Метод отдельных тел

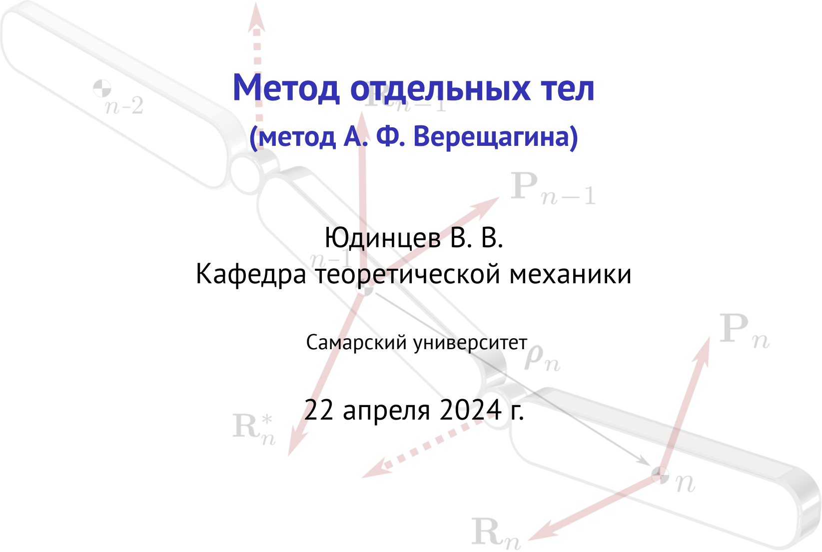
(метод А. Ф. Верещагина)

Юдинцев В. В.

Кафедра теоретической механики

Самарский университет

22 апреля 2024 г.

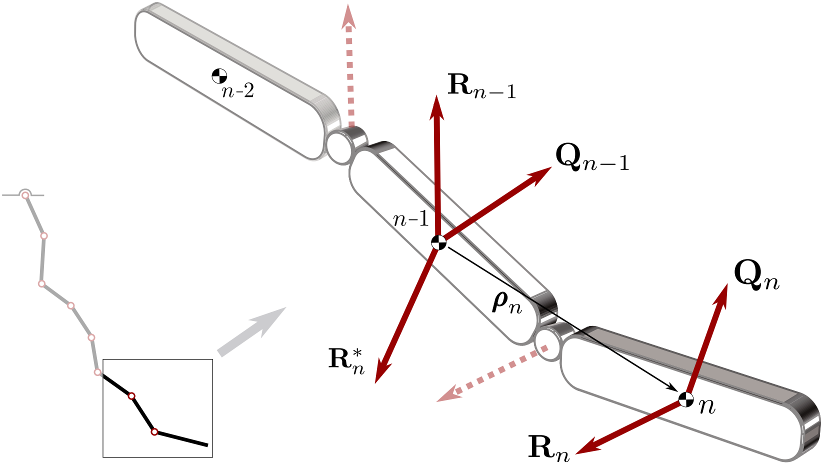


Метод отдельных тел

- Метод разработан Верещагиным А. Ф. в 1974 году:
  Верещагин А. Ф. Компьютерное моделирование динамики сложных механизмов роботов-манипуляторов Инженерная кибернетика, вып. 6, 1974, с. 65-70.
- Используются шарнирные координаты.
- Не формируется матрица масс всей системы: все матричные операции выполняются с матрицами и векторами размерности ≤ 6.
- Вычислительная трудоёмкость пропорциональна количеству тел системы: $O(n)$.

Уравнения движения

Уравнения движения



Уравнения движения тела n

$$\boxed{\mathbf{M}_n \mathbf{a}_n = \mathbf{Q}_n + \mathbf{R}_n} \tag{1}$$

где $\mathbf{M}_n$ – матрица инерции тела:

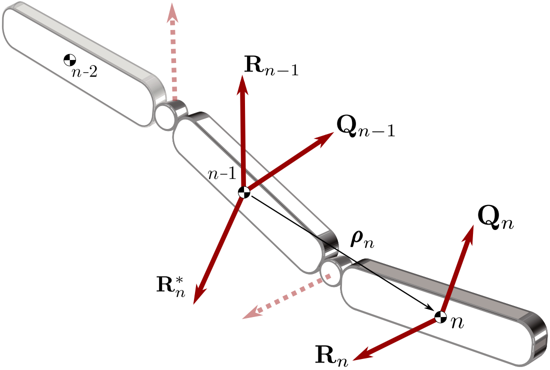
$$\mathbf{M}_n = \begin{bmatrix} m_n \mathbf{E}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{J}_c \end{bmatrix}; \tag{2}$$

