Stepped-frequency pulse-train radar signal

N. Levanon

Abstract: The stepped-frequency radar signal is revisited. The delay–Doppler response of pulse-trains with different interpulse frequency codings is investigated, including linear, nonlinear, up–down and Costas codings. Different approaches to sidelobe reduction are considered. Both matched filter and stretch processing are discussed.

1 Introduction

The range-resolution attainable by a radar signal is inversely related to the signal bandwidth. Linear FM (chirp), a popular pulse compression radar signal, achieves large bandwidth by linearly changing the carrier frequency during the pulse. The result is narrow instant bandwidth, but wide overall bandwidth. The rate of frequency change can be further slowed down by utilising a coherent train of pulses, with each pulse transmitted at a different frequency. Rihaczek ([1], Chap. 8.6) called it 'interpulse frequency shift coding', while in Skolnik ([2], Chap. 10.7) the phrase 'time-frequency-coded waveforms' is used. Monotonically increasing (or decreasing) frequency is usually implemented, but this is not mandatory.

A frequency-stepped coherent pulse-train can be processed using a matched filter. In that case the resulting delay-Doppler response is the ambiguity function of the batch of pulses. Another processing approach is the 'stretch' processor ([3], Chaps. 4.5 and 5). In stretch processing the received signal, reflected from a target, is mixed with a replica of the transmitted signal, resulting in a down-converted baseband signal. If the original signal was frequency-stepped linearly, then the frequency of the baseband signal will be linearly related to the delay of the target return. Fourier analysis is typically used to resolve the delay into range bins.

Step-frequency with stretch processing is especially attractive in radar sensors for short range, like automotive radar [4, 5], for two reasons: (i) The simplicity of the processor, hence its low cost; and (ii) Since the typical delay could be shorter than the pulse duration, and since the receiver is effectively turned off during transmission, not all of the reflected signal is available to the receiver. Missing part of the signal rules out matched processing.

The delay–Doppler response of linear-FM and its stepped-frequency version suffers from range–Doppler coupling and from large delay-sidelobes. The first problem can be mitigated by using two bursts, with up and down frequency slopes. We will also investigate another approach, in which the interpulse frequency coding follows Costas coding scheme [6]. We will show that Costas coding does not rule out stretch processing, but it suffers from relatively high sidelobes.

Delay sidelobes can be reduced through frequency weighting. Frequency weighting can be achieved either by longer dwells in the middle frequencies, namely stepping the frequencies in a nonuniform fashion (nonlinear-FM), or by increasing the amplitude of the pulses corresponding to the middle frequencies. The amount of sidelobe reduction attainable by the first approach (NLFM) is limited. Furthermore, it rules out stretch processing. The second approach (amplitude modulation) is difficult to implement in the transmitter, and if performed only at the receiver, results in SNR loss due to mismatched filtering.

We will demonstrate good sidelobe reduction, while maintaining a fixed frequency-step and constant-amplitude pulses. Frequency weighting will be obtained by controlling the number of pulses at each frequency. This approach can yield any desired weighting law, hence any desired sidelobe reduction. At the same time, due to the single-size frequency step, it allows stretch processing.

2 Ambiguity function of a stepped-frequency pulse-train

The stepped-frequency pulse-train is described in Fig. 1. The burst consists of M pulses at frequencies f_m and amplitudes B_m, $m = 1, 2, \ldots M$. The total coherence time is MT_r. T_r is the pulse repetition interval and t_p is the pulse duration. In its basic linear form the frequencies are equally spaced, i.e. $f_m - f_{m-1} = \Delta f$ and the pulses are of a fixed amplitude, i.e. $B_m = B$.

The zero-Doppler cut of the ambiguity function (namely the magnitude of the autocorrelation function) for delay $|\tau| < t_p$ depends on the number of pulses M, the product $t_p \Delta f$, the amplitudes B_m and the order of the frequencies. In the fixed-amplitude case, the order of frequencies does not

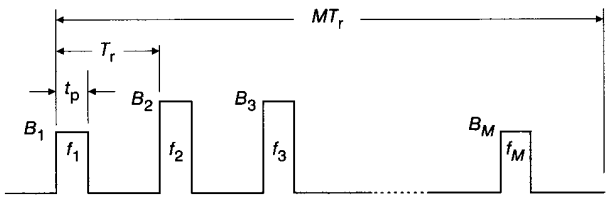

Fig. 1 *Stepped-frequency pulse-train*

IEE Proceedings online no. 20020432

DOI: 10.1049/ip-rsn:20020432

Paper first received 14 October 2001 and in revised form 3 April 2002

The author is with the Department of Electrical Engineering, Tel Aviv University, P.O. Box 39040, Tel Aviv, 69978, Israel

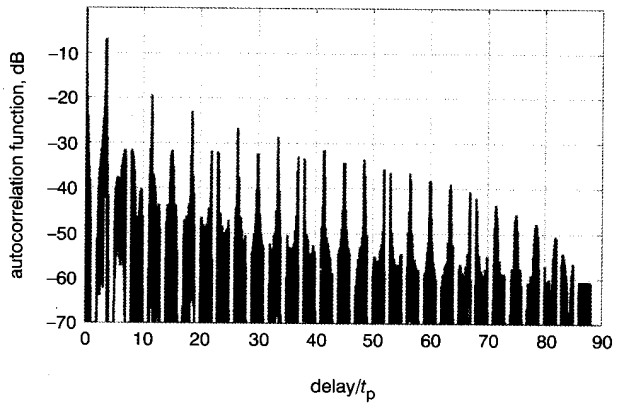

Fig. 2 *Autocorrelation function of 30 pulses with linear frequency steps*

$T_r = 3t_p$, $t_p\Delta f = 0.4$, no amplitude weighting
Frequency order: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

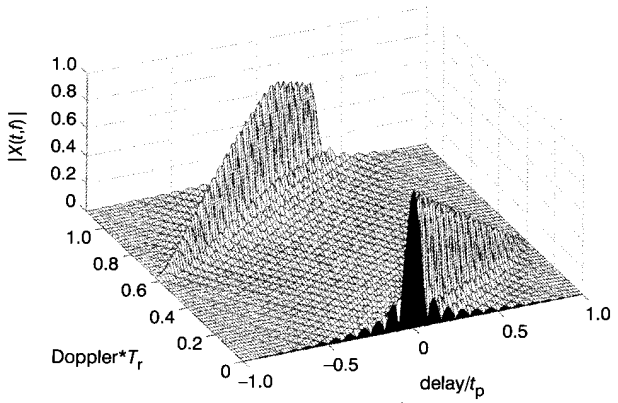

Fig. 4 *Partial ambiguity function of 30 pulses with linear frequency steps*

$T_r = 3t_p$, $t_p\Delta f = 0.4$, no amplitude weighting
Frequency order: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

affect the zero–Doppler cut over $|\tau| < t_p$, but strongly affects the response at higher Doppler shifts.

The next few Figures were derived using the following parameters: $T_r = 3t_p$, $M = 30$ and $t_p\Delta f = 0.4$. Figs. 2–7 pertain to linearly decreasing frequencies. Fig. 2 displays the autocorrelation function in dB, for all positive delays, and Fig. 3 zooms on the delay range $0 \le \tau \le t_p$. Note in Fig. 2 the relatively high recurrent lobes at multiples of the pulse repetition interval. Note in Fig. 3 the first null at

$$\tau_{1\text{st null}} = \frac{1}{\text{bandwidth}} = \frac{1}{M\Delta f} \Rightarrow \frac{\tau_{1\text{st null}}}{t_p} = \frac{1}{Mt_p\Delta f} = 0.083 \quad (1)$$

Note also peak sidelobe of approximately −14 dB, typical of LFM. Two quadrants of the ambiguity function are plotted in Fig. 4, exhibiting the diagonal ridge, typical for LFM. Note the slope of the ridge

$$\frac{f_{\text{ridge}}}{\tau} = \frac{\Delta f}{T_r} \quad (2)$$

With the normalised axes of Fig. 4 the slope of the ridge is given by

$$\frac{f_{\text{ridge}}T_r}{\tau/t_p} = \frac{\Delta f}{T_r}T_r t_p = t_p\Delta f \quad (3)$$

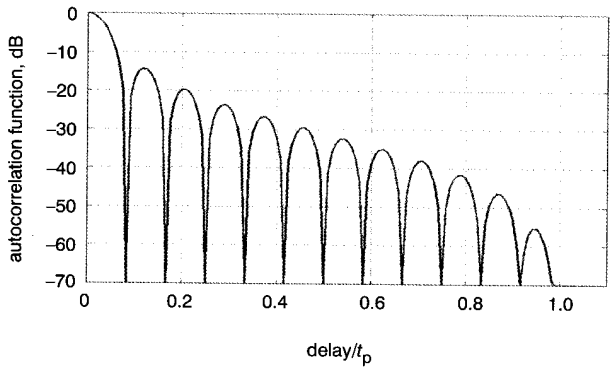

Fig. 3 *Partial autocorrelation function of 30 pulses with linear frequency steps*

$T_r = 3t_p$, $t_p\Delta f = 0.4$, no amplitude weighting
Frequency order: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Equation (3) implies that for a given Doppler shift f_D the delay error, normalised with respect to the pulse width, will be

$$\frac{\tau_{\text{error}}}{t_p} = \frac{T_r}{t_p\Delta f}f_D \quad (4)$$

Fig. 4 contains one more of the many parallel diagonal ridges, spaced in Doppler by $1/T_r$.

Delay sidelobes reduction can be achieved through frequency weighting. Since each pulse represents a different frequency, weighting could be implemented by controlling pulse amplitudes. To prevent SNR loss due to mismatch, the pulse amplitudes, dictated by the weight window, have to be split (square root) between the transmitted and reference signals. Applying a Chebyshev weight window with −50 dB ripple (Fig. 5), yields the partial autocorrelation in Fig. 6. Comparing Figs. 3 and 6 demonstrates both delay sidelobe reduction and main-lobe broadening.

