Demodulation/baseband approach to beamforming

As shown in (A.3) above, for a single reflector at \((r, \theta)\) the output signal from the \(n\)th transducer element is approximately

\[
v_n(t) \approx m_n(t) \cos(\omega_0(t - \tau^T - d_n/c) + \phi_n), \quad n = 1, \ldots, N,
\]

where the “slowly varying” signal component is

\[
m_n(t) \triangleq KR(r, \theta) \left(\frac{\cos \theta}{r}\right)^2 a(t - \tau^T - d_n/c) + \phi_n,
\]

where \(\tau^T\) denotes the transmit propagation delay (typically \(\tau^T = r/c\)). The signals \(v_n(t)\) are (somewhat) narrowband, centered at the carrier frequency \(f_0\). It is easier to manipulate these signals if they are first converted to baseband. This process is called demodulation, i.e., we shift each spectrum to be centered at DC, much like in AM radio.

The following figure shows a block diagram of the demodulation process.

We first demodulate each signal by multiplying by \(\cos\) and \(\sin\) at an approximate demodulation frequency \(\omega_0^\prime\):

\[
I_n'(t) \triangleq 2 v_n(t) \cos(\omega_0^\prime t) = m_n(t) \left[\cos(\Delta \omega t + \omega_0(\tau^T + d_n/c) - \phi_n) + \cos(\omega_0 + \omega_0^\prime) t - \omega_0(\tau^T + d_n/c) + \phi_n)\right],
\]

\[
Q_n'(t) \triangleq 2 v_n(t) \sin(\omega_0^\prime t) = m_n(t) \left[\sin(\Delta \omega t + \omega_0(\tau^T + d_n/c) - \phi_n) + \sin(\omega_0 + \omega_0^\prime) t - \omega_0(\tau^T + d_n/c) + \phi_n)\right],
\]

where \(\Delta \omega \triangleq \omega_0^\prime - \omega_0\), because

\[
2 \cos(x) \cos(y) = \cos(y - x) + \cos(y + x)
\]

\[
2 \cos(x) \sin(y) = \sin(y - x) + \sin(y + x).
\]

For generality we allow for the demodulation frequency \(f_0^\prime\) to differ (slightly) from the carrier frequency \(f_0\). \(I_n'(t)\) is called the in-phase component; \(Q_n'(t)\) is called the quadrature component.
We then low-pass filter \( I'_n(t) \) and \( Q'_n(t) \), removing the term at (roughly) twice the carrier frequency:

\[
I_n(t) \triangleq (I'_n \ast \text{LowPass})(t) = m_n(t) \cos(\Delta \omega t + \omega_0 (\tau_T + d_n/c) - \phi_n)
\]

\[
Q_n(t) \triangleq (Q'_n \ast \text{LowPass})(t) = m_n(t) \sin(\Delta \omega t + \omega_0 (\tau_T + d_n/c) - \phi_n).
\]

Now digitally sample the low-pass signals at times \( t_j = j \Delta t, j = 1, 2, \ldots \), which is easier than sampling \( v_n(t) \) because \( I_n(t) \) and \( Q_n(t) \) are baseband signals, so relatively slowly varying (because \( \omega'_0 \approx \omega_0 \)).

Demodulation can also be done using an FFT with an appropriate shift. In this case the shift \( \omega'_0 \) will be an integer multiple of the fundamental frequency associated with the DFT, which certainly is not likely to exactly match the desired \( \omega_0 \).

After digitally sampling, we can perform many operations (e.g., in a DSP chip), including the following complex operation of forming the complex base-band signal:

\[
b_n(t) \triangleq I_n(t) + jQ_n(t) = m_n(t) e^{i \Delta \omega \tau} e^{i \omega_0 (\tau_T + \frac{d_n}{c})} e^{-i \phi_n} = KR(r, \theta) \left( \frac{\cos \theta}{r} \right)^2 a \left( t - \tau_T - \frac{d_n}{c} \right) e^{i \Delta \omega \tau} e^{i \omega_0 (\tau_T + \frac{d_n}{c})} e^{-i \phi_n}.
\]

The following figure shows two RF signals \( \{v_n(t)\} \) and the corresponding baseband signals \( \{b_n(t)\} \) for an isolated reflector.