$\mathbf{a}_n$ – столбец линейных и угловых ускорений тела n, $\mathbf{Q}_n$ – столбец активных сил и моментов, действующих на тело n, $\mathbf{R}_n$ – столбец сил и моментов реакции, действующих на тело n:

$$\mathbf{a}_n = \begin{bmatrix} \mathbf{a}_n \\ \boldsymbol{\varepsilon}_n \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{r}}_n \\ \boldsymbol{\varepsilon}_n \end{bmatrix}, \qquad \mathbf{Q}_n = \begin{bmatrix} \mathbf{F}_n \\ \mathbf{M}_n - \boldsymbol{\omega}_n \times \mathbf{J}_n \cdot \boldsymbol{\omega}_n \end{bmatrix}, \quad \mathbf{R}_n = \begin{bmatrix} \mathbf{F}_n^c \\ \mathbf{M}_n^c \end{bmatrix}.$$

Уравнения движения тела $n-1$

$$\boxed{\mathbf{M}_{n-1}\mathbf{a}_{n-1} = \mathbf{Q}_{n-1} + \mathbf{R}_{n-1} + \mathbf{R}_n^*} \quad (3)$$



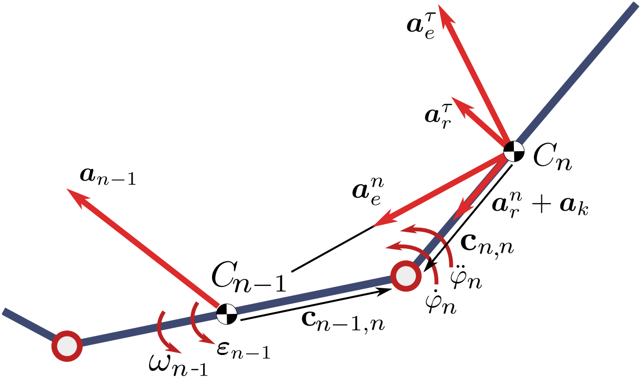
$\mathbf{R}_n^*$ – столбец сил и моментов реакции, приведённых к центру масс тела $n-1$:

$$\mathbf{R}_n^* = \begin{bmatrix} \mathbf{F}_n^c \\ \mathbf{M}_n^c + \boldsymbol{\rho}_n \times \mathbf{F}_n^c \end{bmatrix}.$$

Кинематика относительного движения

Ускорение

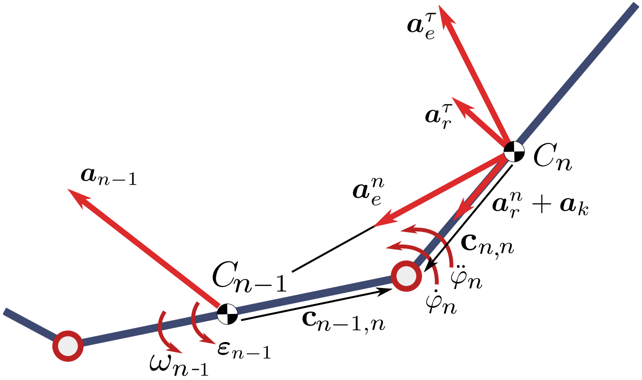
$$\boxed{\mathbf{a}_n = \mathbf{C}_n \mathbf{a}_{n-1} + \mathbf{S}_n \ddot{\mathbf{q}}_n + \mathbf{a}'_n} \tag{4}$$



Матрица C_n

$$\mathbf{a}_n = \mathbf{C}_n \mathbf{a}_{n-1} + \mathbf{S}_n \ddot{\mathbf{q}}_n + \mathbf{a}'_n$$

$$\mathbf{C}_n = \begin{bmatrix} \mathbf{E} & -\boldsymbol{\rho}_n \times \\ \mathbf{0} & \mathbf{E} \end{bmatrix} \to \begin{bmatrix} \mathbf{E} & -\boldsymbol{\rho}_n \times \\ \mathbf{0} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{r}}_{n-1} \\ \boldsymbol{\varepsilon}_{n-1} \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{r}}_{n-1} + \boldsymbol{\varepsilon}_{n-1} \times \boldsymbol{\rho}_n \\ \boldsymbol{\varepsilon}_{n-1} \end{bmatrix} \quad (5)$$



