2006 6th International Conference on ITS Telecommunications Proceedings

A Practical FMCW Radar Signal Processing Method and Its System Implementation

Wu Chang Li Huan Li Yubai
DSP lab of UESTC China

***Abstract*- Frequency Modulated Continuous Wave (FMCW) radar system measures objects and acquires information by changing the sending signal frequency along with the time and by measuring the receiving signal's frequency with respect to the sending signals. This paper introduces a practical FMCW radar signal processing method and system. This system is mainly used for eliminating noise interference from the sampled signal, checking out the objective signal, and sending the data as the processing results to a computer through the PCI Interfaces. The results are finally processed by the computer and displayed on its screen. This paper analyzes the signal processing algorithm, and applies it to the system implementation. This processing system uses the universal DSP chip as the core device, thus making it easier to use and transplant. This system can be applied to many fields (mini near-distance radar, aircraft carried radar, etc) as long as the front antenna and transceiver system are well organized. It can also be applied to the vehicle carried radar for distance-measuring.**

I. INTRODUCTION

The linear modulation radar deals with continuous waves. By changing the frequencies of sending signals along with the time and by measuring the frequency difference of the receiving signals in regard to the sending signals, linear modulation radar can measure the object's distance. It can also measure the object's radial speed using the Doppler Effect.

The radar transmits linear modulated continuous wave signals, which are positive modulated and negative modulated alternatively. Figure 1(b) shows the linear FM of sending signals and the transformation of receiving signals dispersed from the moving object. In this figure, thin real line represents the sending signals and thick real line the receiving signals. The thick dash line represents the receiving signals when using the Doppler Effect. The frequency difference of the wave crests from the two curves is the dispersed signals' Doppler frequency shift that added by the moving object. The time span between two continuous wave crests is the signal round-trip time. The frequency difference of sending and receiving signals is shown in figure 1(c), where real line represents that the object is fixed, and dash line--the object is moving.

Assume the distance to the object is R_0, then the time delay is 2 R_0/C. Assume the frequency difference between sending signals and receiving signals is F_b , when Doppler frequency is f_d, the positive modulation frequency difference F_{b+} is shown as (1)and negative modulation frequency difference F_{b-} is shown as (2), The object distance corresponds to frequency difference F_b is shown as (3).

$F_{b+}=F_b - f_d$ (1) $F_{b-}=F_b+f_d$ (2) $R_0=F_bC/2\mu$ (3)

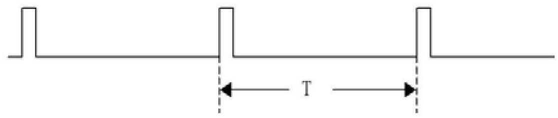

Figure 1 (a) Synchronized Pulse

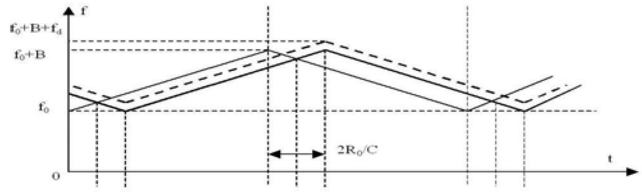

Figure1 (b)frequency as a function of signal time

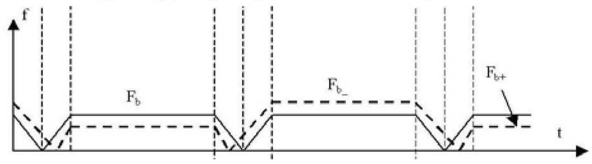

Figure 1 (C)the detector output as a function of time

Where $C=3\times10^8$ (m/s) is the constant transmitting speed of electromagnetic wave, μ is modulation coefficient. μ has a relation with modulation bandwidth B and modulation period T as $\mu=B/T$.

The object's Doppler Effect may lead to the distance measurement error. Based on (1) and (2), and as the Doppler Effect of a fixed object is 0,(F_{b+} - F_{b-}) can then eliminate a fixed object. Through this principle we can eliminate fixed objects. In this system, we eliminate the fixed object in this way.

II. THE SIGNAL PROCESSING METHOD

Aimed at the characteristics of FMCW radar signal, we have designed a new practical signal processing method to detect the object signal. It shows great performance and applicability in validations. This method includes four main aspects as follows: 1) digitize signal, including coherent detection, A/D. 2) frequency difference signal abstraction, including FFT, modular accumulation. 3) pair elimination of fixed objects 4) Doppler frequency distance measuring.

