

# ОБ ИСПОЛЬЗОВАНИИ СКМ «МАТЕМАТИКА» В КАЧЕСТВЕ ПРОГРАММНОЙ СРЕДЫ ДЛЯ МОДЕЛИРОВАНИЯ ДВИЖЕНИЙ АНТРОПОМОРФНОГО РОБОТА

Чигарев А.В.<sup>1</sup>, Борисов А.В.<sup>2</sup>

Белорусский национальный технический университет, г. Минск<sup>1</sup>  
Филиал ГОУ ВПО «Московский энергетический институт  
(Технический университет)», г. Смоленск<sup>2</sup>

*In the article the technique of the automated compiling of differential equations of motion of the anthropomorphic robot is reviewed. The technique of activity with the results of numerical calculations in the system of computer mathematics "Mathematica", its animation capabilities and the export of the results in the generally accepted format of saving and reviewing of the video information "«.gif" is described.*

В настоящее время приходится делать сложные численные расчеты и аналитические преобразования, для описания механических систем с большим числом степеней свободы, сложными геометрическими связями, наличием деформируемости и т.д. Применение систем компьютерной позволило делать численные расчеты, которые ранее были практически недоступны ученым. Громоздкие аналитические преобразования можно провести во много раз быстрее и эффективнее, представив результаты в удобном виде с помощью различных систем компьютерной математики, и, в частности, «Mathematica»<sup>1</sup>. В данной статье рассмотрен один из вариантов таких аналитических преобразований на примере ходьбы антропоморфного механизма. Этот метод является универсальным и его можно применять для описания других механических систем.

Рассмотрим приближенную к человеку одиннадцатизвенную антропоморфную модель. Для рассмотрения плоского движения биомеханической системы в одноопорной фазе введем неподвижную правую декартову систему координат  $xuz$  с началом в точке  $O$  и плоскостью  $xu$ , в которой происходит движение центра масс. Система имеет две трехзвенные весомые ноги, две двухзвенные весомые руки и весомый корпус. Предполагаем, что правые и левые конечности являются одинаковыми. Все элементы структуры являются жесткими. На рис. 1 схематично изображен механизм, и введены соответствующие обозначения [1,4].

Пусть  $O_1A_1 = O_2A_2 = l_1$ ,  $A_1B_1 = A_2B_2 = l_2$ ,  $B_1C = B_2C = l_3$ ,  $CG = l_7$ ,  $DE_1 = DE_2 = l_8$ ,  $E_1F_1 = E_2F_2 = l_9$  – длины звеньев биомеханической системы, моделирующей человека. Положение в одноопорной фазе однозначно определяется углами  $\varphi_i$  ( $i = 1, \dots, 11$ ), поэтому рассматриваемая система имеет одиннадцать степеней свободы. Обозначим через  $M_i$  моменты, развиваемые в  $i$ -том шарнире ( $i = 1, \dots, 11$ ).

Центры масс находятся в точках:  $C_1$  – стопы опорной ноги,  $C_2$  – голени опорной ноги,  $C_3$  – бедра опорной ноги,  $C_4$  – бедра переносной ноги,  $C_5$  – голени переносной ноги,  $C_6$  – стопы переносной ноги,  $C_7$  – корпуса,  $C_8, C_{10}$  – плеч,  $C_9, C_{11}$  – предплечий. Их положения будем задавать в виде отношений длины от начала соответствующего звена до центра масс ко всей длине звена, через множители  $n_i$ , ( $i = 1, \dots, 11$ ), ( $0 < n_i < 1$ ) (если все звенья перенумеровать по номерам индексов у соответствующих углов). Такой способ задания положений центров масс предпочтительней, так как в биомеханике человека положения центров масс конечно-

<sup>1</sup> Лицензионная версия данной программы приобретена за счет средств Гранта Президента Российской Федерации, номер МК-2524.2008.1.

стей определяются эмпирическим путем и задаются в процентном отношении одной части звена к другой.

