EMBEDDED RADAR SIGNAL PROCESSING

Lukáš Maršík
Doctoral Degree Programme (1), FIT BUT
E-mail: xmarsi03@stud.fit.vutbr.cz

Supervised by: Pavel Zemčík
E-mail: zemcik@fit.vutbr.cz

Abstract: This contribution describes radar signal processing using embedded system as a computational unit. The idea presented in this paper is that the basic algorithms of radar signal processing are simple enough to be executed in real time through the inexpensive devices with low power consumption and compact size. Such solution can be utilized in many mobile applications including the multi-modal fusion of radar data with video. This paper describes the overall approach and the achieved results.

Keywords: embedded system, signal processing, radar

1 INTRODUCTION

Nowadays, the ground radar systems are mostly used for controlling airspace or making weather images [2]. These systems consist of large antenna, a lot of the electronic equipment and very powerful computational unit [1]. Smaller versions of these systems are often carried on the board of planes but still they are quite complex devices. Much simpler versions of the systems mentioned above but still using the same basic principles are small compact devices for measuring speed of vehicles, determination distance of a robot from the obstacle, or even a movement detection for opening doors or turning lights on. In these applications, the compact all-in-one solution (the radar module) can be used. Then real time processing of a radar signal can also be much simpler. For the above mentioned simple applications, it is possible and reasonable to have small devices with low power consumption that perform real time processing algorithms. Typical todays embedded *ARM* based processor solutions have enough computational performance and their electric input is also very low. Moreover, the dimensions of processor boards are very compact and they can be easily integrated into very small cases. Thats is why it is good to transfer radar signal processing algorithms to the embedded system.

The recent development in the radar signal processing presents new ideas in frequency modulated radars [3], and in fusion of the Doppler radar and video information for automated traffic surveillance [4].

2 EMBEDDED VERSION OF PROCESSING ALGORITHM

Fundamentals of radar signal processing algorithms is finding significant frequencies in accordance with the speed of the object. Hence, the most important part of the algorithm is the frequency analysis. The result is usually represented by the complex frequency spectrum (including its power), and computed using the short-term discrete Fourier transform. In the computer world, a more efficient variant known as the *FFT* (the fast Fourier transform) is purely used. Although this improved method has an upper bound of computational complexity $O(N \log N)$ (instead of $O(N^2)$), it still consists of many multiply and add operations. Because of the *FFT* is the hearth of the whole signal processing algorithm, and usually it takes the greatest proportion of overall computation time, it is necessary to perform these operations as fast as possible.

2.1 DOPPLER EFFECT

The basic principle of functionality of the Doppler radar is indeed the Doppler effect [5]. The transmitted signal has to cover the way to the object and back. During the reflection from the surface of the object, the frequency of signal can be modified proportionately to the speed of this object (the double Doppler effect occurs). The frequency deviation is called the Doppler shift and can be computed as

$$f_d = 2v\frac{f_t}{c} \tag{1}$$

where f_t is a transmitting frequency, c is the speed of light, and v is a speed of the object. Then it is possible to determine the object speed on the basis of knowledge of the Doppler shift.

2.2 TIME-FREQUENCY ANALYSIS

The processed signal is infinitely long, so it is necessary to divide it into the short segments (e.g. 256 samples) and then the *FFT* is subsequently applied. The part of the analyzed signal is typically selected by the window (most often the Hamming window) with specific frequency properties. It can be moved over the whole width of the *FFT* or less (the common overlay is 50%). The quantity of samples per second is usually greater than ten thousand, so the number of executed algorithm iterations is large.

2.3 FIXED POINT ARITHMETIC

One of the criterion mentioned above has been the fast execution of arithmetic operations. However, the floating point operations are usually several times slower than the fixed ones. This effect is amplified in case of using the processor without the floating point unit. Then the floating point operations have to be emulated through the fixed point instructions. That is why it is good to keep the critical parts of the algorithm directly in the fixed point arithmetic [6].

