DETERMINATION OF PARTICLE DISPLACEMENT OR VELOCITY FROM PARTICLE ACCELERATION MEASUREMENTS

BACKGROUND

[0001] In the oil and gas industry, geophysical prospecting is commonly used to aid in the search for and evaluation of subterranean formations. Geophysical prospecting techniques yield knowledge of the subsurface structure of the earth, which is useful for finding and extracting valuable mineral resources, particularly hydrocarbon deposits such as oil and natural gas. One technique associated with geophysical prospecting is a seismic survey. In a land-based seismic survey, a seismic signal is generated on or near the earth's surface and then travels downward into the subsurface of the earth. In a marine seismic survey, the seismic signal may also travel downward through a body of water overlying the surface of the earth. Seismic energy sources are used to generate the seismic signal which, after propagating into the earth, is at least partially reflected by subsurface seismic reflectors. Such seismic reflectors typically are interfaces between subterranean formations having different elastic properties, such as sound wave velocity and rock density, which lead to differences in acoustic impedance at the interfaces. The reflected seismic energy is detected and recorded by seismic sensors (also called seismic receivers) at or near the surface of the earth, in an overlying body of water, or at known depths in boreholes.

[0002] Example seismic sources for generating the seismic signal in land seismic surveys include, without limitation, explosives, vibrators, or other devices capable of generating seismic waves. Marine seismic surveys typically employ a submerged seismic source towed by a ship and periodically activated to generate an acoustic wavefield. The seismic source generating the wavefield may be of several types including, without limitation, a small explosive charge, an electric spark or arc, a marine vibrator or a gun. The seismic source gun may be a water gun, a vapor gun or an air gun. In many cases, the seismic source consists not of a single source element, but of a spatially-distributed array of source elements.

[0003] The appropriate types of seismic sensors are also diverse and may depend on the application. Example seismic sensors include, without limitation, particle velocity sensors, particularly in land surveys, and water pressure sensors, particularly in marine surveys. Seismic sensors may be deployed by themselves, but are more commonly

deployed in sensor arrays. Additionally, different types of sensors, such as pressure sensors and particle acceleration sensors, may be deployed together in a seismic survey, collocated in pairs or pairs of arrays.

[0004] The resulting seismic data obtained in performing the survey is processed to yield information relating to the geologic structure and properties of the subterranean formations in the area being surveyed. For example, the processed seismic data may be processed for display and analysis of potential hydrocarbon content of these subterranean formations. The goal of seismic data processing is to extract from the seismic data as much information as possible regarding the subterranean formations in order to adequately image or otherwise characterize the geologic subsurface. Accurate characterizations of the geologic subsurface may greatly facilitate geophysical prospecting for petroleum accumulations or other mineral deposits.

[0005] The approaches described in this section are approaches that could be pursued, but not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated, it should not be assumed that any of the approaches described in this section qualify as prior art merely by virtue of their inclusion in this section.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] In the drawings:

[0007] FIG. 1 is a graph illustrating example results of calculating particle displacement data from noise-free particle acceleration data;

[0008] FIG. 2 is a graph illustrating example results of calculating particle velocity data from noise-free particle acceleration data;

[0009] FIG. 3 is a graph illustrating an example of the effects of noise on calculation of particle displacement data from particle acceleration data;

[0010] FIG. 4 is a graph illustrating an example of the effects of noise on calculation of particle velocity data from particle acceleration data;

[0011] FIG. 5 is a graph of example amplitude responses of a double integration filter as applied to acceleration data with noise;

[0012] FIG. 6 is a graph of example amplitude responses of a single integration filter as applied to acceleration data with noise;

[0013] FIG. 7 is a graph illustrating an example calculation of particle displacement data from noisy acceleration data using a deterministic Wiener filter;

[0014] FIG. 8 is a graph illustrating an example calculation of particle velocity data form noisy acceleration data using a deterministic Wiener filter;

[0015] FIG. 9 is a flowchart illustrating a process for calculating particle displacement data from particle acceleration data using a deterministic Wiener filter;

[0016] FIG. 10 is a flowchart illustrating a process for calculating particle velocity data from particle acceleration data using a deterministic Wiener filter;

[0017] FIG. 11 is a table that summarizes the use of filters for the calculation of one measurement from another measurement;

[0018] FIG. 12 is a flowchart illustrating a process for selecting a damping factor when estimating particle displacement;

[0019] FIG. 13 is a flowchart illustrating a process for selecting a damping factor when estimating particle velocity;

[0020] FIG. 14 is a graph showing different scanned damping factors;

[0021] FIG. 15 is a graph illustrating the results of different scanned damping factors when estimating particle velocity;

[0022] FIG. 16 is a block diagram that illustrates an example computer system upon which an embodiment may be implemented;

[0023] FIG. 17 is an illustration depicting a side view of an example marine seismic survey environment in which an embodiment may be implemented.

DETAILED DESCRIPTION

[0024] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention. Various aspects of the invention are described hereinafter in the following sections:

I. **OVERVIEW**

II. **ACCELERATION MEASUREMENTS AND THE EFFECTS OF NOISE**

III. CALCULATION OF PARTICLE DISPLACEMENT OR PARTICLE VELOCITY IN THE PRESENCE OF NOISE

IV. FILTER OPTIMIZATION FOR STABLE CALCULATIONS

V. IMPLEMENTATION EXAMPLES

VI. EXTENSIONS AND ALTERNATIVES

I. OVERVIEW

[0025] Techniques are described for determining particle displacement or particle velocity data from particle acceleration measurements. In an embodiment, an apparatus is configured to receive particle acceleration data. The apparatus may further be configured to determine one or more of particle velocity data or particle displacement data, based upon the particle acceleration data, by processing the particle acceleration data using a filter that at least approximates a Wiener filter and that uses one or more damping factors selected to provide stability in the presence of noise in the particle acceleration data.

[0026] In an embodiment, the apparatus is configured to automatically determine a damping factor based on the received particle acceleration data. To determine the damping factor, the apparatus may be configured to determine calculated acceleration data based on the particle velocity or particle displacement data and determine whether a threshold is satisfied by comparing the calculated acceleration data with the particle acceleration data. In response to determining that the threshold is not satisfied, the apparatus adjusts the damping factor. In response to determining that the threshold is satisfied, the apparatus determines that the damping factor is a satisfactory damping factor. When comparing the calculated acceleration data with the particle acceleration data, the apparatus may be configured to determine a first energy value from the calculated acceleration data, determine a second energy value from the particle acceleration data, and determine a difference between the first energy value and the second energy value. The apparatus may determine that the threshold is satisfied if the difference between the first energy value and the second energy value is less than a threshold. In another embodiment, the apparatus may be configured to automatically determine the damping factor for each distinct set of acceleration data that is received.

