

К ЧИСЛЕННОМУ МОДЕЛИРОВАНИЮ ЭФФЕКТА ДЖАНИБЕКОВА

В.И. Олифер

Эффект теннисной ракетки или теорема о промежуточной оси следуют из результатов классической механики, описывающих свободное вращение твердого тела с тремя различными главными моментами инерции. Это также называется эффектом Джанибекова после того, как российский космонавт Владимир Джанибеков в 1985 году на космической станции «Салют-7» заметил одно из логических следствий этой теоремы в космосе, хотя теоретически этот эффект был известен как минимум за 150 лет до этого. Эффект Джанибекова заключается в том, что тело свободно вращающееся в невесомости и имеющее определённые главные моменты инерции и начальные скорости вращения, ведёт себя следующим образом: сначала оно (тело) вращается вокруг одной оси, потом эта ось вдруг неожиданно переворачивается в противоположенную сторону (совершает кувырок), после чего тело продолжает вращаться как и до переворота, потом ось опять переворачивается в противоположенную сторону, возвращаясь в исходное положение, и тело опять вращаться как в начале, а потом опять ось переворачивается и так далее. Заслуга Джанибекова в том, что он обнаружил особые начальные условия (определённые моменты инерции и угловые скорости) при которых тело ведёт себя столь странным образом.

Если тело вращается в условиях, когда какое-либо воздействие извне отсутствует, то устойчивым оказывается только вращение вокруг главных осей, соответствующих максимальному и минимальному значениям момента инерции, то есть векторы момента импульса и угловой скорости совпадают по направлению вдоль этих осей в течение всего движения и никакого кувыркания не происходит. Вращение же вокруг оси, соответствующей промежуточному по величине моменту инерции, будет неустойчивым и вызывает прецессию (изменение) вектора угловой скорости вокруг вектора момента импульса, и в результате возникают кувырки тела в полете. Т.е. при определенном даже малом воздействии происходит явление кувырка. Видео материал демонстрирующий описанный эффект можно найти в [1].

1. Постановка задачи и схема ее решения

Рассмотрим проявление этого эффекта на примере тонкой прямоугольной пластинки размером $a \times b$. Обычно при рассмотрении вращательного движения твердого тела вводятся две правые прямоугольные системы координат: $O123$ и $Oxyz$ (см. рис. 1.1).

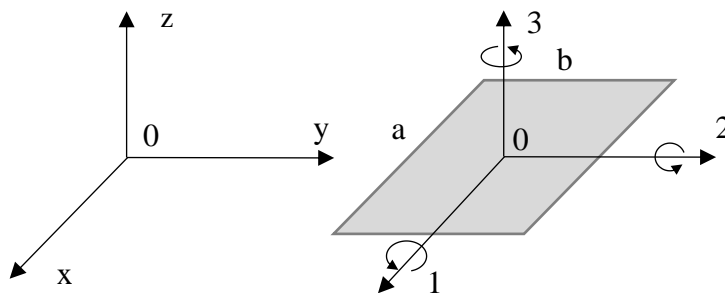


Рис. 1.1. Системы координат

Первая из них является подвижной системой координат, и она жестко связана с самим телом. Ее начало совпадает с центром тяжести тела. Оси подвижной системы являются главными осями инерции – осями тела. Вторая система координат связана с наблюдателем и называется абсолютной системой координат. Система $Oxyz$ фиксирована в трехмерном пространстве, а система координат $O123$ вращается относительно абсолютной системы $Oxyz$. В начальный момент времени обе системы координат совпадают.

Система уравнений Эйлера для вращений осесимметричного тела с тремя различающимися главными моментами инерции $I_1 < I_2 < I_3$ и угловыми скоростями $\omega_1, \omega_2, \omega_3$ вокруг главных осей инерции 1, 2 и 3 имеет вид начальной краевой задачи [2] ($k = 1, 2, 3$):

$$\dot{\omega}_k(t) = p_k \cdot \omega_n(t) \cdot \omega_m(t) + N_k(t, \omega_1, \omega_2, \omega_3)/I_k \quad (1.1)$$

с

$$p_k = (I_n - I_m)/I_k, \quad (1.2)$$

$$n, m = \begin{cases} 2, 3 & \text{при } k = 1 \\ 3, 1 & \text{при } k = 2 \\ 1, 2 & \text{при } k = 3 \end{cases}$$

где: I_k – главные моменты инерции; ω_k – проекции вектора абсолютной угловой скорости вращения; N_k – внешние вращающие моменты; $\dot{\omega}_k(t) = d\omega_k(t)/dt$. Начальными условиями указанной системы уравнений служат следующие соотношения $\omega_k(0) = \omega_{k0}$.

