

Demonstration of a Partial Reconfiguration based Microphone Array Network Emulator

Bruno da Silva*, Federico Dominguez[§], An Braeken* and Abdellah Touhafi*

*Dept. of Industrial Sciences (INDI), Vrije Universiteit Brussel (VUB), Brussels, Belgium

[§]Escuela Superior Politecnica del Litoral (ESPOL), Guayaquil, Ecuador

Abstract—The current Micro-Electro-Mechanical System (MEMS) technology allows to deploy relatively low-cost Wireless Sensor Networks (WSN) composed of MEMS microphone arrays for accurate sound-source localization. However, the evaluation and the selection of the most accurate and power-efficient network’s topology is not trivial when considering dynamic MEMS microphone arrays. Despite software simulators are usually considered, they are high-computational intensive tasks which require hours to days to be completed. Our demonstrator is an FPGA-based network emulator, which provides a fast network design-space exploration. The user can easily evaluate a network’s topology with different nodes’ configurations and multiple sound sources in matter of seconds. An intuitive graphical user interface hides from the user the dynamic partial reconfigurations needed by the network emulator to set the node’s configurations. As a result, a probability map generated from the fusion of the output data from the nodes and an error on the estimation of the sound-source location are graphically represented.

I. DESCRIPTION

Our network emulator [1] is composed of several MEMS microphone arrays acting as WSN nodes. Each node monitors the surrounding acoustic environment and outputs the normalized acoustic power in a *polar steering response map* format. The flexibility of the nodes offers multiple possible configurations [2], [3] which can be modified in runtime.

Our network emulator consists of a front end running on a host and an FPGA as a back end (Figure 1). The front end is composed of a Matlab GUI, the XDMA driver [4] and the support for the reconfiguration through PCIe[5]. The front end generates the audio signals, calculates the signal propagation

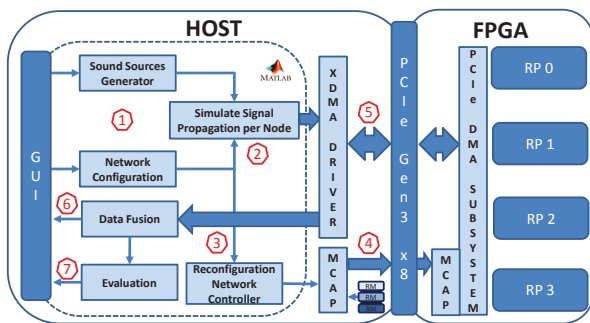
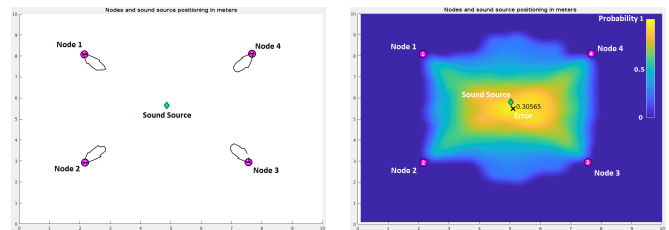


Fig. 1. Demonstration framework and execution flow. 1) The user defines the network topology, the node’s configuration, the sound-sources locations and their characteristics. 2) The sound sources are generated and the propagation of the signal to each node is simulated. 3) The configuration of each node is processed by the reconfiguration network controller to select the correct configuration of each RP. 4) The reconfiguration network controller interacts with the MCAP driver at the host side to select the new RM and to reconfigure the RPs through PCIe. 5) Once the reconfiguration of the RPs has finalized the data communication starts. The host formats the audio data to be compatible with the nodes at the FPGA side. 6) The data fusion of the *polar steering response maps* generated by each node in the FPGA is computed at the front-end. 7) The accuracy errors are displayed based on the known sound-source location and the data-fusion estimation.



(a) Fetching *polar steering response map* from the FPGA (b) Fusing the data to locate the sound source

Fig. 2. The data fusion is capable of simulating a sound field with multiple sound sources (green diamond) and multiple nodes (red circles). The FPGA generates the corresponding *polar steering response map* (a) which are then fed to the data fusion algorithm to generate a probability map (b) and estimate the localization error.

for each microphone in each node, selects the reconfigurable modules (RMs) for each reconfigurable partition (RP) on the FPGA and controls the data communications through PCIe. The back end is composed of a Xilinx UltraScale KU115 FPGA, where four RPs are defined. Each RP supports multiple nodes with the same RTL like the one presented in [2], [3]. The active RM per RP is determined by the front end, based on the node’s configuration selected by the user. The front end fuses the data received from each node via PCIe and estimates the localization error. Both results are displayed in the GUI (Figure 2). Based on the results, the user can change the network topology, the node’s configuration, the multiple sound-sources locations and their frequency characteristics to evaluate new solutions.

The main objective of our demonstrator is to present our partial-reconfiguration based network emulator by allowing the users to experience how the network’s topology and the node’s configuration determine the accuracy and the overall power consumption of the network.

REFERENCES

- [1] da Silva, B., et al. *A partial reconfiguration based microphone array network emulator*. Field Programmable Logic and Applications (FPL), 2017 27th International Conference on. UGent, 2017.
- [2] da Silva, B., et al. *Runtime reconfigurable beamforming architecture for real-time sound-source localization*. Field Programmable Logic and Applications (FPL), 2016 26th International Conference on. EPFL, 2016.
- [3] da Silva, B., et al. *Design Considerations when Accelerating an FPGA-Based Digital Microphone Array for Sound-Source Localization*. Journal of Sensors (2017): 2.
- [4] Xilinx, *Xilinx PCI Express DMA Drivers and Software Guide*; AR# 65444, <https://www.xilinx.com/support/answers/65444.html>. 2016.
- [5] Xilinx, *Bitstream Loading across the PCI Express Link in UltraScale Devices for Tandem PCIe and Partial Reconfiguration*; AR# 64761, <https://www.xilinx.com/support/answers/64761.html>. 2016.