Adaptive Beamforming Algorithm in Real Numbers Arithmetic

Ilya V. Korogodin, Sergey P. Ippolitov, Ivan V. Lipa

National Research University Moscow Power Engineering Institute, Russia

Abstract— Controlled reception pattern antennas (CRPA, serpers) are very useful for telecommunication, radar and navigation receivers. They allow the forming of several independent virtual radiation patterns and to control the patterns' form. As a result, the receiver can gain useful signals and mitigate interferences and multipath signals. The mitigation performance is very high, so, for example, it's the only known way to improve GNSS receivers' antijam capability radically (from 40-50 dB to 90 and more dB).

CRPA operation method is quite simple. Both the useful signals and the interference signals are captured by several antennas. The signals are considered as relatively narrowband, so, the signal samples can be described as complex numbers with different arguments. After covariance matrix accumulation and inverse matrix problem solving, it is possible to compute complex weights for the signals combination. The combination adds the interference signals antiphasely and mitigates them.

The complex representation and the narrowband assumption for the signals are a convenient mathematical model. The model allows describing and implements signal delays as phase shifts. On other hand, it limits mitigated interferences bandwidths and imposes algorithm implementation in complex numbers.

The complex representation increases a computational resource overspending. In that case, it is required to use quadrature front-ends, four times more multipliers are needed to calculate the covariance matrix and implement filters. Inverse matrix problem solving is much more complicated for complex numbers matrices.

In the study, we propose an adaptive beamforming algorithm in real numbers arithmetic. It doesn't use narrowband assumption for the signals. The algorithm operates by finite impulse response filters for the signals alignment, real numbers for the covariance matrix and the signals representation.

The real number representation allows decreasing algorithm complexity: number of multipliers, inverse problem solving time and memory consumption.

1. INTRODUCTION

Let's consider an GNSS antenna array. A typical one contains between two and seven elements. Each element is a rectangular patch antenna. The bandwidth of the elements is quite tight, it is about several tens of MHz. The polarization is circular for the central frequency, but it degenerates to linear polarization on the edges of the frequency band.

The elements form either a square or a hexagon with the central element (see Fig. 2). Each patch is rotated on ninety degrees relative to the previous one for symmetry. The elements can be located both on a plane or on the spherical part. They are separated by a half wavelength usually, but this is not a strict constraint.

The antenna system is constructed to receive GNSS signals in the L-band (about 20 centimeters of wavelength). The size of the antenna system is from 10 to 50 centimeters.

The signals have between 2 and 20 MHz of bandwidth. As a result, the space correlation of the modulation function is between 30 to 300 meters and is much more then the antenna array size. In this condition, the antenna element signals have almost identical envelope delays, slightly different amplitudes, and dramatically different phases.

The GNSS signals can be separated by about a 30 MHz gap (for example, the GPS and GLONASS ones). So, the full frontend bandwidth can be up to about 50 MHz. The value determines the maximum interference bandwidth, which must be considered in the signal processing stage. The spatial correlation of such interference is about 10 meters. As a result, our assumption about envelopes delays, amplitudes, and phases are still appropriate.

The interference can be of both intentional or unintentional nature. A narrowband harmonic interference is the easiest case and can be mitigated without space processing. It is a typical

unintentional interference due to EMI troubles. The antijam capability is defined by the linear properties of the receiver frontend and the analog-digital converters. It is about 40 dB of interference to signal power ratio for modern civilian receivers. The value can be significantly increased up to 80 and more dB by means of frequency domain rejection techniques.

Wideband pseudorandom processes are a more complicated case. They are intended interferences usually. The interference has either chirp, BPSK or BOC spread-spectrum pseudorandom modulation. The interference spectrum covers the whole signal bandwidth and it cannot be rejected by the frequency domain algorithms appropriately. Space-Time adaptive processing is a very effective solution in this case. The antenna arrays allow for an increase in the receiver antijam capability from 45 up to 90 or more dB for such interferences [3].