Для тонкой прямоугольной пластины единичной массы главные моменты инерции определяются по формулам [3]: $I_2 = a^2/12$, $I_1 = \alpha^2 I_2$, $I_3 = I_1 + I_2$, $\alpha = b/a$. Тогда $p_1 = -1$, $p_2 = 1$ и $p_3 = I_2(\alpha^2 - 1)/I_3$.

Уравнения (1.1) образуют замкнутую систему уравнений, интегрирование которой дает зависимость величин ω_k от времени t и начальных условий ω_{k0} . При $N_k = 0$ имеем так называемый случай Эйлера, для которого решение может быть получено (при дополнительных ограничениях) через эллиптические функции Якоби [2]. В общем случае интегрирование этой системы весьма сложно, поэтому решение системы (1.1) обычно производится численными методами решения задачи Коши. Для получения численного решения (1.1) (в виде табличных функций) разобьем интервал интегрирования $[0, T]$ отрезками (шагом) малой длины Δt на s частей. В результате получим расчетные точки (узлы) t_0, t_1, \dots, t_s , где $t_i = i \cdot \Delta t$, $i = 0, 1, \dots, s$. Пусть в узле t_i решение известно, тогда решение в узле t_{i+1} представим первыми четырьмя членами ряда Тейлора [4]:

$$\omega_k(t_{i+1}) = \omega_k(t_i + \Delta t) = \omega_k(t_i) + \dot{\omega}_k(t_i)\Delta t + \frac{1}{2}\ddot{\omega}_k(t_i)\Delta t^2 + \frac{1}{6}\dddot{\omega}_k(t_i)\Delta t^3 \quad (1.3)$$

В точке t_0 решение определяется начальными условиями $\omega_k(0) = \omega_{k0}$.

Для реализации (1.3) помимо значений $\omega_{ki} = \omega_k(t_i)$ необходимо иметь величины $\dot{\omega}_k(t_i)$, $\ddot{\omega}_k(t_i)$ и $\ddot{\omega}_k(t_i)$, которые можно вычислить последовательно используя формулы:

$$\begin{aligned}\dot{\omega}_{ki} &= p_k \omega_{ni} \omega_{mi} + N_{ki}/I_k, & \ddot{\omega}_{ki} &= p_k (\dot{\omega}_{ni} \omega_{mi} + \omega_{ni} \dot{\omega}_{mi}) + \dot{N}_{ki}/I_k, \\ \ddot{\omega}_{ki} &= p_k (\ddot{\omega}_{ni} \omega_{mi} + 2\dot{\omega}_{ni} \dot{\omega}_{mi} + \omega_{ni} \ddot{\omega}_{mi}) + \ddot{N}_{ki}/I_k\end{aligned}\quad (1.4)$$

Величины ω_{ki} являются мгновенной информацией об абсолютном вращательном движении и по ним можно определить соответствующие интегральные характеристики вращения используя те или иные интерполяционные формулы, например, $\theta_{ki} = \int \omega_k dt \approx \theta_{ki-1} + \Delta t (\omega_{ki-1} + \omega_{ki})/2$. Величины θ_{1i} , θ_{2i} и θ_{3i} ассоциируются с углами поворотов вокруг соответствующих осей и образуют псевдовектор угла поворота $\vec{\theta}_i = \theta_{1i} \vec{e}_1 + \theta_{2i} \vec{e}_2 + \theta_{3i} \vec{e}_3$. При достаточно малых углах поворота для сложение таких векторов можно использовать правило сложение истинных векторов.

Векторы $\vec{\omega}_i = \omega_{1i} \vec{e}_1 + \omega_{2i} \vec{e}_2 + \omega_{3i} \vec{e}_3$ указывают направление оси вращения тела в момент времени t_i .