Comparing the new ambiguity function (Fig. 7) with the previous one (Fig. 4) we note a reduction of both delay and Doppler sidelobes. Doppler sidelobes can be reduced by adding a weighting window in the slow-time, namely from pulse to pulse. Owing to the linear frequency stepping, applying a weight window in the slow-time achieves both slow-time weighting and frequency weighting. The real envelope of the transmitted pulses was shown in Fig. 5.

According to (3) higher values of $t_p\Delta f$ would have yielded higher pulse compression ratio, so why have we used $t_p\Delta f = 0.4$? The answer is found in Fig. 8, which summarises the ridge structure. As Fig. 8 and (3) show, increasing $t_p\Delta f$ will increase the slope of the ridges. For $t_p\Delta f \ge 0.5$ there will be Doppler shift values at which the corresponding Doppler cut of the ambiguity function will exhibit two peaks. To avoid such a problem requires $t_p\Delta f < 0.5$. Because the ridge has some width, the upper limit on $t_p\Delta f$ should be somewhat lower, e.g. 0.4. Choosing $t_p\Delta f = 1$ will cause the upper and lower ridges in Fig. 8, to hit the delay axis exactly at $\tau/t_p = \pm 1$. This will raise two large autocorrelation sidelobes near $\tau/t_p = \pm 1$.

If delay–Doppler coupling is undesirable, using different frequency ordering can eliminate it. Costas [6] ordering is a possible alternative [7]. Figs. 9–11 pertain to ordering the 30 frequencies in a Costas sequence {3 9 27 19 26 16 17 20 29 25 13 8 24 10 30 28 22 4 12 5 15 14 11 2 6 18 23 7 21 1}. The order of frequencies does not affect the autocorrelation within $|\tau| < t_p$. Hence, with the same

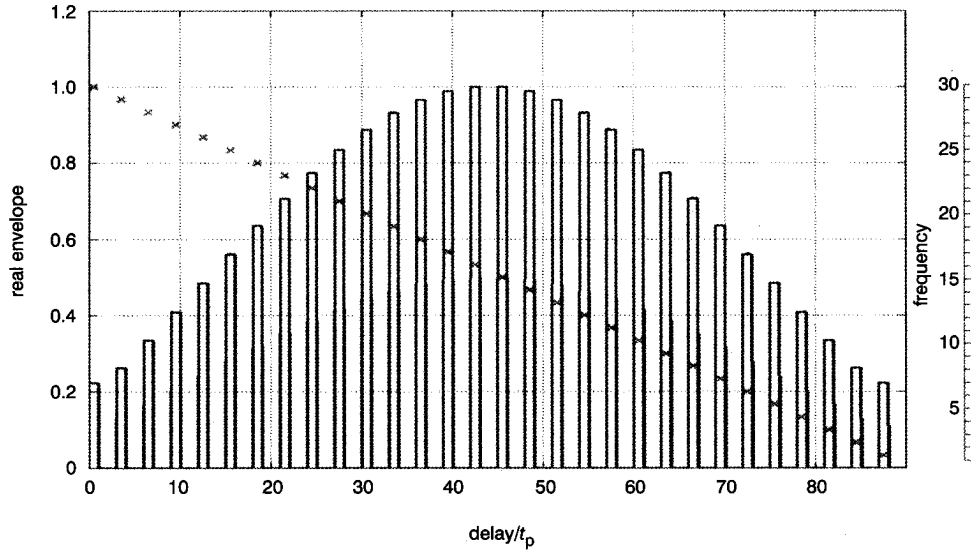

Fig. 5 *Real envelope and frequency (×) of 30 pulses with Chebyshev weighting, linear order*
$T_r = 3t_p$, $t_p\Delta f = 0.4$, frequency weight power = 0.5, Chebyshev weighting (ripple 50 dB)
Frequency order: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

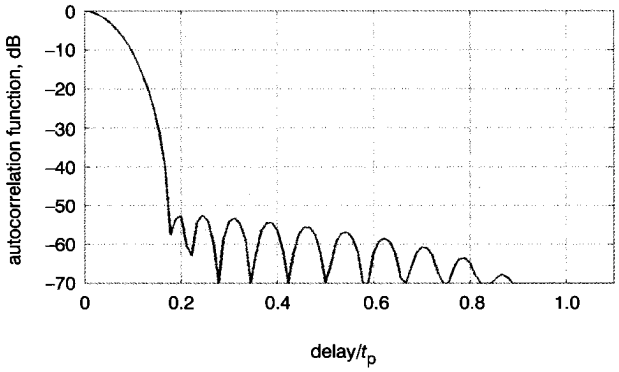

Fig. 6 *Partial autocorrelation function of 30 pulses with linear frequency steps, Chebyshev weighting*
$T_r = 3t_p$, $t_p\Delta f = 0.4$, frequency weight power = 0.5, Chebyshev weighting (ripple 50 dB)
Frequency order: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

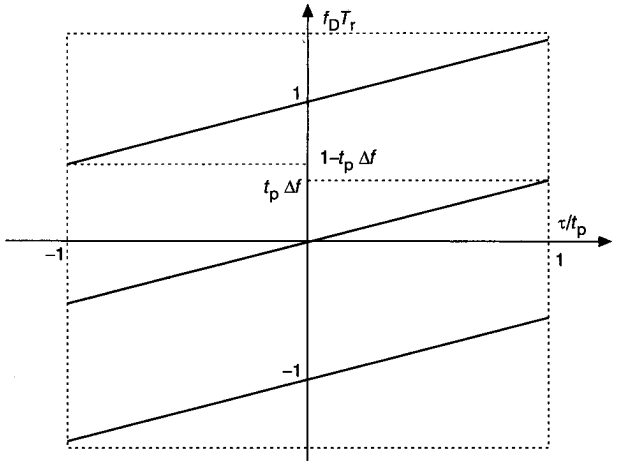

Fig. 8 *Ridges in the ambiguity function of a linear stepped-frequency pulse-train*

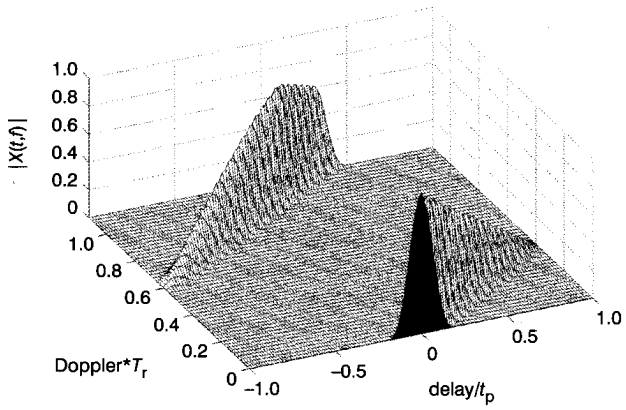

Fig. 7 *Partial ambiguity function of 30 pulses with linear frequency steps, Chebyshev weighting*
$T_r = 3t_p$, $t_p\Delta f = 0.4$, frequency weight power = 0.5, Chebyshev weighting (ripple 50 dB)
Frequency order: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Chebyshev frequency weighting, the Costas ordered signal would exhibit the same autocorrelation as plotted in Fig. 6 for a linear order of frequencies. However, to achieve the same frequency weighting, the pulse amplitudes should be as shown in Fig. 9. The Costas order does affect the ambiguity function at non-zero Doppler, as seen in Fig. 10. A thumbtack shape replaced the ridge shape. To maintain constant volume under the ambiguity function squared, the volume removed from the ridge is now found in the higher sidelobes plateau. When the frequencies are ordered according to Costas, frequency weighting does not affect a worthwhile slow-time weight window (see Fig. 9). Hence, there is no Doppler sidelobe reduction along the zero-delay cut. Another property of a Costas order is some reduction of the autocorrelation recurrent lobes, as seen by comparing Fig. 11 with Fig. 2.

The relatively high sidelobes plateau, observed in Fig. 10, points to a severe problem from neighbouring targets with different Doppler. Sidelobes from such targets may be too high, and could mask targets of interest.

Another approach to mitigate the range–Doppler coupling of LFM is to use separate coherent batches, one with up-

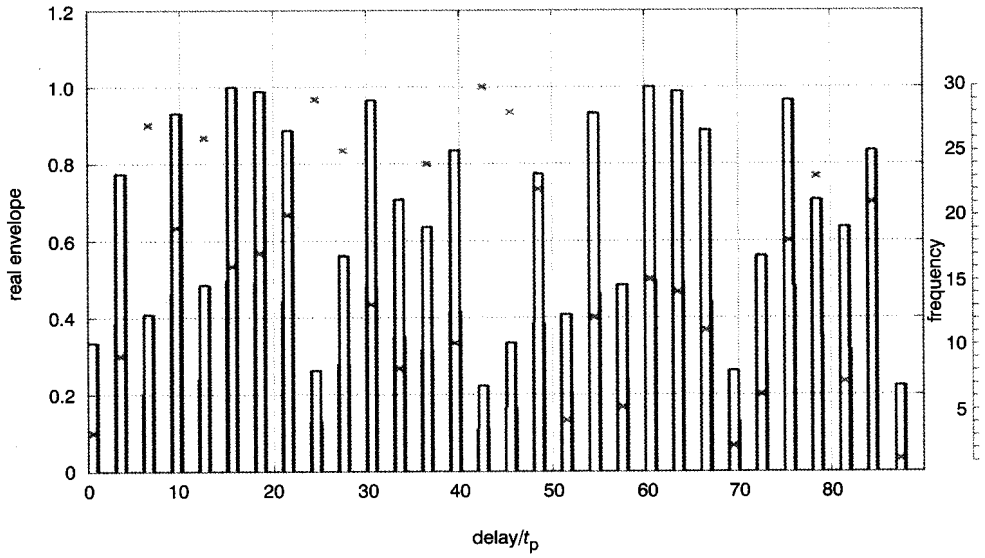

Fig. 9 *30 pulses with Costas frequency steps and Chebyshev weighting: real envelope and frequency (×)*
$T_r = 3t_p$, $t_p\Delta f = 0.4$, frequency weight power = 0.5, Chebyshev weighting (ripple 50 dB)
Frequency order: 3 9 27 19 26 16 17 20 29 25 13 8 24 10 30 28 22 4 12 5 15 14 11 2 6 18 23 7 21 1

