Determining rigid body motion from accelerometer data through the square-root of a negative semi-definite tensor, with applications in mild traumatic brain injury

Yang Wan, Alice Lux Fawzi, Haneesh Kesari*

Brown University School of Engineering, 184 Hope St., Providence, RI, USA

Abstract

Mild Traumatic Brain Injuries (mTBI) are caused by violent head motions or impacts. Most mTBI prevention strategies explicitly or implicitly rely on a "brain injury criterion". A brain injury criterion takes some descriptor of the head's motion as input and yields a prediction for that motion's potential for causing mTBI as the output. The inputs are descriptors of the head's motion that are usually synthesized from accelerometer and gyroscope data. In the context of brain injury criterion, the head is modeled as a rigid body. We present an algorithm for determining the complete motion of the head using data from only four head mounted tri-axial accelerometers. In contrast to inertial measurement unit based algorithms for determining rigid body motion, the presented algorithm does not depend on data from gyroscopes; which consume much more power than accelerometers. Several algorithms that also make use of data from only accelerometers already exist. However, those algorithms, except for the recently presented AO (accelerometer-only) algorithm [Rahaman MM, Fang W, Fawzi AL, Wan Y, Kesari H (2020): *J Mech Phys Solids* 104014], give the rigid body's acceleration field in terms of the body frame, which in general is unknown. Compared to the AO-algorithm the presented algorithm is much more insensitive to bias type errors, such as those that arise from inaccurate measurement of sensor positions and orientations.

Keywords: mTBI; Tensor square root; Rigid body motion; Accelerometers; Inertial navigation; Continuum mechanics

1. Introduction

Mild Traumatic Brain Injury (mTBI) is the most common injury among military personnel and it is estimated that as many as 600 per 100,000 people experience mTBIs each year across the world [1, 2]. Mild Traumatic Brain Injuries are caused by violent head motions, that may occur from intense blunt impacts to the head in contact sports, motor vehicle crashes, falls following blasts, etc. In mTBI, the motion of the head causes the soft tissue of the brain to deform. The magnitude and time rate of brain deformation can cause brain cells to die [3, 4, 5, 6, 7].

There have been many strategies aimed at preventing mTBI. In sports, new rules aim to modify player behavior in order to decrease or eliminate exposure to blunt impacts [8]. Helmets and neck collars are examples of equipment that can alter the motion experienced by the head. A jugular vein compression collar aims to change the stiffness of the brain making it less susceptible to injury [9].

Most mTBI prevention strategies explicitly or implicitly rely on a "brain injury criterion" for their effective synthesis, implementation, and evaluation. A brain injury criterion takes some descriptor of the head's motion as input and yields a prediction for that motion's potential for causing mTBI as the output.

When we refer to any aspect of the head's motion, we, in fact, are referring to that aspect as it pertains to the skull's motion; since it is the skull's motion that is, at least currently, observable and quantifiable in the field, either using video recording equipment or inertial sensor systems. The Young's modulus of bone from human skulls generally lies in the 2–13 GPa range [10, 11, 12]. In comparison, brain tissue is extremely compliant. Recent indentation tests on brain slices that were kept hydrated

---

show that the Young's modulus of brain tissue lies in the 1–2 kPa range (white matter 1.9 ± 0.6 kPa, and gray matter 1.4 ± 0.3 kPa) [13]. Due to this large disparity between the skull's and the brain's stiffnesses, in biomechanical investigations of mTBI the skull is usually modeled as a rigid body [14, 15]. Thus, inputs to the brain injury criteria are rigid body motion descriptors, such as angular velocity time series, translational acceleration times series, etc., or a combination of such time series.

Rigid body motion can be thought of as a composition of translatory and rotatory motions. In initial brain injury criteria the focus was on the head's translatory motion. Two of the first published injury criteria are the Gadd Severity Index (SI) and the Head Injury Criterion (HIC) [16, 17]. Both SI and HIC ignore the head's rotations and take the head's translational acceleration as their input. Later, however, it was realized that in the context of mTBI the head's rotations play an even more important role in causing injury than its translations. The first brain injury criterion to take the rotational aspect of the head's motion into consideration was GAMBIT [18]. The input to GAMBIT is the tuple of center-of-mass-acceleration and angular-acceleration time series. Following the development of GAMBIT, brain injury criteria that use descriptors that only depend on the head's rotational motion as inputs have also been put forward. One such criteria is the Brain Injury Criteria (BrIC) [19]. Aiming to compliment HIC, BrIC only uses the head's angular velocity time series as input. We also note that there is currently significant activity in applying finite element modeling using 2D/3D anatomically consistent discrete geometry head models to evaluate or develop new brain injury criteria [20, 21, 22].

Irrespective of which existing, or yet to be developed, brain injury criterion will be used in the future, its successful application will hinge on the availability of a robust algorithm for constructing the motion descriptor that the criterion takes as input from measurable data. Currently, different algorithms are used to obtain the descriptors taken by the injury criteria as inputs. The inputs to GAMBIT can be obtained from the measurements of one tri-axial accelerometer and one tri-axial gyroscope mounted in a mouthguard [23] if the center-of-mass-acceleration and angular-acceleration are obtained by processing the data using the algorithm in [24]. In another example the input to BrIC (i.e., angular velocity) is prepared by numerically integrating the angular acceleration, which is determined by applying the 6DOF algorithm [25] to the data from 12 single-axis accelerometers mounted in a helmet [26]. Interestingly the inputs to most of the currently employed brain injury criteria can be prepared from the knowledge of a few key rigid body motion descriptors. To make this idea more concrete, consider the following equation, which is often used to describe rigid body motion,

$$\mathbf{x}(\tau) = \mathbf{Q}(\tau)\mathbf{X} + \mathbf{c}(\tau). \tag{1.1}$$

In (1.1) τ is a real number that denotes a non-dimensional time instant; $\mathbf{X}$ is a column matrix of real numbers that denotes the initial position vector of a rigid body material particle; $\mathbf{x}(\tau)$ is the column matrix of real numbers that denotes that material particle's position vector at the time instance τ; $\mathbf{Q}(\tau)$ is a time dependent square matrix of real numbers with positive determinant whose transpose equals its inverse; and $\mathbf{c}(\tau)$ is a time dependent column matrix of real numbers. The matrix $\mathbf{Q}(\tau)$ quantifies the rotation or orientation of the rigid body at the time instance τ, while $\mathbf{c}(\tau)$ quantifies the rigid body's translation at that time instance. The inputs to most current brain injury criteria can be computed from a knowledge of the maps $\mathbf{Q}$ and $\mathbf{c}$ and their first and second-order time derivatives, i.e., $\mathbf{Q}'$, $\mathbf{c}'$, $\mathbf{Q}''$, $\mathbf{c}''$. In this manuscript we present an algorithm for determining these maps and their derivatives using data from only four tri-axial accelerometers. The only requirement on the accelerometers' positions is that all the accelerometers not lie on the same plane.

The presented algorithm has some similarities to the one recently presented by Rahaman *et al.* [27][1], which is referred to as the AO (accelerometer-only) algorithm. For reasons that will become clear shortly, we refer to the algorithm that we present in this manuscript as the $\sqrt{\text{AO}}$-algorithm. The AO-algorithm also presents a framework for completely determining the rigid body's motion, i.e., for constructing the maps $\mathbf{Q}$ and $\mathbf{c}$ and their time derivatives, using data only from four tri-axial accelerometers not lying on the same plane. The reasons why it is required that the accelerometers not lie on a plane in the $\sqrt{\text{AO}}$ and the AO algorithms are the same. They are discussed in [27, §2.1, §3.1, §5]. The $\sqrt{\text{AO}}$-algorithm has all the advantages of the AO-algorithm.

There are existing algorithms for completely determining a rigid body's motion using sensor data.

---

[1]A graphical user interface for applying the AO algorithm to different types of data sets and visualizing its results is freely available [28]

However, these algorithms take data from sensor systems called inertial measurement units (IMUs). These units contain one or more gyroscopes. One of the primary advantages of the AO and $\sqrt{\text{AO}}$ algorithms is their non-dependence on gyroscopes. For a detailed discussion on why accelerometers are preferable over gyroscopes in the context of mTBI please see §1 in [27]. Briefly, gyroscopes' power requirements are much higher than those of accelerometers (gyroscopes consume approximately 25 times more power than accelerometers [29]), and algorithms that aim to construct descriptors of a rigid body's acceleration using data from gyroscopes add a significant amount of noise to those descriptors [30, 31].

Several algorithms exist for constructing inputs to brain injury criteria that that too only make use of data from accelerometers (Padgaonkar *et al.* [32], Genin *et al.* [33] and Naunheim *et al.* [34]). These algorithms, however, give much more limited information than is given by the AO and the $\sqrt{\text{AO}}$ algorithms. For example, all these algorithms give the rigid body's acceleration field in terms of the body frame, which is a set of vectors that are attached to the rigid body, and hence move with it. These algorithms do not provide any information of how the body frame is oriented in space. However, that information is critical for constructing inputs for the upcoming finite element based brain injury criteria. The AO and the $\sqrt{\text{AO}}$ algorithms provide complete information of how the body frame is oriented in space. See §1 in [27] for further discussion on the advantages of the AO and the $\sqrt{\text{AO}}$ algorithms over other algorithms that also make use of only accelerometer data.

Despite its many advantages we note that the AO-algorithm has one critical limitation. It is quite sensitive to bias type errors in the accelerometer data. Bias type errors are distinct from random errors in that they do not arise as a consequence of stochastic processes. For accelerometers, bias type errors can arise as a consequence of inaccurately defining sensor position and orientation (see Fig. 1). As we explain below, the advantage of the $\sqrt{\text{AO}}$-algorithm over the AO-algorithm is that it is far less sensitive to bias type errors than the AO-algorithm.

One of the critical steps in the $\sqrt{\text{AO}}$ and the AO algorithms is the determination of the map $\tau \mapsto \overline{\mathbf{W}}(\tau)$. Here $\overline{\mathbf{W}}(\tau)$ is time dependent skew-symmetric matrix of real numbers that is related to the rigid body's angular velocity. In the AO-algorithm $\overline{\mathbf{W}}$ is determined by numerically integrating the equation ([27, §3.12])

$$\overline{\mathbf{W}}'(\tau) = \text{skew part of } \mathbf{P}(\tau). \tag{1.2}$$

Here $\mathbf{P}(\tau)$ is a square matrix of real numbers that is to be computed from the accelerometers' data, relative locations, and orientations. Due to numerical integration any bias type errors in $\mathbf{P}$ will give rise to errors in $\overline{\mathbf{W}}$ that grow with time. In the $\sqrt{\text{AO}}$-algorithm we alternatively determine $\overline{\mathbf{W}}$ by taking the square-root of the equation

$$\overline{\mathbf{W}}(\tau)\overline{\mathbf{W}}(\tau) = \text{symmetric part of } \mathbf{P}(\tau). \tag{1.3}$$

We derive (1.3) in Appendix B. Due to the elimination of the numerical integration step associated with the solution of (1.2), the $\sqrt{\text{AO}}$-algorithm gives much better persistent accuracy over time when applied to the data containing bias type errors, compared to the AO-algorithm.

In §2 we present the mathematics and mechanics of rigid body motion from [27, §2] that is needed for the development of the $\sqrt{\text{AO}}$-algorithm. In §3 we review the AO-algorithm as preparation for the development of the $\sqrt{\text{AO}}$-algorithm. In §4 we detail the $\sqrt{\text{AO}}$-algorithm and present a procedure for taking the square root of (1.3). In §5 we check the validity and robustness of the $\sqrt{\text{AO}}$-algorithm. We do so by feeding in virtual accelerometer data, to which differing amounts of bias and noise type errors have been added, to both the $\sqrt{\text{AO}}$ and AO algorithms and comparing their predictions. Using those predictions in §6 we show that the $\sqrt{\text{AO}}$-algorithm is less sensitive to bias type errors than the AO-algorithm. We make a few concluding remarks in §7.

2. Preliminary mathematics and kinematics of rigid body motion

In this section we briefly recapitulate the mathematics and kinematics of rigid body motion from [27, §2] that are needed for the development of the proposed $\sqrt{\text{AO}}$-algorithm.

2.1. Notation

Let $\mathbb{E}$ be a finite dimensional, oriented, Hilbert space, i.e., a Euclidean vector space. The Euclidean point space $\mathscr{E}$ has $\mathbb{E}$ as its associated vector space. Let $o \in \mathscr{E}$ be $\mathscr{E}$'s origin. The spaces $\mathbb{E}$ and $\mathscr{E}$ are related to each other such that for any point $x \in \mathscr{E}$ there exists a vector $\boldsymbol{x} \in \mathbb{E}$ such that $o + \boldsymbol{x} = x$. The
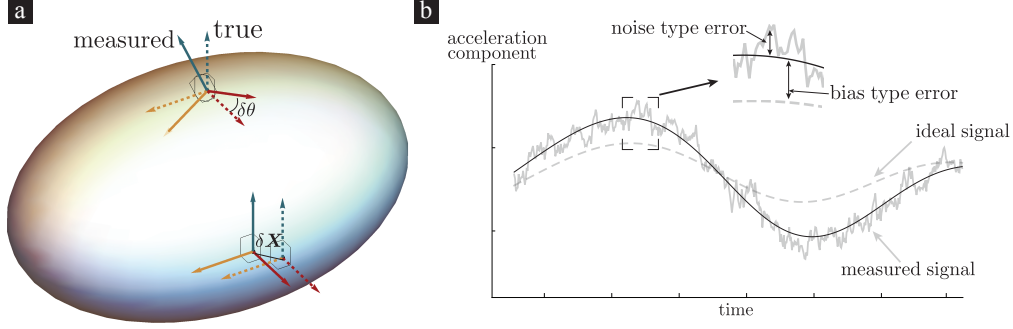
Figure 1: Bias and noise type errors in acceleration component measurements. When there are errors $\delta\boldsymbol{X}$ and $\delta\theta$ in accurately defining an accelerometer's position and orientation, respectively (see (a)), bias type errors can occur in the measurement of acceleration components (see (b)). Noise type errors in the acceleration component measurements are usually a consequence of seismic, electrical, and other types of noise.

topological space $\mathscr{B}$ serves as our model for a rigid body that executes its motion in $\mathscr{E}$. For that reason, we refer to $\mathbb{E}$ and $\mathscr{E}$ as the physical Euclidean vector space and point space, respectively. The spaces $\mathbb{E}_\mathrm{R}$ and $\mathscr{E}_\mathrm{R}$ are another pair of Euclidean vector and point spaces, respectively, that are related to each other in the same way that $\mathbb{E}$ and $\mathscr{E}$ are related to each other. We refer to $\mathbb{E}_\mathrm{R}$ and $\mathscr{E}_\mathrm{R}$ as the reference Euclidean vector and point spaces, respectively. The spaces $\mathbb{E}$, $\mathscr{E}$, $\mathbb{E}_\mathrm{R}$, and $\mathscr{E}_\mathrm{R}$ have the same dimension, which we denote as n_sd. The dimension of $\mathscr{B}$ is less than or equal to n_sd. We call a select continuous, injective map from $\mathscr{B}$ into $\mathbb{E}_\mathrm{R}$ the reference configuration and denote it as $\boldsymbol{\kappa}_\mathrm{R}$. The elements of $\mathscr{B}$ are called material particles. We call $\boldsymbol{X} = \boldsymbol{\kappa}_\mathrm{R}(\mathscr{X})$, where $\mathscr{X} \in \mathscr{B}$, the reference position vector of the material particle $\mathscr{X}$, and we call the set $\boldsymbol{\kappa}_\mathrm{R}(\mathscr{B}) = \left\{ \boldsymbol{\kappa}_\mathrm{R}(\mathscr{X}) \in \mathbb{E}_\mathrm{R} \,\middle|\, \mathscr{X} \in \mathscr{B} \right\}$ the reference body (see Fig. 2). When we refer to $\boldsymbol{X}$ as a material particle we in fact mean the material particle $\boldsymbol{\kappa}_\mathrm{R}^{-1}(\boldsymbol{X}) \in \mathscr{B}$. We model time as a one-dimensional normed vector space $\mathbb{T}$ and denote a typical element in it as $\boldsymbol{\tau} = \tau\boldsymbol{s}$, where $\tau \in \mathbb{R}$ and $\boldsymbol{s}$ is a fixed vector in $\mathbb{T}$ of unit norm. We model the rigid body's motion using the one-parameter family of maps $\boldsymbol{x}_\tau : \mathbb{E}_\mathrm{R} \to \mathbb{E}$ (see Fig. 2). We call $\boldsymbol{x}_\tau$ the deformation map and $\boldsymbol{x} = \boldsymbol{x}_\tau(\boldsymbol{X})$ the material particle $\boldsymbol{X}$'s position vector at the time instance $\boldsymbol{\tau}$. The set $\boldsymbol{\kappa}_\tau(\mathscr{B}) = \left\{ \boldsymbol{x}_\tau(\boldsymbol{X}) \in \mathbb{E} \,\middle|\, \boldsymbol{X} \in \boldsymbol{\kappa}_\mathrm{R}(\mathscr{B}) \right\}$ (see Fig. 2) is called the current body.

2.2. Components

The sets $(\boldsymbol{E}_i)_{i\in\mathscr{I}}$ and $(\boldsymbol{e}_i)_{i\in\mathscr{I}}$, where $\mathscr{I} = (1,\ldots,n_\mathrm{sd})$, are orthonormal sets of basis vectors for $\mathbb{E}_\mathrm{R}$ and $\mathbb{E}$, respectively. By orthonormal we mean that the inner product between $\boldsymbol{E}_i$ and $\boldsymbol{E}_j$, or $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$, where $i,j \in \mathscr{I}$, equals δ_{ij}, the Kronecker delta symbol, which equals unity iff $i = j$ and zero otherwise. We call X_i the component of $\boldsymbol{X}$ w.r.t. $\boldsymbol{E}_i$ iff $X_i = \boldsymbol{X} \cdot \boldsymbol{E}_i$, where the dot denotes the inner product in $\mathbb{E}_\mathrm{R}$. The dot in other expressions is to be similarly interpreted noting the space to which the vectors belong. We call the ordered set $(X_i)_{i\in\mathscr{I}}$ the component form of $\boldsymbol{X}$ w.r.t. $(\boldsymbol{E}_i)_{i\in\mathscr{I}}$ and denote it as X or $\mathscr{M}\boldsymbol{X}$. We denote the space of all $m \times n$ real matrices, where $m, n \in \mathbb{N}$, $\mathscr{M}_{m,n}(\mathbb{R})$; here $\mathbb{N}$ and $\mathbb{R}$ denote the set of natural numbers and the space of real numbers, respectively. Thus, $\mathsf{X} \in \mathscr{M}_{n_\mathrm{sd},1}(\mathbb{R})$. We access the ith component, where $i \in \mathscr{I}$, of X, which of course is X_i, as $(\mathsf{X})_i$. Similarly, we denote the component of $\boldsymbol{x}$ w.r.t. $\boldsymbol{e}_i$ as x_i and call $\mathsf{x} = (x_i)_{i\in\mathscr{I}} \in \mathscr{M}_{n_\mathrm{sd},1}(\mathbb{R})$ the component form of $\boldsymbol{x}$ w.r.t. $(\boldsymbol{e}_i)_{i\in\mathscr{I}}$.