[0027] In an embodiment, the apparatus may be configured to determine both the particle velocity data and the particle displacement data, based upon the particle acceleration data, by processing the particle acceleration data using the filter that at least

approximates a Wiener filter and that uses one or more damping factors selected to provide stability in the presence of noise in the particle acceleration data. The apparatus may be configured to determine the particle velocity data by processing the particle acceleration data using the filter and a first damping factor and the particle displacement data by processing the particle acceleration data using the filter and a second damping factor. The first damping factor may be different than the second damping factor. Thus, different damping factors may be used to determine particle velocity data and particle displacement data.

[0028] In an embodiment, the Wiener filter is of a form represented by $F(\omega) = \frac{-i\omega}{\omega^2+\varepsilon}$, wherein $F(\omega)$ represents the Wiener filter, i represents an imaginary number, ω represents an angular frequency, and ε represents a damping factor.

[0029] In an embodiment, the apparatus may be configured to characterize subterranean geological features based on at least one of the particle velocity data or the particle displacement data.

II. ACCELERATION MEASUREMENTS AND THE EFFECTS OF NOISE

[0030] Recently, there has been increased interest in accelerometers for various applications such as seismic exploration and other industrial uses. In particular, micro-electro mechanical systems (MEMS) accelerometers have proved effective for efficiently recording particle acceleration, which is often helpful in determining the geologic structure and properties of the subterranean formations in an area being surveyed.

[0031] Acceleration measurements recorded by MEMS accelerometers typically have a flat amplitude response from very low to very high frequencies. Despite these positive characteristics, the calculation of particle displacement and velocity by integrating acceleration is not stable in the presence of noise. The instability manifests itself in the form of very high amplitude low frequency noise masking the displacement or velocity response after integration.

[0032] Equations (1) through (6), provided below, characterize the relationship between particle acceleration, particle displacement, and particle velocity. Reference is made to these equations to highlight the effects of noise on calculating particle velocity and displacement values from acceleration data.

[0033] The relationship between particle acceleration and particle displacement may be characterized as follows:

$$a(t) = \frac{d^2u(t)}{dt^2} \qquad (1)$$

where a(t) represents particle acceleration as a function of time (t), u(t) represents particle displacement as a function of time (t), and $\frac{d^2u(t)}{dt^2}$ represents the second derivative of u(t) with respect to time (t).

[0034] The relationship between particle acceleration and particle velocity may be characterized as follows:

$$a(t) = \frac{dv(t)}{dt} \qquad (2)$$

where a(t) represents particle acceleration as a function of time (t), v(t) represents particle velocity as a function of time (t), and $\frac{dv(t)}{dt}$ represents the first derivative of v(t) with respect to time (t).

[0035] In the frequency domain, equation (1) may be expressed as:

$$A(\omega) = -\omega^2 U(\omega) \qquad (3)$$

and equation (2) may be expressed as:

$$A(\omega) = i\omega V(\omega) \qquad (4)$$

where $A(\omega)$, $U(\omega)$ and $V(\omega)$ represent particle acceleration, displacement, and velocity respectively in the angular frequency domain (ω), and i is an imaginary unit that may be expressed as $i = \sqrt{-1}$.

[0036] When the recorded data are acceleration measurements, and particle displacement or velocity is to be calculated, the particle acceleration may be: 1) double integrated to obtain particle displacement; or 2) single integrated to obtain particle velocity. The frequency domain expressions in equations (3) and (4) may be inverted, as expressed in equations (5) and (6), to achieve double or single integration, respectively.

[0037] For particle displacement calculations, the double integration may be represented in the frequency domain as:

$$U(\omega) = \frac{1}{-\omega^2} A(\omega) \qquad \text{for } \omega > 0 \qquad (5)$$

For particle velocity, the single integration may be represented in the frequency domain as:

$$V(\omega) = \frac{1}{i\omega} A(\omega) \qquad \text{for } \omega > 0 \qquad (6)$$

[0038] Equations (5) and (6) are stable when the particle acceleration measurements are noise free, which is demonstrated in the synthetic data examples shown in FIGS. 1 and

2 for the estimation of particle displacement and velocity, respectively. However, when noise is added to the particle acceleration measurements, equations (5) and (6) become unstable, which is demonstrated in the synthetic data examples shown in FIGS. 3 and 4.

[0039] FIG. 1 is a graph illustrating example results of calculating particle displacement data from noise-free particle acceleration data. Referring to FIG. 1, trace (a) 102 shows synthetic ('observed') particle acceleration data. Trace (a) 102 and other traces labeled (a) referenced herein simulate acceleration data that may be recorded or otherwise measured by a MEMS or other accelerometer. Particle displacement is estimated by applying equation (5) to trace (a) 102. The results of this calculation are displayed in trace (e) 110. Trace (d) 108 shows the 'true' particle displacement data and trace (f) 112 is the difference between the calculated displacement data shown by trace (e) 110 and the 'true' particle displacement data shown by trace (d) 108. Because there is no noise in this simulated case, trace (f) 112 is very small. Trace (b) 104 shows the calculated particle acceleration data using the forward equation (3) on trace (e) 110. Trace (c) 106 shows the difference between the 'observed' and 'calculated' particle acceleration data, which is also very small. This simulation demonstrates that in the absence of noise, equation (5) is very stable in calculating particle displacement.

[0040] FIG. 2 is a graph illustrating example results of calculating particle velocity from noise-free particle acceleration data. Referring to FIG. 2, trace (a) 202 shows synthetic ('observed') particle acceleration data. Particle velocity is estimated using this trace and equation (6). The results of this calculation are displayed in trace (e) 210. Trace (d) 208 shows the 'true' particle velocity data and trace (f) 212 is the difference between the calculated velocity data shown in trace (e) 210 and the 'true' particle velocity data shown in trace (d) 208. Because there is no noise in this simulated case, trace (f) 212 is very small. Trace (b) 204 shows the calculated particle acceleration data using the forward equation (4) on trace (e) 210. Trace (c) 206 shows the difference between the 'observed' and 'calculated' particle acceleration data, which is very small as well. This demonstrates that when there is no noise, equation (6) is very stable in calculating particle velocity.

