A Lie Group Variational Integrator for Rigid Body Dynamics Using Dual Quaternion

Jiafeng Xu[1] and Karl H. Halse[2]
Department of Ocean Operations and Civil Engineering,

Norwegian University of Science and Technology, Aalesund, Norway, 6015

In rigid body dynamic simulations, often the algorithm is required to deal with general situations where both reference point and inertia matrix are arbitrarily defined. We introduce a novel Lie group variational integrator using dual quaternion for simulating rigid body dynamics in all six degrees of freedom. Dual quaternion is used to represent rigid body kinematics and one-step Lie group method is used to derive dynamic equations. The combination of these two becomes the first Lie group variational integrator for rigid body dynamics without decoupling translations and rotations. Newton-Raphson method is used to solve the recursive dynamic equation. Our method is suitable for rigid body simulations with high accuracy under large time step. The entire dynamics-control system can now be established based on dual quaternions without re-projection from Cartesian frame. A numerical example of spacecraft-cargo separation process proves that our method respects the symplectic structure of the system with excellent long-term conservation of geometry structure, momentum and energy. In instants where separation happens, reference point transformation is no longer required.

[1] PhD candidate, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Aalesund, Norway

[2] Associate Professor, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Aalesund, Norway

Nomenclature

$\square$	= blank filler
$\overline{\square}$	= quaternions
$\hat{\square}$	= pure quaternions
$\circ$	= quaternion multiplication
$\overline{\square}^{\dagger}$	= conjugate of a quaternion
$\tilde{\square}$	= dual quaternions
$\check{\square}$	= pure dual quaternions
$\tilde{\square}_a$	= first quaternion of a dual quaternion
$\tilde{\square}_b$	= second quaternion of a dual quaternion
$\otimes$	= dual quaternion multiplication
$\tilde{\square}^{\dagger}$	= quaternion conjugate of a dual quaternion
$\tilde{\square}^{*}$	= dual transpose of a dual quaternion
t	= time, s^{-1}
h	= time step, s
k	= time index
δ	= variational derivative
$\square_d$	= discrete value mark
$\boldsymbol{\eta}$	= arbitrary perturbation
$\boldsymbol{e}_W$	= world-fixed coordinate frame
$\boldsymbol{e}_B$	= body-fixed coordinate frame
$\boldsymbol{v}$	= translational velocity, $m \cdot s^{-1}$
$\boldsymbol{\omega}$	= angular velocity, $rad \cdot s^{-1}$
$\boldsymbol{\chi}$	= 6-by-1 angular and translational velocity
$\boldsymbol{J}$	= moment of inertia matrix,$kg \cdot m^2$
$\boldsymbol{M}$	= general inertia matrix
$\boldsymbol{\zeta}, \boldsymbol{P}$	= angular and translational momentum
$\boldsymbol{\tau}$	= angular and translational forces in Cartesian frame

$\boldsymbol{f}$ = incremental pose

$\boldsymbol{\Phi}, \boldsymbol{\Psi}$ = parameters for dual quaternion

$\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}$ = right hand side of recursive dynamic equations

$\boldsymbol{\alpha}, \boldsymbol{\beta}$ = left hand side of recursive dynamic equations

$\boldsymbol{\mathcal{J}}$ = jacobian matrix

I. Introduction

In rigid body dynamic simulations, often the algorithm is required to deal with coupled translations and rotations, meaning the reference point may not be on the center of mass. For example, in space missions, the center of mass of a spacecraft will change when a cargo is released, while the reference point of spacecraft motion is preferred to be constant. This paper introduces a novel Lie group variational integrator based on dual quaternion for handling this situation in a both symplectic and generic manner.

Different from traditional differential equations, variational integrators first temporally discretize a Hamiltonian system to its symplectic form, then solve the recursive discrete equations. Lie group variational integrator is a more efficient method that makes the discrete system evolve automatically on Lie groups without the use of re-projection, constraints, differential equations and local coordinates[1]. A generic mathematical foundation of Lie group variational integrators for geometric mechanics have been developed based on Euler-Poincare and Lie-Poisson equations[2–5]. However in the practical subject of using Lie group variational integrator for single rigid body dynamics, 3-by-3 rotation matrix is still widely used[6, 7], the translation and rotation of the rigid body are also decoupled that only the rotation group $SO(3)$ and its Lie algebra $\mathfrak{so}(3)$ are effectively used[8, 9], meaning the translation is still evolving on linear space $\mathbb{R}^3$[6].

On the other hand, quaternions are commonly used in both rigid body kinematics and dynamics for representing $SO(3)$ due to its clarity and compactness. Dual quaternions are an extension of the quaternion concept that includes translations into the formulation. The advantages dual quaternions have over many other formulations can be summarized as 1) Singularity-free; 2) Un-ambiguous; 3) Shortest path interpolation; 4) Unified representation of translation and rotation in a single invariant coordinate frame[10]; 5) Computationally efficient; 6) Intuitive connection of its exponential map to the screw motion. Many papers have used dual quaternions in kinematic representation for motion tracking and control of rigid body[11–14]. However, only a few papers have discussed the application of dual quaternion in rigid body dynamics and most of them used dual quaternion frame for a modified Newton-Euler or Lagrangian differential formulation[15, 16].

This paper is inspired by Manchester and Peck's work[17], who for the first time used quaternions to directly represent rotation in Lie group variational integrator. Having noticed that the *Special Euclidean Group* $SE(3)$ for spatial displacement of rigid bodies(referred as *pose*[18] in this paper) is a Lie Group by itself, we introduce the Dual Quaternion Variational Integrator(DQVI), a combination of Lie group variational integrator and the unit dual quaternion group, which is a 2-to-1 homomorphism to $SE(3)$. To the best of our knowledge, this is the first practice of using dual quaternion in Lie group variational integrator for rigid body dynamics without decoupling the translation and rotation. The primary improvement of DQVI over other algorithms is the application of dual quaternion state variables that allows a fully coupled dynamic simulation in $SE(3)$. Both the reference point and the full 6-by-6 inertia matrix can be arbitrarily defined. The entire dynamics-control loop can now be established in dual quaternion frame directly without re-projection from Cartesian frame. DQVI is particularly suitable for real-time motion dynamics, estimation, and control applications with high-order accuracy under large time step.

Section II briefly introduce the background of quaternions and dual quaternions. Section III formulates the discretized dynamic equations for single rigid body, which will be used in the variational integrator. Section IV focuses on solving the dynamic equations using Newton-Raphson method. A numerical example of spacecraft-cargo separation process is given in Section V in order to demonstrate the performance of DQVI. Section VI is the conclusion.

II. Background

This section reviews the background of the algebraic and kinematic properties of quaternions and dual quaternions in the context of Lie group and screw motion theory. More detailed introduction can be found in [19, 20].

A. Quaternions

Quaternions form a four-dimensional vector space $\mathbb{H}$ over the real numbers with one real element and three imaginary elements.

$$\overline{\boldsymbol{q}} = \begin{bmatrix} q_w \\ \boldsymbol{q}_{xyz} \end{bmatrix}; \quad \boldsymbol{q}_{xyz} = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}$$

Based on its definition, the quaternions in their vector form have the following unique operators

- Multiplication(noncommutative)

$$\overline{\boldsymbol{q}}_1 \circ \overline{\boldsymbol{q}}_2 = \begin{bmatrix} q_{1w}q_{2w} - \boldsymbol{q}_{1n} \cdot \boldsymbol{q}_{2xyz} \\ q_{1w}\boldsymbol{q}_{2xyz} + q_{2w}\boldsymbol{q}_{1xyz} + \boldsymbol{q}_{1xyz} \times \boldsymbol{q}_{2xyz} \end{bmatrix}$$

- Conjugate

$$\overline{\boldsymbol{q}}^{\dagger} = \begin{bmatrix} q_w \\ -\boldsymbol{q}_{xyz} \end{bmatrix}$$

Remark 1. In general, the notation system for quaternions and dual quaternions has not been standardized yet, readers may find discrepancies in this paper from other works.

There is a useful algebraic property about the quaternion multiplication and the vector dot product

$$(\overline{\boldsymbol{q}}_1 \circ \overline{\boldsymbol{q}}_2) \cdot \overline{\boldsymbol{q}}_3 = (\overline{\boldsymbol{q}}_1^{\dagger} \circ \overline{\boldsymbol{q}}_3) \cdot \overline{\boldsymbol{q}}_2 = (\overline{\boldsymbol{q}}_3 \circ \overline{\boldsymbol{q}}_2^{\dagger}) \cdot \overline{\boldsymbol{q}}_1 \tag{1}$$

The *Unit Quaternions* $\mathbb{H}_u \subset \mathbb{H}$ are quaternions with unit length. $\mathbb{H}_u$ is also an non-abelian Lie group homomorphic to $SO(3)$ under the quaternion multiplication $\circ : \mathbb{H}_u \times \mathbb{H}_u \to \mathbb{H}_u$. The inverse is its conjugate $\overline{\boldsymbol{q}}^{\dagger}$ and the identity is

$$\overline{\boldsymbol{I}} = \overline{\boldsymbol{q}} \circ \overline{\boldsymbol{q}}^{\dagger} = \overline{\boldsymbol{q}}^{\dagger} \circ \overline{\boldsymbol{q}} = [1, 0, 0, 0]^T$$

A vector $\boldsymbol{r} \in \mathbb{R}^3$ can be represented as a *Pure Quaternion* whose "real part" equals zero.

$$\hat{\boldsymbol{r}} = [0, \boldsymbol{r}]^T$$

the rotation of $\boldsymbol{r}$ can be achieved by a combination of two quaternion rotations in $SO(3)$.

$$\overline{\boldsymbol{q}} = \begin{bmatrix} \cos\frac{\theta}{2} \\ \boldsymbol{n}\sin\frac{\theta}{2} \end{bmatrix} = exp\left(\hat{\boldsymbol{n}} \cdot \frac{\theta}{2}\right); \quad \hat{\boldsymbol{r}}' = \overline{\boldsymbol{q}} \circ \hat{\boldsymbol{r}} \circ \overline{\boldsymbol{q}}^{\dagger} \tag{2}$$

$\boldsymbol{n}$ is the rotating axis, θ is the angle of rotation. The exponential map indicates the Lie algebra of unit quaternions $exp : \mathfrak{H}_u \to \mathbb{H}_u$ is pure quaternion.