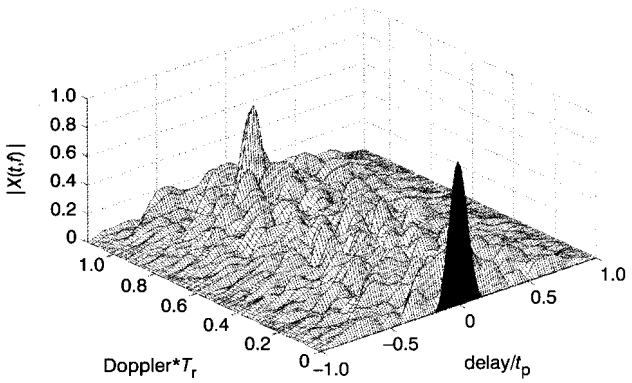

Fig. 10 *30 pulses with Costas frequency steps and Chebyshev weighting: partial ambiguity function*
$T_r = 3t_p$, $t_p\Delta f = 0.4$, frequency weight power = 0.5, Chebyshev weighting (ripple 50 dB)
Frequency order: 3 9 27 19 26 16 17 20 29 25 13 8 24 10 30 28 22 4 12 5 15 14 11 2 6 18 23 7 21 1

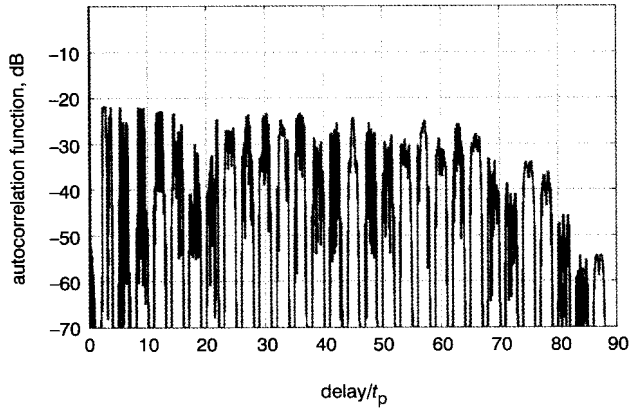

Fig. 11 *Autocorrelation function of 30 pulses with Costas frequency steps and Chebyshev weighting*
$T_r = 3t_p$, $t_p\Delta f = 0.4$, frequency weight power = 0.5, Chebyshev weighting (ripple 50 dB)
Frequency order: 3 9 27 19 26 16 17 20 29 25 13 8 24 10 30 28 22 4 12 5 15 14 11 2 6 18 23 7 21 1

slope and the other with down-slope. Post detection algebraic manipulation of the two results can separate delay from Doppler. We can expect a similar result from coherent matched processing of the two batches together. The real envelope of a weighted, up–down frequency-stepped, coherent pulse-train, is given in Fig. 12, together with the frequency order. The corresponding ambiguity function is given in Fig. 13. Indeed, a large single lobe at the origin plus two lower diagonal ridges replaced the large single diagonal ridge of LFM (Fig. 7).

So far sidelobe reduction has involved pulse amplitude modulation in addition to the pulse frequency shifts. In most practical systems transmitting pulses of different amplitudes is a problem. In such systems the amplitude weighting could be restricted to the receiver. The delay–Doppler response would be identical to the ambiguity function, except for a significant reduction in SNR.

In the single-pulse case, another approach to control the power spectrum of the signal, to reduce range sidelobes, is to change the intrapulse frequency in a nonlinear fashion [2]. In nonlinear FM the spectrum shape is controlled according to the weight window by longer dwell at the mid-frequencies. In the stepped-frequency pulse-train, the nonlinearity could be implemented using two different approaches: (i) nonuniform frequency steps between pulses; and (ii) different number of repetitions of the same frequencies. With the first approach, we were not able to lower the sidelobes below −35 dB. The second approach allowed us to reach practically any desired sidelobe level. The second approach for creating NLFM is demonstrated in Figs. 14 and 15. Fig. 14 shows how a typical nonlinear FM curve using 13 equally spaced frequencies can be implemented by controlling the number of pulses at a given frequency, from 1 the pulse at the lowest (and highest) frequency to 18 consecutive pulses at the middle frequency. Fig. 15 compares the 13 elements of a theoretical Chebyshev window (with ripple of −50 dB) with the integer number of pulses at each one of the 13 frequencies. Since the lowest possible number of pulses at a given frequency is one, the value one should represent the lowest weight in the window. Clearly the total number of pulses in this approach must be much larger (120 in the

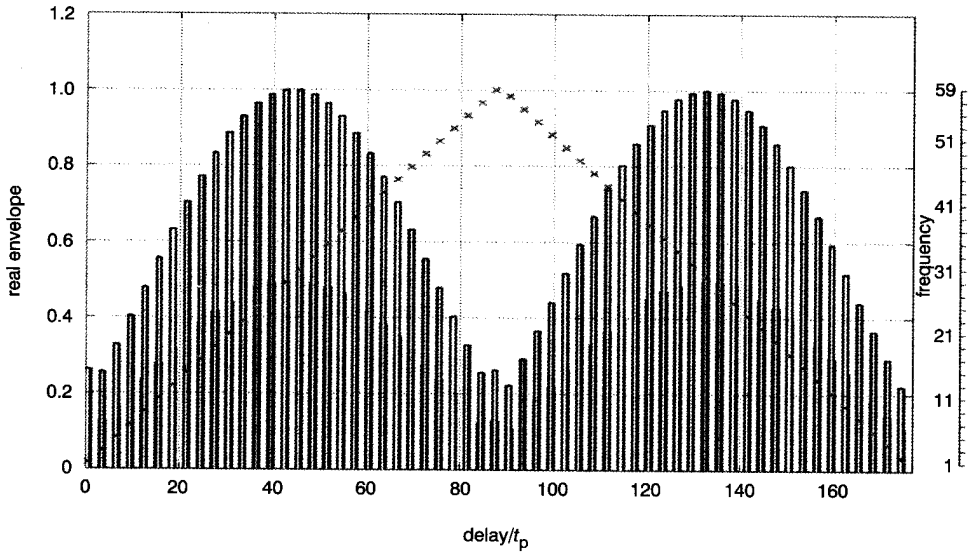

Fig. 12 *59 pulses with up–down linear frequency coding and Chebyshev weighting: real envelope and frequency (×)*
$T_r = 3t_p$, $t_p\Delta f = 0.2$, frequency weight power = 0.5, Chebyshev weighting (ripple = 50 dB)
Frequency order: 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 58 56 54 52 50 48 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16

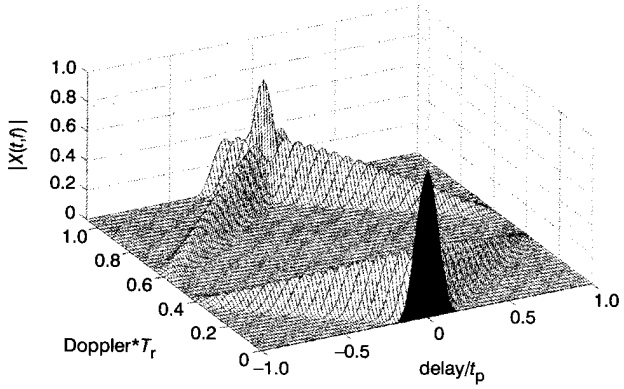

Fig. 13 *59 pulses with up–down linear frequency coding and Chebyshev weighting: partial ambiguity function*
$T_r = 3t_p$, $t_p\Delta f = 0.2$, frequency weight power = 0.5, Chebyshev weighting (ripple = 50 dB)
Frequency order: 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 58 56 54 52 50 48 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16

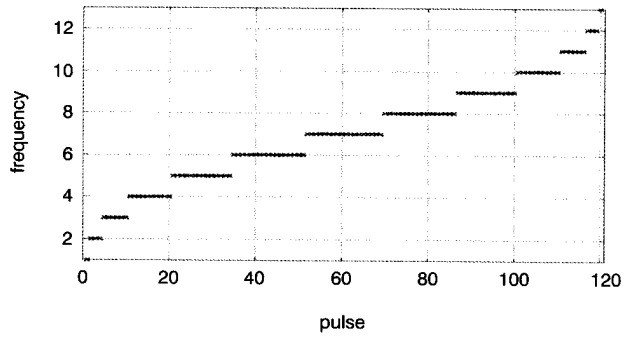

Fig. 14 *Creating nonlinear FM by repetition of frequencies: frequency law*
$M = 120$, number of frequencies = 13, $T_r = 3t_p$, $t_p\Delta f = 0.78$, no amplitude weighting, Chebyshev nonlinear ripple = 50 dB, nonlinear power 1

example) than the number of frequencies (13 in the example) to create the desired weight window.

The performances of the 'NLFM by repetition' approach to sidelobe reduction are presented in Figs. 16–18. Fig. 16 displays the partial autocorrelation function, demonstrating −50 dB sidelobes (in agreement with the weight window's ripple parameter). Fig. 17 displays the partial ambiguity function. The Doppler scale extends beyond the inverse of the pulse interval to show the recurrent Doppler peaks at $f_D = 1/T_r$. Because of the large number of pulses ($M = 120$), the Doppler main-lobe is very narrow $f_{D\ 1st\ null}T_r = 1/120$.

