Sonar Workbench User's Guide

Thomas J. Deal

Naval Undersea Warfare Center, Newport, RI

April 9, 2020

Abstract

Abstract goes here.

Contents

List of Tables

List of Figures

1 Introduction

Sonar Workbench is a suite of Matlab tools for the design and analysis of sonar systems. Much of the content is adapted from *An Introduction to Sonar Systems Engineering* [1], which is recommended as a companion resource for the user interested in understanding more of the theory.

2 Coordinate System and Reference Frames

Sonar Workbench uses a right-handed, Cartesian coordinate system known as North-East-Down (NED), as shown in Fig. 1. In this coordinate system, the first coordinate, x, points north, the second coordinate, y, points east, and the third coordinate, z, points down. Roll, γ, is rotation about the x axis, pitch, θ, is rotation about the y axis, and yaw, ψ, is rotation about the z axis. The NED coordinate system is ideal for underwater applications, because depth is measured downward from the surface, yaw is measured clockwise from north, and pitch is measured relative to the horizontal plane.

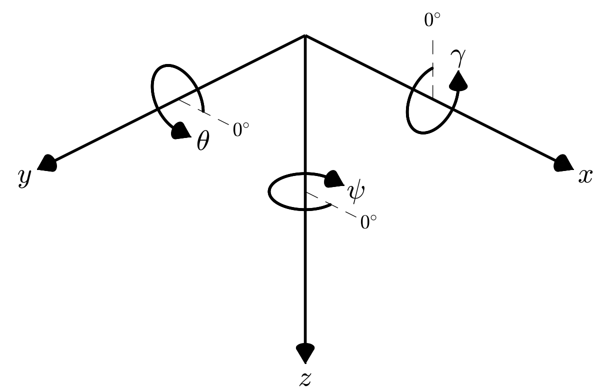

Figure 1: NED coordinate system

Sonar Workbench uses three reference frames: the element frame, the array frame, and the body frame. All frames use the NED coordinate system, and each frame can be located relative to another by a combination of translations[1] and rotations.

The element frame is always located at the center of the element, with the element's maximum response axis aligned with the $+x$ axis. Exceptions to this alignment are the omnidirectional element, which has no maximum response axis, and the linear element, which the user specifies as initially parallel to one of the three axes (x,y,z) in the element frame. For planar piston elements, the element face lies in the element frame y-z plane.

Each element in an array can have arbitrary translation and rotation in the array frame. The array frame origin and orientation is entirely up to the user, but it is typical for planar arrays to be located in center of the array

[1]displacement along the x, y, or z axes

frame's y-z plane and for volumetric arrays' geometric center to be located at the origin of the array frame.

The entire array can also be arbitrarily translated and rotated relative to the body frame. For example, a planar array on the nose of a torpedo might have a simple translation along the body frame x axis, while a flank array might have translations along the body x and y axes plus a rotation ψ about the body frame z axis. Figure 2 shows an example of element, array, and body frames for a conformal flank array.

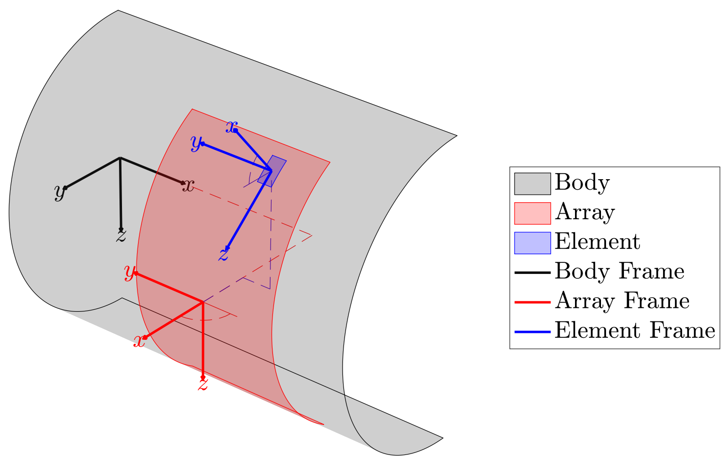

Figure 2: Body, array, and element frames for flank array example

Beam patterns are always computed in the body frame as a function of body azimuth angles ψ and elevation angles θ. Azimuth and elevation angles are measured from the body frame $+x$ axis. For simple analyses, the array and body frames can be aligned and colocated to produce beam patterns in the array frame. In this case, beam pattern angles ψ and θ are measured from the array frame $+x$ axis. More details about element, array, and body frame alignments will be explained in Sections 3 and 4.

