Software Defined RADAR a State of the Art

Thibault Debatty
Royal Military Academy
Brussels, Belgium
thibault.debatty@rma.ac.be

***Abstract*—A software-defined radar is a versatile radar system, where most of the processing, like signal generation, filtering, up- and down conversion etc. is performed by a software. This paper presents a state of the art of software-defined radar technology. It describes the design concept of software-defined radars and the two possible implementations. A global assessment is presented, and the link with the Cognitive Radar is explained.**

I. INTRODUCTION

In many military operations radar, with its "day and night" and "all weather" capabilities, is a most valuable, even indispensable sensor. Traditionally, a radar system is built using dedicated hardware, such as ASIC-circuits. Such hardware is tailored to the task to be achieved and offers little or no reconfigurability.

Radar's possibilities are very broad and cover different fields like long- and short range air- and ground surveillance, target detection, target recognition and classification, weapon guidance, etc. These applications demand for diverse radar capabilities, leading today to the multiplication of highly specialised radar systems and antennas on the same platform (aircraft, ship, or other) [1].

Furthermore, new platforms are also equipped with a growing number of other RF sensors, such as electronic countermeasure (ECM) systems, electronic support measure systems (ESM), communication systems, navigations systems,... [2]

At the other hand, the place and weight available on any platform is limited, especially on unmanned aerial vehicles (UAV).

In order to accommodate these contradicting constraints, future radars will thus have to present multifunction capabilities. They should of course also achieve an increased accuracy, at a reduced price.

A new concept called "software-defined radar" might bring an answer to these needs. A software-defined radar is a versatile radar system, where most of the processing, like signal generation, filtering, up- and down conversion etc. is performed by a software. This software can be easily adapted, what gives software-defined radars many advantages such as:

- the possibility to create a *multipurpose radar*;
- the possibility to *re-use the same hardware*;
- an easier implementation of *advanced signal processing algorithms*;
- a *faster development and reduced price*.

Software-defined radar is a relatively new technology and, to the author's knowledge, no state of the art has been written yet. This paper will thus give a broad overview of software-defined radars. It is organized as follows. The design concept and the two main implementation possibilities are discussed in section II. Section III presents a global assessment of the technology. The link with the Cognitive Radar is explained in section IV and finally, conclusions are provided in section V.

II. DESIGN CONCEPT

A software-defined radar applies the same principles as a software-defined radio: components that have typically been implemented in hardware (e.g. mixers, filters, modulators, demodulators, detectors etc.) are implemented using software on a computer or other programmable device, usually a field-programmable gate array (FPGA) [3], [4].

This kind of implementation has been made possible quite recently thanks to the huge progress made in computational power of programmable devices and computers.

The basic design concept of a receiving and transmitting module consists of a wideband RF front end, a down- respectively up-converter, an AD/DA converter, a FPGA and a computer [5].

Two variants exist:

- most of the processing takes place on the FPGA, and the computer only controls the radar and displays the results;
- most of the processing takes place on the computer, and the FPGA only performs some basic operations (e.g. decimation).

A. FPGA implementation

A field-programmable gate array (FPGA) [6] is an integrated circuit designed to be configured after manufacturing and is hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL) such as VHDL and Verilog. FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together". In most FPGAs, the logic blocks also include memory elements.

The radar presented in [3] is a pulse compression radar built using a commercially available Tekmicro Triton VXS-1 card. The card is equipped with a Xilinx Virtex-II Pro FPGA, a Gigabit Ethernet controller for the connection with

978-1-4244-6459-3/10/$26.00 ©2010 IEEE

the computer, a 10 bit ADC and a 12 bit DAC. Both converters can operate at 2 GSamples/sec, giving a Nyquist digital signal bandwitdth of 1 GHz. Most of the processing operations take place on the FPGA:

- transmission of the emitted waveform to the DAC at a steady pulse repetition frequency (PRF);
- recording of corresponding echoes;
- matched filtering and quadrature demodulation of the echoes.

The desired waveform is selected and generated on the computer and transferred to the Triton card together with the rest of the radar parameters (PRF, pulse length, etc.). The radar can operate using any kind of waveform within the maximum 1 GHz bandwidth.