B. Dual Quaternions

The *Dual Quaternions* form a eight-dimensional space $\mathbb{DH}$ comprised of two quaternions for its real and dual part respectively.

$$\tilde{\boldsymbol{p}} = \begin{bmatrix} \overline{\boldsymbol{p}}_a \\ \overline{\boldsymbol{p}}_b \end{bmatrix}$$

Based on its definition, the dual quaternions in their vector form also have the following unique operators

- Multiplication(noncommunitative)

$$\tilde{\boldsymbol{p}}_1 \otimes \tilde{\boldsymbol{p}}_2 = \begin{bmatrix} \overline{\boldsymbol{p}}_{1a} \circ \overline{\boldsymbol{p}}_{2a} \\ \overline{\boldsymbol{p}}_{1a} \circ \overline{\boldsymbol{p}}_{2b} + \overline{\boldsymbol{p}}_{1b} \circ \overline{\boldsymbol{p}}_{2a} \end{bmatrix}$$

- Quaternion Conjugate

$$\tilde{\boldsymbol{p}}^{\dagger} = \begin{bmatrix} \overline{\boldsymbol{p}}_a^{\dagger} \\ \overline{\boldsymbol{p}}_b^{\dagger} \end{bmatrix}$$

- Dual Transpose

$$\tilde{\boldsymbol{p}}^{*} = \begin{bmatrix} \overline{\boldsymbol{p}}_b \\ \overline{\boldsymbol{p}}_a \end{bmatrix}$$

Remark 2. Here we originally propose the notion of *Dual Transpose*, which shall be distinguished from conventional vector transpose.

Based on Equation 1, there is also an algebraic property, which will be used multiple times in this paper, about the dual quaternion multiplication and the vector dot product involving dual transpose.

$$(\tilde{\boldsymbol{p}}_1 \otimes \tilde{\boldsymbol{p}}_2) \cdot \tilde{\boldsymbol{p}}_3 = (\tilde{\boldsymbol{p}}_1^{\dagger} \otimes \tilde{\boldsymbol{p}}_3^{*}) \cdot \tilde{\boldsymbol{p}}_2^{*} = (\tilde{\boldsymbol{p}}_3^{*} \otimes \tilde{\boldsymbol{p}}_2^{\dagger}) \cdot \tilde{\boldsymbol{p}}_1^{*} \tag{3}$$

The *Pure Dual Quaternions* are dual quaternions composed by two pure quaternions. A vector $[\boldsymbol{r}_a, \boldsymbol{r}_b]^T \in \mathbb{R}^6$ can be expressed in pure dual quaternion form

$$\check{\boldsymbol{r}} = \begin{bmatrix} \hat{\boldsymbol{r}}_a \\ \\ \hat{\boldsymbol{r}}_b \end{bmatrix}$$

The *Unit Dual Quaternions* $\mathbb{DH}_u \subset \mathbb{DH}$ form a Lie group[20] with identity

$$\tilde{\boldsymbol{I}} = \tilde{\boldsymbol{p}} \otimes \tilde{\boldsymbol{p}}^\dagger = \tilde{\boldsymbol{p}}^\dagger \otimes \tilde{\boldsymbol{p}} = [1, 0, 0, 0, 0, 0, 0, 0]^T$$

$\mathbb{DH}_u$ can be used to represent poses in $SE(3)$, which is the semidirect product of the translation group and the rotation group $SE(3) = \mathbb{R}^3 \rtimes SO(3)$. If the translation displacement is denoted as $\boldsymbol{l} \in \mathbb{R}^3$ and the rotation is represented by unit quaternion $\varphi(\overline{\boldsymbol{q}}) = SO(3)$, the unit dual quaternion representation of a pose $\psi(\tilde{\boldsymbol{p}}) = SE(3)$ can be formed as

$$\tilde{\boldsymbol{p}} = \begin{bmatrix} \overline{\boldsymbol{q}} \\ \\ \frac{1}{2}\hat{\boldsymbol{l}} \circ \overline{\boldsymbol{q}} \end{bmatrix} = exp\left(\check{\boldsymbol{s}} \cdot \frac{\Theta}{2}\right) \tag{4}$$

Similar to the exponential map of $\mathbb{H}_u$, the exponential map $exp : \mathfrak{D}\mathfrak{H}_u \to \mathbb{DH}_u$, where $\mathfrak{D}\mathfrak{H}_u$ is the Lie algebra of $\mathbb{DH}_u$, employs the dual angle $\Theta = [\Theta_a, \Theta_b]^T$ and pure dual quaternions $\check{\boldsymbol{s}} = [\hat{\boldsymbol{s}}_a, \hat{\boldsymbol{s}}_b]^T$. The product $\check{\boldsymbol{s}} \cdot \Theta = [\Theta_a \hat{\boldsymbol{s}}_a, \Theta_a \hat{\boldsymbol{s}}_b, \Theta_b \hat{\boldsymbol{s}}_a]^T$. The geometric interpretation of those four quantities is related to *screw motion*, i.e. a rotation and a translation about the same axis. According to Chasle's theorm[21], any rigid transformation can be described by a screw motion. θ_a is the angle of rotation, unit vector $\boldsymbol{s}_a$ is the direction of the axis of rotation, θ_b is the magnitude of translation along the axis and $\boldsymbol{s}_b$ is the *moment of the axis*, which is given by equation $\boldsymbol{s}_b = \boldsymbol{c} \times \boldsymbol{s}_a$. $\boldsymbol{c}$ is the position vector of a point that the axis passes, the exact position of the point is irrelevant since $\boldsymbol{s}_b$ will remain constant. The resemblance in the exponential maps of $\mathbb{H}_u$ and $\mathbb{DH}_u$ reveals that quaternions are only a special case of dual quaternions where the rotation axes pass through the origin, while dual quaternions can represent rotations with arbitrary axes[22].

A pose $\tilde{\boldsymbol{p}}$ can also be used to transform a vector $\boldsymbol{r} \in \mathbb{R}^3$ in $SE(3)$. Construct a pose with identity of $\mathbb{H}_u$

$$\tilde{\boldsymbol{r}} = \begin{bmatrix} \overline{\boldsymbol{I}} \\ \\ \frac{1}{2}\hat{\boldsymbol{r}} \end{bmatrix}$$

the new vector $\boldsymbol{r}'$ after transformation is

$$\tilde{\boldsymbol{r}}' = \tilde{\boldsymbol{p}} \otimes \tilde{\boldsymbol{r}}; \quad \hat{\boldsymbol{r}}' = 2\overline{\boldsymbol{r}}'_b \circ \overline{\boldsymbol{r}}'^{\dagger}_a \tag{5}$$

The transformation also applies to nested poses that are commonly used in multibody kinematics.

$$\tilde{\boldsymbol{r}}' = \tilde{\boldsymbol{p}}_1 \otimes \tilde{\boldsymbol{p}}_2 \otimes \tilde{\boldsymbol{r}} \tag{6}$$

Finally, the relationship of all groups can be represented as in Figure 1

$$\begin{tikzcd}
\mathfrak{H}_u \arrow[r, "d\varphi"] \arrow[d, "exp"] & \mathfrak{so}(3) \arrow[d, "exp"] \\
\mathbb{H}_u \arrow[r, "\varphi"] & SO(3) \arrow[d, "\mathbb{R}^3 \rtimes"] \\
\mathbb{DH}_u \arrow[r, "\psi"] & SE(3) \\
\mathfrak{D}\mathfrak{H}_u \arrow[r, "d\psi"] \arrow[u, "exp"] & \mathfrak{se}(3) \arrow[u, "exp"]
\end{tikzcd}$$

Fig. 1: Relationship between groups

C. Translational and Angular Velocity

The translational and angular velocities $\boldsymbol{v}_B, \boldsymbol{\omega}_B$ in $\boldsymbol{e}_B$ and $\boldsymbol{v}_W, \boldsymbol{\omega}_W$ in $\boldsymbol{e}_W$ are essential to the rigid body simulation. Construct a 6-by-1 vector $\boldsymbol{\chi}$ with $\boldsymbol{v}$ and $\boldsymbol{\omega}$

$$\boldsymbol{\chi}_B = \begin{bmatrix} \boldsymbol{\omega}_B \\ \boldsymbol{v}_B \end{bmatrix}; \quad \boldsymbol{\chi}_W = \begin{bmatrix} \boldsymbol{\omega}_W \\ \boldsymbol{v}_W \end{bmatrix} \tag{7}$$

Its pure dual quaternion form has a key relation with the pose[19]

$$\check{\boldsymbol{\chi}}_B = 2\tilde{\boldsymbol{p}}^{\dagger} \otimes \dot{\tilde{\boldsymbol{p}}}; \quad \check{\boldsymbol{\chi}}_W = 2\dot{\tilde{\boldsymbol{p}}} \otimes \tilde{\boldsymbol{p}} \tag{8}$$

III. Dynamic Equations for Single Rigid Body

This section formulates the recursive discrete-time equation for rigid body dynamics in dual quaternion coordinate frame. The classical variational approach using homogeneous transformation matrix can be found in [23].