Once again we have arrived at a signal equation relating a (processed) measured signal to the quantity of interest, so we can contemplate image formation methods. Again assuming that the peak amplitude of \( a(t) \) is at \( t = 0 \), a natural generalization of the reflectivity estimate (A.4) would be:

\[
\hat{R}(r, \theta) = \text{gain}(r) \left| \frac{1}{N} \sum_{n=1}^{N} b_n(\tau_T + d(x_n; r, \theta)/c) \right| \\
\approx |R(r, \theta)| \left| \frac{1}{N} \sum_{n=1}^{N} e^{i \Delta \omega (\tau_T + d_n/c)} e^{i \omega_0 (\tau_T + d_n/c)} e^{-i \phi_n} \right| = |R(r, \theta)| \left| \frac{1}{N} \sum_{n=1}^{N} e^{i \omega'_0 d_n/c} e^{-i \phi_n} \right|^2,
\]

where

\[
\text{gain}(r) = \frac{r^2}{K |a(0)| \cos^2 \theta e^{2r \alpha(f_0)}}.
\]

Because \( \omega'_0 \) is large, we cannot ignore the \( e^{i \omega'_0 d_n/c} \) term, and the above method leads to destructive interference.

Instead, we must modify the image formation expression by applying appropriate phases to the baseband signals as we sum them. (This is why ultrasound arrays used for beam forming are called phased arrays.) Specifically, we use the following estimate:

\[
\hat{R}(r, \theta) = \text{gain}(r) \left| \frac{1}{N} \sum_{n=1}^{N} b_n(\tau_T(r) + d(x_n; r, \theta)/c) e^{-i \phi_n} \right|,
\]

for which

\[
\hat{R}(r, \theta) \approx |R(r, \theta)| \left| \frac{1}{N} \sum_{n=1}^{N} e^{-i \phi_n} \right|.
\]

Provided the \( \phi_n \) values are identical or very similar, this method will work well.
We needed both the $I$ and $Q$ signals because otherwise we would have $\cos(\cdot)$ instead of $e^{i\cdot}$ terms; we can “cancel out” $e^{i\cdot}$ using $e^{-i\cdot}$, but one cannot “cancel out” a $\cos(\cdot)$ term.

Using delays

The expression (A.5) is suitable for a “table lookup” implementation in MATLAB or with suitably large digital buffers. Alternatively, we can approximate $d_n$ by $r - x_n \sin \theta$, leading to the estimate

$$\hat{R}(r, \theta) = \text{gain}(r) \left| \frac{1}{N} \sum_{n=1}^{N} b_n \left( \tau^q(r) + \frac{r}{c} - \tau^R_n \right) e^{i\omega_0 \tau^R_n} \right|,$$

where the receive delays are defined by

$$\tau^R_n \triangleq \frac{x_n \sin \theta}{c}.$$

For this simple “real-time” implementation using fixed delays and a fixed phase $e^{i\omega_0 \tau^R_n}$ for each channel (independent of $r$, but dependent on the beam direction $\theta$), we have

$$\hat{R}(r, \theta) \approx |R(r, \theta)| \left| \frac{1}{N} \sum_{n=1}^{N} e^{-i\phi_n} e^{i\omega_0 (d_n/c + \tau^R_n)} \right|,$$

where $d_n/c + \tau^R_n \approx x_n^2 \cos^2 \theta/(2rc)$. We can ignore that term if $\omega_0(D/2)^2/(2rc) \ll 1$, i.e., $r \gg D^2/\lambda$. This is the far field; in the near field there will be some signal cancellation. Hence the preference for dynamic receive focusing, where the delays and phases are functions of $r$, as in (A.5).