The coherent detection and A/D transformation are the front steps of digital signal processing. They are aimed at converting the analog signals received from the radar to processible digital signals. The intermediate frequency signals are converted to video signals through coherent detection (zero intermediate frequency processing), and the spectrum changed to single sideband spectrum. After A/D sampling, the analog video signals are converted to digital signals, then processed with the digital processing procedure discussed later.

A. FFT (Fast Fourier Transform)

DFT (Discrete Fourier Transform) is a powerful mathematic tool to analyze the spectrum of digital signals. As a fast implemented algorithm of DFT, FFT makes DFT practical in implementation.

We usually focus on the resolution variation that FFT brings in a practical application, because the variation of resolution directly affects the accuracy of signal processing and the real resolution capability of the system.

Assume in a time interval τ, there are m samples to be processed. Using N-points FFT, we need at least $(1/\tau)m\log2N$ times complex number adding per second. That is, while the number of sampling points grows, the complexity of computing being added. When points are doubled, the speed of computing must increase $(1/\tau)m$ times per second.

In the processing procedure discussed here, we use FFT to estimate the frequency F_b of frequency difference signal. If the length of FFT is N, then the frequency resolution is $\Delta f=f_s/N$, and the corresponding distance resolution is $\Delta R=C\Delta f/2\mu$.

It is clear that when N grows, the distance resolution grows, with the amount of computing grows too. Lots of factors must be considered to select a proper N in real applications, includes the accuracy requirement of resolution, the speed of signal processing, the storage capacity, the real-time requests, etc.

In our projects, we assign N to be 4096. Considering points of this function and our requirements, before we do FFT computing, we first process the sampled data as follows:

There are 14 efficient bits set after A/D sampling in application. Thus, whenever read out a datum, we add its efficient bits to 20 by left-shifting it a 6 bits length. This proves the accuracy of data later to be processed, and avoids overflow.

To satisfy the request that the input datum be stored alternatively by its real part and imaginary part, we shall follow the method of storing one genuine datum, then storing another by adding a zero. That is, the genuine datum is the real part, while zero imaginary part. With this process, input data become the primary data for FFT calculation.

B. Modular Accumulation

The purpose of modular accumulation processing is to increase the signal to make it easier to the threshold comparison. The received signal is the addition of the signal and the noise, and the signal is comparably weaker. However, since the amplitude of the signal is nearly immutable, and the signal in real world is close to the gauss white noise (the amplitude of which is random), we can do some accumulations in order to increase signal. The result of FFT, $S(k)=S_R(k)+jS_I(k)$, is a complex number, so we first do modular arithmetic then do the compared-filtering. When designing the algorithm, accuracy and the amount of computing should both be considered. A quite low accuracy adds big noise to the system, thus affects the judged results; a big amount of computing brings too many overloads to the system.

So in practice, the approximate calculation is the best way when we intend to make full use of specific digital processing units (DSP, etc) and existed modules. In the algorithm designed here, we use the following modular arithmetic:

$$|S(k)|\approx Max\{|S_R(k),S_I(k)|\}+(3/8)Min\{|S_R(k),S_I(k)|\} \qquad (4)$$

The above algorithm is simple and light cost; it is one of the realistic choices: Only one addition, one multiplication, and one comparison are needed in a modular process. Let:

$$x=Min\{|S_R(k)|,|S_I(k)|\}/Max\{|S_R(k)|,|S_I(k)|\} \qquad (5)$$

make $C=1+xi$, then the actual expression of its modular arithmetic is represented as (6) , the approximate resolution expression is (7)

$$y=|C|=\sqrt{1+x^2} \qquad (6) \qquad\qquad \bar{y}=1+(3/8)x \qquad (7).$$

It is easy to obtain: $|\Delta y/y|_{max}=6.16\%$. The average lost of this algorithm is 0.07dB.

Then we do accumulation of the modular results. The received signal is the addition of the signal and the noise. The signal has very low power. However, several times of accumulation (we here take 3 times) could reduce the noise's randomization, making it smoother; yet hardly could change the signal, making the signal amplified and noise restrained.

We allocate three storage areas for the positive/negative modulations. Data after the modular arithmetic are stored alternatively to their respective area. Then we accumulate the data in the three areas stored during the same modulation, and make them to be the output. For one modulation, let the data first stored as y_1, second as y_2, and third as y_3, then the addition average result is:

$$result=(y_1+y_2+y_3)\times(1/3) \qquad (8)$$

In practice, we often use fixed point processing units. Thus it could waste too much time when divided by 3 since it is a float calculation. To make it faster, we use the following approximation to avoid division:

$$1/3\approx1/4+1/16+1/64\approx0.328125 \qquad (9)$$

Note that1/4, 1/16, 1/64are all integer exponential of 1/2, so bit shift methods can be introduced here to improve performance and avoid float calculation. The accuracy is acceptable in most cases.

