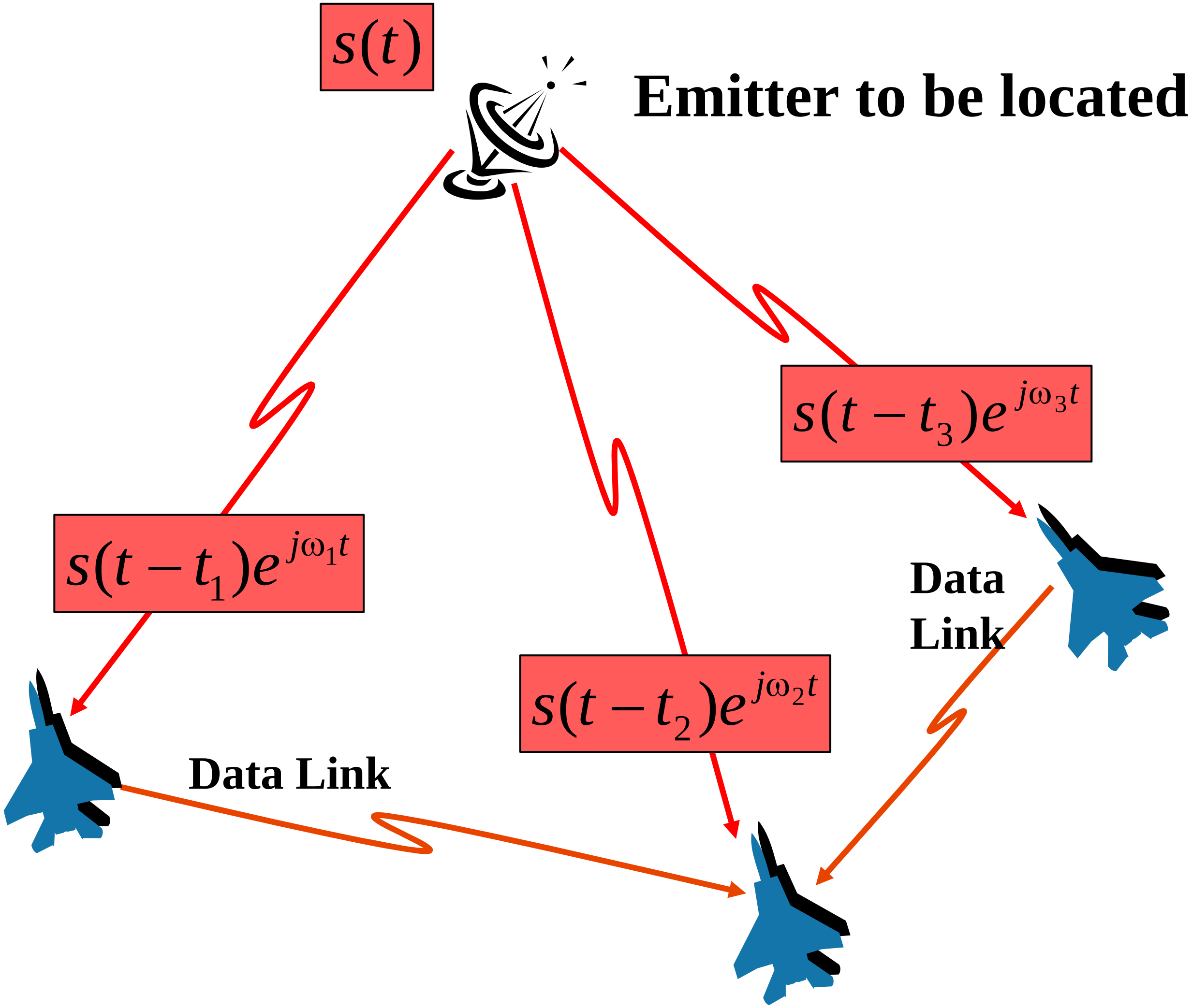
Correlation Processing

An Example Application of:

