$$

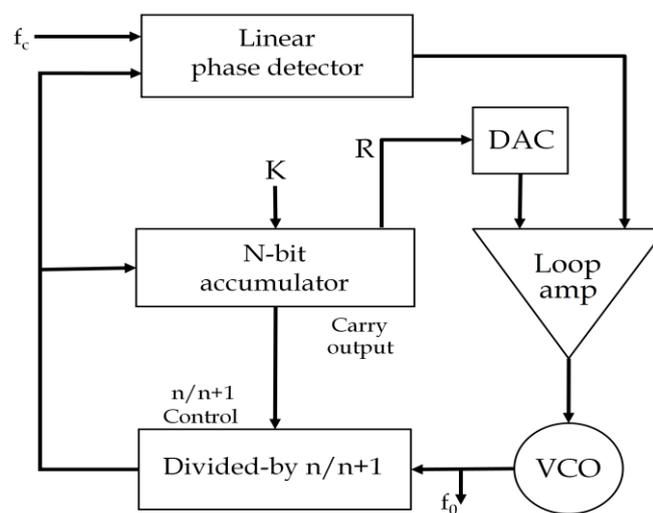


Figure 43 Block diagram of a typical phase interpolation DDS.

The phase jitters and spurs level reductions that are achievable with phase interpolation DDS's are limited by the linearity, accuracy, and resolution of the digital-to-phase conversion process.

Jitter injection DDS: Wheatley (1983) has patented a random jitter injection technique for use on a pulse output DDS which reduces the size of the spurs in the output (Figure 44). This technique reduces the spurs by destroying the periodicity of the phase deviation patterns of the output transitions. A random word, which can vary randomly from 0 to $K-1$, is generated each clock cycle and added to the accumulator register value. The sequence of carriers from this addition then becomes the pulse output. In the original Wheatley circuit, this output is also divided by two to produce a square wave. It can be shown that the spur power in the Wheatley jitter injection DDS is smaller than that in a pulse output DDS.

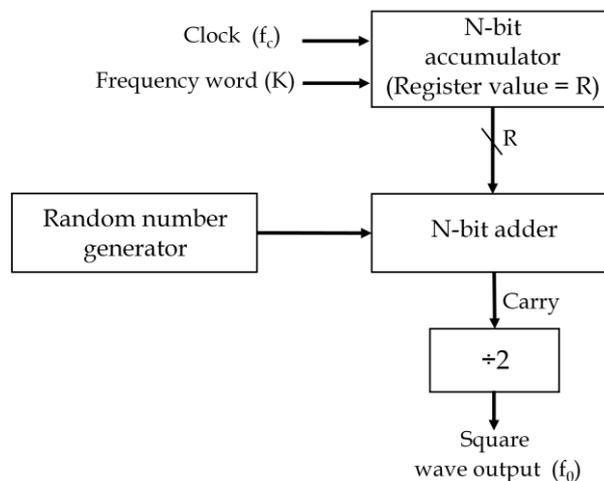


Figure 44 Block diagram of the jitter injection DDS.

Generation of signals for sodar transmitter: The DDS techniques, which are discussed in the previous section, can only be implemented using specially

designed DSP hardware. It is also possible to implement the same techniques through a general-purpose multimedia PC with the help of a few software routines (Petzold, 2013). The low-level waveform audio Application Programming Interfaces (APIs) of Microsoft Visual C++ (or VC++) are used to generate an audio sine wave, within the range of 20 Hz to 20 KHz.

One of the standard modulating techniques, namely, Pulse Code Modulation (PCM), is used to generate audio samples which convert the amplitude and frequency information into digital data. The standard C run-time library includes a function called 'sin' that gives the sine of an angle in radians, generally these values range from -1 to +1. For sine pulse generation, the number of memory locations of RAM is reserved and these calculated sine values are stored in those locations. A double buffer method is employed at this stage, if there is need to generate a continuous wave; otherwise it is an optional parameter.

The important parameter of the signal generation is the sampling rate and using this parameter a signal can be generated by the following formula:

$$s(n) = \sum_{n=0}^{N-1} A \sin(2\pi f n T_s)$$

where $T_s = \frac{1}{f_s}$ = sampling time, f_s - sampling frequency, n -sample number, A - amplitude and f - frequency to be generated. This equation generates a one frequency component. For all the three other frequency components the above equation can be written as follows,

$$s(n) = \sum_{n=0}^{N-1} A \sin(2\pi f_1 n T_s) + \sum_{n=0}^{N-1} A \sin(2\pi f_2 n T_s) + \sum_{n=0}^{N-1} A \sin(2\pi f_3 n T_s)$$

Further above equation can be written as

$$s(n) = A \sum_{n=0}^{N-1} [\sin(2\pi f_1 n T_s) + \sin(2\pi f_2 n T_s) + \sin(2\pi f_3 n T_s)]$$

The ω is calculated in angle radians. i.e.,

$$f_{\text{angle}} = 2\pi f / \text{sampling_rate}$$

This determined f_{angle} value will be multiplied with sample number in order to get the radians per sample. This process repeats to get a continuous wave, but in the case of a Doppler sodar, the system needs a short pulse of 100 ms duration. If the sampling rate is 8000 per second, the required samples to be prepared to generate the pulse of 100 ms are 800. These samples will be prepared just before the sound device is opened and filled in the two buffers and stored in the system primary memory (RAM); this is one time operation in the order of the system execution cycle. The following code snippet explains the implementation of the sine wave generation.

```

/*****
double fAngle1=0;

double fAngle2=0;

double fAngle3=0;

unsigned char *_pbuf;

#define OUT_PUT_BUFFER 800;

#define Freqn1 1750;

#define Freqn2 2000;

#define Freqn3 2250;

```

```

int    GAIN = 12;

for(int i=0; i<OUT_PUT_BUFFER; i++)
    {
        _pbuf[i] =BYTE (127 +GAIN* (sin(fAngle1)+sin(fAngle2)+sin(fAngle3)));

        fAngle1 += 2*PI*Freqn1/SAMPLE_RATE;

        if (fAngle1 >2*PI)

            fAngle1 -= 2*PI;

        fAngle2 += 2*PI*Freqn2/SAMPLE_RATE;

        if (fAngle2 >2*PI)

            fAngle2 -= 2*PI;

        fAngle3 += 2*PI*Freqn3/SAMPLE_RATE;

        if (fAngle3 >2*PI)

            fAngle3 -= 2*PI;

    }

/*****/

```

In the above code, all the three frequency variables are defined as static and phase angles of each frequency component are determined by the given parameters such as sample number, sampling rate, transmit frequency. The flow chart of the cycle is shown in Figure 45. The parameter fAngle calculates the phase angle of the frequency component and that determined phase angle will not be always ends at '0' for cycle for all frequency values. Hence the phase angle is initiated to '0' at the beginning and never be reinitialized to '0' in order to get the continuity of the signal for the entire sodar pulse generation.

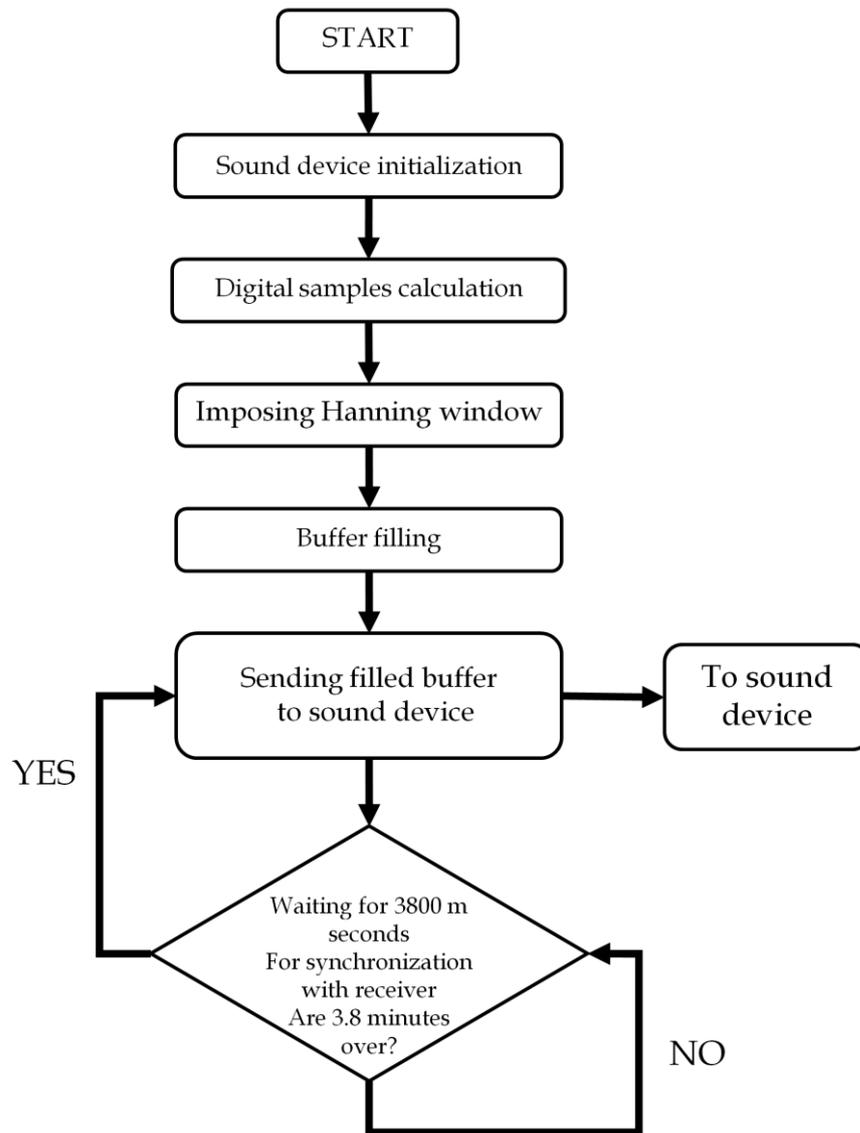


Figure 45 Signal flow chart shows the generation of transmit burst tone.

The system needs to shape the acoustic pulse as shown in Figure 46; the advantages of shaping the signal are two-fold: it protects the acoustic driver unit, and it minimizes harmonic distortion that may take place in the power amplifier section. To get this shape, the samples need to be passed through a Hanning window. Hanning window is somewhat difficult to implement in analog version, but it is easy to implement in software. The process involves multiplication of the predetermined coefficients of Hanning window with a bin of transmitting samples.

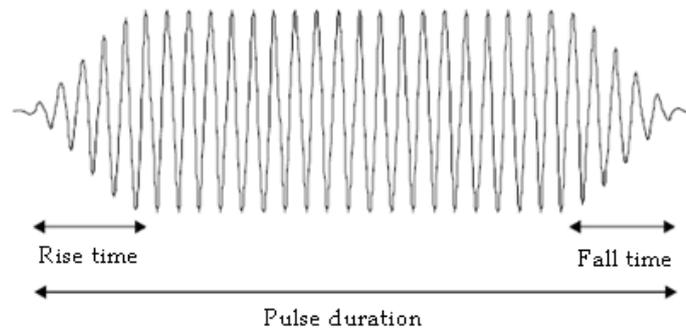


Figure 46 Shape of the transmit burst tone showing tapering on the rise time and fall time of the pulse.