After the accumulation, the S/N ratio has been improved more to meet the stand of judgment.

C. Judgment

The main purpose of judgment is to detect the signal, and eliminate the noise for the successive spectrum matching. We compare the result got before with the threshold we set, and export its amplitude and frequency index when the result is greater, otherwise export frequency index with its amplitude set to zero.

When designing the judgment method, we have to consider the fact: There are lots of strong reflected waves near end radar object. If judging from the near end of radar, these strong objects would be treated as noise, thus submerge some weaker signals. On the opposite, there are fewer objects and weaker reflected waves from the far end of radar, thus the spectrum features are more close to noise. Since judging from the far end is more accurate and appropriate, we adapt this by judging from the maximum index of the spectrum curves, or the so called backward judging, as shown in figure 2.

The way to select the value of threshold is also shown in figure 2. Assume the signal be detected has an amplitude P_a as its main lobe. We may ignore three points: n+2, n+3, n+4. By adding and averaging the spectrum of 16 points from n+5 to n+20, we get the noise evaluation. If $P_a > \eta \times P_j$, we assume signal existed at point n;

If $P_a \leq \eta \times P_j$, we assume no signal existed at point n. Note that η must be assigned a value according to specific applications, and here we set it to be 3.

If there is signal at point n, we export the signal amplitude of this point.

If there is no signal at point n, we export 0, and hold the value of n unchangeable.

The spectrum features are shown after the judgment, which are brought to the Doppler frequency shift formula to generate the expected physical features such as the object distance, etc.

D. Object Detecting

For all the information after judgment, we have to obtain the spectrum index and the contained number of spectrums for every object.

For any two spectrums each with a none-zero amplitude, if there are less than n zero points between them, we then consider them belonging to the same object. The value of n is selected from certain cases, in our system we assign n=3.

For all the 1024 data got after the threshold judgment, we obtain in each group the spectrum lines belonging to the same object using the above method, calculate their central spectrum, find out the spectrum with the maximum amplitude, and finally store all the information to a 2048-points array.

For example, between point 300 and point 304, we figure out a group of spectrums for one same object, and then the central spectrum of this object should be at point 302, the maximum amplitude is 4×10^6. We store the value 4×10^6 at the 302nd location of the 2048-points array, and the spectrum width of this object 4 at 1326 (302+1024).

In addition, we need spectrum matching to eliminate the interferences of fixed objects when detects moving objects.

Spectrum matching refers to cognize (judge) that the positive or negative modulation reflected wave signals come from the same object. This prepares for the following process of fixed wave elimination and Doppler distance detection error correction. Theoretical analysis proves that the positive and negative modulation reflected wave signals generated by the same object have the same spectrum amplitude and shape. Although the amplitude of the two spectrums may be slightly different due to certain cases in practices (the fluctuation of objects, the antenna scanning, interferences, noise, etc), their spectrum shape must have a maximum similarity. This is why we use the spectrum features to detect whether the positive/negative signals are from the same object.

Just then we have processed the data, written every object's maximum amplitude value at its central spectrum point, and stored the spectrum numbers of the object included in the point whose position is 1024 points far from the central spectrum point. Now it is easier for us to do the spectrum matching.

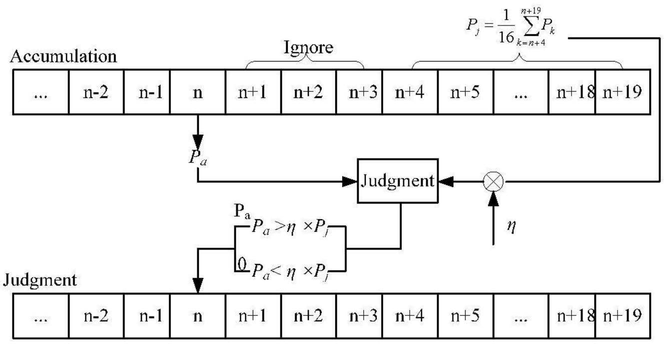

Figure 2 the method of backward judgment

In our algorithm, we consider the spectrums belong to the same object if the positive and negative modulation spectrums meet the following requirements:

1. Assume the difference of the central spectrum index of the positive/negative modulation is m. This m is decided by the maximum object speed the system can handle. If not exceeds m, then consider them belong to the same object. In our system we set m=42.
2. The difference of positive or negative modulation spectrum index is no less than 2, or we consider it as a fixed object.
3. The difference of maximum positive or negative modulation amplitude is no more than 2 times, or we consider it to be different objects.
4. The difference of their spectrum width (spectrum line numbers) is no more than 3.