The third interference type is a spoofing. In this case, interferences simulate useful navigation signals. This is the most dangerous situation due to low receiver antijam capability (low interference power) and possible control consequences.

2. SPACE-TIME ADAPTIVE PROCESSING ALGORITHM

The controlled radiation pattern antennas (CRPA, "serpers") are used to perform space-time adaptive processing (STAP) and to reject interferences. Several STAP algorithms are known, they can be presented as two serial steps: nullforming and beamforming. Nullformers reject interferences and they are described by the next equation [1]:

$$\xi = \mathbf{w}\,(\theta)^{\intercal}\,\mathbf{y}, \tag{1}$$

where $\mathbf{y}$ is an array (vector) of different time and antennas input signal values, ξ is a resulting array (vector) of cleared signals. Details of any certain STAP algorithms are in an approach of matrix $\mathbf{w}$ calculation.

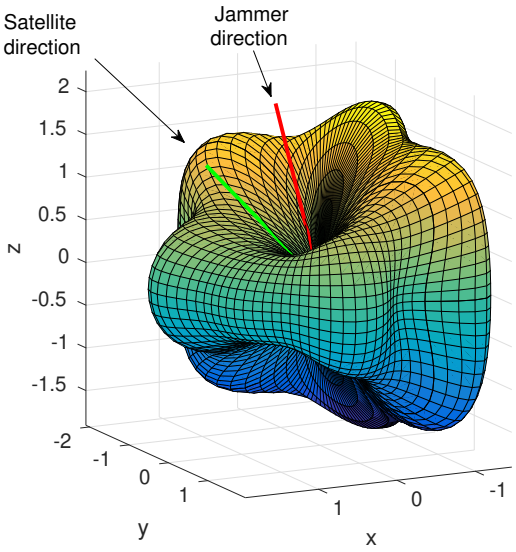

Figure 1: One of beamformer radiation patterns

Navigation satellite signals are weak, they don't even achieve the receiver thermal noise power. As result, antennas' signals almost don't have any correlation among each other in the normal reception conditions. The assumption is disrupted by a strong interference presence. The antenna signals become strongly correlated. The typical space and frequency domain interference mitigation algorithm operates like de-correlator. So, the simplest form of the matrix $\mathbf{w}$ is just an inverse covariance matrix $\mathbf{R} = E\,[\mathbf{y} \cdot \mathbf{y}^{\intercal}]$ for the vector $\mathbf{y}$:

$$\mathbf{w} = \mathbf{R}^{-1}, \tag{2}$$

and the algorithm common view is

$$\xi = \mathbf{R}^{-1}\mathbf{y}, \tag{3}$$

If $\mathbf{y}$ is presented in the form of complex numbers, the matrices $\mathbf{R}$ and $\mathbf{R}^{-1}$ are complex too. In accordance with the formula (3) it is necessary to multiply each element of $\mathbf{y}$ to the corresponding element of the inverse matrix and to add the results. In other words, the algorithm (3) is a space-time finite response filter with adaptive coefficients.

Covariance matrix is not known in practice and should be estimated. For example, we can use direct estimation:

$$\mathbf{R} = \sum_{l=1}^{L} \mathbf{y}\mathbf{y}^{\intercal}/L \tag{4}$$

Cleared signals ξ can be used for signals focusing:

$$y_f = \mathbf{H}\xi \tag{5}$$

where $\mathbf{H} = \mathbf{H}(\alpha, \beta)$ is a steering vector for alpha, beta direction. Several y_f signal for different directions can be form. As a result, we have an antenna system with several virtual radiation patterns and the space-time interference rejection function (see Fig. 1).

The navigation signal gain is relatively small due to a few antenna elements in the typical GNSS antenna array. Such antennas cannot have sharp beams of the radiation pattern. The typical gain is about 5 dB. But the beams can be very useful for multipath mitigation. Consequently, the beamforming technique can increase accuracy for pseudoranges and pseudophase observables [2].