Для нахождения положения подвижной система координат относительно абсолютной системы фактически надо определить положение ортов подвижной системы координат ($\vec{e}_1, \vec{e}_2, \vec{e}_3$) относительно абсолютной системы $Oxuz$ используя ранее вычисленные значения ω_{ki} и θ_{ki} в подвижной система координат. Согласно теореме Эйлера-Даламбера, произвольный поворот твердого тела, имеющего неподвижную точку, можно осуществить одним конечным поворотом (вращением) вокруг некоторой оси, проходящей через эту точку. В нашем случае такую ось вращения задает вектор $\vec{\omega}_i$, а неподвижной точкой служит центр тяжести пластины, связанный с началом подвижной системой координат. Такой поворот (вращение) можно осуществить с помощью матриц ротации или кватернионов [5]. Кватернион представим в виде двухкомпонентного объекта, описывающего поворот в пространстве

$$\mathbf{Q} \equiv \mathbf{Q}(\varphi, \vec{n}) = \left[\cos\left(\frac{\varphi}{2}\right), \sin\left(\frac{\varphi}{2}\right) \vec{n} \right], \quad (1.5)$$

где φ – угол поворота относительно нормализованного вектора \vec{n} .

Вращение вектора \vec{e} на угол φ вокруг оси \vec{n} определяется соотношением

$$\vec{\epsilon}_k = \mathbf{Q} \cdot \vec{e}_k \cdot \bar{\mathbf{Q}}, \quad (1.6)$$

где: $\vec{\epsilon}_k$ – положение ортов \vec{e}_k в абсолютной системе координат; $\bar{\mathbf{Q}}$ – кватернион сопряженный к \mathbf{Q} . В нашем случае в точке t_i имеем ($i > 0$)

$$\begin{aligned}\vec{\epsilon}_{ki} &= \mathbf{Q}_i \cdot \vec{\epsilon}_{ki-1} \cdot \bar{\mathbf{Q}}_i, \\ \varphi_i &= \|\vec{\theta}_i - \vec{\theta}_{i-1}\|, & \vec{n}_i &= \hat{\omega}_i,\end{aligned}\quad (1.7)$$

где $\hat{\omega}_i$ – нормированный вектор $\vec{\omega}_i$.

При $i = 0$ полагаем, что $\vec{\epsilon}_{k0}$ равно базису абсолютной системе координат.

Полученные значения $\vec{\epsilon}_{ki}$ определяют положение пластины относительно абсолютной системы координат.

Для численного анализа влияния разнообразных форм функций $N_k(t)$, $\dot{N}_k(t)$ и $\ddot{N}_k(t)$ на поведение модели целесообразно использовать метод автоматического дифференцирования, основанный на усеченных гипер-дуальных чисел [6]. Выбор оптимальной величины Δt в данной работе не рассматривается.

2. Компьютерная реализация

Компьютерная реализация описанного выше подхода была выполнена на языке программирования SWIFT 5 для macOS 10.15.6. Выбор пал на эту систему по причине наличия достаточно удобной библиотеки `simd`, обеспечивающей оптимальную работу с векторами, матрицами и кватернионами [7]. Так, например, формулы (1.6) реализуются пятью короткими выражениями языка SWIFT.

Процедура `EulerCauchy()` обеспечивает решение поставленной задачи (1.1). Входной информацией служат: временной интервал $[t_0, t_n]$; количество отрезков интегрирования n ; размер пластины a и α ; начальные значения угловой скорости ω_{k0} ; указатель на функцию определения внешних вращающих моментов и их производных $N_k(t)$, которая должна быть сформулирована с использованием усеченных гипер-дуальных чисел [6]. Эта процедура возвращает массивы значений ω_{ki} и θ_{1i} в расчетных точках t_i . Процедура `RotationAxis()` осуществляет поворот подвижного репера на заданный угол. Входными параметрами служат результаты работы процедуры `EulerCauchy()`. На выходе получаем вложенный кортеж (tuple), определяющий положение подвижного репера в неподвижной системе координат.

Программный код описанных процедур и пример их использования даны в Приложении 2.