We take the positive modulation as a benchmark when we do spectrum matching. That is, we find out a spectrum in the positive modulation, and then pick out the matching spectrum in the negative modulation. If a certain spectrum from the negative modulation is considered matching, it is cleared to zero in order to avoid interferences to the next matching.

We find out a spectrum in the positive modulation, set the spectrum index C_P, amplitude A_P, and find out another in the negative modulation with spectrum index C_N and amplitude A_N. Now we can figure out information such as the object distance, direction, speed and the maximum amplitude of the object spectrum:

1. Compute the central spectrum point C (the point with Doppler frequency shift eliminated) through C_P and C_N. Send out C as the object distance information.
2. Acquire the direction information by reading the direction register.
3. Acquire (C- C_N) as object speed information.
4. Take the average of forward/backward modulation amplitude A=(A_P+A_N)/2 as the amplitude information.

The Doppler frequency shift of a moving object may interfere the measuring of the object distance, so we have to revise it.

Because the beat frequencies of a moving object has the following form:

$$\begin{cases} f_{i,up} = f_i + f_d & \textit{positive modulation} \\ f_{i,down} = f_i - f_d & \textit{negative modulation} \end{cases} \tag{10}$$

The beat frequencies introduced by the distance factors can be determined by (11), The corresponding distance is (12)

$f_i=1/2(f_{i,up}+f_{i,down})$ (11) $R_i=(C/2\mu)f_i$ (12)

where C is the speed of light, $\mu=B/T$, B is modulation bandwidth, T is modulation period.

III. System Design

Based on the FMCW signal processing algorithm discussed just then, we use DSP as the core signal processing device in practice. Along with A/D sampler, FIFO, Dual Port RAM, we construct the hardware platform to realize our algorithm. The system design scheme is shown in figure 3:

A/D sampling is a key step before signal processing. In practice, a low pass filter (anti-aliasing filter) must be added in front of the A/D sampler. We must properly select the A/D sampler, with its sampling frequency and accuracy taken into account. In our system, the receiver's bandwidth is 1.5M. Along with the requirements of processor bits, storage size etc, we decide to choose AD9240 from AD plc. We set its sampling clock to 5M/14bit.

$$\textit{Dynamic Range}=20log\,(2^N)=6.02NdB=84.28dB$$

In our implementation, we select 2 chips of FIFO (8k) as the buffer between A/D and DSPI, DSPI and DSPII respectively. The write clock of FIFOI and A/D use the same 5M source clock. The radar signals are positively and negatively modulated alternatively with a period of 4ms. A pulse signal of 2ms will be input to the system. We import this pulse to the 14^{th} bit of FIFOI directly, increasing the data to 15bit. Thus, the most significant bit of the output data will change along with the positive/negative alternation of modulation, and then DSPI can determine whether to use positive or negative modulation by examining the most significant bit of each datum. We use HF (Half Full Flag) of FIFOI to trigger the interruptions of DSP, which brings data exchange between DSP and FIFOI. FIFOII is mainly served as the data buffer between DSPI and DSPII, which makes task allocation available for the 2 DSP chips.

Our system almost includes all the steps of radar signal processing. As there are other constraints such as the real time requirements and the limitation of calculation, here we use 2 DSP chips to divide and handle all the tasks of the system.

DSPI is mainly used for FFT calculation, modular arithmetic, judgment and object recognition, etc.

DSPII is mainly used for spectrum matching, fixed objects paired-elimination directed by user instructions. The way we divide tasks is mainly based on the following points:

1. FFT calculation, modular arithmetic, judgment, object recognition and other relative calculations belong to the basic steps of digital signal processing, with good performance of linking and concatenation, be capable of successive real-time processing, division will decrease the efficiency of processing instead.
2. The spectrum matching and fixed object paired-elimination can only be started after certain amount of data is acquired. The real time requirement is not as strictly as the prior steps.
3. According to the algorithm discussed above, after object recognition, we will store the processed data in a particular format for the convenience of spectrum matching and fixed object paired-elimination. This allows us to take full use of FIFOII, thus saves the on-chip space of the DSP.
4. Unlike DSPI, DSPII not only processes data, but also provides interfaces to PC. Except for EDMA interrupt, it uses another 2 interrupts: one is responsible for executing commands from a PC to take some controls over DSP processing; the other is for receiving pulse signals from the north to gain direction information.
5. To satisfy the real time requirements in practice, we leave some allowance for functional extension, storage and calculation.