Шарнирное ускорение

Подставив в уравнение движения $\mathbf{a}_n$ и умножив результата на $\mathbf{S}_n^T$

$$\mathbf{S}_n^T \cdot \boxed{\mathbf{M}_n \mathbf{a}_n = \mathbf{Q}_n + \mathbf{R}_n} \leftarrow \mathbf{a}_n = \ldots \mathbf{q}_n, \tag{6}$$

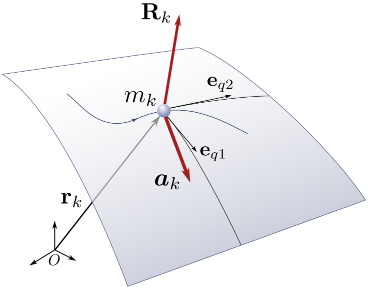
получим:

$$\mathbf{S}_n^T \mathbf{M}_n (\mathbf{C}_n \mathbf{a}_{n-1} + \mathbf{S}_n \ddot{\mathbf{q}}_n + \mathbf{a}'_n) = \mathbf{S}_n^T \mathbf{Q}_n + \mathbf{S}_n^T \mathbf{R}_n. \tag{7}$$

или

$$\ddot{\mathbf{q}}_n = (\underbrace{\mathbf{S}_n^T \mathbf{M}_n \mathbf{S}_n}_{\mathbf{U}_n})^{-1} \mathbf{S}_n^T (\mathbf{Q}_n + \mathbf{R}_n - \mathbf{M}_n (\mathbf{C}_n \mathbf{a}_{n-1} + \mathbf{a}'_n)) \tag{8}$$

Механическая система с идеальными связями

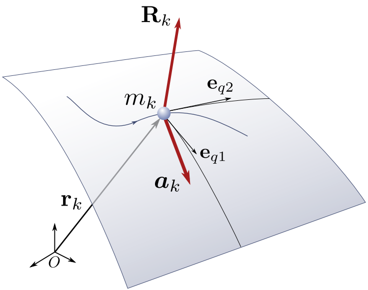


$$\mathbf{r}_k = \mathbf{r}_k(q_1, q_2, \ldots, q_n) = \begin{bmatrix} x_k(q_1, q_2, \ldots, q_n) \\ y_k(q_1, q_2, \ldots, q_n) \\ z_k(q_1, q_2, \ldots, q_n) \end{bmatrix}$$

$$\dot{\mathbf{r}}_k = \begin{bmatrix} \partial x_k/\partial q_1 & \partial x_k/\partial q_2 & \ldots & \partial x_k/\partial q_n \\ \partial y_k/\partial q_1 & \partial y_k/\partial q_2 & \ldots & \partial y_k/\partial q_n \\ \partial z_k/\partial q_1 & \partial z_k/\partial q_2 & \ldots & \partial z_k/\partial q_n \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} + \frac{\partial \mathbf{r}_k}{\partial t} = \mathbf{S}_n \dot{\mathbf{q}} + \frac{\partial \mathbf{r}_k}{\partial t}$$

$$\ddot{\mathbf{r}}_k = \mathbf{S}_n \ddot{\mathbf{q}} + \ldots$$

Матрица S_n и идеальные связи



$$\mathbf{S}_k = \begin{bmatrix} \partial x_k/\partial q_1 & \partial x_k/\partial q_2 & \dots & \partial x_k/\partial q_n \\ \partial y_k/\partial q_1 & \partial y_k/\partial q_2 & \dots & \partial y_k/\partial q_n \\ \partial z_k/\partial q_1 & \partial z_k/\partial q_2 & \dots & \partial z_k/\partial q_n \end{bmatrix}$$

Каждый столбец матрицы $\mathbf{S}_k$ – базисный вектор криволинейного базиса, касательного пространства:

$$\mathbf{S}_k = \begin{bmatrix} \mathbf{e}_{q1} & \mathbf{e}_{q2} & \dots & \mathbf{e}_{qn} \end{bmatrix}$$

Для идеальных связей

$$\mathbf{R}_k \cdot \mathbf{e}_{q1} = 0,\ \mathbf{R}_k \cdot \mathbf{e}_{q2} = 0,\ \dots, \mathbf{R}_k \cdot \mathbf{e}_{qn} = 0$$

