Digital Signal Processing for Every Application
Digital Signal Processing is Everywhere

- High Volume Production Test
- Structural Health Monitoring
- WiMAX, WLAN, GPS
- Space Exploration
- Large Physics Applications
- Avionics Applications
- Digital Signal Processing is Everywhere

ni.com
Case Study in Real-Time Signal Processing
Structural Health Monitoring for Donghai Bridge in China

• Connects Shanghai and the Yangshan deep-water port
• One of the world’s longest cross-sea bridges.
Applying mRSSI to Donghai Bridge

14 PXI systems for data acquisition and control

- Over 500 sensors including accelerometers, Fiber Bragg grating sensors, and anemoscopes
- GPS for system synchronization and time-stamping

LabVIEW for mRSSI algorithm implementation
Tools for Domain-Specific Analysis

- Sound & Vibration Toolkit
- Vision Development Module
- Robotics Module
- Biomedical Toolkit
- Automotive Diagnostic Command Set
- ECU Measurement and Calibration Toolkit

- NI Modulation Toolkit
- Measurement Suite for Fixed WiMAX
- WLAN Measurement Suite
- GPS Simulation Toolkit
What is Digital Signal Processing (DSP)?

1. Ensure signals are correctly digitized

2. Build the right signal processing algorithm
   - Choose the right FFT
   - Digital Signal Processing:
     - The mathematical manipulation of signals after they’ve been converted to a digital form
       - Offline
       - Software
       - Inline

3. Put your analysis in the right place
Sampling Your Signal

1. Ensure your sampled (digital) signal sufficiently represents the analog signal
2. Only take the parts of your signal that you need
   1. Prevent buffer overflow
   2. Reduce burden on on-board memory
Correctly Sampling Your Signal

1. Ensure your sampled (digital) signal is a true representation of your original analog signal

Major components of the analog front end on a high-speed digitizer
Bandwidth

Frequency at which the amplitude of the input signal, passed through the analog front end, is attenuated to 70.7% of its original amplitude

Also known as the -3 dB bandwidth or 3dB point
Typical Bandwidth of an Oscilloscope

Figure 4. NI 5162 50 Ω Frequency Response, 1 V_{pk-pk}, 5 GS/s, Characteristic
Limited Bandwidth Impacts Measurement

Digitizer bandwidth should be 3x to 5x the signal-bandwidth when looking at non sine-wave signals
Sampling Rate

The rate at which a digitizer’s ADC converts the input signal to digital data, after the signal has passed through the analog input path.
**Aliasing**

**Alias**  A false lower frequency component that appears in sampled data acquired at too low a sampling rate.
Shannon-Nyquist theorem

A signal must be sampled at a rate greater than twice the highest frequency component of the signal to accurately reconstruct the waveform.

This is a special case of the theorem that applies to signals from 0 Hz - $F_s/2$ Hz.

Signal of interest in spectrum view

Sampling Frequency

Nyquist Frequency ($F_s/2$)
More generic version of Shannon-Nyquist

A signal must be sampled as least twice as fast as the signal bandwidth.

Signal must be band limited. If not, use a bandpass filter.

“Undersampling”

Sample twice as fast as this bandwidth requires.
Correctly Sampling Your Signal

1. Prevent buffer overflow
2. Reduce burden on on-board memory

Software Trigger

1. Initialize
   Opens a session

2. Configure Trigger Software
   Configures your application

3. Initiate Acquisition
   Begins acquiring pretrigger data

4. Send Software Trigger Edge
   Sends a command to trigger the digitizer

5. Fetch
   Polls the digitizer until the acquisition is complete and retrieves the data

6. Close
   Closes the session

Edge Trigger

Trigger Level

Negative Edge Trigger
Positive Edge Trigger
Available Triggers*

**Software**
Software command starts the acquisition of posttrigger data

**Edge**
A signal crosses a trigger threshold that you specify

**Windowed**
A signal either enters or leaves a window you specify

**Hysteresis**
Removes incorrect trigger behavior as a result of noise

**Digital**
Occurs on either a rising or falling edge of a digital signal
What is Digital Signal Processing (DSP)?

1. Ensure signals are correctly digitized

2. Build the right signal processing algorithm
   - Choose the right FFT
   - Select the best window

3. Put your analysis in the right place
   - Offline
   - Software
   - Inline
Frequency Domain Analysis

Fourier’s Theorem  Any waveform in the time domain can be represented by the weighted sum of sines and cosines
Frequency Spacing and Symmetry of the FFT

