Rigid Body Dynamics and Conformal Geometric Algebra

Anthony Lasenby, Chris Doran and Robert Lasenby

University of Cambridge

June 15, 2010
AGACSE Meeting, Amsterdam

Introduction

- Geometric Algebra is a good mathematical framework for Rigid Body Dynamics
- Makes the Euler equations and its solutions look pretty simple
- Will discuss the application of conformal geometric algebra (CGA) to this area. Will seek covariant formulation based on a Lagrangian action principle
- Will take this through to point of considering interactions of multiple bodies, and implementation of a CGA version of the 'Fast Frictional Dynamics' approach
- This is done in a '2 up' implementation of CGA — will also look at a formulation in a '1 up' approach — penalty here is working in a curved space, but definitely interesting

The conformal geometric algebra — notation!

- Here we will adhere to David Hestenes' original notation (as in 'Clifford Algebra to Geometric Calculus')
- We adjoin to ordinary 3d space:

$$e: \quad e^2 = +1$$
$$\bar{e}: \quad \bar{e}^2 = -1$$

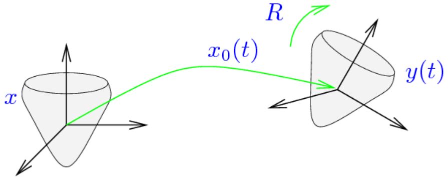
  and this allows us to create

$$\begin{cases} n = e + \bar{e} & = \text{point at infinity} \\ \bar{n} = e - \bar{e} & = \text{origin} \end{cases}$$

- So in terms of the Amsterdam Protocol we have

$$e = e_+, \quad \bar{e} = e_-$$
$$n = n_\infty \quad \bar{n} = n_0$$

- Will start by looking at treatment of rigid body motion in ordinary 3d GA
- Have a rigid body moving through space. Relate the vector position of points in the moving body $y(t)$ back to a fixed 'reference' body.



x_0 is the position in space of the centre of mass. Have

$$y(t) = R(t)x\tilde{R}(t) + x_0(t)$$

Places the rotational motion in the time-dependent rotor $R(t)$.

Rigid Body Dynamics II

- Next need an expression for the angular velocity. This must be a bivector as well.
- Find that $\Omega_S = -2\dot{R}\tilde{R}$ (can check this is a bivector)
- Dynamics reduces to the single rotor equation

$$\dot{R} = -\frac{1}{2}\Omega_S R \quad \text{or} \quad \dot{\tilde{R}} = \frac{1}{2}\tilde{R}\Omega_S$$

- Equations like this are very common in physics.
- Can also express in terms of the body angular velocities, Ω_B, *i.e.* Ω_S expressed back in the 'reference' copy

$$\Omega_S = R\Omega_B\tilde{R}, \quad \Omega_B = \tilde{R}\Omega_S R$$

Rigid Body Dynamics III

- We need the angular momentum bivector L

$$
\begin{aligned}
L &= \int d^3x\, \rho(y - x_0)\wedge v \\
&= \int d^3x\, \rho(Rx\tilde{R})\wedge(R\, x\cdot\Omega_B\, \tilde{R} + v_0) \\
&= R\left(\int d^3x\, \rho x\wedge(x\cdot\Omega_B)\right)\tilde{R}
\end{aligned}
$$

- From this we extract inertia tensor $\mathcal{I}$

$$\mathcal{I}(B) = \int d^3x\, \rho x\wedge(x\cdot B)$$

A linear function mapping bivectors to bivectors.