3 Elements

The transducer element is the fundamental building block for arrays and the starting point for analysis in Sonar Workbench. For the purpose of generating and analyzing beam patterns, specific electromechanical transduction methods do not need to be modeled; instead, the element's acoustic properties can be captured by modeling vibrations of the element's wetted surface. Hereafter, references to an element's geometry refer to the geometry of the element's wetted surface or face. Sonar Workbench treats each element as a uniformly vibrating surface, which is to say that it only models each surface's fundamental mode of vibration. Analyzing element response requires first, defining the element geometry, and second, evaluating that geometry at a specific acoustic wavelength to produce an element pattern.

3.1 Element definition

Sonar Workbench includes support for the element types listed in Table 1.

Table 1: Included element types

Element Type	**Type String**
Omnidirectional	`'OmnidirectionalElement'`
Uniform Line	`'LinearElement'`
Cosine	`'CosineElement'`
Circular Piston	`'CircularPistonElement'`
Rectangular Piston	`'RectangulerPistonElement'`
Annular Piston	`'AnnularPistonElement'`
Hexagonal Piston	`'HexagonalPistonElement'`

An element structure holds the parameters that define the element geometry. The element structure must contain the `.type` field with a string corresponding to the name of a `.m` file that generates the corresponding element pattern. Table 1 lists the built-in element type strings, but the user can define additional element types by generating their own element pattern generator script with the same interface.

The field `.baffle` dictates whether the element should be baffled by the element frame y-z plane (e.g. arrays mounted to platforms) or unbaffled (e.g. towed arrays, sonobuoys). The omnidirectional, uniform line, and cosine elements can all be used with and without baffling. The piston elements' element patterns are all derived from equations assuming the piston is mounted in an infinite rigid baffle; therefore, it is recommended that the user set these element's `.baffle=1` for best results. Diffraction effects caused by finite baffle dimensions are beyond the scope of Sonar Workbench.

For visualization purposes, fields `.shapex`, `.shapey`, and `shapez` contain vectors of element shape coordinates. The script `AddElementShape.m` generates these vectors for the element types listed in Table 1. The other fields in the element structure depend on the element type, as listed in Table 2.

Table 2: Element structure fields

Element Type	**Field**	**Description**
Uniform Line	`.L`	length (m)
	`.axis`	aligned axis `'x'`, `'y'`, `'z'`
Circular Piston	`.a`	radius (m)
Rectangular Piston	`.w`	width (m)
	`.h`	height (m)
Annular Piston	`.a`	outer radius (m)
	`.b`	inner radius (m)
Hexagonal Piston	`.a`	inscribed circle radius (m)

Listing 1 shows the contents of `SampleElement.m`, which defines a rectangular piston element.

Listing 1: `SampleElement.m`

```
%% Element Design
Element.type = 'RectangularPistonElement';
Element.w = lambda/2;                  % Element face width, m
Element.h = lambda/4;                  % Element face height, m
Element.baffle = 1;                    % Hard Baffle
Element = AddElementShape(Element);
```

The user is free to add additional fields to the element structure for their own purposes. These will be ignored by Sonar Workbench.

3.2 Element patterns

Element geometry, coupled with an acoustic wavelength, defines the element pattern. The element pattern is the element's far-field directional response as a function of wavelength, azimuth and elevation, and it can be thought of as a spatial filter. Elements are assumed to be transducers, capable of transmitting and receiving sound, so there is no distinction between transmit and receive element patterns.

Sonar Workbench uses the acoustic wavelength, λ, to calculate element patterns because it combines the frequency and sound speed into a single term. It is related to sound speed c, frequency f in Hz or ω in rad/s, and wavenumber k in m^{-1} by

$$\lambda = \frac{c}{f} = \frac{2\pi c}{\omega} = \frac{2\pi}{k}.$$

At its most general, the element pattern is the three-dimensional Fourier transform of the element's complex aperture function, $A(\lambda, x, y, z)$. For the piston elements, the element pattern reduces to a two-dimensional Fourier transform, since the element face lies in the y-z plane,