1D FFT is defined as:

\[ Y_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N} \quad \forall n = 0, 1, 2, \ldots, N - 1 \]

where \( x = \) input sequence, \( N = \) number of elements of \( x \), \( Y = \) transform result

Frequency spacing:

\[ \Delta f = \frac{F_s}{N} \quad \text{where } F_s = \text{sampling rate} \]
Most real-world signals have no imaginary components, and these purely real signals produce an FFT that is symmetric about the DC component at 0 Hz. This means that values at negative frequencies are exactly the same as their positive counterparts, and these points are considered redundant.

### 1D FFT: Even Number of Samples

<table>
<thead>
<tr>
<th>Array Element</th>
<th>Corresponding Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_0$</td>
<td>DC component</td>
</tr>
<tr>
<td>$Y_1$</td>
<td>1$^{st}$ Harmonic</td>
</tr>
<tr>
<td>$Y_2$</td>
<td>2$^{nd}$ Harmonic</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$Y_{(k-1)}$</td>
<td>(k-1)$^{th}$ Harmonic</td>
</tr>
<tr>
<td>$Y_k$</td>
<td>Nyquist Frequency</td>
</tr>
<tr>
<td>$Y_{-(k-1)}$</td>
<td>-(k-1)$^{th}$ Harmonic</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$Y_{n-2}$</td>
<td>-2$^{nd}$ Harmonic</td>
</tr>
<tr>
<td>$Y_{n-1}$</td>
<td>-1$^{st}$ Harmonic</td>
</tr>
</tbody>
</table>

FFT($X$), $n=512$

- Positive harmonics
- Negative harmonics

DC Component

Nyquist Component

Points are considered redundant.
1D FFT: Even Number of Samples

Consider an example:
1024 samples of a real signal sampled at 100 kS/s

\[ \Delta f = \frac{100 \text{ kHz}}{1024} = 97.65625 \text{ Hz} \]

<table>
<thead>
<tr>
<th>Array Element</th>
<th>Corresponding Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y_0 )</td>
<td>0 Hz</td>
</tr>
<tr>
<td>( Y_1 )</td>
<td>97.65625 Hz</td>
</tr>
<tr>
<td>( Y_2 )</td>
<td>195.3125 Hz</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>( Y_{511} )</td>
<td>49.90234 kHz</td>
</tr>
<tr>
<td>( Y_{512} )</td>
<td>50 kHz</td>
</tr>
<tr>
<td>( Y_{513} )</td>
<td>-49.90234 kHz</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>( Y_{1023} )</td>
<td>-97.65625 Hz</td>
</tr>
</tbody>
</table>

FFT\{X\}, n=1024

DC Component
Nhuyquist Component
Positive harmonics
Negative harmonics
1D FFT: Odd Number of Samples

<table>
<thead>
<tr>
<th>Array Element</th>
<th>Corresponding Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_0$</td>
<td>DC component</td>
</tr>
<tr>
<td>$Y_1$</td>
<td>1$^{st}$ Harmonic</td>
</tr>
<tr>
<td>$Y_2$</td>
<td>2$^{nd}$ Harmonic</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$Y_{(k-1)}$</td>
<td>$(k-1)^{th}$ Harmonic</td>
</tr>
<tr>
<td>$Y_k$</td>
<td>$k^{th}$ Harmonic</td>
</tr>
<tr>
<td>$Y_{-(k-1)}$</td>
<td>$-(k-1)^{th}$ Harmonic</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$Y_{n-2}$</td>
<td>-2$^{nd}$ Harmonic</td>
</tr>
<tr>
<td>$Y_{n-1}$</td>
<td>-1$^{st}$ Harmonic</td>
</tr>
</tbody>
</table>

FFT\{X\}

- Positive harmonics
- Negative harmonics

DC Component

$Y_k$

$k^{th}$ Harmonic
The Power Spectrum

The power at a particular frequency component is the square of the magnitude at that component.