A. Continuous Formulation

The equations for a single rigid body motion satisfies the integral Lagrange d'Alembert's principle

$$\delta \int_{t_0}^{t_1} L\left(\tilde{\boldsymbol{p}}, \dot{\tilde{\boldsymbol{p}}}, t\right) dt + \int_{t_0}^{t_1} \tilde{\boldsymbol{F}}\left(\tilde{\boldsymbol{p}}, \dot{\tilde{\boldsymbol{p}}}, t\right) \cdot \delta \tilde{\boldsymbol{p}} dt = 0 \tag{9}$$

where L is the Lagrangian of the system described by a pose and its derivative. $\tilde{\boldsymbol{F}} \in \mathbb{R}^8$ is the generalized non-conservative force also expressed as dual quaternion. $\delta \tilde{\boldsymbol{p}}$ is the infinitesimal variation of pose that vanishes at the end point but otherwise arbitrary. This obviously leads to the forced Euler-Lagrange equations

$$\frac{\partial L}{\partial \tilde{\boldsymbol{p}}} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\tilde{\boldsymbol{p}}}} + \tilde{\boldsymbol{F}} = 0 \tag{10}$$

The kinetic energy T of a single rigid body is

$$T = \frac{1}{2}\int_m \left(\boldsymbol{v}_B + \boldsymbol{\omega}_B \times \boldsymbol{\rho}\right)^2 dm = \frac{1}{2}\boldsymbol{\chi}_B \cdot \boldsymbol{M}_{6\times 6} \cdot \boldsymbol{\chi}_B \tag{11}$$

where $\boldsymbol{\rho}$ is the position vector of a mass point to the reference point in $\boldsymbol{e}_B$ and

$$\boldsymbol{M}_{6\times 6} = \begin{bmatrix} \boldsymbol{J}_{3\times 3} & m \cdot S(\boldsymbol{r}_g) \\ -m \cdot S(\boldsymbol{r}_g) & \boldsymbol{m}_{3\times 3} \end{bmatrix} \tag{12}$$

$\boldsymbol{J}_{3\times 3}$ is the moment of inertia tensor with respect to the reference point. m is the mass of the rigid body and $\boldsymbol{m}_{3\times 3}$ is the diagonal mass matrix. $\boldsymbol{r}_g$ is the position vector of center of mass of the rigid body in body-fixed frame and $S(\cdot)$ is the skew-symmetric matrix of cross product operator. In the context of unit dual quaternion pose, Equation 12 can be extended into an 8 × 8 matrix with extra rows and columns filled with zeros.

$$\boldsymbol{M}_{8\times 8} = \begin{bmatrix} 0 & \cdots & 0 & \cdots \\ \vdots & \boldsymbol{J}_{3\times 3} & \vdots & m \cdot S(\boldsymbol{r}_g) \\ 0 & \cdots & 0 & \cdots \\ \vdots & -m \cdot S(\boldsymbol{r}_g) & \vdots & \boldsymbol{m}_{3\times 3} \end{bmatrix} \tag{13}$$

Use Equation 8, Equation 12 can be written with unit dual quaternions

$$T = \frac{1}{2}\check{\boldsymbol{\chi}}_B^T \cdot \boldsymbol{M}_{8\times 8} \cdot \check{\boldsymbol{\chi}}_B = 2\left(\tilde{\boldsymbol{p}} \otimes \dot{\tilde{\boldsymbol{p}}}\right)^T \cdot \boldsymbol{M}_{8\times 8} \cdot \left(\tilde{\boldsymbol{p}} \otimes \dot{\tilde{\boldsymbol{p}}}\right) \tag{14}$$

Remark 3. The major improvement here for kinetic energy formulation, compared with previous works[6, 9], is that the reference point and can now be arbitrarily defined without being situated at the center of mass. $\boldsymbol{M}_{6\times6}$ can also be arbitrarily defined as long as the determinant is not zero. This is especially important for hydrodynamic related problems involving added mass. Other formulations based on 3-by-3 rotation matrix on $SO(3)$ used a modified moment of inertia tensor $\boldsymbol{J}'_{3\times3} = det(\boldsymbol{J}_{3\times3}) \cdot \boldsymbol{I}_{3\times3} - \boldsymbol{J}_{3\times3}$, while our method keeps the original $\boldsymbol{J}_{3\times3}$ and the fillers for the extra rows and columns can actually be random values since $\tilde{\boldsymbol{\chi}}_B \in \mathbb{DH}_u$.

Since the reference point is not necessarily on the center of mass, the potential energy of the single rigid body is generally a function of both translation and rotation of the rigid body $U = U(\tilde{\boldsymbol{p}})$. The Lagrangian of the system is

$$L = T - U = 2\left(\tilde{\boldsymbol{p}}^{\dagger} \otimes \dot{\tilde{\boldsymbol{p}}}\right)^{T} \cdot \boldsymbol{M}_{8\times8} \cdot \left(\tilde{\boldsymbol{p}}^{\dagger} \otimes \dot{\tilde{\boldsymbol{p}}}\right) - U\left(\tilde{\boldsymbol{p}}\right) \tag{15}$$

B. Discrete Formulation

Equation 9 can be discretized into

$$\delta \sum_{k=0}^{N-1} L_{dk} + \sum_{k=0}^{N-1} W_{dk} = 0 \tag{16}$$

where L_{dk} and W_{dk} represent the action integral and virtual work integral respectively.

1. *Kinetic Energy Integral Variation*

Equation 15 shows that the single rigid body system is time invariant. $\dot{\tilde{\boldsymbol{p}}}_k$ can be approximated by trapezoidal rule with a fixed time step $h = t_{k+1} - t_k$.

$$\dot{\tilde{\boldsymbol{p}}}_k = \left(\tilde{\boldsymbol{p}}_{k+1} - \tilde{\boldsymbol{p}}_k\right)/h \tag{17}$$

Higher order quadrature rules can also be applied, but in real-time simulations, usually there is an “Update" function that overwrites data from last time frame. One-step quadrature rules work better for its simplicity and low cost. The integral of the kinetic energy can be represented as

$$T_{dk}\left(\tilde{\boldsymbol{p}}_k, \dot{\tilde{\boldsymbol{p}}}_k\right) \approx \frac{2}{h}\left(\tilde{\boldsymbol{p}}_k^{\dagger} \otimes \left(\tilde{\boldsymbol{p}}_{k+1} - \tilde{\boldsymbol{p}}_k\right)\right)^{T} \cdot \boldsymbol{M}_{8\times8} \cdot \left(\tilde{\boldsymbol{p}}_k^{\dagger} \otimes \left(\tilde{\boldsymbol{p}}_{k+1} - \tilde{\boldsymbol{p}}_k\right)\right) \tag{18}$$

Simplify the expression by using the fact that $\tilde{\boldsymbol{p}}_k^{\dagger} \otimes \tilde{\boldsymbol{p}}_k = \tilde{\boldsymbol{I}}$ and $\boldsymbol{M}_{8\times8} \cdot \tilde{\boldsymbol{I}} = (\tilde{\boldsymbol{I}} \cdot \boldsymbol{M}_{8\times8})^T = \tilde{\boldsymbol{0}}$

$$T_{dk} = \frac{2}{h}\left(\tilde{\boldsymbol{f}}_k^{T} \cdot \boldsymbol{M}_{8\times8} \cdot \tilde{\boldsymbol{f}}_k\right) \tag{19}$$

where $\tilde{\boldsymbol{f}}_k = \tilde{\boldsymbol{p}}_k^\dagger \otimes \tilde{\boldsymbol{p}}_{k+1}$, so that $\tilde{\boldsymbol{p}}_{k+1} = \tilde{\boldsymbol{p}}_k \otimes \tilde{\boldsymbol{f}}_k$. By requiring that $\tilde{\boldsymbol{f}}_k \in \mathbb{DH}_u$, $\tilde{\boldsymbol{p}}_k$ is ensured to be evolving in $\mathbb{DH}_u$ as well. The variation of $\tilde{\boldsymbol{f}}_k$ is

$$\delta\tilde{\boldsymbol{f}}_k = \delta\tilde{\boldsymbol{p}}_k^\dagger \otimes \tilde{\boldsymbol{p}}_{k+1} + \tilde{\boldsymbol{p}}_k^\dagger \otimes \delta\tilde{\boldsymbol{p}}_{k+1} = \tilde{\boldsymbol{f}}_k \otimes \check{\boldsymbol{\eta}}_{k+1} - \check{\boldsymbol{\eta}}_k \otimes \tilde{\boldsymbol{f}}_k \tag{20}$$

where $\check{\boldsymbol{\eta}}_k \in \mathfrak{DH}_u$. So the variation of action integral can be written as

$$\delta \sum_{k=0}^{N-1} T_{dk} = \sum_{k=0}^{N-1} \frac{4}{h}\tilde{\boldsymbol{\zeta}}_k \cdot \left(\tilde{\boldsymbol{f}}_k \otimes \check{\boldsymbol{\eta}}_{k+1} - \check{\boldsymbol{\eta}}_k \otimes \tilde{\boldsymbol{f}}_k\right) \tag{21}$$

where

$$\tilde{\boldsymbol{\zeta}}_k = \tilde{\boldsymbol{f}}_k^T \cdot \boldsymbol{M}_{8\times 8}$$

Due to the structure of $\boldsymbol{M}_{8\times 8}$, $\tilde{\boldsymbol{\zeta}}_k$ is guaranteed to be pure dual quaternion $\check{\boldsymbol{\zeta}}_k$. With some index manipulation

$$\begin{aligned}\delta \sum_{k=0}^{N-1} T_{dk} =& \frac{4}{h}\check{\boldsymbol{\zeta}}_{N-1} \cdot \left(\tilde{\boldsymbol{f}}_{N-1} \otimes \check{\boldsymbol{\eta}}_N\right) - \frac{4}{h}\check{\zeta}_0 \cdot \left(\check{\boldsymbol{\eta}}_0 \otimes \tilde{\boldsymbol{f}}_0\right) \\ &+ \sum_{k=1}^{N-1} \frac{4}{h}\check{\boldsymbol{\zeta}}_{k-1} \cdot \tilde{\boldsymbol{f}}_{k-1} \otimes \check{\boldsymbol{\eta}}_k - \check{\boldsymbol{\zeta}}_k \cdot \check{\boldsymbol{\eta}}_k \otimes \tilde{\boldsymbol{f}}_k\end{aligned} \tag{22}$$

The variations are computed with the boundary points $\tilde{\boldsymbol{p}}_0$ and $\tilde{\boldsymbol{p}}_N$ held fixed, meaning $\check{\boldsymbol{\eta}}_k$ vanishes at $k = 0$ and $k = N$

$$\check{\boldsymbol{\eta}}_0 = \check{\boldsymbol{\eta}}_N = \check{\boldsymbol{0}} \tag{23}$$

Also use Equation 3, Equation 22 is then modified as

$$\delta \sum_{k=0}^{N-1} T_{dk} = \sum_{k=1}^{N-1} \frac{4}{h}\left(\tilde{\boldsymbol{f}}_{k-1}^\dagger \otimes \check{\boldsymbol{\zeta}}_{k-1}^* - \check{\boldsymbol{\zeta}}_k^* \otimes \tilde{\boldsymbol{f}}_k^\dagger\right) \cdot \check{\boldsymbol{\eta}}_k^* \tag{24}$$