The small antenna array cannot have sharp beams, but it can have very deep nulls for interference rejection. As result, nullformers dramatically increase the receiver antijam capability. The gain is about 50 dB for wideband interferences [3].

The additional bonus of nullforming technique is that it can reject spoofing interferences. It is necessary to form ten or more signals from the certain direction to perform a spoofing attack. Each signal can be such low as the real navigation signal. But common power of all ten signals is rather big and can be detected by a nullforming algorithm.

3. STAP ALGORITHM IMPLEMENTATION

STAP can be divided to high speed real-time ASIC/FPGA processing and low priority CPU firmware for process management. It is proposed to use three types of FPGA/ASIC logic modules (see Fig. 2):

- covariance matrix estimator (CVM),
- nullformer channel (NF),
- beamformer channel (BF).

The control unit can be implemented as CPU firmware.

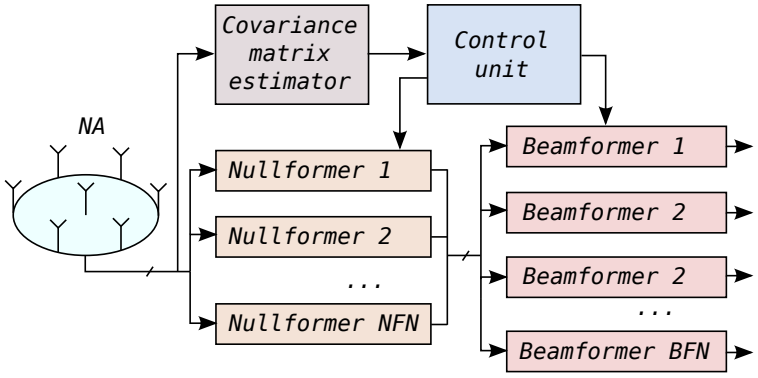

Figure 2: Beamforming/nullforming algorithm

The main parameters of the considering STAP algorithm are the number of antenna elements NA and the time depth NT (length of NF impulse response). The $\mathbf{y}$ vector length is the multiplication result of the parameters:

$$NY = NT \cdot NA. \tag{6}$$

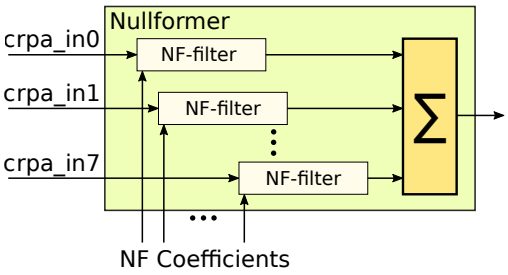

Figure 3: Nullformer module structure

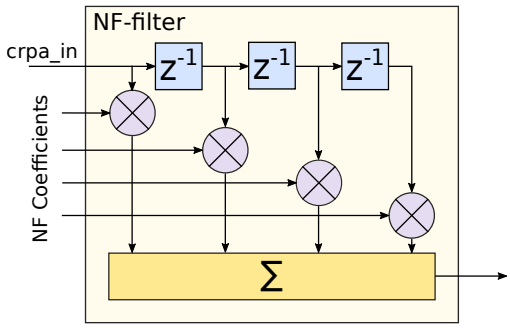

Figure 4: NF-filter module structure for $NT = 4$

The number of antennas NA is equal to the number NFN of NF instances. Each NF contains NA NF-filters (see Fig. 3).

Each NF filter is a finite impulse response filter of NT order (see Fig. 4).

The total multiplier numbers for all nullformers can be calculated as:

$$M_{NF} = NFN \cdot NA \cdot NT. \tag{7}$$

For example, a 7-element CRPA with $NT = 4$ demands $M_{NF} = 196$ multipliers.