или

$$\mathbf{S}_k^T \mathbf{R}_k = 0.$$

Обобщённые ускорения

С учётом

$$\mathbf{S}_n^T \mathbf{R}_n = \mathbf{0} \tag{9}$$

Выражение для вторых производных обобщённых координат

$$\ddot{\mathbf{q}}_n = (\underbrace{\mathbf{S}_n^T \mathbf{M}_n \mathbf{S}_n}_{\mathbf{U}_n})^{-1} \mathbf{S}_n^T (\mathbf{Q}_n + \mathbf{R}_n - \mathbf{M}_n(\mathbf{C}_n \mathbf{a}_{n-1} + \mathbf{a}'_n)) \tag{10}$$

принимает вид:

$$\boxed{\ddot{\mathbf{q}}_n = \mathbf{U}_n^{-1} \mathbf{S}_n^T \left[\mathbf{Q}_n - \mathbf{M}_n(\mathbf{C}_n \mathbf{a}_{n-1} + \mathbf{a}'_n) \right]} \tag{11}$$

где

$$\mathbf{U}_n = \mathbf{S}_n^T \mathbf{M}_n \mathbf{S}_n. \tag{12}$$

Обратный ход алгоритма метода отдельных тел

$$\ddot{\mathbf{q}}_n = \mathbf{U}_n^{-1}\mathbf{S}_n^T\left[\mathbf{Q}_n - \mathbf{M}_n(\mathbf{C}_n\mathbf{a}_{n-1} + \mathbf{a}'_n)\right], \tag{13}$$

$$\mathbf{a}_n = \mathbf{C}_n\mathbf{a}_{n-1} + \mathbf{S}_n\ddot{\mathbf{q}}_n + \mathbf{a}'_n, \tag{14}$$

$$\mathbf{M}_n\mathbf{a}_n = \mathbf{Q}_n + \mathbf{R}_n, \tag{15}$$

$$\mathbf{M}_{n-1}\mathbf{a}_{n-1} = \mathbf{Q}_{n-1} + \mathbf{R}_{n-1} + \mathbf{R}^*_n. \tag{16}$$

$$\ddot{\mathbf{q}}_n \to \mathbf{a}_n(\ddot{\mathbf{q}}_n) \to \boxed{\mathbf{m}_n\mathbf{a}_n = \ldots} \to \mathbf{R}_n(\mathbf{a}_{n-1}) \to \boxed{\mathbf{m}_{n-1}\mathbf{a}_{n-1} = \ldots}$$

$$\mathbf{M}^*_{n-1}\mathbf{a}_{n-1} = \mathbf{Q}^*_{n-1} + \mathbf{R}_{n-1}, \quad n \to k$$

где

$$\mathbf{M}^*_{n-1} = \mathbf{M}_{n-1} + \mathbf{C}_n^T\mathbf{M}_n\mathbf{C}_n - \mathbf{C}_n^T\mathbf{M}_n\mathbf{S}_n\mathbf{U}_n^{-1}\mathbf{S}_n^T\mathbf{M}_n\mathbf{C}_n,$$

$$\mathbf{Q}^*_{n-1} = \mathbf{Q}_{n-1} + \mathbf{C}_n^T\left(\mathbf{M}_n\left(\mathbf{S}_n\mathbf{U}_n^{-1}\mathbf{S}_n^T(\mathbf{Q}_n - \mathbf{M}_n\mathbf{a}'_n) + \mathbf{a}'_n\right) - \mathbf{Q}_n\right).$$

Алгоритм

Обратный ход алгоритма ($k = n, n-1, n-2, \ldots, 2$)

$$\mathbf{M}^*_{k-1} = \mathbf{M}_{k-1} + \mathbf{C}^T_k \mathbf{M}^*_k \mathbf{C}_k - \mathbf{C}^T_k \mathbf{M}^*_k \mathbf{S}_k \mathbf{U}^{-1*}_k \mathbf{S}^T_k \mathbf{M}^*_k \mathbf{C}_k, \tag{17}$$

$$\mathbf{Q}^*_{k-1} = \mathbf{Q}_{k-1} + \mathbf{C}^T_k \left(\mathbf{M}^*_k \left(\mathbf{S}_k \mathbf{U}^{-1*}_k \mathbf{S}^T_k (\mathbf{Q}^*_k - \mathbf{M}^*_k \mathbf{a}'_k) + \mathbf{a}'_k \right) - \mathbf{Q}^*_k \right), \tag{18}$$

$$\mathbf{U}^*_k = \mathbf{S}^T_k \mathbf{M}^*_k \mathbf{S}_k. \tag{19}$$

$$\boxed{\mathbf{M}_n, \mathbf{Q}_n \to \mathbf{M}^*_{n-1}, \mathbf{Q}^*_{n-1} \to \mathbf{M}^*_{n-2}, \mathbf{Q}^*_{n-2} \to \ldots \to \mathbf{M}^*_1, \mathbf{Q}^*_1}$$