- Lowpass Equivalent Signal
- DFT-Based Processing
- Decimation

Multiple-Platform Location



$s(t)$

Emitter to be located

$s(t-t_1)e^{j\omega_1 t}$

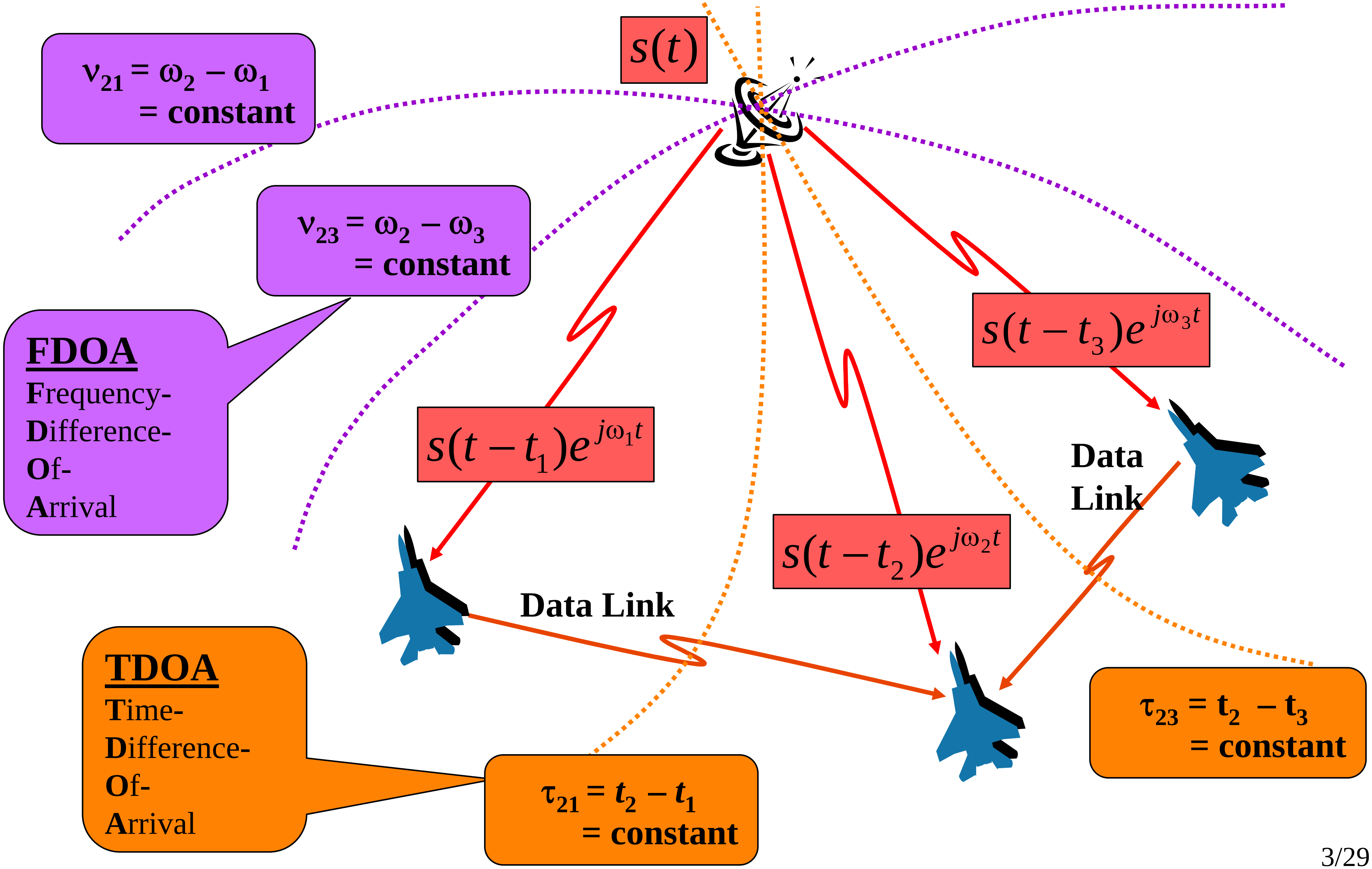
$s(t-t_2)e^{j\omega_2 t}$

$s(t-t_3)e^{j\omega_3 t}$

Data Link

Data Link

TDOA/FDOA LOCATION



$\nu_{21} = \omega_2 - \omega_1$ **= constant**

$\nu_{23} = \omega_2 - \omega_3$ **= constant**

FDOA
Frequency-
Difference-
Of-
Arrival

$s(t)$

$s(t - t_1)e^{j\omega_1 t}$

$s(t - t_2)e^{j\omega_2 t}$

$s(t - t_3)e^{j\omega_3 t}$

Data Link

Data Link

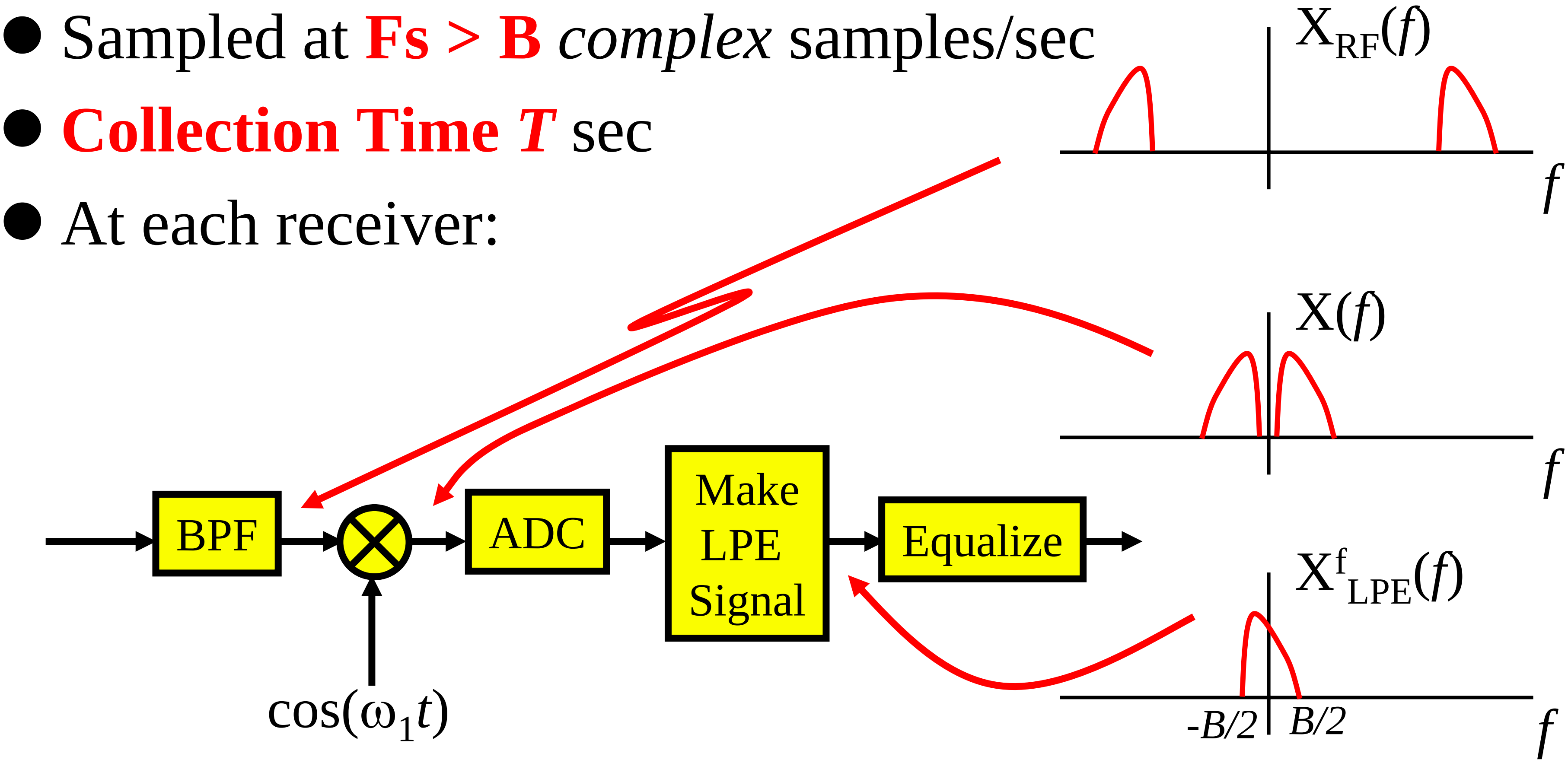
TDOA
Time-
Difference-
Of-
Arrival

$\tau_{21} = t_2 - t_1$ **= constant**

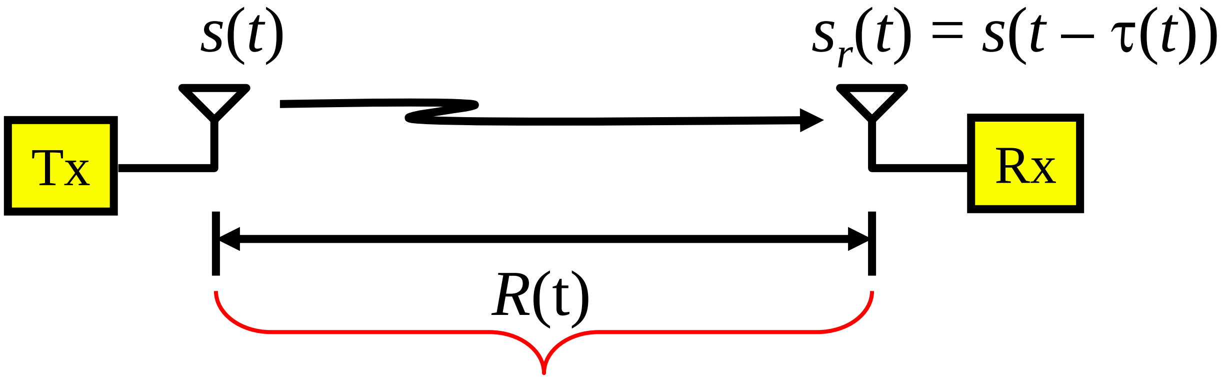
$\tau_{23} = t_2 - t_3$ **= constant**

SIGNAL MODEL

- Will Process Equivalent Lowpass signal, **BW = *B* Hz**
  - Representing RF signal with RF BW = B Hz
- Sampled at **Fs > B** *complex* samples/sec
- **Collection Time *T*** sec
- At each receiver:



Doppler & Delay Model



$s(t)$

$s_r(t) = s(t - \tau(t))$

Tx

Rx

$R(t)$

Propagation Time: $\tau(t) = R(t)/c$

$$R(t) = \underbrace{R_o + vt}_{\text{Use linear approximation – assumes small change in velocity over observation interval}} + (a/2)t^2 + \cdots$$

For Real BP Signals:

$$s_r(t) = s(t - [R_o + vt]/c) = s(\underbrace{[1 - v/c]}_{\text{Time Scaling}}t - \underbrace{R_o/c}_{\text{Time Delay: } \tau_d})$$

DOPPLER & DELAY MODEL (continued)

Analytic Signals Model

$$\tilde{s}(t) = E(t)e^{j[\omega_c t + \phi(t)]}$$

Analytic Signal of Tx

$$\tilde{s}_r(t) = \tilde{s}([1 - v/c]t - \tau_d)$$

$$= E([1 - v/c]t - \tau_d)e^{j\{\omega_c([1-v/c]t-\tau_d) + \phi([1-v/c]t-\tau_d)\}}$$

Analytic Signal of Rx

Now what? Notice that $v << c$ ➔ $(1 - v/c) \approx 1$

Say $v = -300$ m/s (–670 mph) then $v/c = -300/3\text{x}10^8 = -10^{-6}$ ➔ $(1 - v/c)=1.000001$

Now assume $E(t)$ & $\phi(t)$ vary slowly enough that

$$E([1 - v/c]t) \approx E(t)$$

$$\phi([1 - v/c]t) \approx \phi(t)$$

For the range of v of interest

Called Narrowband Approximation

DOPPLER & DELAY MODEL (continued)

Narrowband Analytic Signal Model

$$
\boxed{
\begin{aligned}
\tilde{s}_r(t) &= E(t-\tau_d)e^{j\{\omega_c t-\omega_c (v/c)t-\omega_c \tau_d+\phi(t-\tau_d)\}} \\
&= \underbrace{e^{-j\omega_c\tau_d}}_{\substack{\text{Constant}\\ \text{Phase}\\ \text{Term}\\ \alpha=-\omega_c\tau_d}}\;\underbrace{e^{-j\omega_c(v/c)t}}_{\substack{\text{Doppler}\\ \text{Shift}\\ \text{Term}\\ \omega_d=\omega_c v/c}}\;\underbrace{e^{j\omega_c t}}_{\substack{\text{Carrier}\\ \text{Term}}}\;\underbrace{E(t-\tau_d)e^{j\phi(t-\tau_d)}}_{\substack{\text{Transmitted Signal's}\\ \text{LPE Signal}\\ \text{Time-Shifted by } \tau_d}}
\end{aligned}
}
$$

Narrowband Lowpass Equivalent Signal Model

$$\boxed{\hat{s}_r(t) = e^{j\alpha}e^{-j\omega_d t}\,\hat{s}(t-\tau_d)}$$

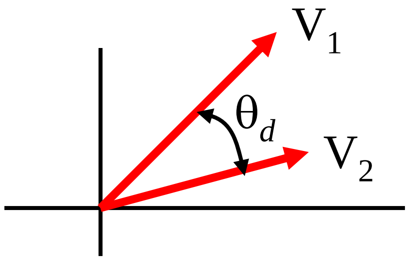
This is the signal that actually gets processed digitally

Estimating Doppler & Delay

Consider C-T view first for simplicity, then switch to D-T
(Note: all signals are LPE signals, but we don’t use “hat”)

Problem: Given LPE $s_1(t) = s(t) \quad \& \quad s_2(t) = e^{j\alpha} e^{-j\omega_d t} s(t - \tau_d)$
for $t \in [0,T]$, compute an estimate of delay τ_d and doppler ω_d.

Motivation: Think about vectors in $\boldsymbol{R}^2$:



Note: For notational purposes we will often use $s(t) = s_1(t)$

To measure θ_d: Let $V_2(\theta)$ be V_2 rotated clockwise by θ…
and for each θ compute **$< V_1, V_2(\theta)> = A(\theta)$** as a function of θ.

Note: $A(\theta)$ has a maximum at $\theta = \theta_d$, so we measure θ_d by **finding the peak of $A(\theta)$**.

ESTIMATING DOPPLER & DELAY (cont.)

