

SIDESCAN SONAR RAW DATA ACQUISITION SOFTWARE INTEGRATION IN AN UNDERWATER VEHICLE

Pablo Rivero¹, Ignacio Gonzalez², Higinio Gonzalez-Jorge¹, Pablo Alvarez², Ana Almecija², Abigail Buceta², and Silvia Torres²

(1) Escola de Enxeñería Industrial, Universidade de Vigo, Rúa Maxwell, s/n. Campus Universitario Lagoas-Marcosende 36310 Vigo (Pontevedra)

(2) Unidad de Tecnologías Marinas (UTMAR), Fundación CETMAR (Centro Tecnológico del Mar), C/Eduardo Cabello s/n 36208 Vigo (Pontevedra)

E-mail: igonzalez@cetmar.org

Tel:(+34) 986 247047

Abstract – *The integration of a sidescan sonar into the set of sensors of an Autonomous Underwater Vehicle (AUV) highly increases its possibilities for seabed cartography. CETMAR, in collaboration with the UVIGO, has developed a specific software in Python that interacts with the Tritech SideScan Sonar, running in a computer embedded in the vehicle itself, and enables the use of the device without installing proprietary software.*

Keywords – *AUV, Sidescan Sonar, Python*

I. INTRODUCTION

The development of a programme for the control and data acquisition from a sidescan sonar is part of a much larger project: the construction of a low cost autonomous underwater vehicle by the *Centro Tecnológico del Mar – Fundación Cetmar*, with the purpose of observation and analysis of the marine environment and any underwater infrastructure. The incorporation of a sidescan sonar increases the operational envelope significantly, thus enhancing the vehicle's chances. Therefore the gathering of images, the realization of accurate cartographies of the seabed, and the localization of objects is possible.

The sidescan sonar used is the *Tritech Seaking UAV/ROV*, specifically designed for its integration in an underwater vehicle. It consists of two transducers and a small control unit,

which integrates the rest of the electronics. The working frequency is 675 kHz, which allows making high-resolution soundings in a maximum range of 100 meters. Communication with the equipment can be achieved via ARCNET protocol (*Attached Resource Computer Network*) or RS-232 protocol. This allows communication data rates up to 115,200 Baud. In most of the commercial sonars the images gathered are displayed using proprietary software provided by the company itself. In this case, the program had limited features. It is not a multi-platform suite, which limited the chances of the sonar.

For this reason, a Python-based application has been developed from the scratch, thus making possible the control and display of the Seaneet subsea devices, adding new features to enable users to release the sonar's full potential. The program can run directly on the vehicle's main computer (ARM cortex A8 at 700 MHz with an operative system Debian GNU/LINUX). In this way, the program can serve as a basis for possible future developments related with the onboard processing of the sidescan imagery.

II. SOFTWARE IMPLEMENTATION

Tritech sonars are controlled by means of the exchange of packets with a pre-established order and structure. Their maximum extension is of

900 bytes. Each of them develops a specific task, which can range from running the sonar, changing operation parameters or sending the information from each pixel of the image. 73 different types of packets can be found, being most of them used for advanced functions such as the configuration or the calibration of the device. They begin with the character '@', followed by two blocks that represent the length of the packet. The eleventh byte indicates which type of packet it is.

The main task of the programme is to obtain data from the sonar and its graphic representation. Due to the manufacturer's limited amount of information about the sonar communication protocol, it was decided to build a sniffer, a bus monitor which intercepts the data sent through the serial port and sends them to another computer, where they are saved in a binary file. This allows to know exactly which data is being transferred between the host computer and the sonar head via RS-232 protocol. After studying the documents provided by the manufacturer and analysing the data obtained by the sniffer, the implementation of the programme can be carried out.

#	Length	Length	Tx Mode	Rx Mode	No. Byte	ID Pack.	Seq.	Node	Count	Sonar head	Status	Range	TAN
08	30	30	35	42	10	00	00	00	00	00	00	00	00

Sain	Slope	Heading offset		L. Line#	R. Line#	Step	Bearing	1500 BINS	1500 BINS
00	73	00	00	00	00	00	00	00	00

Fig. 1. Example of a 'mtHeadData' packet.

First, a library is created as a source for the main programme to create the packets that have to be sent to the device. These packets are created byte by byte, according to certain parameters such as resolution or rank. Furthermore, it is also necessary to parse the data received from the sonar via serial port: the algorithm looks for the packets and identifies them.

The *mtHeadData* packets contain all the

device sample data and, in addition, a copy of the device parameters. These packets are the replies to *mtSendData* command, which instructs the sonar to perform two sample sets. Each time the subsea device sends two data packages (corresponding to each of the sides), they are processed in order to obtain a new image line.

The programme has a PyQt-based Graphic User Interface (GUI), which allows the interaction with the software by means of objects and graphic figures, such as windows, icons or buttons, which eases the use of the programme for unexperienced users. The interface uses tabs for switching between the different tools of the programme:

- *Stream*, where the images from the sonar are displayed. After adjusting the device's parameters such as range scale and resolution, the sonar starts scanning. The image is automatically displayed on the screen, with a maximum resolution of up to 1630x700 pixels. In addition, the application allows data logging, saving the survey data (imagery, time and GPS data) in a binary file (.rlog)
- *Analysis*, where the processing of the images taken by the sonar takes place. It allows to play back recorded log data for its later editing or to apply any type of filters. These filters are applied in order to improve the quality of the images taken by the sidescan sonar for a later visual analysis performed by the user. The outcome of the images taken by sidescan sonars is usually poor in quality: high levels of noise, low definition of the edges and of other details are the most common issues.



Fig. 2. Screenshot of the user interface.

III. CONCLUSION

The objective of the implementation of a programme intended for data acquisition was to set it as a starting point for further developments, increasing its usefulness, which the proprietary

programme lacked. Thus, the vehicle could extract information about the surroundings and make autonomous decisions. By means of developing a simple low-cost software, allows for endless possibilities. Moreover, this software is multi-platform, user-friendly and easily adaptable to many other subsea devices.

IV. REFERENCES

- [1] X. Lurton, An introduction to Underwater Acoustics. London: Springer Praxis Publishing, 2003.
- [2] P. Blondel, The Handbook of SideScan Sonar. London: Springer Praxis Publishing, 2008.
- [3] Richard N. Czerwinski, Douglas L. Jones. William D.O. "Ultrasound Speckle Reduction by Directional Median Filtering", in International Conference on Image Processing, pp: 358 -361, Jun. 1995.
- [4] C. Agreda Ninot, "Técnicas de procesado de datos de sónar de barrido lateral". Proyecto final de carrera, Universidad Politécnica de Catalunya, 2007.
- [5] The Scipy Community (2014), "SciPy Reference Guide" [online]. Disponible en: <http://docs.scipy.org/doc/scipy/reference/index.html>