Прямой ход алгоритма ($k = 1, 2, 3, \ldots, n$)

$$\ddot{\mathbf{q}}_k = \mathbf{U}^{*-1}_k \mathbf{S}^T_k \left(\mathbf{Q}^*_k - \mathbf{M}^*_k (\mathbf{C}_k \mathbf{a}_{k-1} + \mathbf{a}'_k) \right) \tag{20}$$

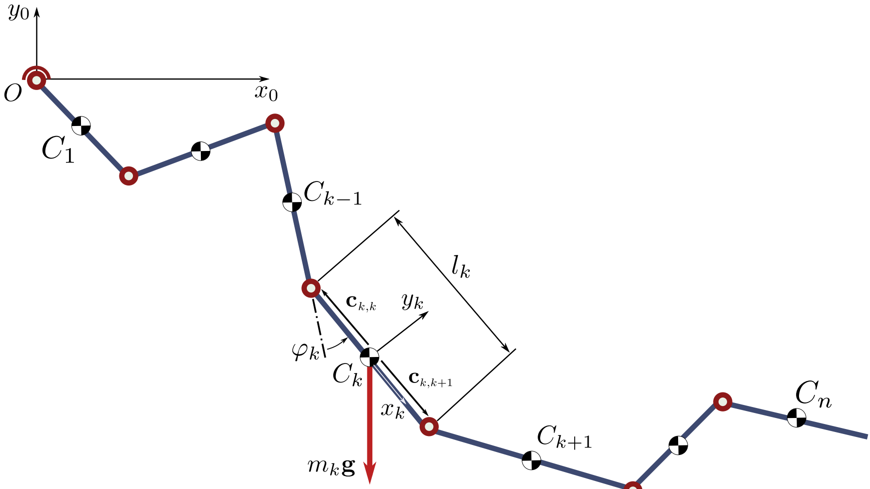
$$\mathbf{a}_k = \mathbf{C}_k \mathbf{a}_{k-1} + \mathbf{S}_k \ddot{\mathbf{q}}_k + \mathbf{a}'_k. \tag{21}$$

$$\boxed{\ddot{\mathbf{q}}_1(\mathbf{w}_0), \mathbf{w}_1 \to \ddot{\mathbf{q}}_2(\mathbf{a}_1), \mathbf{a}_2, \to \ldots \to \ddot{\mathbf{q}}_n(\mathbf{a}_{n-1}), \mathbf{a}_n}$$

Пример

Схема системы

Плоская система стержней, последовательно соединённых цилиндрическими шарнирами



Структура файлов

`main.m`

`dqdt.m`

`getSk.m` — $\mathbf{S}_k$

`getCk.m` — $\mathbf{C}_k$

`getWprim.m` — $\mathbf{a}'_k$

`getMkQkiUk.m` — $\mathbf{M}^*_k, \mathbf{Q}^*_k, \mathbf{U}_k$

`getA.m` — $\mathbf{A}_k$

Главный файл-скрипт I

```
global Model;
Model=struct;
```

Количество тел

```
n=10; Model.n=n;
```

Шарнирные векторы: $\mathbf{c}_{k,k+1}^{(k)} = \begin{bmatrix} l_k/2 \\ 0 \end{bmatrix} = \begin{bmatrix} l/2 \\ 0 \end{bmatrix}$, $\mathbf{c}_{k,k}^{(k)} = -\begin{bmatrix} l_k/2 \\ 0 \end{bmatrix} = -\begin{bmatrix} l/2 \\ 0 \end{bmatrix}$

```
c=mat2cell(zeros(n*2,n),ones(1,n)*2,ones(1,n));
for i=1:n
    c{i,i}=0.5*[-1;0]/n;
    if i~=n
        c{i,i+1}=0.5*[+1;0]/n;
    end
end
Model.c=c;
```