A few on-board sound devices (RealTeck-ALC888) of a PC can be configured for more than two channels; mono, stereo and 2.1, 5.1 etc. The VC++ can call a low-level API `WAVEFROMTEX` to configure mono or stereo channel configuration and to configure more than two channels the API is `WAVEFORMATEXTENSIBLE`. The other API call to configure the kernel mixer is `waveOutOpen()`. This method or function is defined as follows:

```
MMRESULT waveOutOpen(
    LPHWAVEOUT phwo,
    UINT_PTR uDeviceID,
    LPWAVEFORMATEX pwx,
    DWORD_PTR dwCallback,
    DWORD_PTR dwCallbackInstance,
    DWORD fdwOpen );
```

The method takes six parameters and sets the soundcard DAC parameters, when the user clicks the start button through GUI. In the subsequent steps, one of the worker threads of the program prepares the sound samples according to

the given formula and fills up in the allotted buffers. Then, in the preceding step, the program calls another API called `waveOutWrite()`. This method is defined as follows:

```
MMRESULT waveOutWrite(
    HWAVEOUT hwo,
    LPWAVEHDR pwh,
    UINT cbwh );
```

The `waveOutWrite()` method carries the given digital sound sample to the sound device, which converts these digital sample into its equivalent analog signal. Here digital samples are filled in the buffer in the following format as shown in the Tables 2 and 3.

Table 2 Order of the sound samples in the buffer for 16-bit depth

Sample #1	Sample #2	Sample #3	Sample #1	... so on
Channel #1	Channel #2	Channel #3	Channel #1	... so on
8 bits	8 bits	8 bits	8 bits	... so on

Table 3 Order of the sound samples in the channel buffer for 16-bit length

Sample #1		Sample #2		Sample #3	
Channel #1	Channel #1	Channel #2	Channel #2	Channel #3	Channel #3
Low-order byte	High-order byte	Low-order byte	High-order byte	Low-order byte	High-order byte

After the receipt of the API call `waveOutOpen()`, the sound device gets initialized to the given parameters such as the number of channels and the sampling rate. The DAC converts the digital data into analog form when prepared buffer samples are received.

The resultant waveform usually has sharp edges that contain high frequency components. For this reason, play back hardware generally includes a lowpass filter following the DAC. The characteristics of the filter are given in Table 4.

Table 4 Filter characteristics of the RealTek ALC888

Filter	Description	Minimum	Typical	Maximum	Units
Filter	Pass-band	0	---	$0.45 \cdot F_s$	kHz
	Stop-band	$0.60 \cdot F_s$	---	---	kHz
	Stop-band rejection	---	-76	---	dB
	Pass-band frequency response	---	± 0.02	---	dB

F_s = sample rate

The filter removes the high frequency components and produces the final waveform. On the input side, a lowpass filter comes first. The output waveform of the system under test is recorded using a digital storage oscilloscope and a screenshot is presented in Figure 47.

The sampling rate significantly influences, as mentioned earlier, the signal quality and determines the maximum frequency of the sound that can be produced by the sound device.

Bits-per-sample is another parameter, which influences the signal quality at the output. Timing and accuracy of the signal generation is achieved by the chosen sound device. A photo shot of the 100 ms pulse taken with a digital storage oscilloscope is shown in Figure 48.

Power amplifier: A commercially available (Ahuja make) power amplifier is initially used for the power requirements in the present experiment (Figure 49). It is more typical to give above 280 W (electrical) to the acoustic array antenna. A few of the important features of the power amplifier are listed in Table 5. An

in-house power amplifier is designed and built. However, when it is integrated with the array antenna, a few problems related to the heating of power transistors took a lot of time.

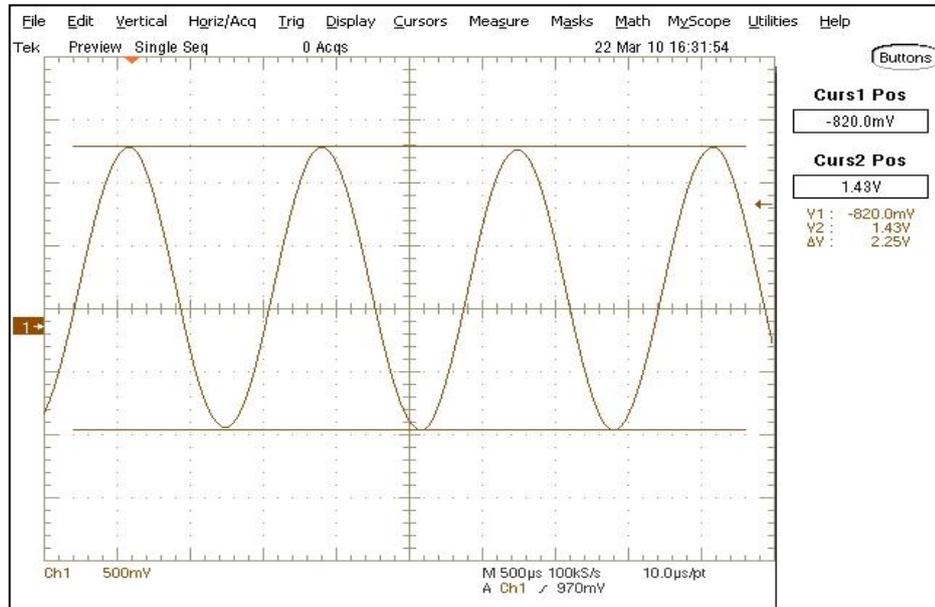


Figure 47 Screenshot of the digital storage oscilloscope showing the sine wave output.

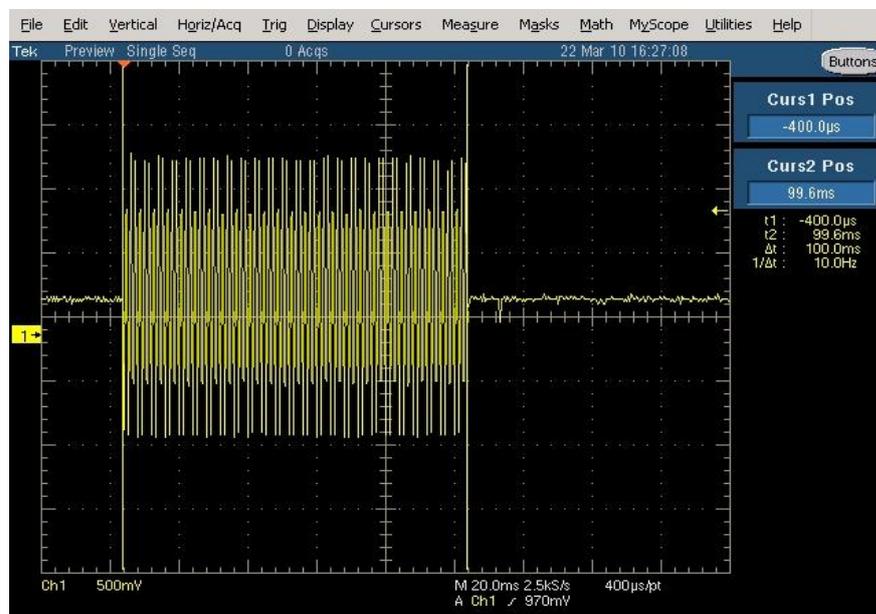


Figure 48 Screenshot of the digital storage oscilloscope displaying the 100 ms duration transmit tone burst.



Figure 49 Photograph of the sodar power amplifier of Ahuja make.

Table 5 A few important specifications of the power amplifier

Power output	500 W + 500 W max, 350 W + 350 W RMS at 10% THD 300 W + 300 W RMS at 5% THD, 280 W + 280 W at 2% THD
Input channels	Mono 2 × 200 mV/100 K Ω ∷ Stereo L/R 200 mV/100 K Ω
Signal to noise ratio	60 dB
Frequency response	50 – 15,000 Hz \pm 3 dB
Speaker outputs	4 Ω , 8 Ω , 70 V and 100 V for each zone

Receiver

The front-end electronics part of the Doppler sodar system includes a preamplifier unit. This protects the sensitive receiving section of the Doppler sodar from high-voltage signals during the transmission, and permit the extremely low-voltage (of the order of a few hundred microvolts) echo-signal to pass through for further amplification.

The circuit diagram of the preamplifier used in the present experiment is shown in Figure 50. The power amplifier output is connected to the preamplifier through a pair of back-to-back (D1) diodes and the interconnecting cable to a diode bridge in the preamplifier. The IN4004 diodes serve to reduce the hum and noise generated by the power amplifier to less than 0.6 V, while the diode B-1 bridge is biased by the 2 K resistors to pass the resulting low-noise voltage

to chassis ground. When a tone burst (transmit signal) is generated, this B-1 bridge is biased off and D1 pair of back-to-back diodes passes the signal to the input terminals of the antenna array. The second B-2 diode bridge, which is in the array leg of the preamplifier, is also biased off from this high voltage signal. In the absence of a tone burst, the back-to-back D2 diodes that were previously short-circuited are now open circuited, and the voltage present at the array antenna terminals is passed to the ferrite pot core transformer.

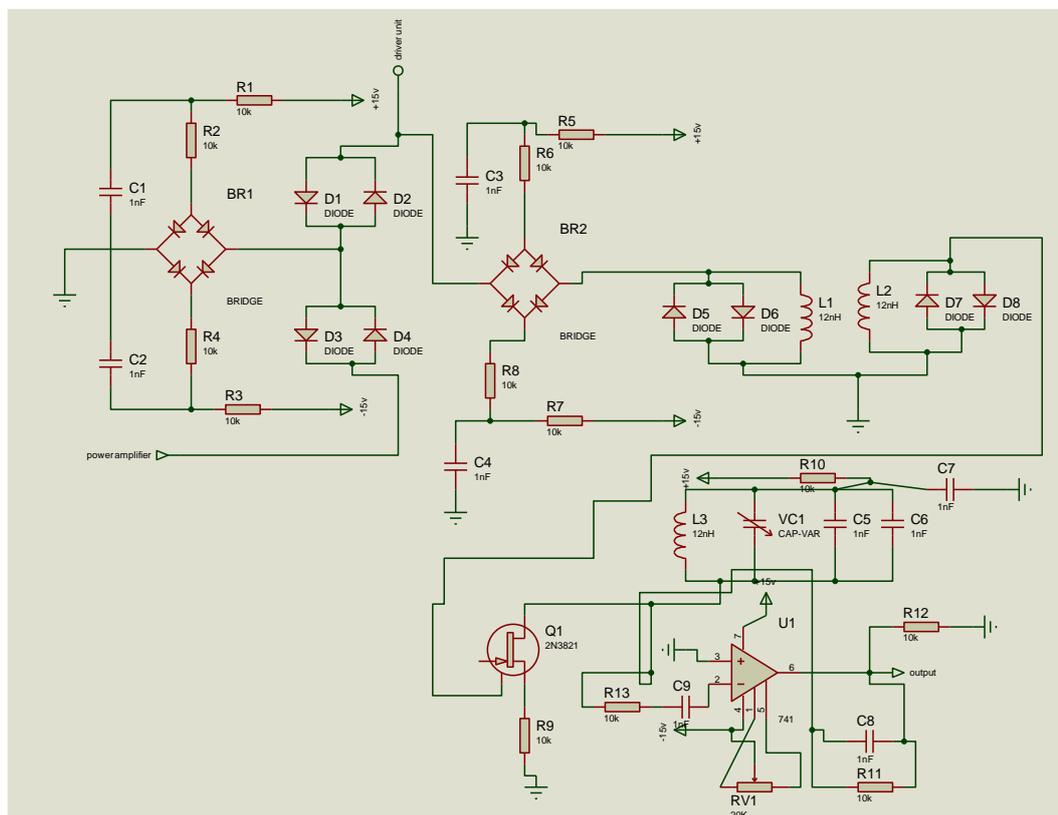


Figure 50 Circuit diagram of the preamplifier unit used in the present Doppler sodar system.

The step-up transformer has 1:60 ratio of winding over a ferrite pot core. It is connected in the reception path through the B-2 diodes bridge. A third set of back-to-back D3 diodes are used to limit the leak-through signal form the B-2 diode bridge to the step-up transformer during the tone burst. The step-up

transformer is used to obtain 40 dB of noiseless amplification of echo-signals varying from 0.1 to 400 μV . The BFW10 is a low-noise FET operational amplifier. It provides another 40 dB gain, raising the signal level into mV range.