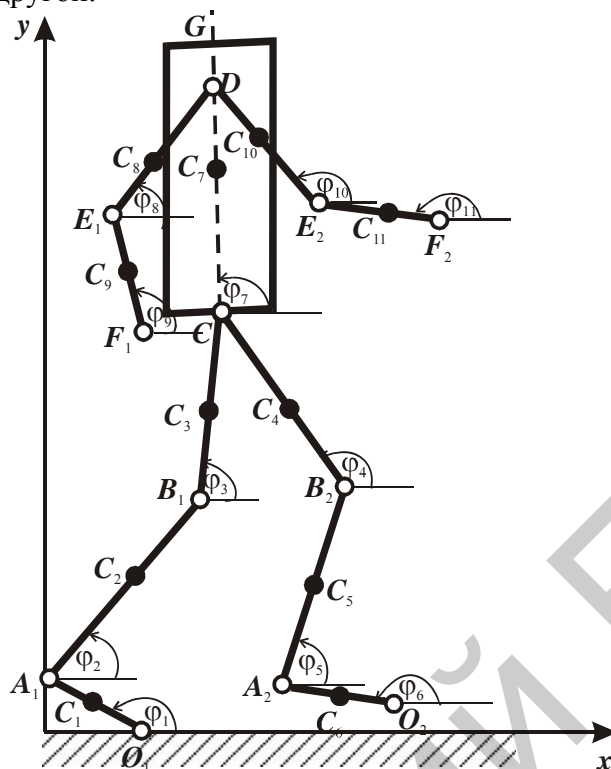


Рис. 1. Модель плоского движения стержневой механической системы с изменяемой геометрией звеньев в одноопорной фазе

Массы:  $m_1, m_2, m_3, m_8, m_9, m_7$  – масса стопы, голени, бедра, плеч, предплечий и корпуса, соответственно. Моменты инерции:  $I_1, I_2, I_3, I_8, I_9, I_7$  – стопы, голени, бедра, плеч, предплечий и корпуса, соответственно. Правую и левую конечности механизма будем считать одинаковыми.

Составление дифференциальных уравнений движения происходит следующим образом. Все строим в векторно-матричной форме. Справочную информацию об используемых командах программы «Mathematica» можно найти в работах [2,3].

Вводим вектор обобщенных координат (команды, вводимые в программе «Mathematica» будем выделять моноширинным шрифтом), для удобства обозначим индексы 10 и 11 через 8p и 9p:

```
q = {{φ1[t]}, {φ2[t]}, {φ3[t]}, {φ4[t]}, {φ5[t]}, {φ6[t]},
{φ7[t]}, {φ8[t]}, {φ9[t]}, {φ8p[t]}, {φ9p[t]}}
```

Формируем вектор угловых скоростей, дифференцируя вектор обобщенных координат:

```
qv = D[q, t]
```

Формируем таблицу производных по времени от обобщенных координат, необходимую для составления функции Лагранжа.

```
DS[T_, q_] := {Table[D[T, q[[i, 1]]], {i, 1, Length[q]}]}
```

Формируем таблицу производных по времени от производных обобщенных координат, необходимую для составления функции Лагранжа.

```
DV[X_, q_] := Table[D[X[[i, 1]], q[[j, 1]]], {i, 1, Length[X]},
{j, 1, Length[q]}
```

