SeismicAirgun User Guide

Leighton M. Watson
Stanford University
leightonwatson@stanford.edu

June 2, 2019

I. Introduction

SeismicAirgun computes airgun/bubble dynamics following a similar treatment to the seminal work of Ziolkowski (1970). A lumped parameter model is used where the internal properties of the airgun and bubbles are assumed to be spatially uniform. **SeismicAirgun** is written in `MATLAB` and runs efficiently on a standard desktop/laptop computer. For more details and examples of the application of **SeismicAirgun** see:

- Chelminski, S., Watson, L. M., and Ronen, S. (2019) Low-frequency pneumatic seismic sources, *Geophysical Prospecting*, https://doi.org/10.1111/1365-2478.12774.
- Watson, L. M., Dunham, E. M., and Ronen, S. (2016) Numerical modeling of seismic airguns and low-pressure sources, *SEG Technical Program Expanded Abstracts*, https://doi.org/10.1190/-segam2016-13846118.1.

II. Directory

- **demo** - script files for demonstration.
- **doc** - documentation including user guide and license file.
- **source** - function files associated with numerical implementation.

SeismicAirgun is freely available online at https://github.com/leighton-watson/SeismicAirgun and is distributed under the MIT license (see `license.txt` for details).

III. Model Description

Here, I describe the mathematical model of **SeismicAirgun**, which is a lumped parameter model where the internal properties of the airgun and bubble are assumed to be spatially uniform. The model can be divided into three components; i.) bubble, ii.) airgun, and iii.) acoustic radiation. For more details see Chelminski et al. (2019) and Watson et al. (2016) and the references therein.

i. Bubble

The motion of the bubble wall is governed by the modified Herring equation (Herring, 1941; Cole, 1948; Vokurka, 1986):

$$R\ddot{R} + \frac{3}{2}\dot{R}^2 = \frac{p_b - p_\infty}{\rho_\infty} + \frac{R}{\rho_\infty c_\infty}\dot{p}_b, \tag{1}$$

where R, $\dot{R} = dR/dt$, and $\ddot{R} = d^2R/dt^2$ are the radius, velocity, and acceleration of the bubble wall, respectively, p_b is the pressure inside the bubble, $\dot{p}_b = dp_b/dt$, and p_∞, ρ_∞ and c_∞ are the

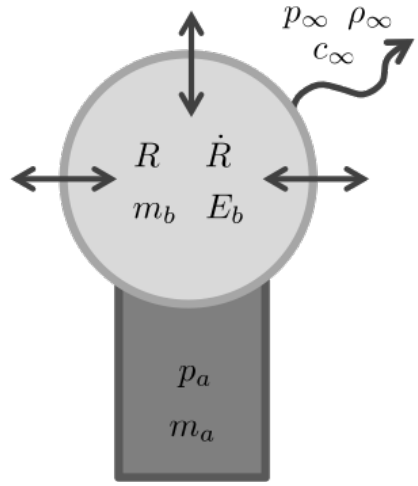

Figure 1: *Model schematic. The airgun and bubble are treated as lumped parameter objects. The internal properties are assumed to be spatially uniform and described by a single value that evolves in time.*

pressure, density, and speed of sound, respectively, in the water infinitely far from the bubble. The ambient pressure at the depth of the airgun, D, is given by $p_\infty = p_{\text{atm}} = \rho_\infty g D$, where p_{atm} is the atmospheric pressure and g is the gravitational acceleration.

Mass flow from the airgun to the bubble is:

$$\frac{dm_b}{dt} = \begin{cases} p_a A\left(\frac{\gamma}{QT_a}\right)^{1/2}\left(\frac{2}{\gamma-1}\right)^{1/2}\left[\left(\frac{p_a}{p_b}\right)^{(\gamma-1)/\gamma} - 1\right] & \text{if flow is unchoked} \\ p_a A\left(\frac{\gamma}{QT_a}\right)^{1/2} & \text{if flow is choked} \end{cases} \tag{2}$$

where m_b is the mass of the bubble, A is the airgun port area, Q is the specific gas constant, and p_a and T_a are the airgun pressure and temperature, respectively. Flow through the port is choked (choked flow is when the flow through a nozzle has a velocity equal to the sound speed, the maximum possible velocity for fluid flow through a nozzle) if (Babu, 2014)

$$\frac{p_a}{p_b} \geq \left(\frac{\gamma+1}{2}\right)^{\gamma/(\gamma-1)}, \tag{3}$$

where γ is the ratio of heat capacities. If the pressure ratio is less than this critical value then flow through the port will be unchoked.