A band-pass filter is used to filter the other noise components in the echo-signal and amplify up to a gain of 60 dB. The effects of ground loop currents are avoided by feeding a sample of control unit ground current back to the preamplifier through the non-inverting input of the last stage of amplification. This filter is mixed-up with the low noise amplifier and placed in the feedback of the operational amplifier.

'Proteus Professional' design combines the ISIS schematic capture and ARES PCB layout programmes to provide a powerful, integrated and easy to use tools suite for education and professional PCB design. As a professional PCB design software with integrated shape based auto router, it provides features such as fully featured schematic capture, highly configurable design rules, interactive SPICE circuit simulator, extensive support for power planes, industry standard CAD, CAM & ODB++ output and integrated 3D viewer. The PCB layout for the preamplifier unit is shown in Figure 51. The copper line PCB is shown in Figure 52. The PCB with all the components soldered to the board is shown in Figure 53. Three such preamplifiers are fabricated and placed in an aluminium box with rain protection (Figure 54). The box is housed in a wooden shelter and placed close to the antennas. The output of the preamplifier is fed to an adder circuit shown in Figure 55.

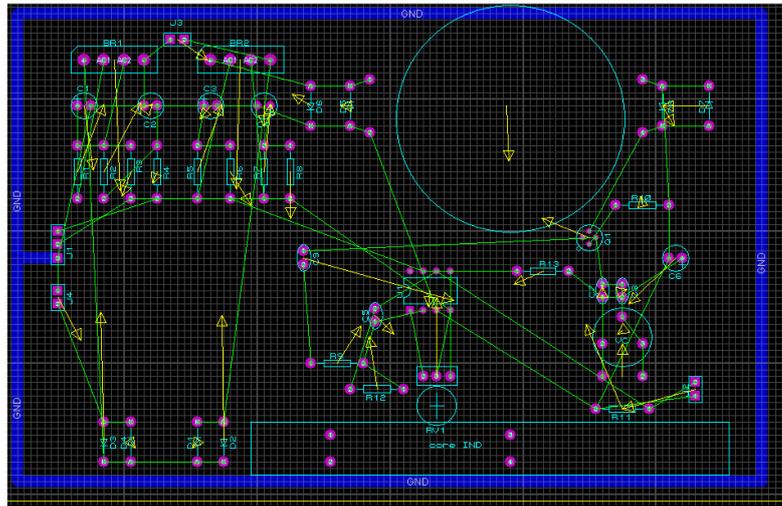


Figure 51 Computer-designed PCB layout of the preamplifier unit.

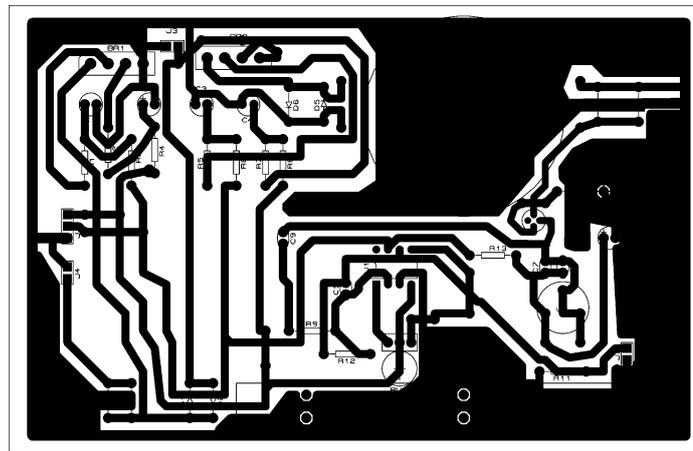


Figure 52 Copper line layout of the preamplifier unit.



Figure 53 Preamplifier PCB with all the components on the board.



Figure 54 Picture showing the preamplifier units for the three antennae.

The role of the adder circuit is to combine all the three echo-signals received on the three antennas and send the composite signals to the line-in port of the sound card. Before fed to the sound card, the echo-signal is further amplified up to 6 dB, as the antenna site and the processing room are 36 m away.

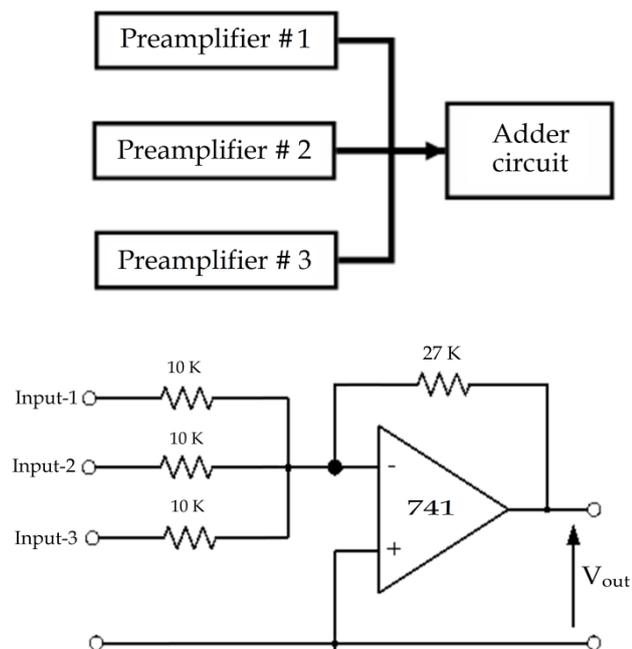


Figure 55 The circuit diagram of the adder used for combining the three echo-signals received from the three antennae operated at three different frequencies.

On the reception side, time duration for reception far exceeds the hardware buffer capacity on the PC soundcard, hence the buffer management routines of the DirectX sound API buffer functionality is used by the application layer.

DirectX sound API internally manages all low-level buffers on their respective interfaces to soundcard hardware (Youngblood, 2002). The application layer will have to manage the high-level DirectX buffers, namely, DirectSoundBuffer and DirectSoundCaptureBuffer to provide uninterrupted operation in multitasking operating system of the Windows environment. The DirectSoundCaptureBuffer stores the digitized echo-signals from the stereo ADC in a circular buffer and notifies the application upon the occurrence of predefined events. Once captured in buffer, one can read the data, perform the necessary modulation and demodulation functions using DSP routines at the application level, and then send the processed data to application layers GUI interface for visualization purpose and storing if necessary. The operation can be represented as a finite state machine as shown in Figure 56.

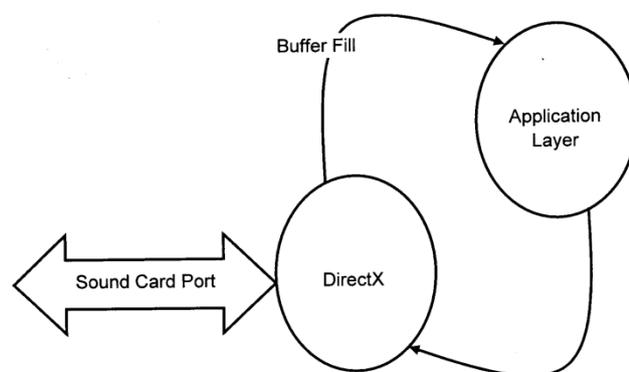


Figure 56 Block diagram showing the circular buffer notification.

Figures 57 and 58 illustrate the concept of circular buffer, which is used for DirectSoundCaptureBuffer.

The hardware buffer is really a linear array in memory as shown in Figure 58, but one can visualize it as circular as shown in Figure 57. This is so because DirectX manages the buffer so that as soon as each cursor reaches the end of the array, the device driver resets the cursor to the beginning of the buffer.

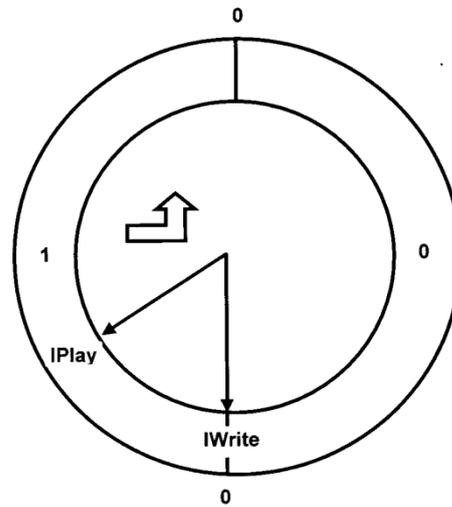


Figure 57 Explanation of the concept of circular buffer.

The DirectSoundCaptureBuffer is split into two blocks, each equal in size to the amount of data to be captured and processed between each event. The event is much like an interrupt. This buffer length is 1024 bytes. In the present experiment, mono-channel with 8 bits per byte is used for capturing echo-signal data, which is equivalent to 1024 bytes per block [1024*1 (Channel)*1 (byte)]. Therefore, the capture buffer is double this size and 2048 bytes.

Since the DirectSoundCaptureBuffer is divided into two blocks, one needs setup notification event after each block gets captured. The DirectX driver maintains the cursors that track the capture operation at all times. The driver provides a means of setting specific locations within the buffer that cause an event to trigger, thereby telling the application to retrieve the data (Kumar *et al.*,

2005). The two cursors resemble the hands on a clock face rotating in clockwise direction, as shown in Figure 57. The capture cursor, *IPlay*, represents the point at which data is currently being captured. The read cursor, *IWrite*, trails the capture cursor and indicates the point up to which data can be safely read. The data after *IWrite* and up to and including *IPlay* are not necessarily good data because of hardware buffering. It there on receives two events per each revolution telling the application layer to read respective block of data.

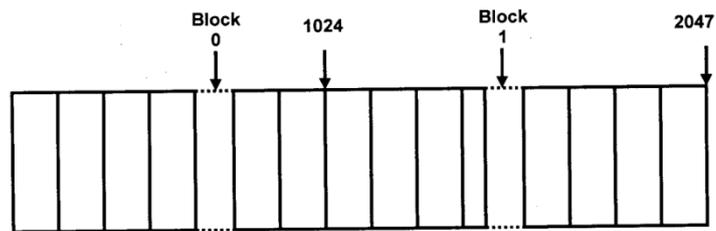


Figure 58 Explanation of the concept of linear array in memory.

The DirectSoundCapture object *IDirectSoundCaptureBuffer* is used to query the capabilities of sound capture devices and to create buffers for capturing audio from an input source. DirectSoundCapture allows capturing of data in PCM or compressed formats.

The DirectSound capture buffer object *IDirectSoundCaptureBuffer* represents a buffer used for receiving data from the input device. Like playback buffers, this buffer is conceptually circular: when input reaches the end of the buffer, it automatically starts again at the beginning.

The methods of *IDirectSoundCaptureBuffer* enable to retrieve the properties of the buffer, start and stop audio capture, and lock portions of the memory, so that one can safely retrieve data for saving to a file or for some other purpose.

For an application that is simply going to capture sounds through the user's preferred capture device, there is no need to enumerate the available devices. By passing NULL or a pre-defined GUID to `DirectSoundCaptureCreate` or `DirectSoundFullDuplexCreate`, it is possible to create a default device. If for a particular kind of device or wish to offer the user a choice of devices, program must enumerate the devices available on the system.

- Reports what hardware is available.
- Supplies a GUID for each device.
- Enables the user to initialize each device as it is enumerated, so that the user can check the capabilities of the device.

Doppler sodar software tools

The first-generation sodar systems used the standard horizontal sweep facsimile recorders for recording the amplitude of the echo-signals in the form of a time-height plot. These recorders used chemically-coated electro-sensitive thermal paper as the medium of writing. In the facsimile recorder, each pulse produces a vertical trace whose darkness varies with the echo-signal intensity from a particular height in the atmosphere, a height proportional to the ordinate of the chart. Since the traces from successive pulses are closely spaced along the abscissa, the eye integrates corresponding features and detects the evolution of atmospheric structure.

