

Метод отдельных тел

Динамика твердого тела и систем тел

Юдинцев В. В.

Кафедра теоретической механики

28 апреля 2026 г.



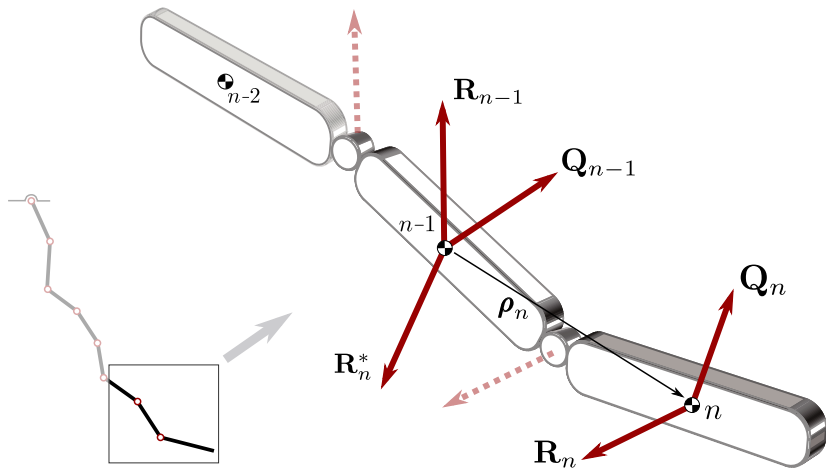
САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

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

$$\boxed{\mathbf{M}_n \mathbf{a}_n = \mathbf{Q}_n + \mathbf{R}_n} \quad (1)$$

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

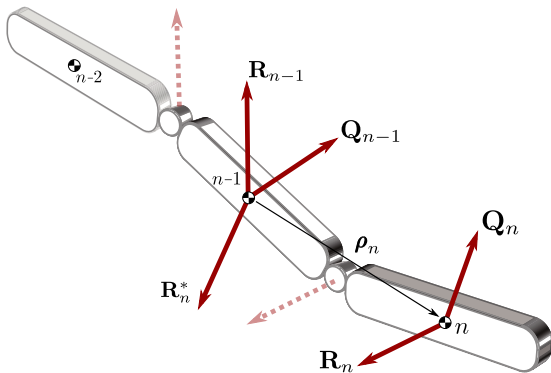
$$\mathbf{M}_n = \begin{bmatrix} m_n \mathbf{E}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{J}_c \end{bmatrix}; \quad (2)$$

\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 :

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

Уравнения движения тела $n - 1$

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



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

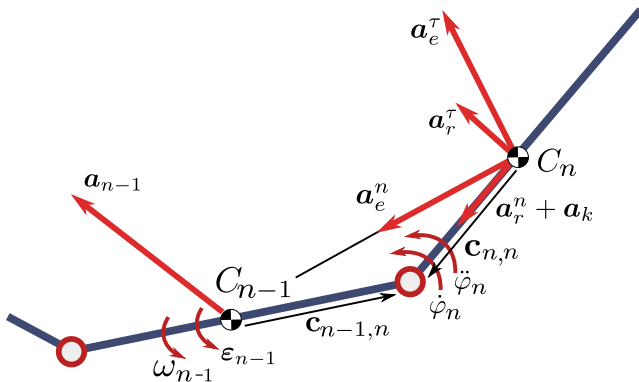
$$\mathbf{R}_n^* = \begin{bmatrix} \mathbf{F}_n^c \\ \mathbf{M}_n^c + \rho_n \times \mathbf{F}_n^c \end{bmatrix}.$$

Кинематика относительного движения

Ускорение

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

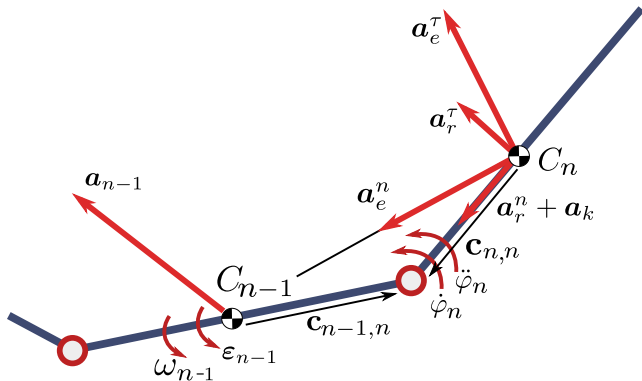
(4)