The internal energy of the bubble, E_b, changes according to the first law of thermodynamics for an open system:

$$\frac{dE_b}{dt} = c_p T_a \frac{dm_b}{dt} - 4\pi M \kappa R^2 (T_b - T_\infty) - p_b \frac{dV_b}{dt}, \tag{4}$$

where c_p is the heat capacity at constant volume, κ is the heat transfer coefficient, M is a constant that accounts for the increased effective surface area over which heat transfer can occur as a result of turbulence at the bubble walls (Laws et al., 1990), and $V_b = (4/3)\pi R^3$ is the volume of the bubble, which is assumed to be spherical.

ii. Airgun

The airgun and bubble are coupled by conservation of mass:

$$\frac{dm_a}{dt} = -\frac{dm_b}{dt}, \tag{5}$$

where m_a is the mass of air inside the airgun, and conservation of energy:

$$\frac{dE_a}{dt} = -c_p T_a \frac{dm_b}{dt}, \tag{6}$$

where E_a is the internal energy of the airgun.

The airgun and bubble governing equations are closed with the ideal gas equation of state:

$$p = \frac{mQT}{V}, \tag{7}$$

and the relationship between the internal energy and temperature:

$$E = c_v mT, \tag{8}$$

where c_v is the heat capacity at constant volume.

iii. Acoustic Radiation

The pressure perturbation in the water is related to the bubble dynamics by (Keller and Kolodner, 1956)

$$\Delta p(r,t) = \rho_\infty \left[\frac{\ddot{V}(t - r/c_\infty)}{4\pi r} - \frac{\dot{V}(t - r/c_\infty)^2}{32\pi^2 r^4} \right], \tag{9}$$

where Δp is the pressure perturbation in the water, r is the distance from the center of the bubble to the receiver, and V is the volume of the bubble. The second term on the right side is a near-field term that decays rapidly with distance. In addition to the direct arrival there is a ghost arrival, which is the initially upgoing wave that is reflected with negative polarity from the sea surface. For a receiver directly below the source the observed pressure perturbation is

$$\Delta p_{\text{obs}}(r,t) = \Delta p_D(r,t) - \Delta p_G(r + 2D, t), \tag{10}$$

where Δp_D is the direct signal and Δp_G is the ghost arrival, which is assumed to have -1 reflection coefficient from the sea surface. D is the depth of the seismic airgun (Figure 2).

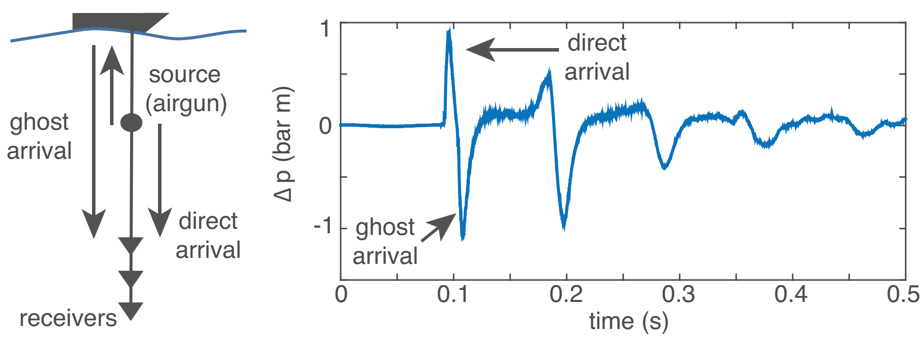

Figure 2: *Schematic showing direct and ghost arrivals along with example source signature from Watson et al. (2016).*

IV. Numerical Implementation

SeismicAirgun solves a system of six ordinary differential equations using `MATLAB`'s built-in solver, `ode45`, which is explicit Runge-Kutta (4,5) method. The variables that are evolved are (1) R, bubble radius, (2) $\dot{R}$, bubble wall velocity, (3) m_b, the mass of bubble, (4) T_b, temperature of bubble, (5) p_a, pressure inside the airgun, and (6) m_a, mass of air inside the airgun.