As it was mentioned above, a covariance matrix estimation is used for NF's coefficients calculation. The covariance matrix is a $NY \times NY$ matrix. But due to correlation matrix properties and in assumption of stationarity of the random process the number of independent cells is much fewer:

$$NC = NA^2 \cdot (NT - 1) + \frac{NA^2 + NA}{2}. \tag{8}$$

For example, there are $NC = 175$ individual cells for 7-element antenna array ($NA = 7$) and 4-sample time depth ($NT = 4$). Reducing the number of antennas to 4 items decreases the number to 58 cells. The total multiplier number for covariance matrix estimation module is equal to the number of individual cells:

$$M_{CVM} = NC. \tag{9}$$

The beamformer instance structure is presented in fig. 5.

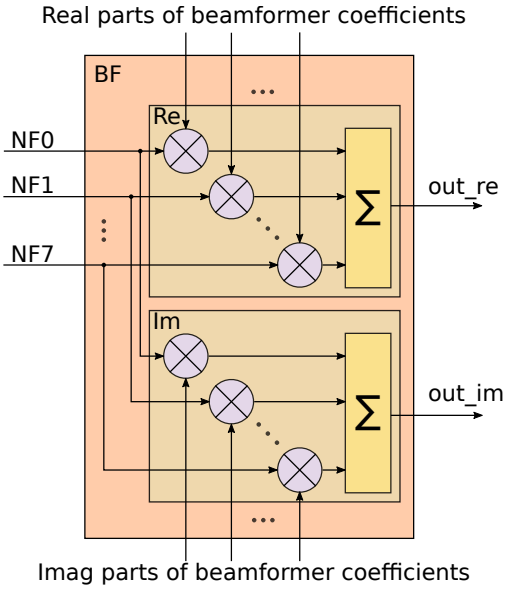

Figure 5: Beamformer module structure

There are $2 \cdot NFN$ multipliers per each nullformer instance. The total amount of multipliers for the beamformers implementation is:

$$M_{BF} = 2 \cdot BFN \cdot NFN. \tag{10}$$

For example, if the number of virtual radiation patterns BFN is equal to 12, then the total amount of multipliers for beamformers is 168 items.

The total big amount of multipliers is a challenging issue for the STAP implementation. Conventional approach supposes a complex signals utilization,and as a result, the multiplications must be performed in the complex form.

Let's consider the complex multipliers FPGA/ASIC implementation. In the case, the E and F values must be found for:

$$E + jF = (A + jB) \cdot (C + jD) = (A \cdot C - B \cdot D) + j(A \cdot D + B \cdot C) \tag{11}$$

There are two ways to perform the multiplication. The first one requires 4 real number multipliers and 2 real number adders and it is represented by the formula (11). The second one allows the change of a multiplier to three extra adders:

$$E + jF = (A \cdot C - B \cdot D) + j((A + B) \cdot (C + D) - A \cdot C - B \cdot D) \tag{12}$$

In the second case we need 3 real number multipliers and 5 real number adders to perform the complex number multiplication.

Both schemes (11) and (12) can be used in practice. The choice depends on certain hardware features. In the study we were aimed to Xilinx Zynq and ASIC implementations. After additional studies the first scheme was chosen.

Let's assume a real number multiplier can be implemented by one DSP-block of FPGA. Then the total number of consumed DSP blocks can be presented like:

$$N_{cmlx} = 4 \cdot M_{NF} + 2 \cdot M_{BF} + 4 \cdot M_{CVM}. \tag{13}$$

Substitutions give the next result:

$$N_{cmlx} = 4 \cdot NFN \cdot NA \cdot NT + 4 \cdot BFN \cdot NFN + 4 \cdot NA^2 \cdot (NT - 1) + 2 \cdot NA^2 + 2 \cdot NA. \tag{14}$$

For example, we need 1820 DSPs for $NFN = 7$, $NA = 7$, $BFN = 12$, $NT = 4$ and 812 DSPs for $NFN = 1$, $NA = 7$, $BFN = 0$, $NT = 4$.