- Rotate to

$$L = R\mathcal{I}(\Omega_B)\tilde{R}$$

- Now $\dot{L} = T$ (the couple as a bivector), so form

$$\begin{aligned}\dot{L} &= \dot{R}\mathcal{I}(\Omega_B)\tilde{R} + R\mathcal{I}(\Omega_B)\dot{\tilde{R}} + R\mathcal{I}(\dot{\Omega}_B)\tilde{R} \\ &= R[\mathcal{I}(\dot{\Omega}_B) - \tfrac{1}{2}\Omega_B\mathcal{I}(\Omega_B) + \tfrac{1}{2}\mathcal{I}(\Omega_B)\Omega_B]\tilde{R} \\ &= R[\mathcal{I}(\dot{\Omega}_B) - \Omega_B\times\mathcal{I}(\Omega_B)]\tilde{R}.\end{aligned}$$

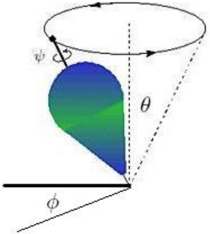
- The torque-free equation $\dot{L} = 0$ reduces to

$$\mathcal{I}(\dot{\Omega}_B) - \Omega_B\times\mathcal{I}(\Omega_B) = 0$$

- So this is the 3d GA version of the Euler equations.
- Use this to get Ω, and can get neat rotor solutions, e.g. in the case of the symmetric top:

$$R(t) = \exp\left(-\frac{1}{2} i_1^{-1} L t\right) R(0) \exp\left(-\frac{1}{2}\omega_3 \left(1 - i_3/i_1\right) I e_3 t\right)$$

- Fully describes the motion of a symmetric top. Here $i_1 = i_2$ are the two equal moments of inertia, and i_3 is the moment of inertia about the symmetry axis.
- We have an 'internal' rotation in the $e_1 e_2$ plane (a symmetry of the body), followed by a rotation in the angular-momentum plane.



- The representation function we will use has the origin as $-\bar{n}/2$, i.e.

$$X = \frac{1}{2\lambda^2}\left(\boldsymbol{x}^2 n + 2\lambda \boldsymbol{x} - \lambda^2 \bar{n}\right)$$

  where $\boldsymbol{x}$ is the ordinary position vector in Euclidean 3-space.
- Note that X so defined is covariantly normalised since it satisfies $X \cdot n = -1$.
- Overall idea is to set up a Lagrangian which is covariant with respect to the 5d geometry, but for which the energy is just the ordinary 3d rigid body energy.
- Then we should get equations of motion which are correct at the 3d level, but covariantly expressed in 5d.

- The state of the body can be represented by

$$R(t) = R_1(t)R_2(t)$$

- Here $R_2(t)$ is an ordinary 3d rotor, giving the *attitude* of the body, and $R_1(t)$ is the translation rotor which takes the origin of the reference copy of the body to where the centre of mass is actually located at time t.
- Explicitly

$$R_1 = 1 - \frac{1}{2\lambda}\boldsymbol{x}_c(t)n$$

where $\boldsymbol{x}_c(t)$ is the 3d position of the centre of mass at time t.

Rigid body dynamics in 5d CGA III

- If X_{ref} represents the position of a point in the reference copy of the body, then the null vector corresponding to the actual position of this point is

$$X = RX_{\mathrm{ref}}\tilde{R}$$

- Now proceed in analogy to the GA treatment of rigid body dynamics in 3d. The *body angular velocity* bivector is still defined by

$$\Omega_B = -2\tilde{R}\dot{R}$$

- Using this, we get that the time derivative of the 5d null vector of position is

$$\dot{X} = R\, X_{\mathrm{ref}} \cdot \Omega_B\, \tilde{R}$$

- This means

$$\dot{X}^2 = (X_{\mathrm{ref}} \cdot \Omega_B)^2$$

- Evaluating $\dot{X}$ explicitly, we have

$$\dot{X} = \frac{1}{2\lambda^2}\left(2\boldsymbol{x}\cdot\dot{\boldsymbol{x}}\, n + 2\lambda\dot{\boldsymbol{x}}\right)$$

- Note the form of this — 3d vector plus multiple of n
- Multiplied by m (for a point mass) this would be momentum — corresponds to general form of free vectors
- Get

$$\dot{X}^2 = \frac{\dot{\boldsymbol{x}}^2}{\lambda^2}$$

- This has therefore returned the 3d velocity squared, which is what we need to evaluate the kinetic energy. We can thus write

$$T = \frac{\lambda^2}{2} \int d^3x_b \, \rho \, (X_{\mathrm{ref}} \cdot \Omega_B)^2$$

where the integration is over the 3d reference body, and ρ is the density (in general a function of 3d position).