For more details, Fig. 18 zooms with respect to the Doppler dimension. We note that the slope of the ridge is much lower than the slope without repetitions. This should be expected. If the typical number of repetitions at the mid-frequencies is $\bar{N}_{rep} \approx 16$ (see Figs. 14 and 15), then the expected slope of the ridge should be

$$\frac{f_{ridge}}{\tau} = \frac{\Delta f}{T_r \bar{N}_{rep}} \tag{5}$$

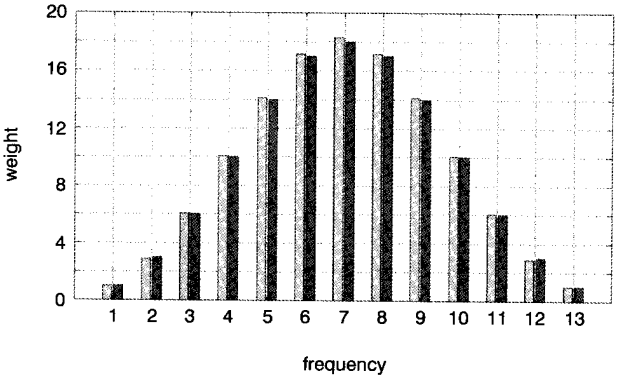

Fig. 15 *Creating nonlinear FM by repetition of frequencies: histogram*
Integer number of repetitions (right bars) and theoretical Chebyshev weight (left bars)
$M = 120$, number of frequencies = 13, $T_r = 3t_p$, $t_p\Delta f = 0.78$, no amplitude weighting, Chebyshev nonlinear ripple = 50 dB, nonlinear power 1

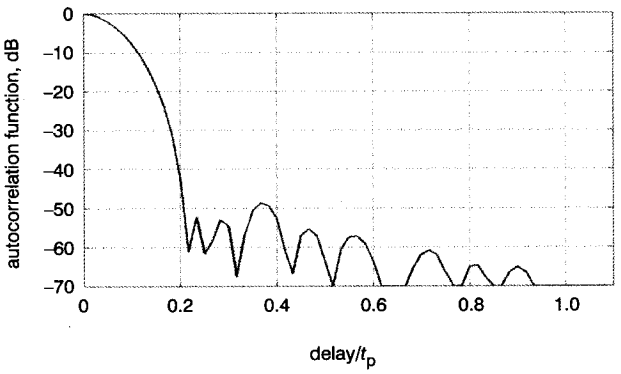

Fig. 16 *Partial autocorrelation function of 120 pulses with the frequency steps shown in Fig. 14*

$M=120$, number of frequencies = 13, $T_r=3t_p$, $t_p\Delta f=0.78$, no amplitude weighting, Chebyshev nonlinear ripple = 50 dB, nonlinear power 1

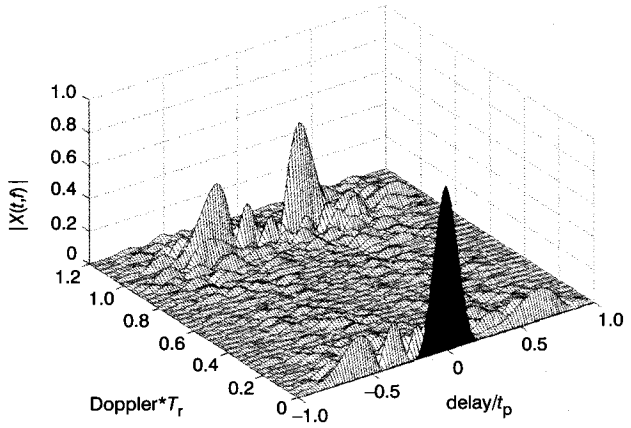

Fig. 17 *Partial ambiguity function of the NLFM signal defined in Fig. 14*

$M=120$, number of frequencies = 13, $T_r=3t_p$, $t_p\Delta f=0.78$, no amplitude weighting, Chebyshev nonlinear ripple = 50 dB, nonlinear power 1

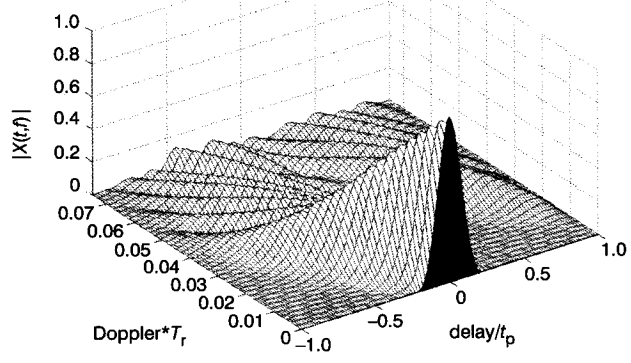

Fig. 18 *Doppler zoom of data in Fig. 17*

$M=120$, number of frequencies = 13, $T_r=3t_p$, $t_p\Delta f=0.78$, no amplitude weighting, Chebyshev nonlinear ripple = 50 dB, nonlinear power 1

With the normalised axes of Fig. 18 the slope of the ridge is given by

$$\frac{f_{\text{ridge}}T_r}{\tau/t_p}=\frac{\Delta f}{T_r\bar{N}_{\text{rep}}}T_rt_p=\frac{t_p\Delta f}{\bar{N}_{\text{rep}}}=\frac{0.78}{16}\approx 0.05 \qquad (6)$$

Indeed, note in Fig. 18 that at $\tau/t_p=-1$, the ridge hits the normalised Doppler value of 0.05.

3 Stretch processing of stepped-frequency pulse-train

Because of the large overall bandwidth of a stepped-frequency pulse-train, matched processing will require a relatively high sampling rate, which may be impractical in many applications. A common alternative is to perform stretch processing [3]. A generic block diagram of stretch processing, suitable for automotive radar application [4, 5], is described in Figs. 19 and 20. Fig. 19 presents the transmitted $x(t)$ and reference $s(t)$ signals. The transmitted signal pulses (top) exhibit fix amplitude A. The duty cycle t_p/T may be as high as $\frac{1}{2}$. The frequency coding f_m, $m=1,\ldots,M$ may follow one of the schemes discussed in the previous section. The reference signal (bottom) time-complements the transmitted one, indicating that the receiver is turned off during transmission. The variable-pulse amplitudes represent the option to add a weight window at the receiver (the weights are inserted prior to the IFFT).

A conceptual radar is described in Fig. 20. The voltage staircase feeding the VCO will generate the frequency coding. The stair period is T_r and the stair height will cause a frequency step of Δf. The switch on the left generates the two signals $x(t)$ and $s(t)$. Synchronously, the switch on the right connects the antenna to the VCO or to the mixer. Ignoring attenuation, the signal received

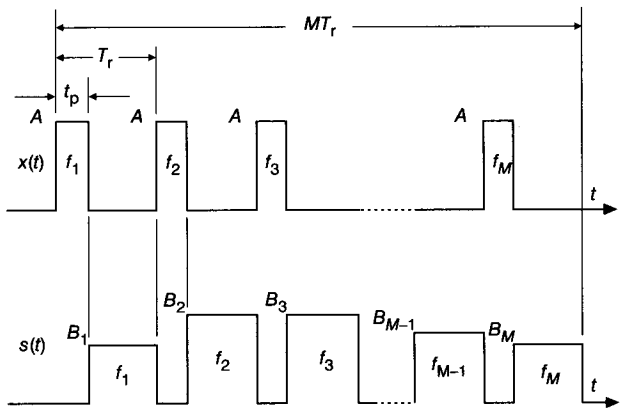

Fig. 19 *Stepped-frequency pulse-train, transmitted (top) and reference (bottom) signals*

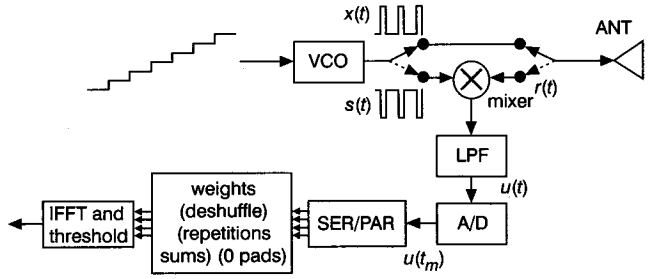

Fig. 20 *Stretch processing of a stepped-frequency pulse-train*

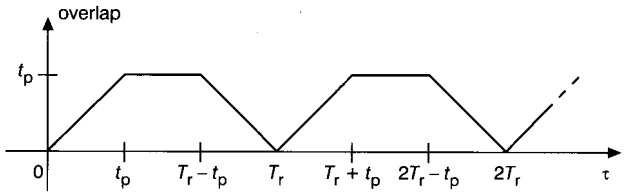

Fig. 21 *Overlap between a received signal delayed by τ and a reference signal*

from a stationary point target is a delayed version of the transmitted one

$$r(t) = x(t - \tau) \tag{7}$$

The mixer, low-pass filter and analogue-to-digital converter create one complex sample of the low-pass filtered mixer output, for each pulse. These samples could be approximated by the expression

$$\boldsymbol{u}(mT_r) = \int_{(m-1)T_r}^{mT_r} x(t - \tau)s^*(t)dt \tag{8}$$

The M complex samples corresponding to the outcome of the M pulses in a batch are multiplied by the M weights representing the different amplitudes B_m of the reference signal. If the frequency coding was other than linear (e.g. Costas) then order deshuffle is required before weighting and feeding the M complex numbers to an inverse FFT, which then yields the intensities of the returns at the effective range bins. If the frequency coding were 'NLFM by repetitions' then the weighting would have been implemented by summing the samples from same-frequency pulses. The 'weights' block represents these other functions (deshuffle and repetitions sums) when needed. That block, if desired, also represents zero-padding the input to the IFFT.

Some results from a simulated stretch processor are given in Figs. 21 to 25. To be able to see waveform details, the batch was constructed from only 16 pulses. A practical system, like the automotive radar described in [5], utilises a 1000-pulse batch. The transmitter duty–cycle is 1/3, namely $T_r = 3t_p$. Note that even if the delay τ is less than the pulse duration t_p there is already some overlap between the received signal and the reference signal. Assuming infinitesimal rise-times the overlap as a function of delay is given in Fig. 21.