The number expressed in the fixed point arithmetic has constant bit width for the integral and fractional part (the exact ratio depends on an application). In other words, the number cannot be indefinitely big or small, as in case of the floating point format. Let us see the example of an unsigned 16-bit number with the 12-bit integral part:

$$n = b_{11}b_{10}b_9b_8b_7b_6b_5b_4b_3b_2b_1b_0.b_{-1}b_{-2}b_{-3}b_{-4}, \quad n \in \{0, 0.0625, \ldots, 4095.9375\}.$$

However, the drawback of using the fixed point arithmetic is the limited accuracy of computation.

3 SYSTEM ARCHITECTURE AND IMPLEMENTATION

The designed system is possible to divide into several disjunct parts where some of them are platform dependent and others are independent. So the system architecture logically consists of three parts. Each of them is running in the separate thread that is implemented using the *Boost::Thread* [7] cross-platform library. Let us mention, that the use of an effective object language is assumed. In our case it is the *C++*. The block scheme of the designed software architecture can be seen on Figure 1 bellow.

The input block takes care of signal input, and separate out the process block from the platform dependent reading of data samples. For the purpose of the input audio buffer reading the *Linux* sound library *ALSA* [8] is used. Then the blocks of collected data are transmitted through the pool structures to the process block.

The process block covers platform independent algorithms for the real time preprocessing and processing of radar signal. As an implementation of the Fourier transform has been chosen the *Kiss FFT*

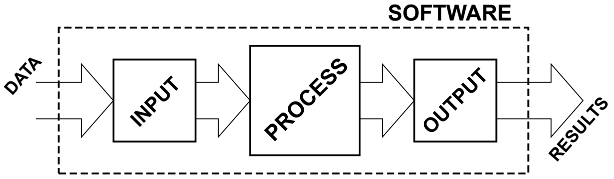

Figure 1: System architecture block diagram

library [9], designed especially for the simple and efficient frequency analysis of real and complex signals. It supports computation using the floating point and also the fixed point arithmetic (with 16-bit and 32-bit basic width). Possible results of performed analysis are reported through another pool structure to the output block.

The output block is typically platform independent. It only transforms received results to the specific message format and writes this reports to the serial console. Then the superior computer can read the messages through the serial link and post-process them. Alternatively the results can be sent over the *TCP/IP*.

4 EMBEDDED IMPLEMENTATION

The presented embedded radar system physically consist of two major components (interconnected through the stereo audio channel). One of them is the radar module, a sensor able to detect objects using microwave signal. Second and equally important part of the system is the embedded processor board that is mainly concentrated on execution operations of the processing algorithm.

4.1 RADAR MODULE

The radar module transmits a narrow beam of microwave energy with the specific frequency. After the moving object reflection, the integrated antenna (situated on the front side of the module) receives appropriate echoes with a slightly different frequency because of the Doppler effect. These two signals are mixed and amplified inside, and the output of this black-box is the signal with the appropriate Doppler frequency. More precisely, the sign of the Doppler shift is indicated by the phase difference of equal frequency components of two corresponding signals on the complex output channel pair (exactly it is $+90^\circ$ and -90°). For the purpose of the embedded signal processing, the output voltage levels of the radar module are adapted using stereo amplifiers and further it is possible to process signal the same way as audio.

Particular module, that has been chosen as an ideal candidate for our embedded system, is the *K-MC1* (see Figure 2) manufactured by the Switzerland company *RFbeam* [10]. Its key features are:

- basic transmitting frequency $24.150\,GHz$, tunable in range $\pm 70\,MHz$
- dispersion of beam 25° in horizontal and 12.5° in vertical direction
- low power consumption mode
- $6\,mm$ thin body

4.2 EMBEDDED PROCESSOR BOARD

The processor board is a platform for real time execution of processing algorithms. These boards usually run one of number of embedded Operating Systems, typically it is the *Linux*, *Windows CE*, etc. Compilation of custom applications is performed with the technique called cross compilation. It

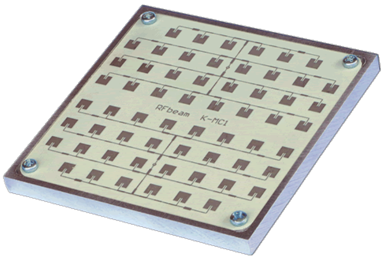

Figure 2: K-MC1 radar module

means that the building of programs is realized on the side of the host PC, and then it is copied to the embedded device. These processor boards have almost the same functionality as a common computer, only the computational performance is usually significantly lower. Moreover, these processors usually do not have the floating point unit, so the drop of performance in case of the floating point operations is large. On the other side, the best advantage of these processor boards is a very low power consumption and compact dimensions.