2. Potential Energy Integral Variation

The potential energy is only a function of $\tilde{\boldsymbol{p}}$, so the potential energy integral variation can be represented as

$$\delta \sum_{k=0}^{N-1} U_{dk} \approx \sum_{k=0}^{N-1} \frac{h}{2}\frac{\partial U(\tilde{\boldsymbol{p}}_k)}{\partial\tilde{\boldsymbol{p}}_k} \cdot \delta\tilde{\boldsymbol{p}}_k + \sum_{k=0}^{N-1} \frac{h}{2}\frac{\partial U(\tilde{\boldsymbol{p}}_{k+1})}{\partial\tilde{\boldsymbol{p}}_{k+1}} \cdot \delta\tilde{\boldsymbol{p}}_{k+1} \tag{25}$$

After index manipulation and also using Equation 3 & 23 , Equation 25 is simplified as

$$\delta \sum_{k=0}^{N-1} U_{dk} = \sum_{k=1}^{N-1} h\left(\tilde{\boldsymbol{p}}_k^\dagger \otimes \left(\frac{\partial U(\tilde{\boldsymbol{p}}_k)}{\partial\tilde{\boldsymbol{p}}_k}\right)^*\right) \cdot \check{\boldsymbol{\eta}}_k^* \tag{26}$$

3. Virtual Work Integral Variation

The virtual work integral is approximated as

$$\sum_{k=0}^{N-1} W_d \approx \sum_{k=0}^{N-1} \frac{h}{2}\left(\tilde{\boldsymbol{F}}_k \delta\tilde{\boldsymbol{p}}_k + \tilde{\boldsymbol{F}}_{k+1}\delta\tilde{\boldsymbol{p}}_{k+1}\right) \tag{27}$$

After index manipulation and also using Equation 3 & 23, Equation 27 is simplified as

$$\sum_{k=0}^{N-1} W_{dk} = \sum_{k=1}^{N-1} h\left(\tilde{\boldsymbol{F}}_k \cdot \left(\tilde{\boldsymbol{p}}_k \otimes \check{\boldsymbol{\eta}}_k\right)\right) = \sum_{k=1}^{N-1} h\left(\tilde{\boldsymbol{p}}_k^{\dagger} \otimes \tilde{\boldsymbol{F}}_k^{*}\right) \cdot \check{\boldsymbol{\eta}}_k^{*} \tag{28}$$

Alternatively, the generalized force in dual quaternion space can be transformed from classical torque and force in Cartesian frame by using the identity of work. Let $\boldsymbol{\tau}_B$ and $\boldsymbol{\tau}_W$ denote torque and force of all 6-DOF in $\boldsymbol{e}_B$ and $\boldsymbol{e}_W$ respectively.

$$W = \boldsymbol{\tau}_B \cdot \boldsymbol{\chi}_B = \boldsymbol{\tau}_W \cdot \boldsymbol{\chi}_W = \tilde{\boldsymbol{F}} \cdot \dot{\tilde{\boldsymbol{p}}} \tag{29}$$

By using Equation 3 and 8

$$\left(\tilde{\boldsymbol{p}}^{\dagger} \otimes \tilde{\boldsymbol{F}}^{*}\right) \cdot \check{\boldsymbol{\chi}}_B^{*} = 2\check{\boldsymbol{\tau}}_B^{*} \cdot \check{\boldsymbol{\chi}}_B^{*}; \quad \left(\tilde{\boldsymbol{F}}^{*} \otimes \tilde{\boldsymbol{p}}\right) \cdot \check{\boldsymbol{\chi}}_W^{*} = 2\check{\boldsymbol{\tau}}_W^{*} \cdot \check{\boldsymbol{\chi}}_W^{*} \tag{30}$$

Since $\check{\boldsymbol{\chi}}_B$ and $\check{\boldsymbol{\chi}}_W$ are both arbitrary pure dual quaternions, the 2nd-4th and 6th-8th equations of

$$\tilde{\boldsymbol{F}}^{*} = 2\tilde{\boldsymbol{p}} \otimes \check{\boldsymbol{\tau}}_B^{*} = 2\check{\boldsymbol{\tau}}_W^{*} \otimes \tilde{\boldsymbol{p}}^{\dagger} \tag{31}$$

are valid. Also, because $\check{\boldsymbol{\eta}}_k$ is a pure dual quaternion, Equation 27 can be simplified as

$$\sum_{k=0}^{N-1} W_{dk} = \sum_{k=1}^{N-1} 2h\check{\boldsymbol{\tau}}_{Bk\_external}^{*} \cdot \check{\boldsymbol{\eta}}_k^{*} \tag{32}$$

Notice that Equation 26 can also be simplified by using body-fixed conservative torque and force in Cartesian frame. The the 2nd-4th and 6th-8th equations of

$$\check{\boldsymbol{\tau}}_{Bk\_conservative}^{*} = -\frac{1}{2}\tilde{\boldsymbol{p}}_k^{\dagger} \otimes \left(\frac{\partial U(\tilde{\boldsymbol{p}}_k)}{\partial \tilde{\boldsymbol{p}}_k}\right)^{*} \tag{33}$$

are valid. Hence

$$\delta \sum_{k=0}^{N-1} U_{dk} = \sum_{k=1}^{N-1} 2h\check{\boldsymbol{\tau}}_{Bk\_conservative}^{*} \cdot \check{\boldsymbol{\eta}}_k^{*} \tag{34}$$

Return Equation 24, 32 and 34 back to Equation 16

$$\sum_{k=1}^{N-1} \left[\frac{4}{h}\left(\tilde{\boldsymbol{f}}_{k-1}^{\dagger} \otimes \check{\boldsymbol{\zeta}}_{k-1}^{*} - \check{\boldsymbol{\zeta}}_k^{*} \otimes \tilde{\boldsymbol{f}}_k^{\dagger}\right) + 2h\left(\check{\boldsymbol{\tau}}_{Bk\_conservative}^{*} + \check{\boldsymbol{\tau}}_{Bk\_external}^{*}\right)\right] \cdot \check{\boldsymbol{\eta}}_k^{*} = 0 \tag{35}$$

Since $\check{\boldsymbol{\eta}}_k$ are arbitrary perturbations in every time frame, Equation 35 must hold

$$\tilde{\boldsymbol{f}}_{k-1}^{\dagger} \otimes \check{\boldsymbol{\zeta}}_{k-1}^{*} - \check{\boldsymbol{\zeta}}_{k}^{*} \otimes \tilde{\boldsymbol{f}}_{k}^{\dagger} + \frac{h^2}{2}\left(\check{\boldsymbol{\tau}}_{Bk\_conservative}^{*} + \check{\boldsymbol{\tau}}_{Bk\_external}^{*}\right) = \begin{bmatrix} \lambda_1 \\ \mathbf{0} \\ \lambda_2 \\ \mathbf{0} \end{bmatrix} \tag{36}$$

λ_1 and λ_2 are unknown values, meaning Equation 36 is only useful with its 2nd-4th and 6th-8th equations. Equation 36 forms a recursive map for getting $\tilde{\boldsymbol{p}}_{k+1}$ from $\tilde{\boldsymbol{p}}_k$ and $\tilde{\boldsymbol{p}}_{k-1}$. The evolution of the system is defined given the initial values of $\tilde{\boldsymbol{p}}_0$ and $\tilde{\boldsymbol{p}}_1$.

C. Parametrization

Since $\tilde{\boldsymbol{f}}_k \in \mathbb{DH}_u$ is homomorphic to $SE(3)$, it can be parametrized by 6 independent variables. Based on the definition of $\mathbb{DH}_u$ in Equation 4 and an unconstrained vector representation of $\mathbb{H}_u$ [17], $\tilde{\boldsymbol{f}}$ can be expressed as

$$\overline{\boldsymbol{f}}_a = \begin{bmatrix} \sqrt{1-\boldsymbol{\Phi}^2} \\ \boldsymbol{\Phi} \end{bmatrix}, \quad \overline{\boldsymbol{f}}_b = \begin{bmatrix} -\frac{\boldsymbol{\Psi}\cdot\boldsymbol{\Phi}}{\sqrt{1-\boldsymbol{\Phi}^2}} \\ \boldsymbol{\Psi} \end{bmatrix} \tag{37}$$

Other ways for parametrization also exists. Put Equation 37 back to Equation 36 and only treat $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Psi}_k$ as variables

$$\begin{cases} \boldsymbol{\mathcal{A}}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k) - \boldsymbol{\alpha}_k = \mathbf{0} \\ \boldsymbol{\mathcal{B}}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k) - \boldsymbol{\beta}_k = \mathbf{0} \end{cases} \tag{38}$$

where

$$\begin{aligned} \boldsymbol{\mathcal{A}}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k) &= \left(\sqrt{1-\boldsymbol{\Phi}_k^2}\boldsymbol{I} + S(\boldsymbol{\Phi}_k)\right)\cdot(\boldsymbol{M}_{11}\boldsymbol{\Phi}_k + \boldsymbol{M}_{12}\boldsymbol{\Psi}_k) \\ &+ \left(-\frac{\boldsymbol{\Psi}_k\cdot\boldsymbol{\Phi}_k}{\sqrt{1-\boldsymbol{\Phi}_k^2}}\boldsymbol{I} + S(\boldsymbol{\Psi}_k)\right)\cdot(\boldsymbol{M}_{21}\boldsymbol{\Phi}_k + \boldsymbol{M}_{22}\boldsymbol{\Psi}_k) \end{aligned} \tag{39}$$

$$\boldsymbol{\mathcal{B}}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k) = \left(\sqrt{1-\boldsymbol{\Phi}_k^2}\boldsymbol{I} + S(\boldsymbol{\Phi}_k)\right)\cdot(\boldsymbol{M}_{21}\boldsymbol{\Phi}_k + \boldsymbol{M}_{22}\boldsymbol{\Psi}_k) \tag{40}$$

$$\begin{aligned} \boldsymbol{\alpha}_k &= \left(\sqrt{1-\boldsymbol{\Phi}_{k-1}^2}\boldsymbol{I} - S(\boldsymbol{\Phi}_{k-1})\right)\cdot(\boldsymbol{M}_{11}\boldsymbol{\Phi}_{k-1} + \boldsymbol{M}_{12}\boldsymbol{\Psi}_{k-1}) \\ &+ \left(-\frac{\boldsymbol{\Psi}_{k-1}\cdot\boldsymbol{\Phi}_{k-1}}{\sqrt{1-\boldsymbol{\Phi}_{k-1}^2}}\boldsymbol{I} - S(\boldsymbol{\Psi}_{k-1})\right)\cdot(\boldsymbol{M}_{21}\boldsymbol{\Phi}_{k-1} + \boldsymbol{M}_{22}\boldsymbol{\Psi}_{k-1}) \\ &+ \frac{h^2}{2}\left(\boldsymbol{\tau}_{ak\_conservative} + \boldsymbol{\tau}_{ak\_external}\right) \end{aligned} \tag{41}$$

$$\begin{aligned}\boldsymbol{\beta}_k = &\left(\sqrt{1-\boldsymbol{\Phi}_{k-1}^2}\boldsymbol{I} - S(\boldsymbol{\Phi}_{k-1})\right)\cdot\left(\boldsymbol{M}_{21}\boldsymbol{\Phi}_{k-1}+\boldsymbol{M}_{22}\boldsymbol{\Psi}_{k-1}\right)\\ &+\frac{h^2}{2}\left(\boldsymbol{\tau}_{bk\_conservative}+\boldsymbol{\tau}_{bk\_external}\right)\end{aligned} \tag{42}$$