In practice, reflectivity images are usually displayed on a logarithmic scale, so constants like $\log |a(0)|$ in the gain term just add a DC shift that is irrelevant when mapping to grayscale values from 0 to 255, so they can be ignored.
Image formation and sector scans

A conventional 2D sector scan ultrasound reflectivity image is formed by using transmit delays to steer the transmitted beam in a sequence of angular directions. (We will consider the appropriate angular spacing soon.) For each angle (or beam), a pulse is transmitted, and then echos are collected and sampled. Receive “delays” (or table lookups) are applied to estimate the reflectivity as a function of range \( r \) along the particular angular direction \( \theta \). Thus, the initial image that is formed is in polar coordinates, i.e., we have acquired

\[
R_{ij} = \hat{R}(r_i, \theta_j) = \hat{R}(r_i, \arcsin(s_j))
\]

for several range values \( r_i \) and angles \( \theta_j = \arcsin(s_j) \).

- The range values typically are uniformly spaced: \( r_i = i\Delta_r \).
- The angular samples typically are uniformly spaced in \( s = \sin \theta \), i.e., \( \theta_j = \arcsin(s_j) \), where \( s_j = j\Delta_s \).

For display, these polar coordinate samples must be converted to (samples on) Cartesian coordinates. One simple way to perform this conversion is bilinear interpolation (provided in MATLAB’s \texttt{interp2} function).

First a 1D example. If \( g_n \) are samples of a function \( g(x) \) at locations \( n\Delta_x \), then we can estimate \( g(x) \) from the samples by the following interpolation formula:

\[
\hat{g}(x) = \sum_n g_n \text{tri}
\left( \frac{x - n\Delta_x}{\Delta_x} \right),
\]

where the linear interpolator is

\[
\text{tri}(x) = \left\{ \begin{array}{ll} 
1 - |x|, & |x| \leq 1 \\
0, & \text{otherwise.}
\end{array} \right.
\]

Note that \( \hat{g}(x_n) = g(x_n) \), i.e., the interpolated function matches the original function at the sample points, but between the samples \( \hat{g}(x) \) is a linear approximation to \( g(x) \).

In 2D, we want to estimate \( R(x, z) = R(x, 0, z) \) from the samples \( \{R_{ij}\} \).

Because \( r = \sqrt{x^2 + z^2} \) and \( s = \sin \theta = x/r = x/\sqrt{x^2 + z^2} \), the bilinear interpolation estimate is:

\[
\hat{R}(x, z) = \sum_i \sum_j R_{ij} \text{tri}
\left( \frac{\sqrt{x^2 + z^2} - i\Delta_r}{\Delta_r} \right) \text{tri}
\left( \frac{x}{\sqrt{x^2 + z^2}} - j\Delta_s \right).
\]

For display we do not evaluate the above expression for all possible \( x \) and \( z \), but only for \( (x, z) \) pairs on a uniformly spaced lattice. MATLAB’s \texttt{interp2} function accepts as input the \( R_{ij} \) values at the known \( (r_i, s_j) \) coordinates as well as a list of the desired \( (x_n, z_m) \) locations, and returns \( \hat{R}(x_n, z_m) \) for those locations by evaluating the above expression.

Many of the \( (x, z) \) values on the desired lattice may lie outside of the area covered by the angular and radial sampling acquired, so the reflectivity at those locations is unknown. By default, the \texttt{interp2} function returns a NaN (not a number) for these locations, which the user should reassign prior to some constant, e.g., 0, prior to display. Newer version of \texttt{interp2} have optional arguments to control how the interpolator works in the exterior regions.

Because the data is uniformly sampled in \( r \) and \( s \), there is no need to use \texttt{griddata}.

Example. The following figure shows \( \hat{R}(r, \sin \theta) \) and \( \hat{R}(x, 0, z) \), i.e., before and after scan conversion.
Example. The following figure shows some of the received signals $v_n(t)$ for an object consisting of a few point reflectors. Also shown is the corresponding beamformed images $\hat{R}(r, \theta)$. One can see that deeper scatterers are blurred more.
We now analyze the PSF for a phased array with the following typical geometry, viewed from along $z$.