Say $\mathbb{W}$ and $\mathbb{U}$ are two arbitrary, oriented, finite dimensional Hilbert spaces; for instance, they can be $\mathbb{E}_\mathrm{R}$ and $\mathbb{E}$. We denote the space of all linear maps (transformations/operators) from $\mathbb{W}$ to $\mathbb{U}$ as $\mathscr{L}(\mathbb{W},\mathbb{U})$[2]. We denote the norm of a vector $\boldsymbol{w}_1$ in $\mathbb{W}$ that is induced by $\mathbb{W}$'s inner product, i.e., $(\boldsymbol{w}_1 \cdot \boldsymbol{w}_1)^{1/2}$, as $\|\boldsymbol{w}_1\|$. For $\boldsymbol{u}_1 \in \mathbb{U}$, the expression $\boldsymbol{u}_1 \otimes \boldsymbol{w}_1$ denotes the linear map from $\mathbb{W}$ to $\mathbb{U}$ defined as

$$(\boldsymbol{u}_1 \otimes \boldsymbol{w}_1)\,\boldsymbol{w}_2 = \boldsymbol{u}_1\,(\boldsymbol{w}_1 \cdot \boldsymbol{w}_2), \tag{2.1}$$

where $\boldsymbol{w}_2 \in \mathbb{W}$. If the sets $(\boldsymbol{u}_i)_{i\in\mathscr{I}}$ and $(\boldsymbol{w}_i)_{i\in\mathscr{I}}$ provide bases for $\mathbb{U}$ and $\mathbb{W}$, respectively, then it can be shown that $\left((\boldsymbol{u}_i \otimes \boldsymbol{w}_j)_{j\in\mathscr{I}} \right)_{i\in\mathscr{I}}$, which we will henceforth abbreviate as $(\boldsymbol{u}_i \otimes \boldsymbol{w}_j)_{i,j\in\mathscr{I}}$, provides a basis

---

[2]In our previous paper [27], we denoted the set of bounded linear operators from $\mathbb{U}$ to $\mathbb{W}$ as $B(\mathbb{U},\mathbb{W})$. As a linear operator on a finite dimensional normed space is automatically bounded, here we use $\mathscr{L}(\mathbb{U},\mathbb{W})$ instead of $B(\mathbb{U},\mathbb{W})$ to denote the set of all linear operators from $\mathbb{U}$ to $\mathbb{W}$.

for $\mathscr{L}(\mathbb{W}, \mathbb{U})$. The number T_{ij}, where $i, j \in \mathscr{I}$, is called the component of $\boldsymbol{T} \in \mathscr{L}(\mathbb{W}, \mathbb{U})$ w.r.t. $\boldsymbol{u}_i \otimes \boldsymbol{w}_j$ iff $T_{ij} = \boldsymbol{u}_i \cdot (\boldsymbol{T} \boldsymbol{w}_j)$. We call the nested ordered set $(T_{ij})_{i,j \in \mathscr{I}}$ the component form of $\boldsymbol{T}$ w.r.t. $(\boldsymbol{u}_i \otimes \boldsymbol{w}_j)_{i,j \in \mathscr{I}}$, and denote it as $\mathscr{M}\boldsymbol{T}$, or, when possible, briefly as T. We sometimes access the ith, j^{th} component of T, where $i, j \in \mathscr{I}$, as $(\mathsf{T})_{ij}$.

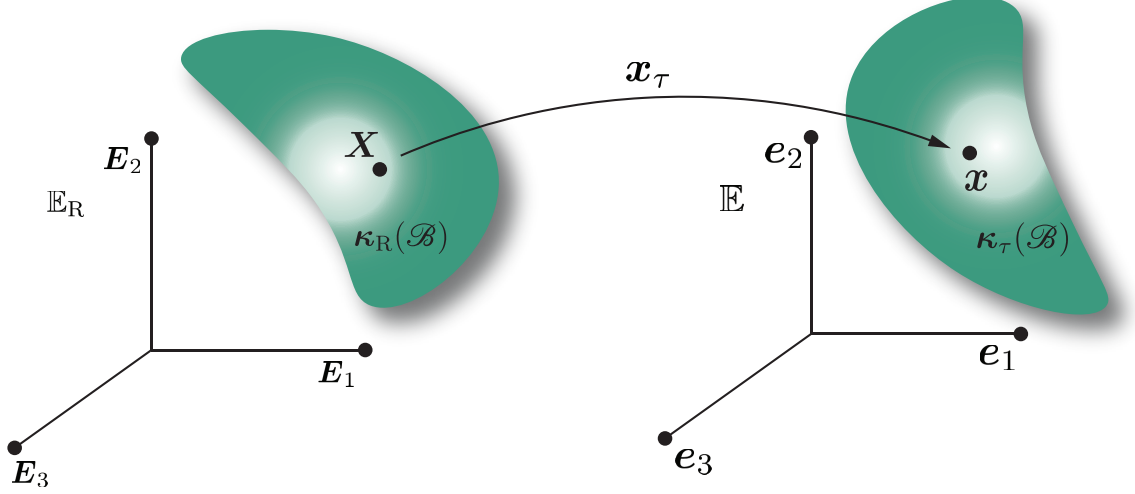
Figure 2: Some mathematical quantities used in the description of motion. Illustration of the reference Euclidean vector space $\mathbb{E}_{\text{R}}$, reference body $\boldsymbol{\kappa}_{\text{R}}(\mathscr{B})$, a material particle $\boldsymbol{X}$, the deformation map $\boldsymbol{x}_\tau$, current body $\boldsymbol{\kappa}_\tau(\mathscr{B})$, the (physical) Euclidean vector space $\mathbb{E}$, and the location of the material particle $\boldsymbol{X}$ in $\mathbb{E}$, i.e., the material particle $\boldsymbol{X}$'s spatial position vector $\boldsymbol{x}$. See §2.1 for details.

From here on, unless otherwise specified, we will be following the Einstein summation convention. As per this convention a repeated index in a term will imply a sum over that term with the repeated index taking values in $\mathscr{I}$. For example, the expression $X_i \boldsymbol{E}_i$ represents the sum $\sum_{i \in \mathscr{I}} X_i \boldsymbol{E}_i$. And an unrepeated index in a term will signify a set of $\mathrm{n_{sd}}$ terms. For example, the term $\boldsymbol{E}_i$ represents the set $\{\boldsymbol{E}_i \mid i \in \mathscr{I}\}$.

2.3. Velocities and Accelerations

For the case of rigid body motion $\boldsymbol{x}_\tau$ takes the form

$$\boldsymbol{x}_\tau(\boldsymbol{X}) = \boldsymbol{Q}_\tau \boldsymbol{X} + \boldsymbol{c}(\boldsymbol{\tau}), \tag{2.2}$$

where $\boldsymbol{Q}_\tau$ is a proper (orientation preserving), linear isometry from $\mathbb{E}_{\text{R}}$ into $\mathbb{E}$ and $\boldsymbol{c}(\boldsymbol{\tau}) = c_i(\tau)\boldsymbol{e}_i$, where c_i belongs to the space of twice continuously differentiable real valued functions over $\mathbb{R}$, i.e., to $C^2(\mathbb{R}, \mathbb{R})$. The operator $\boldsymbol{Q}_\tau$ can be written as $Q_{ij}(\tau)\boldsymbol{e}_i \otimes \boldsymbol{E}_j$, where $Q_{ij} \in C^2(\mathbb{R}, \mathbb{R})$ and satisfy $Q_{ki}(\tau)Q_{kj}(\tau) = \delta_{ij}$ for all $\tau \in \mathbb{R}$. We abbreviate $(Q_{ij}(\tau))_{i,j \in \mathscr{I}} \in \mathscr{M}_{\mathrm{n_{sd}},\mathrm{n_{sd}}}(\mathbb{R})$, $(c_i(\tau))_{i \in \mathscr{I}} \in \mathscr{M}_{\mathrm{n_{sd}},1}(\mathbb{R})$, and $(\delta_{ij})_{i,j \in \mathscr{I}} \in \mathscr{M}_{\mathrm{n_{sd}},\mathrm{n_{sd}}}(\mathbb{R})$ as $\mathbf{Q}(\tau)$, $\mathbf{c}(\tau)$, and $\mathbf{I}$, respectively. The component or non-dimensional form of (2.2) is (1.1). Since $\boldsymbol{Q}_\tau$ is a proper isometry, it follows that $\mathbf{Q}(\tau)$, which we refer to as the rotation matrix, belongs to the special orthogonal group $SO(\mathrm{n_{sd}})$. As a consequence of belonging to $SO(\mathrm{n_{sd}})$ the matrix $\mathbf{Q}(\tau)$ satisfies the equations

$$\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}(\tau) = \mathbf{I}, \tag{2.3a}$$

and

$$\mathbf{Q}(\tau)\,\mathbf{Q}^{\mathsf{T}}(\tau) = \mathbf{I}, \tag{2.3b}$$

where $\mathbf{Q}^{\mathsf{T}}(\tau)$ is the transpose of $\mathbf{Q}(\tau)$, i.e., $\mathbf{Q}^{\mathsf{T}}(\tau) = (\mathbf{Q}(\tau))^{\mathsf{T}}$.

We call $\mathscr{L}\left(\mathbb{T}, \mathbb{E}\right)$ the physical velocity vector space and denote it as $\mathbb{V}$. It can be shown that the set $\left(\boldsymbol{v}_i\right)_{i\in\mathscr{I}}$, where $\boldsymbol{v}_i \in \mathbb{V}$ and are defined such that $\boldsymbol{v}_i\boldsymbol{\tau} = \tau\boldsymbol{e}_i$, provides an orthonormal basis for $\mathbb{V}$. The velocity of a material particle $\boldsymbol{X}$ executing its motion in $\mathbb{E}$ lies in $\mathbb{V}$. The velocity of the material particle $\boldsymbol{X}$ at the instant $\boldsymbol{\tau}$, which we denote as $\boldsymbol{V}_\tau(\boldsymbol{X})$, equals the value of the Fréchet derivative[3] of the map $\mathbb{T} \ni \boldsymbol{\tau} \mapsto \boldsymbol{x_X}(\boldsymbol{\tau}) \in \mathbb{E}$, where $\boldsymbol{x_X}(\boldsymbol{\tau}) = \boldsymbol{x}_\tau(\boldsymbol{X})$, at the time instance $\boldsymbol{\tau}$. Thus, it follows from (2.2) that

$$\boldsymbol{V}_\tau(\boldsymbol{X}) = \boldsymbol{L}_\tau\boldsymbol{X} + \boldsymbol{c}'(\boldsymbol{\tau}), \tag{2.4}$$

where $\boldsymbol{L}_\tau := Q'_{ij}(\tau)\boldsymbol{v}_i \otimes \boldsymbol{E}_j$ and $\boldsymbol{c}'(\boldsymbol{\tau}) := c'_i(\tau)\boldsymbol{v}_i$, and Q'_{ij} and c'_i are the derivatives of Q_{ij} and c_i, respectively. We abbreviate $\left(Q'_{ij}(\tau)\right)_{i,j\in\mathscr{I}} \in \mathscr{M}_{\mathrm{n_{sd}},\mathrm{n_{sd}}}(\mathbb{R})$ and $\left(c'_i(\tau)\right)_{i\in\mathscr{I}} \in \mathscr{M}_{\mathrm{n_{sd}},1}(\mathbb{R})$ as $\mathbf{Q}'(\tau)$ and $\mathbf{c}'(\tau)$, respectively. Using (2.4) and (2.2) it can be shown that the velocity at the time instance τ of the material particle occupying the spatial position $\boldsymbol{x} \in \mathbb{E}$ at the time instance $\boldsymbol{\tau}$ is $\boldsymbol{W}_\tau\left(\boldsymbol{x} - \boldsymbol{c}(\boldsymbol{\tau})\right) + \boldsymbol{c}'(\boldsymbol{\tau})$, where the linear map $\boldsymbol{W}_\tau : \mathbb{E} \to \mathbb{V}$ is defined by the formula

$$\boldsymbol{W}_\tau\boldsymbol{x} = \boldsymbol{L}_\tau\boldsymbol{Q}^*_\tau\boldsymbol{x}, \tag{2.5}$$

for all $\boldsymbol{x} \in \mathbb{E}$. The operator $\boldsymbol{Q}^*_\tau$ is the Hilbert-adjoint of $\boldsymbol{Q}_\tau$ and is equal to $Q_{ji}(\tau)\boldsymbol{E}_i \otimes \boldsymbol{e}_j$. Let the component form of $\boldsymbol{W}_\tau$ w.r.t. $\left(\boldsymbol{v}_i \otimes \boldsymbol{e}_j\right)_{i,j\in\mathscr{I}}$ be $\left(W_{ij}(\tau)\right)_{i,j\in\mathscr{I}} \in \mathscr{M}_{\mathrm{n_{sd}},\mathrm{n_{sd}}}(\mathbb{R})$, which we abbreviate as $\mathbf{W}(\tau)$. It follows from (2.5) that $W_{ij}(\tau) = Q'_{ik}(\tau)Q_{jk}(\tau)$, or equivalently,

$$\mathbf{W}(\tau) = \mathbf{Q}'(\tau)\,\mathbf{Q}^\mathsf{T}(\tau). \tag{2.6}$$

We call $\mathscr{L}\left(\mathbb{T}, \mathbb{V}\right)$ the physical acceleration vector space and denote it as $\mathbb{A}$. It can be shown that the set $\left(\boldsymbol{a}_i\right)_{i\in\mathscr{I}}$, where $\boldsymbol{a}_i \in \mathbb{A}$ and are defined such that $\boldsymbol{a}_i\boldsymbol{\tau} = \tau\boldsymbol{v}_i$, provides an orthonormal basis for $\mathbb{A}$. The acceleration of a material particle $\boldsymbol{X}$ executing its motion in $\mathbb{E}$ lies in $\mathbb{A}$. The acceleration of $\boldsymbol{X}$ at the time instance $\boldsymbol{\tau}$ equals the value of the Fréchet derivative of the map $\mathbb{T} \ni \boldsymbol{\tau} \mapsto \boldsymbol{V_X}(\boldsymbol{\tau}) \in \mathbb{V}$, where $\boldsymbol{V_X}(\boldsymbol{\tau}) = \boldsymbol{V}_\tau(\boldsymbol{X})$, at the time instance $\boldsymbol{\tau}$. Thus, it follows from (2.4) that

$$\boldsymbol{A}_\tau(\boldsymbol{X}) = \boldsymbol{M}_\tau\boldsymbol{X} + \boldsymbol{c}''(\boldsymbol{\tau}), \tag{2.7}$$

where the map $\boldsymbol{M}_\tau : \mathbb{E}_\mathrm{R} \to \mathbb{A}$ is defined by the equation

$$\boldsymbol{M}_\tau := Q''_{ij}(\tau)\boldsymbol{a}_i \otimes \boldsymbol{E}_j \tag{2.8}$$

and $\boldsymbol{c}''(\boldsymbol{\tau}) := c''_i(\tau)\boldsymbol{a}_i$, where Q''_{ij} and c''_i are the derivatives of Q'_{ij} and c'_i, respectively. Let $A_{\tau i}(\boldsymbol{X})$ be the component of $\boldsymbol{A}_\tau(\boldsymbol{X})$ w.r.t. $\boldsymbol{a}_i$. We abbreviate the ordered sets $\left(A_{\tau i}(\boldsymbol{X})\right)_{i\in\mathscr{I}} \in \mathscr{M}_{\mathrm{n_{sd}},1}(\mathbb{R})$, $\left(Q''_{ij}(\tau)\right)_{i,j\in\mathscr{I}} \in \mathscr{M}_{\mathrm{n_{sd}},\mathrm{n_{sd}}}(\mathbb{R})$, and $\left(c''_i(\tau)\right)_{i\in\mathscr{I}} \in \mathscr{M}_{\mathrm{n_{sd}},1}(\mathbb{R})$ as $\mathbf{A}_\tau(\mathbf{X})$, $\mathbf{Q}''(\tau)$, and $\mathbf{c}''(\tau)$, respectively.

We will predominantly be presenting the ensuing results in component form. The component form can be converted into physical or dimensional form. Therefore, from here on we will be often omit explicitly using the qualification "is the component form of" when referring to the component form of a physical quantity. For example, instead of saying "$\mathbf{A}_\tau(\mathbf{X})$ as the component form of the acceleration of the material particle $\boldsymbol{X}$ at the time instance $\boldsymbol{\tau}$", we will often write "$\mathbf{A}_\tau(\mathbf{X})$ is the acceleration of the material particle $\mathbf{X}$ at the time instance $\boldsymbol{\tau}$". The acceleration $\mathbf{A}_\tau(\mathbf{X})$ can be interpreted as the value of the (non-dimensional) acceleration field $\mathbf{A}_\tau : B_\mathrm{R}(\mathscr{B}) \to \mathbb{R}^3$, where we call $B_\mathrm{R}(\mathscr{B}) := \left\{(X_1, X_2, X_3) \in \mathbb{R}^3 \,\middle|\, X_i\boldsymbol{E}_i \in \boldsymbol{\kappa}_\mathrm{R}(\mathscr{B})\right\}$ the non-dimensional reference body.

3. Review of the AO-algorithm

Let $\overline{\boldsymbol{Q}}_\tau : \mathbb{A} \to \mathbb{E}_\mathrm{R}$ be defined by the equation

$$\overline{\boldsymbol{Q}}_\tau = Q_{ji}(\tau)\boldsymbol{E}_i \otimes \boldsymbol{a}_j, \tag{3.1}$$

---

[3]For the definition of Fréchet derivative in the context of the current work see [27, §2.1]

then we call the map $\overline{A}_\tau : \kappa_{\mathrm{R}}(\mathscr{B}) \to \mathbb{E}_{\mathrm{R}}$ defined by the equation

$$\overline{A}_\tau(X) = \overline{Q}_\tau A_\tau(X) \tag{3.2}$$

the "Pseudo-acceleration field". Say $\overline{A}_{\tau i}(X)$ is the component of $\overline{A}_\tau(X)$ w.r.t. E_i, then we abbreviate $\left(\overline{A}_{\tau i}(X)\right)_{i \in \mathscr{J}} \in \mathscr{M}_{n_{\mathrm{sd}},1}(\mathbb{R})$, the component form of $\overline{A}_\tau(X)$ w.r.t. E_i, as $\overline{\mathbf{A}}_\tau(\mathbf{X})$. From the definitions of the pseudo acceleration field $\overline{A}_\tau$ (3.2), and $\overline{\mathbf{A}}_\tau(\mathbf{X})$, and the definitions of $\mathbf{A}_\tau(\mathbf{X})$, and $\mathbf{Q}(\tau)$, which are given in §2.3, it follows that

$$\mathbf{A}_\tau(\mathbf{X}) = \mathbf{Q}(\tau)\overline{\mathbf{A}}_\tau(\mathbf{X}). \tag{3.3}$$

In [27, §2.1.1] it was shown that

$$\overline{\mathbf{A}}_\tau(\mathbf{X}) = \mathbf{P}(\tau)\mathbf{X} + \mathbf{q}(\tau), \tag{3.4}$$

where

$$\mathbf{P}(\tau) = \mathbf{Q}^{\mathsf{T}}(\tau)\mathbf{Q}''(\tau) \tag{3.5}$$

is the component form of the linear map $P_\tau := \overline{Q}_\tau \circ M_\tau$ w.r.t. $(E_i \otimes E_j)_{i,j \in \mathscr{J}}$, and $\mathbf{q}(\tau) \in \mathscr{M}_{n_{\mathrm{sd}},1}(\mathbb{R})$ is the component form of

$$q(\tau) := \overline{Q}_\tau c''(\tau) \tag{3.6}$$

w.r.t. $(E_i)_{i \in \mathscr{J}}$. Thus, the acceleration field $\mathbf{A}_\tau$ is taken to be fully determined once $\mathbf{Q}(\tau)$, $\mathbf{P}(\tau)$, and $\mathbf{q}(\tau)$ have been computed.