Do the same thing with signals: Let $s_{\omega,\tau}(t) = e^{j\omega t} s_2(t+\tau)$

$$\begin{aligned} A(\omega,\tau) &= < s_1(t), s_{\omega,\tau}(t) > \\ &= \int_0^T s_1(t)\,\overline{s_{\omega,\tau}(t)}\,dt \\ &= \int_0^T s_1(t)\,\overline{s_2(t+\tau)}\,e^{-j\omega t}\,dt \end{aligned}$$

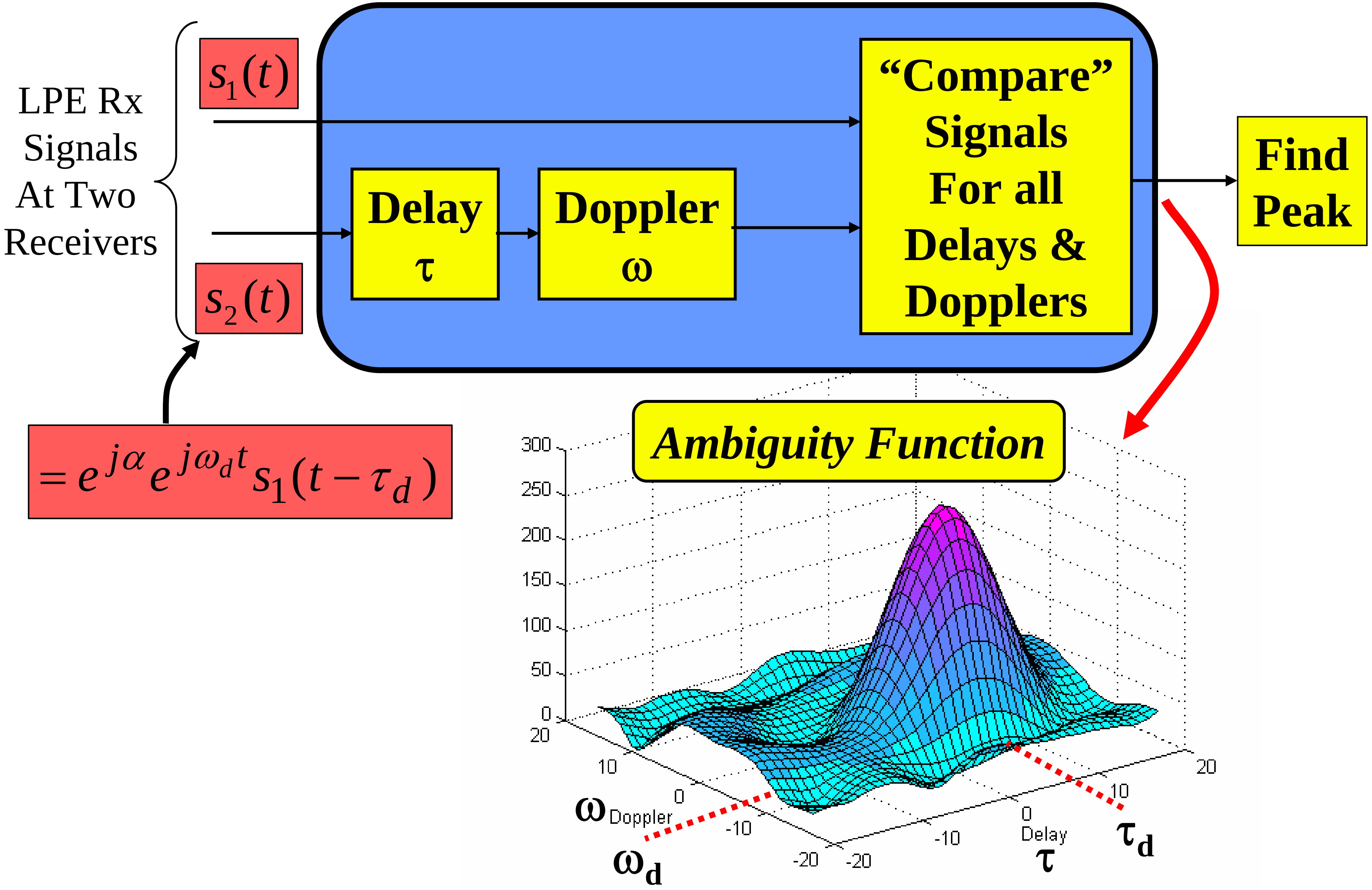
We know from the inner product view that |A(ω,τ)| has a maximum at $\omega = \omega_d$ and $\tau = \tau_d$. Also note:

$$\left|A(\omega,\tau)\right| = \left|\int_0^T s(t)\,\overline{s(t-\tau_d+\tau)}\,e^{-j(\omega-\omega_d)t}\,dt\right|$$

$$\Rightarrow \quad \left|A(\omega_d,\tau_d)\right| = \int_0^T \left|s(t)\right|^2 dt = E_s$$

Signal Energy

ESTIMATING DOPPLER & DELAY (cont.)



LPE Rx Signals At Two Receivers

$s_1(t)$

$s_2(t)$

$= e^{j\alpha}e^{j\omega_d t}s_1(t-\tau_d)$

Delay τ

Doppler ω

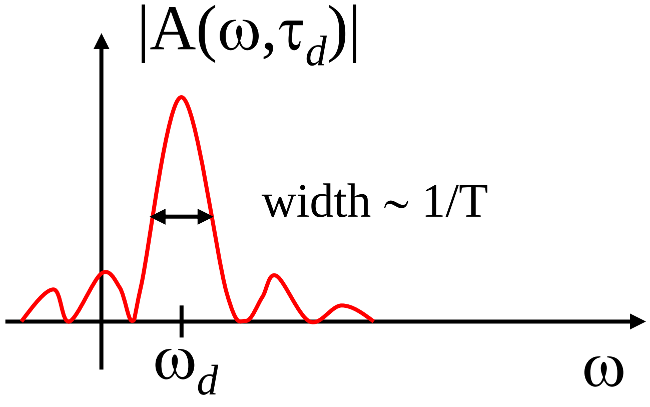
"Compare" Signals For all Delays & Dopplers

Find Peak

Ambiguity Function

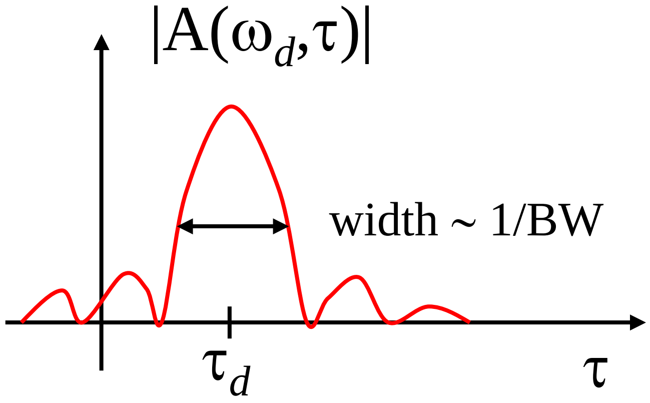
ESTIMATING DOPPLER & DELAY (cont.)

- Consider when $\tau = \tau_d$



$$\boxed{\left|A(\omega,\tau_d)\right| = \underbrace{\left|\int_0^T \left[\left|s(t)\right|^2 e^{j\omega_d t}\right] e^{-j\omega t}\,dt\right|}_{\text{like windowed FT of sinusoid where window is } |s(t)|^2}}$$

- Consider when $\omega = \omega_d$



$$\boxed{\left|A(\omega_d,\tau)\right| = \left|\int_0^T \underbrace{s(t)\,\overline{s(t-\tau_d+\tau)}\,dt}_{\text{correlation}}\right|}$$

Computing The Ambiguity Function

Obviously we can only compute |A(ω,τ)| for discrete values of ω and τ:

Need to know a priori values for:

Max/Min Doppler (from largest expected velocity difference)
Max/Min Delay (from largest expected range difference)
Delay Spacing (from expected/measured signal BW)
Doppler Spacing (from observation time T)

Recall: $$\boxed{A(\omega,\tau) = \int_0^T s_1(t)\,\overline{s_2(t+\tau)}\,e^{-j\omega t}\,dt}$$

View as a FT ➔ Implies Use of DFT

Define "lag-product" signal : $f_\tau(t) = s_1(t)\,\overline{s_2(t+\tau)}$

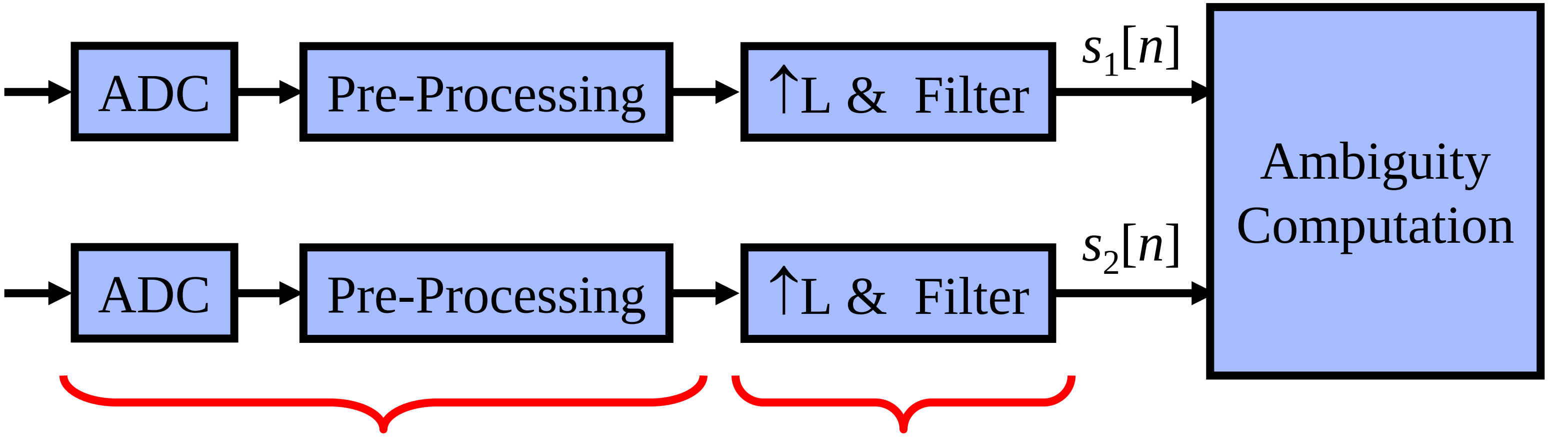
Thus, for each delay τ_m of interest : $$\boxed{A(\omega,\tau_m) = \mathscr{F}\{ f_{\tau_m}(t)\}}$$

Computing The Ambiguity Function (cont.)