Главный файл-скрипт II

Матрицы масс

```
mass=cell(n,1); m=10;
for i=1:n
    mass{i}=[m/n 0 0; 0 m/n 0; 0 0 m/(12*n*n*n)];
end
Model.mass=mass;
```

Начальные условия

```
q0=zeros(n*2,1); q0(1)=-1.0;
```

Интегрирование

```
[t,q]=ode113(@dqdt,[0 10],q0);
```

Файл-функция правых частей ДУ I

$$[\dot{\varphi}_1, \dot{\varphi}_2, \ldots, \dot{\varphi}_n, \ddot{\varphi}_1, \ddot{\varphi}_2, \ldots, \ddot{\varphi}_n] = \mathsf{dqdt}(t, [\varphi_1, \varphi_2, \ldots, \varphi_n, \dot{\varphi}_1, \dot{\varphi}_2, \ldots, \dot{\varphi}_n])$$

```
function dx = dqdt(t,x)
  global Model;
  n=size(x,1)/2;
  q =x(1:n);
  dq=x(n+1:2*n);
  dx=[dq;dq];
```

Матрицы преобразования координат $\mathbf{A}_i$ из базиса тела i в базис 0:

```
  A0=cell(n,1);
  A0{1}=getA(q(1));
  for i=2:n
    A0{i}=A0{i-1}*getA(q(i));
  end
```

Файл-функция правых частей ДУ II

Силы и моменты

```
Q=cell(n,1);
for i=1:n
  Q{i}=[0;-Model.mass{i}(1,1)*9.81;0];
end
```

Обратный ход алгоритма.
Вычисление матриц $\mathbf{M}_k^*, \mathbf{Q}_k^*, \mathbf{U}_k^{-1}$

```
[Mk,Qk,iUk]=getMkQkiUk(q,dq,Model.mass,Model.c,Q,A0);
```

Файл-функция правых частей ДУ III

Прямой ход алгоритма. Вычисление $\ddot{\mathbf{q}}_k, \mathbf{a}_k$.

```
  w=[0;0;0];
  for i=1:n
    Ck=getCk(i, q, dq, Model.c, A0);
    Sk=getSk(i, q, dq, Model.c, A0);
    Wp=getWprim(i, q, dq, Model.c, A0);
    dx(n+i)=iUk{i}*Sk'*(Qk{i}-Mk{i}*(Ck*w+Wp));
    w=Ck*w+Sk*dx(n+i)+Wp;
  end
end
```

Функция getMkQkiUk (обратный ход алгоритма) I

```
function [Mk,Qk,iUk] = getMkQkiUk(q,dq,masses,c,Q,A0)
 n=size(q,1);
 Mk=cell(n,1);
 Qk=cell(n,1);
 iUk=cell(n,1);
 Mk{n}=masses{n};
 Qk{n}=Q{n};
```

Функция getMkQkiUk (обратный ход алгоритма) II

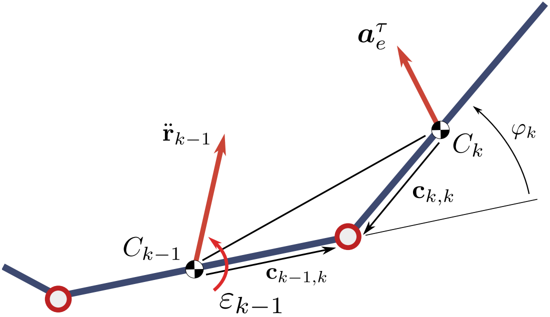
```
  for k=n:-1:1
   Ck=getCk(k,q,dq,c,A0);
   Sk=getSk(k,q,dq,c,A0);
   Wprim=getWprim(k, q, dq, c, A0);
   U=Sk'*Mk{k}*Sk;
   iUk{k}=inv(U);
   if k>1
    Mk{k-1}=masses{k-1}+Ck'*Mk{k}*Ck-Ck'*Mk{k}*...
                Sk*iUk{k}*Sk'*Mk{k}*Ck;
    Qk{k-1}=Q{k-1}-Ck'*(Mk{k}*(Sk*iUk{k}*Sk'*...
                (Qk{k}-Mk{k}*Wprim)+Wprim)-Qk{k});
   end
  end % for

end
```