Матрица C_n

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

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



Шарнирное ускорение

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

$$\mathbf{S}_n^T \cdot \boxed{\mathbf{M}_n \mathbf{a}_n = \mathbf{Q}_n + \mathbf{R}_n} \leftarrow \mathbf{a}_n = \dots \mathbf{q}_n, \quad (6)$$

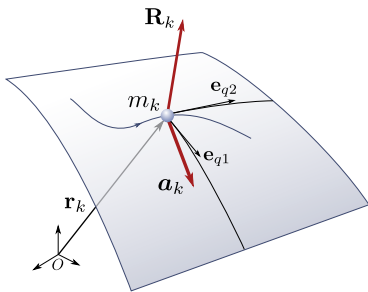
получим:

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

или

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

Механическая система с идеальными связями

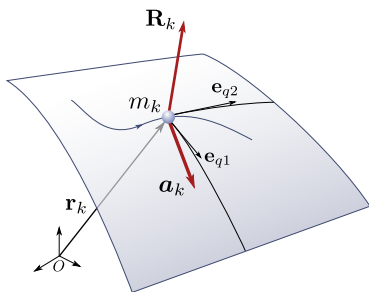


$$\mathbf{r}_k = \mathbf{r}_k(q_1, q_2, \dots, q_n) = \begin{bmatrix} x_k(q_1, q_2, \dots, q_n) \\ y_k(q_1, q_2, \dots, q_n) \\ z_k(q_1, q_2, \dots, q_n) \end{bmatrix}$$

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

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

Матрица S_n и идеальные связи



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

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

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

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

$$\mathbf{S}_k = [\mathbf{e}_{q1} \quad \mathbf{e}_{q2} \quad \dots \quad \mathbf{e}_{qn}]$$

Обобщённые ускорения

С учётом

$$\mathbf{S}_n^T \mathbf{R}_n = \mathbf{0} \quad (9)$$

Выражение для вторых производных обобщённых координат

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

принимает вид:

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

где

$$\mathbf{U}_n = \mathbf{S}_n^T \mathbf{M}_n \mathbf{S}_n. \quad (12)$$

Обратный ход алгоритма метода отдельных тел

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

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

$$\mathbf{M}_n \mathbf{a}_n = \mathbf{Q}_n + \mathbf{R}_n, \quad (15)$$

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

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

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

где

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

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

Алгоритм

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

$$\mathbf{M}_{k-1}^* = \mathbf{M}_{k-1} + \mathbf{C}_k^T \mathbf{M}_k^* \mathbf{C}_k - \mathbf{C}_k^T \mathbf{M}_k^* \mathbf{S}_k \mathbf{U}_k^{-1} \mathbf{S}_k^T \mathbf{M}_k^* \mathbf{C}_k, \quad (17)$$

$$\mathbf{Q}_{k-1}^* = \mathbf{Q}_{k-1} + \mathbf{C}_k^T (\mathbf{M}_k^* (\mathbf{S}_k \mathbf{U}_k^{-1} \mathbf{S}_k^T (\mathbf{Q}_k^* - \mathbf{M}_k^* \mathbf{a}'_k) + \mathbf{a}'_k) - \mathbf{Q}_k^*), \quad (18)$$

$$\mathbf{U}_k^* = \mathbf{S}_k^T \mathbf{M}_k^* \mathbf{S}_k. \quad (19)$$

$$\mathbf{M}_n, \mathbf{Q}_n \rightarrow \mathbf{M}_{n-1}^*, \mathbf{Q}_{n-1}^* \rightarrow \mathbf{M}_{n-2}^*, \mathbf{Q}_{n-2}^* \rightarrow \dots \rightarrow \mathbf{M}_1^*, \mathbf{Q}_1^*$$

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

$$\ddot{\mathbf{q}}_k = \mathbf{U}_k^{*-1} \mathbf{S}_k^T (\mathbf{Q}_k^* - \mathbf{M}_k^* (\mathbf{C}_k \mathbf{a}_{k-1} + \mathbf{a}'_k)) \quad (20)$$