[0041] FIG. 3 is a graph illustrating an example of the effects of noise on calculation of particle displacement data from particle acceleration data. Referring to FIG. 3, trace (a) 302 shows synthetic ('observed') particle acceleration data. Unlike trace (a) 102, trace (a) 302 includes noise that has been added to the acceleration data. Particle displacement is estimated using trace (a) 302 and equation (5). The results of this calculation are displayed

in trace (e) 310. Trace (d) 308 shows the 'true' particle displacement data and trace (f) 312 is the difference between the calculated displacement data shown by trace (e) 310 and the 'true' particle displacement data shown by trace (d) 308. Because of the added noise in this simulated case, trace (f) 312 is very large. Trace (b) 304 shows the calculated particle acceleration data using the forward equation (3) on trace (e) 310. Trace (c) 306 shows the difference between the 'observed' and 'calculated' particle acceleration data, which is very small. This simulation demonstrates that in the presence of noise, equation (5) is unstable in calculating particle displacement.

[0042] FIG. 4 is a graph illustrating an example of the effects of noise on calculation of particle velocity data from particle acceleration data. Referring to FIG. 4, trace (a) 402 shows synthetic ('observed') particle acceleration data. Unlike trace (a) 202, trace (a) 402 includes noise that has been added to the acceleration data. Particle velocity is estimated using trace (a) 402 and equation (6). The results of this calculation are displayed in trace (e) 410. Trace (d) 408 shows the 'true' particle velocity data and trace (f) 412 is the difference between the calculated velocity data shown in trace (e) 410 and the 'true' particle velocity data shown in trace (d) 408. Because of the added noise in this simulated case, trace (f) 412 is very large. Trace (b) 404 shows the calculated particle acceleration data using the forward equation (4) on trace (e) 410. Trace (c) 406 shows the difference between the 'observed' and 'calculated' particle acceleration data, which is very small. This demonstrates that when there is no noise, equation (6) is very stable in calculating particle velocity.

[0043] As illustrated in FIGS. 3 and 4, a small amount of noise makes the integration process unstable. The calculated particle displacement data (trace (e) 310) and velocity (trace (e) 410) show high amplitude low frequency components created by noise. Trace (f) 312 and trace (f) 412 highlight the low frequencies built up by the integration performed with equations (5) and (6) when noise is present in the acceleration data.

[0044] The origin of the instability can be explained by inspecting the amplitude response of the integration filters for the calculation of particle displacement $\left[\frac{1}{-\omega^2}\right]$ and velocity $\left[\frac{1}{i\omega}\right]$ used in equations (5) and (6) respectively. FIG. 5 illustrates example amplitude responses of a double integration filter. Referring to FIG. 5 , amplitude response 502 corresponds to the amplitude response of squared Wiener filter 506 when the value of $\varepsilon=0$. The use of a squared Wiener filter for providing stability in the presence of

noise is discussed in more detail below. In the present case, when $\varepsilon=0$, squared Wiener filter 506 is the same as the double integration filter of equation (5). As illustrated by amplitude response 502, as the frequency approaches zero a singularity amplifies the energy of the lower end of the frequency spectrum. When there is noise in the particle acceleration measurements, the amplification shown in amplitude response 502 is responsible for the abnormal low frequency noise boosting in displacement calculations.

[0045] FIG 6 illustrates example amplitude responses of a single integration filter. Referring to FIG. 6, amplitude response 602 corresponds to the amplitude response of Wiener filter 606 when the value of $\varepsilon=0$, resulting in the single integration filter of equation (6). As in the case of the amplitude response of the double integration filter, amplitude response 602 shows that as the frequency approaches zero, the energy of the lower end of the frequency spectrum is amplified. When there is noise in the particle acceleration measurements, the amplification shown in amplitude response 602 is responsible for the abnormal low frequency noise boosting in velocity calculations.

III. CALCULATION OF PARTICLE DISPLACEMENT OR PARTICLE VELOCITY IN THE PRESENCE OF NOISE

[0046] In order to reduce the effects of noise highlighted above, a noise-reduction filter may be used to estimate particle displacement data and/or particle velocity data from the noisy acceleration data. In one embodiment, the respective inverse filters $\frac{1}{-\omega^2}$ and $\frac{1}{i\omega}$ associated with equations (5) and (6), respectively, may be formulated as a deterministic Wiener filter. As illustrated below, a Wiener filter has good performance characteristics in the presence of noise and may achieve stability during the double and single integration of the particle acceleration data for the calculation of particle displacement and velocity. Further, the use of a filter that at least approximates a Wiener filter, as described below, avoids the need to apply filtering after integration and may operate in the presence of noise without estimating or knowing about the noise.

[0047] In an embodiment, a Wiener filter as used herein may refer to any deterministic filter that approximates a Wiener filter. Typically, a Wiener filter is stochastically derived. However, the Wiener filter described herein may also be deterministically derived such that it mathematically adopts the form of a statistical Wiener filter. For example, the filter may be derived by multiplying the inverse filters $\frac{1}{-\omega^2}$ and $\frac{1}{i\omega}$ by their complex conjugates, which results in equations (7) and (8) below.

[0048] In an embodiment, the double integration filter in equation (5) is modified to the following form:

$$\frac{1}{-\omega^2} \approx \left[\frac{-i\omega}{\omega^2+\epsilon}\right]^2 = F_u(\omega) \qquad (7)$$

[0049] In another embodiment, the single integration filter in equation (6) is modified to the following form:

$$\frac{1}{i\omega} \approx \frac{-i\omega}{\omega^2+\varepsilon} = F_v(\omega) \qquad (8)$$

[0050] In the middle term of equations (7) and (8), ε is a damping factor that is selected to provide stability to the double integration and single integration process in the presence of noise. The value of ε may be manually or automatically selected, depending on the implementation. Techniques for selecting a damping factor are described in further detail below.

[0051] In one embodiment, the value of ε may be any suitable constant to dampen undesirable amplification in the amplitude response of the integration filters. For example, the damping effects of ε are illustrated in FIGS. 5 and 6. Referring to FIG. 5, amplitude response 504 shows the amplitude response of Wiener filter 506 when a damping factor of 4000 is selected. Similarly, in FIG. 6, amplitude response 604 shows the amplitude response of Wiener filter 606 when a damping factor of 4000 is selected. In contrast to amplitude responses 502 and 602, amplitude response 504 and 604 are dampened at low frequencies.