The memory of the Xilinx Virtex-II Pro FPGA installed on the Triton card consists of 328 blocks of 18 Kb each, which limits the radar to the recording and processing of only 16 kSamples. If the radar operates with a full bandwidth waveform (i.e. 2 GSamples/sec), this results in a sampling range of 1229 meters, what is quite limited for a real radar.

The implementation of a radar algorithm on a single FPGA is thus usually limited by the number of logic blocks and memory elements available on the FPGA. If the algorithm to implement is more complex, the radar will require multiple FPGAs, what makes the design of the radar more complex, and reduces its reusability.

B. Computer implementation

In this case, most of the processing takes place on a computer, while only some basic operations (e.g. decimation) take place on the FPGA. This is a very flexible solution, that makes it much easier to implement very advanced algorithms. The only limit is then usually the processing speed of the computer.

The Reconfigurable Software Defined Passive Radar described in [7], is a passive radar that uses FM radio emitters. It relies on 8 omni-directional antennas and Digital Beam Forming (DBF) to form reference beams in the direction of FM radio emitters and surveillance beams in the direction of targets. The complete algorithm is run on a PC equipped with an Intel Core 2 Quad processor. The processing power of this kind of PC limits the system to the real-time processing of only one emitter and one surveillance beam.

Some, usually experimental, software-defined radars are based on the GNU Radio project [8], such as the radar described in [9].

GNU is a recursive acronym that stands for "GNU's Not Unix". The GNU Project rally volunteers whose original goal was to develop a completely free operating system, which is now known as "Linux". The GNU Project currently reassembles many free software development projects, including the GNU Radio project.

GNU Radio is a free software development toolkit that, when combined with minimal hardware, allows for the construction of RF devices. It is compatible with many commercially available ADC/DAC boards, but the GNU Radio developers created a dedicated board, called Universal Software Radio Peripheral (USRP [10]). It consists of:

- Four 12 bit, 64 MSample/s ADCs;
- Four 14 bit, 128 MSamble/s DACs;
- An Altera Cyclone FPGA;
- A USB 2.0 controller for the connection with the computer;
- Four extension sockets in order to connect 2 or 4 daughterboards.

Daughterboards serve as RF front end. There are various models that work at different frequencies.

GNU Radio and the USRP have already been used to build a GPS receiver, an FM radio receiver and an FM radio transmitter, a digital television decoder, a Digital Audio Broadcasting transmitter, and so on.

The USRP is an inexpensive board (about 700$ [11]), but has limited performance. The GNU Radio team also created another board, the USRP2 [12] that is slightly more expensive (about 1400$ [11]), but performs better:

- Two 14 bit, 100 MSample/s ADCs;
- Two 16 bit, 400 MSample/s DACs;
- A Xilinx Spartan 3-2000 FPGA;
- A Gigabit Ethernet interface for the connection with the computer;
- Two extension sockets for 1 or 2 daughterboards.

The greatest limitation of these cards is the connection with the computer, which, as can be seen in table I limits applications to maximum 30 and respectively 62.5 MSamples/s.

TABLE I
THROUGHPUT IN MBITS/S AND MSAMPLES/S OF SOME COMMON COMPUTER INTERFACES, FOR 16 BITS/SAMPLE

	Mbit/s	**MS/s (16bit/S)**
Fast Ethernet	100	6.25
Gigabit Ethernet	1000	62.5
USB2.0	480	30
PCI 32-bit/33 MHz	1067	66.69
PCI Express x1	2000	125
PCI Express x8	16000	1000
(e)SATA	2400	150

III. GLOBAL ASSESMENT

A. Advantages

Software-defined radars present many advantages such as:

- the possibility to create a *multipurpose radar*;
- the possibility to *re-use the same hardware*;
- an easier implementation of *advanced signal processing algorithms*;
- a *faster development and reduced price*.

All these advantages are detailed below.

1) Multipurpose Radar: With software-defined radars, the processing algorithm is implemented in software, which can easily be loaded during operation. It is thus possible to dynamically switch between different radar modes, or between a radar and another RF application. The same antenna and transmission (Tx) - reception (Rx) module can thus be used for different RF applications.