The phase of the received signal reflected from a point target with a delay of $\tau = 0.6t_p$ is shown in Fig. 22 and compared to the reference signal (centre). The phase is measured with respect to the phase of a virtual sine wave at the centre of the frequency band, namely at $(f_8 + f_9)/2$. This choice of reference frequency explains why the phase ramps of the outer pulses are higher than those of the inner pulses, and why the phase slope reverses between pulses 8 and 9. The phase of the mixer output is given at the bottom of Fig. 22. Note the constant phase during the entire overlap of a given pulse. Hence, the integral in (8) adds phasors that are all in the same direction. This constant phase during each pulse at the mixer output, gives results only for a stationary point target. Thus the $M = 16$ samples of the LPF (or integrator) output, one for each pulse, will have phases identical to those plotted in the bottom section of Fig. 22. The magnitude will depend on the mixer output pulse duration (the overlap) and the intensity of the reflected signal. The overlap chart (Fig. 21) indicates a useful inherent gain control, which linearly suppresses close targets, up to a delay equal to the transmitted pulse-width t_p.

To demonstrate the IFFT output for different scenes, the number of pulses (and frequencies) was raised to $M = 64$. Zero-padding increased the length of the IFFT output vector by a factor of $ZPF = 4$, to 256. Fig. 23 displays the output assuming two stationary targets at normalised delays of $\tau/t_p = 0.5$ and 2. The relationship between the

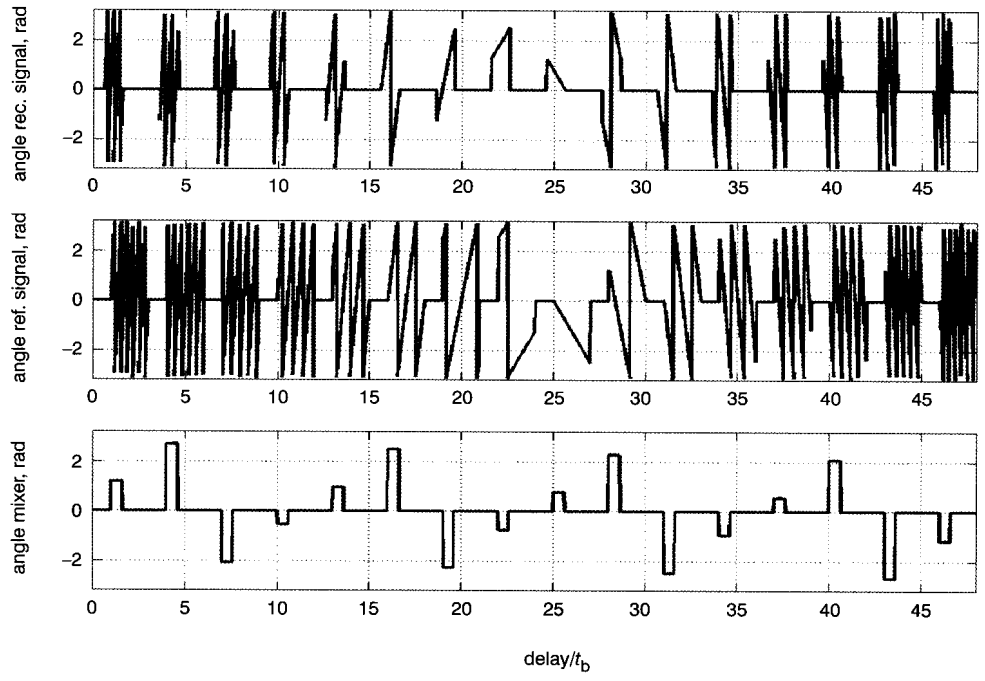

Fig. 22 *Phase of transmitted (top), received (middle) and mixer output (bottom) signals*

$M = 16$, number of frequencies = 16, $T_r = 3t_p$, $t_p\Delta f = 0.4$, frequency weight power = 1, Chebyshev weighting (ripple 50 dB) $\tau = 0.6t_p$
Frequency order: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

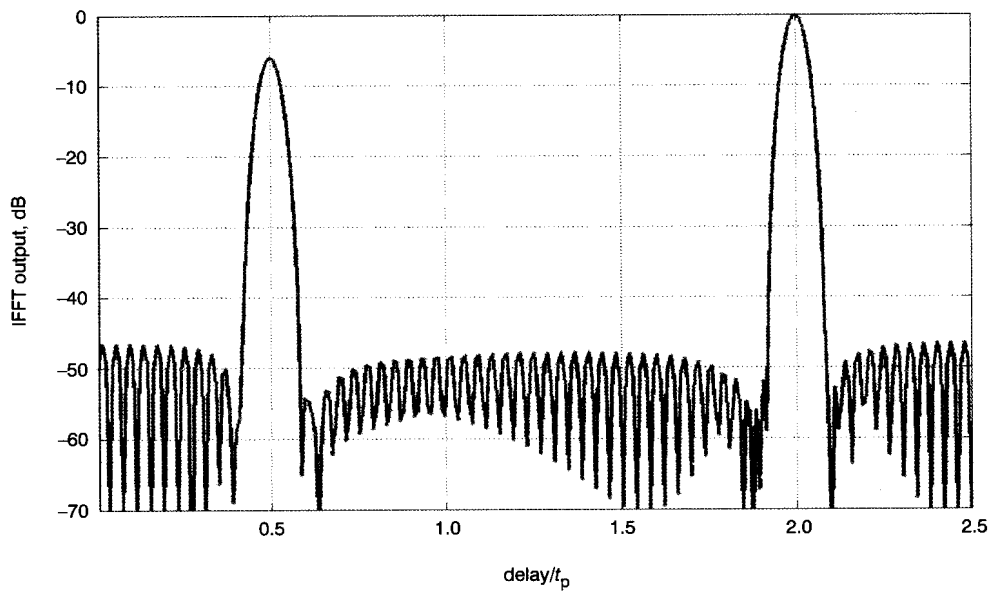

Fig. 23 *IFFT output with two targets at normalised delays of $\tau/t_p = 0.5$ and 2, LFM coding*
The targets exhibit identical received intensities; both targets are stationary
$M = 64$, number of frequencies = 64, $T_r = 3t_p$, $t_p\Delta f = 0.4$, Chebyshev weighting (ripple = 50 dB), 1st delay = $0.5t_p$, 2nd delay = $2t_p$

IFFT output bin number n_b and the normalised delay is given by

$$\frac{\tau}{t_p} = \frac{n_b}{ZPFt_p\Delta fM} \tag{9}$$

The two signals were LFM-coded, with Chebyshev weighting (−50 dB ripple), only on receive. The signals exhibited equal intensities at the receiver input. The peak representing the closer target is lower because of the reduced overlap.

The scene that produced Fig. 24 is identical to the scene that produced Fig. 23, except that the closer target is moving, causing a normalised Doppler shift of $f_DT_r/(t_p\Delta f) = 0.25$. Note that the peak corresponding to this target has moved by $\tau/t_p = 0.25$ from its bin position without Doppler. This result was predicted by (4).

A completely different effect is observed when Costas coding is used. Figs 25 and 26 were obtained with 57 stepped-frequency pulses, coded according to Costas. They refer to the same scenes as Figs 23 and 24. When the two targets are stationary (Fig. 25), the stretch processor clearly resolves them, with sidelobes around the planed −50 dB level. However, when the first target exhibits a normalised Doppler shift of $f_DT_r/(t_p\Delta f) = 0.25$, we note from Fig. 26 that the main lobe of that first target was replaced by sidelobes extending over the entire delay span. The relative level of the raised sidelobes is about −11 dB below the replaced peak. This result was predicted qualitatively by the ambiguity function (Fig. 10).

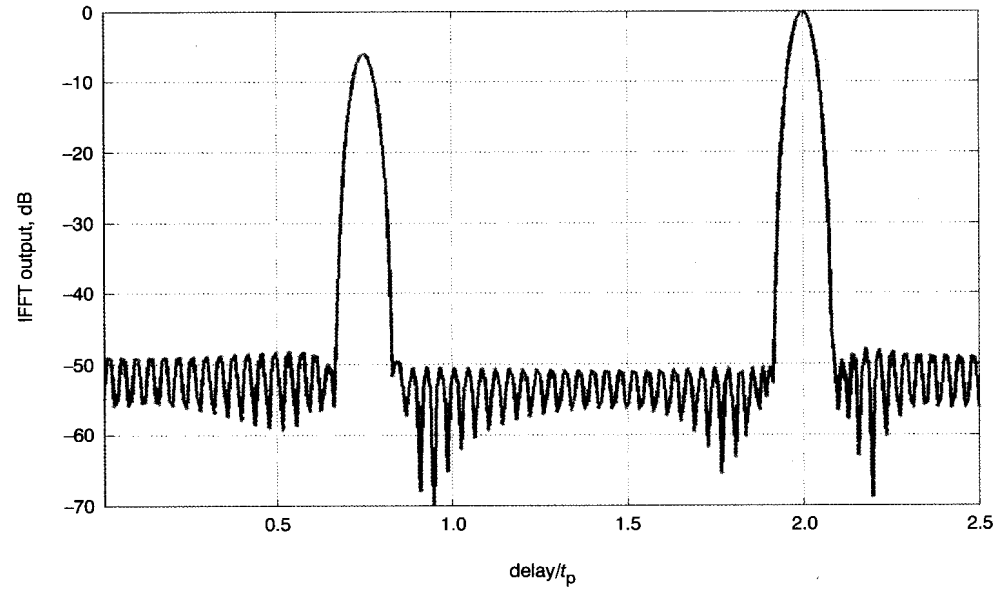

Fig. 24 *IFFT output with two targets at normalised delays of $\tau/t_p = 0.5$ and 2, LFM coding*
The targets exhibits identical received intensities; the target at $\tau/t_p = 0.5$ is moving with a normalised Doppler of $f_DT_r(t_p\Delta f) = 0.25$
$M = 64$, number of frequencies = 64, $T_r = 3t_p$, $t_p\Delta f = 0.4$, Chebyshev weighting (ripple = 50 dB), 1st delay = $0.5t_p$, 2nd delay = $2t_p$

