Biomedical Ultrasound Fundamentals of Imaging and Micromachined Transducers

Course Instructor : Karla Mercado-Shekhar

Department of Electronic Systems Engineering

Indian Institute of Science, Bangalore

Lecture - 41

Hello, and welcome to today's lecture. We will be discussing how to use Field II for conducting ultrasound imaging simulations. To start, we'll talk about how phantoms are created in Field II. These phantoms are numerical models, where each object to be imaged is represented as a collection of point scatterers. Essentially, Field II models each object as a set of point scatterers, and if the object contains small scatterers, it will exhibit diffuse Rayleigh scattering characteristics.

Each target in the imaging field acts as a Rayleigh scatterer, and its position in space can be defined. Let's revisit the schematic of our transducer array to better understand this. In the z-direction, we have the axial direction, where the ultrasound beam moves away from the transducer. We also have the lateral direction, which scans from left to right, and the elevational direction, which lies outside the axial-lateral (z-x) plane. Each scatterer has a specific position within these coordinates, and we can also assign an amplitude or strength to it. If you're imaging a hyper-echoic region, the amplitude will be positive, and for a hypo-echoic region, the amplitude will be negative.

The process of image simulation follows steps similar to what we discussed in a previous lecture on Field II. First, you need to define the transducer type and create a handle for it. We have previously covered how to create a handle for a circular piston single-element transducer and a focused ultrasound linear array. The next step is to define the transducer's electromechanical impulse response, followed by the definition of the excitation signal that will be transmitted to the transducer elements. Afterward, you'll specify the scan locations where each point target will be imaged, and use a function to calculate the echoes from those targets.

To illustrate this, we'll go through an example where we compute an A-mode signal using a piston transducer. A-mode, or amplitude mode, generates a one-dimensional ultrasound echo signal. For this example, we define the transducer parameters: an unfocused circular piston transducer with a 6mm radius, a center frequency of 1.25 MHz, and a -6 dB bandwidth of 60%.

We are working with a bandwidth ratio and imaging targets located at axial depths of 3, 3.5, 4, and 4.5 centimeters. First, we need to create the transducer handle using the XDC_piston function. Here, we input the parameters of the transducer, such as the 6 mm radius, with all units following the MKS system (meters). We'll also divide the transducer element into smaller mathematical subdivisions, approximately 0.5 mm in size.

After creating the transducer handle, the next step is to define the impulse response. We'll set the sampling frequency to 100 MHz to satisfy the Nyquist criterion. The center frequency is 1.25 MHz, and the bandwidth ratio will be specified in terms of percentage. The bandwidth level (BW level) corresponds to the -6 dB full-width half maximum of the peak. To define the impulse response, we'll use the Gauss_pulse function from MATLAB's signal processing toolbox. We'll then send this impulse response to the transducer using the XDC_impulse command, with inputs being the transducer handle and the impulse response signal. These steps have been covered previously, so we'll move on.

Next, we use the XDC_excitation command to excite the transducer. After excitation, we define the positions of the point targets at depths of 3, 3.5, 4, and 4.5 cm. To set the scatterer positions, we create a vector where each row represents the x, y, and z coordinates of a scatterer. Each subsequent row will define the location of another scatterer. In this case, we'll assign all point targets an amplitude of one using the ones function. If you want to scale the amplitudes, you can multiply the vector by a desired amplitude factor.

Finally, we calculate the echoes from the point targets using the calc_scat command. The inputs for this function are the transducer (used for both transmitting and receiving), the positions of the point targets, and their assigned amplitudes.

The calc_scat command outputs the echo signals at each location, as well as the start time. The start time is crucial for positioning the scatterer echo within the signal. After obtaining the echo signals, we typically compute the envelope of the signal to map the brightness of the echo. This is done using the absolute value of the Hilbert transform.

Next, we create a time axis vector and use the range equation to calculate the depth corresponding to each time value. The range equation states that the depth equals the speed of sound multiplied by the echo arrival time, divided by two. The factor of two accounts for the pulse-echo system, where the sound travels twice the distance (to the scatterer and back).

Here, we convert the speed of sound from 1540 meters per second to 1540 × 100 to obtain centimeters per second. After creating the time and depth vectors, we can plot the RF signal and its envelope as a function of depth.