Actually, the complex STAP algorithm requires the input signal preparation. It is typical to implement high-linear frontends in non-complex forms. So, we should perform the Hilbert transformation to get complex input signals. The order of each antenna filter is about 8, so the transformations require $8 \cdot NA$ real-number multipliers.

4. REAL NUMBERS STAP

As shown above, conventional complex number STAP requires about 2000 DSP blocks for the 7-elements CRPA case. It's a serious disadvantage and challenging issue to place the project to both FPGAs and ASICs. Such ASIC suffers from the overheating problem. In the case of FPGA implementation, the number of DSPs demand a high-end FPGA version (Zynq Z7100 [5] or Zynq ZU6 and higher [5]).

To solve the implementation problem, it's proposed in the paper to use a real number STAP algorithm modification. The modification is rather simple, all complex units are changed by real number versions: complex multipliers by real multipliers, complex adders by real number adders, complex signals by real parts of the signals. In this case, the total number of DSP-blocks can be presented as:

$$N_{real} = \cdot NFN \cdot NA \cdot NT + 2 \cdot BFN \cdot NFN + \cdot NA^2 \cdot (NT - 1) + \frac{\cdot NA^2 + \cdot NA}{2}. \tag{15}$$

For example, we need 539 (instead of 1820) DSPs for $NFN = 7$, $NA = 7$, $BFN = 12$, $NT = 4$ and 203 (instead of 812) DSPs for $NFN = 1$, $NA = 7$, $BFN = 0$, $NT = 4$.

It's not obvious, but the algorithm keeps working after the modification. We just should consider the algorithm as a space-time de-correlator instead of traditional antenna array with phase shifters.

Let's introduce an indicator of the interference rejection as the efficiency coefficient:

$$K_{eff} = \frac{P_{s,out}}{P_{n,out} + P_{j,out}} \Big/ \frac{P_{s,in}}{P_{n,in} + P_{j,in}} \tag{16}$$

where $P_{s,x}$, $P_{n,x}$, $P_{j,x}$ are the navigation signal power, the noise power, and interference power before and after the space-time processing.

A simulation was performed for both the complex number and real number algorithms. The simulation shows that the real number algorithm keeps efficiency in the case of the small number of interference sources. The efficiency coefficient in dependency of interference direction of arrival (DOA) is depicted on the Fig. 6. The simulation is performed for $NA = 7$, $NT = 4$ and $J/S = P_{j,in}/P_{s,in} = 90$ dB, the useful navigation DOA was 40 degrees. The model contained one interference source with a 40 degree DOA.

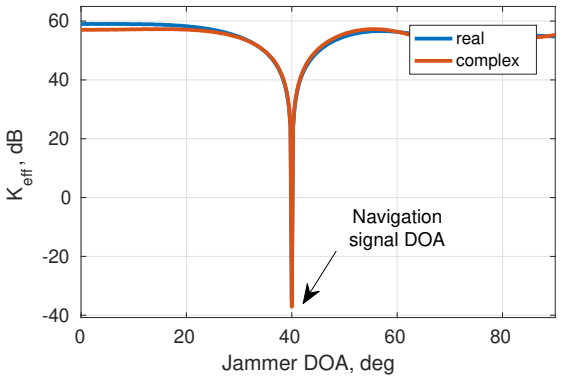

Figure 6: The efficiency coefficient for real number and complex number algorithms for different interference's direction of arrival

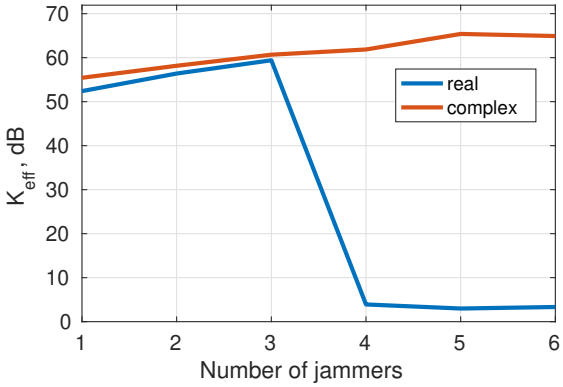

Figure 7: The efficiency coefficient for real number and complex number algorithms for different number of jammers

A practice rule is known for complex STAP: number of mitigated jammers is equal to number of antennas minus one [4]. For example, a seven element CRPA can reject up to six interferences.