Both the AO and the $\sqrt{\text{AO}}$ algorithms can be described as consisting of three primary steps. The AO-algorithm's three steps can briefly be described as follows:

AO-Step 1 Compute (time discrete versions of) the maps $\tau \mapsto \mathbf{P}(\tau)$ and $\tau \mapsto \mathbf{q}(\tau)$ using the measurements and the geometry of the arrangement of the four tri-axial accelerometers.

AO-Step 2 Compute the map $\tau \mapsto \overline{\mathbf{W}}(\tau)$, where

$$\overline{\mathbf{W}}(\tau) := \mathbf{Q}^{\mathsf{T}}(\tau)\mathbf{W}(\tau)\mathbf{Q}(\tau), \tag{3.7}$$

using the map $\mathbf{P}$ computed in *AO-Step 1* and numerically integrating (1.2). From Lemma Appendix B.1 we have that the matrix $\overline{\mathbf{W}}(\tau)$ belongs to the space of $n_{\mathrm{sd}} \times n_{\mathrm{sd}}$ real skew-symmetric matrices, which we denote as $\mathfrak{so}(\mathbb{R}, n_{\mathrm{sd}})$.

AO-Step 3 Compute the map $\tau \mapsto \mathbf{Q}(\tau)$ using the $\overline{\mathbf{W}}$ map computed in *AO-Step 2* and numerically integrating the equation

$$\mathbf{Q}'(\tau) = \mathbf{Q}(\tau)\overline{\mathbf{W}}(\tau). \tag{3.8}$$

Equation (3.8) is from [27], where it appears as equation 3.14.

Step one of the $\sqrt{\text{AO}}$-algorithm has two sub-steps: the *predictor* step and the *corrector* step. The predictor step is the same as *AO-Step 1* of the AO-algorithm. The corrector step is necessary for carrying out step two of the $\sqrt{\text{AO}}$-algorithm. In step two of the $\sqrt{\text{AO}}$-algorithm instead of obtaining $\overline{\mathbf{W}}$ from (1.2), as is done in the AO-algorithm, we obtain it from (1.3). More precisely, in the $\sqrt{\text{AO}}$-algorithm $\overline{\mathbf{W}}(\tau)$ is obtained as the square root of the symmetric part of $\mathbf{P}(\tau)$. We use $\mathrm{sym}\,(\mathbf{P}(\tau))$ to denote the symmetric part of $\mathbf{P}(\tau)$. The derivation of (1.3) is presented in Appendix B. A procedure for determining $\overline{\mathbf{W}}(\tau)$ as the square root of $\mathrm{sym}\,(\mathbf{P}(\tau))$, i.e., for solving (1.3) for $\overline{\mathbf{W}}(\tau)$ with given $\mathbf{P}(\tau)$, is presented in §4.2. The goal of step three of the $\sqrt{\text{AO}}$-algorithm is to compute $\mathbf{Q}$ using the $\overline{\mathbf{W}}$ computed in step two. It involves using a slightly modified version of the numerical integration scheme described by equations 3.15, 3.16, and 3.17 in [27, §3.2] to solve (3.8). We discuss it in §4.3.

4. The $\sqrt{\text{AO}}$-algorithm

As we mentioned in §3 the $\sqrt{\text{AO}}$ algorithm consists of three primary steps. Those steps are as follows:

Step 1 Compute (time discrete versions of) the maps $\tau \mapsto \mathbf{P}(\tau)$ and $\tau \mapsto \mathbf{q}(\tau)$ using the measurements and the geometry of the arrangement of the four tri-axial accelerometers (see §4.1 for details).

Step 2 Use (1.3) and the **P** map obtained from Step 1 to solve for $\tau \mapsto \overline{\mathbf{W}}(\tau)$. That is, for each τ in a discrete sequence of time instances, compute $\overline{\mathbf{W}}(\tau)$ as the square root of the symmetric part of $\mathbf{P}(\tau)$ (for details see §4.2).

Step 3 Compute (a time discrete version of) the map $\tau \mapsto \mathbf{Q}(\tau)$ using the $\overline{\mathbf{W}}$ map computed in Step 2 and numerically integrating (3.8) (details in §4.3).

4.1. $\sqrt{AO}$-algorithm, Step 1 of 3

In §3.1 of [27] a method was presented to estimate $\mathbf{P}(\tau)$ and $\mathbf{q}(\tau)$ from the accelerometer measurements corresponding to the time instance τ. Applying that method for each τ in a discrete time sequence yields a numerical approximation for the maps $\tau \mapsto \mathbf{P}(\tau)$ and $\tau \mapsto \mathbf{q}(\tau)$. We present here an augmented version of that method for computing similar numerical approximations. The primary difference between our method and that presented in [27] is that the estimate for $\mathbf{P}(\tau)$ yielded by our method is certain to retain some of the mathematical properties that are expected of $\mathbf{P}(\tau)$ based on our theoretical analysis. Specifically, it follows from Lemmas Appendix C.1 and Appendix C.2 that $\mathrm{sym}\,(\mathbf{P}(\tau))$ is a negative semidefinite matrix with its negative eigenvalues, if any, being of even algebraic multiplicities. These mathematical properties of $\mathbf{P}(\tau)$ are critical for carrying out Step 2 of the $\sqrt{AO}$-algorithm. We found that experimental noise and errors can cause the estimate for $\mathbf{P}(\tau)$ provided by the method presented in [27] to lose the aforementioned mathematical properties. Our method, on the contrary, ensures that the symmetric part of the estimated $\mathbf{P}(\tau)$ is negative semidefinite and that its negative eigenvalues, when they exist, are of even algebraic multiplicities. Once $\mathbf{P}(\tau)$ is estimated, our method to estimate $\mathbf{q}(\tau)$ is exactly the same as that in [27]. We review it in 4.1.2.

4.1.1. Estimating $\mathbf{P}(\tau)$

Our method for estimating $\mathbf{P}(\tau)$ can be described as consisting of two steps: a *predictor step* and a *corrector step*. In the predictor step we use the method presented in [27, §3.1] for estimating $\mathbf{P}(\tau)$ to compute a prediction for $\mathbf{P}(\tau)$. We denote this prediction as $\mathbf{P}(\tau)^{\mathrm{p}}$. In the corrector step we estimate $\mathbf{P}(\tau)$ as the sum of $\mathbf{P}(\tau)^{\mathrm{p}}$ and a correction term, which we construct using $\mathbf{P}(\tau)^{\mathrm{p}}$. The correction term is constructed such that the estimated $\mathbf{P}(\tau)$ is as close as possible to $\mathbf{P}(\tau)^{\mathrm{p}}$ under the constraint that the estimated $\mathbf{P}(\tau)$'s symmetric part is negative semidefinite and its negative eigenvalues (if they exist) are of even algebraic multiplicities.

Predictor step. Say the four tri-axial accelerometers are attached to the rigid body $\mathscr{B}$ at the material particles $\left({}^{\ell}\mathcal{X}\right)_{\ell \in \mathscr{J}}$, where $\mathscr{J} := (1, \ldots, 4)$, and let the position vectors of those particles in $\mathbb{E}_{\mathrm{R}}$, respectively, be $\left({}^{\ell}\boldsymbol{X}\right)_{\ell \in \mathscr{J}}$ (see Fig. 3). A tri-axial accelerometer is capable of measuring the components of its acceleration in three mutually perpendicular directions. We refer to those directions as the accelerometer's measurement directions. The measurement directions are usually marked on the accelerometer package by the manufacturer as arrows that are labeled x, y, and z. As $\mathscr{B}$ moves in $\mathbb{E}$, the attached accelerometers move with it, and, therefore, the measurement directions (in $\mathbb{E}$) can change with time. For an accelerometer ℓ, where $\ell \in \mathscr{J}$, we denote its time varying measurement directions in $\mathbb{E}$ using the orthonormal set $\left({}^{\ell}\boldsymbol{e}_{\tau i}\right)_{i \in \mathscr{J}}$. Assuming that the accelerometers remain rigidly attached to $\mathscr{B}$, i.e., their positions and orientations w.r.t. $\mathscr{B}$ do not change as $\mathscr{B}$ moves in $\mathbb{E}$, it can be shown that $\boldsymbol{Q}_{\tau}^{*}\,{}^{\ell}\boldsymbol{e}_{\tau i}$, where $\ell \in \mathscr{J}, i \in \mathscr{I}$, is a constant vector in $\mathbb{E}_{\mathrm{R}}$, which we denote as ${}^{\ell}\boldsymbol{E}_i$. The position vectors $\left({}^{\ell}\boldsymbol{X}\right)_{\ell \in \mathscr{J}}$ and the directions $\left({}^{\ell}\boldsymbol{E}_i\right)_{\ell \in \mathscr{J}, i \in \mathscr{I}}$ are known from the arrangement and orientation of the accelerometers at the experiment's beginning.

For $\ell \in \mathscr{J}$, let ${}^{\ell}\overline{\mathbf{A}}(\tau) := \left({}^{\ell}\alpha_j(\tau){}^{\ell}\boldsymbol{E}_j \cdot \boldsymbol{E}_i\right)_{i \in \mathscr{I}}$ (no sum over ℓ), where ${}^{\ell}\alpha_i(\tau)$, $i \in \mathscr{I}$, is the measurement reported by accelerometer ${}^{\ell}\boldsymbol{X}$ for the (non-dimensional) component of its acceleration in the ${}^{\ell}\boldsymbol{e}_{\tau i}$ direction[4] at the time instance τ. And let ${}^{\ell}\mathbf{X} := \left({}^{\ell}\boldsymbol{X} \cdot \boldsymbol{E}_i\right)_{i \in \mathscr{I}}$. Then, we compute $\mathbf{P}(\tau)^{\mathrm{p}}$ as $\mathbf{P}(\tau)$ is estimated in [27] using the equation

$$\mathbf{P}(\tau)^{\mathrm{p}} = \left(\left({}^i\Delta\overline{\mathbf{A}}(\tau)\right)^{\mathsf{T}} \left({}^j\Delta\mathbf{X}\right)\right) \left(\left({}^i\Delta\mathbf{Y}\right) \left({}^j\Delta\mathbf{Y}\right)^{\mathsf{T}}\right), \tag{4.1}$$

---

[4]Or to be mathematically precise, in the ${}^{\ell}\boldsymbol{a}_{\tau i} \in \mathbb{A}$ direction that is defined such that $\left({}^{\ell}\boldsymbol{a}_{\tau i}\boldsymbol{s}\right) \boldsymbol{s} = {}^{\ell}\boldsymbol{e}_{\tau i}$.

where $^i\Delta\overline{\mathbf{A}}(\tau) := {}^{i+1}\overline{\mathbf{A}}(\tau) - {}^1\overline{\mathbf{A}}(\tau)$, $^i\Delta\mathbf{X} := {}^{i+1}\mathbf{X} - {}^1\mathbf{X}$. The ordered sets $^i\Delta\mathbf{Y}$ belong to $\mathscr{M}_{n_{sd},1}(\mathbb{R})$ and are defined by the equation

$$\left({}^1\Delta\mathbf{Y},\ldots,{}^{n_{sd}}\Delta\mathbf{Y}\right) = \left({}^1\Delta\mathbf{X},\ldots,{}^{n_{sd}}\Delta\mathbf{X}\right)^{-\mathsf{T}}, \tag{4.2}$$

where $(\cdot)^{-\mathsf{T}}$ is the operator that acts on an invertable element of $\mathscr{M}_{n_{sd},n_{sd}}(\mathbb{R})$ and returns the transpose of its inverse.
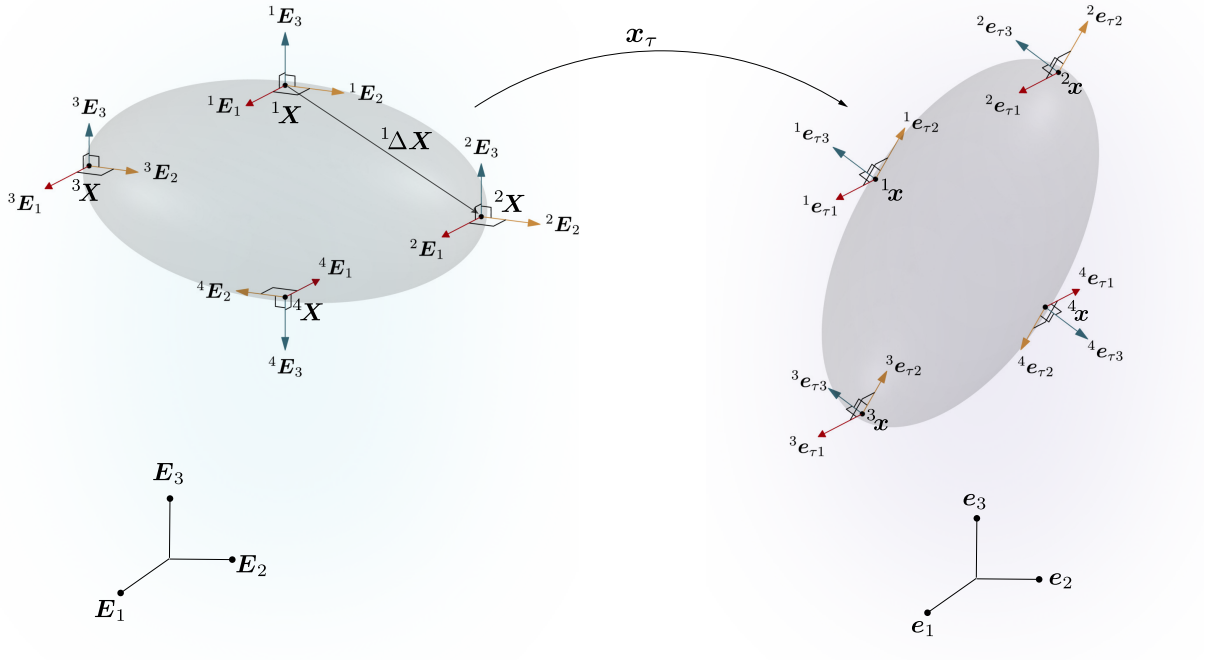
Figure 3: Schematic of the locations and orientations of four tri-axial accelerometers (left) and their motion (right) (modified from [27], copyright 2020, Elsevier).

Corrector step. In Appendix D.1 we show that sym($\mathbf{P}(\tau)$) allows itself to be decomposed as

$$\mathbf{N}(\tau)\,\mathbf{D}(\tau)\,\mathbf{N}^{\mathsf{T}}(\tau), \tag{4.3a}$$

where $\mathbf{N}(\tau) \in \mathscr{M}_{n_{sd},n_{sd}}(\mathbb{R})$ is an orthogonal matrix, i.e.,

$$\mathbf{N}^{\mathsf{T}}(\tau)\mathbf{N}(\tau) = \mathbf{I}, \tag{4.3b}$$

and $\mathbf{D}(\tau) \in \mathscr{M}_{n_{sd},n_{sd}}(\mathbb{R})$ is a diagonal matrix that for $n_{sd} = 2$ and 3, respectively, has the form

$$\mathbf{D}(\tau) = \mathrm{diag}\left(-\lambda(\tau)^2, -\lambda(\tau)^2\right) \text{ and } \mathrm{diag}\left(0, -\lambda(\tau)^2, -\lambda(\tau)^2\right), \tag{4.3c}$$

where $\lambda(\tau) \in \mathbb{R}$ and the function $\mathrm{diag}(\cdot) : \mathbb{F}^{n_{sd}} \to \mathscr{M}_{n_{sd},n_{sd}}(\mathbb{F})$, where $\mathbb{F}$ is either $\mathbb{R}$ or $\mathbb{C}$, is defined such that $\mathrm{diag}(a_1,\ldots,a_{n_{sd}})$ is a diagonal matrix with diagonal entries $a_1,\ldots,a_{n_{sd}}$.

The matrix sym($\mathbf{P}(\tau)$) allowing the decomposition (4.3) is critical for carrying out Step 2 of the $\sqrt{\text{AO}}$-algorithm. In an ideal scenario, in which there are no experimental errors or noise in the accelerometer measurements, $\mathbf{P}(\tau)^{\mathrm{p}}$ would be the same as $\mathbf{P}(\tau)$. However, due to the experimental noise and other errors $\mathbf{P}(\tau)^{\mathrm{p}}$ will generally be different from $\mathbf{P}(\tau)$. In general, such a deviation would not be of much consequence, since, experimental measurements of physical quantities, more often than not, are different from the true values of those quantities. Thus, generally, we would, as done by [27], take $\mathbf{P}(\tau)^{\mathrm{p}}$ to be the final estimate for $\mathbf{P}(\tau)$ and no longer distinguish between $\mathbf{P}(\tau)^{\mathrm{p}}$ and $\mathbf{P}(\tau)$. However, in the present case the deviation of $\mathbf{P}(\tau)^{\mathrm{p}}$ from $\mathbf{P}(\tau)$ has an important consequence which requires us to not take $\mathbf{P}(\tau)^{\mathrm{p}}$ as $\mathbf{P}(\tau)$'s final estimate. The important consequence is that in general sym($\mathbf{P}(\tau)^{\mathrm{p}}$) will not allow a decomposition of the form (4.3). In general, it will only allow itself to be decomposed as

$\mathbf{N}_p(\tau) \operatorname{diag}\left(\lambda_i(\tau), \ldots, \lambda_{\mathrm{n_{sd}}}(\tau)\right) \mathbf{N}_p^{\mathsf{T}}(\tau)$, where $\mathbf{N}_p(\tau)$'s columns are the eigenvectors of $\operatorname{sym}(\mathbf{P}(\tau)^{\mathrm{p}})$ that are chosen such that $\mathbf{N}_p(\tau)$ is orthogonal and their corresponding eigenvalues $\lambda_i(\tau) \in \mathbb{R}$ form a non-increasing sequence, i.e., $\lambda_1(\tau) \geqslant \ldots \geqslant \lambda_{\mathrm{n_{sd}}}(\tau)$. Therefore, instead of taking $\mathbf{P}(\tau)^{\mathrm{p}}$ as the final estimate of $\mathbf{P}(\tau)$ we derive the final estimate for $\mathbf{P}(\tau)$, as we detail next, using $\mathbf{P}(\tau)^{\mathrm{p}}$ so that its symmetric part does allow a decomposition of the form (4.3).