- Now

$$(X_{\mathrm{ref}} \cdot \Omega_B)^2 = -\Omega_B \cdot (X_{\mathrm{ref}} \wedge (X_{\mathrm{ref}} \cdot \Omega_B))$$

- Thus have

$$T = -\frac{1}{2}\Omega_B \cdot I(\Omega_B) \quad \text{where} \quad I(\Omega_B) = \lambda^2 \int d^3x_b \, \rho \, X_{\mathrm{ref}} \wedge (X_{\mathrm{ref}} \cdot \Omega_B)$$

- This is therefore our version of the moment of inertia tensor.

Setting up the Lagrangian I

- Need to set up our 5d Lagrangian
- Because we are working with 5d spinors, need some further restrictions to ensure that we are working with a rotor.
- In addition, we only want to include rotors of a specific type, namely those that preserve the point at infinity n, since these correspond to the rigid body motions we are considering
- Our full Lagrangian is thus taken as

$$\mathcal{L} = \left\langle -\frac{1}{2}\Omega_B \cdot I(\Omega_B) - \mu(\psi\tilde{\psi} - 1) - IU\psi\tilde{\psi} - V(\psi n\tilde{\psi} - n) \right\rangle$$

where μ is a scalar, I the 5d pseudoscalar, and U and V are general 5d vectors.

- Here Ω_B is taken to be defined in terms of the (general) spinor ψ as follows:

$$\Omega_B = -\tilde{\psi}\dot{\psi} + \dot{\tilde{\psi}}\psi$$

  which can be seen to be always a bivector.
- The function of the general grade 4 Lagrange multiplier IU is to ensure that the grade 4 part of $\psi\tilde{\psi}$ is zero, and the function of V is to restrict to rigid body motions.
- With all these restrictions in place, enforced by the multipliers, a general ψ turns into a rotor of the desired form.

The equations of motion and conservation laws I

We employ the usual techniques (e.g. Lasenby et al. 'Multivector derivatives and Lagrangian Field Theory, 1993) to get the equations of motion. These are

$$I(\dot{\Omega}_B) - \Omega_B I(\Omega_B) = \mu + I\tilde{\psi}U\psi + \tilde{\psi}V\psi n$$

Assuming that ψ is of the desired form, and restricting to the various grades we have

$$\begin{aligned} -\Omega_B \cdot I(\Omega_B) &= \mu + V \cdot n && \text{scalar part} \\ I(\dot{\Omega}_B) - \Omega_B \times I(\Omega_B) &= \tilde{\psi}(V \wedge n)\psi && \text{bivector part} \\ -\Omega_B \wedge I(\Omega_B) &= \tilde{\psi} I U \psi && \text{grade 4 part} \end{aligned} \tag{1}$$

- The middle one of these is the direct analogue of the Euler equations.

The equations of motion and conservation laws II

- Shows us that the effect of the Lagrange multiplier enforcing the restriction to rigid body motions, is to introduce the 'space torque' $V \wedge n$, which appears in the Euler equation back-rotated to the body frame.
- If we employ the techniques in 'A Multivector Derivative Approach to Lagrangian Field Theory' to find the angular momentum, and the conservation law it satisfies, then under the change

$$\psi \mapsto e^{\alpha B/2}\psi, \quad V \mapsto e^{\alpha B/2} V e^{-\alpha B/2}, \quad U \mapsto e^{\alpha B/2} U e^{-\alpha B/2}$$

  where α is a scalar and B is a constant bivector, we find that all the terms in the Lagrangian are invariant except the final one, $V \cdot n$.