3. Численный эксперимент

Далее приводятся некоторые численные решения на основе разработанного программного обеспечения. Графические результаты расчета для пластины с размерами $a = 20$, $\alpha = 0.5$ и $t_0 = 0$, $t_n = 4.3$, $n = 50$, $\omega_{10} = 0.1$, $\omega_{20} = 5.0$, $\omega_{30} = 0$, $N_k \equiv 0$ приведены в Приложении 1 (рис. П.1, П.2 и П.3). Из рис. П.1 видно, что в точке А пластина совершает кувырок т.е. ω_2 меняет направление вращения на противоположное, а затем в точке В возвращается к исходному направлению. Рис. П.2 и П.3 отображают изменения фазовой траектории и годографа вектора угловой скорости соответственно.

Для пластины с размерами $a = 20$, $\alpha = 0.5$ и $t_0 = 0$, $t_n = 20.3$, $n = 300$, $\omega_{10} = 0.1$, $\omega_{20} = 5.0$, $\omega_{30} = 0$, $N_k \equiv 0$ результаты даны Приложении 1 (рис. П.4). С течением времени интервал между кувырками уменьшается т.е. их частота увеличивается, что соответствует видео материалам, полученным с космической станции «Салют-7». Выше полученные

численные решения следует интерпретировать как эффект Джанибекова в вакууме при отсутствии гравитации.

В реальных условиях на тело действуют возмущающие моменты внешних и внутренних сил, в частности гравитационные силы, силы сопротивления среды и внутренние диссипативные силы, что может привести к постепенной эволюции движения. Так, например, со временем вращение должно прекратиться, если учитывать определенные внешние воздействия, например, сопротивление воздуха. Учет указанных воздействий можно имитировать заданием соответствующих функций внешних вращающих моментов N_k , которые должны зависеть от ряда параметров, в том числе от размеров пластины, ее материала и угловой скорости. В качестве примера, рассмотрим самую простую линейную функцию диссипации и $N_2 = -10 \times \omega_2$ для ранее описанной пластины. Результаты расчета приведены на рис. П.5. Как видно с каждым кувырком пластины происходит уменьшение абсолютной величины ω_2 , что свидетельствует об монотонном рассеивании энергии.