One of the major problems in recording the data on those special papers is related to their relatively shorter shelf-life. The contents of the record get faded away within a relatively short-time. They must also be handled carefully because of the wear and tear with the paper medium. The advent of

microcomputers, modern digital circuits and new digital signal processing techniques made the recording/display of the echo-signal more user-friendly and ease in the retrieval. An attempt is made here to develop software tools to produce the echo-signal intensity time-height plots on a real-time basis. The tool provides the user to display the plot for a pre-selected time and height off-line and print using the present-day dot-matrix, inkjet and laser printers. A provision is also made to remove any unwanted noise patches in the plot caused by either man-made or instrumental.

Windows programming requires extensive calls to resources provided in the hardware. MFC library makes full use of GDI, which is a layer in the Windows architecture that insulates the application from direct interaction with hardware. This interface has an extensive set of high level functions that can be linked from its objects for drawing and managing graphics (Swanke, 1999). A class that is commonly derived from CObject is CDC. In MFC, CDC is the base class for providing an interface with other class, including CPaintDC, CClientDC, CWindowDC, and CMetaFileDC. An object abstraction called device-context has been provided through these four classes, which links to the functions in CDC for supporting all the basic graphical and drawing functions. Device-context is an object that is responsible for displaying text and graphics as output. In reality, a device-context is a logical device that acts as an interface between a physical device (such as monitor and printer) and the application. For achieving this task, a device-context has a set of tools or attributes for putting text and drawing graphics on the screen using GDI functions.

Device-context is created from class CClientDC. The tools in device-context are represented as graphic objects such as pens, brushes, fonts, and bitmaps. There are four types of device contexts in GDI: display-context, memory-context, information-context, and printer-context. A display-context supports operations for displaying text and graphics on a video display. Before displaying text and graphics, a display-context links with MFC functions for creating a pen, brush, font, color, palette, and other device. Members of the device-context class MoveTo(x, y) and LineTo(x, y) functions are used to draw coordinate axes.

In MFC, the color of each pixel is controlled using the function RGB() from the CDC class. RGB() consists of three arguments in the order from left to right as the red, green, and blue components. The function is declared as RGB(r,g,b);. Each component in RGB() is an integer that represents the monotone scale from 0 to 255, with 0 as darkest value and 255 as the lightest. The monotone scale for solid black is RGB(0,0,0) and for white it is RGB(255,255,255). Gray color is obtained by setting r=g=b in RGB(r,g,b). The function SetPixel(x, y, RGB(r, g, b)) is used to illuminate a pixel on video display. VC++ graphics drawing capabilities are used to develop the facsimile record for the Doppler sodar system.

The timing information related to both the transmission and reception is shown in Figure 59. The pulse repetition time of the Doppler sodar is 4 s. The speed of sound in atmosphere is taken as 340 ms^{-1} in the present experiment. The maximum unambiguous range is computed as 680 m ($340 \text{ ms}^{-1} \times 4 \text{ s} / 2 = 680 \text{ m}$). The minimum range depends on the transmitted pulse width (100 ms in

the present experiment) and the subsequent blanking time (100 ms in the present experiment). The total amount of time is 200 ms and the minimum detectable range of the system is computed as 34 m ($340 \text{ ms}^{-1} \times 0.2/2 = 34 \text{ m}$). The range resolution of the system is 17 m for the given pulse width of 100 ms in the present experiment ($340 \text{ ms}^{-1} \times 0.1/2=17 \text{ m}$).

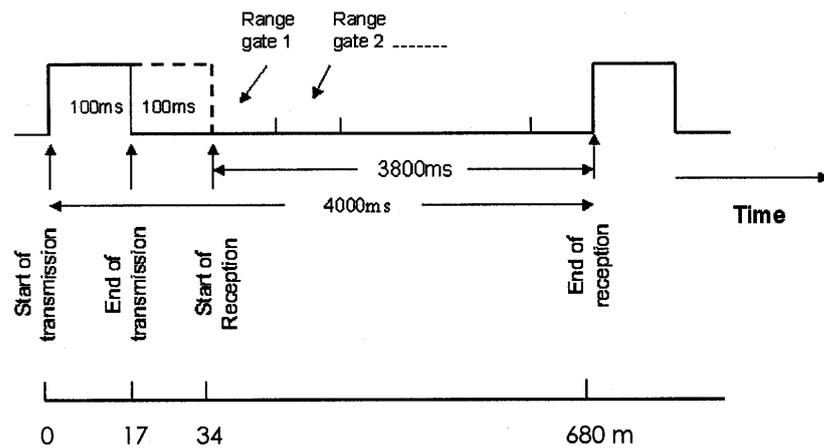


Figure 59 Time-range diagram of the present Doppler sodar system.

The echo-signal is sampled at 8 KHz. The number of digital samples available during the receiving time of 3800 ms is $8000 \times 3.8 = 30,400$. The vertical resolution for the measurement of wind vector profiles is dependent on the number of samples selected for the computation of FFT and the sampling frequency. The vertical resolution, Δh in the present experiment is computed as follows:

$$\Delta h = \frac{c}{2} \times \frac{\Delta n}{n} = \frac{340}{2} \times \frac{1024}{8000} = 21.76 \text{ m}$$

where Δn is the number of samples and n is the sampling frequency. In general, Doppler sodar systems using frequencies around 2000 Hz, height intervals in the range 20-30 m are most common. These intervals are acceptable for wind

profile measurements. However, the associated echo-signal intensity cannot be used to produce time-height plots that are comparable, in terms of resolution, with the analog facsimile recorders (Mastrantonio and Argentini, 1997). For this purpose, a second harmonic analysis is performed over a shorter signal length. The previously considered 1024 samples, which are used to retrieve radial wind, is divided by 8 and a harmonic analysis is again performed on the 128 samples to get the time-height intensity plot. This step corresponds to a height resolution of 2.72 m as shown below:

$$\Delta h = \frac{c}{2} \times \frac{\Delta n}{n} = \frac{340}{2} \times \frac{128}{8000} = 2.72m$$

Therefore range information corresponding to the first range gate is $34+2.72=36.72$ m. For second range gate, it is $36.72+2.72 = 39.44$ m. The digitized data are gated sequentially and processed in real time using FFT to obtain discrete frequency spectra. FFT bin Peak amplitude echo-signal is measured and gray color corresponding to that value illuminates the pixel at that height on the graph at that instance of time as shown in the Figure 60. This process repeats for every 4s for all the range gates, thus, forming a traditional-type facsimile record (Kumar *et al.*, 2008).

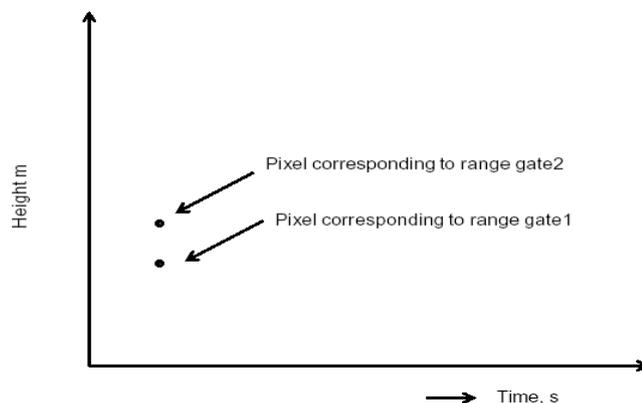


Figure 60 Description of the development of the digital facsimile record.

The x- and y-axis are set to indicate the local time and height, respectively. The gray color assigned to each pixel indicates the amplitude of the echo-signal at the corresponding range-height. Black color corresponds to the strongest echo-signal and white for weakest. Echo-signals that have the intermediate amplitudes are represented by the varying shades of gray color.

Doppler shift calculation

There are two basic ways to acquire data from the real world signal, either one sample at a time (continuous processing), or one frame at a time (batch processing). Here the Doppler sodar is frame-based system, thus, it acquires a frame of samples and processing occurs on the entire data and results in a frame of transformed output data. In order to maintain real time operation, the entire FFT must be calculated during the frame period. This assumes that the sound device which is used to collect the data for the next frame, while the computer is calculating the FFT for the current frame of the data.

Another important step is to determine the number of points in the FFT to achieve the desired frequency resolution. The frequency resolution is obtained by dividing the sampling rate f_s by N , the number of points in FFT. The number of FFT points also dictates noise floor of the FFT with respect to the broadband noise level, and this may also be a consideration. The FFT acts like an analog spectrum analyzer with sweep bandwidth of f_s/N . Increasing the number of points per bin, increases the FFT resolution and narrows its bandwidth, thereby reducing the noise floor.

This analysis neglects the FFT round off error. In practice, the ADC that is used to digitize the echo-signal produces quantization noise, which is a dominant

noise source. One important consideration is the time required for FFT processing to make sure that real-time operation can be achieved. This means that FFTs must be calculated during the acquisition time for one frame of data which is N/f_s .

The power spectrum from the range gate i consists of value R_{ik} at frequencies f_{ik} ($k=1,2,\dots,N_s$) can be calculated using the following formula.

$$f_{ik} = \frac{f_s}{N_s} \left(k - \frac{N_s}{2} \right)$$

The power spectrum of the subsequent range gates for single-axis monostatic system is labeled and illustrated in the Figure 61.

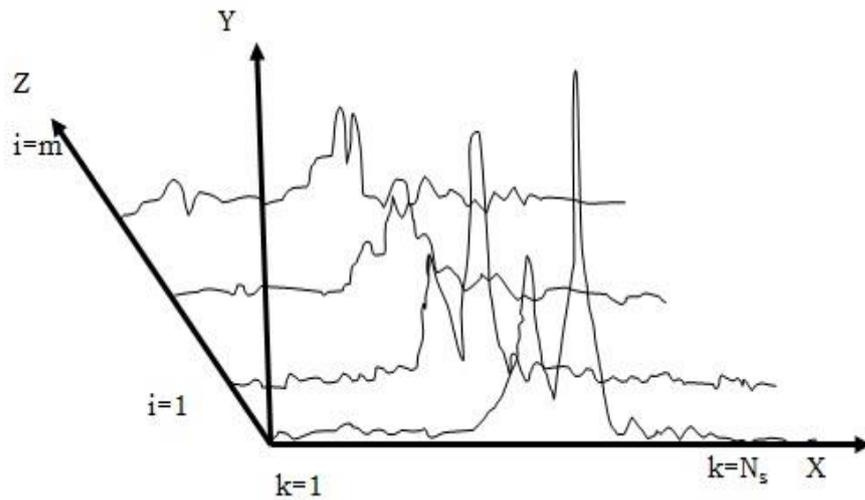


Figure 61 Labeling of power spectrum values in the profile.

A sample of the power spectrum for the first 15 range gates obtained from the present tri-axial monostatic Doppler sodar system is shown Figure 62. The spectrum shows a composite signal of the above system consisting of ± 50 Hz band of signals of the three transmitted frequencies.

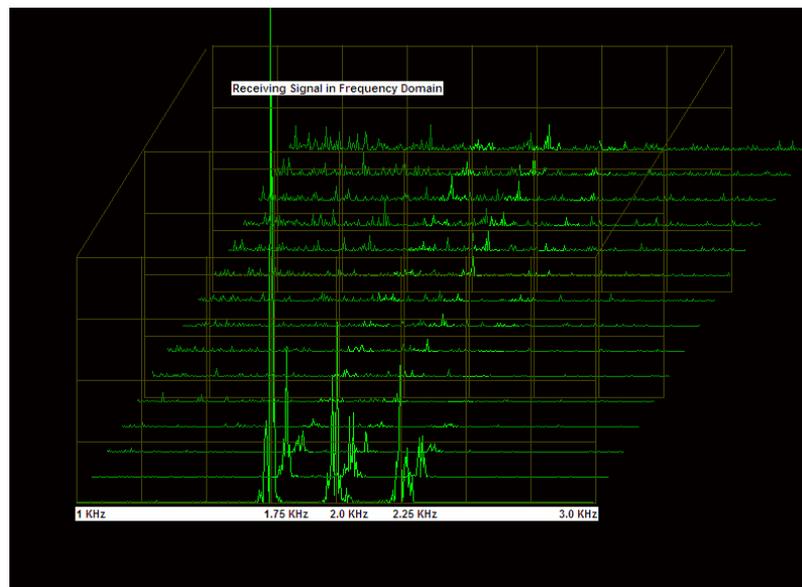


Figure 62 A sample power spectrum of the present Doppler sodar system.