This rule is disrupted for the real number algorithm case. The efficiency coefficient as function of the number of jammers is pictured on the Fig. 7.

Consequently from graphs, the real number version of the algorithm has more strict limitations to the number of jammers - about twice. And it is the main disadvantage of real number approach.

The real number STAP algorithm was implemented in a FPGA and ASIC. NF, BF and CVM modules were written in SystemVerilog language. Control unit was written in C++.

The common ASIC/FPGA logic module structure is depicted on the Fig. 8.

An example of FPGA resource utilization for the CVM (175 individual cells) and NF-filter modules in real number arithmetic is presented in the Table 1. The number of consumed DSP is rather close to estimated value: 182 vs 175 for CVM and 4 vs 4 for NF-filter. Total resources utilization is about 20% of the Xilinx Zynq 7045 chip for the whole project.

Table 1: Xilinx Zynq 7045 resources utilization for real number modules

Module	Slice LUTs	Slice Registers	Slice	LUT as logic	LUT as memory	LUT FIFO pairs	Block RAM tiles	DSP
FPGA	218600	437200	54650	218600	70400	218600	545	900
NF-filt	30	80	30	30	0	70	0	4
CVM	25877	28092	19632	25877	0	38798	0	182

A 7-element CRPA prototype was developed. The prototype utilizes 14-bit ADC and FPGA with the presented algorithms. A series of experiments was conducted. Experiments confirmed