The equations of motion and conservation laws III

- This leads to a non-conservation of angular momentum, with the result

$$\frac{dL}{dt} = V \wedge n$$

  which makes sense since we've already identfied $V \wedge n$ as a torque operating on the system.
- The angular momentum itself is given by

$$L = \psi I(\Omega_B)\tilde{\psi}$$

  in the usual way.
- The conserved Hamiltonian is found to be simply the kinetic energy

$$H = -\frac{1}{2}\Omega_B \cdot I(\Omega_B)$$

Solving the equations I

- Appearance of V in the main Euler equation is awkward
- We don't have a dynamical equation for it, and secondly since it introduces an explicit ψ in what should be the bivector update equation.
- Solution of this is to wedge both sides of the bivector equation with n. This yields

$$\boxed{\Big(I(\dot{\Omega}_B) - \Omega_B \times I(\Omega_B)\Big) \wedge n = 0} \tag{2}$$

- It is this equation which appears to give the required dynamics most quickly, since it contains neither V nor ψ.

An example — the dumbbell I

- To make the above concrete, we consider a simple example, a 2d dumbbell.
- This is taken to consist of unit mass points at $x = \pm x_0$, $y = z = 0$, and be moving just in the (x, y) plane. The inertia tensor is then just

$$I(\Omega_B) = \lambda^2 \left(X_1 \wedge (X_1 \cdot \Omega_B) + X_2 \wedge (X_2 \cdot \Omega_B)\right)$$

  where $X_1 = X(x = x_0, y = 0, z = 0)$ and $X_2 = X(x = -x_0, y = 0, z = 0)$.
- As an example of how one could proceed, we parameterise Ω_B as

$$\Omega_B = f(t) e_1 e_2 + g(t) e_1 n + h(t) e_2 n$$

An example — the dumbbell II

- This is the most general angular velocity bivector compatible with the constraints on ψ and the fact the motion is 2 dimensional.
- Equation (2) reads

$$x_0^2 \dot{f}\, e_1 e_2 n - \lambda^2 \left((-\dot{g} + fh) e_1 + (\dot{h} + fg) e_2 \right) e\bar{e} = 0$$

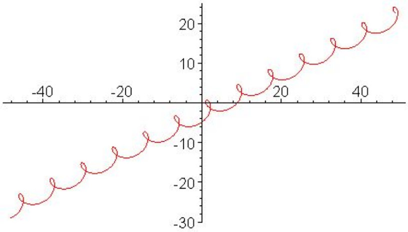
- From this we can read off the solution $f = \text{const.}$ and

$$g(t) = c_1 \cos(ft) + c_2 \sin(ft), \quad h(t) = c_2 \cos(ft) - c_1 \sin(ft)$$

- These are the correct 'angular velocities' for a body moving with constant translational velocity $(\lambda c_1, \lambda c_2)$ and rotating at constant rate f.

An example — the dumbbell III

- Going back to find the vector Lagrange multiplier V for this case, one finds that its $\bar{n}$ component is constant, and equal to minus the translational kinetic energy.
- The spatial part carries out a type of helical motion, at twice the underlying frequency f, looking rather like zitterbewegung. (David Hestenes' favourite topic!).



Plot of the locus of the spatial part of the Lagrange multiplier V for the 2d dumbbell case. $(V_1(t), V_2(t))$ is plotted against t. The $\bar{n}$ component of V is constant and equal to minus the translational kinetic energy.

Including moments, forces and collisions I

- At this point, we need to start generalising to more complicated cases
- We need to consider the effects of moments, forces and collisions
- Turns out the covariant notion of a force is an F satisfying $F \cdot n = 0$. Specifically we write

$$F = \boldsymbol{f} + \alpha n$$

where $\boldsymbol{f}$ is the normal Euclidean 3d force vector, and α is a multiplier. (Corresponds to the notion of Euclidean boundary points in Lasenby, 2004.)

