WORKSHOP ON COMPRESSIVE SENSING AND ITS APPLICATIONS
21st July – 4th August, 2015, Kotor, Montenegro

An Architecture for Hardware Realization of Compressive Sensing Gradient Algorithm

Stefan Vujović, Miloš Daković, Irena Orović, Srdjan Stanković
University of Montenegro
Faculty of Electrical Engineering,

CSET
Compressive Sensing
and Emerging Technologies

Funded by the Ministry of Science of Montenegro under the World Bank loan.

Introduction

- Compressive sensing (CS) atracts significant research interest in last decade. Idea is to reconstruct sparse signal from reduced set od samples (measurements).

- Sparse signals are of interest in many applications like radars, sonars, biomedicine, etc.

- Two main research directions in CS:
    1. How to take measurements
    2. **How to reconstruct signal**

- This paper consider an arhitecture for hardware realization of gradient based reconstruction algorithm for sparse signals

Reconstruction algorithms

- There are many groups of algorithms that deals with sparse reconstruction problems: pursuit methods, convex relaxation, nonconvex relaxation, brute force based methods

- Convex relaxation algorithms are the most commonly used ones and are based on l_1-norm optimization

- This paper consider an hardware realization of one such algorithm [1]. Since considered algorithm is iterative, it is very important to find an optimal hardware realization in order to reduce computational time.

[1] LJ. Stanković, M. Daković, and S. Vujović, "Adaptive Variable Step Algorithm for Missing Samples Recovery in Sparse Signals," IET Signal Processing, vol. 8, no. 3, pp. 246 -256, 2014. (arXiv:1309.5749v1).

Gradient algorithm

- Start with signal where all missing samples are set to 0.
- For each missing sample we form two signals:

$$y_1^{(k)}(n) = \begin{cases} y^{(k)}(n) + \Delta & \text{for } n = n_i \\ y^{(k)}(n) & \text{for } n \neq n_i \end{cases}$$

$$y_2^{(k)}(n) = \begin{cases} y^{(k)}(n) - \Delta & \text{for } n = n_i \\ y^{(k)}(n) & \text{for } n \neq n_i \end{cases}$$

Δ is used here in order to decide should current sample value be increased or decreased

and calculate approximation of measure gradient as:

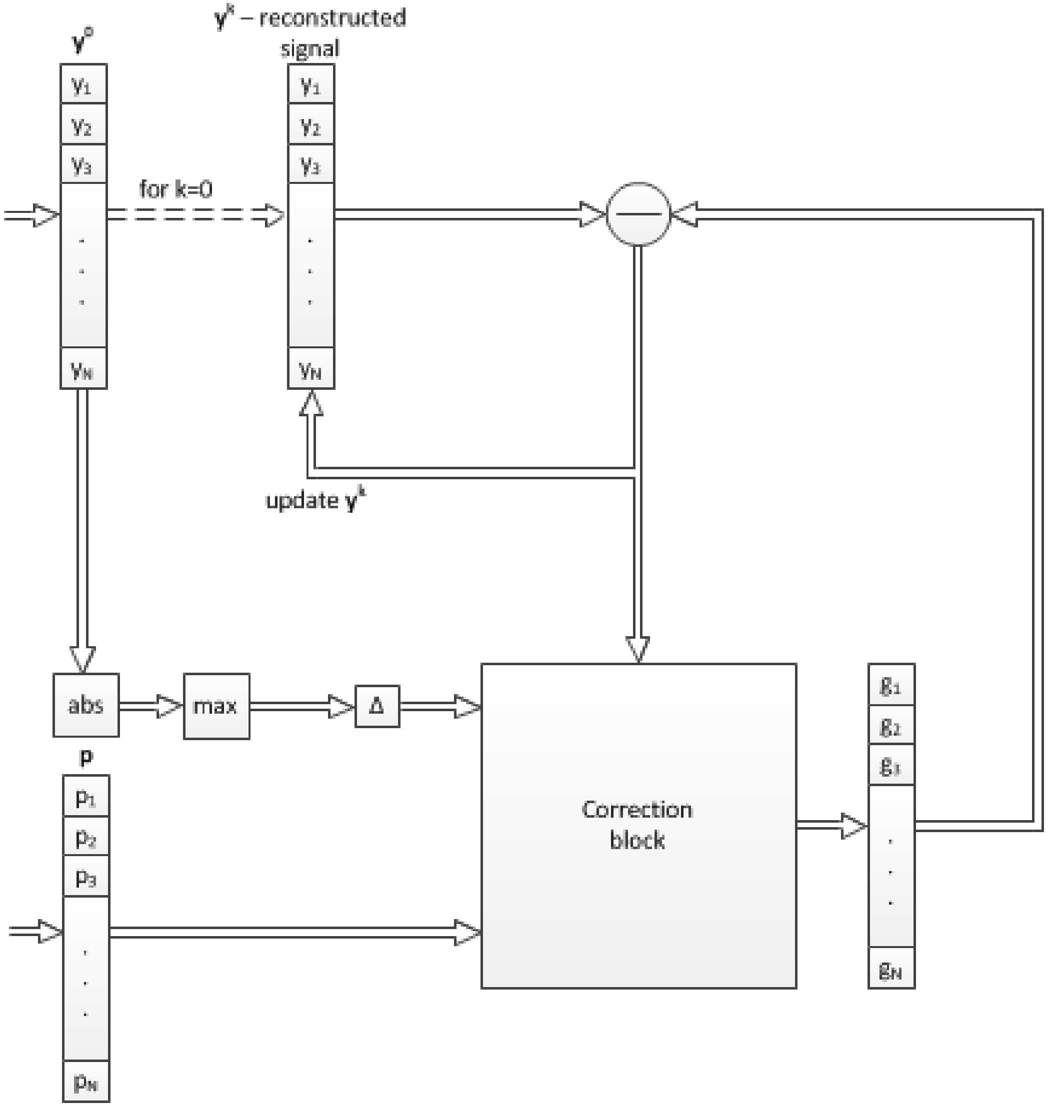
$$g(n_i) = \frac{\mathcal{M}_p\left[T[y_1^{(k)}(n)]\right] - \mathcal{M}_p\left[T[y_2^{(k)}(n)]\right]}{2\Delta}$$