Вводим координаты центров масс звеньев механизма

```
C1 = {{l1*n1*Cos[φ1[t]], {l1*n1*Sin[φ1[t]]}};
```

```

C2 = {{11*Cos[φ1[t]] + 12*n2*Cos[φ2[t]], {11*Sin[φ1[t]] +
12*n2*Sin[φ2[t]]}};
C3 = {{11*Cos[φ1[t]] + 12*Cos[φ2[t]] + 13*n3*Cos[φ3[t]],
{11*Sin[φ1[t]] + 12*Sin[φ2[t]] + 13*n3*Sin[φ3[t]]}};
C4 = {{11*Cos[φ1[t]] + 12*Cos[φ2[t]] + 13*Cos[φ3[t]] + 13*(1 -
n3)*Cos[π + φ4[t]], {11*Sin[φ1[t]] + 12*Sin[φ2[t]] + 13*Sin[φ3[t]] +
13*(1 - n3)*Sin[π + φ4[t]]}};
C5 = {{11*Cos[φ1[t]] + 12*Cos[φ2[t]] + 13*Cos[φ3[t]] + 13*Cos[π
+ φ4[t]] + 12*(1 - n2)*Cos[π + φ5[t]], {11*Sin[φ1[t]] + 12*Sin[φ2[t]]
+ 13*Sin[φ3[t]] + 13*Sin[π + φ4[t]] + 12*(1 - n2)*Sin[π + φ5[t]]}};
C6 = {{11*Cos[φ1[t]] + 12*Cos[φ2[t]] + 13*Cos[φ3[t]] + 13*Cos[π
+ φ4[t]] + 12*Cos[π + φ5[t]] + 11*(1 - n1)*Cos[π + φ6[t]],
{11*Sin[φ1[t]] + 12*Sin[φ2[t]] + 13*Sin[φ3[t]] + 13*Sin[π + φ4[t]] +
12*Sin[π + φ5[t]] + 11*(1 - n1)*Sin[π + φ6[t]]}};
C7 = {{11*Cos[φ1[t]] + 12*Cos[φ2[t]] + 13*Cos[φ3[t]] +
17*n7*Cos[φ7[t]], {11*Sin[φ1[t]] + 12*Sin[φ2[t]] + 13*Sin[φ3[t]] +
17*n7*Sin[φ7[t]]}};
C8 = {{11*Cos[φ1[t]] + 12*Cos[φ2[t]] + 13*Cos[φ3[t]] +
17*n7R*Cos[φ7[t]] + 18*(1 - n8)*Cos[π + φ8[t]], {11*Sin[φ1[t]] +
12*Sin[φ2[t]] + 13*Sin[φ3[t]] + 17*n7R*Sin[φ7[t]] + 18*(1 - n8)*Sin[π
+ φ8[t]]}};
C9 = {{11*Cos[φ1[t]] + 12*Cos[φ2[t]] + 13*Cos[φ3[t]] +
17*n7R*Cos[φ7[t]] + 18*Cos[π + φ8[t]] + 19*(1 - n9)*Cos[π + φ9[t]],
{11*Sin[φ1[t]] + 12*Sin[φ2[t]] + 13*Sin[φ3[t]] + 17*n7R*Sin[φ7[t]] +
18*Sin[π + φ8[t]] + 19*(1 - n9)*Sin[π + φ9[t]]}};
C8p = {{11*Cos[φ1[t]] + 12*Cos[φ2[t]] + 13*Cos[φ3[t]] +
17*n7R*Cos[φ7[t]] + 18*(1 - n8)*Cos[π + φ8p[t]], {11*Sin[φ1[t]] +
12*Sin[φ2[t]] + 13*Sin[φ3[t]] + 17*n7R*Sin[φ7[t]] + 18*(1 - n8)*Sin[π
+ φ8p[t]]}};
C9p = {{11*Cos[φ1[t]] + 12*Cos[φ2[t]] + 13*Cos[φ3[t]] +
17*n7R*Cos[φ7[t]] + 18*Cos[π + φ8p[t]] + 19*(1 - n9)*Cos[π + φ9p[t]],
{11*Sin[φ1[t]] + 12*Sin[φ2[t]] + 13*Sin[φ3[t]] + 17*n7R*Sin[φ7[t]] +
18*Sin[π + φ8p[t]] + 19*(1 - n9)*Sin[π + φ9p[t]]}};

```

**Вычисляем скорости центров масс:**

```

vC1 = D[C1, t];
vC2 = D[C2, t];
vC3 = D[C3, t];
vC4 = D[C4, t];
vC5 = D[C5, t];
vC6 = D[C6, t];
vC7 = D[C7, t];
vC8 = D[C8, t];
vC9 = D[C9, t];
vC8p = D[C8p, t];
vC9p = D[C9p, t];