A good candidate for our embedded system, with all mentioned features, is a solution with the name *Leopard Board* (see Figure 3) [11]. Its key parameters are:

- 32-bit ARM9 based processor *Texas Instruments TMS320DM355*, clocked at $216 MHz$
- *128 MB* DDR RAM and SD card slot for storage memory
- integrated sound card
- Linux Operating System
- power consumption $2W$ (with connected camera module)

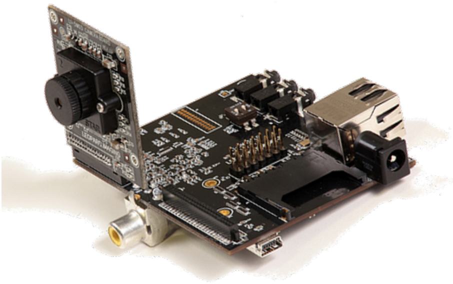

Figure 3: Leopard board with attached camera module

5 RESULTS

The performance and the power consumption of the embedded system has been properly tested. The main algorithm loop (executed in real time) has contained a bit of preprocessing, the *FFT* (16-bit fixed point) applied to the both channels, computation of power of the frequency spectrum (only one channel) and post-processing performing detection of the objects. The exact parameters of the algorithm and the results can be seen in Table 1. Functionality of the complete embedded system and reliability of this solution has been verified performing the real time pedestrian detection application at the fair *Intertraffic 2010* at the stand of the *Camea* company. Finally, the system has run for four days without any crash or getting stuck.

Parameter	Value	Value
Sampling frequency [Hz]	22050	44100
FFT length [samples]	256	512
FFT overlay [samples]	128	0
Processor time [%]	72	80
Power consumption [W]	4.2	4.3

Table 1: Parameters of processing algorithm and appropriate results

6 CONCLUSIONS

It has been demonstrated that the radar signal processing algorithms can be easily executed on the low cost and not so powerful embedded processors. Due to using the fixed point arithmetic for the implementation of the critical parts of the algorithm, it is possible to perform the signal processing in real time. Moreover, these systems has very low consumption and thanks to its compact dimensions it can be mounted on the robots, integrated to the front mask of cars, used as a movement detector, etc. Further, the radar data can be fused with the video data for improved detection of the objects in environment.

ACKNOWLEDGEMENT

This work has been supported by the Artemis JU under the research program R3-COP with the project number 100233.

REFERENCES

[1] Purdy, R. J., Blankenship, P. E., Muehe, Ch. E., Rader, Ch. M., Stern, E., Williamson, R. C.: Radar Signal Processing, Lincoln Laboratory Journal, vol.12, no.2, pp.297-320, 2000

[2] Skolnik,M.: Radar Handbook, Third Edition, New York, NY, USA, McGraw-Hill, 2008, ISBN 00-71485-47-3

[3] Griffiths, H.D.: New ideas in FM radar, Electronics Communication Engineering Journal, vol.2, no.5, pp.185-194, Oct 1990

[4] Roy, A., Gale, N., Hong, L.: Fusion of Doppler Radar and Video Information for Automated Traffic Surveillance, 12th Internation Conference on Information Fusion, Seattle, WA, USA, pp.1989-1996, Jul 2009

[5] Varshney, L.: Radar Principle, 2002

[6] Yates, R.: Fixed-Point Arithmetic: An Introduction, Digital Signal Labs, 2009

[7] Williams, A.: Boost::Thread, Boost.org, 2008

[8] Nagorni, M.: ALSA Programming HOWTO, 2010

[9] Borgerding, M.: Kiss FFT ver. 1.2.8, 2008

[10] RFbeam Microwave GmbH, St.Gallen: K-MC1 Radar Transciever Datasheet rev. 1.3, 2010

[11] DesignSomething.org: LeopardBoard Hardware Guide rev. 1.1, Texas Instruments, Dallas, 2009