D. Retrieving Velocity

In practice, the translational and angular velocity in Cartesian frame are used more frequently than $\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k$ in $\tilde{\boldsymbol{f}}_k$. The transformation between $\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k$ and $\boldsymbol{\chi}_{Bk}$ can be achieved from the identity of kinetic energy.

$$T_k = \frac{1}{2}\check{\boldsymbol{\chi}}_{Bk}^T \cdot \boldsymbol{M}_{8\times 8} \cdot \check{\boldsymbol{\chi}}_{Bk} = \frac{1}{2}\tilde{\boldsymbol{P}}_{dk} \cdot \dot{\tilde{\boldsymbol{p}}}_k \tag{43}$$

where $\check{\boldsymbol{P}}_{Dk}$ denote the momenta in dual quaternion frame. Hence

$$\left(\check{\boldsymbol{\chi}}_{Bk}^T \cdot \boldsymbol{M}_{8\times 8}\right) \cdot \check{\boldsymbol{\chi}}_{Bk} = \left(\tilde{\boldsymbol{p}}_k^\dagger \otimes \check{\boldsymbol{P}}_{dk}^*\right)^* \cdot \check{\boldsymbol{\chi}}_{Bk} \tag{44}$$

Based on Equation 17, the same trapezoidal approximation can be made for $\check{\boldsymbol{P}}_{dk}$ in discrete form

$$\check{\boldsymbol{P}}_{dk} = \left(\frac{\partial T}{\partial \dot{\tilde{\boldsymbol{p}}}}\right)_k \approx \frac{\partial T_{dk}}{\partial \tilde{\boldsymbol{p}}_k} \tag{45}$$

Equation 45 shows that $\check{\boldsymbol{P}}_{dk}$ is not a function of $\check{\boldsymbol{\chi}}_{Bk}$. Since $\check{\boldsymbol{\chi}}_{Bk}$ is a pure dual quaternion, after taking derivative of $\check{\boldsymbol{\chi}}_{Bk}$ on both side of Equation 44, a simplification can be made that the 2nd-4th and 6th-8th equations of

$$\boldsymbol{M}_{8\times 8} \cdot \check{\boldsymbol{\chi}}_{Bk} = \frac{1}{2}\left(\tilde{\boldsymbol{p}}^\dagger \otimes \check{\boldsymbol{P}}_{dk}^*\right)^* \tag{46}$$

are valid. After tedious but straightforward simplification, a familiar formulation appears

$$\boldsymbol{M}_{6\times 6} \cdot \boldsymbol{\chi}_{Bk} = \frac{2}{h} \cdot \begin{bmatrix} \boldsymbol{\mathcal{A}}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k) \\ \\ \boldsymbol{\mathcal{B}}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k) \end{bmatrix} \tag{47}$$

Equation 47 reveals that Equation 38 is actually the evolution map of momentum in $\boldsymbol{e}_B$. And in each time frame with given $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Psi}_k$, the translational and angular velocity in $\boldsymbol{e}_B$ can be retrieved by

$$\boldsymbol{\chi}_{Bk} = \frac{2}{h} \cdot \boldsymbol{M}_{6\times 6}^{-1} \cdot \begin{bmatrix} \boldsymbol{\mathcal{A}}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k) \\ \\ \boldsymbol{\mathcal{B}}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k) \end{bmatrix} \tag{48}$$

IV. Newton-Raphson Solver

To solve Equation 38, Newton-Raphson method is used, which involves finding the Jacobian matrix of the equation and making proper initial estimation of the starting point.

A. Jacobian Matrix

The 6-by-6 Jacobian matrix is

$$\boldsymbol{\mathcal{J}}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k) = \begin{bmatrix} \frac{\partial \boldsymbol{\mathcal{A}}_k}{\partial \boldsymbol{\Phi}_k} & \frac{\partial \boldsymbol{\mathcal{A}}_k}{\partial \boldsymbol{\Psi}_k} \\ \\ \frac{\partial \boldsymbol{\mathcal{B}}_k}{\partial \boldsymbol{\Phi}_k} & \frac{\partial \boldsymbol{\mathcal{B}}_k}{\partial \boldsymbol{\Psi}_k} \end{bmatrix} \tag{49}$$

where

$$\begin{aligned} \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{\Phi}} =& (\boldsymbol{M}_{21}\boldsymbol{\Phi} + \boldsymbol{M}_{22}\boldsymbol{\Psi}) \cdot \frac{(\boldsymbol{\Phi} \times (\boldsymbol{\Psi} \times \boldsymbol{\Phi}) - \boldsymbol{\Psi})^T}{(1 - \boldsymbol{\Phi}^2)^{\frac{3}{2}}} + \left(-\frac{\boldsymbol{\Psi} \cdot \boldsymbol{\Phi}}{\sqrt{1 - \boldsymbol{\Phi}^2}} \boldsymbol{I} + S(\boldsymbol{\Psi}) \right) \cdot \boldsymbol{M}_{21} \\ & - \frac{(\boldsymbol{M}_{11}\boldsymbol{\Phi} + \boldsymbol{M}_{12}\boldsymbol{\Psi}) \cdot \boldsymbol{\Phi}^T}{\sqrt{1 - \boldsymbol{\Phi}^2}} - S(\boldsymbol{M}_{11}\boldsymbol{\Phi} + \boldsymbol{M}_{12}\boldsymbol{\Psi}) + \left(\sqrt{1 - \boldsymbol{\Phi}^2} \boldsymbol{I} + S(\boldsymbol{\Phi}) \right) \cdot \boldsymbol{M}_{11} \end{aligned} \tag{50}$$

$$\begin{aligned} \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{\Psi}} =& - \frac{(\boldsymbol{M}_{21}\boldsymbol{\Phi} + \boldsymbol{M}_{22}\boldsymbol{\Psi}) \cdot \boldsymbol{\Phi}^T}{\sqrt{1 - \boldsymbol{\Phi}^2}} + \left(-\frac{\boldsymbol{\Psi} \cdot \boldsymbol{\Phi}}{\sqrt{1 - \boldsymbol{\Phi}^2}} \boldsymbol{I} + S(\boldsymbol{\Psi}) \right) \boldsymbol{M}_{22} \\ & + \left(\sqrt{1 - \boldsymbol{\Phi}^2} \boldsymbol{I} + S(\boldsymbol{\Phi}) \right) \cdot \boldsymbol{M}_{12} - S(\boldsymbol{M}_{21}\boldsymbol{\Phi} + \boldsymbol{M}_{22}\boldsymbol{\Psi}) \end{aligned} \tag{51}$$

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial \boldsymbol{\Phi}} = - \frac{(\boldsymbol{M}_{21}\boldsymbol{\Phi} + \boldsymbol{M}_{22}\boldsymbol{\Psi}) \cdot \boldsymbol{\Phi}^T}{\sqrt{1 - \boldsymbol{\Phi}^2}} - S(\boldsymbol{M}_{21}\boldsymbol{\Phi} + \boldsymbol{M}_{22}\boldsymbol{\Psi}) + \left(\sqrt{1 - \boldsymbol{\Phi}^2} \boldsymbol{I} + S(\boldsymbol{\Phi}) \right) \cdot \boldsymbol{M}_{21} \tag{52}$$

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial \boldsymbol{\Psi}} = \left(\sqrt{1 - \boldsymbol{\Phi}^2} \boldsymbol{I} + S(\boldsymbol{\Phi}) \right) \cdot \boldsymbol{M}_{22} \tag{53}$$

B. Initial Estimation

In order to achieve faster convergence in Newton-Raphson solver, a proper starting point should be estimated in every time frame. The initial condition of the system is usually given as the pose-velocity or pose-momentum form. In either form, the initial pose and momentum are given. The definition of $\tilde{\boldsymbol{f}}_k$ gives out that

$$\tilde{\boldsymbol{f}}_k = \tilde{\boldsymbol{p}}_k^\dagger \otimes \tilde{\boldsymbol{p}}_{k+1} \approx \tilde{\boldsymbol{p}}_k^\dagger \otimes (\tilde{\boldsymbol{p}}_k + h\dot{\tilde{\boldsymbol{p}}}_k) = \tilde{\boldsymbol{I}} + \frac{h}{2}\check{\boldsymbol{\chi}}_{Bk} \tag{54}$$

Hence

$$\begin{bmatrix} \boldsymbol{\Phi}_k \\ \\ \boldsymbol{\Psi}_k \end{bmatrix} \approx \frac{h}{2}\boldsymbol{\chi}_{Bk} \tag{55}$$

Equation 55 reflects a constraint in this parametrization that $|\boldsymbol{\Phi}| < 1$, meaning the time step h must be chosen small enough to ensure that the incremental rotations between adjacent times steps are less than 180 degrees. Equation 55 also serves as the starting point for Newton-Raphson solver to get the initial value of $\boldsymbol{\Phi}_0$ and $\boldsymbol{\Psi}_0$. Then for every time frame, put $\boldsymbol{\Phi}_{k-1}$ and $\boldsymbol{\Psi}_{k-1}$ in Equation 38 to find $\boldsymbol{\mathcal{A}}_k$ and $\boldsymbol{\mathcal{B}}_k$. Finally, use Equation 48 & 55 to get the starting point of solving Equation 38. If the reference point is at the center of mass and there is no rotation, the first estimation of $\boldsymbol{\Psi}_k$ will be the exact solution, meaning the Newton-Raphson solver for pure translations is equivalent to the one-step forward Euler method. Iteration times requirement is studied in the next section.