```

**Определяем квадраты скоростей и сразу упрощаем их:**

```
Simplify[Transpose[vC1].vC1]
```

**Находим кинетическую энергию механизма и сразу упрощаем ее:**

```

T = 1/2*m1*(Transpose[vC1].vC1) + 1/2*m2*(Transpose[vC2].vC2) +
1/2*m3*(Transpose[vC3].vC3) + 1/2*m3*(Transpose[vC4].vC4) +
1/2*m2*(Transpose[vC5].vC5) + 1/2*m1*(Transpose[vC6].vC6) +
1/2*m7*(Transpose[vC7].vC7) + 1/2*m8*(Transpose[vC8].vC8) +
1/2*m9*(Transpose[vC9].vC9) + 1/2*m8*(Transpose[vC8p].vC8p) +
1/2*m9*(Transpose[vC9p].vC9p);
T = Simplify[T][[1, 1]] + 1/2*I1*D[φ1[t], t]*D[φ1[t], t] +
1/2*I2*D[φ2[t], t]*D[φ2[t], t] + 1/2*I3*D[φ3[t], t]*D[φ3[t], t] +
1/2*I3*D[φ4[t], t]*D[φ4[t], t] + 1/2*I2*D[φ5[t], t]*D[φ5[t], t] +
1/2*I1*D[φ6[t], t]*D[φ6[t], t] + 1/2*I7*D[φ7[t], t]*D[φ7[t], t] +
1/2*I8*D[φ8[t], t]*D[φ8[t], t] + 1/2*I9*D[φ9[t], t]*D[φ9[t], t] +
1/2*I8*D[φ8p[t], t]*D[φ8p[t], t] + 1/2*I9*D[φ9p[t], t]*D[φ9p[t], t];

```

Записываем потенциальную энергию:

$$\Pi = m1 * g * C1[[2, 1]] + m2 * g * C2[[2, 1]] + m3 * g * C3[[2, 1]] + m3 * g * C4[[2, 1]] + m2 * g * C5[[2, 1]] + m1 * g * C6[[2, 1]] + m7 * g * C7[[2, 1]] + m8 * g * C8[[2, 1]] + m9 * g * C9[[2, 1]] + m8 * g * C8p[[2, 1]] + m9 * g * C9p[[2, 1]];$$

Составляем функцию Лагранжа:

$$L = T - \Pi;$$

Записываем уравнения движения в общем виде:

$$Eq = \text{Simplify}[\text{Transpose}[D[DS[L, qv], t] - DS[L, q]]];$$

Выписываем левые части уравнений:

$$\begin{aligned} E1 &= Eq[[1, 1]]; \\ E2 &= Eq[[2, 1]]; \\ E3 &= Eq[[3, 1]]; \\ E4 &= Eq[[4, 1]]; \\ E5 &= Eq[[5, 1]]; \\ E6 &= Eq[[6, 1]]; \\ E7 &= Eq[[7, 1]]; \\ E8 &= Eq[[8, 1]]; \\ E9 &= Eq[[9, 1]]; \\ E8p &= Eq[[10, 1]]; \\ E9p &= Eq[[11, 1]]; \end{aligned}$$

Записываем уравнения с правыми частями:

$$\begin{aligned} Eqn1 &= (E1 - (M1[t] - M2[t])); \\ Eqn2 &= (E2 - (M2[t] - M3[t])); \\ Eqn3 &= (E3 - (M3[t] + M4[t] + M7[t])); \\ Eqn4 &= (E4 - (M4[t] - M5[t])); \\ Eqn5 &= (E5 - (M5[t] - M6[t])); \\ Eqn6 &= (E6 - M6[t]); \\ Eqn7 &= (E7 - (M3[t] + M4[t])); \\ Eqn8 &= (E8 - (M8[t] - M9[t])); \\ Eqn9 &= (E9 - M9[t]); \\ Eqn8p &= (E8p - (M8p[t] - M9p[t])); \\ Eqn9p &= (E9p - M9p[t]); \end{aligned}$$

Окончательно, приравняв их к нулю, получаем уравнения динамики движения антропоморфного механизма.