[0052] Thus, the damping factor provides stability by not amplifying the low frequency end of the spectrum, thereby reducing the effects of noise, and if chosen properly, produces reliable particle displacement or particle velocity amplitudes and phases. The stability is demonstrated by applying the Wiener filters of equations (7) and (8) to the same noisy synthetic particle acceleration data set shown on trace (a) of either FIG 3 or 4.

[0053] FIG. 7 is a graph illustrating an example calculation of particle displacement data from noisy acceleration data using a deterministic Wiener filter. Referring to FIG. 7, trace (a) 702 shows synthetic ('observed') particle acceleration data, which is the same 'observed' noisy particle acceleration data shown in trace (a) 302. Particle displacement is estimated using this trace and equation (7). The results of this calculation are displayed in trace (e) 710. Trace (d) 708 shows the 'true' particle displacement data, which is the same as trace (d) 308. Trace (f) 712 corresponds to the noise residuals obtained from the

difference of the estimated particle displacement of trace (e) 710 and the 'true' particle displacement of trace (d) 708. Trace (b) 704 shows the calculated particle acceleration data using the forward equation (4) on trace (e) 710. Trace (c) 706 shows the difference between the 'observed' and 'calculated' particle acceleration data.

[0054] FIG. 8 is a graph illustrating an example calculation of particle velocity data form noisy acceleration data using a deterministic Wiener filter. Referring to FIG. 8, trace (a) 802 shows synthetic ('observed') particle acceleration data, which is the same 'observed' noisy particle acceleration data shown in trace (a) 402. Particle velocity is estimated using this trace and equation (8). The results of this calculation are displayed in trace (e) 810. Trace (d) 808 shows the 'true' particle velocity data, which is the same as trace (d) 408. Trace (f) 812 corresponds to the noise residuals obtained from the difference of the estimated particle velocity of trace (e) 810 and the 'true' particle velocity of trace (d) 808. Trace (b) 804 shows the calculated particle acceleration data using the forward equation (4) on trace (e) 810. Trace (c) 806 shows the difference between the 'observed' and 'calculated' particle acceleration data.

[0055] Trace (c) in the above figures corresponds to particle acceleration noise residuals for each case of displacement and velocity calculations. This trace may be used to determine the relevant measurement for goodness of fit between the calculated data and the observed data since particle acceleration is the input data. However, in the presence of noise, a perfect fit does not necessarily imply that the particle displacement or velocity estimates are stable and accurate, as demonstrated in the cases illustrated by FIGS. 3 and 4. Furthermore, the difference shown in trace (f) of the above figures is computed assuming the knowledge of the 'true' particle displacement or velocity, which in real data cases may be unknown or otherwise unavailable. Accordingly, the difference calculation (trace (f)) is included for purposes of illustrating the performance of the deterministic Wiener filters, but may be unavailable, depending on the particular implementation.

[0056] FIG. 9 is a flowchart illustrating a process for calculating particle displacement data from particle acceleration data using a deterministic Wiener filter. In step 902, raw acceleration data is input and a damping factor is defined. This step may include, without limitation, recording or otherwise receiving an acceleration signal from a MEMS or other type of accelerometer that measures particle acceleration. The damping factor defined at this step may be manually or automatically selected, depending on the particular implementation. In an embodiment, a damping is selected to provide stability in the

presence of noise in the particle acceleration data. The process illustrated in FIG. 13 may be used to select the damping factor.

[0057] In step 904, frequency domain acceleration data is determined from the received particle acceleration data. This step may include, without limitation, using a Fast Fourier Transform (FFT) to extract or otherwise estimate components of different frequencies that form the particle acceleration signal received by the accelerometer. For example, the FFT may be used to transform time-domain acceleration data of a form approximated by equation (1) to frequency-domain acceleration data of a form approximated by equation (3).

[0058] In step 906, a squared Wiener filter is used to estimate frequency-domain particle displacement data. In an embodiment, the Wiener filter $F_u(\omega)$ of equation (7) is applied to the frequency-domain acceleration data determined at step 904. Applying the Wiener filter may include, without limitation, performing a convolution operation on the frequency-domain acceleration data using the squared Wiener filter.

[0059] In step 908, time-domain particle displacement data is determined from the frequency-domain particle displacement data. This step may include, without limitation, using a Fast Fourier Transform (FFT) to estimate particle displacement as a function of time based on the component frequencies represented in the frequency-domain particle displacement data.

[0060] In step 910, the particle displacement data is output. Outputting the particle displacement data may include, without limitation, displaying, storing, or printing the particle displacement data, or sending the particle data to another device or application program. Furthermore, the particle displacement data that is output may be used to characterize subterranean geological features associated with the acceleration measurements. For example, the particle displacement data may be used in conjunction with other seismic data to estimate the properties of the Earth's subsurface from reflected seismic waves.

[0061] FIG. 10 is a flowchart illustrating a process for calculating particle velocity data from particle acceleration data using a deterministic Wiener filter. In step 1002, raw acceleration data is input and a damping factor is defined. The same acceleration data set input at step 1002 may also be used in the process illustrated in FIG. 10. Alternatively, the process of FIG. 10 may be independent of the process illustrated in FIG. 9. The damping factor defined at this step may be manually or automatically selected, depending on the

particular implementation. In an embodiment, a damping factor is selected to provide stability in the presence of noise in the particle acceleration data. The process illustrated in FIG. 14 may be used to select the damping factor.

[0062] In step 1004, frequency domain acceleration data is determined from the received particle acceleration data. This step may include without limitation using a Fast Fourier Transform (FFT) to extract or otherwise estimate components of different frequencies that form the particle acceleration signal. For example, the FFT may be used to transform time-domain acceleration data of a form approximated by equation (2) to frequency-domain acceleration data of a form approximated by equation (4).

[0063] In step 1006, a Wiener filter is used to estimate frequency-domain particle velocity data. In an embodiment, the Wiener filter $F_V(\omega)$ of equation (8) is applied to the frequency-domain acceleration data determined at step 1004. Applying the Wiener filter may include, without limitation, performing a convolution operation on the frequency-domain acceleration data using the Wiener filter.

[0064] In step 1008, time-domain particle velocity data is determined from the frequency-domain particle velocity data. This step may include, without limitation, using a Fast Fourier Transform (FFT) to estimate particle velocity as a function of time based on the component frequencies represented in the frequency-domain particle velocity data.