C. Algorithm Implementation

The entire algorithm of DQVI can be implemented as in Algorithm 1-2.

Algorithm 1 DQVI

```
define initial pose: $\tilde{\boldsymbol{p}} \leftarrow \tilde{\boldsymbol{p}}_0$
define initial velocity: $\tilde{\boldsymbol{\chi}}_0$
estimate initial increment: $\boldsymbol{\Phi}, \boldsymbol{\Psi} \leftarrow \boldsymbol{\chi}_0 \cdot h/2$    ▷ Equation 55
find converged solution: $\boldsymbol{\Phi}, \boldsymbol{\Psi} \leftarrow \textsc{NewtonRaphson}(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{M}_{6\times 6} \cdot \boldsymbol{\chi}_0)$
loop
    gather external forces: $\boldsymbol{\tau}(\tilde{\boldsymbol{p}})_{conservative}, \boldsymbol{\tau}(\tilde{\boldsymbol{p}})_{external}$
    calculate momentum integral: $\boldsymbol{\alpha}(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\tau}), \boldsymbol{\beta}(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\tau})$    ▷ Equation 41-42
    estimate increment: $\boldsymbol{\Phi}, \boldsymbol{\Psi} \leftarrow inverse(\boldsymbol{M}_{6\times 6}) \cdot [\boldsymbol{\alpha}, \boldsymbol{\beta}]$    ▷ Equation 48
    find converged solution: $\boldsymbol{\Phi}, \boldsymbol{\Psi} \leftarrow \textsc{NewtonRaphson}(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
    calculate dual quaternion increment: $\tilde{\boldsymbol{f}}(\boldsymbol{\Phi}, \boldsymbol{\Psi})$    ▷ Equation 37
    update pose: $\tilde{\boldsymbol{p}} \leftarrow \tilde{\boldsymbol{p}} \otimes \tilde{\boldsymbol{f}}$
end loop
```

Algorithm 2 Newton-Raphson

```
function NewtonRaphson($\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\alpha}, \boldsymbol{\beta}$)
    for $i \leftarrow 1, iteration$ do
        calculate Jacobian: $\boldsymbol{\mathcal{J}}(\boldsymbol{\Phi}, \boldsymbol{\Psi})$        ▷ Equation 49-53
        converge to solution: $[\boldsymbol{\Phi}, \boldsymbol{\Psi}] \leftarrow [\boldsymbol{\Phi}, \boldsymbol{\Psi}] - inverse\,(\boldsymbol{\mathcal{J}}) \cdot [\boldsymbol{\alpha}, \boldsymbol{\beta}]$
    end for
    return $\boldsymbol{\Phi}, \boldsymbol{\Psi}$
end function
```

Based on this standard implementation, variable time step can be set up by retrieving velocity at key frame using Equation 48 then initiating a new simulation using the new time step. DQVI can also be easily modified to handle rigid body separation scenarios. After the key frame where separation happens, the two new rigid bodies created will share the same reference point and velocity, only the 6-by-6 inertia matrix is divided into two matrices. Then without performing reference point coordinate transformation like in many traditional methods, the simulation can be directly resumed.

V. Numerical Example: Spacecraft-Cargo Separation

For the purpose of demonstrating the performance of DQVI, a numerical example of spacecraft-cargo separation process is presented in this section. Before separation, the entire spacecraft is under pure rotation around its stationary center of mass without external torque or force. If the reference point is defined on the center of mass, the algorithm is effectively simplified as quaternion variational integrator[17]. Here the reference point is defined on the separation point O', which has an offset from the center of mass, see Figure 2. The reference point will then follow a “wobbling" trajectory in a ring of certain width on a spherical surface whose radius is the distance from reference point to the center of mass. This trajectory is symmetrical around the angular momentum axis $\boldsymbol{n}_W$ in $\boldsymbol{e}_W$. Use $\boldsymbol{n}_W$ as the normal vector, the normal projection of the trajectory should be restricted in a circle. The spacecraft is in a delicate balance that any accumulated error will break the stationary position of the center of mass. The separation happens without any reaction force or torque, meaning the total mechanical energy is conserved before and after the process, so does the total momentum.

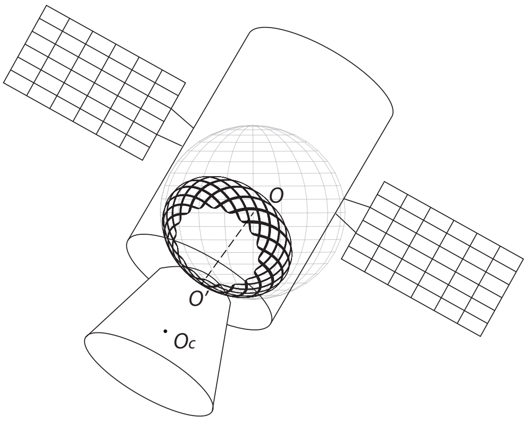


Fig. 2: Pure rotation with reference point offset

A. Before Separation: Sensitivity Study

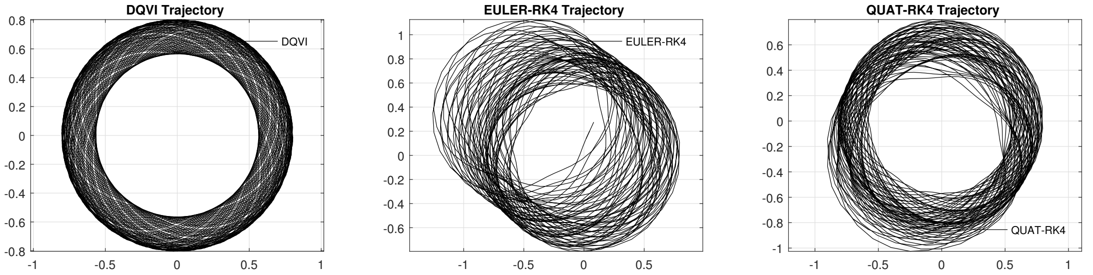
The mass of the spacecraft-cargo system is 1000 kg, with its center of mass relative to the reference point in $\boldsymbol{e}_B$ as $O'O = [1, 0.8, 0.5]m$. The moment of inertia tensor with respect to the center of mass is

$$\boldsymbol{J}_{3\times3} = \begin{bmatrix} 200 & 100 & 100 \\ 100 & 300 & 100 \\ 100 & 100 & 400 \end{bmatrix} kg \cdot m^2$$

The variables in the test are time step and initial angular velocity. The reference methods to be compared with are quaternion and Euler angles representations of classical 2nd-order Newton-Euler ODE [24] using 4th order Runge-Kutta method, denoted as “QUAT-RK4" and “EULER-RK4" respectively. QUAT-RK4 explicitly normalizes the rotation quaternion in every time step and EULER-RK4 uses the sequence in intrinsic “ZYX" order. Also to show the “gimbal lock" phenomenon, the dual-Euler angles representation[25] is intentionally not used. The iteration times of DQVI is 4, and a comparison of different iteration times is conducted in the end.

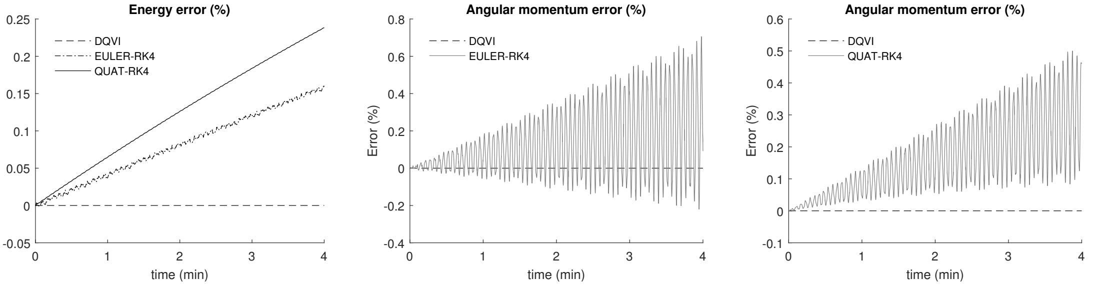
1. Time step

Two time steps are tested in the study: a large time step of 10fps (frames per second) and 60fps typical for real-time simulation. The simulation time is 4 minutes. The initial angular velocity in $\boldsymbol{e}_B$ is fixed as $[1, 1, 1]rad/s$.



(a) DQVI (b) EULER-RK4 (c) QUAT-RK4

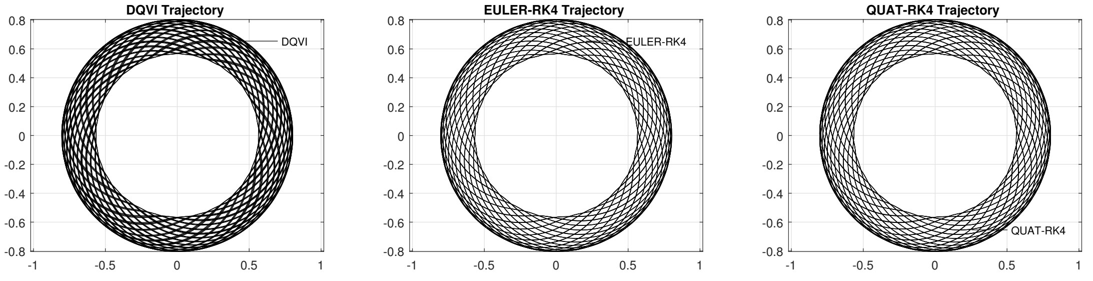
Fig. 3: Normal projection of trajectory (10fps)



(a) Energy (b) Momentum EULER-RK4 (c) Momentum QUAT-RK4

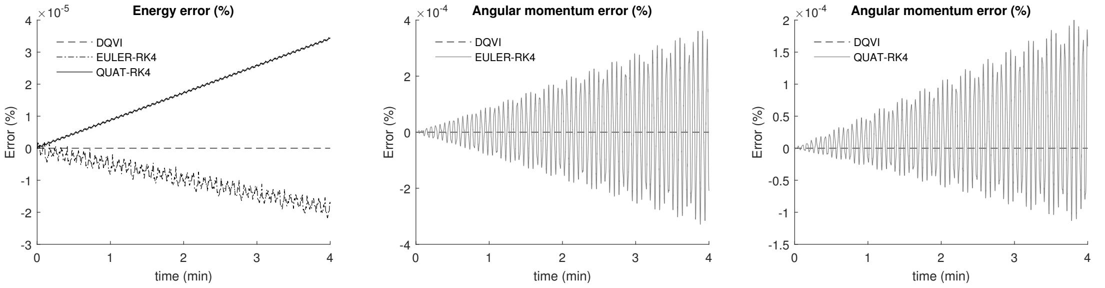
Fig. 4: Error level (10fps)

Figure 3-4 show that for large time step, the trajectory of EULER-RK4 and QUAT-RK4 quikcly drifts away due to the lack of conservation of energy and momentum. On the other hand, DQVI performs well in terms of conservation of the trajectory symmetry, energy and momentum.