For corresponding D-T signals we need: $\boxed{f_m[n] = s_1[n]\overline{s_2[n+m]}}$

m = delay index

Sampling Interval sets Delay Spacing

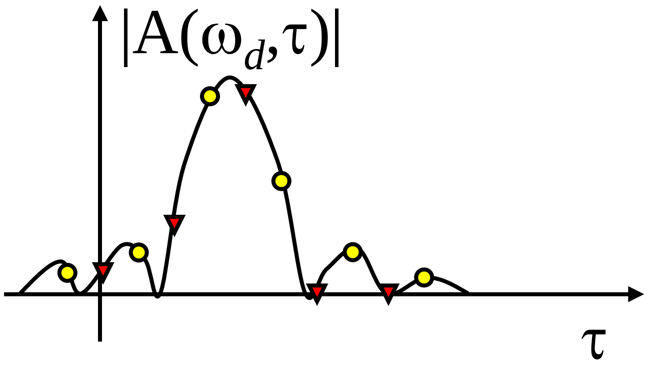


ADC → Pre-Processing → ↑L & Filter → $s_1[n]$ → Ambiguity Computation

ADC → Pre-Processing → ↑L & Filter → $s_2[n]$ → Ambiguity Computation

Sampling rate chosen to match signal BW according to Nyquist

Interpolate to get desired delay spacing



COMPUTING THE AMBIGUITY FUNCTION (cont.)

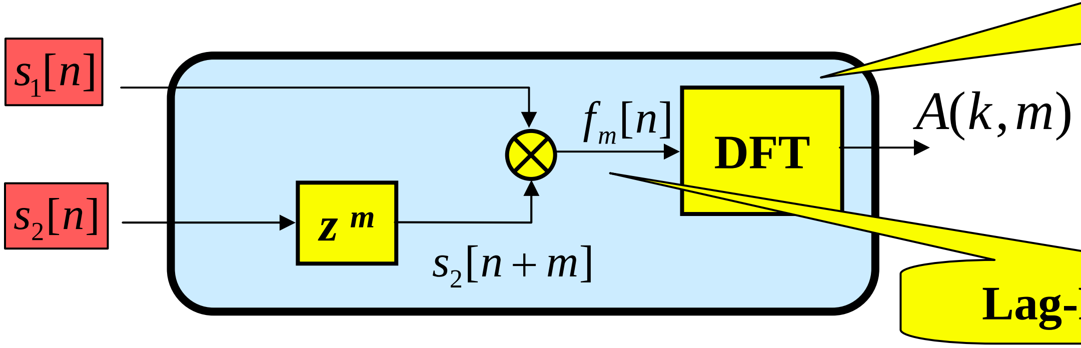
Recall C-T result: For each delay τ_m:

$$\boxed{A(\omega, \tau_m) = \mathscr{F}\{ f_{\tau_m}(t) \}}$$

Thus for D-T: For each delay index m:

$$\begin{aligned} A(k,m) &= \text{DFT}\{ f_m(n) \} \\ &= \text{DFT}\{ s_1[n] \overline{s_2[n+m]} \} \end{aligned}$$

So, computing the ambiguity function for *M* delay indices is nothing more than doing *M* *N*-pt. DFTs, one for each delay index

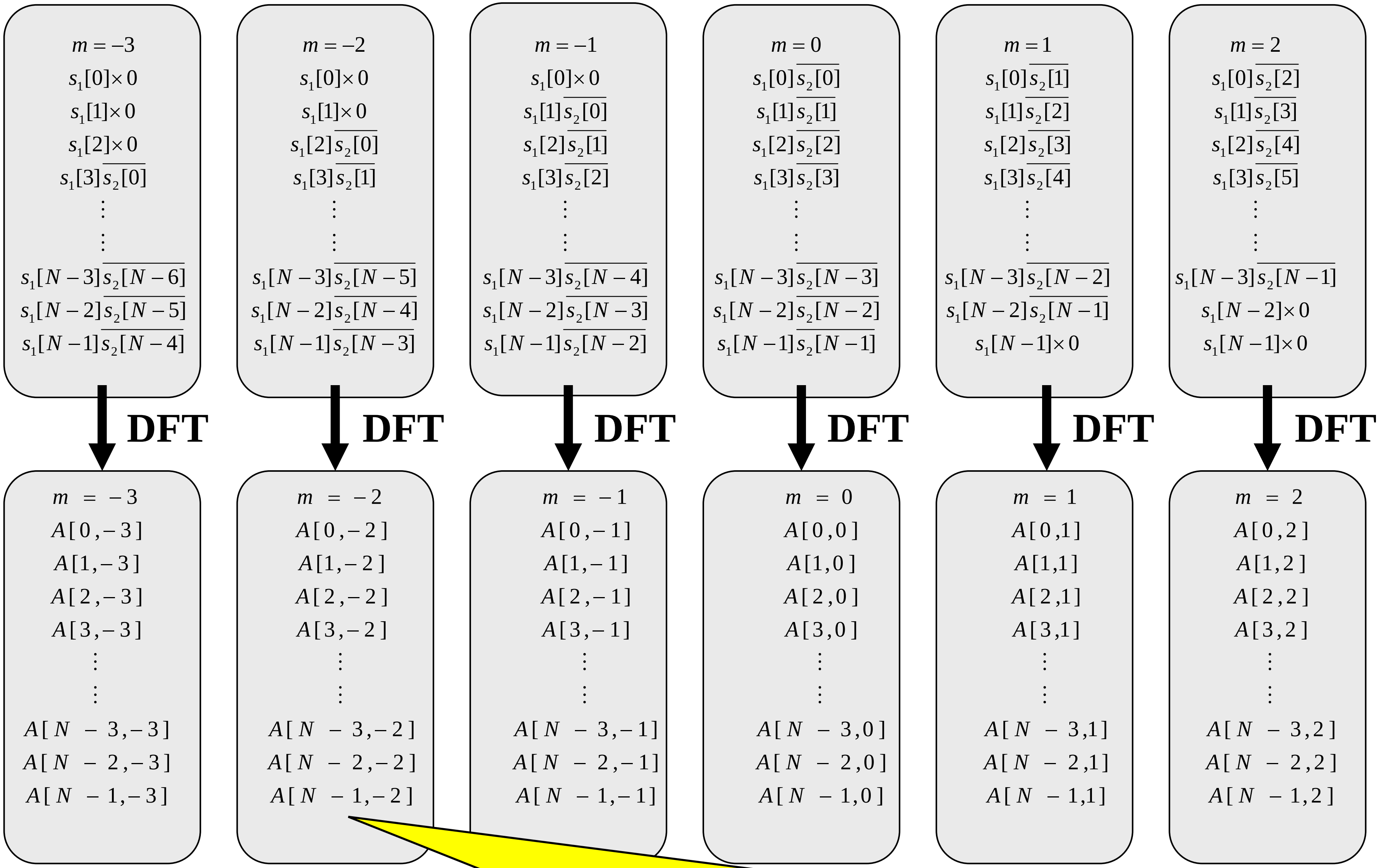


Use a Window for all the same reasons as frequency estimation

Lag-Product Signal

Each Column = Lag Product $s_1(n)s_2^*(n+m)$ for an *m* value

COMPUTING THE AMBIGUITY FUNCTION (cont.)



$m=-3$	$m=-2$	$m=-1$	$m=0$	$m=1$	$m=2$
$s_1[0]\times 0$	$s_1[0]\times 0$	$s_1[0]\times 0$	$s_1[0]\overline{s_2[0]}$	$s_1[0]\overline{s_2[1]}$	$s_1[0]\overline{s_2[2]}$
$s_1[1]\times 0$	$s_1[1]\times 0$	$s_1[1]\overline{s_2[0]}$	$s_1[1]\overline{s_2[1]}$	$s_1[1]\overline{s_2[2]}$	$s_1[1]\overline{s_2[3]}$
$s_1[2]\times 0$	$s_1[2]\overline{s_2[0]}$	$s_1[2]\overline{s_2[1]}$	$s_1[2]\overline{s_2[2]}$	$s_1[2]\overline{s_2[3]}$	$s_1[2]\overline{s_2[4]}$
$s_1[3]\overline{s_2[0]}$	$s_1[3]\overline{s_2[1]}$	$s_1[3]\overline{s_2[2]}$	$s_1[3]\overline{s_2[3]}$	$s_1[3]\overline{s_2[4]}$	$s_1[3]\overline{s_2[5]}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$s_1[N-3]\overline{s_2[N-6]}$	$s_1[N-3]\overline{s_2[N-5]}$	$s_1[N-3]\overline{s_2[N-4]}$	$s_1[N-3]\overline{s_2[N-3]}$	$s_1[N-3]\overline{s_2[N-2]}$	$s_1[N-3]\overline{s_2[N-1]}$
$s_1[N-2]\overline{s_2[N-5]}$	$s_1[N-2]\overline{s_2[N-4]}$	$s_1[N-2]\overline{s_2[N-3]}$	$s_1[N-2]\overline{s_2[N-2]}$	$s_1[N-2]\overline{s_2[N-1]}$	$s_1[N-2]\times 0$
$s_1[N-1]\overline{s_2[N-4]}$	$s_1[N-1]\overline{s_2[N-3]}$	$s_1[N-1]\overline{s_2[N-2]}$	$s_1[N-1]\overline{s_2[N-1]}$	$s_1[N-1]\times 0$	$s_1[N-1]\times 0$