The measurement of the radial wind velocity component in sodar systems is generally carried out by application of the first moment integral to the spectrum of the echo-signals. In unfavorable signal-to-noise conditions, this leads to a large systematic bias and lack of precision. Hence a two-step procedure is adopted, whereby the spectrum of the echo-signal is first localized and then the actual value of its center frequency is determined (Mastrantonio and Fiocco, 1982). Large improvements in accuracy and precision are thus obtained.

The radial wind velocity measurement in any range gate consists of (1) the determination of the center frequency of the echo-signal, and (2) comparing with the center frequency of the transmitted signal. Since the spectral content of the transmitted signal can be assumed to be under control, the uncertainties are confined only to the echo-signals. The echo-signal and the noise are described by their spectral densities.

The procedure is carried out in two stages. The first stage consists of a search over the entire spectrum to determine the approximate location of the echo-

signal. As shown in Figure 63, this is done by a balancing method whereby the entire spectrum is scanned by fixed increments of width δf and index I with two adjacent, moving windows of width $F/2$ (Mastrantonio and Fiocco, 1982).

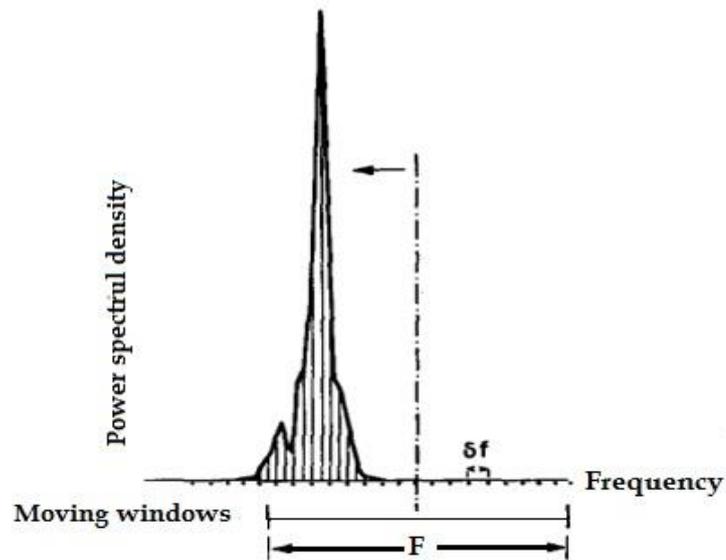


Figure 63 Implementation of the balancing method (Mastrantonio & Fiocco, 1982).

The power contained in each of the two windows is measured. The echo-signal is considered acquired for that location of the windows where the difference in the power levels of the two windows is minimal. Then the ratio between such levels and the average level obtained in any other location is calculated. The measurement is accepted if such a ratio exceeds certain threshold. The measurement is discarded if this condition is not met. A further improvement in the center frequency estimate is obtained by making a parabolic fit for three values around the minimum. A fractional value is thus obtained corresponding to a minimum in the parabola. The second stage in the procedure is the determination of the center frequency by the application of the first moment integral to this narrow region. Thus a best estimate of the Doppler frequency

and, consequently, of the radial wind w is obtained. This region is much narrower and the weight of the noise spectrum is considerably less.

In the earlier Doppler sodar systems, the echo-signal is heterodyned down to 250 Hz and is then sampled at 1 KHz (Owens, 1977). The same technique is used in a tri-axial monostatic Doppler sodar system where the echo-signal is heterodyned down to 750 Hz and is sampled at 1.6 KHz (Elisei *et al.*, 1986). In the present experiment, the ADCs on the sound card support independent 16/20/24 bit and 44.1 KHz/48 KHz/96 KHz/192 KHz sampling frequencies. The ADCs can also be configured through software tools to sampling frequencies down to 8 KHz/11.05 KHz/22.1 KHz. For the present work, both the 8-bit and 16-bit conversions are used. A GUI is designed to select between 8-bit and 16-bit conversion and also for the sampling frequency. All the results presented here are based on the 8 kHz sampling frequency unless otherwise specified.

Aliasing: As an example, assume that the sampling frequency is $f_s = 8000$ Hz. A signal component having frequency $8000 + 8000/3 = 10666.66$ Hz gives the same digitized values as if it had a frequency of $8000/3 = 2666.66$ Hz. This means that higher frequency components can add into the lower frequency spectrum, which is called aliasing. This means that all the frequency components outside of $nf \pm f_s/2$ should be excluded from the signal before digitizing.

Usually this is interpreted as using anti-aliasing low-pass filters to remove all frequency components outside of $\pm f_s/2$, but in fact the criterion is satisfied if band-pass filters remove all components within a $\pm f_s/2$ bandwidth of nf_s . To achieve these conditions a tuned band-pass filter is included in the preamplifier

circuit with bandwidth of 100 Hz. Another low pass filter is also available with the on-board sound card of the PC used in the present experiment.

Storage of raw data and real-time processing: It is a practice in every scientific experiment to store the raw data for off-line analysis. The earlier storage devices include digital tape recorders and the magneto-optical disks. The arrival of large capacity solid state memory devices and external hard disks made the storage of raw data relatively easy.

Nowadays, there are a number of standards being practiced in the industry for storing the audio data such as A.U, VOC, WAVE, AIFF, AIFF-C and IFF/8VX. In the present experiment, the Microsoft .WAV file format is used for the raw echo-signal storage.

Wave file format: A wave file format is a subset of Microsoft RIFF specification for the storage of multimedia files. The Table 6 shows typical RIFF wave file format for Windows operating system for storage and playback.

The format starts out with a file header followed by a sequence of data chunks. A wave file is often just a RIFF file with a single "WAVE" chunk which consists of two sub-chunks; "fmt " chunk specifying the data format and a "data" chunk containing the actual sample data. This wave file format is supported by all available general software for multimedia players, so the data can be visualized with the minimal setting of the player; hence selection of the data for the off-line processing in the present experiment becomes easy.

Ramp implementation: The echo-signal in the receiver chain usually contains a multiplier circuit to which one input is the echo-signal itself and the other is a ramp signal whose repetition time is equal to the pulse repetition time of the

Doppler sodar system. The output of the multiplier circuit is an amplified version of the echo-signal whose magnitude increases linearly with the pulse repetition time. This is necessary due to beam spreading and absorption characteristics of the sound in the atmosphere. This is also known as range compensating circuit. A typical circuit developed by Owens (1977) is reproduced in Figure 64. The ramp signal according to the pulse repetition time of the system is generated by the Burr-Brown 4013/25 module. Wired as an integrated-and-reset generator, the module generates a linear ramp which forms one input to the multiplier AD 532JD. As a result of this process, the echo-signals from lower ranges are amplified less compared to their counterparts from longer ranges.

Table 6 Wave file format

The WAVE file format				
endian	File Offset (bytes)	Field name	Field Size (bytes)	Description
big	0	ChunkID	4	The "RIFF" chunk descriptor The format of concern here is "WAVE", which requires two sub-chunks; "fmt " and "data"
little	4	ChunkSize	4	
big	8	Format	4	
big	12	Subchunk1ID	4	The "fmt " sub-chunk Describes the format of the sound information in the data sub-chunk
little	16	Subchunk1Size	4	
little	20	AudioFormat	2	
little	22	NumChannels	2	
little	24	SampleRate	4	
little	28	ByteRate	4	
little	32	BlockAlign	2	
little	34	BitsPerSample	2	
big	36	SubShunk2ID	4	
little	40	SubChunk2Size	4	The "data" sub-chunk Indicates the size of the sound information and contains the raw sound data
little	44	Data		

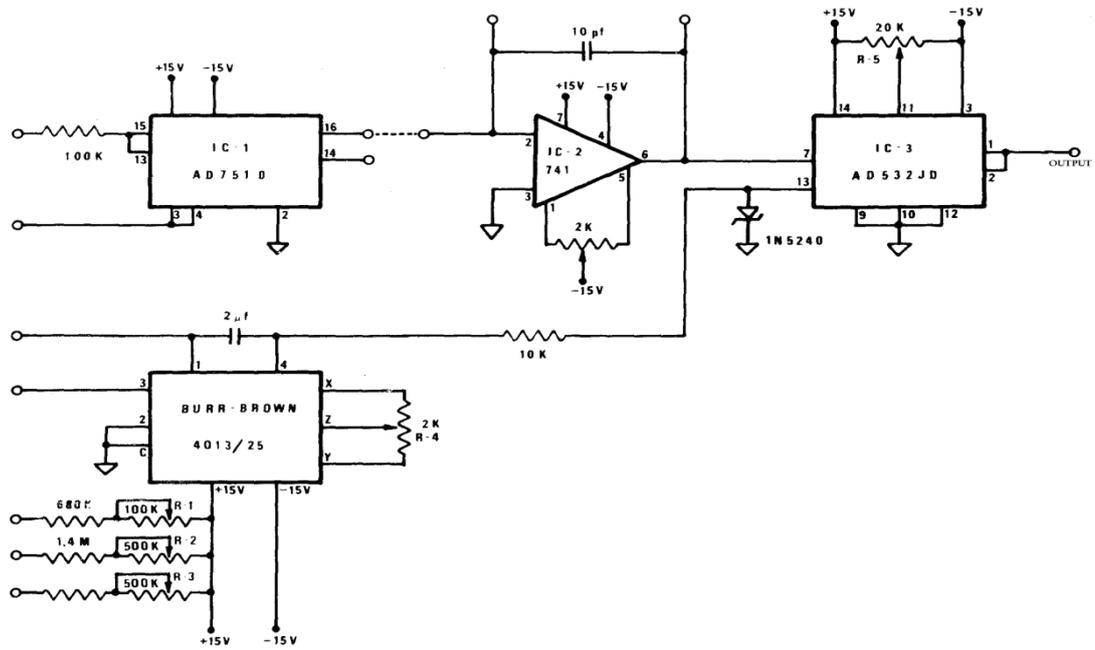


Figure 64 Circuit diagram of the range compensating module.

In the present experiment, the role of the range compensation module is implemented with a newly developed software tool. The digitized echo-signal samples are collected in 15 buffers with each buffer containing 2048 samples. Each sample is added with a ramp_gain factor, which will be incremented linearly with a unit of 0.001. The following VC++ code snippet explains the implementation of the ramp_gain hardware equivalent. The flow chart of the software routine is shown in Figure 65.

```

//*****//
int ramp_gain=0;// defined and initialized parameter outside the for loop.
int NoB = 0; // defined and initialized the NoB for the buffer count
for(int k=0; k<2048; k++) // loop for multiplying with the
    gain_ramp parameter
{
    _pBuf3[NoB+k]=_pBuf2[NoB+k]*ramp_gain;

```

```

// multiplying the raw data in the buffer_2 and copying to the
    new buffer_3.

ramp_gain += 0.001;

    // incrementing the ramp_gain parameter linearly in
    accordance to the increment of bit number in the buffer

    }

NoB ++; // Increment the buffer count

if (NoB == 15) // loop for checking the buffer count

ramp_gain =0; // If buffer count reaches the 15 then the

    ramp_gain will be reset to 0.

//*****//

```

Range gates: The Doppler sodar system measures the wind vector at different height intervals. These height intervals are also known as range gates. The maximum resolution that can be obtained for these range gates is given by:

$$\Delta z = \frac{c\tau}{2}$$

The range resolution for the present experiment can be determined as:

$$\Delta z = \frac{c\tau}{2} = \frac{(340 \times 0.1)}{2} = 17 \text{ m}$$

It can also be defined as:

$$\Delta z = \frac{cN}{2f_s}$$

Here, the number of samples is represented by N. The resolution for the values considered in the present experiment is obtained as:

$$\frac{cN}{2f_s} = \frac{(340 \times 2048)}{2 \times 8000} = \frac{696320}{16000} = 43.52 \text{ m}$$