Including moments, forces and collisions II

- Our fundamental equation is as follows. For a force F acting on a body at the contact position given by X^{cont} (note this is on the surface of the real body, not the reference body) then the total effect of this force is to produce the 'moment'

$$M_{\mathrm{space}} = \lambda\, X^{\mathrm{cont}} \wedge F$$

- This encompasses both the actual torque about the centre of mass and the effect on the c.o.m. motion itself
- A further job would be to add in the effects of gravity — this can be done nicely at the Lagrangian level in terms of a conservative potential, but won't give details

Including moments, forces and collisions III

- Then a central thing that will be needed for a numerical algorithm will be a way of updating the body angular velocity bivector Ω_B at each time step.
- Equation (2), supplemented by various terms on the right hand side representing the external torques in the body frame, at first glance does not look very promising for this purpose
- As well as $\dot{\Omega}_B$ being operated on by the inertia tensor, the result is wedged with n, rasing possible fears about invertibility and uniqueness.
- In fact all is well, and we can proceed as follows. Let's rewrite (2) as

$$I(\dot{\Omega}_B) \wedge n = (\Omega_B \times I(\Omega_B)) \wedge n + T \tag{3}$$

where T is the sum of the trivector torques in the body frame.

- Then turns out we can explicitly invert (3) via

$$\dot{\Omega}_B = [(\Omega_B \times I(\Omega_B)) \wedge n + T] \cdot F^i E_i \tag{4}$$

where a sum over i is taken, and F^j and E_i are sets of bivectors that we can always form, given knowledge of the inertia tensor.

- So this does the job!
- Suppose the body experiences a collision.
- This will give rise to a trivector impulsive torque T_{imp}
- We then input this to the impulsive version of (4), to obtain

$$\boxed{\Delta\Omega_B = T_{\mathrm{imp}} \cdot F^i E_i}$$

- Here $\Delta\Omega_B$ is the sudden jump that will occur in Ω_B due to the impulse. We note that although Ω_B changes, there will be no change in the rotor ψ — it is just the state of motion that alters, not the position or orientation.
- To calculate the value of the impulse, we can use energy conservation. Specifically, if Ω_B^{old} is the value of Ω_B just before the impulse, we require

$$-\frac{1}{2}(\Omega_B^{\mathrm{old}} + \Delta\Omega_B)\cdot I(\Omega_B^{\mathrm{old}} + \Delta\Omega_B) = -\frac{1}{2}\Omega_B^{\mathrm{old}}\cdot I(\Omega_B^{\mathrm{old}})$$

- This provides an equation which is in general quadratic in T_{imp}.
- One solution, however, is that $T_{\mathrm{imp}} = 0$, and so it turns out that the value we want is the root of a linear equation, giving a nice simple answer.

Implemenation

- (In a non-perfectly reflecting case, where the energy changes, then can work by specifying a 'coefficient of restitution' α — details in paper.)
- Several things to be considered in an actual implementation
- We based our approach on the methods described in Fast Frictional Dynamics for Rigid Bodies by Kaufman, Edmunds & Pai (2005)
- This deals with constraints and contact forces, including friction, in a way that scales only linearly with the number of objects involved
- They evolved their own methods for treating moments and forces in a unified fashion — pretty complicated — believe the CGA approach is better as regards conceptual ease in programming

Implemenation II

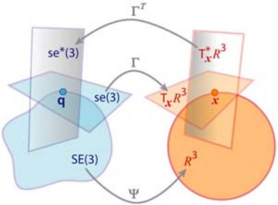


Figure 10: A Diagram of $SE(3)$ mappings.