Уравнения получаются слишком громоздкими. Чтобы привести их к компактному виду, сначала группируем по членам, содержащим первые и вторые производные обобщенных координат. При этом каждый раз, для удобства обращения к промежуточным результатам, уравнению присваиваем новое название, добавляя в конец имени единицу. Так как команды однотипны, а уравнения получены, дальнейшие преобразования будем выписывать только для первого уравнения:

$$Eqn11 = \text{Collect}[Eqn1, \{\varphi1''[t], \varphi2''[t], \varphi3''[t], \varphi4''[t], \varphi5''[t], \varphi6''[t], \varphi7''[t], \varphi8''[t], \varphi9''[t], \varphi8p''[t], \varphi9p''[t], \varphi1'[t]^2, \varphi2'[t]^2, \varphi3'[t]^2, \varphi4'[t]^2, \varphi5'[t]^2, \varphi6'[t]^2, \varphi7'[t]^2, \varphi8'[t]^2, \varphi9'[t]^2, \varphi8p'[t]^2, \varphi9p'[t]^2\}];$$

Затем упрощаем с помощью команды:

$$Eqn111 = \text{Simplify}[Eqn11].$$

В итоге, приравняв к нулю левую часть, получаем систему нелинейных дифференциальных уравнений, описывающих динамику антропоморфного робота. Приведем в качестве примера левую часть первого уравнения после группировки и упрощения:

$$g \ l1 \ m1 \ \text{Cos}[\varphi1[t]] + 2 \ g \ l1 \ m2 \ \text{Cos}[\varphi1[t]] + 2 \ g \ l1 \ m3 \ \text{Cos}[\varphi1[t]] + g \ l1 \ m7 \ \text{Cos}[\varphi1[t]] + 2 \ g \ l1 \ m8 \ \text{Cos}[\varphi1[t]] + 2 \ g \ l1 \ m9 \ \text{Cos}[\varphi1[t]] + g \ l1 \ m1 \ n1 \ \text{Cos}[\varphi1[t]] - M1[t] + M2[t] + l1 \ l2 \ (m1 + m2 + 2 \ m3 + m7 + 2 \ m8 + 2 \ m9 + m2 \ n2) \ \text{Sin}[\varphi1[t] - \varphi2[t]] \ \varphi2'[t]^2 + l1 \ l3 \ (m1 + m2 + m3 + m7$$