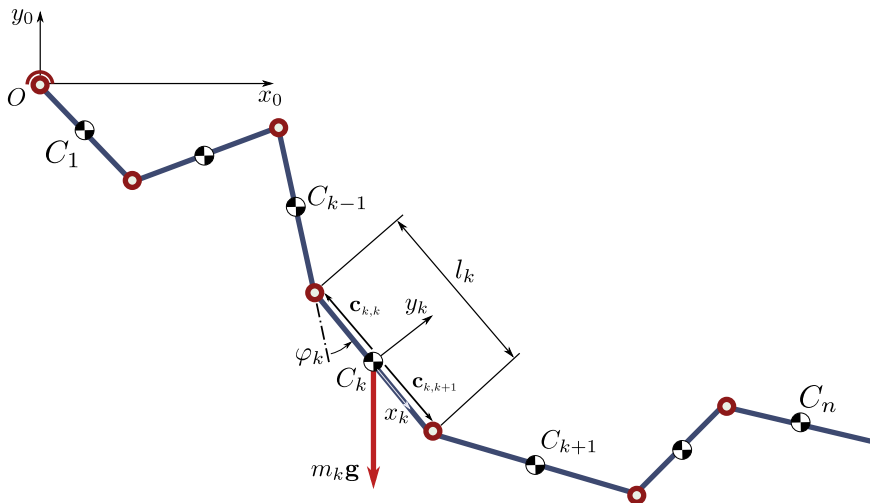
$$\mathbf{a}_k = \mathbf{C}_k \mathbf{a}_{k-1} + \mathbf{S}_k \ddot{\mathbf{q}}_k + \mathbf{a}'_k. \quad (21)$$

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

Пример

Схема системы

Плоская система стержней, последовательно соединённых цилиндрическими шарнирами



Структура файлов



Главный файл-скрипт I

```
1 | global Model ;  
2 | Model=struct ;
```

Количество тел

```
1 | n=10; Model.n=n;
```

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

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

Главный файл-скрипт II

Матрицы масс

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

Начальные условия

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

Интегрирование

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

Файл-функция правых частей ДУ I

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

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

Матрицы преобразования координат A_i из базиса тела i в базис 0:

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

Файл-функция правых частей ДУ II

Силы и моменты

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

Обратный ход алгоритма.

Вычисление матриц M_k^* , Q_k^* , U_k^{-1}

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

Файл-функция правых частей ДУ III

Прямой ход алгоритма. Вычисление \ddot{q}_k, a_k .

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

Функция getMkQkiUk (обратный ход алгоритма) I

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

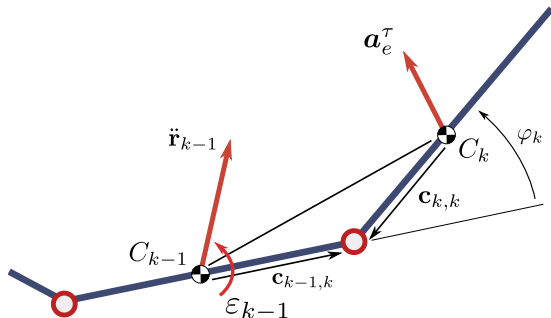
Функция getMkQkiUk (обратный ход алгоритма) II

```
1 for k=n:-1:1
2   Ck=getCk(k,q,dq,c,A0);
3   Sk=getSk(k,q,dq,c,A0);
4   Wprim=getWprim(k,q,dq,c,A0);
5   U=Sk'*Mk{k}*Sk;
6   iUk{k}=inv(U);
7   if k>1
8     Mk{k-1}=masses{k-1}+Ck'*Mk{k}*Ck-Ck'*Mk{k}*...
9             Sk*iUk{k}*Sk'*Mk{k}*Ck;
10    Qk{k-1}=Qk{k-1}-Ck'*(Mk{k}*(Sk*iUk{k}*Sk'*...
11            (Qk{k}-Mk{k}*Wprim)+Wprim)-Qk{k});
12  end
13 end % for
14
15 end
```

Файл-функция getCk

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

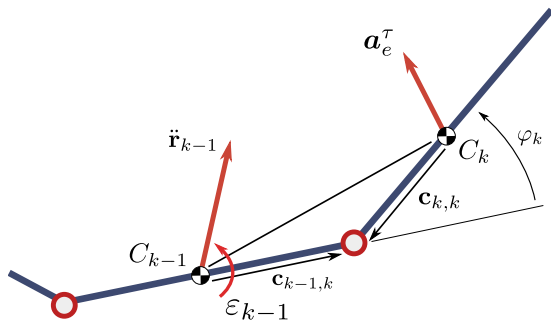
$$\mathbf{a}_k = \mathbf{C}_k \mathbf{a}_{k-1} + \mathbf{S}_k \ddot{\mathbf{q}}_k + \mathbf{a}'_k \quad (22)$$