![Transducer Array](image)

**Array design parameters**
- Pulse amplitude function $a_0(t)$
- Carrier frequency $\omega_0 = 2\pi f_0$, or equivalently wavelength $\lambda$
- Shape of each transducer element $s_n(x, y)$, especially width $w$ for rectangular elements
- Number of elements $N$
- Distance between element centers $d$
- Transmit and receive delays $\tau^x_n(\theta)$, $\tau^y_n(\theta)$
- Transmit and receive apodization gains $\alpha^x_n$, $\alpha^y_n$
- For image formation: What should the angular sample spacing $\Delta \theta$ be? And How many $\theta$-lines to acquire?

We have already seen how to specify the transmit delays $\tau^x_n$ and receive delays $\tau^y_n$ from a purely geometric point of view. Now we examine how these and other parameters affect spatial resolution by determining the **receive** beam pattern $b(x, y, z; s^n)$, concentrating on the Fraunhofer approximation.

Sampling issues are also answered by examining the PSF.

We also briefly consider dynamic focusing using Fresnel approximation.

### PSF of a phased array via diffraction

(It is essential to consider diffraction because transducer element size about $\lambda/2$.)

Suppose we have $N$ (non-overlapping) transducers, each with its own input/output signal wire.
- $s_n(x, y)$: source amplitude distribution (aperture) of $n$th transducer
- $\alpha^x_n p_0(t - \tau^x_n)$: (delayed) pulse applied to $n$th transducer, where as before, $p_0(t) = a_0(t) e^{-i\omega_0 t}$

(It is possible to send different pulse shapes to each transducer: this is an advanced topic.)

Then, in piston mode, the pressure field at the transducer plane is:

$$u(P_0, t) = \sum_{n=1}^{N} \alpha^x_n s_n(x_0, y_0) p_0(t - \tau^x_n).$$

By diffraction formula (Goodman 3-33) (repeated below), the incident pressure at $P_1$ is:

$$u(P_1, t) = \frac{\cos \theta_0}{r_0} \frac{1}{2\pi c} \frac{d}{dt} u\left(P_0, t - \frac{r_0}{c}\right) \, dx_0 \, dy_0 \approx \frac{\cos \theta_1}{r_1} \int \frac{1}{2\pi c} \frac{d}{dt} u\left(P_0, t - \frac{r_0}{c}\right) \, dx_0 \, dy_0$$

$$= \frac{\cos \theta_0}{r_0} \frac{1}{2\pi c} \frac{d}{dt} \int \sum_{n=1}^{N} s_n(x_0, y_0) \alpha^x_n \frac{1}{2\pi c} \frac{d}{dt} p_0 \left(t - \tau^x_n - \frac{r_0}{c}\right) \, dx_0 \, dy_0$$

$$\approx -i\frac{\cos \theta_1}{\lambda r_1} \int \sum_{n=1}^{N} \alpha^x_n s_n(x_0, y_0) p_0 \left(t - \tau^x_n - \frac{r_0}{c}\right) \, dx_0 \, dy_0,$$

again using the **narrowband** approximation, so the **insonification** is superposition of contributions of $N$ sources.
By reciprocity, back at the transducer plane:

\[ u(P'_0, t) = \iiint R(P_1) \frac{\cos \theta_1}{\lambda \rho_1} \frac{1}{2\pi c} \frac{1}{d} u(P_1, t - \frac{r_{10}}{c}) dP_1 \]

\[ \approx \iiint R(P_1) \frac{\cos \theta_1}{\lambda \rho_1} \frac{1}{2\pi c} \frac{1}{d} \left[ \sum_{n=1}^{N} \alpha_n^T s_n(x_0, y_0) p_0 \left( t - \frac{\tau_n^T}{c} - \frac{r_{10} + r}{c} \right) dx_0 dy_0 \right] dP_1 \]

\[ \approx - e^{-i\omega_0 t} \iiint R(P_1) \frac{1}{\lambda} \left( \frac{\cos \theta_1}{\lambda \rho_1} \right)^2 e^{i\omega_0 \tau_1^T} \sum_{n=1}^{N} \alpha_n^T s_n(x_0, y_0) e^{i\omega_0 \tau_1^T} dx_0 dy_0 \]