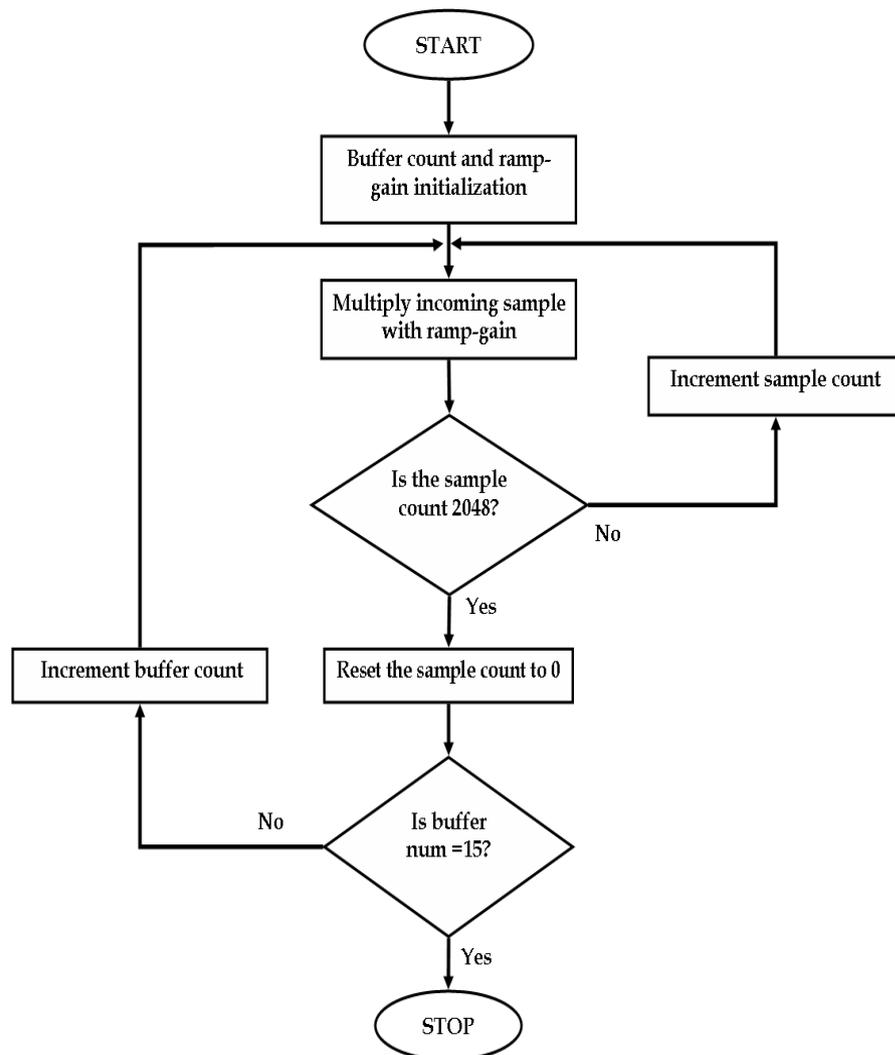


Figure 65 Flow chart of the ramp_gain implementation.

Spectral leakage and windowing: Whenever DFT is to be computed, it is necessary to take only a finite number of data samples covering duration of t seconds. With this finite length of data, discontinuities that occur at the end points of the data window results in spectral leakage in the frequency domain because of the harmonics that are generated. It results in unwanted side lobe levels. In addition to this, the main lobe smears over several frequency bins. This situation would be unsuitable for many applications. This leakage can be

eliminated by multiplying incoming time samples by any suitable windowing function, which brings the signal to zero at the edges of the window. The selection of the window function is a trade-off between main lobe spreading and side lobes roll-off.

The mathematical functions describe four popular window functions. They are Hamming, Blackman, Hanning, and minimum 4-term Blackman Harris. Hanning is well tested for the system, and it achieved the requirements of the system. The computations are straight forward, the window function calculation are pre-calculated and are stored in the memory to minimize their impact on the FFT processing time. The result of the time data samples multiplied by a Hanning window is illustrated in Figure 66.

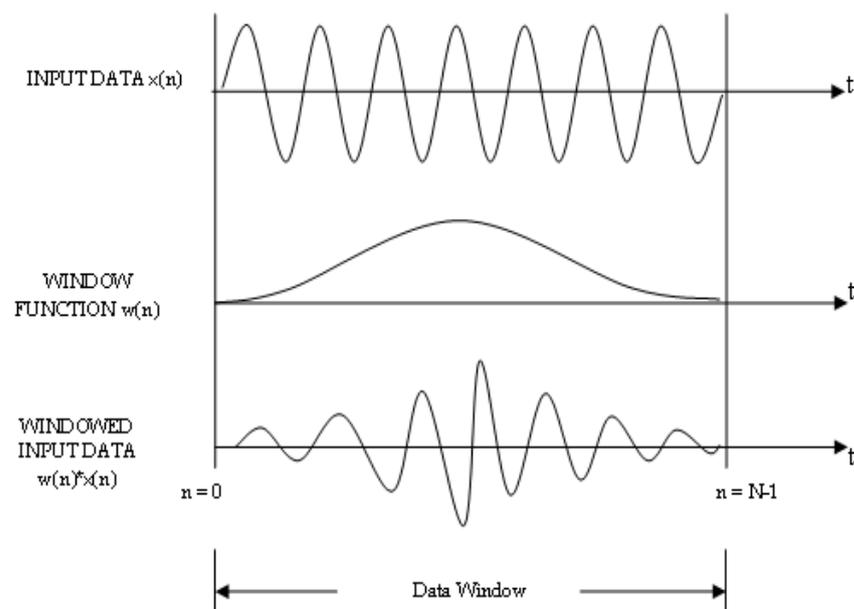


Figure 66 Windowing to reduce spectral leakage.

Vertical wind velocity calculation: The radial component on the antenna pointed to zenith gives the vertical wind speed of the atmosphere. The vertical wind is calculated and sent to the paint thread to draw on the client area of the window.

Figure 67 shows the signal flow graph for the time-height plot of the vertical wind velocity.

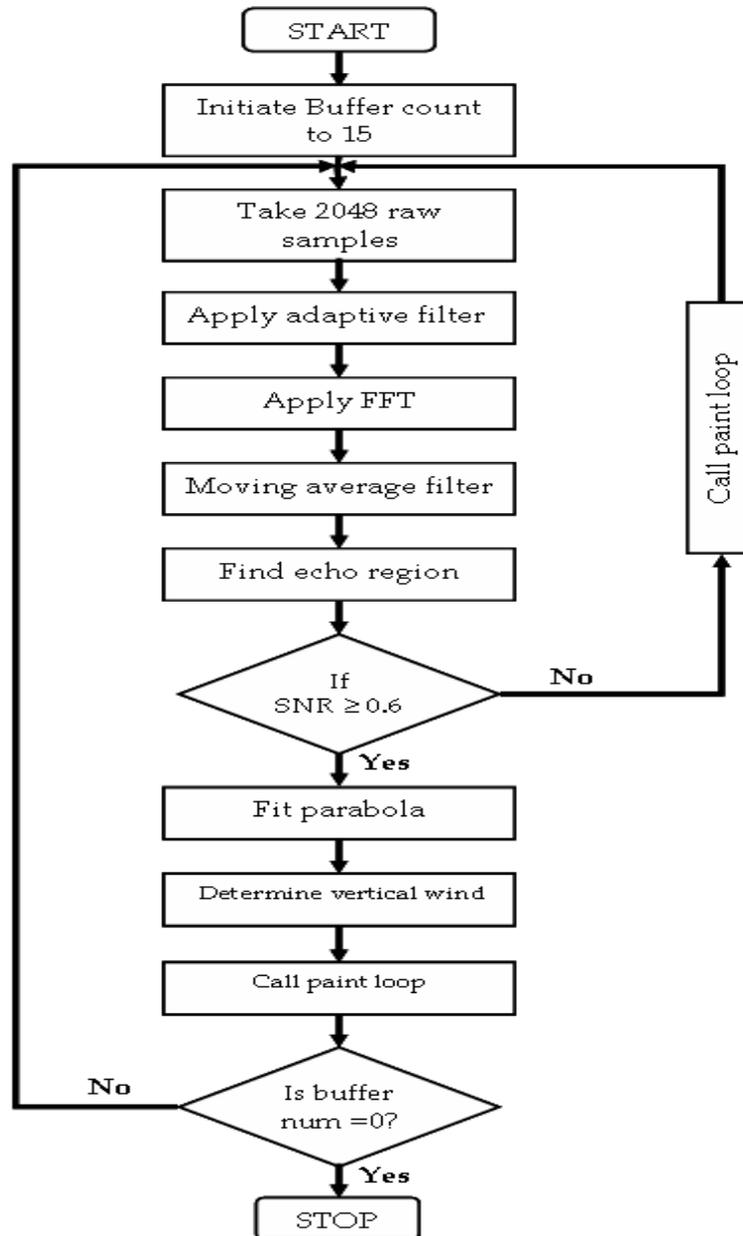


Figure 67 Signal flow chart for the vertical wind velocity.

The maximum and minimum detectable vertical wind velocities, using the present system, are $\pm 5 \text{ ms}^{-1}$. Figure 68 shows a sample of the vertical wind profile obtained on October 5, 2014 between 09:26 and 10:26 hrs. For each cycle,

the echo-signal data is divided into 15 range gate and each range gate corresponds to a height of 34 m covering a total height of 680 m.

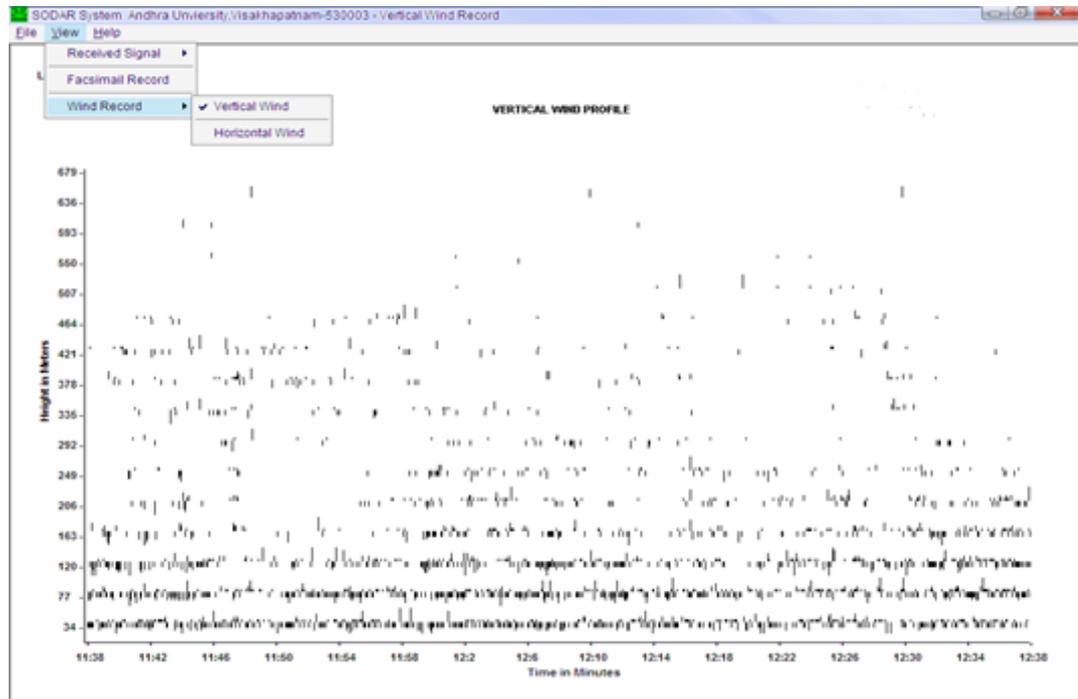


Figure 68 The window showing the time-height plot of vertical wind velocity.