Приложение 1

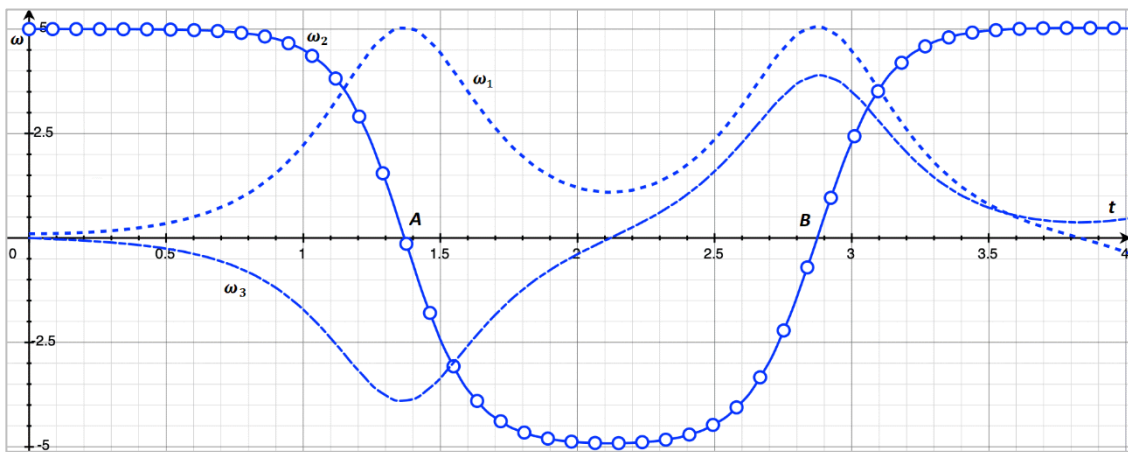


Рис. П.1. Проекции угловой скорости $\omega_1, \omega_2, \omega_3$

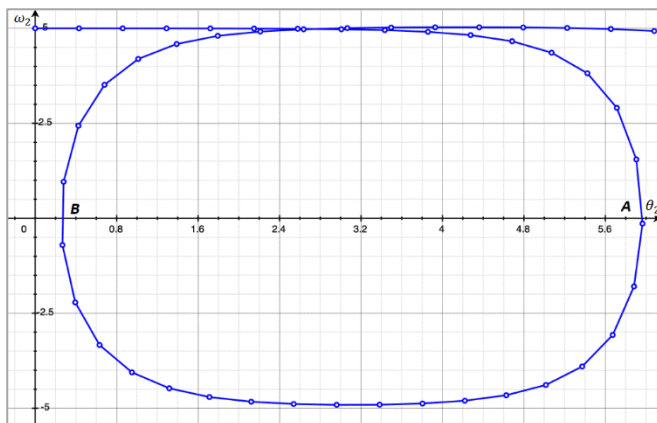


Рис. П.2. Фазовая траектория $[\theta_2, \omega_2]$

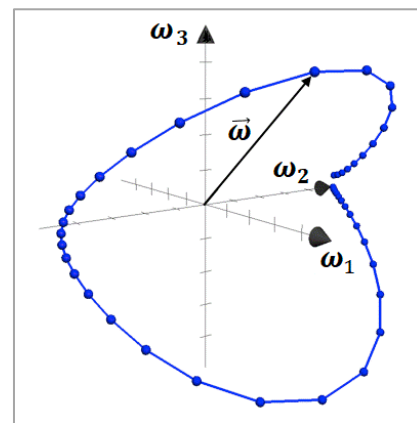


Рис. П.3. Годограф вектора угловой скорости $\vec{\omega}$

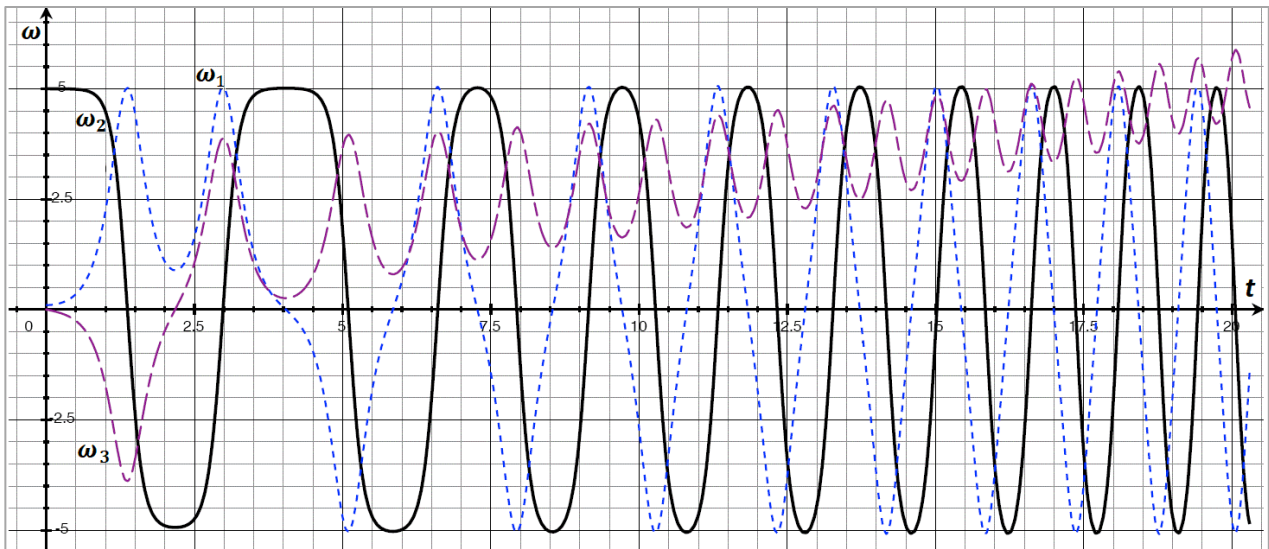


Рис. П.4. Проекция угловой скорости ω_1 , ω_2 , ω_3

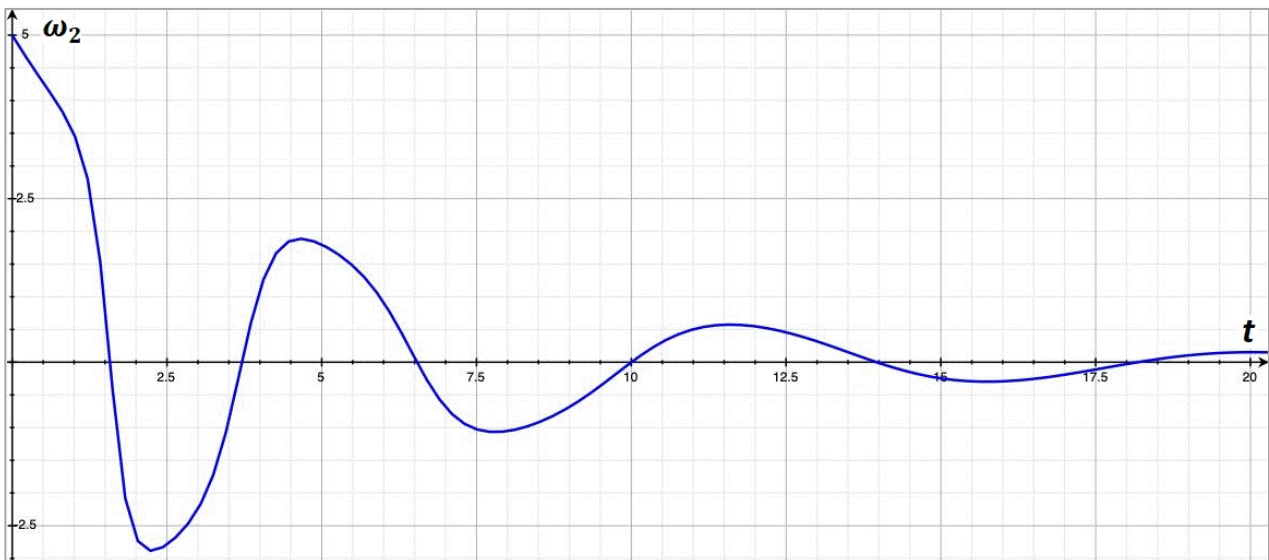


Рис. П.5. Проекция угловой скорости ω_2 при учете диссипативных сил

Приложение 2

// Для реализации приведенного ниже кода необходимо добавить файл, описывающий тип
// данных Thdn (см. [6])

```
import simd
```

```
func EulerCauchy(t0:Double, Tn: Double, n:Int, a:Double, α:Double,  
    ω10:Double, ω20:Double, ω30:Double, N_k:(Double,Double,Double,Double)  
->(N1:Thdn, N2:Thdn, N3:Thdn))->  
    ((t:Double, ω1:Double, ω2:Double, ω3:Double)],  
    [(t:Double, θ1:Double, θ2:Double, θ3:Double)]){
```

```
    func I123(a:Double, α:Double)->(I1:Double,I2:Double,I3:Double,
```

```

        p1:Double,p2:Double,p3:Double){
    let a2 = a*a, α2 = α*α, l2 = a2/12, l1 = l2*α2, l3 = l1 + a2/12.0;