In this system, Dual Port RAM is used to connect DSPII and PCI chip to enable communication between DSP and PC. Results are stored and displayed on the PC screen. Here we use the interruption-based mechanism. Once the PC writes values to the highest address of the Dual Port RAM through PCI9054, it will generate an interruption on INTL, left port of the Dual Port RAM. Once DSP receives this interruption, it reads the address, do some changes of the DSP processing according to the written data from PC.

In the whole system design, we have used CPLD to control and convert some basic logics, which is commonly used in engineering projects. Moreover, there may exist unforeseeable conditions when do system design. The use of CPLD makes our system more adaptive and easier to debug.

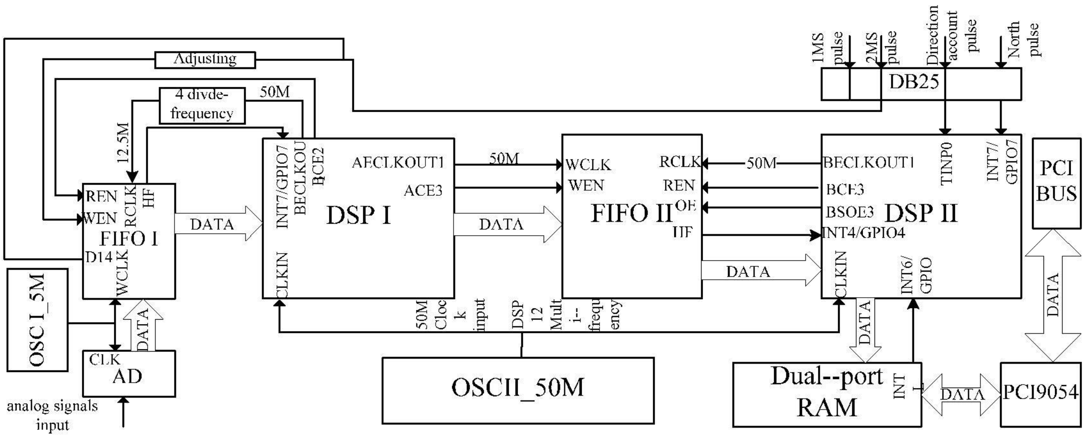

Figure 3 the system infrastructure

IV. CONCLUSION

Linear FM radar has lots of advantages and is broadly applied in many fields.

This paper brings forward practical signal processing methods based on the features of FMCW radar signals, including: FFT, accumulation, backward judgment, etc. It also analyzes how these methods can be applied to engineering projects. Based on all these methods, this paper introduces a new design scheme of FMCW radar signal processing system. Aimed at the features of engineer tasks, it divides the tasks (into two parts), using two DSP chips in the concatenation mode. The paper also gives some solutions to the pertinent problems those may occur during the FMCW radar signal processing.

At last, we adapt a universal DSP- TMS320C6416, one of the fastest processor currently shipped by TI, to implement a real FMCW radar signal processing system. We apply this system to the signal detection of near-distance radars. Testing shows that this system is practical and flexible. It can detect signals with an input S/N of -27dB. PCI interfaces are installed on this system to make it capable to communicate with PC and other devices conveniently.

REFERENCES

[1] Xiang Jingcheng, and Zhang Mingyou, "Radar System", Publishing House of Electronics Industry, 2001.

[2] Stove, A.G., *Linear FMCW radar techniques,* Radar and Signal Processing, IEE Proceedings, Part F. Volume 139, Issue 5, Oct. 1992 Page(s):343 - 350

[3] Peng Qicong, Li Yubai, and Guan Qing, "Development and Application of DSP Technology", Higher Education Press, Sep. 2002

[4] Ding Lufei, and Gen Fulu, "Radar Theory", Xi Dian University Press, Jul, 2002.

[5] Xu Xiaoping, Liu Jianxin, Han Yu, and Ding Qingsheng, *Simulation of Digital Signal Processor on FMCW Radar*, Information and Electronic Engineering, Vol. 2, No. 2, Jun, 2004