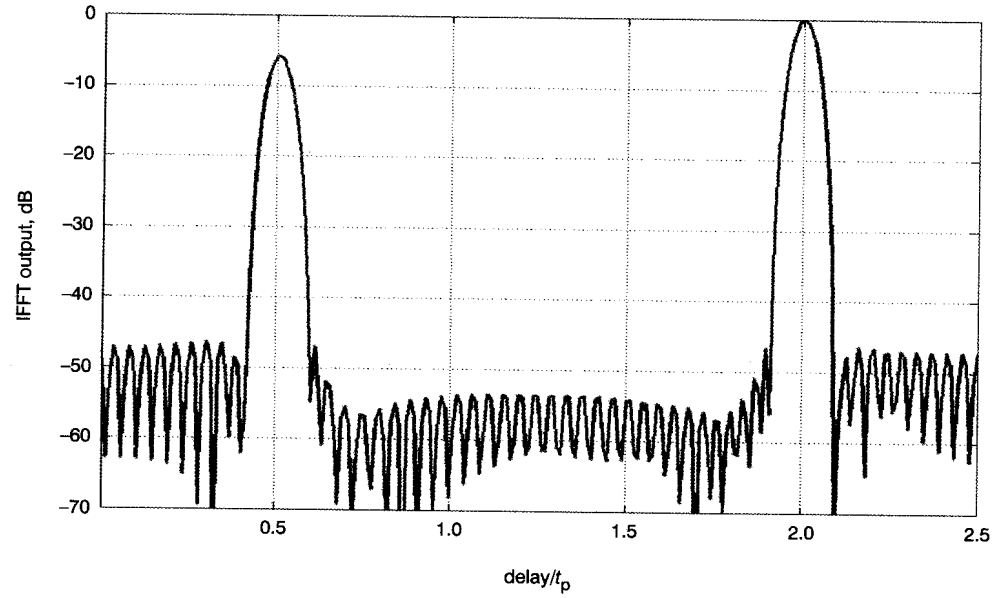

Fig. 25 *IFFT output with two targets at normalised delays of $\tau/t_p = 0.5$ and 2, Costas coding*

The targets exhibit identical received intensities; both targets are stationary
$M = 57$, number of frequencies = 57, $T_r = 3t_p$, $t_p\Delta f = 0.4$, frequency weight power = 1, Chebyshev weighting (ripple = 50 dB), 1st delay = $0.5t_p$, 2nd delay = $2t_p$
Frequency order: 2 1 43 9 48 27 13 31 4 44 55 20 7 53 47 19 17 25 16 12 38 33 45 49 18 39 6 56 41 57 8 42 22 54 51 40 46 21 26 36 29 32 3 10 23 37 15 5 24 52 35 50 14 34 11 28 30

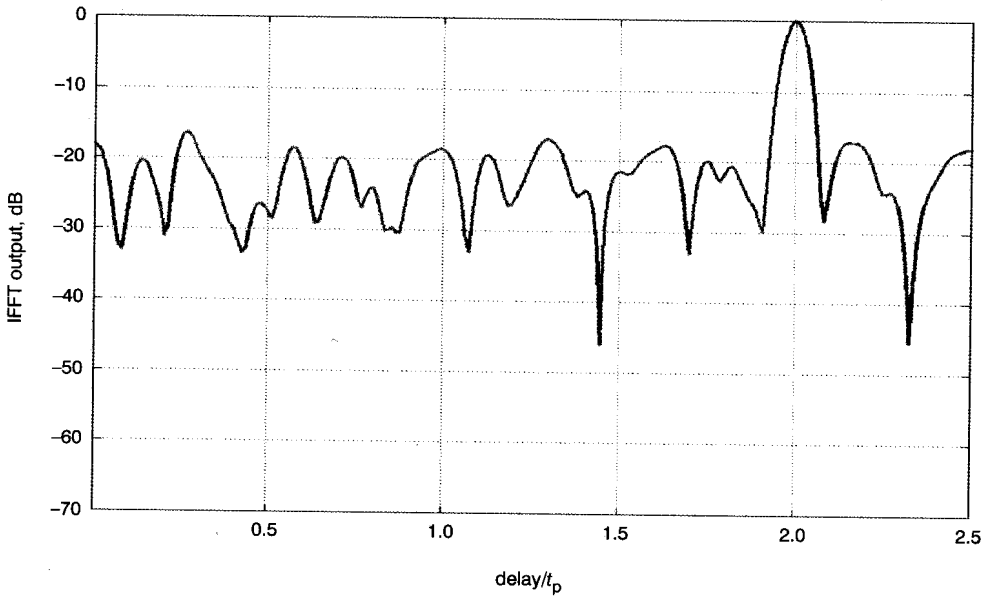

Fig. 26 *IFFT output with two targets at normalised delays of $\tau/t_p = 0.5$ and 2, Costas coding*

The targets exhibits identical received intensities; the target at $\tau/t_p = 0.5$ is moving with a normalised Doppler of $f_D T_r(t_p\Delta f) = 0.25$
$M = 57$, number of frequencies = 57, $T_r = 3t_p$, $t_p\Delta f = 0.4$, frequency weight power = 1, Chebyshev weighting (ripple = 50 dB), 1st delay = $0.5t_p$, 2nd delay = $2t_p$
Frequency order: 2 1 43 9 48 27 13 31 4 44 55 20 7 53 47 19 17 25 16 12 38 33 45 49 18 39 6 56 41 57 8 42 22 54 51 40 46 21 26 36 29 32 3 10 23 37 15 5 24 52 35 50 14 34 11 28 30

4. Resolving Doppler in stretch processing

4.1 Linear frequency coding

The expected scenario in automotive radar applications could involve many targets with different ranges and Doppler shifts. In Sections 2 and 3 we have seen signals whose ambiguity function exhibits diagonal ridges but otherwise low sidelobes (e.g. linear frequency coding) and signals with a thumbtack ambiguity function with a relatively large sidelobe pedestal (e.g. Costas frequency coding). These ambiguity properties affect both types of processors, matched filter and stretched. In this Section we will consider enhancements to stretch processing required to resolve targets in range and Doppler, especially in radar scenes crowded with targets. We will begin with linear frequency coding which is the more popular signal.

In stretch processing, without additional Doppler processing, a target at delay τ and Doppler shift f_D will yield a peak at the IFFT bin number

$$n_{b+} = M\,ZPF(\Delta f\tau + T_r f_D) \tag{10}$$

where M is the number of frequencies (and pulses), ZPF is the zero-padding factor, Δf is the frequency step and T_r is the pulse repetition interval. The two unknowns τ and f_D, cannot be resolved from only one measurement n_{b+}. The recommended solution is to process another batch of pulses with the opposite frequency slope. The new peak will appear at bin number

$$n_{b-} = M\,ZPF(-\Delta f\tau + T_r f_D) \tag{11}$$

If a target could be associated with its two corresponding bins, then its range and Doppler could be resolved from algebraic sum and difference

$$\tau = \frac{n_{b+} - n_{b-}}{2M\,ZPF\,\Delta f} \tag{12}$$

$$f_D = \frac{n_{b+} + n_{b-}}{2M\,ZPF\,T_r} \tag{13}$$

When there are many targets with a spread of delays and Doppler shifts, it may become rather difficult to associate each target with its two corresponding bins n_{b+} and n_{b-}. In [5] a third batch of pulses is suggested with no frequency slope ($\Delta f = 0$). In that case a target will yield an IFFT peak at bin number

$$n_{b0} = M\,ZPF\,T_r f_D \tag{14}$$

An example of the phases of the two signals received from two point targets is given in Fig. 27 together with the phases of the signal at the mixer output. Note that Doppler shift adds an intrapulse ramp to the phase of the mixer output. The corresponding IFFT output is presented in Fig. 28.

The importance of the fixed-frequency batch of pulses can be demonstrated with the help of Fig. 29. The example shows the procedure needed to resolve the two co-ordinates (delay and Doppler) of $N = 4$ targets. For the given example the delay bins detected with the first batch are marked on the horizontal axis as points {A B C D}. Diagonal lines with positive slope are drawn from these points. The lines are wrapped modulo 64. The delay bins obtained from the second batch (opposite frequency slope) were {a b c d}, respectively. From these points on the horizontal axis, diagonal lines with negative slope are drawn. The $2N$ wrapped lines will yield $2N^2$ intersections, of which only N are true. The Doppler values of the N true intersections will be the N Doppler values $\{f_a f_b f_c f_d\}$ obtained from the third batch of pulses (with no frequency slope) and are represented by solid horizontal lines in Fig. 29. Close observation of Fig. 29 will reveal that some of the false intersections are close enough to one of the four horizontal lines to create uncertainty. This problem will aggravate as the number of targets grows. Other targets join clutter and thermal noise to decrease the effective SNR and reduce the accuracy of the estimated range and Doppler. Also note that the radar scene may change between the beginning of the first batch and the end of the third batch of pulses. The changing scene can cause the three-way intersections to be poorly defined.