If the radar is equipped with a wideband fully digital phased array antenna, where digitization of the antenna signals on transmission and reception occurs at the antenna element level, the possibilities become incredibly vast.

Some of the applications that an airborne software-defined radar equipped with such an antenna could perform are described in [13], [2]. They include:

- Ground Moving Target Indication;
- SAR;
- Inverse SAR;
- Communication, navigation and identification systems: Satellite Communications (SatCom), TACtical Air Navigation (TACAN), Radar Altimeter, Identification Friend or Foe (IFF),...
- Electronic Countermeasures (ECM) and Electronic Support Measures (ESM) systems;
- Weather radar;
- Weapon location for the detection and tracking of artillery shells;
- For a UAV: sense and avoid sensor that detects other flying objects to take evasive actions.

The possibility to create a multipurpose radar, that combines different RF applications in one Tx-Rx module and one antenna, is especially interesting for aircrafts, and UAVs in particular, where the available space and weight (e.g. for antennas) are limited.

Examples of multipurpose software-defined radars include the ARS-800-2 Reconfigurable Maritime Patrol Radar and the Multi Purpose RF System Demonstrator.

The ARS-800-2 Reconfigurable Maritime Patrol Radar presented in [14] is an airborne software-defined radar that can switch between different modes during the flight. For each mode, the appropriate algorithm is loaded on a FPGA by means of a PC. The currently available modes are:

- Real-time unfocused SAR;
- Real-time oil slicks and pollution detection (using SAR);
- Real-time representation of AIS (Automatic Identification System) targets.

The Multi Purpose RF System Demonstrator presented in [15] combines two RF applications on the same hardware: a classical pulse compression radar and a jammer. As for the ARS-800-2, the appropriate algorithm is loaded on the FPGA by a PC.

2) Re-use of hardware: Software-defined radar technology allows to re-use radar parts, which can be either new reusable building blocks or old radar parts. In both case, creating a radar consists in choosing the right parts and developing the appropriate software.

As explained in [16], there are projects for the creation of Scalable Multifunction RF (SMRF) building blocks. These building blocks will be designed to be compatible between companies and nations. Ideally, building a new software-defined radar will consist in choosing the right blocks, and developing the appropriate software.

Software-defined radar also allows for the re-use of old hardware. The VHF surveillance radar described in [17] was built using the mechanisms and antenna of an old P-18 radar of Russian make.

3) Advanced signal processing algorithms: For years, researchers have developed and tested new advanced radar signal processing algorithms like Space-Time Adaptive Processing (STAP), Synthetic Aperture Radar (SAR) or Inverse SAR (ISAR) on their computers, while the processing in a real-world radar was performed by specialized hardware circuits [3]. The conversion of the algorithm from the software world of computers to the hardware world of real radars was quite challenging.

Because the algorithm of a software-defined radar is implemented in software, it facilitates the implementation, test and further improvement of an advanced algorithm.

4) Faster development and cheaper radars: The possibility to use the same building blocks for different applications allows a faster and cheaper development because the hardware already exists and only the software part of the radar has to be developed.

Although both the initial investment in research and development and the initial price of the blocks themselves might be higher, it will certainly drop when they become widely used. Ultimately, the use of these standard building blocks will thus reduce the production costs of the radar system.

In addition, the possibility to test the radar under development without having to build a new prototype for each version of the algorithm will also allow a faster and cheaper development.

Finally, the possibility to create a multifunction radar rather than different individual systems will also reduce the total cost.

B. Current shortcomings

Two main challenges remain for creating multifunction software-defined radars running advanced algorithms:

- Designing adapted antennas;
- Increasing the processing speed.

In addition, it will also be important to:

- Reduce the data flow rate to help improve the processing speed;
- Design faster A/D converters in order to increase the bandwidth of software-defined radars.

1) Antennas: Basic software-defined radars can work with any kind of antenna, but in order to leverage all the possibilities described above, multifunction software-defined radars will require antennas responding to various needs [13], [2]:

- Wideband or multiband capacity in order to accommodate the different functions;
- Availability as standard building blocks to simplify the design of new systems;
- Preferably a fully digital array conception.