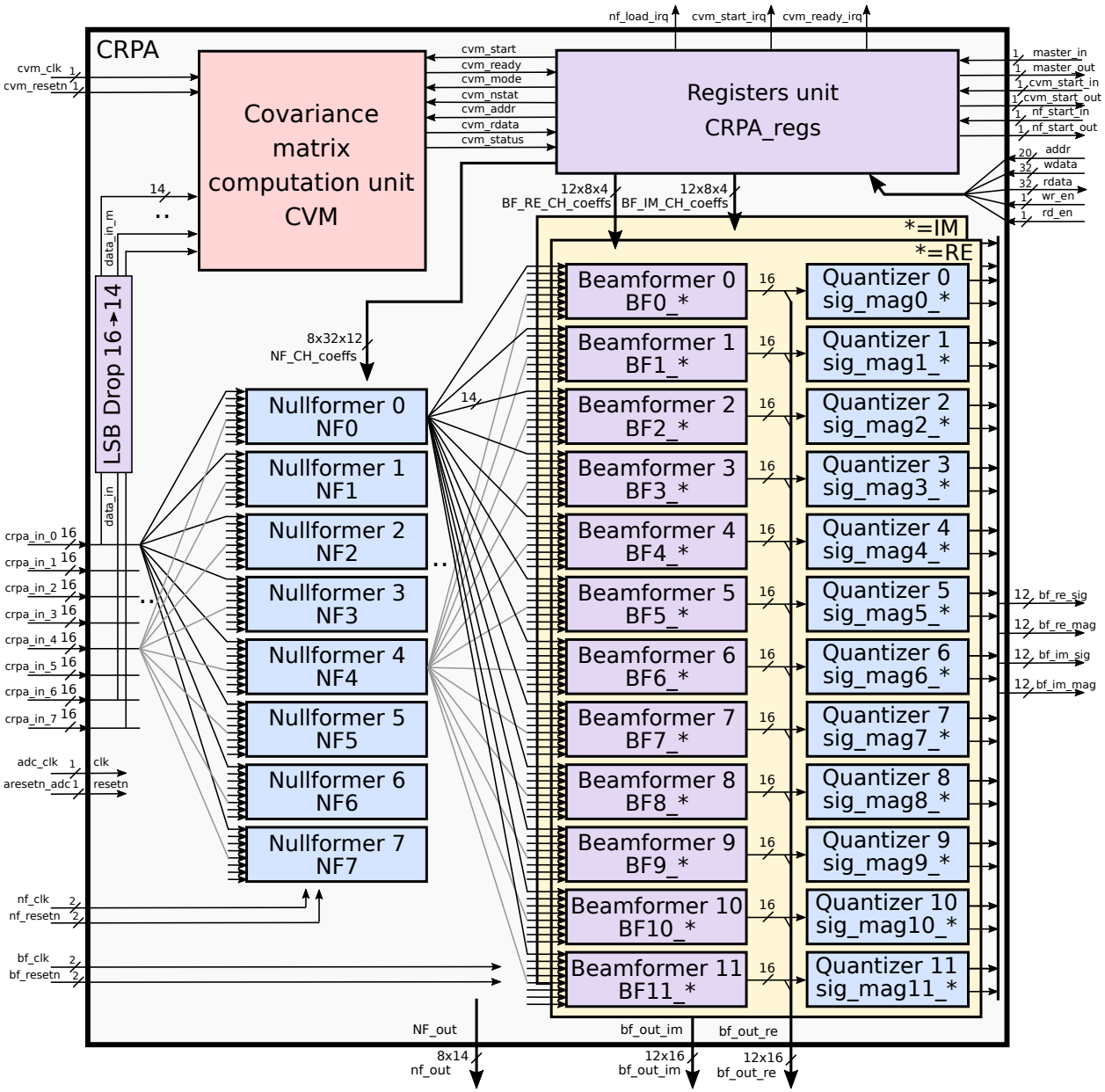

Figure 8: Space-time processing ASIC/FPGA logic module structure

the simulation predictions: the algorithm successfully mitigates interferences and consumes a small number of DSPs.

5. CONCLUSION

Conventional space-time processing algorithms operate in complex numbers. The complex representation allows to perform time shifting for narrow band signals by means of multiplication to complex weight, in other words, by means of phase shifting.

But the complex operations are rather complicated for calculation. The real number space-time processing algorithm is proposed. The real representation of values allows the decrease of FPGA/ASIC resource consumption about four times.

The efficiency of interference mitigation for the real number algorithm is equal to the efficiency of the complex algorithm in the case of small number of jammers. Otherwise, the complex algorithm has better performance.

ACKNOWLEDGMENT

This work was supported by the Ministry of Education and Science of the Russian Federation (project no. 8.9615.2017/BCh)

REFERENCES

1. R. A. Monzingo and T. W. Miller, Introduction to Adaptive Arrays. John Wiley & Sons, 1980.
2. B. D. V. Veen and K. M. Buckley, Beamforming: A versatile approach to spatial filtering, IEEE Signal Processing Mag., vol. 5, no. 2, pp. 4-24, April 1988.
3. Carlson, S.G., Popeck, C.A., Stockmaster, M.H., McDowell, C.E., "Rockwell Collins' Flexible Digital Anti-Jam Architecture," Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003), Portland, OR, September 2003, pp. 1843-1851.
4. Bartone, Chris, Stansell, Tom, "A Multi-Circular Ring CRPA for Robust GNSS Performance in an Interference and Multipath Environment," Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR, September 2011, pp. 1129-1139.
5. Xilinx Zynq-7000 SoC Data Sheet: Overview, DS190 (v1.11.1) July 2, 2018
6. Xilinx Zynq UltraScale+ MPSoC Data Sheet: Overview, DS891 (v1.7) November 12, 2018