    let p1 = (l2 - l3)/l1, p2 = (l3 - l1)/l2, p3 = (l1 - l2)/l3;
    return (l1:l1, l2:l2, l3:l3, p1:p1, p2:p2, p3:p3);
}

func Ω(z:(_:Double, _:Double, _:Double, _:Double))->Double{
    return z.0 + Δt*z.1 + Δt*Δt*z.2/2.0 + Δt*Δt*Δt*z.3/6.0;
}

let l = l123(a: a, α: α);
var :[(t:Double, ω1:Double, ω2:Double, ω3:Double)] = [];
var :[(t:Double, θ1:Double, θ2:Double, θ3:Double)] = [];
var ω11,ω12,ω13, ω21,ω22,ω23, ω31,ω32,ω33, ω_1,ω_2,ω_3,
    Δω1,ω2,Δω3:Double;
var last:(t:Double, ω1:Double, ω2:Double, ω3:Double);
var Nk:(N1:Thdn, N2:Thdn, N3:Thdn);
let Δt:Double = (tn - t0)/Double(n);
ω.append((t:t0, ω1:ω10, ω2:ω20, ω3:ω30));
θ.append((t:t0, θ1:0, θ2:0, θ3:0));
for t in stride(from: t0, to: tn, by: Δt){
    last = ω.last!;
    Nk = N_k(t, last.ω1, last.ω2, last.ω3);
    ω11 = l.p1*last.ω2*last.ω3 + Nk.N1.re/l.l1;
    ω21 = l.p2*last.ω3*last.ω1 + Nk.N2.re/l.l2;
    ω31 = l.p3*last.ω1*last.ω2 + Nk.N3.re/l.l3;

    ω12 = l.p1*(ω21*last.ω3 + last.ω2*ω31) + Nk.N1.im1/l.l1;
    ω22 = l.p2*(ω31*last.ω1 + last.ω3*ω11) + Nk.N2.im1/l.l2;
    ω32 = l.p3*(ω11*last.ω2 + last.ω1*ω21) + Nk.N3.im1/l.l3;