$$E_{piston}(\lambda, \theta, \psi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(\lambda, y, z) e^{j2\pi\left(\frac{\cos\theta \sin\psi}{\lambda} y + \frac{\sin\theta}{\lambda} z\right)} dy dz, \tag{1}$$

and for the uniform line array, it further reduces to a one-dimensional Fourier transform,

$$E_{line}(\lambda, \theta, \psi) = \int_{-\infty}^{\infty} A(\lambda, y) e^{j2\pi \frac{\cos\theta \sin\psi}{\lambda} y} dy, \tag{2}$$

for a line array aligned with the y axis. The finite element extents make the integration limits finite. For the simple elements included with Sonar Workbench, the assumption of uniform surface motion means that the aperture function is real-valued and equal to 1 over the entire element surface. This simplifies the integration for certain element geometries. These integrals have analytic solutions, which Sonar Workbench uses instead of evaluating the integrals numerically.

Element patterns are normalized such that they have unity gain along their maximum response axis. Figure 3 shows element patterns for each of the included element types listed in Table 1 for a wavelength equal to half of the element's maximum dimension. The uniform line element is aligned with the y axis, and the cosine element is aligned with the x axis. Note that the omnidirectional and cosine element patterns do not depend on wavelength.

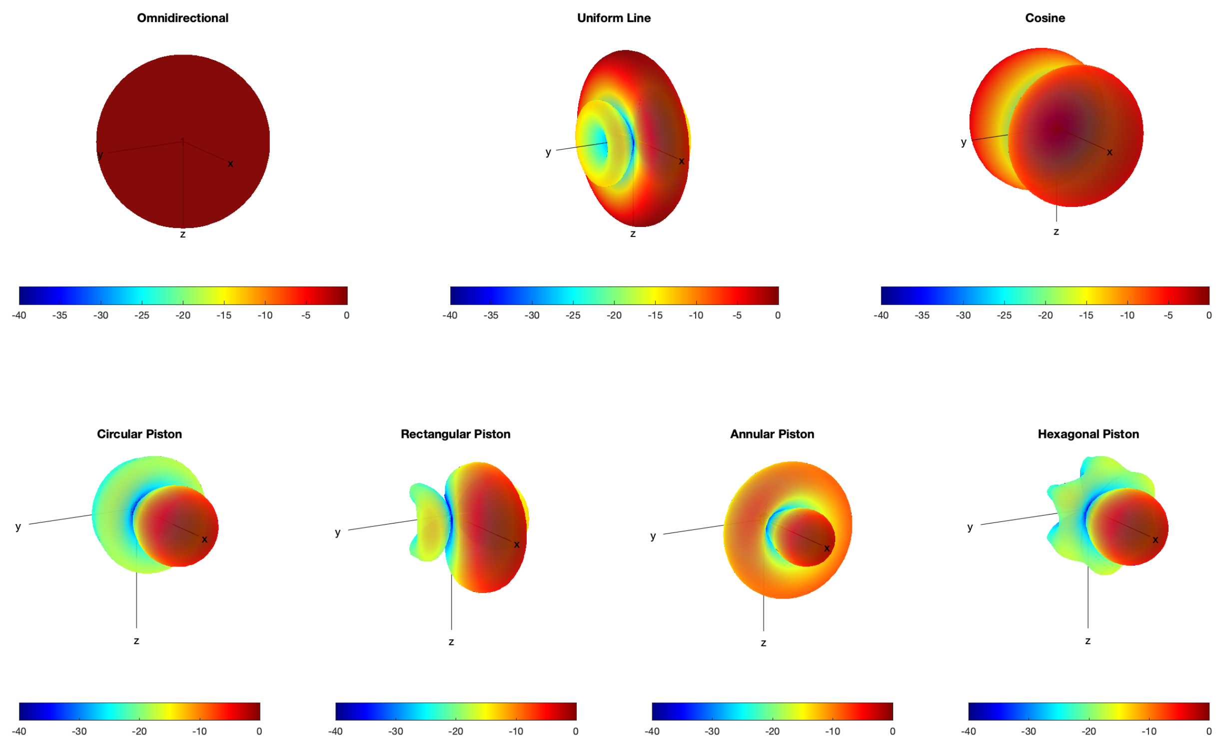

Figure 3: Element patterns for included element types

4 Arrays

5 Beams

References

[1] L. J. Ziomek, *An introduction to sonar systems engineering*. Boca Raton, FL: Taylor & Francis/CRC, 1st ed., 2017.