[0065] In step 1010, the particle velocity data is output. Outputting the particle velocity data may include, without limitation, displaying, storing, or printing the particle velocity data, or sending the particle data to another device or application program. Furthermore, the particle velocity data that is output may be used to characterize subterranean geological features associated with the acceleration measurements. For example, the particle velocity data may be used in conjunction with other seismic data to estimate the properties of the Earth's subsurface from reflected seismic waves.

[0066] The techniques described above highlight the use of a Wiener filter to provide stable calculations of particle velocity and particle displacement data from particle acceleration measurements. However, the techniques may also be used to provide stable calculations from different input sources. For example, if the input signal at step 902 is particle velocity data that is measured or otherwise observed, then the Wiener filter of equation (8) may be applied to the particle velocity data to estimate particle displacement.

[0067] FIG. 11 is a table that summarizes the filters described above for the calculation of one measurement (output) from another measurement (input). Referring to FIG. 11, if

the input measurement is acceleration, then velocity may be calculated using the Wiener filter $\frac{-i\omega}{\omega^2+\varepsilon}$ and displacement may be calculated using the squared Wiener filter $\left[\frac{-i\omega}{\omega^2+\epsilon}\right]^2$. If the input measurement is velocity, then acceleration may be calculated using the first derivative filter $i\omega$ and displacement may be calculated using the Wiener filter $\frac{-i\omega}{\omega^2+\varepsilon}$. If the input measurement is displacement, then acceleration may be calculated using the second derivative filter $-\omega^2$ and velocity may be calculated using the first derivative filter $i\omega$.

IV. FILTER OPTIMIZATION FOR STABLE CALCULATIONS

[0068] The choice of the damping factor may determine the effectiveness of the integration Wiener filters highlighted above. According to the techniques described herein, a damping factor may be determined based on a comparison between the observed acceleration data and calculated acceleration data. For example, the observed acceleration data illustrated in trace (a) 702 or trace (a) 802 may be compared to the calculated acceleration data illustrated in trace (b) 704 or trace (b) 804, respectively, to determine whether a threshold is satisfied. If the threshold is satisfied, then the damping factor may be selected as satisfactory. If the threshold is not satisfied, then a different damping factor may be selected.

[0069] FIG. 12 is a flowchart illustrating a process for selecting a damping factor when estimating particle displacement. In step 1202, raw acceleration data is input similar to step 902 above and a large starting value for ε is initially selected.

[0070] In step 1204, frequency-domain acceleration data is determined from the particle acceleration data received such as described in step 904 above.

[0071] In step 1206, a squared Wiener filter is used to estimate frequency-domain particle displacement data. In an embodiment, the Wiener filter $F_u(\omega)$ of equation (7) is applied to the frequency-domain acceleration data determined such as described in step 906 above.

[0072] In step 1208, frequency-domain acceleration data is calculated based on the frequency-domain particle displacement data. To calculate the acceleration data, this step may include performing a second derivative operation with respect to the frequency-

domain particle displacement data. For example, the second derivative filter of equation (3), $-\omega^2$, may be applied to the frequency-domain particle displacement data.

[0073] In step 1210, the energy between of the observed and calculated particle acceleration data is estimated and compared to a supplied minimum threshold value. In an embodiment, the energy data may be calculated and compared based on the following formulas:

$$EA_{obs} = \frac{1}{Nnyq}\sum_0^{\omega nyq} A^2(\omega) \qquad (9);$$

$$EA_{calc} = \frac{1}{Nnyq}\sum_0^{\omega nyq} A'^2(\omega) \qquad (10);$$

and

$$E_{rat} = 20\log_{10}\left(\frac{EA_{obs}}{EA_{calc}}\right) \qquad (11);$$

where ωnyq represents the Nyquist angular frequency, N_{nyq} represents the number of angular frequency samples in the spectrum up to the Nyquist frequency, S represents a summation operator, EA_{obs} represents the energy of the observed acceleration, EA_{calc} represents the energy of the calculated acceleration, and E_{rat} represents the residual energy. The residual energy may be compared to a supplied threshold value in decibels. If E_{rat} is larger than the threshold then the damping factor may be adjusted. If E_{rat} is smaller than the threshold, then the current value of the damping factor may be saved for subsequent processing.

[0074] When the threshold is not satisfied at step 1210, then in step 1212, the value of the damping factor is adjusted to a lower value. The lower value may be manually chosen or automatically selected based on any suitable criteria. Once the lower value is selected, the process repeats with step 1206, and the particle displacement data is recalculated using the squared Wiener filter modified with the new damping factor. The process repeats until a satisfactory damping factor is determined.

[0075] If the threshold is satisfied at step 1210, then the most recent damping factor may be selected as the damping factor and, at step 1214, time-domain particle displacement data is determined from the frequency-domain particle displacement data that was calculated at step 1206 with the damping factor. The time-domain particle displacement data may be calculated in the same manner described in step 908.

[0076] In step 1216, the particle displacement value is output such as described in step 910.

[0077] FIG. 13 is a flowchart illustrating a process for estimating a damping factor when estimating particle velocity. In step 1302, raw acceleration data is input similar to step 1002 and a large starting value for ε is initially selected.

[0078] In step 1304, frequency-domain acceleration data is determined from the particle acceleration data received such as described in step 1004 above.

[0079] In step 1306, a Wiener filter is used to estimate frequency-domain particle velocity data. In an embodiment, the Wiener filter $F_v(\omega)$ of equation (8) is applied to the frequency-domain acceleration data such as described in step 1006 above.

[0080] In step 1308, frequency-domain acceleration data is calculated based on the frequency-domain particle velocity data. To calculate the acceleration data, this step may include performing a first derivative operation with respect to the frequency-domain particle velocity data. For example, the first derivative filter of equation (4), $i\omega$, may be applied to the frequency-domain particle velocity data.

[0081] In step 1310, the energy between of the observed and calculated particle acceleration is estimated and compared to a supplied minimum threshold value. In an embodiment, the energy data may be calculated and compared based on the formulas specified in equations (9), (10), and (11). If E_{rat} is larger than the threshold then the damping factor may be adjusted. If E_{rat} is smaller than the threshold, then the current value of the damping factor may be saved for subsequent processing.