    ω13 = l.p1*(ω22*last.ω3 + 2.0*ω21*ω31 + last.ω2*ω32) + Nk.N1.im2/l.l1;
    ω23 = l.p2*(ω32*last.ω3 + 2.0*ω31*ω11 + last.ω3*ω12) + Nk.N2.im2/l.l2;
    ω33 = l.p3*(ω12*last.ω1 + 2.0*ω11*ω21 + last.ω1*ω12) + Nk.N3.im2/l.l3;

    ω_1 = (z: (last.ω1, ω11, ω12, ω13));
    ω_2 = (z: (last.ω2, ω21, ω22, ω23));
    ω_3 = (z: (last.ω3, ω31, ω32, ω33));
    ω.append((t:t + Δt, ω1:ω_1, ω2:ω_2, ω3:ω_3));

    Δω1 = 0.5*(last.ω1 + ω_1)*Δt;
    Δω2 = 0.5*(last.ω2 + ω_2)*Δt;
    Δω3 = 0.5*(last.ω3 + ω_3)*Δt;
    θ.append((t:t + Δt, θ1:θ.last!.θ1 + Δω1, θ2:θ.last!.θ2 + Δω2,
        θ3:θ.last!.θ3 + Δω3));
}
return (ω, θ);
}

func RotationAxis(ω: [(t:Double, ω1:Double, ω2:Double, ω3:Double)],
    θ: [(t:Double, θ1:Double, θ2:Double, θ3:Double)])->
    [(e1:simd_double3, e2:simd_double3, e3:simd_double3)]{

func e_123(ω1:Double, ω2:Double, ω3:Double,
    e1:simd_double3, e2:simd_double3, e3:simd_double3,
    φ:Double)->(e1:simd_double3, e2:simd_double3,e3:simd_double3){
    let n = simd_normalize(simd_double3(x: ω1, y:ω2, z: ω3));
    let Q = simd_quatd(angle: φ, axis: n);
    let i1:simd_double3 = Q.act(e1);
    let i2:simd_double3 = Q.act(e2);
}
}