4.2 Costas frequency coding

Both the sensitivity to noise of the algebraic solution in linear frequency coding, and the sensitivity to changing scene during the three consecutive frequency-coded pulse-

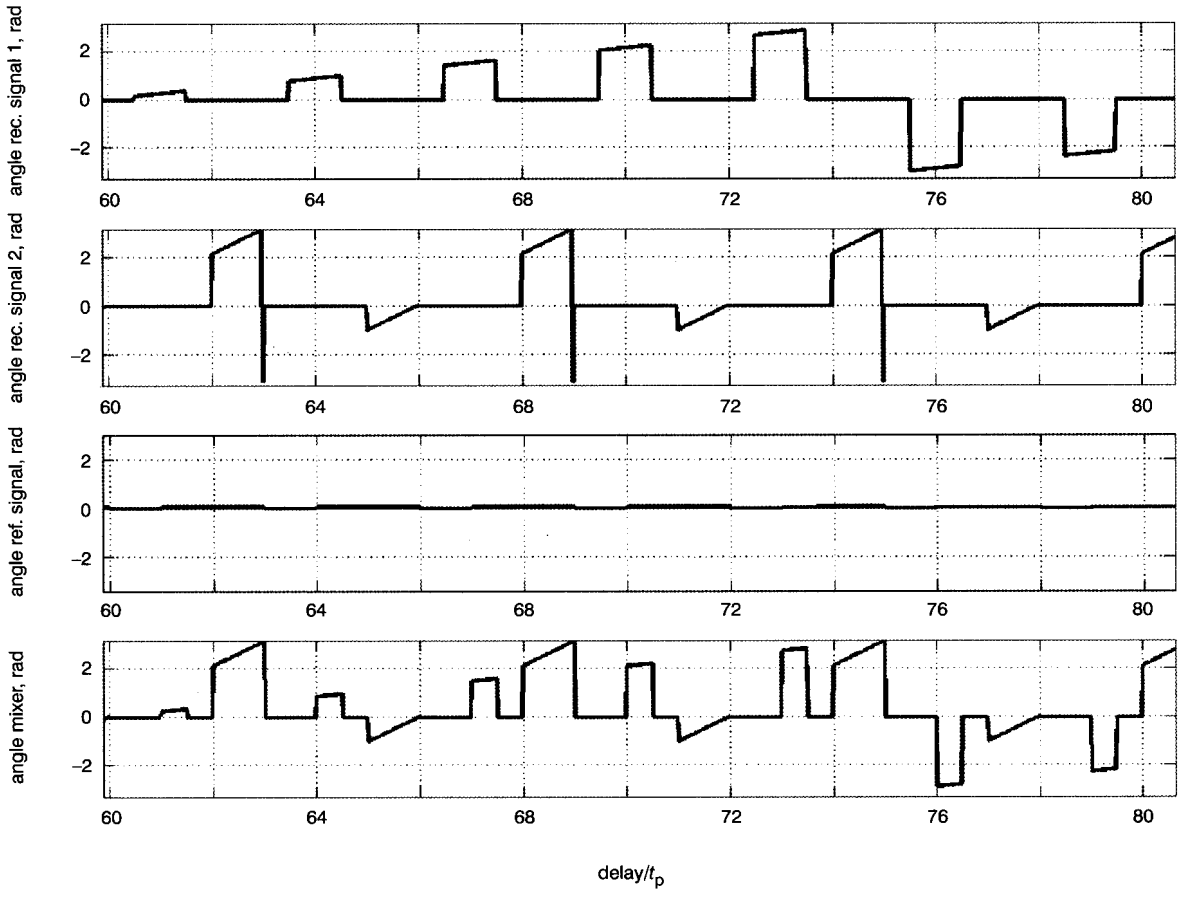

Fig. 27 *Stretch processing of 64 constant-frequency pulses with two targets: (i) $t/t_p = 0.5$, $f_D T_r = 0.1$, (ii) $t/t_p = 2$, $f_D T_r = 0.5$: phases during pulses 20 to 26*

Number of frequencies = 64, $T_r = 3t_p$, frequency weight power = 0.5, Chebyshev weighting (ripple 50 dB)

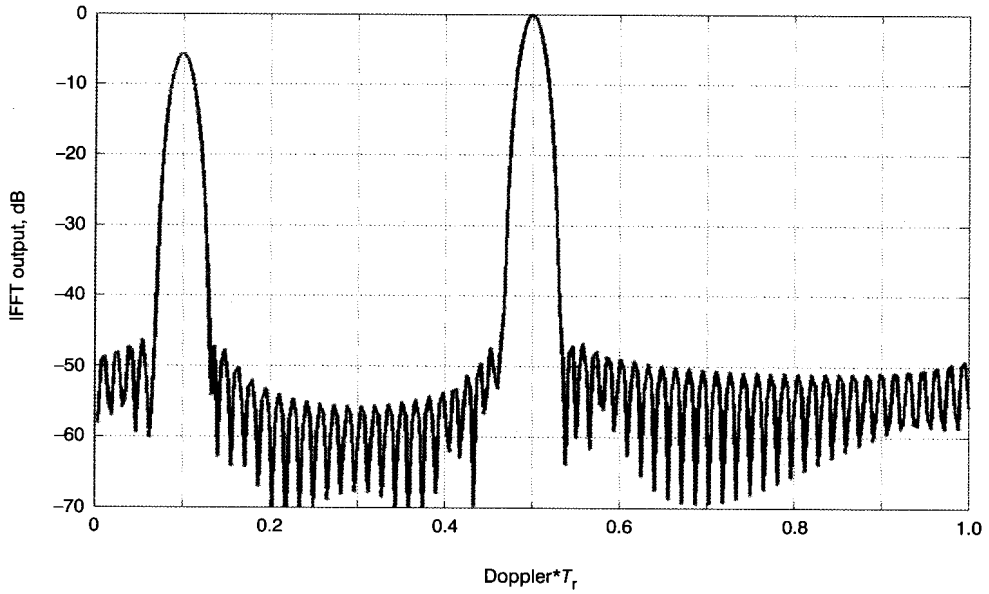

Fig. 28 *Stretch processing of 64 constant-frequency pulses with two targets: (i) $t/t_p = 0.5$, $f_D T_r = 0.1$, (ii) $t/t_p = 2$, $f_D T_r = 0.5$: IFFT output*
Number of frequencies = 64, $T_r = 3t_p$, frequency weight power = 1, Chebyshev weighting (ripple 50 dB)

trains (up, down and flat), can be alleviated by using Costas coding and a multitude of Doppler filters. Due to its thumbtack ambiguity function (see Fig. 10), only one Costas-coded pulse-train is needed. However, the high Doppler resolution imposes a need for a bank of Doppler filters to cover all possible Doppler shifts. Stretch implementation employing a bank of Doppler filters is suggested in Fig. 30, which is a modification of the processor in Fig. 20. The vector $\boldsymbol{u}$, at the output of the serial/parallel converter contains M complex elements, one for each pulse, ordered chronologically. In the original processor (Fig. 20) it was fed to a single IFFT block (after deshuf-

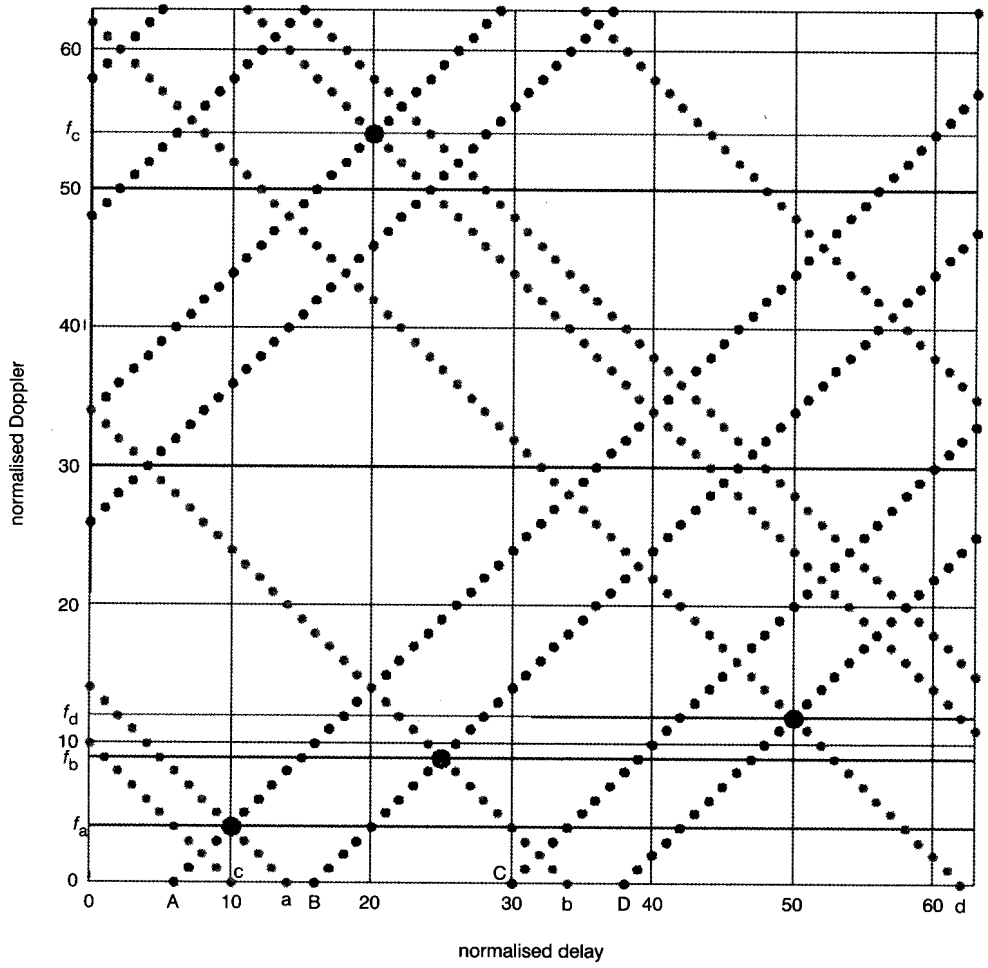

Fig. 29 *Using a third batch of pulses with no frequency slope to identify the N true intersections representing the N targets, from a total of $2N^2$ intersections*