[0082] When the threshold is not satisfied at step 1310, then in step 1312, the value of the damping factor is adjusted to a lower value. The lower value may be manually chosen or automatically selected using any suitable mechanism. Once the lower value is selected, the process repeats with step 1306, and the particle velocity data is recalculated using the Wiener filter and the new damping factor. The process repeats until a satisfactory damping factor is determined.

[0083] If the threshold is satisfied at step 1310, then the most recent damping factor may be selected and, at step 1314, time-domain particle velocity data is determined from the frequency-domain particle velocity data that was calculated at step 1306 with the selected damping factor. The time-domain particle velocity data may be calculated in the same manner described in step 1008.

[0084] In step 1316, the particle velocity value is output such as described in step 1010.

[0085] If particle velocity is the input data rather than particle acceleration data, then the damping value may be calculated based on the difference between observed and calculated particle velocity. For example, the following formulas may be used:

$$EV_{obs} = \frac{1}{Nnyq}\sum_{0}^{\omega nyq} V(\omega) \qquad (12);$$

$$EV_{calc} = \frac{1}{Nnyq}\sum_{0}^{\omega nyq} V'^{2}(\omega) \qquad (13);$$

and

$$E_{rat} = 20\log_{10}\left(\frac{EV_{obs}}{EV_{calc}}\right) \qquad (14);$$

where ωnyq represents the Nyquist angular frequency, N_{nyq} represents the number of angular frequency samples in the spectrum up to the Nyquist frequency, S represents a summation operator, EV_{obs} represents the energy of the observed velocity, EV_{calc} represents the energy of the calculated velocity, and E_{rat} represents the residual energy. The residual energy may be compared to a supplied threshold value in decibels. If E_{rat} is larger than the threshold then the damping factor may be adjusted. If E_{rat} is smaller than the threshold, then the current value of the damping factor may be selected and saved for subsequent processing including estimating particle displacement from the observed particle velocity data.

[0086] An example selection of ε is illustrated in FIGS. 14 and 15. FIG. 14 is a graph showing different scanned damping factors. In this case, the scanned damping factors include the following values: 10^6, 10^5, 10^4, 4000, 10^3, 10^2, 10 and 10^{-5}. Plot 1402 illustrates the different values that were tested to determine a satisfactory value. As indicated by the graph point 1404, the value selected in this case was 4000.

[0087] FIG. 15 is a graph illustrating the results of different scanned damping factors when estimating particle velocity. Graph (a) 1502 shows acceleration traces that were calculated from the estimated particle velocity traces. As illustrated by this graph, for very large ε values, the energy in the calculated acceleration trace is small, indicating inadequate results. As ε increases in value, the energy in the calculated acceleration trace approaches that of the observed acceleration trace. Graph (b) 1504 shows traces that correspond between the difference in the 'observed' and 'calculated' acceleration data. In this example, the value of 4000 was selected for ε when a threshold of 0.2 decibels was reached. Graph (c) 1506 shows the equivalent calculated particle velocity traces. As ε

decreases, the result becomes unstable. This is evident by the presence of very large amplitude and low frequency energy.

[0088] If both the particle velocity and the particle displacement are being calculated from the same set of acceleration data, then in one embodiment, the same damping factor may be used for the calculations. The damping factor may be determined using either the process illustrated in FIG. 12 or the process illustrated in FIG. 13. In an alternative embodiment, the damping factor may be independently derived for particle displacement and particle velocity according to the techniques described above. When independently derived, the damping factor may be different for the particle displacement calculation and the particle velocity calculation. Thus, a first damping factor may be selected to calculate particle displacement from the acceleration data and a second damping factor, different than the first, may be selected to calculate particle velocity from the same set of acceleration data.

[0089] Satisfactory damping factor(s) may also vary between different sets of acceleration data; therefore, it may be beneficial to determine and select the damping factor(s) on a per data set basis. Accordingly, in one embodiment, the damping factor may be calculated according to the processes illustrated in FIGS. 12 and/or 13 for each distinct set of acceleration data received. In an alternative embodiment, to save on computational resources, damping factor(s) selected for one set of acceleration data may be used for particle displacement and/or particle velocity calculations from other sets of acceleration data. In this case, the damping factors may be periodically updated to minimize the risk of any instability during the integration process.

V. IMPLEMENTATION EXAMPLES

[0090] According to one embodiment, the techniques described herein are implemented by one or more special-purpose computing devices. The special-purpose computing devices may be hard-wired to perform the techniques, or may include digital electronic devices such as one or more application-specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs) that are persistently programmed to perform the techniques, or may include one or more general purpose hardware processors programmed to perform the techniques pursuant to program instructions in firmware, memory, other storage, or a combination. Such special-purpose computing devices may also combine custom hard-wired logic, ASICs, or FPGAs with custom programming to accomplish the

techniques. The special-purpose computing devices may be desktop computer systems, portable computer systems, handheld devices, networking devices or any other device that incorporates hard-wired and/or program logic to implement the techniques.

[0091] For example, FIG. 16 is a block diagram that illustrates an example computer system 1600 upon which an embodiment of the invention may be implemented. Computer system 1600 includes a bus 1602 or other communication mechanism for communicating information, and a hardware processor 1604 coupled with bus 1602 for processing information. Hardware processor 1604 may be, for example, a general purpose microprocessor.

[0092] Computer system 1600 also includes a main memory 1606, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 1602 for storing information and instructions to be executed by processor 1604. Main memory 1606 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 1604. Such instructions, when stored in non-transitory storage media accessible to processor 1604, render computer system 1600 into a special-purpose machine that is customized to perform the operations specified in the instructions.

[0093] Computer system 1600 further includes a read only memory (ROM) 1608 or other static storage device coupled to bus 1602 for storing static information and instructions for processor 1604. A storage device 1610, such as a magnetic disk or optical disk, is provided and coupled to bus 1602 for storing information and instructions.

[0094] Computer system 1600 may be coupled via bus 1602 to a display 1612, such as a cathode ray tube (CRT), for displaying information to a computer user. Although bus 1602 is illustrated as a single bus, bus 1602 may comprise one or more buses. For example, bus 1602 may include without limitation a control bus by which processor 1604 controls other devices within computer system 1600, an address bus by which processor 1604 specifies memory locations of instructions for execution, or any other type of bus for transferring data or signals between components of computer system 1600.