```

```

    let i3:simd_double3 = Q.act(e3);
    return (e1:i1,e2:i2,e3:i3);
}

var  $\varphi$ :Double = 0.0;
var e123:[(e1:simd_double3, e2:simd_double3, e3:simd_double3)] =
    [(e1:simd_double3(x: 1,y: 0,z: 0),
      e2:simd_double3(x: 0,y: 1,z: 0),
      e3:simd_double3(x: 0,y: 0,z: 1))];
var  $\Delta\theta_1, \Delta\theta_2, \Delta\theta_3$ :Double;
for i in 1.. $\omega$ .count{
     $\Delta\theta_1$  = [i]. $\theta_1$  -  $\theta$ [i - 1]. $\theta_1$ ;
     $\Delta\theta_2$  = [i]. $\theta_2$  -  $\theta$ [i - 1]. $\theta_2$ ;
     $\Delta\theta_3$  = [i]. $\theta_3$  -  $\theta$ [i - 1]. $\theta_3$ ;
     $\varphi$  = sqrt( $\Delta\theta_1$ * $\Delta\theta_1$  +  $\Delta\theta_2$ * $\Delta\theta_2$  +  $\Delta\theta_3$ * $\Delta\theta_3$ );
    e123.append(e_123( $\omega$ 1:  $\omega$ [i]. $\omega$ 1,  $\omega$ 2:  $\omega$ [i]. $\omega$ 2,  $\omega$ 3:  $\omega$ [i]. $\omega$ 3,
      e1: e123.last!.e1, e2: e123.last!.e2, e3: e123.last!.e3,  $\varphi$ :  $\varphi$ ));
}
return e123;
}

// Пример использования процедур EulerCauchy() и RotationAxis()
func N_k(t:Double  $\omega$ 1:Double,  $\omega$ 2:Double,  $\omega$ 3:Double)->(N1:Thdn, N2:Thdn, N3:Thdn){
    let T = Thdn(re: t, im1: 1, im2: 0);
    let N1 = Thdn(re: 0), N2 = Thdn(re: 0), N3 = Thdn(re: 0);
    return (N1, N2, N3);
}

var WQ = EulerCauchy(t0: 0.0, tn: 4.3, n: 50, a: 20.0,  $\alpha$ : 0.5,
     $\omega$ 10: 0.1,  $\omega$ 20: 5,  $\omega$ 30: 0.0, N_k: N_k);

Let E123 = RotationAxis( $\omega$ : WQ.0,  $\theta$ : WQ.1);

// учет диссипативных сил
func N_kd(t:Double,  $\omega$ 1:Double,  $\omega$ 2:Double,  $\omega$ 3:Double)->(N1:Thdn, N2:Thdn, N3:Thdn){
    let T = Thdn(re: t, im1: 1, im2: 0);
    let N1:Thdn = Thdn(re: 0),N2:Thdn = Thdn(re: -10.0* $\omega$ 2), Thdn = Thdn(re: 0);
    return (N1, N2, N3);
}

WQ = EulerCauchy(t0: 0.0, tn: 20.3, n: 100, a: 20.0,  $\alpha$ : 0.5,
     $\omega$ 10: 0.1,  $\omega$ 20: 5,  $\omega$ 30: 0.0, N_k: N_kd);

```

Литература

1. Эффект Джанибекова [*Veritasium*] – <https://www.youtube.com/watch?v=N9HIQ-XXVnFk>
2. Черноусько Ф. Л., Акуленко Л. Д., Лещенко Д. Д. Эволюция движений твердого тела относительно центра масс. – М.– Ижевск: Ижевский институт компьютерных исследований, 2015, – 308 с.
3. Сивухин Д. В. Общий курс физики. Т.1. Механика. 3-е изд. –М.: Наука, 1989. § 53.
4. Авдюшев В. А. Численные методы интегрирования обыкновенных дифференциальных

уравнений. Томск: Изд-во ТГУ, 2009, 52с.

5. Гордеев В. Н. Кватернионы и бикватернионы с приложениями в геометрии и механике. – Киев: СТАЛЬ, 2016 – 315 с.
6. Олифер В. И. Усеченные гипер-дуальные числа в автоматическом дифференцировании. – http://viosolutions.amerihomesrealty.com/pdf/Усеченные_гипер-дуальные_числа_в_автоматическом_дифференцировании.pdf
7. *Quaternions*. – <https://developer.apple.com/documentation/accelerate/simd>

Абстракт

В данной публикации рассматривается численное моделирование эффекта Джанибекова известного также как эффект теннисной ракетки или теорема о промежуточной оси. Основой служат динамические уравнения Эйлера, решение которых осуществляется методом Коши с использованием ряда Тейлора. Для возможности учета различных форм функций внешних вращающих моментов применен метод автоматического дифференцирования, основанный на усеченных гипер-дуальных чисел. Представлена компьютерная реализация описанной модели для языка SWIFT операционной системы macOS на основе которой проведены численные эксперименты.

Ключевые слова: эффект Джанибекова, эффект теннисной ракетки, теорема о промежуточной оси, численное решение задачи Коши, усеченные гипер-дуальные числа, автоматическое дифференцирование, *Dzhanibekov effect, tennis racket effect, intermediate axis theorem, numerical solution of the Cauchy problem, truncated hyper-dual numbers, automatic differentiation*

7 сентября 2020 г.