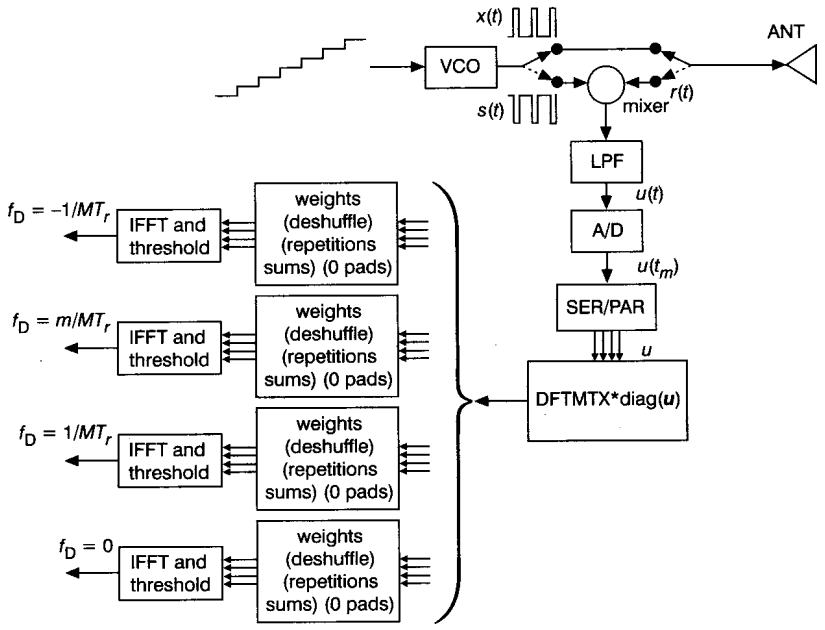

Fig. 30 *Stretch processing of a stepped-frequency pulse-train including a bank of Doppler filters*

fling and weighting) to yield a response matched to zero Doppler. To implement a bank of Doppler filters we need to generate many versions of vector $\boldsymbol{u}$, each one compensating a different Doppler shift. Such an operation is performed in the block marked DFTMTX*diag($\boldsymbol{u}$), which is exactly what this block performs in MATLAB terminology. DFTMTX(M) is the M-by-M complex matrix of values around the unit-circle whose inner product with a column vector of length M yields the discrete Fourier transform of the vector. In our block we multiply this matrix by an M-by-M diagonal matrix, whose diagonal elements are the M elements of $\boldsymbol{u}$. Each raw in the resulted M-by-M complex array, is a different (progressive) Doppler-compensated version of $\boldsymbol{u}$. Each raw is identically processed, namely; deshuffling, weighting and IFFT.

The absolute value (linear scale) of the outputs of M such filters is presented in Fig. 31. The signal was a Costas-coded pulse-train with $M = 144$ pulses. There were two targets, with identical received intensities, and the following delays and Doppler shifts:

$$\text{(i) } t/t_p = 1, f_D T_r = -0.5 \quad \text{(ii) } t/t_p = 2, f_D T_r = 0$$

The peaks corresponding to the two targets are clearly visible at their expected locations, above a pedestal of approximately -20 dB. If the targets were not of equal intensity, then the pedestal level would be approximately -20 dB below the peak of the stronger target. The relative height of the pedestal decreases as the number of pulses increases. In Costas coding there is no good reason to add frequency weighting. Frequency weighting reduces sidelobes from interfering targets which happen to have the same Doppler shift as the processed target. Sidelobes from targets at different Doppler shifts remain high despite

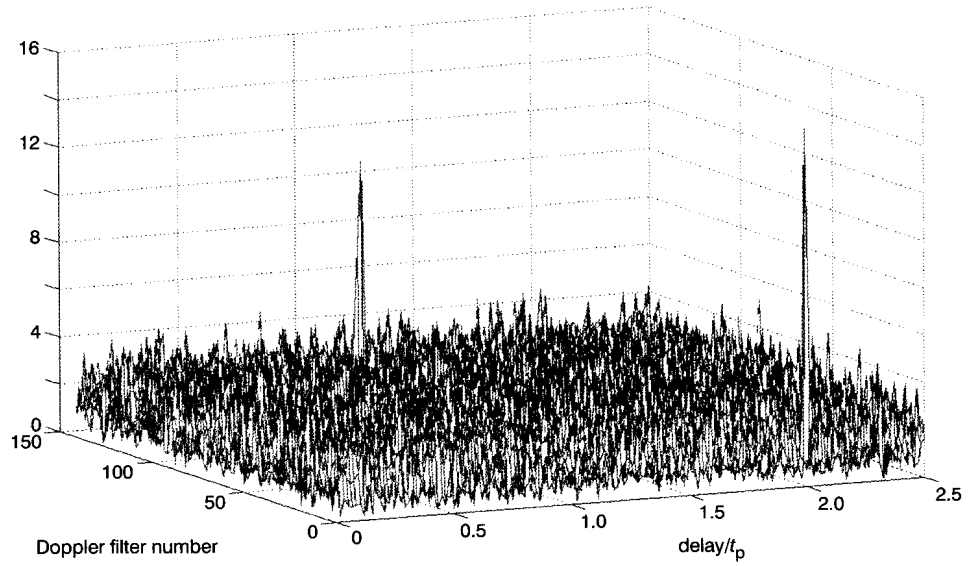

Fig. 31 *Response of all Doppler filters used in stretched-processing of 144 step-frequency pulses, Costas coded, to two equal intensity targets at (i) $t/t_p = 1$, $f_D T_r = -0.5$, (ii) $t/t_p = 2$, $f_D T_r = 0$*

Number of frequencies = 144, $T_r = 3t_p$, $t_p \Delta f = 0.4$, no amplitude weighting

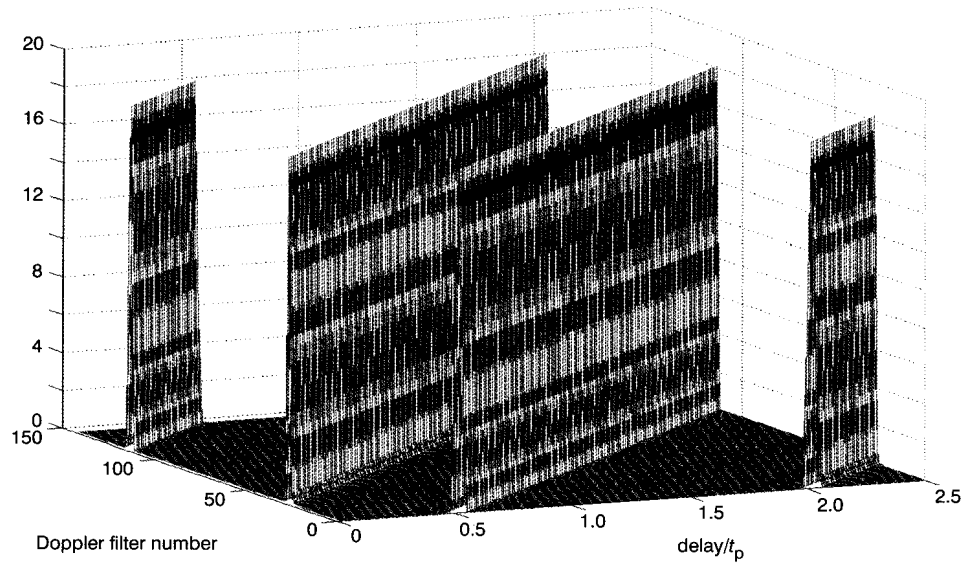

Fig. 32 *Response of all Doppler filters used in stretched-processing of 144 step-frequency pulses, linearly coded with two equal intensity targets at (i) $t/t_p = 1$, $f_D T_r = -0.5$, (ii) $t/t_p = 2$, $f_D T_r = 0$*
Number of frequencies = 144, $T_r = 3t_p$, $t_p \Delta f = 0.4$, Chebyshev weighting (ripple = 50 dB)

weighting. Hence, Fig. 31 was obtained without frequency weighting.

The filter-bank processor can be applied to other signals as well, including linear frequency coding. The outputs for linear frequency coded train of $M = 144$ pulses are presented in Fig. 32 for the same two-target scene. In the linear frequency coding, two more pulse-trains are required, at opposite and at flat-frequency slope, to separate delay from Doppler.

Comparing Figs. 31 and 32 reveals the trade-off between Costas and linear frequency coding. Costas coding exhibits thumbtack response in which the targets are well defined, but the sidelobe pedestal is relatively high, implying possible masking if one target is much stronger than the rest. Linear coding exhibits low sidelobes but extensive ridges, creating delay–Doppler coupling. As the number of targets grows the pedestal will rise in Costas coding, and the number of ridges will increase in linear coding. Which one is more useful depends on the application. It may also be argued that just as we can combine (algebraically) the results from different linear slopes, we can combine linear slopes and Costas coding. A similar approach was suggested in [8] of combining the results from the three slopes (up, down and zero) with a fourth, batch-coded according to residual-class code, rather than Costas.

Finally, we note that the output of stretched processing using a filter bank strongly resembles the ambiguity function. The major difference is the attenuation of targets at short delays. Even though large sections of the signal are not reaching the receiver, the ambiguity function still predicts the output, because of the noncontiguous nature of the pulses. Each pulse still produces its one complex sample. Masking reduces its magnitude but does not affect its phase.

5 References

1 RIHACZEK, A.W.: 'Principles of high resolution radar' (McGraw–Hill, 1969)
2 FARNET, E.C., and STEVENS, G.H.: 'Pulse compression radar' in SKOLNIK, M.I. (Ed.): 'Radar handbook' (McGraw–Hill, 1990, 2nd edn.), Chap. 10
3 WEHNER, D.R.: 'High resolution radar' (Artech House, 1987)
4 KAJIWARA, A.: 'Stepped-FM pulse radar for vehicular collision avoidance', *Electron. Commun. Jpn. 1, Commun.*, 1999, **82**, (6), pp. 1–7
5 ROHLING, H., and MENDE, R.: 'Method for operating a radar system'. US Patent 6,147,638, 14 Nov. 2000
6 COSTAS, J.P.: 'A study of a class of detection waveforms having nearly ideal range-Doppler ambiguity function', *Proc. IEEE*, 1984, **72**, (8), pp. 996–1009
7 COXSON, G.E., private communication
8 ROHLING, H., and LISSEL, E.: '77 GHz radar sensor for car application'. Proceedings of IEEE International Radar Conference, Alexandria, VA, USA, May 1995, pp. 373–379