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

Файл-функция getCk

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

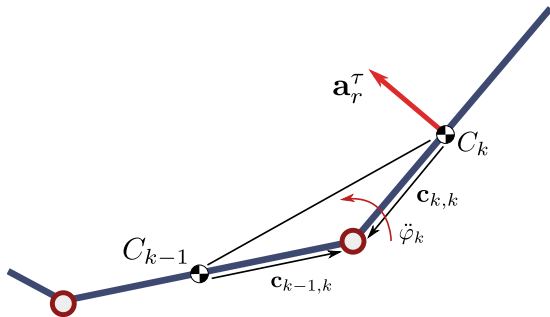


Файл-функция getCk

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

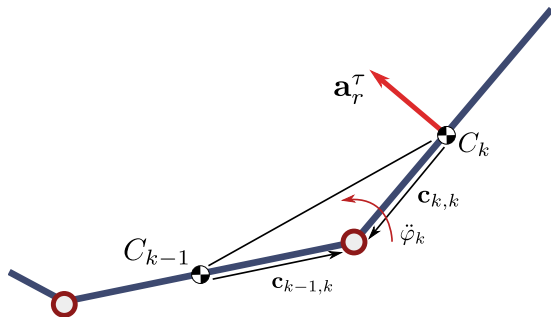
Файл-функция getSk

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



Файл-функция getSk

$$\mathbf{S}_k \ddot{\mathbf{q}}_k = \mathbf{S}_k \ddot{\varphi}_k = \begin{bmatrix} a_{rx}^\tau \\ a_{ry}^\tau \\ \ddot{\varphi}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{-\pi/2} \mathbf{A}_k \mathbf{c}_{kk} \\ 1 \end{bmatrix} \ddot{\varphi}_k. \quad (24)$$



Файл-функция getSk

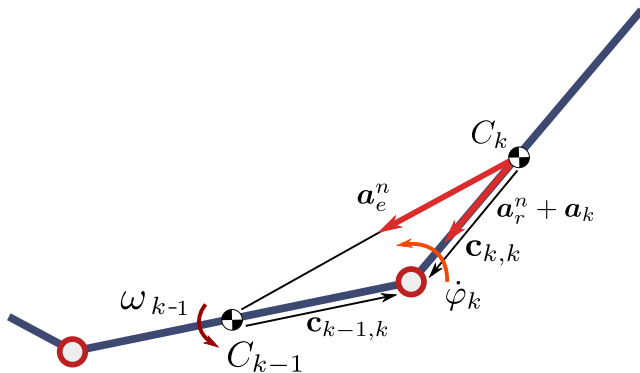
$$\mathbf{S}_k = \begin{bmatrix} \mathbf{A}_{-\pi/2} \mathbf{A}_k \mathbf{c}_{kk} \\ 1 \end{bmatrix}. \quad (25)$$

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

Файл-функция `getWprim`

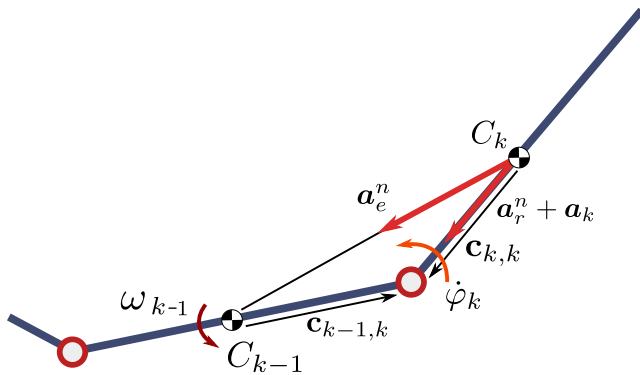
Составляющие ускорения, не зависящие от вторых производных обобщённых координат \ddot{q} .

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



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

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

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

Файл-функция getA

Матрица плоского поворота, преобразующая координаты из базиса e^2 , повернутого вокруг оси z относительно базиса e^1 , в базис e^1 .

```
1 function A = getA(angle)
2     A=[cos(angle) -sin(angle); sin(angle) cos(angle)];
3 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 *Дмитrochenко, 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.