T is signal transformation to sparse domain

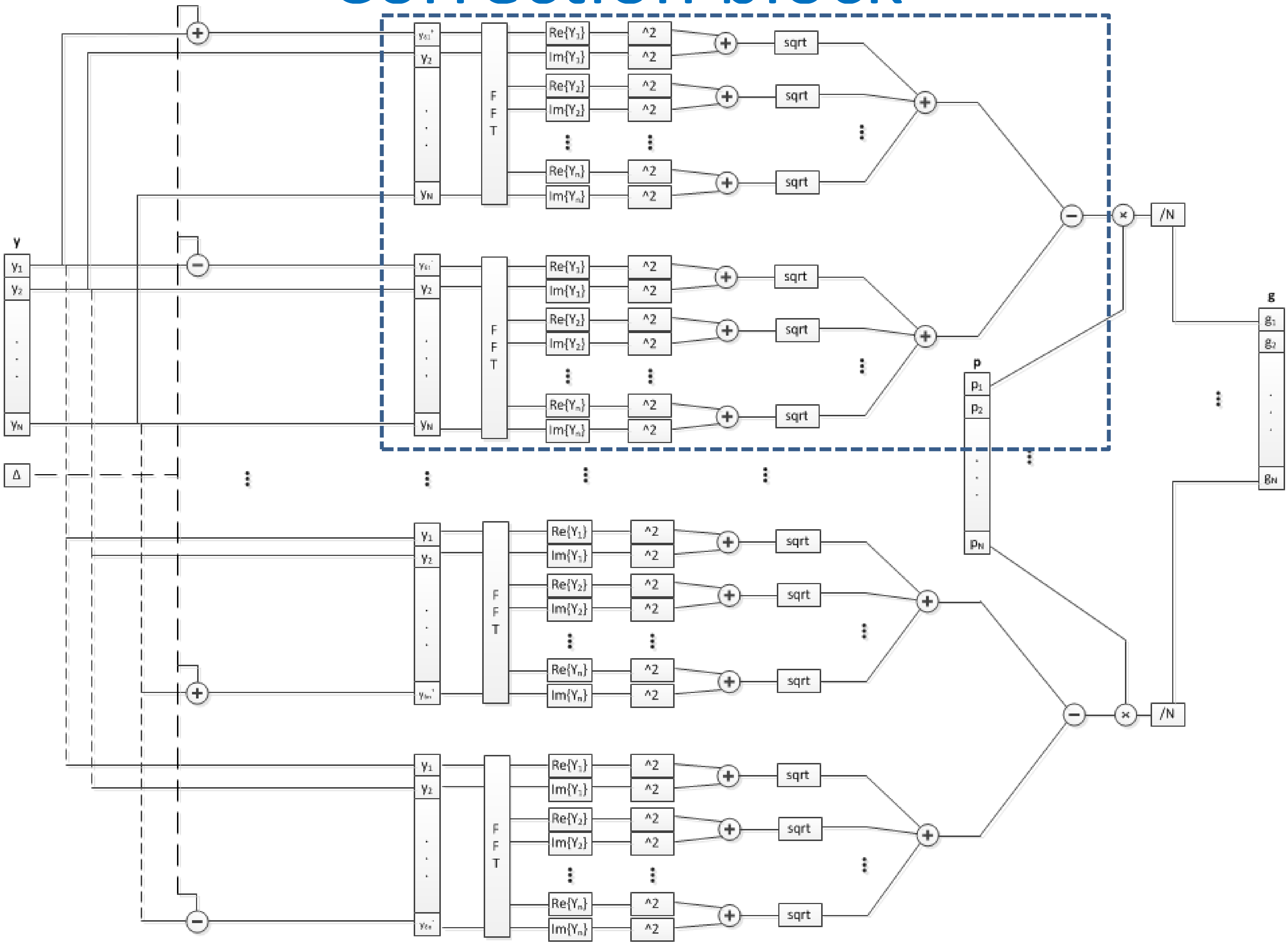
- Form gradient vector G and adjust missing sample values:

$$y^{(k+1)}(n) = y^{(k)}(n) - \mu G(n)$$

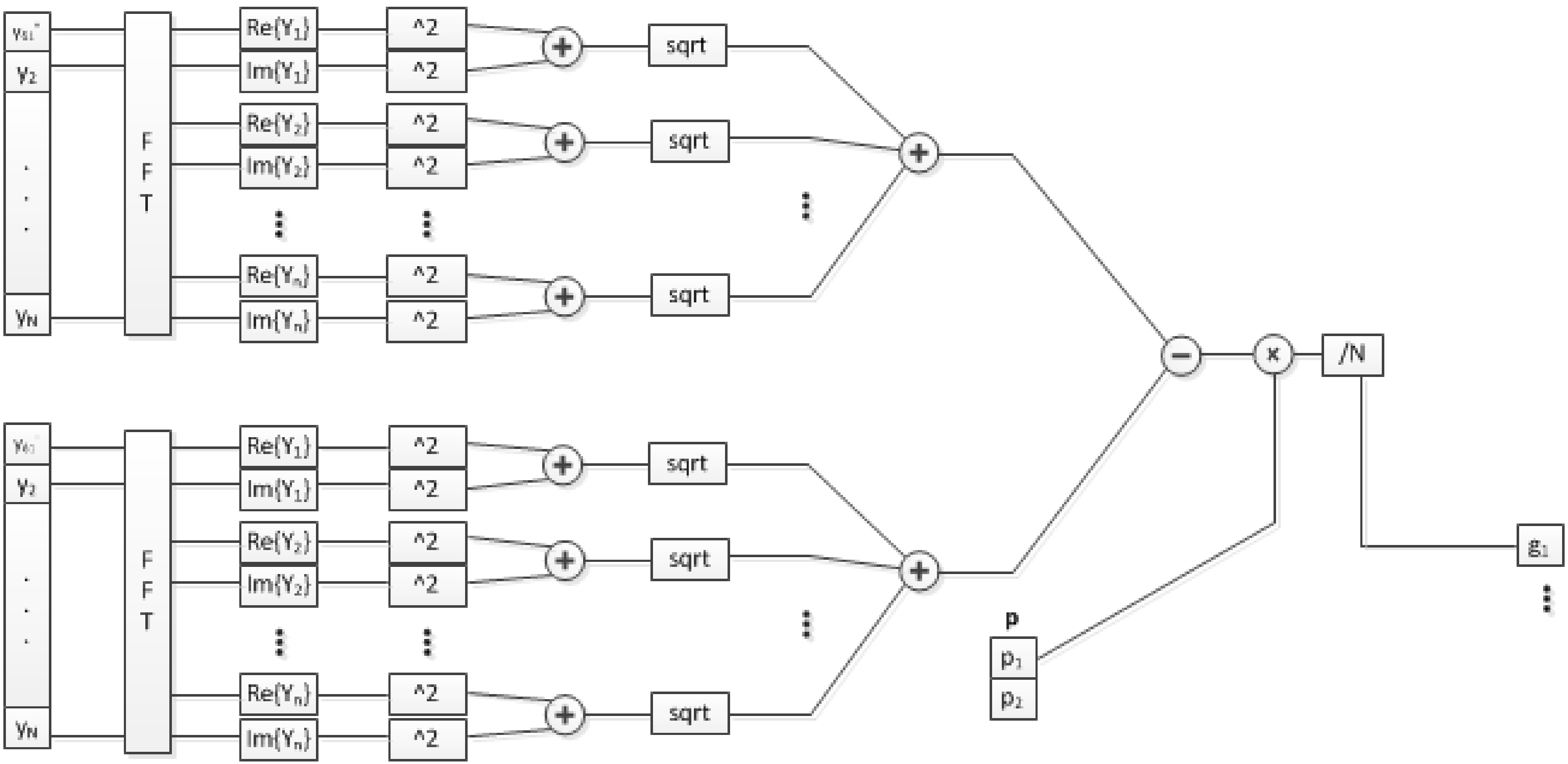
Block scheme



Correction block



Correction block



Clock cycles

- Square root block – 49 clock cycles (25 calculation, 24 issue rate)
- FFT block – $\log_2 N$ clock cycles, where N is signal length
- All other blocks are elementary operations which could be finished within few cycles.
- Note that, all clock cycles are sumed within one iteration of algorithm, and obtained number is multiplied with number of iterations

Conclusion

- Hardware implementations are of big importance for real applications

- One of the most important parameters which is used to determine the quality of proposed implementation is speed. So, it is used here as the measure of quality

- Proposed architecture is suitable for FPGA realization since realizations for all used blocks already exists

- Next step in our research will be to implement proposed realization on hardware

Thank you

Questions?