Файл-функция getCk

Коэффициенты при ускорениях $\mathbf{a}_{k-1}$ в выражении

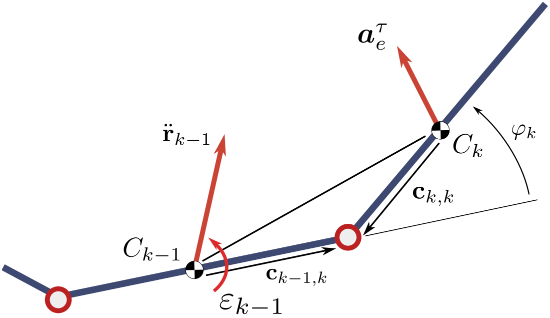
$$\mathbf{a}_k = \mathbf{C}_k \mathbf{a}_{k-1} + \mathbf{S}_k \ddot{\mathbf{q}}_k + \mathbf{a}'_k \qquad (22)$$



$$\mathbf{a}_k = \begin{bmatrix} \ddot{x}_{k-1} \\ \ddot{y}_{k-1} \\ \varepsilon_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{-\pi/2}(\mathbf{A}_k \mathbf{c}_{kk}^{(k)} - \mathbf{A}_{k-1} \mathbf{c}_{k,k-1}^{(k-1)}) \\ 0 \end{bmatrix} \varepsilon_{k-1} + \ldots$$

Файл-функция getCk

$$\mathbf{a}_k = \underbrace{\begin{bmatrix} \mathbf{E}_{2\times 2} & \mathbf{A}_{-\pi/2}(\mathbf{A}_k \mathbf{c}_{kk}^{(k)} - \mathbf{A}_{k-1}\mathbf{c}_{k,k-1}^{(k-1)}) \\ 0 & 1 \end{bmatrix}}_{\mathbf{C}_k} \begin{bmatrix} \ddot{x}_{k-1} \\ \ddot{y}_{k-1} \\ \varepsilon_{k-1} \end{bmatrix} + \ldots$$

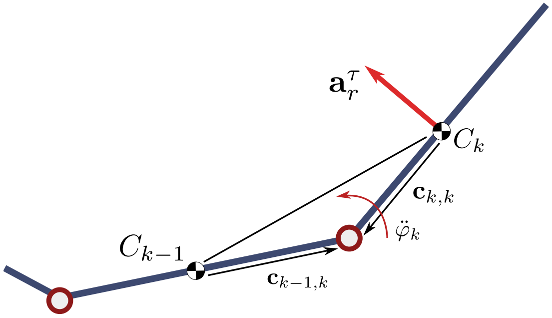


Файл-функция getCk

```
function ck = getCk(k, q, dq, c, A0)
 ck=eye(3,3);
  if k==1
   ck(1:2,3)=-[0 1;-1 0]*(-A0{k}*c{k,k});
  else
   ck(1:2,3)=-[0 1;-1 0]*(A0{k-1}*c{k-1,k}-A0{k}*c{k,k});
end
```

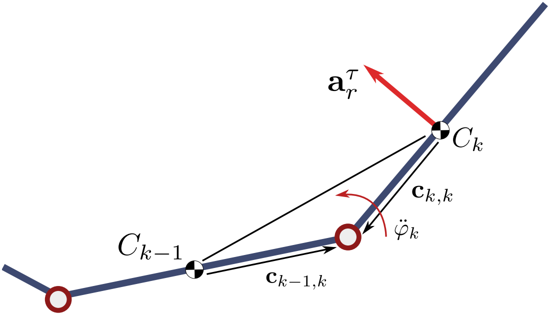
Файл-функция getSk

$$\mathbf{a}_k = \mathbf{C}_k \mathbf{a}_{k-1} + \mathbf{S}_k \ddot{\mathbf{q}}_k + \mathbf{a}'_k \tag{23}$$



Файл-функция getSk

$$\mathbf{S}_k \ddot{\mathbf{q}}_k = \mathbf{S}_k \ddot{\varphi}_k = \begin{bmatrix} a^{\tau}_{r_x} \\ a^{\tau}_{r_y} \\ \ddot{\varphi}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{-\pi/2} \mathbf{A}_k \mathbf{c}_{kk} \\ 1 \end{bmatrix} \ddot{\varphi}_k. \tag{24}$$