TABLE II
COMPARISON OF A MODERN CPU AND GPU

	Intel Core i7 965 Extreme Edition	NVidia GTX 285
Cores:	4 cores	240 shaders, also called CUDA Cores
Frequency:	3200 MHz	648 MHz
Processing power:	70 GFLOPs	1062 GFLOPs

2) Processing speed: For computer implementation, the processing speed is in fact the bottleneck of software-defined radars. One solution might be the use of Graphics Processing Units (GPU). A GPU is a processor specialized in 3D graphics rendering. It has a highly parallel structure that makes it more effective than general-purpose processors for some types of algorithms.

Table II compares a modern desktop CPU (Intel Core i7 965 Extreme Edition) with a modern GPU (NVidia GTX 285) and shows that the GPU can be very useful for massively parallel applications, such as radar signal processing.

The two biggest GPU manufacturers, NVidia and AMD, developed libraries that allow programmers to use their GPU for non graphic applications. This is usually called general-purpose computing on graphics processing units (GPGPU, also referred to as GPGP) or stream processing. NVidia's library is named the Compute Unified Device Architecture (CUDA) [18] and AMD's is called FireStream [19].

Previous studies showed CUDA might speed up Radar Pulse Compression computing by a factor of 4 [20] and SAR processing by a factor of 15 [21], with respect to the reference workstation.

Until recently, CUDA-enabled cards only existed for desktop computers and servers, while real life systems are built using cards specially designed for embedded systems. CUDA was thus not available for real life radars. But the situation changed in 2009 when GE Fanuc released a first 3U-VPX based board [22] featuring a NVidia GT 240 CUDA-enabled GPU [23]. In the future, a rugged version will be available, as well as 6U-VPX cards featuring combinations of Intel dual core processors and NVidia CUDA-enabled GPUs. These kinds of boards will certainly allow the creation of real-life software-defined radars running advanced algorithms resulting in an increased performance.

IV. COGNITIVE RADAR

Software-defined radar is also the enabler that will make possible the implementation of the Cognitive Radar described in [24].

A Cognitive Radar must fulfil three conditions: intelligent signal processing, feedback from receiver to transmitter and detection through tracking.

A. Intelligent signal processing

A cognitive radar uses prior knowledge to perform the best possible signal processing, using two types of information:

- Information about the environment previously acquired by the radar itself. This is usually called adaptive processing.
- Information delivered by external knowledge sources, known as knowledge-based processing. These knowledge sources can be either external databases, other radars or other sensors. The latter is different from data fusion because the information delivered by other sensors is used to adapt the internal signal processing of the radar. In the case of data fusion, the information from other sensors is only 'mixed' with the output signal of the radar, without influencing its behaviour.

An example of knowledge based processing using an external database is the CFAR knowledge-aided Radar Detection presented in [25]. It performs knowledge-aided STAP: the signal from the cell that is being analyzed is compared with the signal from cells that present the same proportion of physical materials at the surface of the earth (i.e. land cover). Land covers include grass, asphalt, trees, bare ground, water, etc. The system uses National Land Cover Data (NLCD) published by the United States Geological Survey (USGS) [26].

B. Feedback from receiver to transmitter

A Cognitive Radar has a feedback system from the receiver to the transmitter, that allows the adaptation of the emitted wave (e.g. frequency, length, code, ..). In the case of digital beam forming or with a phased array antenna, the radiation pattern can naturally also be adapted.

An example of a radar with feedback from receiver to transmitter is the Target Matched Illumination Radar described in [27]. This is a Frequency Modulated Continuous-wave (FMCW) radar that analyses the linearity of the received signal. The best frequency band is then chosen to avoid interferences from other radars, jamming from hostile sources or propagation irregularities.

C. Detection through tracking

Finally, a cognitive radar performs detection through tracking, meaning that:

1) For a given search area, radar returns are collected over a certain period of time;
2) For each range-azimuth resolution cell, the probability of the cell containing a target is computed;
3) With the evolution of target probability distribution resulting from the recursive computation of step 2 over time, target tracks are detected, and corresponding hard decisions on possible targets are subsequently made.

V. CONCLUSION

This paper described the design concept of software-defined radars, along with the two possible implementations. The advantages and shortcomings of this technology were presented, and the link with the Cognitive Radar was explained.