$m=-3$	$m=-2$	$m=-1$	$m=0$	$m=1$	$m=2$
$A[0,-3]$	$A[0,-2]$	$A[0,-1]$	$A[0,0]$	$A[0,1]$	$A[0,2]$
$A[1,-3]$	$A[1,-2]$	$A[1,-1]$	$A[1,0]$	$A[1,1]$	$A[1,2]$
$A[2,-3]$	$A[2,-2]$	$A[2,-1]$	$A[2,0]$	$A[2,1]$	$A[2,2]$
$A[3,-3]$	$A[3,-2]$	$A[3,-1]$	$A[3,0]$	$A[3,1]$	$A[3,2]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A[N-3,-3]$	$A[N-3,-2]$	$A[N-3,-1]$	$A[N-3,0]$	$A[N-3,1]$	$A[N-3,2]$
$A[N-2,-3]$	$A[N-2,-2]$	$A[N-2,-1]$	$A[N-2,0]$	$A[N-2,1]$	$A[N-2,2]$
$A[N-1,-3]$	$A[N-1,-2]$	$A[N-1,-1]$	$A[N-1,0]$	$A[N-1,1]$	$A[N-1,2]$

Each Column = Ambiguity Function $A(k,m)$ for an *m* value

COMPUTING THE AMBIGUITY FUNCTION (cont.)

Problems

1. We end up computing as many doppler bins as there are signal samples – this gives too many doppler bins

2. This usually requires a VERY large FFT

3. The range of doppler bins is from –Fs/2 to Fs/2, which is a far wider range than necessary.

4. The doppler spacing is too coarse: could fix with zero padding, but the FFT is already TOO long

COMPUTING THE AMBIGUITY FUNCTION (cont.)

Example: Signal RF BW = 2 MHz; Center Freq = 10 GHz;
F_s = 2.4 MHz; Collect for T = 0.1 sec; Max Velocity, v = 300 m/s

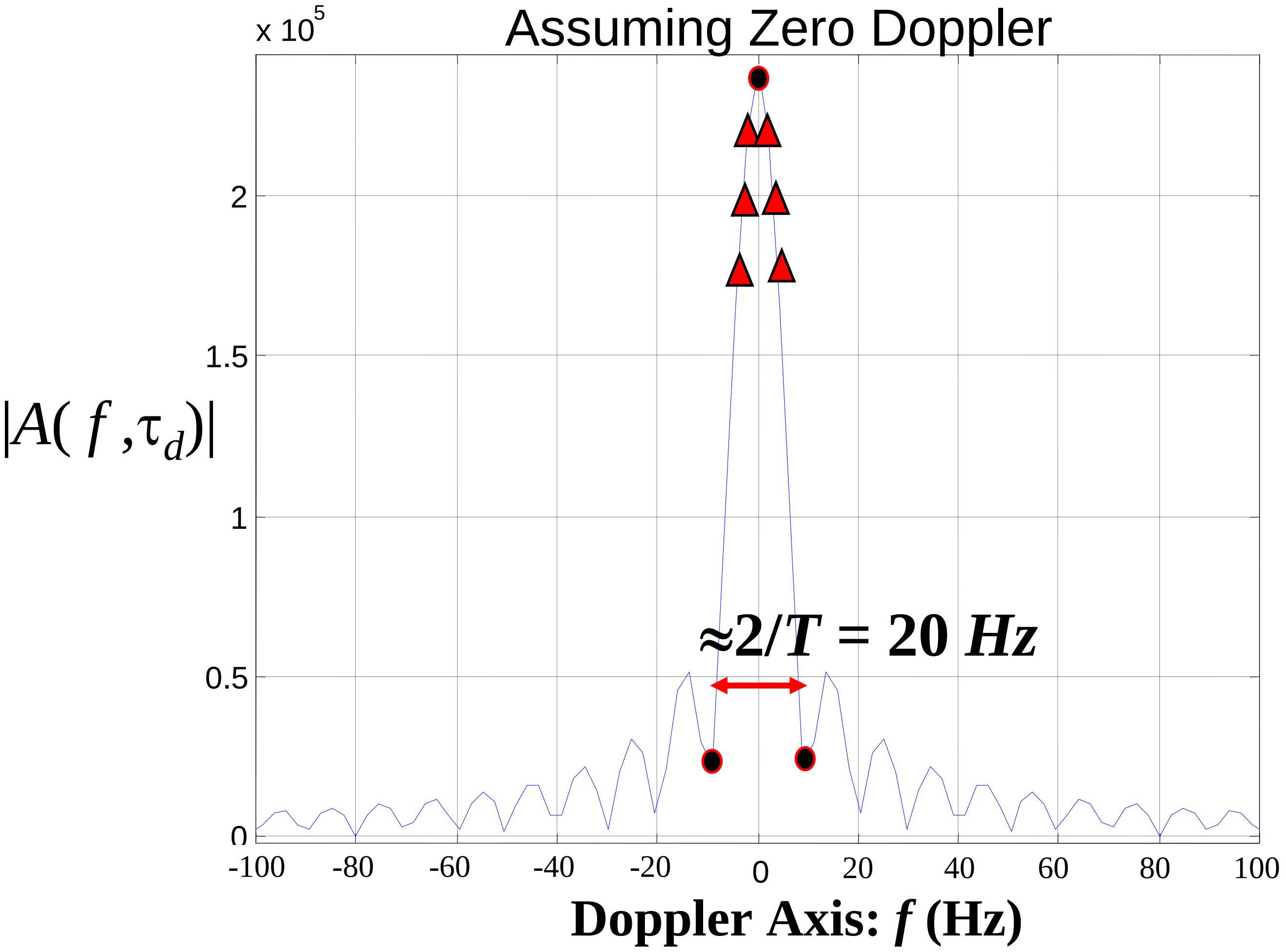
- DFT Frequency Range = $-F_s/2$ to $F_s/2$ = [–2.4/2 , 2.4/2] MHz
  ➔ **Freq Range = [–1.2 , 1.2] MHz**

- $|f_d| = f_c(|v|/c) \leq 10^{10} \times 300/(3 \times 10^8)$ = 10 kHz
  ➔ **max doppler = ±10 kHz** (Less Than 1% of Freq Range!!!)

- $N = F_s\, T$ = (2.4 MSPS)(0.1 sec) = 240,000 samples
  ➔ **# Samples = 240,000** ➔ **FFT size = 2^{18} = 262,144**

- DFT Spacing = F_s/FFT Size = $2.4 \times 10^6/262144$
  ➔ **doppler bin spacing = 9.2 Hz** (Need ≈ 1 Hz spacing)
  Could get that using zero-padding but that would make an 8× longer FFT having 2,097,152 points !!!!!!
  $(2.4 \times 10^6/(8 \times 262144) = 1.1$ Hz

COMPUTING THE AMBIGUITY FUNCTION (cont.)



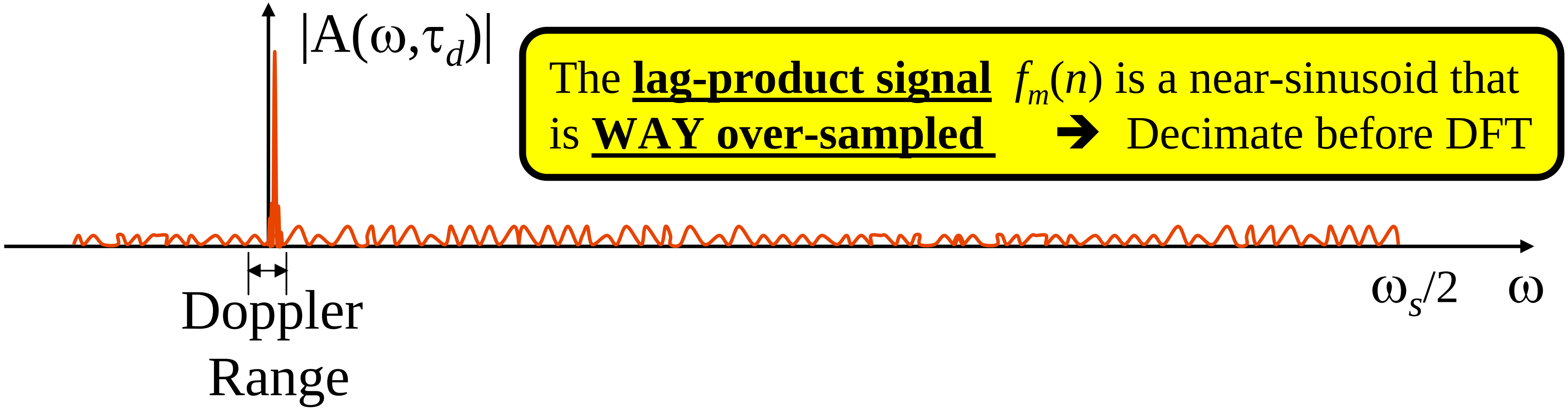
Assuming Zero Doppler

≈2/*T* = 20 *Hz*

Doppler Spacing of 9.2 Hz is not sufficient

Need about 1 Hz spacing to get 5 DFT points near peak

COMPUTING THE AMBIGUITY FUNCTION (cont.)