(a) DQVI (b) EULER-RK4 (c) QUAT-RK4

Fig. 5: Normal projection of trajectory (60fps)



(a) Energy

(b) Momentum EULER-RK4

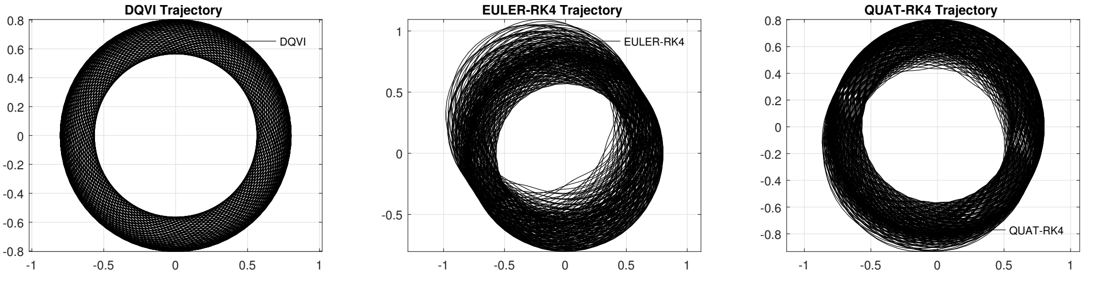
(c) Momentum QUAT-RK4

Fig. 6: Error level (60fps)

Figure 5- 6 show that the conservation of trajectory in EULER-RK4 and QUAT-RK4 is improved when time step drops, but the conservation of energy and momentum is still not achieved. By a closer comparison between Figure 3a and 5a one can find that the trajectory of DQVI under 60fps coincides better than under 10fps. This reflects the property of variational integrators that although the symmetry, the energy and the momentum of the system are conserved, the exact trajectory accuracy of DQVI is still dependent on the integration time step.

2. Angular Velocity

To study the influence of angular velocity to the simulation, two new sets of initial angular velocity in $\boldsymbol{e}_b$, $[3, 3, 3] rad/s$ and $[3, 2, 1] rad/s$ (denoted as *333* and *321*), are tested . Time step is set to be 60fps and the rest parameters remain the same as in time step study.

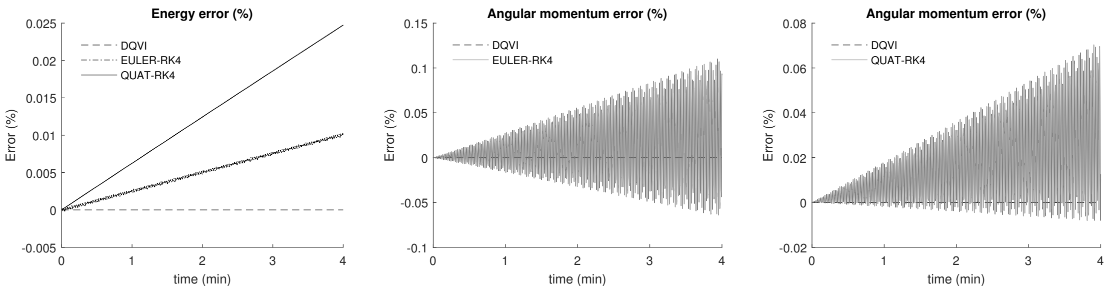


(a) DQVI

(b) EULER-RK4

(c) QUAT-RK4

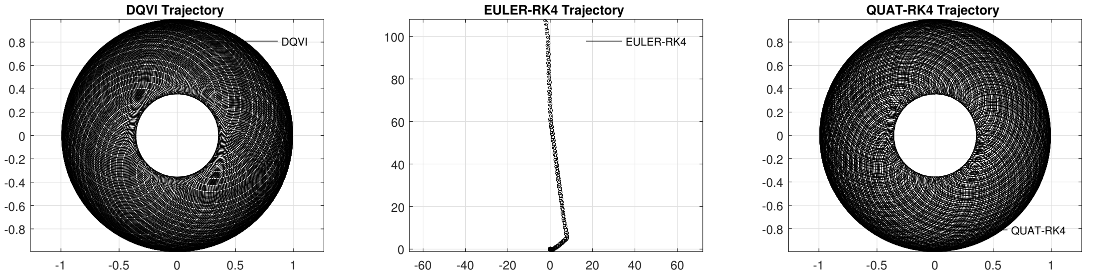
Fig. 7: Normal projection of trajectory (60fps, 333)



(a) Energy error (60fps, 333) (b) Momentum EULER-RK4 (c) Momentum QUAT-RK4

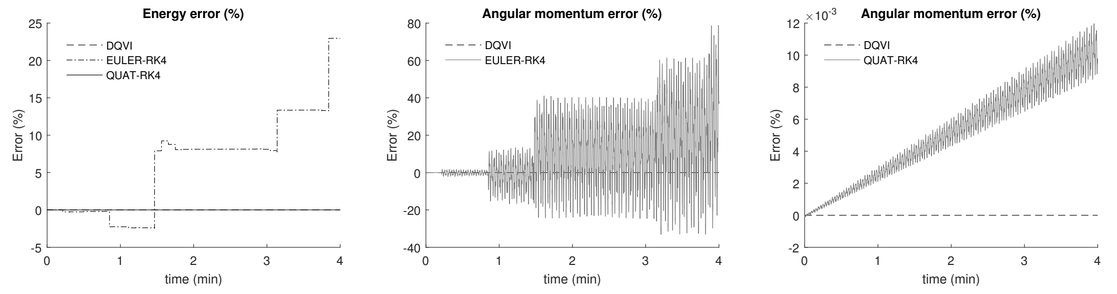
Fig. 8: Error level (60fps, 333)

The comparison between Figure 7-8 and Figure 5-6 shows that under the same time step, when the initial angular velocity increases, the error level of both EULER-RK4 and QUAT-RK4 increases as well. This indicates that the error level of the system is not only determined by time step, but also by angular velocity. On the other hand, although the trajectory of DQVI covers more area of the ring due to more revolutions made in the same time span, it is still restricted in the circle compared to worsened cases of EULER-RK4 and QUAT-RK4. And the energy and momentum are both conserved as it should be.



(a) DQVI (b) EULER-RK4 (c) QUAT-RK4

Fig. 9: Normal projection of trajectory (60fps, 321)



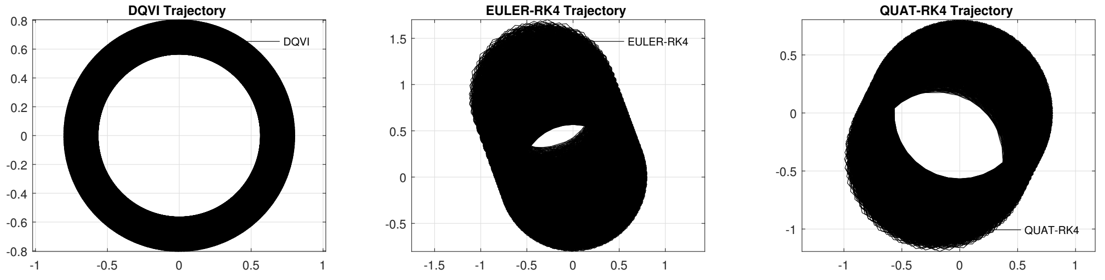
(a) Energy (b) Momentum EULER-RK4 (c) Momentum QUAT-RK4

Fig. 10: Error level (60fps, 321)

Figure 9-10 show that under certain initial conditions, EULER-RK4 will suffer from numerical instability when the second Euler angle is close to ±90 deg. This phenomenon is well known as “gimbal lock". Gimbal lock can be avoided by introducing a complex dual-Euler angles representation, or using quaternions as demonstrated by QUAT-RK4. Nevertheless, DQVI is still superior among other methods in terms of system conservation.

3. Long Term Performance

To study the dissipation/accumulation of energy and momentum one step further, with the same parameters as in time step study, the simulation time in the 60fps case is extended to 3 hours. Figure 11a & 12a show that DQVI has excellent performance in long term simulation in terms of conservation of symmetry, energy and momentum. EULER-RK4 and QUAT-RK4 (Figure 11b & 11c) as comparisons, eventually accumulate enough error to be visible on plot. In view of the conclusion from angular velocity study, the faster the object rotates, the quicker accumulated error will be visible.