V. Using **SeismicAirgun**

The script file `example.m` demonstrates how to run the code. In this file the firing properties of the airgun (pressure, volume, port area, depth and distance from source to receiver) are specified:

```
airgunParams = [pressure, volume, port_area]; % specify firing configuration [psi, in^3,
    in^2]
depth = 10; % depth of air gun [m]
r = 75; % distance from source to receiver [m]
physConst = physical_constants(depth,r); % load physical constants
```

The code is run by using one of the two following script files:

```
AirgunBubbleSolve(airgunParams, physConst); %works for multiple firing configurations
output = AirgunBubbleSolveOutput(airgunParams, physConst, false); %run solver and saves
    outputs (only works for a single firing configuration). True or false flag determines
     if solution is plotted or not.
```

There are two options for running the code. The first option is to use `AirgunBubbleSolve.m`. This function can handle multiple airgun properties (pressure, volume, port area) and will plot the modeled source signatures in the time and frequency domain. However, the outputs are not saved. The second option is `AirgunBubbleSolveOutput.m`. This function will save the simulation outputs to the current workspace but currently only works for a single set of airgun properties. The structure of the code is similar for the two options and is shown below.

The airgun and bubble properties are initialized from the specified `airgunParams` and `physConst`:

```
airgunInit = airgun_initialization(airgunParams(i,:), physConst); %air gun
bubbleInit = bubble_initialization(airgunInit, physConst); %bubble
```

The initial conditions and parameters are saved into vectors that can be loaded into `ode45`:

```
initCond = initCond_save(bubbleInit, airgunInit); %inital conditions [R,U,mb,Tb,pa,ma]
params = params_save(bubbleInit, airgunInit, physConst); %save ambient water, airgun and
    bubble parameters into vector
```

The initial conditions on the airgun variables are determined from the specified `airgunParams`. The pressure is explicitly specified in `airgunParams` while the mass of air inside the airgun is calculated by assuming that the airgun temperature is equal to the ambient temperature in the water. This a reasonable assumption because the air is compressed on the source boat and then pumped to the airgun through umbilical cords that are suspended in the water, providing sufficient time for the air to equilibrate in temperature with the water.

The bubble is initialized with a finite initial volume equal to the volume of the airgun. This is because the modified Herring equation (equation 1) becomes singular when $R \to 0$. The bubble wall velocity is initially zero and the bubble air is at the same temperature and pressure as the ambient water. The mass can then be calculated from equation 7. For more discussion on the bubble initial conditions see Watson et al. (2019).

The governing equations for the airgun and bubble dynamics are specified in `modified_herring_eqn.m` and are solved using `ode45` starting from the initial conditions specified in the vector `initCond`:

```
sol = ode45(@modified_herring_eqn, physConst.time, initCond, [], params);
```

Once the airgun and bubble dynamics are known, the pressure perturbation (source signature) in the water is computed by equation 9. In the current version of **SeismicAirgun** the receiver is assumed to be directly below the source, and hence the ghost is calculated as having a propagation distance of `r+2D`:

```
[tDir, pDir] = pressure_eqn(t, R, U, A, physConst.rho_infty, physConst.c_infty, physConst
    .r); %direct arrival
[tGhost, pGhost] = pressure_eqn(t, R, U, A, physConst.rho_infty, physConst.c_infty,
    physConst.r + 2*physConst.depth); %ghost arrival
```

The direct and ghost arrivals are interpolated onto a common time vector (`tInterp`) and the pressure perturbation in the frequency domain is computed using `pressure_spectra`. Example model outputs for a 1000 in^3 airgun pressurized to 1600 psi with a port area of 16 in^2 fired at a depth of 10 m are shown below in Figure 3.