In the output, the top plot shows the RF echo signal, with amplitude plotted against time in microseconds. This plot reveals the times at which the echoes from the four point targets appear in the A-mode signal. The envelope of the signal, calculated using the Hilbert transform, is plotted according to the depth at which each point target is located. Peaks appear at 3, 3.5, 4, and 4.5 centimeters, corresponding to the previously defined scatterer locations.

To create a B-mode image of point targets, we need to define the positions in two dimensions. For this example, we simulate three point targets located on-axis at axial depths of 4, 5, and 6 centimeters.

Here's a schematic of the simulation: our field of view spans from 3 to 7 centimeters, with scatterers placed at 4, 5, and 6 centimeters in the axial direction. The lateral positions are set along the center of the beam. The amplitudes for all point targets are set to 100.

Let's now proceed to create a transducer. We'll be using a similar setup: an unfocused circular single-element piston transducer. In this case, we simulate a transducer with a 6 mm radius, a center frequency of 2 MHz, and a negative 6 dB bandwidth ratio of 70%. As before, we create the transducer handle, inputting the transducer's specifications into the code.

Next, we set the impulse response, updating the bandwidth ratio to 0.7, and the center frequency to 2 MHz. The rest of the code remains the same. In this example, we want to excite the transducer using a two-cycle sine wave, also known as a sine burst. We define the sine wave by specifying two cycles, and then create a time vector to plot the excitation pulse. The sine function's parameters are set accordingly, and we use the XDC_excitation command to excite the transducer with this pulse.

When it comes to B-mode imaging using a single-element transducer, typically, you acquire one echo signal or RF line at a time. To create a B-mode image, you mechanically scan the transducer across the 2D region of interest. To simulate this mechanical scanning, let's assume we want to scan in the lateral direction from -2 cm to +2 cm, stepping through these positions and calculating the echo at each point. The calc_scat command is used to compute the echo for each point.

In MATLAB, this mechanical scanning is coded by holding the beam stable at one location while shifting the positions of the scatterers. We define a parameter called shifted_points, which simulates this behavior. Unlike for a linear array transducer, in this case, the beam remains fixed while the scatterer positions shift. We define the lateral scan range from -2 cm to +2 cm and loop through each of these lateral positions. At each step, we compute the echo signal using calc_scat, just as we did previously.

Since each echo line may have a different start time, there's a method to align the echo lines in your image to a common start time. You can achieve this by using code that accounts for the computed start position, the minimum start time, the sampling frequency, and the RF data. The patch of code provided allows you to adjust these variables and align the echo lines appropriately.

Once you have the echoes from each scatterer, the next step is to compute the signal envelope using the Hilbert transform. The image is typically displayed as a grayscale image on a decibel (dB) scale. To achieve this, you normalize the echo amplitude by dividing by the maximum value of the envelope, then apply the formula 20 * log10 to convert the image into a log scale.

Finally, the B-mode image is a function of location. The z-axis values are computed using the range equation, allowing you to map each point in space to the z-axial direction. The image is plotted using the imagesc function, with a dynamic range of 40 dB applied to the plot for proper visualization.

Here's how the output appears: you can clearly see the three targets. One is positioned near 4 cm, another near 5 cm, and the third near 6 cm. The image is plotted with a 40 dB dynamic range, which is a common display setting for ultrasound scanners, typically ranging between 40 and 60 dB. The z-axis represents the axial direction, and the x-axis represents the lateral direction.

Now, if you want to simulate a B-mode image of a phantom containing numerous scatterers, like what you would encounter when imaging actual tissue, you'll be dealing with what are known as diffuse scatterers. Tissue consists of many scatterers, which are often modeled as Rayleigh scatterers. We refer to this as a "diffuse phantom," representing a scattering object with a random distribution of point targets across the region of interest.

Here, we will simulate something like a cyst, which appears as a circular anechoic region in the image. In ultrasound imaging, a cyst is hypoechoic compared to the surrounding tissue, meaning it reflects fewer echoes. So, we give the surrounding tissue a higher amplitude while creating a cyst region with a lower amplitude.

To create a diffuse phantom, let's assume we have 1,000 scatterers randomly distributed in the phantom. We define the x, z, and y coordinates for these scatterers. The x and z coordinates correspond to the plane we are imaging, while the y coordinate is set to zero because we assume the phantom exists only in the x-z plane (the elevational direction, or y-axis, is ignored for simplicity).