We take the skew-symmetric part of our final estimate for $\mathbf{P}(\tau)$ to be the same as that of $\mathbf{P}(\tau)^{\mathrm{p}}$. We take its symmetric part to be

$$\mathbf{N}_p(\tau)\,\check{\mathbf{D}}(\tau)\,\mathbf{N}_p^{\mathsf{T}}(\tau), \tag{4.4a}$$

where

$$\check{\mathbf{D}}(\tau) := \operatorname{diag}\left(0, -\check{\lambda}(\tau)^2, -\check{\lambda}(\tau)^2\right), \tag{4.4b}$$

with

$$\check{\lambda}(\tau) := \begin{cases} \sqrt{-\frac{\lambda_2(\tau)+\lambda_3(\tau)}{2}}, & \lambda_2(\tau) + \lambda_3(\tau) \leqslant 0, \\ 0, & \lambda_2(\tau) + \lambda_3(\tau) > 0, \end{cases} \tag{4.4c}$$

for $\mathrm{n_{sd}} = 3$, and

$$\check{\mathbf{D}}(\tau) := \operatorname{diag}\left(-\check{\lambda}(\tau)^2, -\check{\lambda}(\tau)^2\right), \tag{4.4d}$$

with

$$\check{\lambda}(\tau) := \begin{cases} \sqrt{-\frac{\lambda_1(\tau)+\lambda_2(\tau)}{2}}, & \lambda_1(\tau) + \lambda_2(\tau) \leqslant 0, \\ 0, & \lambda_1(\tau) + \lambda_2(\tau) > 0, \end{cases} \tag{4.4e}$$

for $\mathrm{n_{sd}} = 2$.

The orthogonal matrix $\mathbf{N}_p(\tau)$ and the eigenvalues $\lambda_i(\tau)$ can be obtained from the spectral or symmetric-Schur [35, §8] decomposition of $\operatorname{sym}(\mathbf{P}(\tau)^{\mathrm{p}})$. Since $\operatorname{sym}(\mathbf{P}(\tau)^{\mathrm{p}})$ is a real symmetric matrix, it is always possible to carry out $\operatorname{sym}(\mathbf{P}(\tau)^{\mathrm{p}})$'s spectral or symmetric Schur decomposition.

To summarize, we take $\mathbf{P}(\tau)$'s final estimate to be

$$\mathbf{P}^{\mathrm{p}}(\tau) + \Delta\mathbf{P}(\tau), \tag{4.5a}$$

where

$$\Delta\mathbf{P}(\tau) := \operatorname{sym}\left(\check{\mathbf{P}}(\tau) - \mathbf{P}^{\mathrm{p}}(\tau)\right), \tag{4.5b}$$

$$\check{\mathbf{P}}(\tau) := \mathbf{N}_p(\tau)\check{\mathbf{D}}(\tau)\,\mathbf{N}_p^{\mathsf{T}}(\tau). \tag{4.5c}$$

It can be ascertained that the symmetric part of our final estimate for $\mathbf{P}(\tau)$, namely $\check{\mathbf{P}}(\tau)$, allows a decomposition of the form (4.3). In fact, that decomposition is precisely the one given by (4.4). For $\mathrm{n_{sd}} = 2$ or 3 it can be shown that $\check{\mathbf{P}}(\tau)$ is the best approximation, in the Frobenius norm, to $\operatorname{sym}\left(\mathbf{P}^{\mathrm{p}}(\tau)\right)$ in the set of $\mathrm{n_{sd}} \times \mathrm{n_{sd}}$ real symmetric negative-semidefinite matrices whose negative eigenvalues (when they exist) are of even algebraic multiplicities.

4.1.2. Estimating $\mathbf{q}(\tau)$

After estimating $\mathbf{P}(\tau)$ as described in §4.1.1 using (4.5) we estimate $\mathbf{q}(\tau)$ as

$$^\ell\bar{\mathbf{A}}(\tau) - \mathbf{P}(\tau)\,^\ell\mathbf{X}, \tag{4.6}$$

where ℓ is some particular integer in $\mathscr{J}$.

4.2. $\sqrt{AO}$-algorithm, Step 2 of 3

In Appendix D.2 we show using $\mathbf{P}(\tau)$'s decomposition (4.3) that $\overline{\mathbf{W}}(\tau)$ can be computed from (1.3) as

$$\overline{\mathbf{W}}(\tau) = \pm \begin{cases} \mathbf{N}(\tau) \star (\lambda(\tau))\,\mathbf{N}^\mathsf{T}(\tau), & n_{\mathrm{sd}} = 2, \\ \mathbf{N}(\tau) \star ((\lambda(\tau),0,0))\,\mathbf{N}^\mathsf{T}(\tau), & n_{\mathrm{sd}} = 3, \end{cases} \tag{4.7}$$

where the map $\star\,(\cdot)$ is defined in Appendix A.

Using the decomposition (4.4) for the symmetric part of our final estimate for $\mathbf{P}(\tau)$ and similar calculations as those used in §Appendix D.2, it can be shown that if we compute our final estimate for $\overline{\mathbf{W}}(\tau)$ as

$$\pm \begin{cases} \mathbf{N}_p(\tau) \star (\check{\lambda}(\tau))\,\mathbf{N}_p^\mathsf{T}(\tau), & n_{\mathrm{sd}} = 2, \\ \mathbf{N}_p(\tau) \star ((\check{\lambda}(\tau),0,0))\,\mathbf{N}_p^\mathsf{T}(\tau), & n_{\mathrm{sd}} = 3, \end{cases} \tag{4.8}$$

then it and our final estimate for $\mathbf{P}(\tau)$ will satisfy (1.3).

We take the time discrete versions of the $\overline{\mathbf{W}}$ and $\mathbf{P}$ maps to be constant over each time interval $\Delta\tau_n := [n\Delta\tau, (n+1)\Delta\tau)$, where $n \in (0, 1, \cdots)$ and $\Delta\tau \in \mathbb{R}$. We denote the values of these two maps over $\Delta\tau_n$ as $\overline{\mathbf{W}}(n)$ and $\mathbf{P}(n)$, respectively. The quantity $\overline{\mathbf{W}}(0)$ is known from initial conditions. For $n > 0$ we compute $\overline{\mathbf{W}}(n)$ using (4.8). Using the positive and negative signs in (4.8) will give us two different estimates for $\overline{\mathbf{W}}(n)$. Among those two estimates we choose the one that is closer to $\overline{\mathbf{W}}$'s value over the previous time interval. To be precise, we choose the estimate that gives a lower value for the metric $m(\overline{\mathbf{W}}(n), \overline{\mathbf{W}}(n-1))$, where $m : \mathfrak{so}(\mathbb{R}, n_{\mathrm{sd}})^2 \to \mathbb{R}$,

$$m(\overline{\mathbf{W}}(n), \overline{\mathbf{W}}(n-1)) = \arccos\left(\frac{\star\left(\overline{\mathbf{W}}(n)\right) \cdot \star\left(\overline{\mathbf{W}}(n-1)\right)}{\|\star\left(\overline{\mathbf{W}}(n)\right)\| \|\star\left(\overline{\mathbf{W}}(n-1)\right)\|} \right). \tag{4.9}$$

The metric (4.9) is a measure of the difference between between $\overline{\mathbf{W}}(n)$ and $\overline{\mathbf{W}}(n-1)$. Our criterion for choosing between the two estimates given by (4.8) is essentially based on the assumption that in most practical scenarios $\overline{\mathbf{W}}$ should be a continuous function of time, and on the ansatz that due to the continuity of $\overline{\mathbf{W}}$ the true $\overline{\mathbf{W}}(n)$ would be the one that is closer to $\overline{\mathbf{W}}(n-1)$ when $\|\star\left(\overline{\mathbf{W}}(n-1)\right)\|$ is large.

If $\overline{\mathbf{W}}(n-1) = \mathbf{0}$ or when $\|\star\left(\overline{\mathbf{W}}(n-1)\right)\|$ is small, then our above criterion for choosing between the two estimates for $\overline{\mathbf{W}}(n)$ cannot be used. In such cases we first derive a prediction for $\overline{\mathbf{W}}(n)$ by applying the AO-algorithm to the previous time interval and then choose the estimate that is closer to that prediction.

4.3. $\sqrt{AO}$-algorithm, Step 3 of 3

We use a slightly modified version of the numerical integration scheme described by equations 3.15, 3.16, and 3.17 in [27, §3.2] to solve (3.8). As we did with $\overline{\mathbf{W}}$ and $\mathbf{P}$, we assume that the discrete version of $\mathbf{Q}$ remains constant over each time interval $\Delta\tau_n$ and denote its constant values as $\mathbf{Q}(n)$, where $n \in (0, 1, \cdots)$. The matrix $\mathbf{Q}(0)$ is taken to be known from the initial conditions of the experiment. For $n > 0$ the matrix $\mathbf{Q}(n)$ is computed as

$$\mathbf{Q}(n) = \mathbf{Q}(n-1)\,e^{\Delta\tau\,\overline{\mathbf{W}}_{n-\frac{1}{2}}}, \tag{4.10a}$$

where the map $e^{(\cdot)} : \mathfrak{so}(\mathbb{R}, n_{sd}) \to SO(n_{sd})$ is defined by the equation[5]

$$e^{(\cdot)} = \mathbf{I} + \mathrm{sinc}\left(\|\star(\cdot)\|\right)(\cdot) + \frac{1}{2}\left(\mathrm{sinc}\left(\frac{\|\star(\cdot)\|}{2}\right)\right)^2 (\cdot)^2, \tag{4.10b}$$

and

$$\overline{\mathbf{W}}_{n-\frac{1}{2}} := \frac{1}{2}\left(\overline{\mathbf{W}}(n) + \overline{\mathbf{W}}(n-1)\right). \tag{4.10c}$$

The difference between the integration scheme (4.10) and that given by 3.15, 3.16, and 3.17 in [27, §3.2] is the manner in which $\overline{\mathbf{W}}_{n-\frac{1}{2}}$ is computed. In Rahaman *et al.*'s integration scheme, $\overline{\mathbf{W}}_{n-\frac{1}{2}}$ is computed as $\overline{\mathbf{W}}(n-1) + \frac{\Delta\tau}{2}\mathrm{skew}\left(\mathbf{P}(n-1)\right)$, whereas we compute it using (4.10c). Here we use $\mathrm{skew}\left(\mathbf{P}(n-1)\right)$ to denote the skew-symmetric part of the matrix $\mathbf{P}(n-1)$.

5. *In silico* validation, evaluation and comparison of the $\sqrt{\mathrm{AO}}$-algorithm

In this section, we check the validity and robustness of the $\sqrt{\mathrm{AO}}$-algorithm. We do that by feeding in virtual accelerometer data, to which differing amounts of bias and noise type errors have been added, to the $\sqrt{\mathrm{AO}}$ and AO algorithms, and comparing their resulting predictions. We discuss the creation of the virtual accelerometer data in §5.1, and simulating bias and noise type errors in §5.2. We compare the predictions in §5.3.

5.1. *Virtual accelerometer data from the simulation of a rigid ellipsoid impacting an elastic half-space*

The virtual accelerometer data we use for the comparison is from the numerical simulation of a rigid ellipsoid impacting an elastic half-space. This simulation is presented and discussed in detail in [27], starting in §4. However, for the readers convenience we give a very brief description of that simulation here.

In the simulation an ellipsoid, $\mathscr{B}$, is dropped onto an elastic half-space, H, under the action of gravity with the initial angular and translational velocities prescribed (see Fig. 5). In the simulation the Euclidean point space $\mathscr{E}$, in which the ellipsoid and the half-space, respectively, execute their motion and deformation, is taken to be three dimensional, i.e., $n_{sd} = 3$. The vectors $\boldsymbol{E}_i$ and $\boldsymbol{e}_i$, $i \in \mathscr{I}$, are taken to have units of meters and s to have units of seconds. Hence $\boldsymbol{v}_i$ and $\boldsymbol{a}_i$, $i \in \mathscr{I}$, have units of meters-per-second and meters-per-second-squared, respectively.

The reference configuration of the ellipsoid is given in Fig. 4. In $\mathscr{E}_R$ the ellipsoid occupies the region $\left\{(X_1, X_2, X_3) \,\middle|\, (X_1/a)^2 + (X_2/b)^2 + (X_3/c)^2 \leqslant 1\right\}$, where $(a, b, c) = (0.15, 0.10, 0.08)$. The half-space when it is undeformed in $\mathscr{E}$ occupies the region $x_3 < 0$. The initial location and orientation of $\mathscr{B}$ w.r.t. H in $\mathscr{E}$ are shown in Fig. 5. They correspond to the initial conditions $\mathbf{Q}(0) = \mathrm{diag}\,(1, 1, 1)$ and $\mathbf{c}(0) = (0, 0, 0.75)$.

The mechanics of H is modeled using the theory of small deformation linear elasto-statics and taking H's Young's modulus and Poisson's ratio to be 10^4 Pa and 0.3, respectively. The ellipsoid is rigid and homogeneous. Its density and total mass are 1989.44 kg/m^3 and 10 kg, respectively.

The ellipsoid's dynamics are obtained by numerically solving its linear and angular momentum balance equations. The force in those equations arises due to the action of gravity on $\mathscr{B}$ and $\mathscr{B}$'s interaction with H; and the torque exclusively from $\mathscr{B}$'s interaction with H. The interaction between $\mathscr{B}$ and H is modeled using the Hertz contact theory, e.g., [36, 37]. For details on effecting a numerical solution to the balance equations see [27, §B1.1]. For more details regarding the contact modeling see [27, §B1.2].

Four virtual accelerometers are taken to be rigidly attached to the ellipsoid's material particles ${}^\ell\mathscr{X}$, $\ell \in \mathscr{J}$. The locations and orientations of those accelerometers w.r.t. $\mathscr{B}$ in $\mathscr{E}_R$ are shown in Fig. 4. Their position vectors ${}^\ell\boldsymbol{X}$ and orientations $\left({}^\ell\boldsymbol{E}_i\right)_{i \in \mathscr{I}}$, $\ell \in \mathscr{J}$, are given in the caption of Fig. 4. The acceleration of any of the ellipsoid's material particles can be obtained from the simulation results using the procedure outlined in [27, §B2]. For $i \in \mathscr{I}$, the values of ${}^1\alpha_i$, which is the component of ${}^1\mathscr{X}$'s acceleration in the

---

[5]This equation is the corrected version of equation 3.17 in [27, §3.2], which has two typos in it.

$^1e_{\tau i}$ direction, or to be more precise the $^1\boldsymbol{a}_{\tau i}$ direction[4] (see Fig. 5), at a sequence of time instances are shown in Fig. 6(a).

5.2. Adding synthetic errors to virtual accelerometer data

The acceleration components $^\ell\alpha_i$, $\ell \in \mathcal{J}$, from the simulation do not contain any errors; other than, of course, the errors that arise due to numerical discretization of the balance equations, numerical round-off, etc. However, those type of errors are of insignificant magnitude. Using the error free virtual accelerometer data $^\ell\alpha_i$, $\ell \in \mathcal{J}$, from the simulation we generate virtual error-inclusive accelerometer data $^\ell\alpha_i^{\text{Error}}$, $\ell \in \mathcal{J}$, as

$$^\ell\alpha_i^{\text{Error}}(\tau) = {}^\ell\alpha_i(\tau) + \eta_\tau. \tag{5.1}$$

In equation (5.1) η_τ denotes a particular realization of the Ornstein-Uhlenbeck (OU) process [38]. We will describe shortly what we mean by a "realization". The OU process is a continuous time and state stochastic process that is defined by the integral equation

$$\eta_{\tau_1+\tau_2} - \eta_{\tau_1} = \beta \int_{\tau_1}^{\tau_1+\tau_2} (\mu - \eta_\tau)\, d\tau + \sigma \int_{\tau_1}^{\tau_1+\tau_2} dW_\tau, \tag{5.2}$$

where the second integral on the right is an Itô integral and W_τ is the Wiener process [39]. The real number μ is called the mean value, $\sigma \geqslant 0$ the diffusion coefficient, and $\beta > 0$ the drift coefficient. The symbols τ_1, τ_2 denote any two (non-dimensional) time instances. Since the OU process is a stochastic process, a given set of OU parameters, i.e., a particular set of μ, σ, and β values, define an entire family or population of real valued functions on $\mathbb{R}$. For a given OU parameter set, a particular realization of the OU process is obtained by drawing η_0 from a Gaussian distribution of mean μ and variance $\sigma^2/(2\beta)$ and solving (5.2). As a consequence of (5.1), $^\ell\alpha_i^{\text{Error}}$, $\ell \in \mathcal{J}$, too are stochastic processes.

For $i \in \mathcal{I}$ and $\ell \in \mathcal{J}$, when $\mu \neq 0$ and $\sigma = 0$, any particular realization of $^\ell\alpha_i^{\text{Error}}$ will contain only bias type errors. A representative realization of $^1\alpha_2^{\text{Error}}$ for $\mu = 5$, $\sigma = 0$, and $\beta = 10^3$ is shown in Fig. 6(b). Alternatively, when $\mu = 0$ and $\sigma \neq 0$ any particular realization of $^\ell\alpha_i^{\text{Error}}$ will only contain noise type errors. A representative realization of $^1\alpha_2^{\text{Error}}$ for $\mu = 0$, $\sigma = 10^2$, and $\beta = 10^3$ is shown in Fig. 6(c). In general when μ and σ are both non-zero, realizations of $^\ell\alpha_i^{\text{Error}}$ will contain both bias and noise type errors. A representative realization of $^1\alpha_2^{\text{Error}}$ for $\mu = 5$, $\sigma = 10^2$, and $\beta = 10^3$ is shown in Fig. 6(d).

From here on unless otherwise specified the value of β will always be equal to 10^3.

5.3. Comparison of $\sqrt{AO}$ and AO algorithms using virtual error-inclusive accelerometer data

We compare the predictions of the $\sqrt{AO}$ and AO algorithms for the following categories of OU parameter sets.

Category I Exclusively bias type errors: $\mu = 0$, 0.1, 0.2, 0.5, and 1, and $\sigma = 0$ (see Table 1).

Category II Exclusively noise type errors: $\mu = 0$, and $\sigma = 0$, 1, 10, 50, and 10^2 (see Table 2).

Category III Both bias and noise type errors: $\mu = 0$, 0.1, 0.2, 0.5, and 1, and $\sigma = 10$ (see Table 3).

For a given OU parameter set we generate a large number of $^\ell\alpha_i^{\text{Error}}$, $\ell \in \mathcal{J}$, realizations. We apply the $\sqrt{AO}$ and AO algorithms to each of those realizations and derive a population of predictions for the acceleration of the material particle $^5\mathcal{X}$ (see Fig. 4). We denote the error-free (non-dimensional) acceleration of $^5\mathcal{X}$, which we know from the rigid-ellipsoid-impact-simulation's results, at the time instance τ as $^5\mathbf{A}(\tau) \in \mathcal{M}_{3,1}(\mathbb{R})$. The components of $^5\mathbf{A}(\tau)$, i.e., $\left({}^5\mathbf{A}(\tau)\right)_i$, $i \in \mathcal{I}$, for a sequence of time instances are, respectively, shown in subfigures (a), (b), and (c) in each of Figs. 7–9. They are shown using thick gray curves.