The vertical velocity is visualized using $\text{MoveTo}(x_1, y_1)$ and $\text{LineTo}(x_2, y_2)$ as in the case of facsimile record. At every point of drawing y_{oldpoint} is set to zero point to the measurement level, and the y_{newpoint} takes either negative or positive vertical velocity and performs the addition or subtraction from the y_{oldpoint} . The difference is drawn on the screen either upwards or downwards depends on the \pm sign of the velocity, from mean zero level according to the magnitude passed to the function argument in the form of y_{newpoint} . If the magnitude goes upwards it indicates the wind is in upward and vice versa. The following code snippet illustrates the plotting method. In the code snippet, the method “ $\text{DrawVeticalLine}()$ ” of class ‘Record’ takes the magnitude of the vertical wind velocity in the form X-coordinate and Y-

coordinate of pixel value and presents on the screen with a resolution of 1024 x 768.

```
//*****//
void Recorder::DrawVerticalLine(CDC *pdc,int X_OldPoint,int Y_OldPoint,
                               int X_NewPoint,int Y_NewPoint)
{
    CPoint OldPoint;
    CPoint NewPoint;
    OldPoint.x = X_OldPoint;
    OldPoint.y = Y_OldPoint;
    NewPoint.x = X_NewPoint;
    NewPoint.y = Y_NewPoint;
    CDC *pDCVL = pdc;
    pDCVL->MoveTo(OldPoint);
    m_memdc2.MoveTo(OldPoint);
    pDCVL->LineTo(NewPoint);
    m_memdc2.LineTo(NewPoint);
}
//*****//
```

Horizontal wind velocity: The values of the three Doppler shifts estimated from the three echo-signals received on the three antennae are substituted in the following matrix to obtain the three components of the wind vector for each range gate (Bradley, 2008).

$$\begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} = \begin{bmatrix} -\frac{c \Delta f_1}{2 f_T} \\ -\frac{c \Delta f_2}{2 f_T} \\ -\frac{c \Delta f_3}{2 f_T} \end{bmatrix} = \begin{bmatrix} \sin\theta \cos\phi & \sin\theta \sin\phi & \cos\theta \\ -\sin\theta \sin\phi & \sin\theta \cos\phi & \cos\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

All three spectral peak positions are linear related to the three velocity components. If they are expressed in equation notation, above values can be presented as follows:

$$\eta_1 = -\frac{c}{2} \frac{\Delta f_1}{f_T} = u \sin \theta \cos \phi + v \sin \theta \sin \phi + w \cos \theta$$

$$\eta_2 = -\frac{c}{2} \frac{\Delta f_2}{f_T} = -u \sin \theta \sin \phi + v \sin \theta \cos \phi + w \cos \theta$$

$$\eta_3 = -\frac{c}{2} \frac{\Delta f_3}{f_T} = W$$

After substituting the sine and cosine ($\theta = 20^\circ, \phi = 0^\circ$) values in the above equations, they can be written as follows:

$$\eta_1 = -\frac{c}{2} \frac{\Delta f_1}{f_T} = u \sin(20^\circ) \cos(0^\circ) + v \sin(20^\circ) \sin(0^\circ) + w \cos(20^\circ)$$

$$\eta_1 = -\frac{c}{2} \frac{\Delta f_1}{f_T} = u \sin(20^\circ) + w \cos(20^\circ)$$

$$\eta_1 = -\frac{c}{2} \frac{\Delta f_1}{f_T} = u * (0.342020) + w * (0.939692)$$

$$\eta_2 = -\frac{c}{2} \frac{\Delta f_2}{f_T} = -u \sin(20^\circ) \sin(0^\circ) + v \sin(20^\circ) \cos(0^\circ) + w \cos(20^\circ)$$

$$\eta_2 = -\frac{c}{2} \frac{\Delta f_2}{f_T} = v \sin(20^\circ) + w \cos(20^\circ)$$

$$\eta_2 = -\frac{c}{2} \frac{\Delta f_2}{f_T} = v * (0.342020) + w * (0.939692)$$

$$\eta_3 = -\frac{c}{2} \frac{\Delta f_3}{f_T} = w$$

$$\eta_1 = -\frac{c}{2} \frac{\Delta f_1}{f_T} = u^*(0.342020) + w^*(0.939692)$$

$$\eta_2 = -\frac{c}{2} \frac{\Delta f_2}{f_T} = v^*(0.342020) + w^*(0.939692)$$

$$\eta_3 = -\frac{c}{2} \frac{\Delta f_3}{f_T} = w$$

From the above, it can be further written as follows:

$$\begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} = \begin{bmatrix} 0.321394 & 0 & 0.939692 \\ 0 & 0.342020 & 0.939692 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0.321394 & 0 & 0.939692 \\ 0 & 0.342020 & 0.939692 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 3.11210 & 0 & 2.9244 \\ 0 & 2.9244 & -2.74804 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix}$$

Now U, V, W components are as follows:

$$u = 3.11210 \eta_1 + 2.9244 \eta_3$$

$$v = 2.9244 \eta_2 - 2.74804 \eta_3$$

$$w = \eta_3$$

$$\text{Wind Strength (WS): } \sqrt{v^2 + u^2}$$

$$\text{Wind Direction (WD): } \tan^{-1} \frac{u}{v}$$

The obtained wind strength and direction parameters are drawn on a client area of a window. The procedure of wind record generation is discussed below.

In the late 1970s, the VDU resolution is 256 x 256 elements and the individual colors are generated by programming the intensity of the red, blue, and green guns in 16 discrete steps each (Thomson and Scheib, 1976). Although a total of 4096 colors are possible, each screen element is assigned just 4 color bits. Hence, only 16 colors are used to be shown at any one time on the screen. Thomson and Scheib (1978) have experimented by representing colors of increasing intensity in the order of 'ice blue' to 'white hot'. Thus, 'hot' spots of turbulence could be easily identified. It could facilitate a less-experienced observer to be more comfortable in the interpretation of the data. The magnitudes of the wind vector profiles obtained from the Doppler sodar systems are presented in the form of a color scheme (Figure 69).

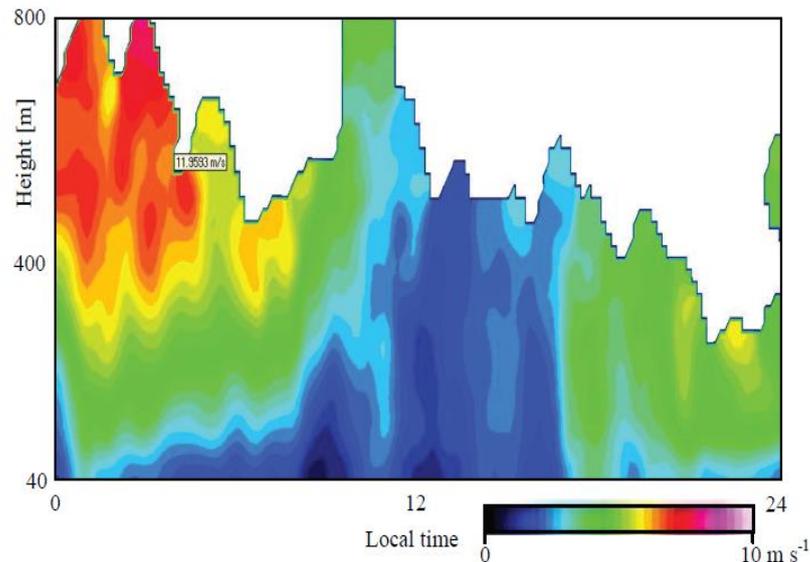


Figure 69 Wind speed profiles presented in the color scheme (Thomson and Scheib, 1978).

At the same time, Doppler sodar wind profiles are often presented in the form of arrow heads. In principle, such display of wind profiles is considered as an extension of the time-height plot of vertical wind velocity. The maximum

detectable wind magnitude and directions are 30 ms^{-1} and 360° , respectively. Each wind vector is displayed on the client area of the window as time-height plot. Wind vectors are derived from the first moments of Doppler spectra incoherently averaged for every 5 minutes, that is, 45 consecutive spectra. The arrow length indicates the magnitude of wind speed and arrow direction indicates the direction of the wind. As explained in the case of vertical wind, the application layer takes care of the communication between the data layer and OS layer to accomplish the wind profiles plotting.

The method `DrawArrowLine()` is developed to draw the geometry of the arrow at the given point of x,y coordinates. The wind magnitude and direction parameters are passed to the above function to draw the arrow head. The mathematical library function 'atan2' is used to calculate the orientation of arrow head lines. For any two real arguments x, y not both equal to zero, $\text{atan}(x,y)$ produces the angle in radians between the positive x -axis of a plane and point given by the coordinates on it. This angle is positive for counter-clockwise angle (upper half plane, $y>0$), and negative for clockwise angles (lower half plane, $y<0$). These positive and negative values are used in drawing the upper and down lines of the arrow head.

After determining the arrow length and head geometry, `LineTo()`, `MoveTo()` functions are used to draw the canvas by taking above values as arguments. Determination process of horizontal wind vector and drawing the time-height plot process is explained in the flowchart given in the Figure 70. In the process of real-time display of horizontal wind vector profiles, the program iterates the paint loop 15 times for each cycle. After completion of these iterations, the

paint thread is terminated and is re-initiated in synchronization with the data display thread. The signal flow chart for the horizontal wind vector plot is shown in Figure 70.

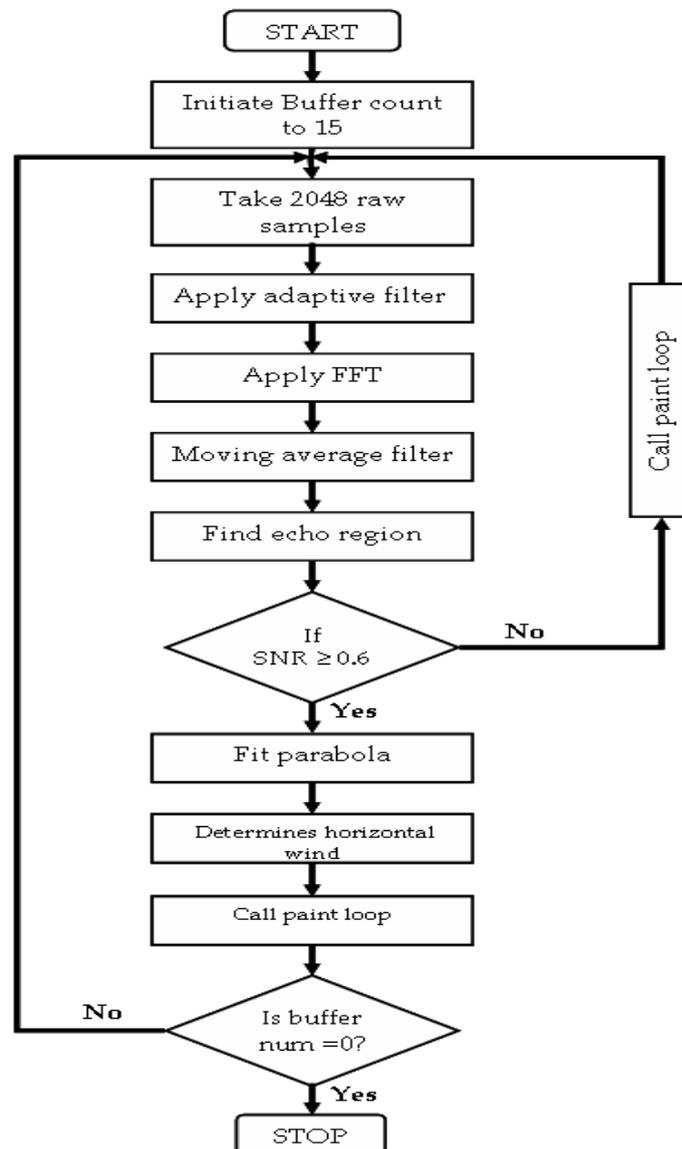


Figure 70 Signal flow chart for the horizontal wind vectors.