The **lag-product signal** $f_m(n)$ is a near-sinusoid that is **WAY over-sampled** ➔ Decimate before DFT

Original Freq Range = [–1.2 , 1.2] MHz

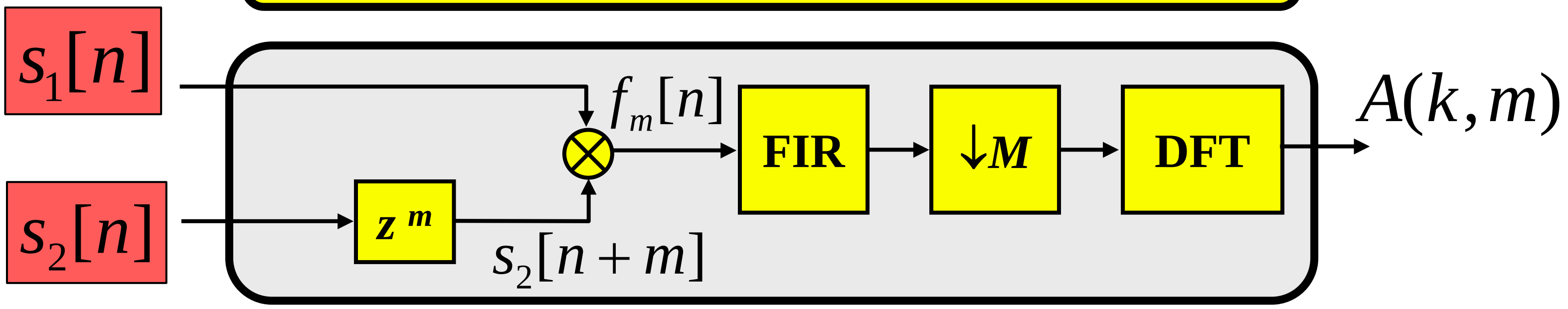
max doppler = 10 kHz ➔ Digital Freq: π(10kHz/1.2MHz)

➔ **Doppler Range = [–0.008π , 0.008π] rad/sample**

So we could filter $f_m(n)$ to [–0.01π , 0.01π]
and then decimate by M = 100 (100 = π/ 0.01π)
New Rate: Fs = 2.4MHz/100 = 24 kHz

Computing The Ambiguity Function (cont.)

Decimate before computing the DFT



New # Samples after Decimating by 100: N = 240,000/100 = 2400

➔ **New # Samples = 2400** ➔ **FFT size = 2^{12} = 4096**

DFT Spacing = New F_s/FFT Size = 24 × 10^3/4096 =5.9

➔ **doppler bin spacing = 5.9 Hz** (Still not 1 Hz!!)

But…Now CAN correct using zero-padding

➔ zero-pad to **16384 point FFT**:

spacing = 24 × 10^3/16384 = 1.5 Hz

Can live with that spacing

COMPUTING THE AMBIGUITY FUNCTION (cont.)

Decimation Filter Design

Could use standard design approach – but it is possible to use a very simple filter/decimation approach[1]. This approach gains efficiency at the cost of performance, but the performance has been found to be satisfactory in most cases.

A simple length-L FIR LPF is made by having all coefficients set to 1's:

$$h(n) = \begin{cases} 1, & n = 0, 1, 2, \ldots, L-1 \\ 0, & \text{otherwise} \end{cases}$$

The output is simply computed as the sum of L input samples:

$$\boxed{y(n) = \sum_{l=0}^{L-1} x(n-l)}$$

No Multiplies Needed

[1] See Seymour Stein, "Algorithms for Ambiguity Function Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, June 1981, pp. 588 – 599.

COMPUTING THE AMBIGUITY FUNCTION (cont.)

Now, for a given length L, what is the frequency response of this filter? Well, that is nothing more than the DTFT of a rectangular window of length *L*, which we have seen to be:

$$H_L(\theta) = \frac{\sin(\theta L/2)}{\sin(\theta/2)}$$

This has

- First nulls at θ = $\pm F_s/L$ Hz
- 3 dB points at θ = (0.89/2) F_s/L Hz
- Stopband height at first lobe is –13.5 dB
- Stopband rolls off at –6 dB/octave

See #2 on p. 167 of textbook

See Table in Harris's Paper

See #4 on p. 167 of textbook

See Table in Harris's Paper

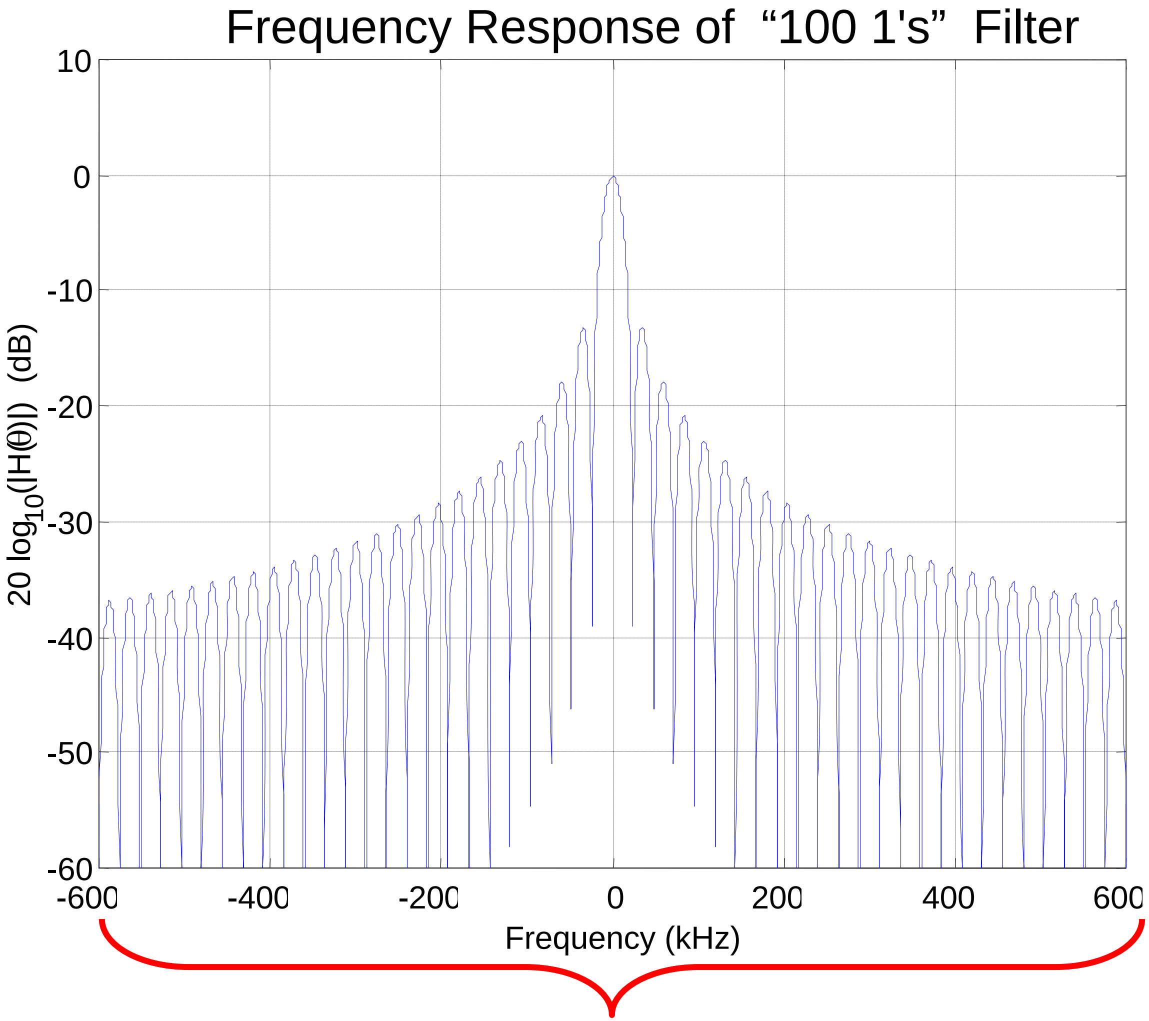
We'll **pick $L = M = 100$** to get a 3-dB passband to cover the doppler range:

$$(0.89/2)\ \underbrace{\underbrace{2.4 \times 10^6}_{\text{Pre-Dec } F_s} /100}_{\text{Post-Dec } F_s} = 10.7 \text{ kHz}$$

M = Dec Factor

COMPUTING THE AMBIGUITY FUNCTION (cont.)