Since the predictions of the $\sqrt{AO}$ and AO algorithms are derived by, respectively, feeding the $\sqrt{AO}$ and AO algorithms the stochastic processes $^\ell\alpha_i^{\text{Error}}$, $\ell \in \mathcal{J}$, they too, in fact, are stochastic processes. Representative realizations of the predictions from the $\sqrt{AO}$ (resp. AO) algorithm for different OU parameter sets are, respectively, shown in Figs. 7–9 in green (resp. red).

6. Results and Discussion

6.1. Category I

Among the OU parameter sets belonging to *Category I* the set corresponding to the most amount of error is $(\mu, \sigma) = (1.0, 0.0)$. Representative realizations of the predictions from the $\sqrt{\text{AO}}$ and AO algorithms for this parameter set are, respectively, shown in green and red in Fig. 7. In Fig. 7 the realization of the $\sqrt{\text{AO}}$-algorithm's prediction appears to be more accurate than that of the AO-algorithm's prediction, especially with increasing time. In order to make a more quantitative comparison between the $\sqrt{\text{AO}}$ and AO algorithms' predictions, we focus on the time interval $[0, 1]$ and make use of the the metrics

$$\epsilon_2 \left(\sqrt{\text{AO}} \right) := \frac{\| \sqrt{\text{AO}} \left({}^5\mathbf{A} \right) - {}^5\mathbf{A} \|_2}{\| {}^5\mathbf{A} \|_2}, \tag{6.1}$$

$$\epsilon_2 \left(\text{AO} \right) := \frac{\| \text{AO} \left({}^5\mathbf{A} \right) - {}^5\mathbf{A} \|_2}{\| {}^5\mathbf{A} \|_2}, \tag{6.2}$$

where $\| f \|_2 := \sqrt{\int_0^1 \| f(\tau) \|^2 \, d\tau}$; and $\mathbb{R} \ni \tau \mapsto \sqrt{\text{AO}} \left({}^5\mathbf{A} \right) (\tau) \in \mathscr{M}_{3,1}(\mathbb{R})$, and $\mathbb{R} \ni \tau \mapsto \text{AO} \left({}^5\mathbf{A} \right) (\tau) \in \mathscr{M}_{3,1}(\mathbb{R})$ are, respectively, particular realizations of the $\sqrt{\text{AO}}$ and AO algorithms' predictions for $\mathbb{R} \ni \tau \mapsto {}^5\mathbf{A}(\tau) \in \mathscr{M}_{3,1}(\mathbb{R})$. The metric $\epsilon_2 \left(\sqrt{\text{AO}} \right)$ (resp. $\epsilon_2 \left(\text{AO} \right)$) is constructed such that the smaller its value the more accurate the realization used in computing it. The values of $\epsilon_2 \left(\sqrt{\text{AO}} \right)$ and $\epsilon_2 \left(\text{AO} \right)$ for the realizations shown in Fig. 7 are, respectively, 8.79% and 47.11%. The metric $\epsilon_2 \left(\sqrt{\text{AO}} \right)$'s smaller value in comparison to that of $\epsilon_2 \left(\text{AO} \right)$ corroborates our earlier assertion that among the $\sqrt{\text{AO}} \left({}^5\mathbf{A} \right)$ and $\sqrt{\text{AO}} \left({}^5\mathbf{A} \right)$ shown in Fig. 7 the realization $\sqrt{\text{AO}} \left({}^5\mathbf{A} \right)$ is more accurate. This comparison between the $\sqrt{\text{AO}}$ and AO algorithms' predictions' particular realizations prompts us to hypothesize that the $\sqrt{\text{AO}}$ algorithm is more accurate than the AO algorithm.

In order to compare the $\sqrt{\text{AO}}$ and AO algorithms' predictions in a more well-balanced and comprehensive manner we calculated $\epsilon_2 \left(\sqrt{\text{AO}} \right)$ and $\epsilon_2 \left(\text{AO} \right)$, respectively, for a large number of realizations (population size $N = 200$) of the predictions from the $\sqrt{\text{AO}}$ and AO algorithms. The mean values of the thus generated populations of $\epsilon_2 \left(\sqrt{\text{AO}} \right)$ and $\epsilon_2 \left(\text{AO} \right)$ are 8.788% and 47.114%, respectively (see row number 5 of Table 1). The mean value of the population of $\epsilon_2 \left(\sqrt{\text{AO}} \right)$ being lower than the mean value of the population of $\epsilon_2 \left(\text{AO} \right)$ further supports our earlier hypothesis that $\sqrt{\text{AO}}$-algorithm is more accurate than the AO-algorithm.

To recall, the discussion so far in this section exclusively relates to the (μ, σ) parameter set $(1.0, 0.0)$. We performed analysis similar to the one discussed in the previous paragraph for the parameter sets $(0.0, 0.0)$, $(0.1, 0.0)$, $(0.2, 0.0)$, and $(0.5, 0.0)$ as well. The means of $\epsilon_2 \left(\sqrt{\text{AO}} \right)$'s and $\epsilon_2 \left(\text{AO} \right)$'s populations for these other parameter sets are, respectively, given in the first and second columns of Table 1. It can be seen from Table 1 that the means of the $\epsilon_2 \left(\sqrt{\text{AO}} \right)$ populations are consistently smaller than those of $\epsilon_2 \left(\text{AO} \right)$ populations across all the parameter sets considered. Furthermore, in Table 1 the difference between the means of a $\epsilon_2 \left(\sqrt{\text{AO}} \right)$ population and a $\epsilon_2 \left(\text{AO} \right)$ population corresponding to the same parameter set increases with the amount of error, i.e., with the magnitude of μ in the present category. Thus for the category of exclusively bias type errors, in addition to the $\sqrt{\text{AO}}$-algorithm appearing to be more accurate than the AO-algorithm, it further appears that the $\sqrt{\text{AO}}$-algorithm's performance over the AO-algorithm increases with increasing amount of error.

6.2. Category II

In *Category II* we consider the (μ, σ) parameter sets $(0.0, 0.0)$, $(0.0, 1.0)$, $(0.0, 10.0)$, $(0.0, 50.0)$, and $(0.0, 100.0)$. The means of $\epsilon_2 \left(\sqrt{\text{AO}} \right)$'s and $\epsilon_2 \left(\text{AO} \right)$'s populations for these parameter sets are, respectively, given in the first and second columns of Table 2. It can be seen from Table 2 that the means of the $\epsilon_2 \left(\sqrt{\text{AO}} \right)$ populations are approximately the same as those of $\epsilon_2 \left(\text{AO} \right)$ populations across all the parameter sets considered. Thus, for the category of exclusively noise type errors the $\sqrt{\text{AO}}$-algorithm appears to perform on par with the AO-algorithm.

Among the OU parameter sets belonging to *Category II* the set corresponding to the most amount of error is $(\mu, \sigma) = (0.0, 100.0)$. Representative realizations of the predictions from the $\sqrt{\text{AO}}$ and AO

algorithms for this parameter set are, respectively, shown in green and red in Fig. 8. In Fig. 8, at least at the earlier time instances, the $\sqrt{AO}$ and AO algorithms' predictions' realizations are almost indistinguishable from one another. However, at later time instances the $\sqrt{AO}$-algorithm seems to be performing better than the AO-algorithm (This feature is likely not reflected in the results presented in Table 2 because they are calculated only using data from the initial time instances, or, to be more precise, from the $[0, 1]$ time interval.). Based on this observation we venture to speculate that even when the errors are predominantly of the noise type, the $\sqrt{AO}$-algorithm will eventually begin to outperform the AO-algorithm.

6.3. Category III

In *Category III* we consider the (μ, σ) parameter sets $(0.0, 10.0)$, $(0.1, 10.0)$, $(0.2, 10.0)$, $(0.5, 10.0)$, and $(1, 10.0)$. The means of $\epsilon_2 \left(\sqrt{AO}\right)$'s and ϵ_2 (AO)'s populations for these parameter sets are, respectively, given in the first and second columns of Table 3. Among the OU parameter sets belonging to *Category III* the set corresponding to the most amount of error is $(\mu, \sigma) = (1.0, 10.0)$. Representative realizations of the predictions from the $\sqrt{AO}$ and AO algorithms for this parameter set are, respectively, shown in green and red in Fig. 9.

It can be seen from Table 3 that the means of the $\epsilon_2 \left(\sqrt{AO}\right)$ populations are consistently smaller than those of ϵ_2 (AO) populations across all the parameter sets considered. Furthermore, in Table 3 the difference between the means of a $\epsilon_2 \left(\sqrt{AO}\right)$ population and a ϵ_2 (AO) population corresponding to the same parameter set increases with the amount of error, i.e., with the magnitudes of σ and μ. Thus, in *Category III* the relative performance of the $\sqrt{AO}$ and AO algorithms is very similar to that in *Category I*.

From the discussion in §6.1 we know that the AO-algorithm is more sensitive to bias type errors than the $\sqrt{AO}$-algorithm and from the discussion in §6.2 we know that the $\sqrt{AO}$ and AO algorithms are, approximately, equally sensitive to noise type errors. From the results in this section it appears that the $\sqrt{AO}$-algorithm outperforms the AO-algorithm as long as the errors have some bias type component in them, irrespective of what the amount of the noise type component in them is.

Table 1: The mean and standard deviation of the error measure ϵ_2 for 200 realizations of the accelerometer data only containing bias type error with $\sigma = 0$. In this case, as the value of standard deviation is quite small compared to the value of mean, we do not show the value in the table

	$\epsilon_2 \times 10^3$ (mean± std)	
μ	$\sqrt{AO}$-algorithm	AO-algorithm
0	0.01	0.02
0.1	8.87	41.02
0.2	17.73	83.63
0.5	44.22	220.28
1	87.88	471.14

Table 2: The mean and standard deviation of the error measure ϵ_2 for 200 realizations of the accelerometer data only containing noise type error with $\mu = 0$

	$\epsilon_2 \times 10^3$ (mean± std)	
σ	$\sqrt{AO}$-algorithm	AO-algorithm
0	0.01	0.02
1	1.03 ± 0.09	1.16 ± 0.21
10	10.22 ± 0.64	11.46 ± 1.80
50	52.79 ± 3.33	56.87 ± 7.79
100	115.85 ± 10.11	112.83 ± 15.54

Table 3: The mean and standard deviation of the error measure ϵ_2 for 200 realizations of the accelerometer data containing bias and noise type errors with $\sigma = 10$