Power is always real, and all phase information is lost.

<table>
<thead>
<tr>
<th>Array Element</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{xx}[0]$</td>
<td>Power in DC component</td>
</tr>
<tr>
<td>$S_{xx}[1] = S_{xx}[N-1]$</td>
<td><em>Power at 1</em>st harmonic</td>
</tr>
<tr>
<td>$S_{xx}[2] = S_{xx}[N-2]$</td>
<td><em>Power at 2</em>nd harmonic</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$S_{xx}[k-2] = S_{xx}[N-(k-2)]$</td>
<td>*Power at (k-2)*th harmonic</td>
</tr>
<tr>
<td>$S_{xx}[k-1] = S_{xx}[N-(k-1)]$</td>
<td>*Power at (k-1)*th harmonic</td>
</tr>
<tr>
<td>$S_{xx}[k]$</td>
<td>*Power at (k)*th harmonic</td>
</tr>
</tbody>
</table>
Array-based Analysis vs Point-by-Point Analysis

**Point-by-point analysis** A subset of inline analysis where results are calculated after every individual sample rather than on a group of samples.
Array-based Analysis vs Point-by-Point Analysis

**Characteristics**

- Limited compatibility with real-time systems
- Specify a buffer
- "Delayed" processing
- Asynchronous work style

**Characteristics**

- Compatible with real-time systems
- No explicit buffers
- Immediate processing
- Synchronous work style
- Must initialize

Such analysis is essential when dealing with control processes featuring high-speed, deterministic, single-point data acquisition. Latency between acquisition and decision is minimized.
Spectral Leakage

Practical signals are finite, but the FFT assumes this time record repeats.

If you have an integral number of cycles in your time record, the repetition is smooth at the boundaries. If not, you get discontinuities.
Effects of Spectral Leakage on an FFT

Power Spectrum

Integer number of periods

Power Spectrum

Non-integer number of periods
Windowing

Windowing can be used to smooth these boundaries, reducing the size of the discontinuity and thus spectral leakage.
Frequency Response: No Window

The black curve shows the frequency response of FFT bin 3, blue is bin 4, etc. Note the poor sideband response of the filters.
Frequency Response: 4-term Blackman Harris

Using a 4-term Blackman Harris window before the FFT minimizes side lobes.

Tradeoff: the main lobe’s -3 dB bandwidth increased from 1 bin to 2 bins.
## Choosing a Filter

<table>
<thead>
<tr>
<th>Signal Characteristic</th>
<th>Window Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong frequency components far from your frequency of interest</td>
<td>Window with a high side lobe roll-off rate</td>
</tr>
<tr>
<td>Strong frequency components near your frequency of interest</td>
<td>Low maximum side lobe level</td>
</tr>
<tr>
<td>Area of interest is 2+ nearby frequencies</td>
<td>Narrow main lobe to maximize spectral resolution</td>
</tr>
<tr>
<td>Amplitude accuracy of a single frequency of interest is important</td>
<td>Window with a wide main lobe</td>
</tr>
<tr>
<td>Flat signal, or broadband in frequency content</td>
<td>Uniform window or no window</td>
</tr>
</tbody>
</table>

Not sure? A Hanning window will do the job most of the time, and is a good starting point.
Learn More Windowing Techniques

Find a handy guide online – search “windowing fft”

Windowing: Optimizing FFTs Using Window Functions

<table>
<thead>
<tr>
<th>Type of Signal</th>
<th>Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transients whose duration is shorter than the length of the window</td>
<td>Rectangular</td>
</tr>
<tr>
<td>Transients whose duration is longer than the length of the window</td>
<td>Exponential, Hanning</td>
</tr>
<tr>
<td>General-purpose applications</td>
<td>Hanning</td>
</tr>
<tr>
<td>Spectral analysis (frequency-response measurements)</td>
<td>Hanning (for random excitation), Rectangular (for pseudorandom excitation)</td>
</tr>
<tr>
<td>Separation of two tones with frequencies very close to each other but with widely differing amplitudes</td>
<td>Kaiser-Bessel</td>
</tr>
<tr>
<td>Separation of two tones with frequencies very close to each other but with almost equal amplitudes</td>
<td>Rectangular</td>
</tr>
<tr>
<td>Accurate single-tone amplitude measurements</td>
<td>Flat top</td>
</tr>
<tr>
<td>Sine wave or combination of sine waves</td>
<td>Hanning</td>
</tr>
<tr>
<td>Sine wave and amplitude accuracy is important</td>
<td>Flat top</td>
</tr>
</tbody>
</table>
What is Digital Signal Processing (DSP)?