$+ 2 m_8 + 2 m_9 + m_3 n_3) \sin[\varphi_1[t] - \varphi_3[t]] \varphi_3'[t]^2 - 11 13 (m_1 + m_2 + m_3 - m_3 n_3) \sin[\varphi_1[t] - \varphi_4[t]] \varphi_4'[t]^2 - 11 12 (m_1 + m_2 - m_2 n_2) \sin[\varphi_1[t] - \varphi_5[t]] \varphi_5'[t]^2 + 11^2 m_1 (-1 + n_1) \sin[\varphi_1[t] - \varphi_6[t]] \varphi_6'[t]^2 + 11 17 (m_7 n_7 + 2 (m_8 + m_9) n_7 R) \sin[\varphi_1[t] - \varphi_7[t]] \varphi_7'[t]^2 + 11 18 (-m_8 - m_9 + m_8 n_8) \sin[\varphi_1[t] - \varphi_8[t]] \varphi_8'[t]^2 + 11 18 (-m_8 - m_9 + m_8 n_8) \sin[\varphi_1[t] - \varphi_{8p}[t]] \varphi_{8p}'[t]^2 + 11 19 m_9 (-1 + n_9) \sin[\varphi_1[t] - \varphi_9[t]] \varphi_9'[t]^2 + 11 19 m_9 (-1 + n_9) \sin[\varphi_1[t] - \varphi_{9p}[t]] \varphi_{9p}'[t]^2 + (I_1 + 11^2 (m_1 + 2 m_2 + 2 m_3 + m_7 + 2 m_8 + 2 m_9 + m_1 n_1^2)) \varphi_1''[t] + 11 12 (m_1 + m_2 + 2 m_3 + m_7 + 2 m_8 + 2 m_9 + m_2 n_2) \cos[\varphi_1[t] - \varphi_2[t]] \varphi_2''[t] + 11 13 (m_1 + m_2 + m_3 + m_7 + 2 m_8 + 2 m_9 + m_3 n_3) \cos[\varphi_1[t] - \varphi_3[t]] \varphi_3''[t] - 11 13 (m_1 + m_2 + m_3 - m_3 n_3) \cos[\varphi_1[t] - \varphi_4[t]] \varphi_4''[t] - 11 12 (m_1 + m_2 - m_2 n_2) \cos[\varphi_1[t] - \varphi_5[t]] \varphi_5''[t] + 11^2 m_1 (-1 + n_1) \cos[\varphi_1[t] - \varphi_6[t]] \varphi_6''[t] + 11 17 (m_7 n_7 + 2 (m_8 + m_9) n_7 R) \cos[\varphi_1[t] - \varphi_7[t]] \varphi_7''[t] + 11 18 (-m_8 - m_9 + m_8 n_8) \cos[\varphi_1[t] - \varphi_8[t]] \varphi_8''[t]^2 + 11 18 (-m_8 - m_9 + m_8 n_8) \cos[\varphi_1[t] - \varphi_{8p}[t]] \varphi_{8p}''[t]^2 + 11 19 m_9 (-1 + n_9) \cos[\varphi_1[t] - \varphi_9[t]] \varphi_9''[t]^2 + 11 19 m_9 (-1 + n_9) \cos[\varphi_1[t] - \varphi_{9p}[t]] \varphi_{9p}''[t]$

Таким образом, описана методика составления дифференциальных уравнений движения модели антропоморфного робота.

Описанная методика является универсальной и может применяться для составления уравнений динамики разнообразных антропоморфных механизмов, с большим или меньшим количеством звеньев, может учитывать различия между правыми и левыми конечностями аппарата, наличие деформаций звеньев.

После того, как уравнения движения составлены, в СКМ «Mathematica», возникает необходимость их решения. Так как полученная система обладает большим количеством неизвестных и при этом существенно нелинейна, решить аналитически ее не представляется возможным, да и необходимости нет. Поэтому проводим ее решение численное решение.

Для того, чтобы решать численно, необходимо все параметры, входящие в уравнения, кроме неизвестных функций, определить численно. Сформируем набор констант, характеризующих длины, массы, моменты инерции, коэффициенты, определяющие положения центров масс звеньев антропоморфного робота и ускорение свободного падения. Все их зададим в системе «СИ» и все дальнейшие вычисления будем вести в этой системе единиц:

```

ss = {m1 -> 0.98, m2 -> 3.1, m3 -> 10.24, m7 -> 36.38, I1 -> 0.01, I2 -> 0.08, I3 -> 0.43, I7 -> 4.61, g -> 9.81, l1 -> 0.26, l2 -> 0.50, l3 -> 0.51, l7 -> 0.76, n1 -> 0.5585, n2 -> 0.5951, n3 -> 0.5451, n7 -> 0.5000, m8 -> 1.95, I8 -> 0.027, l8 -> 0.3, n8 -> 0.5502, m9 -> 1.16, I9 -> 0.013, l9 -> 0.25, n9 -> 0.5726, n7R -> 0.7, mM -> 36.38 + 2*(1.95 + 1.16) + 0.98 + 3.1 + 10.24};

```

Далее, с помощью оператора подстановки «/.» подставим их в левые части дифференциальных уравнений движения:

```

Eqn1 = (E1 - (M1[t] - M2[t])) /. ss;
Eqn2 = (E2 - (M2[t] - M3[t])) /. ss;
Eqn3 = (E3 - (M3[t] + M4[t] + M7[t])) /. ss;
Eqn4 = (E4 - (M4[t] - M5[t])) /. ss;
Eqn5 = (E5 - (M5[t] - M6[t])) /. ss;
Eqn6 = (E6 - M6[t]) /. ss;
Eqn7 = (E7 - (M3[t] + M4[t])) /. ss;
Eqn8 = (E8 - (M8[t] - M9[t])) /. ss;
Eqn9 = (E9 - M9[t]) /. ss;
Eqn8p = (E8p - (M8p[t] - M9p[t])) /. ss;
Eqn9p = (E9p - M9p[t]) /. ss;

```

Управляющие моменты определяются в виде функций от времени, на основании разработанной нами методики [1].