\[ = - e^{-i\omega_0 t} \iiint R(P_1) \frac{1}{\tau_1^T} \frac{\cos \theta_1}{\lambda} e^{i\omega_0 \tau_1^T} b_{\text{Narrowband}}(x_1, y_1, z_1; s^T) a \left( t - \frac{2r_1}{c} \right) dP_1 \]

where as before for any 2D function \( s \) we define the (unitless) narrowband beam pattern as

\[ b_{\text{Narrowband}}(x_1, y_1, z_1; s) \triangleq \frac{\cos \theta_1}{\lambda} \int \int s(x_0, y_0) e^{ikr_0} dx_0 dy_0, \]

and we define the following effective transmit aperture function:

\[ s^T(x, y) \triangleq \sum_{n=1}^{N} \alpha_n^T e^{i\omega_0 \tau_1^T} s_n(x, y). \]

Ideally, the signal output from the \( n \)th transducer is (proportional to) the integral of pressure over its face:

\[ v_n(t) = \frac{1}{\lambda} \int \int s_n(x'_0, y'_0) u(P'_0, t) dx'_0 dy'_0, \]

so from above and simplifying (and accounting for transducer impulse response):

\[ v_n(t) = - e^{-i\omega_0 t} \iiint R(P_1) \frac{1}{\tau_1^T} b_{\text{Narrowband}}(x_1, y_1, z_1; s_n) b_{\text{Narrowband}}(x_1, y_1, z_1; s^T) a \left( t - \frac{2r_1}{c} \right) dP_1. \]

Ultimately beamforming is done by a (gain corrected) sum of the individual transducer output voltages:

\[ v_c(t) = \text{gain}(t) \cdot \sum_{n=1}^{N} \alpha_n^R v_n(t - \tau_n^R), \]

where the receive delays \( \tau_n^R \) may be same or different as transmit delays \( \tau_n^T \).

For the purposes of analyzing the beam pattern, we assume that the delays are small enough that

\[ a \left( t - \tau_n^R - \frac{2r_1}{c} \right) \approx a \left( t - \frac{2r_1}{c} \right). \]

Then the signal processing output is:

\[ v_c(t) \approx e^{-i\omega_0 t} \iiint R(x_1, y_1, z_1) b_{\text{Narrowband}}(x_1, y_1, z_1; s^R) b_{\text{Narrowband}}(x_1, y_1, z_1; s^T) a \left( t - \frac{2r_1}{c} \right) dx_1 dy_1 dz_1, \]

where we define the following effective receive aperture function:

\[ s^R(x_0, y_0) = \sum_{n=1}^{N} \alpha_n^R e^{i\omega_0 \tau_n^R} s_n(x_0, y_0). \]
If the delays have been chosen to steer the beam in a desired direction \( \theta \), then we will see soon that \( s^R \) and \( s^T \), which both depend on \( \theta \), will be peaked in the direction \( \theta \). Thus the natural estimate of reflectivity in that direction \( \theta \) is:

\[
\hat{R}(r \sin \theta, 0, r \cos \theta) = |v_c(2r/c)| \\
\approx \left| \iiint R(x_1, y_1, z_1) b_{\text{Narrowband}}(x_1, y_1, z_1; s^R(\theta)) b_{\text{Narrowband}}(x_1, y_1, z_1; s^T(\theta)) a\left(\frac{2(r - r_1)}{c}\right) \, dx_1 \, dy_1 \, dz_1 \right|.
\]

Again, depth resolution is determined by the pulse envelope \( a(t) \), whereas the lateral resolution is determined by beam patterns associated with the effective apertures \( s^R \) and \( s^T \).

From the above, we see that the \( b_{\text{Narrowband}}^2 \) terms in the PSF analysis for a single transducer were the product of one transmit term and one receive term. Here the **overall beam pattern** is the product

\[
b_{\text{Narrowband}}(x_1, y_1, z_1; s^R) b_{\text{Narrowband}}(x_1, y_1, z_1; s^T).
\]

Note: typically \( \tau_n^R \) is very small, but \( \omega_0 \tau_n^R \) is sufficiently large to change phase of \( \exp(i\omega_0 \tau_n^R) \).