μ	$\epsilon_2 \times 10^3$ (mean$\pm$ std)	
	$\sqrt{\text{AO}}$-algorithm	AO-algorithm
0	10.22 ± 0.64	11.46 ± 1.80
0.1	13.42 ± 1.14	42.61 ± 5.05
0.2	20.19 ± 1.43	84.52 ± 5.57
0.5	45.18 ± 1.47	221.08 ± 6.02
1	88.27 ± 1.44	472.26 ± 6.47
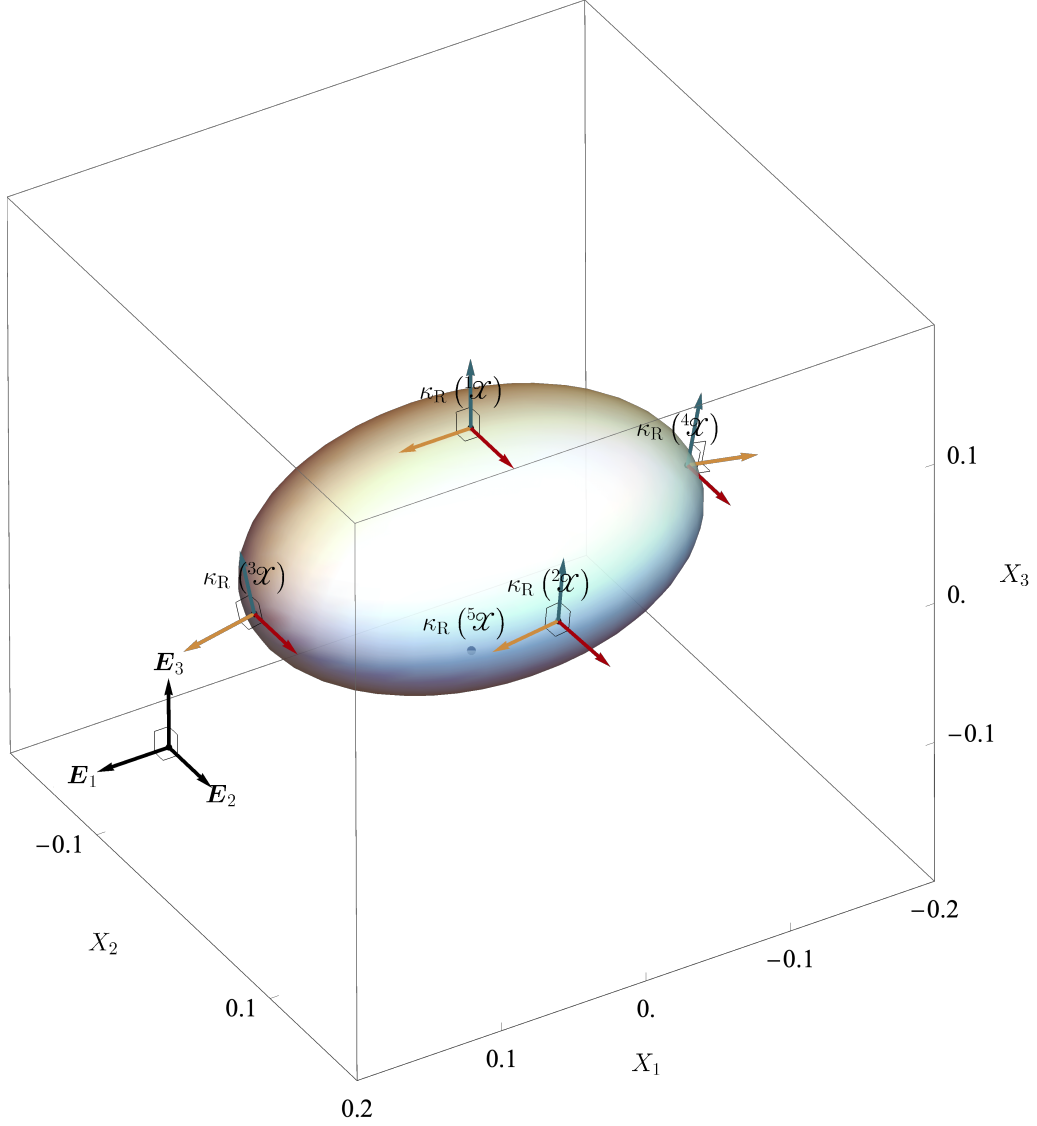
Figure 4: Accelerometer arrangement and orientation in numerical simulation of a rigid ellipsoid impacting an elastic half-space (see §5.1 for details). In the reference point space $\mathscr{E}_{\mathrm{R}}$ the ellipsoid, $\mathscr{B}$, occupies the region $\left\{ (X_1, X_2, X_3) \,\middle|\, (X_1/a)^2 + (X_2/b)^2 + (X_3/c)^2 \leqslant 1 \right\}$, where $(a, b, c) = (0.15, 0.10, 0.08)$. Four virtual accelerometers are, respectively, attached to the ellipsoid's material particles ${}^{\ell}\mathcal{X}$, $\ell \in \mathscr{J}$. The reference position vectors of ${}^{\ell}\boldsymbol{X}$, $\ell \in \mathscr{J}$, are, respectively, $c\boldsymbol{E}_3$, $b\boldsymbol{E}_2$, $a\boldsymbol{E}_1$, and $-a\boldsymbol{E}_1$, where $(\boldsymbol{E}_i)_{i \in \mathscr{J}}$ are shown in the figure as well. The accelerometers' orientations are given by $\left({}^{\ell}\boldsymbol{E}_i \right)_{i \in \mathscr{J}}$, $\ell \in \mathscr{J}$. The component representation of $\left({}^{1}\boldsymbol{E}_i \right)_{i \in \mathscr{J}}$ w.r.t. $(\boldsymbol{E}_i)_{i \in \mathscr{J}}$ is $((0, 1, 0), (1, 0, 0), (0, 0, 1))$; of $\left({}^{2}\boldsymbol{E}_i \right)_{i \in \mathscr{J}}$ is $\left(\left(-\frac{2}{\sqrt{229}}, \frac{225}{229}, -\frac{30}{229} \right), \left(\frac{15}{\sqrt{229}}, \frac{30}{229}, -\frac{4}{229} \right), \left(0, \frac{2}{\sqrt{229}}, \frac{15}{\sqrt{229}} \right) \right)$; of $\left({}^{3}\boldsymbol{E}_i \right)_{i \in \mathscr{J}}$ is $\left((0, 1, 0), \left(\frac{5}{\sqrt{26}}, 0, -\frac{1}{\sqrt{26}} \right), \left(\frac{1}{\sqrt{26}}, 0, \frac{5}{\sqrt{26}} \right) \right)$; and of $\left({}^{4}\boldsymbol{E}_i \right)_{i \in \mathscr{J}}$ is $\left((0, 1, 0), \left(-\frac{5}{\sqrt{26}}, 0, -\frac{1}{\sqrt{26}} \right), \left(-\frac{1}{\sqrt{26}}, 0, \frac{5}{\sqrt{26}} \right) \right)$. We apply the $\sqrt{\text{AO}}$-algorithm to the accelerometer data from the four virtual accelerometers ${}^{\ell}\mathcal{X}$, $\ell \in \mathscr{J}$ to predict the acceleration of the material particle ${}^{5}\mathcal{X}$. The reference position vector of ${}^{5}\mathcal{X}$ is $-c\boldsymbol{E}_3$. (modified from [27], copyright 2020, Elsevier)
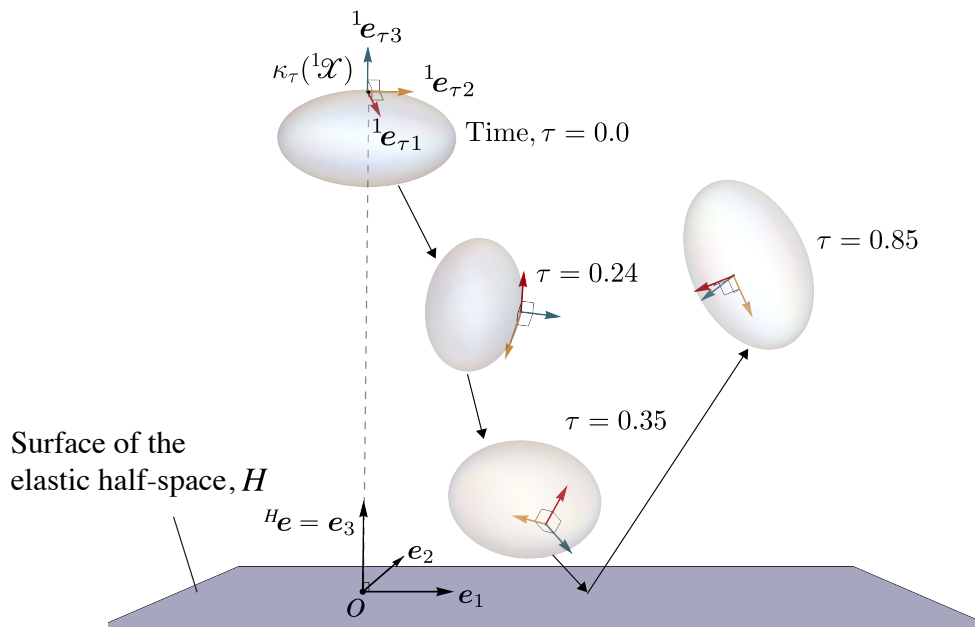
Figure 5: Configuration of the rigid ellipsoid at different time instances in the simulation of it impacting an elastic half space (see §5.1 for details). In the simulation, the ellipsoid, $\mathscr{B}$, is dropped onto an elastic half-space, H, under the action of gravity with the initial angular and translational velocities prescribed. The ellipsoid's initial position in $\mathscr{E}$ is prescribed by taking $\mathbf{c}(0) = (0, 0, 0.75)$, and $\mathbf{Q}(0) = \mathrm{diag}\,(1, 1, 1)$. Its initial velocities are prescribed by setting $\star(\mathbf{W}(0)) = (5, 5, 5)$, and $\mathbf{c}'(0) = (0.75, 0, 0)$. (modified from [27], copyright 2020, Elsevier)
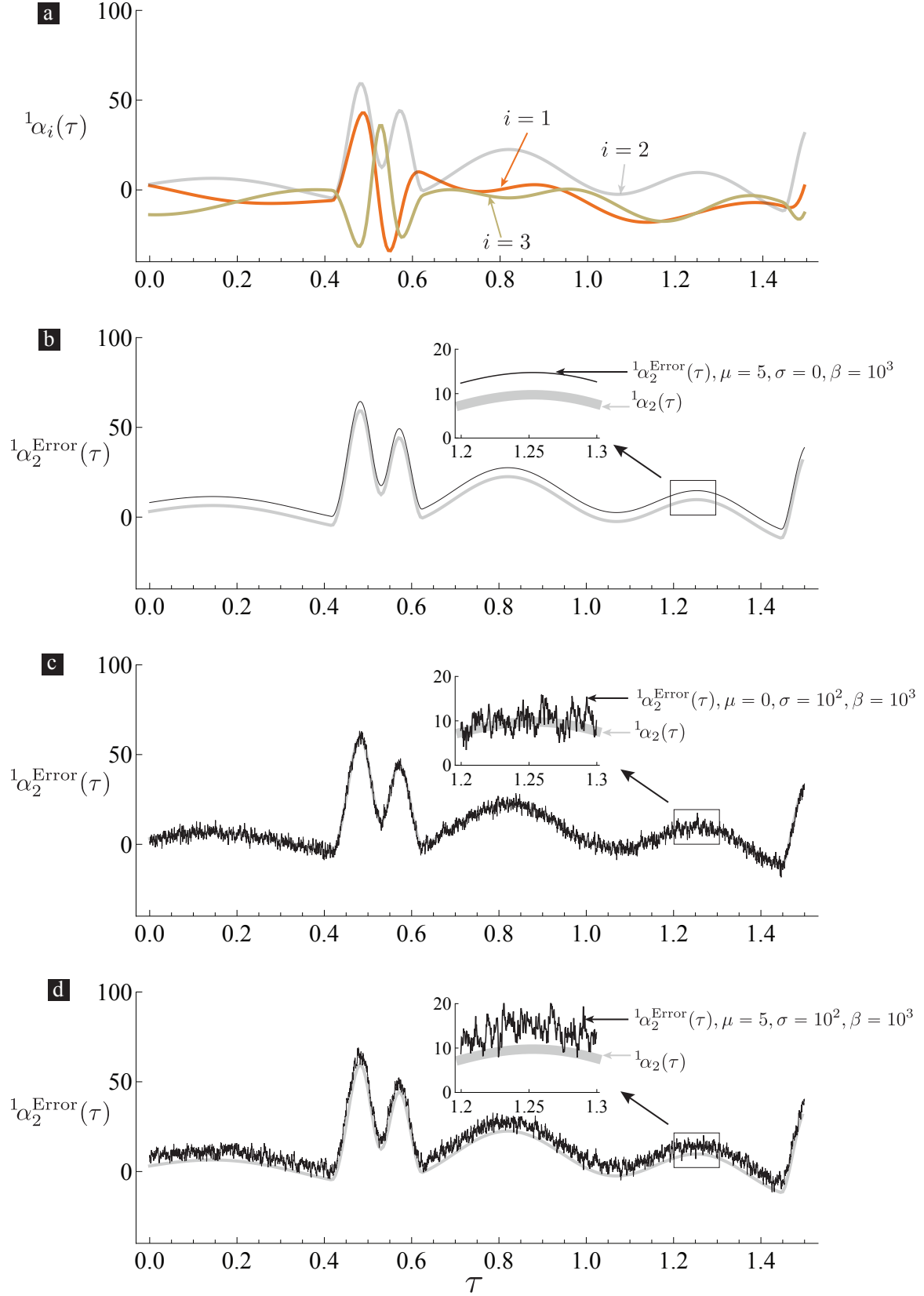
Figure 6: The acceleration components $^1\alpha_i(\tau)$, $i \in \mathscr{I}$, of the virtual accelerometer $^1\mathscr{X}$ before and after addition of synthetic errors (see §5.2 for details). (a) shows the acceleration components before the addition of synthetic errors. (b)–(d) show the error-inclusive acceleration component $^1\alpha_2^{\mathrm{Error}}$, which is generated by adding different errors to the acceleration component $^1\alpha_2$. In (b), (c), and (d) the error time signals are particular realizations of the OU process for the OU parameter sets $(\mu, \sigma, \beta) = (5, 0, 10^3)$, $(0, 10^2, 10^3)$, and $(5, 10^2, 10^3)$, respectively. The error in (b) corresponds to *Category I* (exclusively bias type errors); in (c) to *Category II* (exclusively noise type errors); and in (d) to *Category III* (a combination of bias and noise type errors).
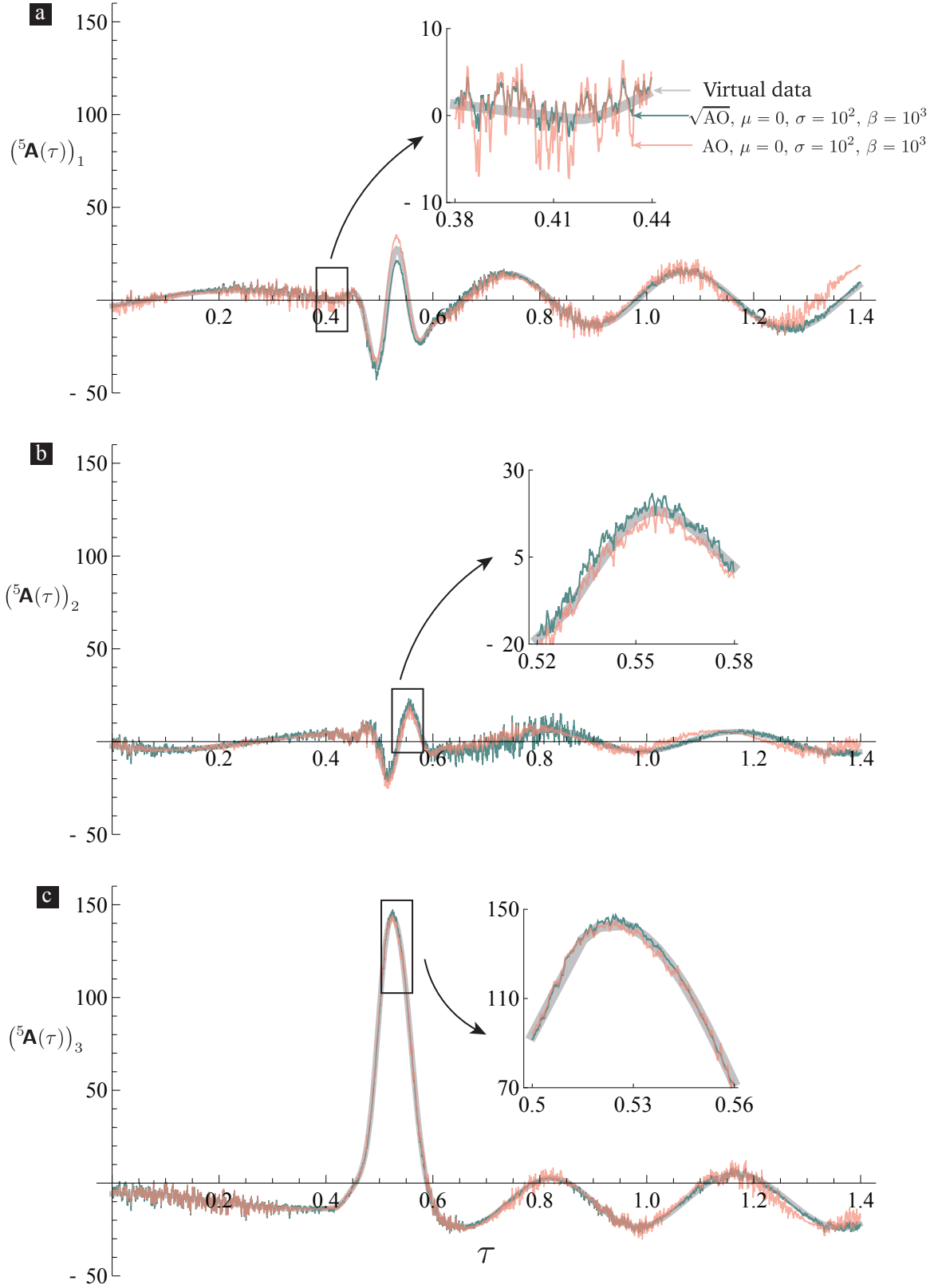
19
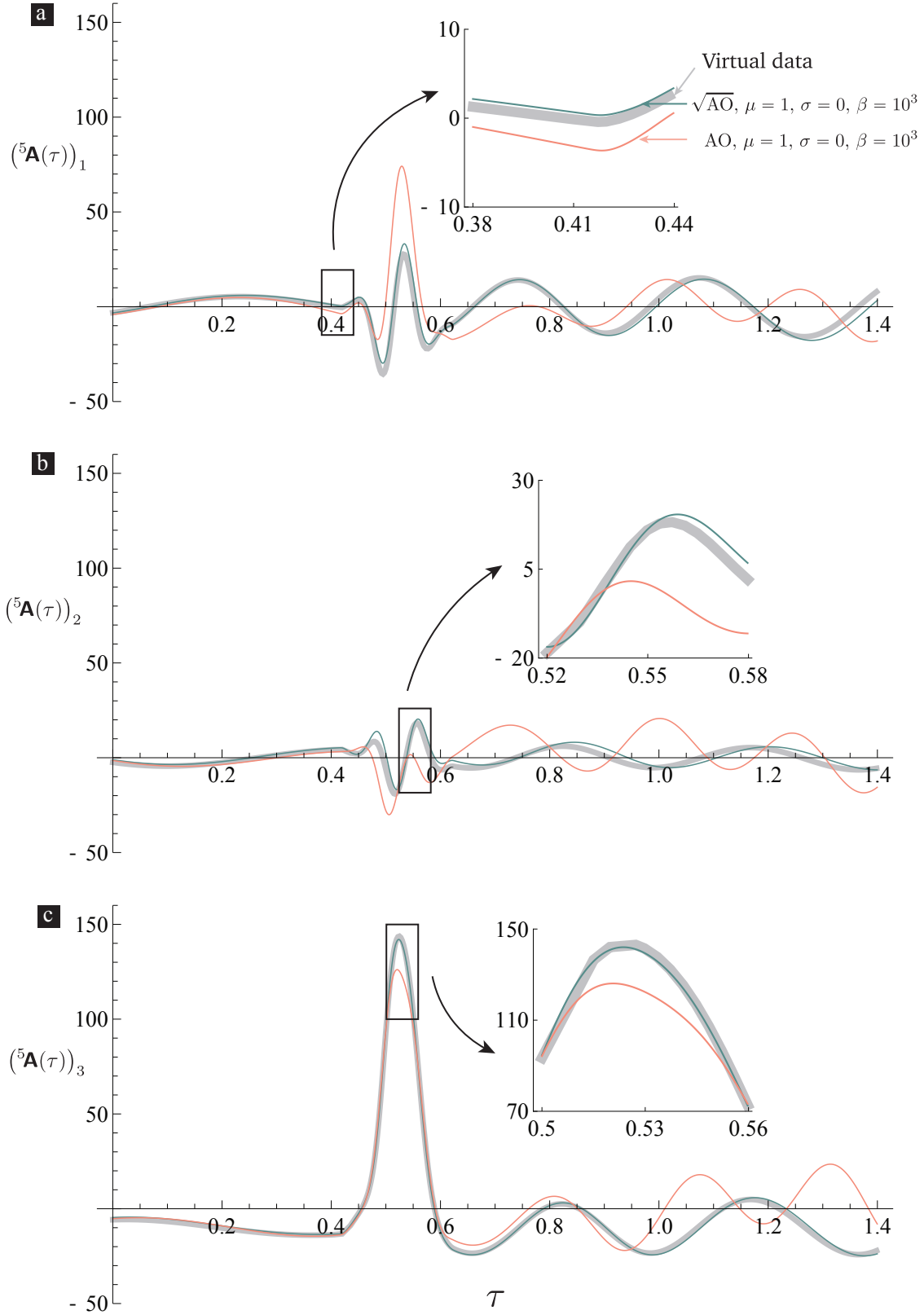
Figure 7: Comparison of the predictions from the $\sqrt{\text{AO}}$ and AO algorithms for the acceleration of the material particle $^5\mathcal{X}$ (see Fig. 4) in the rigid ellipsoid impact simulation (see §5.1 for details). Both the $\sqrt{\text{AO}}$ and AO algorithms were fed the same virtual error-inclusive accelerometer data. The data was generated by adding a particular realization of the OU process to the virtual accelerometer data from the rigid ellipsoid impact simulation. The OU realization corresponded to the OU parameter set $(\mu, \sigma, \beta) = (1, 0, 10^3)$. Subfigures (a), (b), and (c), respectively, show the comparison for the component of $^5\mathcal{X}$'s acceleration in the $\boldsymbol{e}_i$, $i \in \mathscr{I}$, directions.
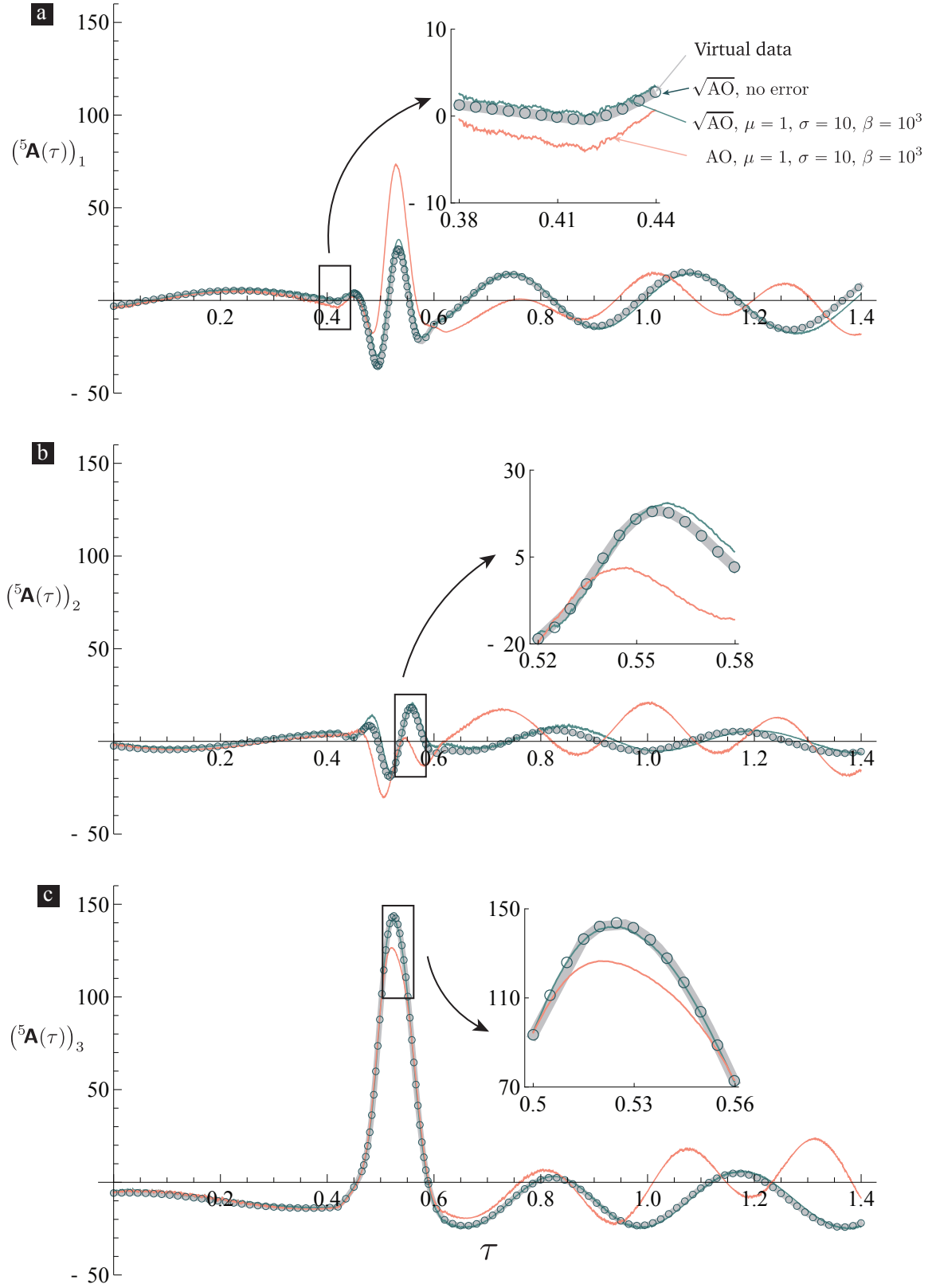
Figure 8: Comparison of the predictions from the $\sqrt{\text{AO}}$ and AO algorithms for the acceleration of the material particle $^5\mathscr{X}$ (see Fig. 4) in the rigid ellipsoid impact simulation (see §5.1 for details). Both the $\sqrt{\text{AO}}$ and AO algorithms were fed the same virtual error-inclusive accelerometer data. The data was generated by adding a particular realization of the OU process to the virtual accelerometer data from the rigid ellipsoid impact simulation. The OU realization corresponded to the OU parameter set $(\mu, \sigma, \beta) = (0, 10^2, 10^3)$. Subfigures (a), (b), and (c), respectively, show the comparison for the component of $^5\mathscr{X}$'s acceleration in the $\boldsymbol{e}_i$, $i \in \mathscr{I}$, directions.