We input these coordinates into a matrix that contains the positions of the diffuse scatterers. Next, we assign an amplitude for each scatterer, setting an initial amplitude of 1. However, for the scatterers located within a defined cyst region, we set their amplitudes to zero, simulating the anechoic region. We determine which scatterers fall within this one-centimeter diameter cyst and adjust their amplitudes accordingly.

Once the diffuse phantom is generated, we perform mechanical scanning with the transducer. In this case, we scan laterally from -2 cm to +2 cm in 0.1 cm steps. As before, we define the positions of the beam for each scan line, shifting the scatterer positions while keeping the beam location fixed. This simulates the mechanical scanning environment.

Using the calc_scat function, we calculate the echo lines for each lateral position, obtaining the RF data. Afterward, we compute the signal envelope, display the image on a grayscale decibel scale, and use the range equation to compute the axial distance vector, or z-axis, which represents the depth.

This process is similar to the earlier example with the three-point targets. The output shows a diffuse phantom, with the center of the cyst located around the 6 cm axial depth. The image is displayed in a 40 dB dynamic range. Notice that individual scatterers are not visible because the scattered waves from each point target interfere with one another, either constructively or destructively, producing the characteristic speckle pattern observed in typical B-mode images.

The final result looks like this: a diffuse phantom with a clear cyst in the middle. The speckle pattern simulates what you would see in real ultrasound imaging. This example demonstrates how single-element transducers can mechanically scan and create B-mode images of different types of phantoms.

Next, we'll move on to a more commonly used approach in ultrasound imaging—linear or phased arrays. We'll set up the process for simulating a typical linear or phased array imaging scenario.

When setting up a transducer handle for array transducers, it's important to account for the distinct behaviors of arrays during transmission and reception. For example, a transducer may have a fixed transmit focus while using dynamic receive focusing, or it might feature a fixed transmit apodization with dynamic receive apodization. These configurations are commonly found in B-mode imaging with linear arrays. Field II allows for the simulation of arrays as separate entities for transmission and reception, enabling the creation of configurations such as a fixed transmit focus combined with dynamic receive focusing.

In the following example, a focused linear array transducer with a center frequency of 7 MHz and a -6 dB bandwidth of 70% is simulated. The transducer consists of 128 elements, an f-number of 6.5, and a natural focus at a 4 cm axial depth. Both transmission and reception use fixed electronic focusing at this depth. To begin, the transducer parameters are defined, including the center frequency of 7 MHz and the 70% bandwidth. The 128-element array is specified, and the elevation focus is set, which is a fixed characteristic of the transducer's design and does not change during imaging.

Key dimensions, such as element width, height, and kerf (the space between elements), are established. The kerf fraction, representing the ratio of the kerf to the element width, can be adjusted based on the transducer design. The pitch, which is the sum of the element width and kerf, is essential for defining the spacing between elements in the array. The speed of sound and sampling frequency, typically predefined in Field II, are explicitly defined to ensure their availability in the workspace for later computations. These parameters are critical for determining time delays and focal points.

To improve simulation accuracy, each element in the transducer is divided into smaller mathematical subdivisions. This finer division enhances the modeling of the acoustic field by allowing more precise control over the distribution of sound waves from each element. Once these parameters are set, further properties for transmission and reception can be defined, enabling advanced transducer configurations such as dynamic receive focusing and fixed transmit apodization for more accurate ultrasound imaging simulations.

In this part of the process, we are ensuring proper element subdivision in the code. The rule of thumb for accurate simulation is to subdivide the transducer elements into smaller mathematical elements, each approximately half the size of the wavelength. This ensures that all scatterers in the simulation lie in the far field of these smaller elements. By dividing the transducer elements into smaller subdivisions, the wavefronts from each element combine according to Huygens' principle. Ensuring the subdivisions are half a wavelength in size guarantees that the waves are accurately simulated within the far-field approximation, which Field II relies on for optimal performance. If the size criterion is not met, the simulation may introduce errors. This recommendation can be coded automatically in the simulation, as shown in the two lines provided.

Next, we proceed to create the transmit and receive arrays using the XDC_focused_array function. The inputs for this function include the number of elements, element width, height, kerf (space between elements), elevation focus, and the number of subdivisions that we previously defined. In this example, the transmit and receive codes are quite similar because we are assuming fixed transmit and receive focuses.