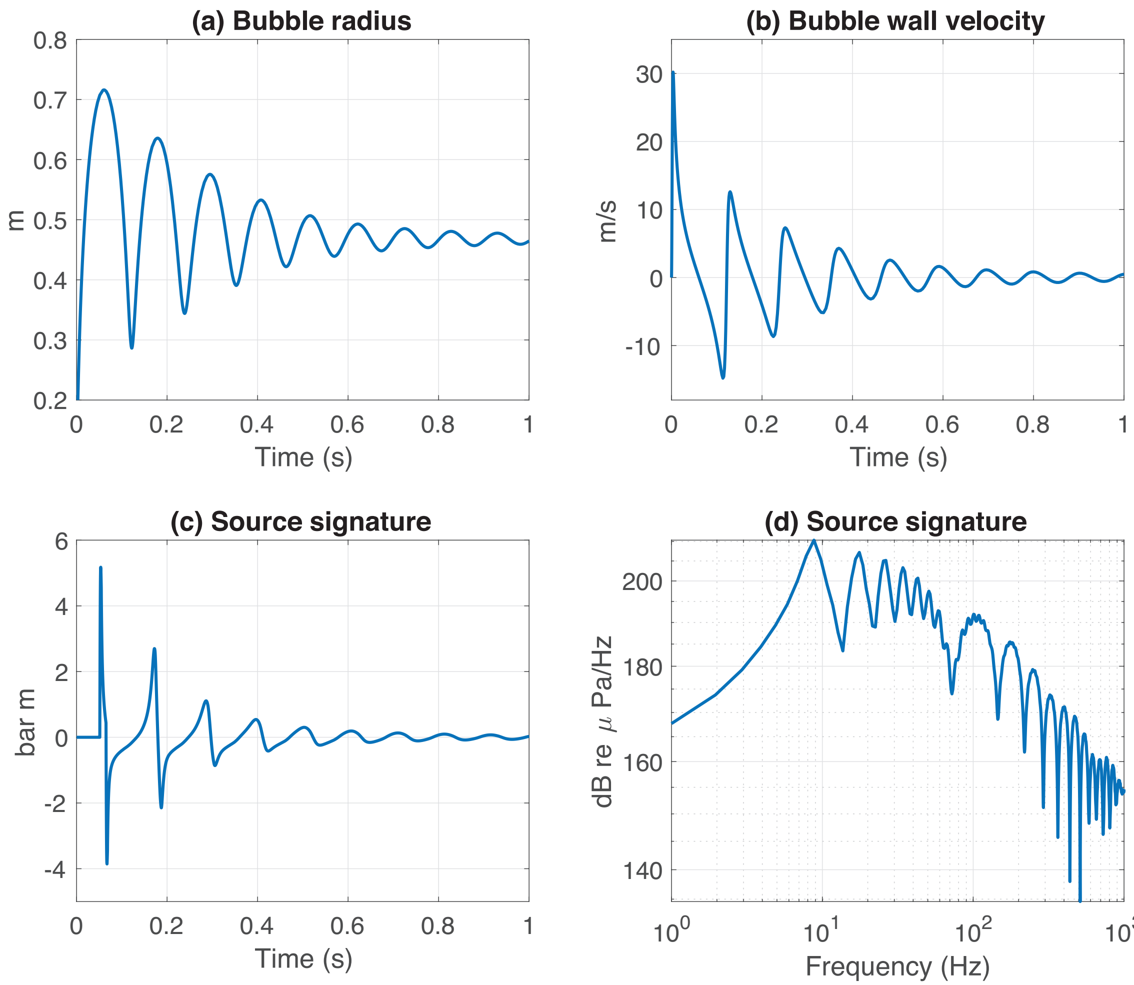

Figure 3: *Example model outputs for a 1000 in^3 airgun pressurized to 1600 psi with a port area of 16 in^2 fired at a depth of 10 m showing (a) bubble radius, (b) bubble wall velocity, and source signature (acoustic pressure perturbation in the water) in the (c) time and (d) frequency domains. The source signature in the time domain is displayed in bar m, which enables comparison between observations at different source-receiver distances.*

References

Babu, V., 2014. Fundamentals of Gas Dynamics, 2nd Edition. John Wiley & Sons, Chichester.

Chelminski, S., Watson, L. M., Ronen, S., 2019. Low frequency pneumatic seismic sources. Geophysical Prospecting, 1–10.

Cole, R. H., 1948. Underwater explosions. Princeton University Press, Princeton.

Herring, C., 1941. Theory of the pulsations of the gas bubble produced by an underwater explosions. Tech. rep., Office of Scientific Research and Development.

Keller, J. B., Kolodner, I. I., 1956. Damping of underwater explosion bubble oscillations. Journal of Applied Physics 27 (10), 1152–1161.

Laws, R. M., Hatton, L., Haartsen, M., 1990. Computer modelling of clustered airguns. First Break 8 (9), 331–338.

Vokurka, K., 1986. Comparison of Rayleigh's , Herring's , and Gilmore's Models of Gas Bubbles. Acta Acustica united with Acustica 59 (3), 214–219(6).

Watson, L. M., Dunham, E. M., Ronen, S., 2016. Numerical modeling of seismic airguns and low-pressure sources. In: SEG Technical Program Expanded Abstracts 2016. No. 1. pp. 219–224.

Watson, L. M., Werpers, J., Dunham, E. M., 2019. What controls the initial peak of an air gun source signature? Geophysics 84 (2), P27–P45.

Ziolkowski, A., 1970. A Method for Calculating the Output Pressure Waveform from an Air Gun. Geophysical Journal of the Royal Astronomical Society 21, 137–161.