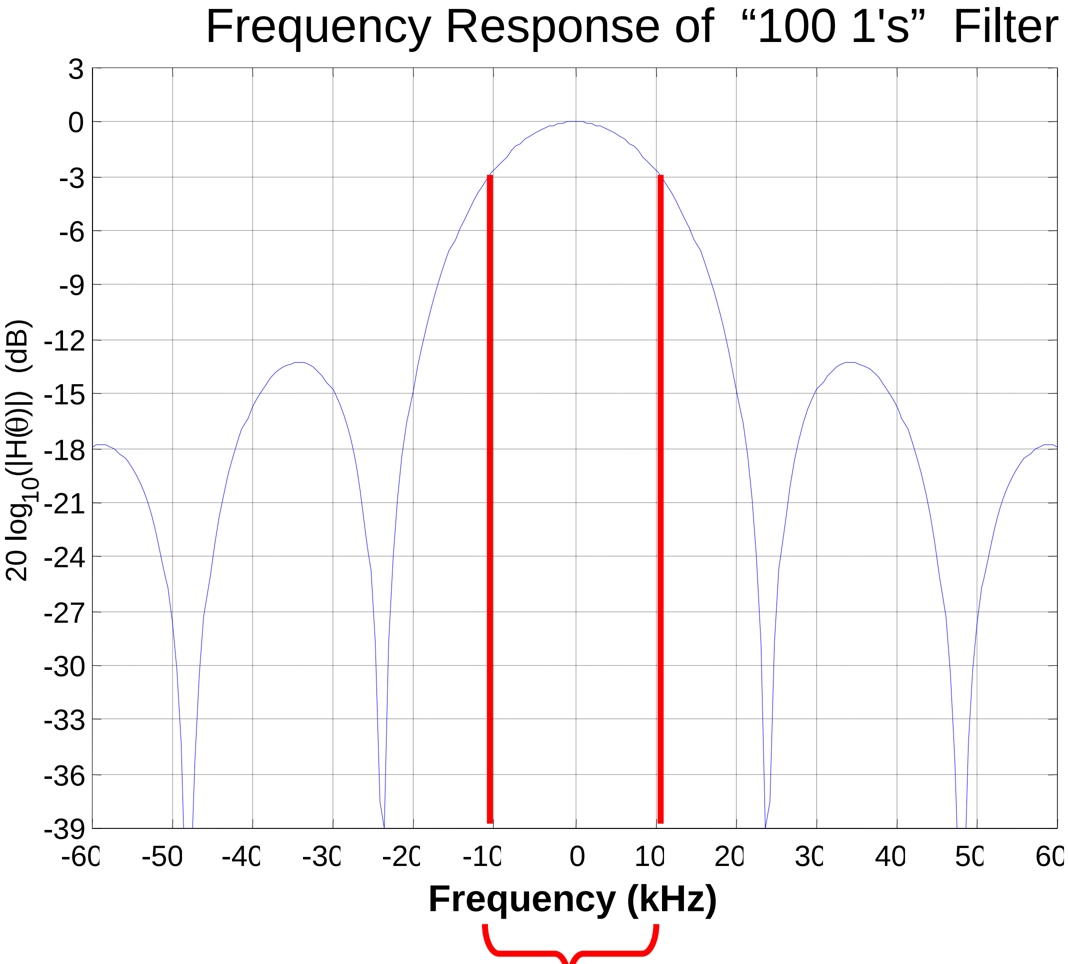
This looks like this for a length of L = 100



Shown over a very limited part of [-1.2 MHz, 1.2 MHz]

COMPUTING THE AMBIGUITY FUNCTION (cont.)

Zoom in to see passband better…



Passband Covers Doppler Range

COMPUTING THE AMBIGUITY FUNCTION (cont.)

Now we decimate by factor M = 100 and what happens?

We get the part of the signal that goes through the stopband aliased into the passband (see Eq. (12.14) in the book). To get a rough idea of what impact this has we can look at the filter's stopband response "aliased" back into the passband.

This corresponds to plotting the frequency shifted versions $H((\theta - 2\pi m)/M)$ for integer m on top of $H(\theta)$ – again, see Eq. (12.14) in the book.

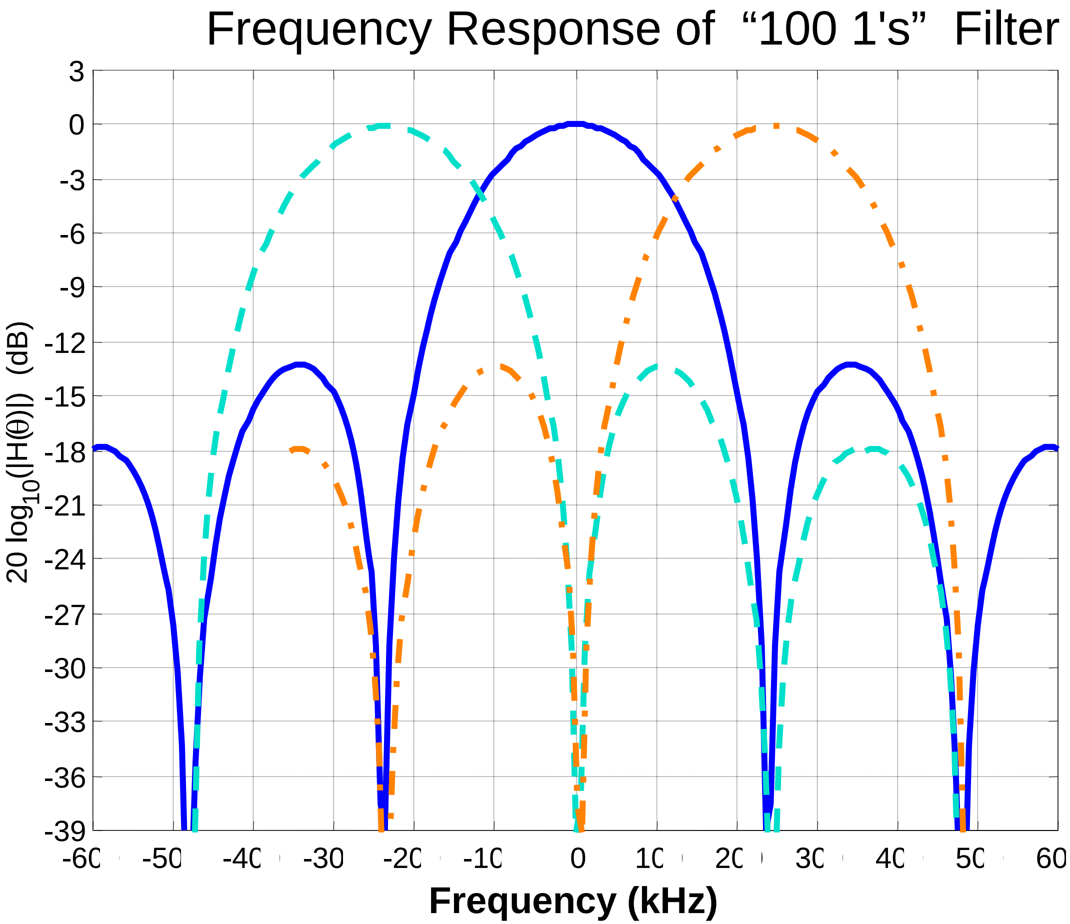
In Hz, these shifts are integer multiples of

$$F_s/M = 2.4\text{ MHz}/100 = 24\text{ kHz}$$

When $M = L$

When we do this we see that **This Particular** filter has a nice property that shifts the nulls to the center of the passband, thus minimizing the impact of aliasing, even though this filter is not really a very good lowpass filter!!

COMPUTING THE AMBIGUITY FUNCTION (cont.)



From this we see that in the middle of the doppler band we have very little impact from the aliasing, but at the edges of the doppler band there can be some severe aliasing. **Can we live with that?**

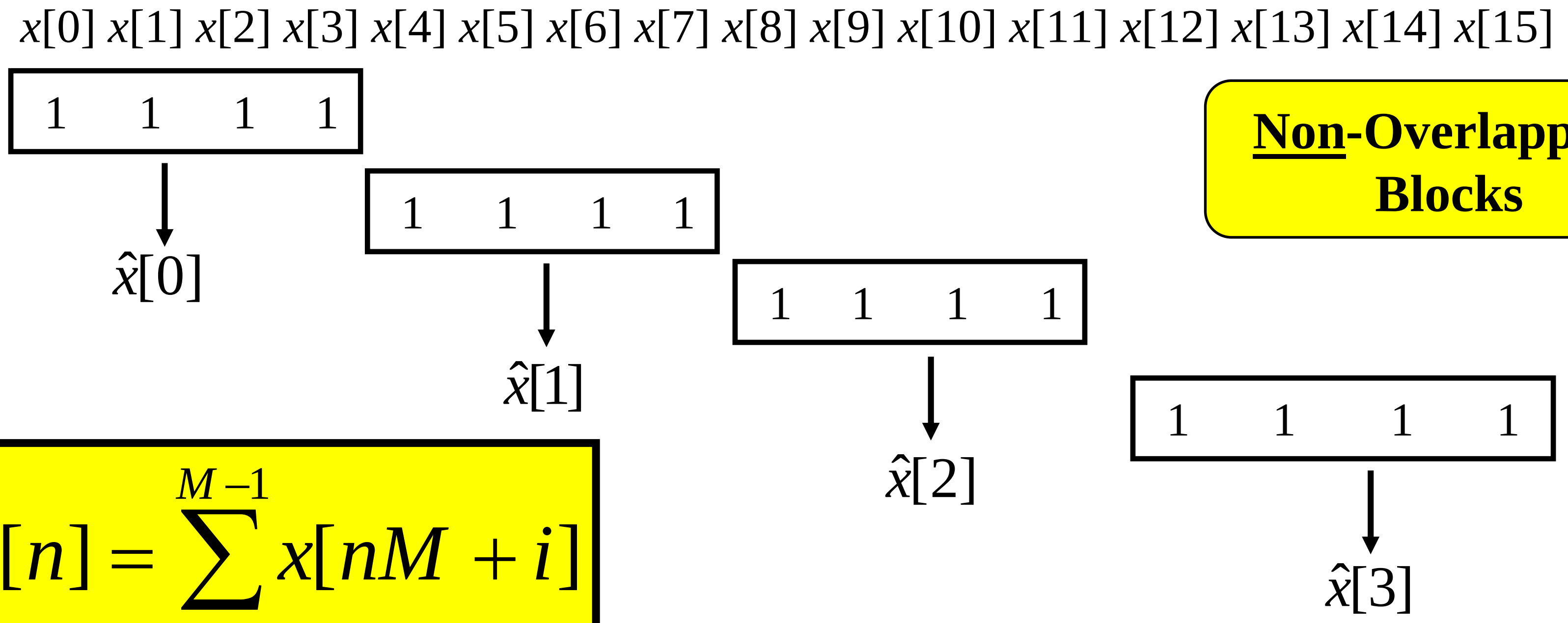
That is a design decision: we must trade computation vs. performance!!!