1. Ensure signals are correctly digitized

2. Build the right signal processing algorithm
   - Choose the right FFT
   - Select the best window

3. Put your analysis in the right place
   - Offline
   - Software
   - Inline
Analysis in a Generic System Architecture

- Offline
- Inline with acquisition hardware
- On processor during acquisition
When Should Analysis be Offline?

Offline Analysis | Analysis performed after data acquisition

Benefits
- Not limited by timing and memory constraints of data acquisition
- Greater data interactivity. You have the ability to freely explore raw data and results of analysis
- Doesn’t bottleneck your acquisition. Offline analysis might be necessary for intensive algorithms operating on large data sets

Typical Use Cases
- Only useful when you do not need to make decisions as a result of this analysis as you acquire data
- Often used to identify the cause and effect of variables by correlating multiple data sets
- Histograms, trending, and curve-fitting are common offline analysis tasks
Performing Analysis in Software During Acquisition

General Purpose OS
- Can run many tasks at a time
- Familiar environment
- “General Purpose”

Real-Time OS
- Precise timing
- Higher level of determinism
- Typically run one program at a time
- Usually no user interface
Analyze Inline with Acquisition Hardware

Put analysis algorithms directly onto an FPGA

- True parallelism
- High reliability as designs become a custom circuit
- Runs algorithms at deterministic rates on the order of nanoseconds
- Reconfigurable
FPGA Technology

What is an FPGA?

- Software defined hardware
- No operating system is needed for execution of logic

Dedicated logic in silicon for highest reliability

High-speed and deterministic control

Inline signal processing with minimal latency
Mapping LabVIEW to an FPGA
FPGA Enables Data Reduction & In-line DSP

- Filtering
- Peak-detect
- FFTs
- Custom triggering

- Algorithmic pattern generation
- Co-processing
- Modulation/demodulation

Image Sensor

LCD Display

NI 6581 Adapter

Virtex-5 LX85 FPGA Modules

ni.com
High Performance Application: Real-Time OCT Imaging

Goal: Create a medical instrument to detect cancer without the patient undergoing a stressful biopsy.

A 3D optical coherence tomography (OCT) imaging system is a non-invasive solution with better resolution than traditional MRI or PET scans.

(a) (b)
High Performance Application: Real-Time OCT Imaging

Researchers used LabVIEW to perform analysis on highly parallel computing architectures. These technologies enabled them to achieve the fast processing necessary for 3D imaging.

Used LabVIEW with NI FlexRIO to compute over 700,000 512-point FFTs every second

Real-time 3D image rendering and display was performed using a high performance NVIDIA Quadro GPU
Analysis in a 3D OCT Application

- Offline data exploration using LabVIEW
- 3D Image Rendering
- Intensive FFT Processing
- Synchronized multichannel acquisition

Diagram:

- GPU
- CPU
- Real-Time Processor
- FPGA
- I/O
- I/O
- I/O
- I/O
Already CLAD Certified?

You’re immediately eligible to take the **Certified LabVIEW Developer** exam. Start preparing now!

- Join a local user group
- Prepare using resources on Developer Zone [ni.com/training/certification_prep](http://ni.com/training/certification_prep)
- Time yourself during practice exams

Note: CLAD certification must be current to take the CLD exam

Email [certification@ni.com](mailto:certification@ni.com) to register for an exam near you.
How would you rate 'Digital Signal Processing for Every Application'?

Text a **CODE** to 22333

<table>
<thead>
<tr>
<th>Rating</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>789893</td>
</tr>
<tr>
<td>Good</td>
<td>789894</td>
</tr>
<tr>
<td>Okay</td>
<td>789895</td>
</tr>
<tr>
<td>Poor</td>
<td>789896</td>
</tr>
</tbody>
</table>