- Our programming, using wholly CGA approach, carried out by Robert Lasenby
- Discovered that the FFD approach has some serious limitations — basically doesn't deal properly with the interactions, and can get some anomalous results — however on a surface level looks pretty good, and is definitely fast

An alternative approach I

- Instead of the '2 dimensions up' approach, one can also think about carrying out rigid body computations in a '1-d up' approach
- Idea here is to work in curved space
- Remember the setup we used for CGA: we adjoin to ordinary 3d space:

$$e: \quad e^2 = +1$$
$$\bar{e}: \quad \bar{e}^2 = -1$$

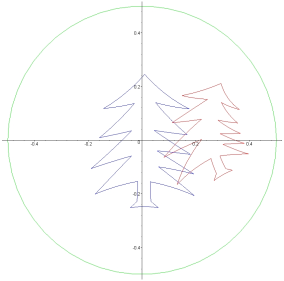
and this allows us to create

$$\begin{cases} n = e + \bar{e} & = \text{point at infinity} \\ \bar{n} = e - \bar{e} & = \text{origin} \end{cases}$$

- Euclidean geometry keeps n constant

An alternative approach II

- We can do spherical geometry by keeping $\bar{e}$ constant and hyperbolic geometry by keeping e constant!
- An advantage is that then only need one extra dimension (compared to 3d space) for computations
- E.g., if work in spherical space, then since $\bar{e}$ kept constant under transformations, don't need to include it anywhere in computations, and can work in just a 4d space.
- Note all the advantages of translations and rotations being unified in a rotor structure are still present



An alternative approach III

- The 'radius of curvature' of the space can be taken to $\mapsto \infty$ at the end of the computations (if desired) and we will get back to Euclidean space-type answers.
- Interesting point of using such an approach for rigid bodies, is that the 'counting' is now exactly right
- Translations and rotations in 3d have 6 d.o.f and this matches number of bivectors (and therefore generators of motion) in 4d (i.e. in 1-d up)
- Works generally — to describe rotation in n dimensions, we need $1/2n(n-1)$ components to describe rotation, whilst position needs another n
- Therefore the total is $1/2n(n+1)$, which is the same as the number of bivector (and therefore independent rotor) components in $n+1$ dimensions

An alternative approach IV

- The other interest in working in such spaces is effectively one from physics: does the notion of a rigid body and its dynamics even make sense in a curved space?
- In a space of constant curvature (such as the spherical or hyperbolic spaces we are using) the answer seems to be yes, and the CGA provides a consistent way to formalise it
- (Probably a good job, since out current space probably is curved — in fact think we are heading towards an asymptotic de Sitter phase in which the universe is closed spatially, with constant curvature everywhere.)
- Currently working on rigid body motion in curved space, but here are just two equations and an example:

An alternative approach

- We take as our Lagrangian

$$\mathcal{L} = \left\langle -\frac{1}{2}\Omega_B \cdot I(\Omega_B) - \mu(\psi\tilde{\psi} - 1) - \nu I \psi\tilde{\psi} \right\rangle$$

  where μ and ν are scalars, and ψ a general spinor in 4d.
- μ and ν jointly enforce the constraint that ψ should be a rotor ($\psi\tilde{\psi} = 1$)
- Working in a $(+,+,+,+)$ space (therefore spherical geometry) can set up our free dumbbell example again, and see how it moves
- Note all the expected features of rigid body motion work here — e.g. can define 4d angular momentum and energy, and they are conserved
- Remains to be investigated whether this approach offers any genuine computational advantages over 2-d up approach — work in progress!

Summary

- Conformal Geometric Algebra, employed in a 2d-up approach, allows us to describe and encode rigid-body dynamics efficiently
- Gives an alternative (integrated fully with the rest of GA) for the many mathematical schemes used to cope with joint treatment of translations and rotations
- Lagrangian formulation means physics can be incorporated properly, and that all equations will be covariant
- Fast Frictional Dynamics approach indeed fast, but some problems — probably this whole area still needs to be sorted out
- 1-d up approach in curved space interesting and clearly can be made to work
- However, computational advantages still need to be assessed