Computing The Ambiguity Function (cont.)

Now what impact does using this filter have on computing the ambiguity function? Recall how filtering is done for decimation: you move the filter coefficients ahead by *M* to get each decimated output. But for our case, the filter is length *M* so there is no overlap between consecutive placements of the coefficients:

For Illustration: Use ***M* = 4**

A Small *M* for Illustration!!



$x[0]$ $x[1]$ $x[2]$ $x[3]$ $x[4]$ $x[5]$ $x[6]$ $x[7]$ $x[8]$ $x[9]$ $x[10]$ $x[11]$ $x[12]$ $x[13]$ $x[14]$ $x[15]$

1 1 1 1 → $\hat{x}[0]$

1 1 1 1 → $\hat{x}[1]$

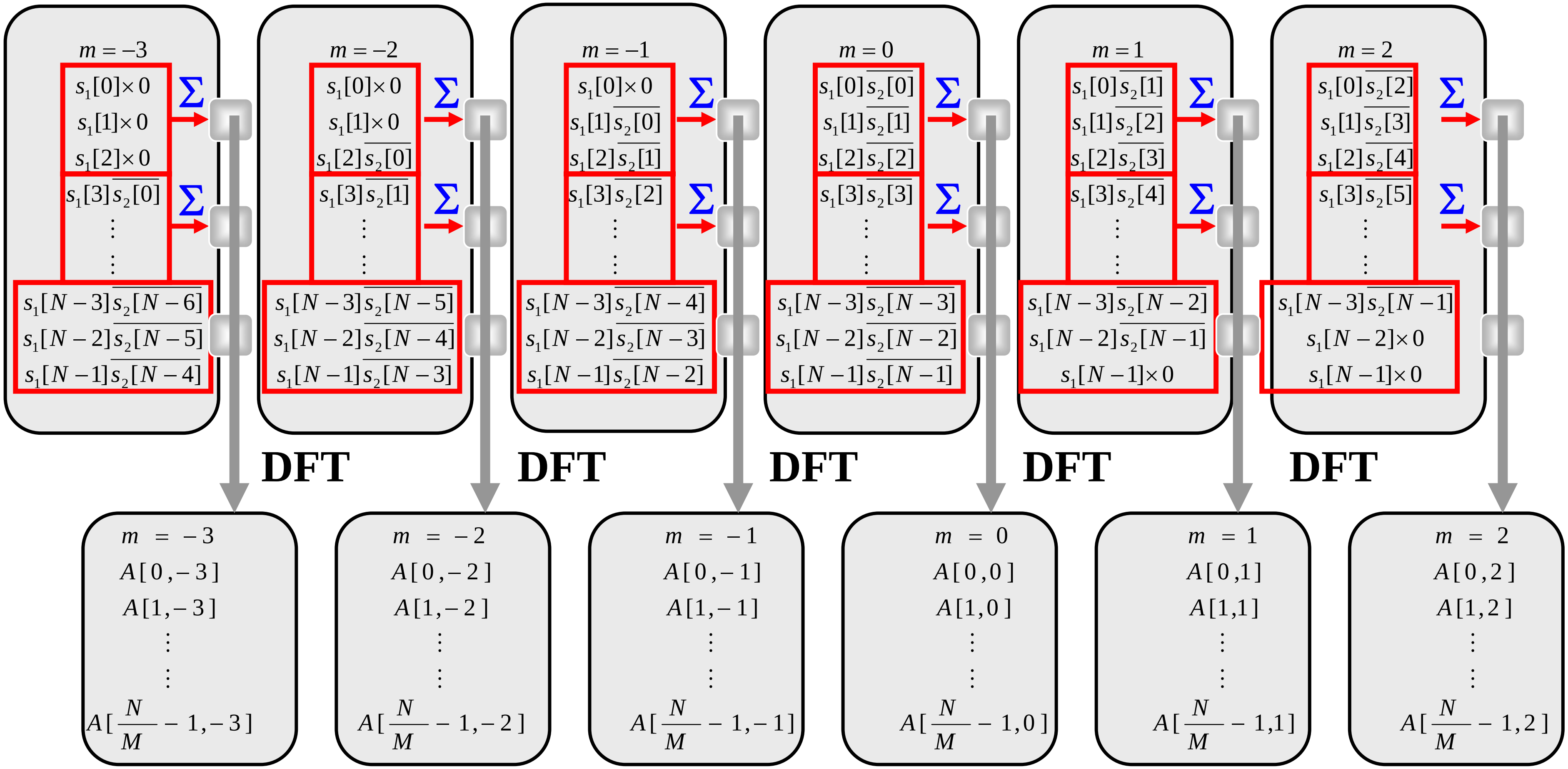
1 1 1 1 → $\hat{x}[2]$

1 1 1 1 → $\hat{x}[3]$

Non-Overlapping Blocks

$$\hat{x}[n] = \sum_{i=0}^{M-1} x[nM + i]$$

COMPUTING THE AMBIGUITY FUNCTION (cont.)



$m = -3$

$s_1[0]\times 0$, $s_1[1]\times 0$, $s_1[2]\times 0$, $s_1[3]\overline{s_2[0]}$, $\vdots$, $\vdots$, $s_1[N-3]\overline{s_2[N-6]}$, $s_1[N-2]\overline{s_2[N-5]}$, $s_1[N-1]\overline{s_2[N-4]}$

$m = -2$

$s_1[0]\times 0$, $s_1[1]\times 0$, $s_1[2]\overline{s_2[0]}$, $s_1[3]\overline{s_2[1]}$, $\vdots$, $\vdots$, $s_1[N-3]\overline{s_2[N-5]}$, $s_1[N-2]\overline{s_2[N-4]}$, $s_1[N-1]\overline{s_2[N-3]}$

$m = -1$

$s_1[0]\times 0$, $s_1[1]\overline{s_2[0]}$, $s_1[2]\overline{s_2[1]}$, $s_1[3]\overline{s_2[2]}$, $\vdots$, $\vdots$, $s_1[N-3]\overline{s_2[N-4]}$, $s_1[N-2]\overline{s_2[N-3]}$, $s_1[N-1]\overline{s_2[N-2]}$

$m = 0$

$s_1[0]\overline{s_2[0]}$, $s_1[1]\overline{s_2[1]}$, $s_1[2]\overline{s_2[2]}$, $s_1[3]\overline{s_2[3]}$, $\vdots$, $\vdots$, $s_1[N-3]\overline{s_2[N-3]}$, $s_1[N-2]\overline{s_2[N-2]}$, $s_1[N-1]\overline{s_2[N-1]}$

$m = 1$

$s_1[0]\overline{s_2[1]}$, $s_1[1]\overline{s_2[2]}$, $s_1[2]\overline{s_2[3]}$, $s_1[3]\overline{s_2[4]}$, $\vdots$, $\vdots$, $s_1[N-3]\overline{s_2[N-2]}$, $s_1[N-2]\overline{s_2[N-1]}$, $s_1[N-1]\times 0$

$m = 2$

$s_1[0]\overline{s_2[2]}$, $s_1[1]\overline{s_2[3]}$, $s_1[2]\overline{s_2[4]}$, $s_1[3]\overline{s_2[5]}$, $\vdots$, $\vdots$, $s_1[N-3]\overline{s_2[N-1]}$, $s_1[N-2]\times 0$, $s_1[N-1]\times 0$

$m = -3$: $A[0,-3]$, $A[1,-3]$, $\vdots$, $\vdots$, $A[\frac{N}{M}-1,-3]$

$m = -2$: $A[0,-2]$, $A[1,-2]$, $\vdots$, $\vdots$, $A[\frac{N}{M}-1,-2]$

$m = -1$: $A[0,-1]$, $A[1,-1]$, $\vdots$, $\vdots$, $A[\frac{N}{M}-1,-1]$

$m = 0$: $A[0,0]$, $A[1,0]$, $\vdots$, $\vdots$, $A[\frac{N}{M}-1,0]$

$m = 1$: $A[0,1]$, $A[1,1]$, $\vdots$, $\vdots$, $A[\frac{N}{M}-1,1]$

$m = 2$: $A[0,2]$, $A[1,2]$, $\vdots$, $\vdots$, $A[\frac{N}{M}-1,2]$

- Form matrix w/ columns of lag products
- Break into *M*-pt subblocks along columns
- Sum elements in each subblock
- Take DFT of each column of subblock sums

COMPUTING THE AMBIGUITY FUNCTION (cont.)

Without Decimation

$$A(k,m) = \mathrm{DFT}\{ f_m(n)\} = \sum_{n=0}^{N-1} f_m(n)\, e^{j2\pi kn/N}$$

Example:

Computation per Delay Bin is:

- **2,097,152 point FFT**

With Decimation

$$A(k,m) \approx \mathrm{DFT}\{ \hat{f}_m(n)\} = \sum_{n=0}^{(N/M)-1} \left[\sum_{i=0}^{M-1} f_m(nM+i) \right] e^{j2\pi kn/N}$$

Example:

Computation per Delay Bin is:

- **(N/M) M-pt sums:**
  - ▸ **2400 100-pt sums**
- **16,384 point FFT**

Using Eq. (5.23) and (5.24) for FFT Counts

Without Decimation

For Each Delay Bin:

- **# Real Multiplies = 79,691,780**
- **# Real Additions = 127,926,274**

With Decimation

For Each Delay Bin:

- **# Real Multiplies = 393,220**
- **# Real Additions = 1,135,362**

Decimation Reduces Counts to < 1% of “No Dec” Values