[0095] An input device 1614, including alphanumeric and other keys, is coupled to bus 1602 for communicating information and command selections to processor 1604. Another type of user input device is cursor control 1616, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 1604 and for controlling cursor movement on display 1612. This input device

typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0096] Computer system 1600 may implement the techniques described herein using customized hard-wired logic, one or more ASICs or FPGAs, firmware and/or program logic which in combination with the computer system causes or programs computer system 1600 to be a special-purpose machine. According to one embodiment, the techniques herein are performed by computer system 1600 in response to processor 1604 executing one or more sequences of one or more instructions contained in main memory 1606. Such instructions may be read into main memory 1606 from another storage medium, such as storage device 1610. Execution of the sequences of instructions contained in main memory 1606 causes processor 1604 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions.

[0097] The term "storage media" as used herein refers to any non-transitory media that store data and/or instructions that cause a machine to operate in a specific fashion. Such storage media may comprise non-volatile media and/or volatile media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 1610. Volatile media includes dynamic memory, such as main memory 1606. Common forms of storage media include, for example, a floppy disk, a flexible disk, hard disk, solid state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium, any physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, NVRAM, any other memory chip or cartridge.

[0098] Storage media is distinct from but may be used in conjunction with transmission media. Transmission media participates in transferring information between storage media. For example, transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 1602. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0099] Various forms of media may be involved in carrying one or more sequences of one or more instructions to processor 1604 for execution. For example, the instructions may initially be carried on a magnetic disk or solid state drive of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system

1600 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 1602. Bus 1602 carries the data to main memory 1606, from which processor 1604 retrieves and executes the instructions. The instructions received by main memory 1606 may optionally be stored on storage device 1610 either before or after execution by processor 1604.

[0100] Computer system 1600 also includes a communication interface 1618 coupled to bus 1602. Communication interface 1618 provides a two-way data communication coupling to a network link 1620 that is connected to a local network 1622. For example, communication interface 1618 may be an integrated services digital network (ISDN) card, cable modem, satellite modem, or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 1618 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 1618 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0101] Network link 1620 typically provides data communication through one or more networks to other data devices. For example, network link 1620 may provide a connection through local network 1622 to a host computer 1624 or to data equipment operated by an Internet Service Provider (ISP) 1626. ISP 1626 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 1628. Local network 1622 and Internet 1628 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 1620 and through communication interface 1618, which carry the digital data to and from computer system 1600, are example forms of transmission media.

[0102] Computer system 1600 can send messages and receive data, including program code, through the network(s), network link 1620 and communication interface 1618. In the Internet example, a server 1630 might transmit a requested code for an application program through Internet 1628, ISP 1626, local network 1622 and communication interface 1618.

[0103] The received code may be executed by processor 1604 as it is received, and/or stored in storage device 1610, or other non-volatile storage for later execution.

[0104] Exploration seismology is routinely performed both on land and at sea. At sea, seismic survey ships deploy streamers behind the ship as depicted in FIG. 17, which is an illustration of a side view of an example marine seismic survey environment in which seismic data may be acquired. Each streamer 1710 trails behind ship 1700 as the ship moves forward (in the direction of arrow 1702), and each streamer includes multiple evenly-spaced receivers 1714. Each streamer 1710 may further include one or more diverters 1718 and programmable depth controllers that pull the streamer out to an operating offset distance from the ship's path and down to an operating depth.

[0105] Streamers 1710 may be up to several kilometers long, and are usually constructed in sections 25 to 100 meters in length that include groups of up to 35 or more uniformly spaced receivers. Each streamer 1710 includes electrical or fiber-optic cabling for interconnecting receivers 1714 and the seismic equipment on ship 1700. Data may be digitized near receivers 1714 and transmitted to ship 1700 through the cabling at rates of 7 (or more) million bits of data per second.

[0106] As depicted in FIG. 17, seismic survey ship 1700 also tows a source 1712. Source 1712 may be an impulse source or a vibratory source. Receivers 1714 may include hydrophones and/or geophones. Source 1712 and receivers 1714 typically deploy below the ocean's surface 1704. Processing equipment aboard the ship controls the operation of the source and receivers and records the acquired data.

[0107] Seismic surveys provide data for imaging below the ocean surface 1704 and include subsurface structures such as structure 1706, which lies below the ocean floor 1708. Certain seismic characteristics of recorded seismic data are indicative of oil and/or gas reservoirs.

[0108] To image the subsurface structure 1706, source 1712 emits seismic waves 1716 that are reflected where there are changes in acoustic impedance contrast due to subsurface structure 1706 (and other subsurface structures). The reflected waves are detected by a pattern of receivers 1714. By recording, among other things, the elapsed time for the seismic waves 1716 to travel from source 1712 to subsurface structure 1706 to receivers 1714, an image of subsurface structure 1706 can be obtained after appropriate data processing. Data processing may include the techniques for calculating particle acceleration, velocity, and/or displacement data described above.

VI. EXTENSIONS AND ALTERNATIVES

[0109] In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. The sole and exclusive indicator of the scope of the invention, and what is intended by the applicants to be the scope of the invention, is the literal and equivalent scope of the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction.

CLAIMS

What is claimed is:

1. An apparatus comprising:

an interface configured to receive particle acceleration data;

one or more processors;

one or more storage media storing instructions, which, when executed by the one or more processors, cause the apparatus to perform:

determining one or more of particle velocity data or particle displacement data, based upon the particle acceleration data, by processing the particle acceleration data using a filter that at least approximates a Wiener filter and that uses one or more damping factors selected to provide stability in the presence of noise in the particle acceleration data.

2. The apparatus of Claim 1, wherein the instructions, when executed, further cause the apparatus to perform: determining a particular damping factor based on the received particle acceleration data.

3. The apparatus of Claim 2, wherein instructions for determining the particular damping factor comprise instructions for:

determining, based on the particle velocity or particle displacement data, calculated acceleration data;

determining whether a threshold is satisfied by comparing the calculated acceleration data with the particle acceleration data;

in response to determining that the threshold is not satisfied, adjusting a damping factor of the one or more damping factors;

in response to determining that the threshold is satisfied, selecting the damping factor as the particular damping factor.

4. The apparatus of Claim 3, wherein instructions for comparing the calculated acceleration data with the particle acceleration data comprise instructions for:

determining a first energy value from the calculated acceleration data;

determining a second energy value from the particle acceleration data;

determining a difference between the first energy value and the second energy value;

wherein the threshold is satisfied if the difference is less than a threshold difference.