Файл-функция getSk

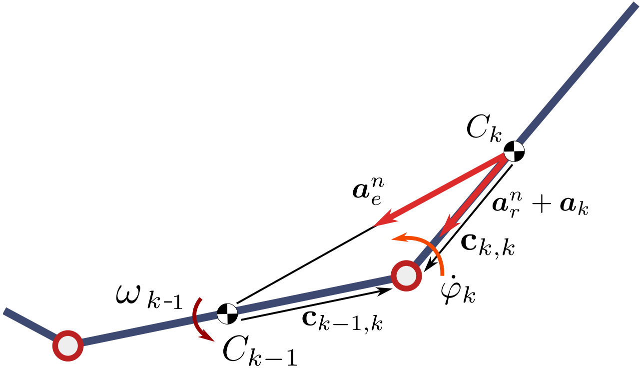
$$\mathbf{S}_k = \begin{bmatrix} \mathbf{A}_{-\pi/2}\mathbf{A}_k\mathbf{c}_{kk} \\ 1 \end{bmatrix}. \tag{25}$$

```
function sk = getSk(k, q, dq, c, A0)
  sk=[0;0;1];
  sk(1:2)=[0 1;-1 0]*A0{k}*c{k,k};
end
```

Файл-функция getWprim

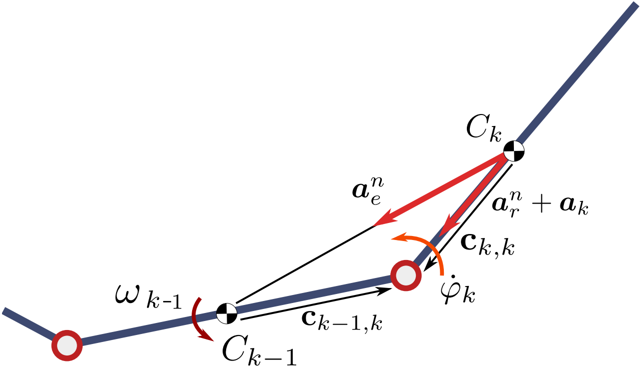
Составляющие ускорения, не зависящие от вторых производных обобщённых координат $\ddot{\mathbf{q}}$.

$$\mathbf{a}_k = \mathbf{C}_k \mathbf{a}_{k-1} + \mathbf{S}_k \ddot{\mathbf{q}}_k + \mathbf{a}'_k \tag{26}$$



Файл-функция getWprim

$$\mathbf{a}'_k = \underbrace{-\mathbf{c}_{k-1,k}\omega_{k-1}^2 + \mathbf{c}_{kk}\omega_{k-1}^2}_{\mathbf{a}_e^n} + \underbrace{2\mathbf{c}_{kk}\dot{\varphi}_k\omega_{k-1}}_{\mathbf{a}_k} + \underbrace{\mathbf{c}_{kk}\dot{\varphi}_k^2}_{\mathbf{a}_r^n} \quad (27)$$



Файл-функция getWprim

$$\mathbf{a}'_k = -\mathbf{c}_{k-1,k}\omega^2_{k-1} + \mathbf{c}_{kk}\omega^2_{k-1} + 2\mathbf{c}_{kk}\dot{\varphi}_k\omega_{k-1} + \mathbf{c}_{kk}\dot{\varphi}^2_k. \qquad (28)$$

```
function  wprim=getWprim(k, q, dq, c, A0)
 if k==1
    wprim=[(dq(k)+0)^2*A0{k}*c{k,k};0];
 else
    wkp=sum(dq(1:k-1));
    wprim=[(dq(k)+wkp)^2*A0{k}*c{k,k}-wkp*wkp*A0{k-1}*c{k-1,k};0];
 end
end
```

Файл-функция getA

Матрица плоского поворота, преобразующая координаты из базиса $\mathbf{e}^2$, повёрнутого вокруг оси z относительно базиса $\mathbf{e}^1$, в базис $\mathbf{e}^1$.

```
function A = getA(angle)
  A=[cos(angle) -sin(angle);sin(angle) cos(angle)];
end
```

Список использованных источников

1. *Верещагин, А. Ф.* Компьютерное моделирование динамики сложных механизмов роботов-манипуляторов / А. Ф. Верещагин // Инженерная кибернетика, вып. 6. — 1974. — С. 65–70.
2. *Дмитроченко, О. Н.* Эффективные методы численного моделирования динамики нелинейных систем абсолютно твёрдых и деформируемых тел: Дис... канд. физ. мат. наук: 01.02.01. – М., 2003. – 125 с.
3. *Joel Storch and Stephen Gates*, Motivating Kane's Method for Obtaining Equations of Motion for Dynamic System. Engineering Notes, Vol. 12, N. 4, July-August 1989.