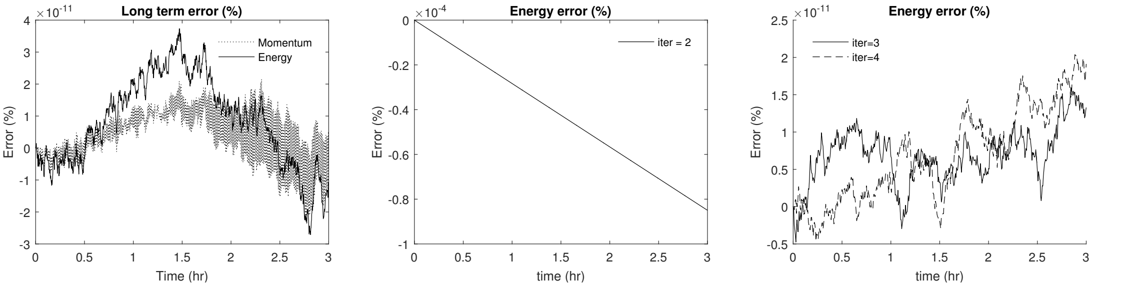


(a) DQVI

(b) EULER-RK4

(c) QUAT-RK4

Fig. 11: Normal projection of trajectory (60fps, 3-hr)



(a) DQVI (60fps)

(b) Energy DQVI (10fps, iter = 2)

(c) Energy DQVI (10fps, iter = 3,4)

Fig. 12: Long term error level (3-hr)

4. Iteration times

Since the initial estimation in Newton-Raphson solver is based on translational and angular velocity in $\boldsymbol{e}_B$ (Equation 55), the solution has a close initial point and should reach convergence in a few iteration times. Here iteration times of 2, 3 and 4 are tested with time step of 10fps in 3 hours. The rest of the parameters are the same as in time step test. Figure 12b shows that when iteration times equals 2, the energy error increases proportionally with time, meaning the solution has yet to reach convergence. Figure 12c shows once iteration time reaches 3, the energy error not only drops significantly, but also has a random pattern that is not proportional to time. This is a signal that the solution has converged to machine precision. Therefore, a iteration time of 3 is recommended for the Newton-Raphson solver. Under the same time step and without code optimization, DQVI(iter=3) has a roughly equivalent simulation speed as EULER-RK4 and DQVI(iter=4) is roughly equivalent to QUAT-RK4, also about 33% slower than DQVI(iter=3) as expected.

B. Separation

The cargo weights 400 kg with the center of mass $O'O_c = [-0.3, -0.1, -0.1]m$. The moment of inertia tensor of the cargo with respect to O_c is

$$\boldsymbol{J}_{c3\times3} = \begin{bmatrix} 100 & 20 & 20 \\ 20 & 100 & 20 \\ 20 & 20 & 100 \end{bmatrix} kg \cdot m^2$$

The entire 6-by-6 inertia matrix of the spacecraft-cargo system is separated into two matrices. Time step is set as 60fps and the rest parameters are identical in time step study. The simulation lasts 6 seconds and the separation happens at $t = 3s$. The gray lines indicate the trajectory of the center of mass vector of the spacecraft, and the black lines indicate the cargo, see Figure 13. After separation, the reference points of both spacecraft and cargo are still on the separation point. No velocity coordinate transformation is required at separation key frame.

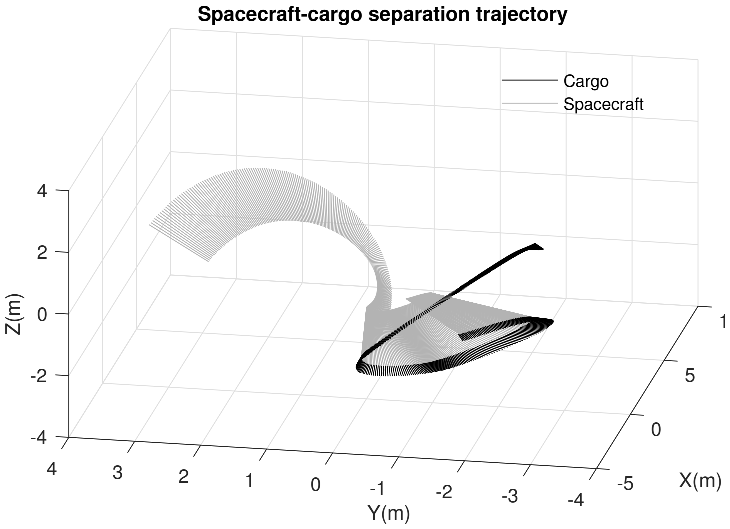


Fig. 13: Spacecraft-cargo separation

VI. Conclusion

The Dual Quaternion Variational Integrator introduced in this paper treats the rigid body pose in $SE(3)$ as an entire Lie group without decoupling translation and rotation. A parametrization of the incremental pose with 6 independent variables guarantees that the system will evolve in $SE(3)$. The numerical example shows that under a variety of settings, DQVI consistently outperforms the traditional 2nd-order Newton-Euler differential equations in terms of conservation of geometric structure, energy and momentum. DQVI allows reference point and 6-by-6 inertia matrix to be arbitrarily defined. In situations when separation happens, no coordinate transformation is required, providing more generality to possible spacecraft applications. Multi-step and variable time step integration method can be developed based on our work.

Reference

[1] Leok, M., "An overview of Lie group variational integrators and their applications to optimal control," in "International Conference on Scientific Computation and Differential Equations," , 2007, p. 1.

[2] Marsden, J. E., Pekarsky, S., and Shkoller, S., "Discrete euler-poincare and Lie-poisson equations," *Nonlinearity*, p. 1647,
doi:10.1.1.124.4644.

[3] Cendra, H., Marsden, J. E., Pekarsky, S., and Ratiu, T. S., "Variational Principles for Lie-Poisson and Hamilton-PoincarÃľ Equations," *Moscow Mathematical Journal*, Vol. 3, No. 3, 2003, pp. 833–867,
doi:10.1.1.131.219.

[4] Bloch, A., Krishnaprasad, P., Marsden, J. E., and Ratiu, T. S., "The Euler-Poincare Equations and Double Bracket Dissipation," *Communications in Mathematical Physics*, pp. 1–42,
doi:10.1.1.142.6349.

[5] Lewis, D. and Simo, J., "Conserving Algorithms for the Dynamics of Hamiltonian Systems on Lie Groups," *Nonlinear Science*, Vol. 4, 1994, pp. 253–299,
doi:10.1007/BF02430634.

[6] Nordkvist, N. and Sandyal, A. K., "A Lie Group Variational Integrator for Rigid Body Motion in SE(3) with Applications to Underwater Vehicle Dynamics," in "49th IEEE Conference on Decision and Control," , 2010, pp. 5414–5419,
doi:10.1109/CDC.2010.5717622.

[7] Kobilarov, M., Crane, K., and Desbrun, M., "Lie Group Integrators for Animation and Control of Vehicles," Tech. rep., Caltech, 2009.

[8] Lee, T. and McClamroch, N. H., "A Lie Group Variational Integrator for the Attitude Dynamics of a Rigid Body with Applications to the 3D Pendulum," in "Proceedings of the 2005 IEEE Conference on

Control Applications," , 2005, pp. 962–967,
doi:10.1109/CCA.2005.1507254.

[9] Lee, T., Leok, M., and McClamroch, N. H., "Lie group variational integrators for the full body problem," *Computer Methods in Applied Mechanics and Engineering*, Vol. 196, No. 29-30, 2007, pp. 2907–2924,
doi:10.1016/j.cma.2007.01.017.

[10] Kenwright, B., "A Beginners Guide to Dual-Quaternions: What They Are, How They Work, and How to Use Them for 3D Character Hierarchies," in "The 20th International Conference on Computer Graphics, Visualization and Computer Vision," , 2012, pp. 1–13,
doi:10.1.1.407.9047.

[11] Filipe, N., Kontitsis, M., and Tsiotras, P., "Extended Kalman Filter for Spacecraft Pose Estimation Using Dual Quaternions," *Journal of Guidance, Control, and Dynamics*, Vol. 38, 2015, pp. 1625–1641,
doi:10.2514/1.G000977.

[12] Lee, U. and Mesbahi, M., "Dual Quaternions, Rigid Body Mechanics, and Powered-Descent Guidance," in "2012 IEEE 51st IEEE Conference on Decision and Control (CDC)," , 2012, pp. 3386–3391,
doi:10.1109/CDC.2012.6425996.

[13] Kwon, J.-W., Lee, D., and Bang, H., "Virtual Trajectory Augmented Landing Control Based on Dual Quaternion for Lunar Lander," *Journal of Guidance, Control, and Dynamics*, Vol. 39, 2016, pp. 2044–2057,
doi:10.2514/1.G001459.

[14] Reynolds, R. G., "Quaternion Parameterization and a Simple Algorithm for Global Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 21, 2016, pp. 669–672,
doi:10.2514/2.4290.

[15] Dooley, J. and McCarthy, J., "Spatial Rigid Body Dynamics using Dual Quaternion Components," in "IEEE International Conference on Robotics and Automation," , 1991, pp. 90–95,
doi:10.1109/ROBOT.1991.131559.

[16] Jing, Y., Zhihao, C., and Wang, Y., "Modeling of The Quadrotor UAV Based on Screw Theory via Dual Quaternion," in "AIAA Modeling and Simulation Technologies (MST) Conference," , 2013,
doi:10.2514/6.2013-4594.

[17] Manchester, Z. R. and Peck, M. A., "Quaternion Variational Integrators for Spacecraft Dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 39, 2016, pp. 69–76,
doi:10.2514/1.G001176.

[18] Blanco, J.-L., "A tutorial on SE(3) transformation parameterizations and on-manifold optimization,"

Tech. rep., MAPIR Group, 2014.

[19] Wang, X., Han, D., Yu, C., and Zheng, Z., "The geometric structure of unit dual quaternion with application in kinematic control," *Journal of Mathematical Analysis and Applications*, Vol. 389, 2012, pp. 1352–1364,
doi:10.1016/j.jmaa.2012.01.016.

[20] Akyar, B., "Dual quaternions in spatial kinematics in an algebraic sense," *Turkish Journal of Mathematics*, pp. 373–391.

[21] Daniilidis, K., "Hand-eye calibration using dual quaternions," *International Journal of Robotics Research*, pp. 18, 286–298,
doi:10.1.1.18.366.

[22] Kavan, L., Collins, S., O'Sullivan, C., and Zara, J., "Dual Quaternions for Rigid Transformation Blending," Tech. rep., Trinity College Dublin, 2006.

[23] Holm, D. D., *Geometric Mechanics, Part II: Rotating, Translating and Rolling*, Imperial College Press, London, 2011.

[24] Fossen, T. I., *Handbook of Marine Craft Hydrodynamics and Motion Control*, Wiley, Trondheim, Norway, 2011.

[25] Huang, X., "A dual-Euler method for solving all-attitude angles of the aircraft," in "Flight Simulation and Technologies," , 1993, pp. 257–262,
doi:10.2514/6.1993-3589.