After setting up the transducer, we define the impulse response, which follows the same approach discussed earlier. We then set the positions of the point targets, with six targets spaced one centimeter apart, each having an amplitude of 100. You can adjust the amplitude as needed based on your simulation requirements.

For obtaining echoes from the point targets, we simulate the beam scanning. Unlike the single-element transducer case, where we moved the scatterer positions, here, we move the beam across the transducer's aperture in the lateral direction. This simulates what happens in real linear array imaging, where a small group of elements focuses on a specific region before moving to another lateral position. Thus, the scatterers remain stationary, but the beam is moved.

A key command for this simulation is XDC_center_focus, which defines the starting point of the beam on the transducer aperture. This function helps to simulate the lateral beam movement across the region of interest during imaging.

In this process, the XDC_focus command is used to define the location of the beam focus. Here, we set the electronic focus to be at 3 centimeters, while scanning from -2 to 2 centimeters laterally.

For each lateral position, both transmit and receive focusing are implemented. First, the center of the focus is defined using the XDC_center_focus command for both the transmit and receive arrays. After that, XDC_focus is executed. It's crucial to follow this sequence, as placing the XDC_center_focus command after XDC_focus will result in an error.

Next, the Calc_Scat command is employed to gather the echoes from all locations within the imaging field. Once the echoes are collected, they are aligned to have a common start time. The envelope of the signal is then computed, and the B-mode image is displayed on a decibel scale. The resulting image shows a well-defined point target at the 3-centimeter focal depth, where the beam is concentrated. At other depths, the lateral spread of the signal is broader, which is due to the beam focusing primarily at the 3-centimeter depth. Consequently, the point spread function becomes wider at locations outside this focus.

In cases where multiple focal depths are used or dynamic receive focusing is applied, the procedure is adjusted. Using the same transducer and phantom (with six point targets spaced 1 centimeter apart along the axis), we now introduce two transmit focal points at 2 and 4 centimeters. These focal zones are defined in the code, and start times for the focal zones are computed. Dynamic receive focusing is achieved by adjusting the focus over time, so that all on-axis points remain in focus. The XDC_dynamic_focus command is used for this, taking inputs such as the transducer handle, the time at which dynamic focusing becomes valid, and the angles for focusing in both the z-x and z-y planes.

The code for dynamic focusing involves scanning through the lateral locations in the region of interest. The center focus and focal depths for the transmit side (at 2 and 4 centimeters) are defined first, as dynamic focusing is not applied to the transmit phase in this example. Then, for the receive side, dynamic focusing is applied. For each lateral location, the XDC_center_focus command is first used, followed by the dynamic focusing procedure to adjust the receive focus as time progresses, ensuring all points are in focus.

In this example, we use the XDC_dynamic_focus command to input the receive transducer handle. We then set the start time for dynamic focusing to begin near the transducer. Since we're working with a linear array, which cannot steer the beam, the steering angle is set to zero. However, if we were using a phased array, the steering angle would be adjustable based on the phased array's capabilities. After setting these parameters, we calculate the scatterer echoes at each position in the imaging field, as in the previous examples.

The remaining code follows the same structure as before. The resulting image reveals the point targets, particularly at 2 and 4 centimeters, where we set the transmit focus. These point targets appear much finer compared to those at other locations, showing the enhanced resolution at the transmit focal depths. Compared to the previous example, where we had a fixed receive focus at 3 centimeters, you can observe that dynamic receive focusing results in skinnier, more refined point echoes in the lateral direction. This improved lateral resolution is due to focusing at each point along the axis, enhancing the point spread function. As discussed in earlier lectures, dynamic receive focusing significantly enhances image resolution.

This demonstration highlights how to simulate dynamic receive focusing using Field II. Throughout the lecture, we covered how to define numerical phantoms in Field II, including the simulation of point targets and diffuse scattering phantoms. The Field II website offers many examples and resources, including setting scatterer positions to mimic anatomical structures in the tissue being simulated.

Additionally, we explored how to simulate an A-mode scan using a single-element piston transducer and performed mechanical scanning using the same transducer. We also covered linear array imaging, including multiple transmit foci and dynamic receive focusing. Beyond this, you can use Field II to simulate phased arrays, concave transducers, and more.

I highly recommend referring to the Field II website, as it provides a comprehensive user guide with examples and scripts to help you simulate different types of anatomy. This concludes our lecture, and I look forward to seeing you in the next class. Thank you.