5. The apparatus of Claim 1, wherein the instructions, when executed, further cause the apparatus to perform:

determining both the particle velocity data and the particle displacement data, based upon the particle acceleration data, by processing the particle acceleration data using the filter that at least approximates a Wiener filter and that uses one or more damping factors selected to provide stability in the presence of noise in the particle acceleration data;

wherein determining the particle velocity data comprises processing the particle acceleration data using the filter and a first damping factor;

wherein determining the particle displacement data comprises processing the particle acceleration data using the filter and a second damping factor;

wherein the first damping factor is different than the second damping factor.

6. The apparatus of Claim 1, wherein the Wiener filter is of a form represented by $F(\omega) = \frac{-i\omega}{\omega^2+\varepsilon}$; wherein $F(\omega)$ represents the Wiener filter, i represents an imaginary number, ω represents an angular frequency, and ε represents a damping factor.

7. The apparatus of Claim 1, wherein the instructions, when executed, further cause the apparatus to perform: determining a damping factor for each distinct set of acceleration data that is received through the interface.

8. One or more non-transitory storage media storing instructions which, when executed by one or more computing devices, cause:

receiving particle acceleration data;

determining one or more of particle velocity data or particle displacement data, based upon the particle acceleration data, by processing the particle acceleration data using a filter that at least approximates a Wiener filter and that uses one or

more damping factors selected to provide stability in the presence of noise in the particle acceleration data.

9. The one or more non-transitory storage media of Claim 8, wherein the instructions, when executed, further cause: determining a particular damping factor based on the received particle acceleration data.

10. The one or more non-transitory storage media of Claim 9, wherein instructions for determining the particular damping factor comprise instructions for:

determining, based on the particle velocity or particle displacement data, calculated acceleration data;

determining whether a threshold is satisfied by comparing the calculated acceleration data with the particle acceleration data;

in response to determining that the threshold is not satisfied, adjusting a damping factor of the one or more damping factors;

in response to determining that the threshold is satisfied, selecting the damping factor as the particular damping factor.

11. The one or more non-transitory storage media of Claim 10, wherein instructions for comparing the calculated acceleration data with the particle acceleration data comprise instructions for:

determining a first energy value from the calculated acceleration data;

determining a second energy value from the particle acceleration data;

determining a difference between the first energy value and the second energy value;

wherein the threshold is satisfied if the difference is less than a threshold difference.

12. The one or more non-transitory storage media of Claim 8, wherein the instructions, when executed, further cause:

determining both the particle velocity data and the particle displacement data, based upon the particle acceleration data, by processing the particle acceleration data using the filter that at least approximates a Wiener filter and that uses one or more damping factors selected to provide stability in the presence of noise in the particle acceleration data;

wherein determining the particle velocity data comprises processing the particle acceleration data using the filter and a first damping factor;

wherein determining the particle displacement data comprises processing the particle acceleration data using the filter and a second damping factor;

wherein the first damping factor is different than the second damping factor.

13. The one or more non-transitory storage media of Claim 8, wherein the Wiener filter is of a form represented by $F(\omega) = \frac{-i\omega}{\omega^2+\varepsilon}$; wherein $F(\omega)$ represents the Wiener filter, i represents an imaginary number, ω represents an angular frequency, and ε represents a damping factor.

14. The one or more non-transitory storage media of Claim 8, wherein the instructions, when executed, further cause: characterizing subterranean geological features based on the particle velocity data or the particle displacement data.

15. A method comprising:

receiving particle acceleration data;

determining one or more of particle velocity data or particle displacement data, based upon the particle acceleration data, by processing the particle acceleration data using a filter that at least approximates a Wiener filter and that uses one or more damping factors selected to provide stability in the presence of noise in the particle acceleration data.

16. The method of Claim 15, further comprising determining a particular damping factor based on the received particle acceleration data.

17. The method of Claim 16, wherein determining the particular damping factor comprises:

determining, based on the particle velocity or particle displacement data, calculated acceleration data;

determining whether a threshold is satisfied by comparing the calculated acceleration data with the particle acceleration data;

in response to determining that the threshold is not satisfied, adjusting a damping factor of the one or more damping factors;

in response to determining that the threshold is satisfied, selecting the damping factor as the particular damping factor.

18. The method of Claim 17, wherein comparing the calculated acceleration data with the particle acceleration data comprises:

determining a first energy value from the calculated acceleration data;

determining a second energy value from the particle acceleration data;

determining a difference between the first energy value and the second energy value;

wherein the threshold is satisfied if the difference is less than a threshold difference.

19. The method of Claim 15, further comprising:

determining both the particle velocity data and the particle displacement data, based upon the particle acceleration data, by processing the particle acceleration data using the filter that at least approximates a Wiener filter and that uses one or more damping factors selected to provide stability in the presence of noise in the particle acceleration data;

wherein determining the particle velocity data comprises processing the particle acceleration data using the filter and a first damping factor;

wherein determining the particle displacement data comprises processing the particle acceleration data using the filter and a second damping factor;

wherein the first damping factor is different than the second damping factor.

20. The method of Claim 15, wherein the Wiener filter is of a form represented by $F(\omega) = \frac{-i\omega}{\omega^2+\varepsilon}$; wherein $F(\omega)$ represents the Wiener filter, i represents an imaginary number, ω represents an angular frequency, and ε represents a damping factor.

21. A method comprising:

applying a filter including one or more damping factors to particle acceleration data to determine one or more of particle velocity data or particle displacement data;

in response to determining that stability of the particle velocity data or the particle displacement data does not satisfy a threshold, adjusting one or more of the one or more damping factors of the filter;

wherein the method is performed by one or more computing devices.

22. The method of Claim 21, further comprising repeating said steps of applying and adjusting until the stability of the particle velocity data or the particle displacement data satisfies the threshold.

23. The method of Claim 21, wherein determining that the particle displacement data does not satisfy a threshold comprises:

after applying the filter including one or more damping factors to the particle acceleration data to determine one or more of particle velocity data or particle displacement data:

calculating second particle acceleration data from the particle velocity data or particle displacement data;

determining that a difference between a first energy value of the particle acceleration data and a second energy value of the second particle acceleration data does not satisfy a threshold.

24. The method of Claim 23 wherein the difference between the first energy value and the second energy value does not satisfy a threshold when greater than a threshold value.

25. The method of Claim 21, wherein the filter at least approximates a Wiener filter.

26. The method of Claim 21 further comprising measuring the particle acceleration data using one or more seismic sensors.