Software-defined radar technology is a very promising concept, that will make it easier to create multipurpose, high performance radars at a reduced price. Challenges remain concerning adapted antennas and global processing power. Despite the current limitations, software-defined radars will certainly be the radars of the future.

References

[1] "Software defined radar - executive summary," in *RTO-MP-SET-136*, 2009.

[2] C. Renard and M. Soiron, "Wideband multifunction airborne antennas," in *International Radar Conference*, 2009.

[3] H. Nicolaisen, T. Holmboe, K. V. Hoel, and S. Kristoffersen, "High resolution range-doppler radar demonstrator based on a commercially available fpga card," in *IEEE RADAR 2008*, 2008.

[4] "Software defined radio," http://en.wikipedia.org/wiki/Software-defined_radio.

[5] J. Cechak and P. Bojda, "Software defined radar and FPGA limits," in *RTO-MP-SET-136*, 2009.

[6] "Field-programmable gate array," http://en.wikipedia.org/wiki/Field-programmable_gate_array.

[7] M. Malanowski, K. Kulpa, M. Mordzonek, and P. Samczynski, "Parade - reconfigurable software defined passive radar," in *RTO-MP-SET-136*, 2009.

[8] "GNU radio," http://gnuradio.org.

[9] L. K. Patton, "A GNU radio based software-defined radar," 2007.

[10] "GNU radio - USRP," http://gnuradio.org/redmine/wiki/gnuradio/USRP.

[11] "Ettus research llc," http://www.ettus.com.

[12] "GNU radio - USRP2," http://gnuradio.org/redmine/wiki/gnuradio/USRP2.

[13] A. Huizing, M. Otten, W. van Rossum, R. van Dijk, A. Maas, E. van der Houwen, and R. Bolt, "Compact scalable multifunction RF payload for UAVs with FMCW radar and ESM functionality," in *International Radar Conference*, 2009.

[14] K. Kulpa, J. Misiurewicz, M. Smolarczyk, and A. Gados, "Ars-800-2 - reconfigurable maritime patrol x-band radar," in *RTO-MP-SET-136*, 2009.

[15] T. Holmboe, S. Johnsrud, K. V. Hoel, and S. Kristoffersen, "Multi purpose rf system demonstrator base on a commercially available adc/fpga/dac board: High-resolution radar and drfm-base radar jammer in the same hardware," in *RTO-MP-SET-136*, 2009.

[16] A. Huizing, "Scalable multifunction radio frequency systems," in *Defence Sensors Technology Conference*, 2009.

[17] I. Biro, B. Babjak, and B. Eged, "Sofware defined vhf surveillance radar system," in *RTO-MP-SET-136*, 2009.

[18] "CUDA zone – the ressource for cuda developers," http://www.nvidia.com/object/cuda_home.html.

[19] "AMD FireStream," http://en.wikipedia.org/wiki/AMD_FireStream.

[20] S. Bash, D. Carpman, and D. Holl, "Radar pulse compression using nvidia cuda framework," in *HPEC*, 2008.

[21] C. Clemente, M. di Bisceglie, and M. D. Santo, "Processing of synthetic aperture radar data with gpu," Universita degli Studio del Sannio, Tech. Rep., 2009.

[22] http://www.ge-ip.com/products/3429.

[23] "Ge to revolutionize military with nvidia cuda technology," http://benchmarkreviews.com/index.php?option=com_content&task=view&id=8373&Itemid=47.

[24] S. Haykin, "Cognitive radar : A way of the future," *IEEE Signal Processing Magazine*, 2006.

[25] C. T. Capraro, G. T. Capraro, A. D. Maio, A. Farina, and M. C. Wicks, "Cfar knowledge-aided radar detection and it demonstration using measured airborne data," in *Knowledge Based Radar Detection, Tracking and Classification*. John Wiley & Sons, Inc., 2008, pp. 103–128.

[26] "Land use and land cover (LULC)," http://eros.usgs.gov/#/Find_Data/Products_and_Data_Available/LULC.

[27] I. Koukos and I. Moraitakis, "Target matched illumination driven by adaptive software radar algorithms," in *RTO-MP-SET-136*, 2009.