Figure 9: Comparison of the predictions from the $\sqrt{AO}$ and AO algorithms for the acceleration of the material particle $^5\mathscr{X}$ (see Fig. 4) in the rigid ellipsoid impact simulation (see §5.1 for details). Both the $\sqrt{AO}$ and AO algorithms were fed the same virtual error-inclusive accelerometer data. The data was generated by adding a particular realization of the OU process to the virtual accelerometer data from the rigid ellipsoid impact simulation. The OU realization corresponded to the OU parameter set $(\mu, \sigma, \beta) = (1, 10, 10^3)$. Subfigures (a), (b), and (c), respectively, show the comparison for the component of $^5\mathscr{X}$'s acceleration in the $\boldsymbol{e}_i$, $i \in \mathscr{I}$, directions. The predictions of the $\sqrt{AO}$ algorithm when fed just the virtual accelerometer data, i.e., with no added errors, is also shown in (a), (b), and (c) using black open circles.

7. Concluding remarks

1. The results discussed in §6 show that the $\sqrt{\text{AO}}$-algorithm provides a valid approach to determine the complete motion of a rigid body using only data from four tri-axial accelerometers. We plan on carrying out more simulations in the future for gauging the robustness of the $\sqrt{\text{AO}}$ algorithm. Those simulations will be more complicated than the one we considered in this paper, e.g., in the future simulations we will consider non-symmetric geometries for the rigid body and non-planar surfaces for the elastic solid. The $\sqrt{\text{AO}}$-algorithm's practical validity in the field too still remains to be explored. In the future, we plan to conduct an experimental evaluation of the $\sqrt{\text{AO}}$-algorithm to compliment its *in silico* validation that we presented in this paper.

2. The comparison in §6 shows that for the cases we considered the $\sqrt{\text{AO}}$-algorithm is less sensitive to bias type errors compared to the AO-Algorithm. However, we have not provided a mathematical proof that the $\sqrt{\text{AO}}$-algorithm is better than the AO-algorithm with regard to bias type errors. Thus, though the comparison presented in §6 provides strong support to the hypothesis that the $\sqrt{\text{AO}}$-algorithm is less sensitive to bias type errors than the AO-algorithm, it by no means provides a proof for the hypothesis. A definitive resolution to the question of whether the hypothesis is true requires an error analysis of both the $\sqrt{\text{AO}}$-algorithm as well as the AO-algorithm. We currently have not carried out such analyses. Nevertheless, irrespective of the relative merit of the $\sqrt{\text{AO}}$ over the AO algorithm, it is quite clear from its derivation and the results discussed in §6 that it provides a valid approach for determining the complete motion of a rigid body from accelerometer data.

3. The $\sqrt{\text{AO}}$-algorithm retains all the benefits of the AO-algorithm. Both algorithms provide the complete motion of the rigid body in the fixed laboratory frame. Without integration or differentiation, both algorithms are able to determine the pseudo acceleration field, providing the magnitude of acceleration for all material particles. Both algorithms can be applied to any arrangement of four tri-axial accelerometers as long as they do not lie in the same plane. There is no restriction on the orientation of the tri-axial accelerometers.

4. In the *in silico* validation, evaluation, and comparison of the $\sqrt{\text{AO}}$-algorithm that we set up in §5 we used the OU process to model experimental errors. Even more specifically, we took the magnitude of the parameter μ in the OU process as a measure of the bias type errors in the OU process' realizations. There of course exist bias type errors that cannot be modeled in this manner. Thus, our evaluation of the relative sensitivities of the $\sqrt{\text{AO}}$ and the AO algorithms to bias type errors was carried out using a limited form of bias type errors. A more general method to represent bias type errors in the virtual accelerometer data would provide a more comprehensive comparison of the relative sensitivities of the $\sqrt{\text{AO}}$ and the AO algorithms to bias type errors.

5. We envision the $\sqrt{\text{AO}}$ algorithm being used in design efforts that are aimed at providing better protection against mTBI. Design efforts that are aimed at providing better protection against mTBI, typically make use of *brain injury criteria*. Currently, several finite element (FE) based brain injury criteria are being developed with an aim of providing a better estimate of a motion's potential for causing injury, e.g., see [40]. Strain and strain rate have been shown to play an important role in brain injury. The FE based injury criteria connect the risk of injury to time histories of different stress and strain fields within the head. Application of the FE based injury criteria requires simulating the motion that is being evaluated on a computational head model and determining the time histories of the stress and strains fields within it. The developed $\sqrt{\text{AO}}$ algorithm provides a means for accurately simulating varieties of motions that are important in the context of mTBI. Since, it makes it possible to measure even 3D motions that are difficult to reproduce out of field, such as the landing of a paratrooper.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Definition of the map $\star\left(\cdot\right)$

For $n_{sd} = 2$, the map $\star\left(\cdot\right) : \mathfrak{so}(\mathbb{R}, 2) \to \mathbb{R}$ is defined by the equation $\star\left(\cdot\right) = (\cdot)_{21}$. The inverse of $\star\left(\cdot\right)$ is the map $\star^{-1}\left(\cdot\right) : \mathbb{R} \to \mathfrak{so}(\mathbb{R}, 2)$ defined by the equation

$$\star\left(\alpha\right) = \begin{pmatrix} 0 & -\alpha \\ \alpha & 0 \end{pmatrix}. \tag{A.1}$$

For $n_{sd} = 3$, the map $\star\left(\cdot\right) : \mathfrak{so}(\mathbb{R}, 3) \to \mathscr{M}_{3,1}(\mathbb{R})$ is defined by the equation $\star\left(\cdot\right) = ((\cdot)_{32}, (\cdot)_{13}, (\cdot)_{21})$. The inverse of $\star\left(\cdot\right)$ is the map $\star^{-1}\left(\cdot\right) : \mathscr{M}_{3,1}(\mathbb{R}) \to \mathfrak{so}(\mathbb{R}, 3)$ defined by the equation

$$\star\left((\alpha_1, \alpha_2, \alpha_3)\right) = \begin{pmatrix} 0 & -\alpha_3 & \alpha_2 \\ \alpha_3 & 0 & -\alpha_1 \\ -\alpha_2 & \alpha_1 & 0 \end{pmatrix}. \tag{A.2}$$

To make our notation appear less cumbersome we denote $\star^{-1}\left(\cdot\right)$ too as $\star\left(\cdot\right)$. Whether we mean $\star\left(\cdot\right)$ or $\star^{-1}\left(\cdot\right)$ will be clear from the argument of $\star\left(\cdot\right)$.

Appendix B. Derivation of (1.3), i.e., proof of the statement that square of $\overline{\mathbf{W}}(\tau)$ is equal to the symmetric part of $\mathbf{P}(\tau)$

The following lemmas can be shown to be equivalent to some of the standard results in the mechanics of rigid solids, see the work in [41, §2.5.2] and [42, §6.4], which treats the rigid body motion in a modern continuum mechanics style; or see the work in [43, §9.4] and [44, §15], which treats the rigid body motion from a perspective of geometric mechanics. However, at a cursory level, due to our notation and formalism, those results might appear to be different from the below lemmas. The differences in notation and formalism are primarily due to the fact that in our work we distinguish between the vector spaces to which the various physical quantities, e.g., the rotation operation, belong to and the (non-dimensional) matrix vector spaces to which the component representations of those quantities belong to. For that reason, we believed that it would be helpful to the reader if we presented the following lemmas using the notation and formalism that we use in the current work.

Appendix B.1. Skew symmetry of $\overline{\mathbf{W}}(\tau)$

Lemma Appendix B.1. *The matrix $\overline{\mathbf{W}}(\tau)$, defined in (3.7), is skew-symmetric.*

Proof. It can be shown using $\overline{\mathbf{W}}(\tau)$'s definition (3.7) and equations (2.6) and (2.3a) that

$$\overline{\mathbf{W}}(\tau) = \mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}'(\tau). \tag{B.1}$$

Differentiating (2.3a) we get

$$\left(\mathbf{Q}^{\mathsf{T}}\right)'(\tau)\,\mathbf{Q}(\tau) + \mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}'(\tau) = \mathbf{0}. \tag{B.2}$$

Noting that $\left(\mathbf{Q}^{\mathsf{T}}\right)'(\tau) = \left(\mathbf{Q}'\right)^{\mathsf{T}}(\tau)$ we see that the first term on the left hand side of (B.2) is equal to $\left(\mathbf{Q}'\right)^{\mathsf{T}}(\tau)\,\mathbf{Q}(\tau)$, which, in fact, is equal to the transpose of the second term on the left hand side of (B.2).

Thus, it follows from (B.2) that $\mathrm{sym}\left(\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}'(\tau)\right) = \mathbf{0}$. That is, that $\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}'(\tau)$ is skew-symmetric. The result that $\overline{\mathbf{W}}(\tau)$ too is skew-symmetric now immediately follows from (B.1).

Appendix B.2. Derivation of equation (1.3)

Lemma Appendix B.2. *The symmetric part of $\mathbf{P}(\tau)$ is equal to the square of $\overline{\mathbf{W}}(\tau)$.*

Proof. Differentiating (2.3a) twice and rearranging we get that

$$\left(\mathbf{Q}^{\mathsf{T}}\right)''(\tau)\,\mathbf{Q}(\tau) + \mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}''(\tau) = -2\left(\mathbf{Q}^{\mathsf{T}}\right)'(\tau)\,\mathbf{Q}'(\tau). \tag{B.3}$$

The equation (B.3) on noting that the first and second terms on its left hand side are in fact transposes of each other simplifies to

$$\mathrm{sym}\left(\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}''(\tau)\right) = -\left(\mathbf{Q}^{\mathsf{T}}\right)'(\tau)\,\mathbf{Q}'(\tau). \tag{B.4}$$

Writing the term on the right hand side of (B.4) as $-\left(\mathbf{Q}^{\mathsf{T}}\right)'(\tau)\,\mathbf{I}\,\mathbf{Q}'(\tau)$, and then using (2.3b) and replacing the $\mathbf{I}$ in the resulting equation with $\mathbf{Q}(\tau)\,\mathbf{Q}^{\mathsf{T}}(\tau)$, we get

$$\mathrm{sym}\left(\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}''(\tau)\right) = -\left(\left(\mathbf{Q}^{\mathsf{T}}\right)'(\tau)\,\mathbf{Q}(\tau)\right)\left(\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}'(\tau)\right). \tag{B.5}$$

Noting that $\left(\mathbf{Q}^{\mathsf{T}}\right)'(\tau) = \left(\mathbf{Q}'\right)^{\mathsf{T}}(\tau)$ we see that the first factor on the right hand side of (B.5) is equal to $\left(\mathbf{Q}'\right)^{\mathsf{T}}(\tau)\,\mathbf{Q}(\tau)$, which is the transpose of the second factor on the right hand side of (B.5), namely $\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}'(\tau)$. We, however, know from (B.1) that this second factor is equal to $\overline{\mathbf{W}}(\tau)$. Thus, we get from (B.5) that

$$\mathrm{sym}\left(\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}''(\tau)\right) = -\overline{\mathbf{W}}^{\mathsf{T}}(\tau)\,\overline{\mathbf{W}}(\tau), \tag{B.6a}$$

which simplifies on using Lemma Appendix B.1 to

$$\mathrm{sym}\left(\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}''(\tau)\right) = \overline{\mathbf{W}}^2(\tau). \tag{B.6b}$$

Using (3.5) and replacing the quantity $\mathbf{Q}^{\mathsf{T}}(\tau)\,\mathbf{Q}''(\tau)$ appearing on the left hand side of (B.6b) with $\mathbf{P}(\tau)$, we get

$$\mathrm{sym}\left(\mathbf{P}(\tau)\right) = \overline{\mathbf{W}}^2(\tau). \tag{B.7}$$

Appendix C. The matrix $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$ is negative semidefinite and its negative eigenvalues, if they exist, have even algebraic multiplicities

The entries of $\overline{\mathbf{W}}(\tau)$ and $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$ are all real numbers. However, in this section we consider $\overline{\mathbf{W}}(\tau)$ and $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$ to be elements of $\mathscr{M}_{\mathrm{n_{sd}},\mathrm{n_{sd}}}(\mathbb{C})$, where $\mathscr{M}_{\mathrm{n_{sd}},\mathrm{n_{sd}}}(\mathbb{C})$ is the space of all $\mathrm{n_{sd}} \times \mathrm{n_{sd}}$ matrices whose entries belong to $\mathbb{C}$, the space of complex numbers.

Appendix C.1. Negative semi-definiteness of $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$

Lemma Appendix C.1. *The matrix $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$ is negative semidefinite.*

Proof. The matrix $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$ is self-adjoint since it is equal to its transpose, which, as all of $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$'s entries are real, is equal to its conjugate-transpose, i.e., to its adjoint. Hence it follows from [45, 7.31] that the matrix $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$ is negative semidefinite iff the inner product $\langle\mathrm{sym}\left(\mathbf{P}(\tau)\right)\mathbf{X}, \mathbf{X}\rangle$, where $\mathbf{X} \in \mathscr{M}_{\mathrm{n_{sd}},1}(\mathbb{C})$ but is otherwise arbitrary, is always non-positive.

It follows from (1.3) that

$$\langle \mathrm{sym}\left(\mathbf{P}(\tau)\right)\mathbf{X}, \mathbf{X}\rangle = \langle \overline{\mathbf{W}}^2(\tau)\mathbf{X}, \mathbf{X}\rangle. \tag{C.1}$$

Since we know from Lemma Appendix B.1 that $\overline{\mathbf{W}}(\tau)$ is skew-symmetric, we can write $\overline{\mathbf{W}}^2(\tau)$ on the right hand side of (C.1) as $-\overline{\mathbf{W}}^\mathsf{T}(\tau)\overline{\mathbf{W}}(\tau)$. On doing so and using the properties of the inner product we get

$$\langle \mathrm{sym}\left(\mathbf{P}(\tau)\right)\mathbf{X}, \mathbf{X}\rangle = -\langle \overline{\mathbf{W}}(\tau)\mathbf{X}, \overline{\mathbf{W}}(\tau)\mathbf{X}\rangle. \tag{C.2}$$

It also follows from the properties of the inner product that $\langle \overline{\mathbf{W}}(\tau)\mathbf{X}, \overline{\mathbf{W}}(\tau)\mathbf{X}\rangle$ is always non-negative. Therefore we get from (C.2) that $\langle \mathrm{sym}\left(\mathbf{P}(\tau)\right)\mathbf{X}, \mathbf{X}\rangle$ is always non-positive, or equivalently that $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$ is negative semidefinite.

Appendix C.2. Form of the eigenvalues of $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$

Lemma Appendix C.2. *The matrix* $\mathrm{sym}\left(\mathbf{P}(\tau)\right)$*'s negative eigenvalues, if they exist, have even algebraic multiplicities.*

Proof. Say $\mathbf{S} \in \mathscr{M}_{\mathrm{n_{sd}, n_{sd}}}(\mathbb{C})$ then $\mathbf{S}$ is said to be normal when it commutes with its conjugate-transpose $\mathbf{S}^\mathsf{H}$, i.e., when $\mathbf{S}\,\mathbf{S}^\mathsf{H} = \mathbf{S}^\mathsf{H}\,\mathbf{S}$. Note that

$$\overline{\mathbf{W}}(\tau)\,\overline{\mathbf{W}}^\mathsf{T}(\tau) = \overline{\mathbf{W}}(\tau)\left(-\overline{\mathbf{W}}(\tau)\right) = \left(-\overline{\mathbf{W}}(\tau)\right)\overline{\mathbf{W}}(\tau) = \overline{\mathbf{W}}^\mathsf{T}(\tau)\overline{\mathbf{W}}(\tau). \tag{C.3}$$

The first and the third equalities in (C.3) follow from the fact that $\overline{\mathbf{W}}(\tau)$ is skew-symmetric (Lemma Appendix B.1). Since all the entries of $\overline{\mathbf{W}}(\tau)$ are real, the transpose of $\overline{\mathbf{W}}(\tau)$ is equal to its conjugate-transpose. For this reason it follows from (C.3) that $\overline{\mathbf{W}}(\tau)\,\overline{\mathbf{W}}^\mathsf{H}(\tau) = \overline{\mathbf{W}}^\mathsf{H}(\tau)\overline{\mathbf{W}}(\tau)$, or equivalently that $\overline{\mathbf{W}}(\tau)$ is normal.

Since $\overline{\mathbf{W}}(\tau)$ is normal it follows from the *Complex Spectral Theorem* [e.g., see 45, 7.24] that there exists an unitary matrix $\mathbf{U}(\tau) \in \mathscr{M}_{\mathrm{n_{sd}, n_{sd}}}(\mathbb{C})$ such that $\mathbf{U}^\mathsf{H}(\tau)\overline{\mathbf{W}}(\tau)\mathbf{U}(\tau) = \mathrm{diag}\left(\mu_i(\tau)\right)_{i \in \mathscr{I}}$, where $\mu_i(\tau) \in \mathbb{C}$ and $\mathrm{diag}\left(\mu_i(\tau)\right)_{i \in \mathscr{I}}$ is a diagonal matrix whose diagonal entries are $\mu_1(\tau), \mu_2(\tau) \dots, \mu_{\mathrm{n_{sd}}}(\tau)$. That is,

$$\mathbf{U}^\mathsf{H}(\tau)\overline{\mathbf{W}}(\tau)\mathbf{U}(\tau) = \begin{pmatrix} \mu_1(\tau) & 0 & \dots & 0 \\ 0 & \mu_2(\tau) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mu_{\mathrm{n_{sd}}}(\tau) \end{pmatrix}. \tag{C.4}$$

The complex numbers $\mu_i(\tau)$, not necessarily distinct, are the eigenvalues of $\overline{\mathbf{W}}(\tau)$ and the columns of $\mathbf{U}(\tau)$ are the eigenvectors of $\overline{\mathbf{W}}(\tau)$. To be more specific,

$$\overline{\mathbf{W}}(\tau)\mathbf{u}_i(\tau) = \mu_i(\tau)\mathbf{u}_i(\tau) \text{ (no sum over } i), \tag{C.5}$$

where $\mathbf{u}_i(\tau) = \left((\mathbf{U}(\tau))_{ji}\right)_{j \in \mathscr{I}}$. Applying the operation of complex-conjugation to both sides of (C.5) and noting that the entries of $\overline{\mathbf{W}}(\tau)$ are all real we get that

$$\overline{\mathbf{W}}(\tau)\mathbf{u}_i^*(\tau) = \mu_i^*(\tau)\,\mathbf{u}_i^*(\tau) \text{ (no sum over } i), \tag{C.6}$$

where $\mu_i^*(\tau)$ and $\mathbf{u}_i^*(\tau)$ are, respectively, the complex-conjugates of $\mu_i(\tau)$ and $\mathbf{u}_i(\tau)$. It follows from (C.6) that if $\mu_i(\tau)$ is an eigenvalue of $\overline{\mathbf{W}}(\tau)$ then so is $\mu_i^*(\tau)$. Thus $(\mu_i(\tau))_{i \in \mathscr{I}}$ has the form $\varsigma\left(z_1(\tau), z_1^*(\tau), z_2(\tau), z_2^*(\tau), \dots, z_k(\tau), z_k^*(\tau), \alpha_1(\tau), \alpha_2(\tau), \dots, \alpha_l(\tau)\right)$, where $\varsigma(\cdot)$ is the permutation operation, $z_i(\tau) \in \mathbb{C}$ with $\mathrm{Im}\left(z_i(\tau)\right) \neq 0$, $z_i^*(\tau)$ is the complex-conjugate of $z_i(\tau)$, $0 \leqslant k \leqslant \lfloor \mathrm{n_{sd}}/2 \rfloor$[6], $\alpha_i(\tau) \in \mathbb{R}$, and $l = \mathrm{n_{sd}} - 2k$. It is not necessary that the complex numbers $z_i(\tau)$ be distinct from one another. The same is the case with the real numbers $\alpha_i(\tau)$.