Winds are named according to the compass direction of their source. Thus, a wind from the north is called northerly wind. Figure 71 describes the sixteen principal bearings of wind direction. Most meteorological observations report wind direction using one of these sixteen bearings. The present system

produces the wind in terms of degrees azimuth values, and these values are converted to wind direction as described in the Figure 72.

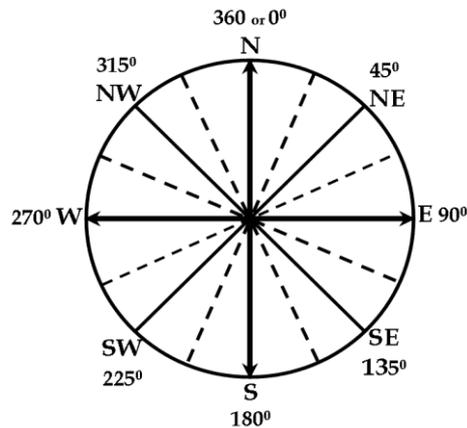


Figure 71 The wind compass describing the sixteen principle bearings used to measure wind direction.

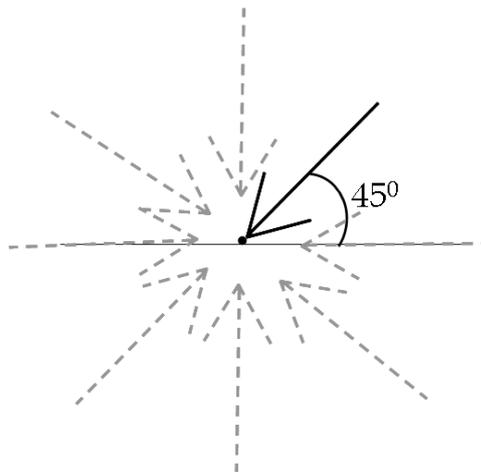


Figure 72 Arrow head in accordance with a wind direction and a magnitude.

The wind compass indicates the wind direction in terms of degrees from true north (0 or 360°). For instance, if the wind direction found to be 45° with a magnitude of 20 m/s, then an arrow is drawn on the client area of the window indicating northeast (NE) direction as depicted in Figure 72. For one cycle of sodar operation, the Y-axis is divided by the total number of range gates and

mark the centre points along the axis with the mean values of the corresponding range gate to draw the respective arrows. Here, the arrow head is drawn towards that point (Figure 72).

The snippet of the program to make the time-height plot of the horizontal wind is given below.

```
//*****//
Void Recorder::DrawArrowLine (CDC *pDCA, int X_AOldPoint, int Y_AOldPoint,double WS, double
WD)
{
    CPoint AOldPoint;
    CPoint ANewPoint;
    CDC *pDCHL = pDCA;
    double slopy, cosy, siny;

    AOldPoint.x = X_AOldPoint;
    AOldPoint.y = Y_AOldPoint;

    ANewPoint.x = X_AOldPoint+(WS*sin(WD));
    ANewPoint.y = Y_AOldPoint-(WS*cos(WD));

    pDCHL->MoveTo(AOldPoint);
    m_memdc3.MoveTo(AOldPoint);
    pDCHL->LineTo(ANewPoint);
    m_memdc3.LineTo(ANewPoint);

    pDCHL->MoveTo(ANewPoint);
    m_memdc3.MoveTo(ANewPoint);
    slopy = atan2(float(ANewPoint.y-AOldPoint.y),float(ANewPoint.x-AOldPoint.x));
    cosy = cos(slopy);
    siny = sin(slopy);
    WS = WS/2.5;
    pDCHL->LineTo(ANewPoint.x+int(-WS*cosy-(WS/2.0*siny)),
ANewPoint.y+int(-WS*siny+(WS/2.0*cosy)));
    m_memdc3.LineTo(ANewPoint.x+int(-WS*cosy-(WS/2.0*siny)),
ANewPoint.y+int(-WS*siny+(WS/2.0*cosy)));
    pDCHL->MoveTo(ANewPoint);
    m_memdc3.MoveTo(ANewPoint);
}
```

```

pDCHL->LineTo(ANewPoint.x+int(-WS*cosy+(WS/2.0*siny)),
ANewPoint.y-int(WS*siny+(WS/2.0*cosy)));

m_memdc3.LineTo(ANewPoint.x+int(-WS*cosy+(WS/2.0*siny)),
ANewPoint.y-int(WS*siny+(WS/2.0*cosy)));

}
//*****//

```

A sample horizontal wind profile plot obtained on September 1, 2014 between 16:21 and 17:21 hrs is illustrated in Figure 73. It shows different sizes of arrow heads indicating the wind speed and the corresponding direction. The date and time of the observations are given on top-right corner of the plot.

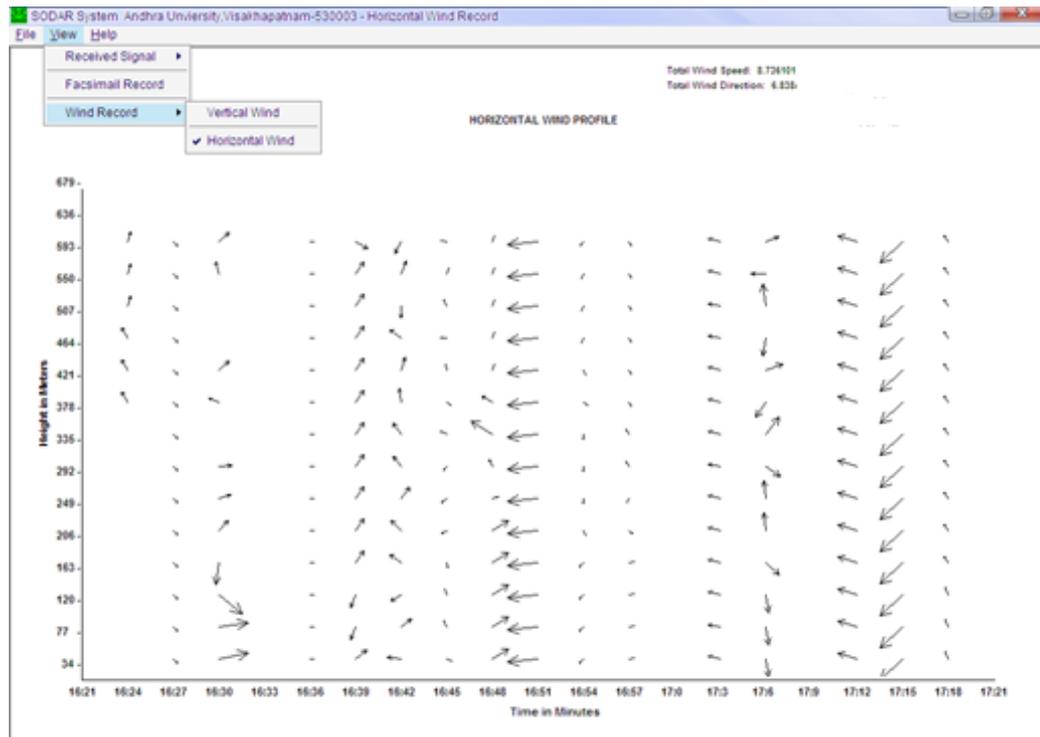


Figure 73 The window showing the horizontal wind vector plot.

The methodologies involved in the development of the tools are described with snippets of the programs and the flow charts. A sample plot for each of the vertical wind and horizontal wind vector obtained during the experimentation is also given. The interpretation of the data from the view point of atmospheric dynamics is done elsewhere. The two plots are of 60 min duration and are saved in .bmp file format. The raw data is saved in .wav file format. The tools are found to be more user-friendly and robust.

References

- Bradley, S., 2008: *Atmospheric Acoustic Remote Sensing*, CRC Press, USA.
- Elisei, G., Maini, M., Marzorati, A., Morselli, M.G., Fiocco, G., Cantarano, S., and Mastrantonio, G., 1986: Implementation of a multi-axial Doppler sodar system with advanced data processing, *Atmos. Res.* **20**, 109-118.
- Essenwanger, K.A. and Reinhardt, V.S., 1998: Sine output DDSs: A survey of the state of the art, *Proceedings of the 1998 IEEE International Frequency Control Symposium (Cat.No.98CH36165)*, 27-29 May 1998, pp 370-378, Pasadena, CA, USA, DOI: 10.1109/FREQ.1998.717930
- Kumar, V.V., Murthy, K.S.R.N., Vani, M., Rao, M.L.P., and Rao, M.P., 2005: Development of software utilities for on-board soundcard for Doppler sodar operation, *J. Acoust. Soc. India* **33**, 102-105.
- Kumar, V.V., Prasad, G., Madhu, K., Vani, M., Ramana, K.V., Rao, K.T. and Rao, M. P., 2008: Real-time display of Doppler sodar echo-signal intensity in a time-height plot, *Proc. National Symposium on Acoustics (NSA 2008)*, 525-529.
- Mastrantonio, G. and Argentini, S., 1997: A modular PC-based multiband sodar system, *Acoustic Remote Sensing Applications*, Narosa Publishing House, New Delhi, pp 105-116.
- Mastrantonio, G. and Fiocco, G., 1982: Accuracy of wind velocity determinations with Doppler sodar, *J. Appl. Meteorol.* **21**, 820-830.
- Owens, E.J., 1977: Microcomputer-controlled acoustic echo sounder, NOAA Technical Memorandum ERL WPL – 21, Wave Propagation Laboratory, Boulder, Colorado, USA, 1-76.
- Petzold, C., 2013: *Programming Windows*, 6th Edition, Microsoft Press, Washington, 1105 pp, ISBN: 978-0-7356-7176-8
- Rahim, R.A., Heng, T.C., Sulaiman, S., and Manaf, M.S.A., 2006: PC controlled function generator using direct digital synthesis (DDS) technology for electrical capacitance tomography, *Jurnal Teknologi*, **45**, 19-39, DOI: 10.11113/jt.v45.329
- Reinhardt, V.S., 1985: Direct digital synthesizers, *Proceedings of the Seventeenth Annual Precise Time and Timer Interval (PTTI) Systems and Applications*, 3-5 December 1985, Washington, DC, USA, pp 345-374,

- Reinhardt, V.S., 1993: Spur reduction techniques in direct digital synthesizers, *Proceedings of the 1993 International Frequency Control Symposium*, 2-4 June 1993, Salt Lake City, UT, USA, 1-12, DOI: 10.1109/FREQ.1993.367401
- Reinhardt, V.S., Gould, K., McNab, K., and Bustamante, M., 1986: A short survey of frequency synthesizer techniques, *Proceedings of the 40th Annual Symposium on Frequency Control*, 355-365, DOI: 10.1109/FREQ.1986.200968
- Reinhardt, V.S., Verdes, R.P., Shahriary, I., and Monica, S., 1989: Spurless fractional divider direct digital synthesizer and method, *U.S. Patent* No. 4,815,018 dated March 21, 1989, 1-8.
- Swanke, J.E., 1999: *VC++ MFC Extensions by Example*, CMP Books, 643 pp.
- Thomson, D.W. and Scheib, J.P., 1976: 'Applications of digital color display techniques for sodar', *J. Acoust. Soc. Am.* 60, Suppl., S32 pp.
- Thomson, D.W. and Scheib, J.P., 1978: 'Improved display techniques for sodar measurements', *Bull. Amer. Meteorol. Soc.* 59, 147-152.
- Wheatley, C.E., 1983: Digital frequency synthesizer with random jittering for reducing discrete spectral spurs, *U.S. Patent* No. 4,410,954 dated October 18, 1983, 1-10.
- Youngblood, G., 2002: A Software-Defined Radio for the Masses, Part 1, *QEX*, Jul/Aug, 1-10.
- Youngblood, G., 2002: A Software-Defined Radio for the Masses, Part 2, *QEX*, Sep/Oct, 1-9.