Taking the square on both sides of (C.4) and using our knowledge about the form of $(\mu_i(\tau))_{i \in \mathscr{I}}$ we

---

[6]Here $\lfloor \cdot \rfloor$ is the *floor* function.

get that

$$\left(\mathbf{U}^{\mathsf{H}}(\tau)\overline{\mathbf{W}}(\tau)\mathbf{U}(\tau) \right)^2 = \operatorname{diag}_\varsigma \left(z_1^{\,2}(\tau), z_1^{*\,2}(\tau), \ldots, z_k^{\,2}(\tau), z_k^{*\,2}(\tau), \alpha_1^2(\tau), \ldots, \alpha_l^2(\tau) \right). \qquad \text{(C.7)}$$

The expression on the left hand side of (C.7) can be simplified as $\left(\mathbf{U}^{\mathsf{H}}(\tau)\overline{\mathbf{W}}(\tau)\mathbf{U}(\tau) \right)^2 =$ $\mathbf{U}^{\mathsf{H}}(\tau)\overline{\mathbf{W}}(\tau)\mathbf{U}(\tau)\,\mathbf{U}^{\mathsf{H}}(\tau)\overline{\mathbf{W}}(\tau)\mathbf{U}(\tau) = \mathbf{U}^{\mathsf{H}}(\tau)\overline{\mathbf{W}}^2(\tau)\mathbf{U}(\tau) = \mathbf{U}^{\mathsf{H}}(\tau)\operatorname{sym}(\mathbf{P}(\tau))\,\mathbf{U}(\tau)$, where the second equality follows from the fact that $\mathbf{U}(\tau)$ is an unitary matrix, and the third equality from (1.3). Thus we can get from (C.7) that

$$\mathbf{U}^{\mathsf{H}}(\tau)\operatorname{sym}(\mathbf{P}(\tau))\,\mathbf{U}(\tau) = \operatorname{diag}_\varsigma \left(z_1^{\,2}(\tau), z_1^{*\,2}(\tau), \ldots, z_k^{\,2}(\tau), z_k^{*\,2}(\tau), \alpha_1^2(\tau), \ldots, \alpha_l^2(\tau) \right), \qquad \text{(C.8a)}$$

which implies that $\mathbf{u}_i(\tau)$ are the eigenvectors of $\operatorname{sym}(\mathbf{P}(\tau))$ as well, but with their corresponding eigenvalues being $\varsigma \left(z_1^{\,2}(\tau), z_1^{*\,2}(\tau), \ldots, z_k^{\,2}(\tau), z_k^{*\,2}(\tau), \alpha_1^2(\tau), \ldots, \alpha_l^2(\tau) \right)$. We know from Lemma Appendix C.1 that all of $\operatorname{sym}(\mathbf{P}(\tau))$'s eigenvalues are non-positive. Therefore, each $z_i(\tau)$ must be of the form $\lambda_i(\tau)\sqrt{-1}$, where $\lambda_i(\tau) \in \mathbb{R}$ and $\lambda_i(\tau) \neq 0$, and $\alpha_i(\tau) = 0$. Hence, we get from (C.8a) that

$$\mathbf{U}^{\mathsf{H}}(\tau)\operatorname{sym}(\mathbf{P}(\tau))\,\mathbf{U}(\tau) = \operatorname{diag}_\varsigma \left(-\lambda_1^2(\tau), -\lambda_1^2(\tau), \ldots, -\lambda_k^2(\tau), -\lambda_k^2(\tau), 0, \ldots, 0 \right), \qquad \text{(C.9)}$$

where, to reiterate, $0 \leqslant k \leqslant \lfloor \mathrm{n_{sd}}/2 \rfloor$ and $\lambda_i(\tau)$, when they exist, are non-zero and not necessarily distinct. It can be noted from this last assertion that all of $\operatorname{sym}(\mathbf{P}(\tau))$'s negative eigenvalues, specifically those corresponding to $\lambda_i(\tau)$, are of even geometric multiplicities. For the case of symmetric matrices, algebraic and geometric multiplicities are one and the same. Therefore, $\operatorname{sym}(\mathbf{P}(\tau))$'s negative eigenvalues, when they exist, are also of even algebraic multiplicities.

Appendix D. Calculating $\overline{\mathbf{W}}(\tau)$ as the square-root of $\operatorname{sym}(\mathbf{P}(\tau))$

Appendix D.1. A spectral decomposition of $\operatorname{sym}(\mathbf{P}(\tau))$

Since $\operatorname{sym}(\mathbf{P}(\tau))$ is a real symmetric matrix it follows from the *Real Spectral Theorem* [45, 7.29] that it can be decomposed as

$$\mathbf{N}(\tau)\,\mathbf{D}(\tau)\,\mathbf{N}^{\mathsf{T}}(\tau), \qquad \text{(D.1)}$$

where $\mathbf{D}(\tau)$ and $\mathbf{N}(\tau)$ belong to $\mathscr{M}_{\mathrm{n_{sd}},\mathrm{n_{sd}}}(\mathbb{R})$. We first describe $\mathbf{N}(\tau)$ and then $\mathbf{D}(\tau)$, in the next paragraph. The matrix $\mathbf{N}(\tau) := (\mathbf{n}_1(\tau), \ldots, \mathbf{n}_{\mathrm{n_{sd}}}(\tau))^{\mathsf{T}}$ where $\mathbf{n}_i(\tau) \in \mathscr{M}_{\mathrm{n_{sd}},1}(\mathbb{R})$ are $\operatorname{sym}(\mathbf{P}(\tau))$'s eigenvectors that are constructed such that $\mathbf{n}_i(\tau) \cdot \mathbf{n}_j(\tau) = \delta_{ij}$, or equivalently

$$\mathbf{N}(\tau)\,\mathbf{N}^{\mathsf{T}}(\tau) = \mathbf{I}. \qquad \text{(D.2)}$$

Using (C.9) it can be shown that for $\mathrm{n_{sd}} = 1$, $\mathbf{D}(\tau) = (0)$; for $\mathrm{n_{sd}} = 2$, $\mathbf{D}(\tau) = \operatorname{diag}(0,0)$ or $\operatorname{diag}\left(-\lambda_1^2(\tau), -\lambda_1^2(\tau) \right)$, where $\lambda_1(\tau) \neq 0$; and for $\mathrm{n_{sd}} = 3$, $\mathbf{D}(\tau) = \operatorname{diag}(0,0,0)$ or $\operatorname{diag}_\varsigma \left(-\lambda_1^2(\tau), -\lambda_1^2(\tau), 0 \right)$. The last two results can be summarized by saying that when $\mathrm{n_{sd}} = 2$,

$$\mathbf{D}(\tau) = \operatorname{diag}\left(-\lambda^2(\tau), -\lambda^2(\tau) \right), \qquad \text{(D.3)}$$

where $\lambda(\tau) \in \mathbb{R}$, and when $\mathrm{n_{sd}} = 3$, $\mathbf{D}(\tau) = \operatorname{diag}_\varsigma(-\lambda(\tau)^2, -\lambda(\tau)^2, 0)$. Without loss of generality, we can choose the order of $\mathbf{n}_i(\tau)$ so that their respective eigenvalues form a non-increasing sequence[7]. Therefore, for concreteness in the case of $\mathrm{n_{sd}} = 3$ we take

$$\mathbf{D}(\tau) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\lambda^2(\tau) & 0 \\ 0 & 0 & -\lambda^2(\tau) \end{pmatrix}. \qquad \text{(D.4)}$$

---

[7]The matrix $\overline{\mathbf{W}}(\tau)$ as the square root of $\operatorname{sym}(\mathbf{P}(\tau))$ does not depend on the order of $\mathbf{n}_i(\tau)$. Different orders will lead to the same $\overline{\mathbf{W}}(\tau)$.

Appendix D.2. Calculation of $\overline{\mathbf{W}}(\tau)$ from sym$\,(\mathbf{P}(\tau))$ using (1.3)

Let

$$\mathbf{F}(\tau) := \mathbf{N}^\mathsf{T}(\tau)\,\overline{\mathbf{W}}(\tau)\,\mathbf{N}(\tau). \tag{D.5}$$

It can be shown using $\mathbf{F}(\tau)$'s definition, equations (D.2), and (1.3), and sym$\,(\mathbf{P}(\tau))$'s decomposition that is derived in Appendix D.1 and summarized in (4.3) that

$$\mathbf{F}^2(\tau) = \mathbf{D}(\tau). \tag{D.6}$$

Substituting $\mathbf{D}(\tau)$ in (D.6) from (D.3) and (D.4) for then noting from Lemma Appendix B.1 and $\mathbf{F}(\tau)$'s definition that $\mathbf{F}(\tau)$ is skew-symmetric, it can be shown that for $n_{sd} = 2$ and 3

$$\mathbf{F}(\tau) = \pm \star (\lambda(\tau)) \tag{D.7}$$

and

$$\mathbf{F}(\tau) = \pm \star ((\lambda(\tau), 0, 0)), \tag{D.8}$$

respectively. Equations 4.7 follow from (D.5), (D.7), and (D.8).

References

[1] CDC, NIH, DoD, and VA Leadership Panel, Report to congress on traumatic brain injury in the united states: Understanding the public health problem among current and former military personnel., Tech. rep., Centers for Disease Control and Prevention (CDC), the National Institutes of Health (NIH), the Department of Defense (DoD), and the Department of Veterans Affairs (VA) (2013).

[2] J. D. Cassidy, L. Carroll, P. Peloso, J. Borg, H. Von Holst, L. Holm, J. Kraus, V. Coronado, Incidence, risk factors and prevention of mild traumatic brain injury: results of the WHO collaborating centre task force on mild traumatic brain injury, Journal of Rehabilitation Medicine 36 (0) (2004) 28–60.

[3] S. Ganpule, N. P. Daphalapurkar, K. T. Ramesh, A. K. Knutsen, D. L. Pham, P. V. Bayly, J. L. Prince, A three-dimensional computational human head model that captures live human brains dynamics, Journal of Neurotrauma 34 (13) (2017) 2154–2166.

[4] A. C. Bain, D. F. Meaney, Tissue-level thresholds for axonal damage in an experimental model of central nervous system white matter injury, Journal of Biomechanical Engineering 122 (6) (2000) 615–622.

[5] D. F. Meaney, B. Morrison, C. D. Bass, The mechanics of traumatic brain injury: a review of what we know and what we need to know for reducing its societal burden, Journal of Biomechanical Engineering 136 (2) (2014) 021008.

[6] S. Kleiven, Why most traumatic brain injuries are not caused by linear acceleration but skull fractures are, Frontiers in Bioengineering and Biotechnology 1 (2013) 15.

[7] E. Bar-Kochba, M. T. Scimone, J. B. Estrada, C. Franck, Strain and rate-dependent neuronal injury in a 3D in vitro compression model of traumatic brain injury, Scientific Reports 6 (2016) 30550.

[8] P. K. Kriz, S. J. Staffa, D. Zurakowski, M. MacAskill, T. Kirchberg, K. Robert, J. Baird, G. Lockhart, Effect of penalty minute rule change on injuries and game disqualification penalties in high school ice hockey, The American Journal of Sports Medicine 47 (2) (2019) 438–443.

[9] R. Mannix, N. J. Morriss, G. M. Conley, W. P. Meehan III, A. Nedder, J. Qiu, J. Float, C. A. DiCesare, G. D. Myer, Internal jugular vein compression collar mitigates histopathological alterations after closed head rotational head impact in swine: A pilot study, Neuroscience 437 (2020) 132–144.

[10] J. H. McElhaney, J. L. Fogle, J. W. Melvin, R. R. Haynes, V. L. Roberts, N. M. Alem, Mechanical properties of cranial bone, Journal of Biomechanics 3 (5) (1970) 495 – 511.

[11] R. Delille, D. Lesueur, P. Potier, P. Drazetic, E. Markiewicz, Experimental study of the bone behaviour of the human skull bone for the development of a physical head model, International Journal of Crashworthiness 12 (2) (2007) 101–108.

[12] J. A. Motherway, P. Verschueren, G. Van der Perre, J. Vander Sloten, M. D. Gilchrist, The mechanical properties of cranial bone: The effect of loading rate and cranial sampling position, Journal of Biomechanics 42 (13) (2009) 2129 – 2135.

[13] S. Budday, R. Nay, R. de Rooij, P. Steinmann, T. Wyrobek, T. C. Ovaert, E. Kuhl, Mechanical properties of gray and white matter brain tissue by indentation, Journal of the Mechanical Behavior of Biomedical Materials 46 (2015) 318 – 330.

[14] R. Willinger, H. Kang, B. Diaw, Three-dimensional human head finite-element model validation against two experimental impacts, Annals of Biomedical Engineering 27 (3) (1999) 403–410.

[15] R. M. Wright, A. Post, B. Hoshizaki, K. Ramesh, A multiscale computational approach to estimating axonal damage under inertial loading of the head, Journal of Neurotrauma 30 2 (2013) 102–18.

[16] C. W. Gadd, Use of a weighted-impulse criterion for estimating injury hazard, Tech. rep., SAE Technical Paper (1966).

[17] C. C. Chou, G. W. Nyquist, Analytical studies of the head injury criterion (HIC), SAE Transactions (1974) 398–410.

[18] J. A. Newman, A generalized acceleration model for brain injury threshold (GAMBIT), in: Proceedings of the 1986 International IRCOBI Conference on the Biomechanics of Impact, 1986.

[19] E. G. Takhounts, M. J. Craig, K. Moorhouse, J. McFadden, V. Hasija, Development of brain injury criteria (BrIC), Tech. rep., SAE Technical Paper (2013).

[20] K. Laksari, M. Fanton, L. C. Wu, T. H. Nguyen, M. Kurt, C. Giordano, E. Kelly, E. O'Keeffe, E. Wallace, C. Doherty, M. Campbell, S. Tiernan, G. Grant, J. Ruan, S. Barbat, D. B. Camarillo, Multi-directional dynamic model for traumatic brain injury detection, Journal of Neurotrauma 37 (7) (2020) 982–993.

[21] L. F. Gabler, J. R. Crandall, M. B. Panzer, Development of a metric for predicting brain strain responses using head kinematics, Annals of Biomedical Engineering 46 (7) (2018) 972–985.

[22] L. F. Gabler, J. R. Crandall, M. B. Panzer, Development of a second-order system for rapid estimation of maximum brain strain, Annals of Biomedical Engineering 47 (2019) 1971–1981.

[23] F. Hernandez, L. C. Wu, M. C. Yip, K. Laksari, A. R. Hoffman, J. R. Lopez, G. A. Grant, S. Kleiven, D. B. Camarillo, Six degree-of-freedom measurements of human mild traumatic brain injury, Annals of Biomedical Engineering 43 (8) (2015) 1918–1934.

[24] D. B. Camarillo, P. B. Shull, J. E. Mattson, R. Shultz, D. Garza, An instrumented mouthguard for measuring linear and angular head impact kinematics in american football, Annals of Biomedical Engineering 41 (2013) 1939–1949.

[25] J. Chu, J. Beckwith, J. Crisco, R. Greenwald, A novel algorithm to measure linear and rotational head acceleration using single-axis accelerometers, Journal of Biomechanics 39 (Supplement 1) (2006) S534.

[26] S. Rowson, G. Brolinson, M. Goforth, D. Dietter, S. Duma, Linear and angular head acceleration measurements in collegiate football, Journal of Biomechanical Engineering 131 (6) (2009) 061016.

[27] M. M. Rahaman, W. Fang, A. L. Fawzi, Y. Wan, H. Kesari, An accelerometer-only algorithm for determining the acceleration field of a rigid body, with application in studying the mechanics of mild traumatic brain injury, Journal of the Mechanics and Physics of Solids 143 (2020) 104014.

[28] W. Fang, Y. Wan, H. Kesari, AO-Desktop-App, https://github.com/AppliedMechanicsLab/AO-Desktop-App.git (2021).

[29] H. Lee, H. Chung, J. Lee, Motion artifact cancellation in wearable photoplethysmography using gyroscope, IEEE Sensors Journal 19 (3) (2019) 1166–1175.

[30] S. J. Ovaska, S. Valiviita, Angular acceleration measurement: A review, in: IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation is Going (Cat. No. 98CH36222), Vol. 2, IEEE, 1998, pp. 875–880.

[31] F. Alonso, J. Castillo, P. Pintado, Application of singular spectrum analysis to the smoothing of raw kinematic signals, Journal of Biomechanics 38 (5) (2005) 1085–1092.

[32] A. J. Padgaonkar, K. W. Krieger, A. I. King, Measurement of angular acceleration of a rigid body using linear accelerometers, Journal of Applied Mechanics 42 (3) (1975) 552–556.

[33] J. Genin, J. Hong, W. Xu, Accelerometer placement for angular velocity determination, Journal of Dynamic Systems, Measurement, and Control 119 (3) (1997) 474–477.

[34] R. Naunheim, P. Bayly, J. Standeven, J. Neubauer, L. Lewis, G. Genin, Linear and angular head accelerations during heading of a soccer ball, Medicine & Science in Sports & Exercise 35 (8) (2003) 1406–1412.

[35] G. Golub, C. Van Loan, Matrix Computations, Johns Hopkins University Press, 2013.

[36] H. Kesari, A. J. Lew, Adhesive frictionless contact between an elastic isotropic half-space and a rigid axi-symmetric punch, Journal of Elasticity 106 (2) (2012) 203–224.

[37] H. Kesari, A. J. Lew, Effective macroscopic adhesive contact behavior induced by small surface roughness, Journal of the Mechanics and Physics of Solids 59 (12) (2011) 2488 – 2510.

[38] M. Dimian, P. Andrei, Noise-driven phenomena in hysteretic systems, Springer, 2014.

[39] E. Bibbona, G. Panfilo, P. Tavella, The Ornstein–Uhlenbeck process as a model of a low pass filtered white noise, Metrologia 45 (6) (2008) S117–S126.

[40] R. W. Carlsen, A. L. Fawzi, Y. Wan, H. Kesari, C. Franck, A quantitative relationship between rotational head kinematics and brain tissue strain from a 2-D parametric finite element analysis, Brain Multiphysics 2 (2021) 100024.

[41] C. Jog, Continuum Mechanics: Volume 1: Foundations and Applications of Mechanics, Cambridge University Press, 2015.

[42] O. M. O'Reilly, Intermediate Dynamics for Engineers: Newton-Euler and Lagrangian Mechanics, Cambridge University Press, 2020.

[43] J. Maruskin, Dynamical Systems and Geometric Mechanics: An Introduction, De Gruyter, 2018.

[44] J. E. Marsden, T. S. Ratiu, Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems, Vol. 17, Springer Science & Business Media, 2013.

[45] S. Axler, Linear algebra done right, Springer, 2015.