Также необходимо задать время, на котором численно будет интегрироваться система нелинейных дифференциальных уравнений движения:

```
tk = 0.3.
```

Задаем начальное положение и скорости антропоморфного механизма:

```
φ10 = φ1i[0]; φ20 = φ2i[0]; φ30 = φ3i[0]; φ40 = φ4i[0]; φ50 = φ5i[0]; φ60 = φ6i[0]; φ70 = φ7i[0]; φ80 = φ8i[0]; φ90 = φ9i[0]; φ80p = φ8pi[0]; φ90p = φ9pi[0];
```

Теперь проводим численное ее решение с использованной встроенной функции для численного решения дифференциальных уравнений и их систем NDSolve. В ней в качестве аргументов задаются сами уравнения, начальные значения углов (положение механизма) и начальные скорости, т.е. ставится задача Коши, также задается список неизвестных функций, относительно которых необходимо проводить решение системы дифференциальных уравнений, параметр  $t$ , являющийся аргументом неизвестных функций и пределы его изменения:

```
sol7 = NDSolve[{Eqn1 == 0, Eqn2 == 0, Eqn3 == 0, Eqn4 == 0, Eqn5 == 0, Eqn6 == 0, Eqn7 == 0, Eqn8 == 0, Eqn9 == 0, Eqn8p == 0, Eqn9p == 0, φ1[0] == φ10, φ2[0] == φ20, φ3[0] == φ30, φ4[0] == φ40, φ5[0] == φ50, φ6[0] == φ60, φ7[0] == φ70, φ8[0] == φ80, φ9[0] == φ90, φ8p[0] == φ80p, φ9p[0] == φ90p, φ1'[0] == 0, φ2'[0] == 0, φ3'[0] == 0, φ4'[0] == 0, φ5'[0] == 0, φ6'[0] == 0, φ7'[0] == 0, φ8'[0] == 0, φ9'[0] == 0, φ8p'[0] == 0, φ9p'[0] == 0}, {φ1, φ2, φ3, φ4, φ5, φ6, φ7, φ8, φ9, φ8p, φ9p}, {t, 0, tk}]
```

Ответ получаем в виде интерполяционных функций:

```
{φ1 -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ2 -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ3 -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ4 -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ5 -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ6 -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ7 -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ8 -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ9 -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ8p -> InterpolatingFunction[{{0., 0.3}}, "<>"], φ9p -> InterpolatingFunction[{{0., 0.3}}, "<>"]}}
```

В качестве ответа получаем набор интерполяционных функций для каждой из неизвестных функций.

Далее, стоит задача извлечения полезной информации из этих интерполяционных функций. Конечно, можно пойти традиционным путем и построить их графики (рис. 1 а). На них изображены зависимости углов поворота звеньев антропоморфного робота от времени. Так как это аналитические функции, их можно продифференцировать. Однократное дифференцирование по времени даст нам зависимости угловой скорости звеньев от времени (рис. 1 б), а повторное дифференцирование – зависимости углового ускорения звеньев механизма от времени (рис. 1 в). Построение графиков также имеет свою специфику по извлечению из переменной sol7, в которой, хранится все решение данной системы нелинейных дифференциальных уравнений движения.

```
ris1 = Plot[φ1[t] /. sol7, {t, 0, tk}, Background -> RGBColor[1, 1, 1], AxesLabel -> "φ1[t]", PlotStyle -> {RGBColor[0, 0, 0], Thickness[.01]}, DisplayFunction -> Identity]; ris11 = Plot[φ1'[t] /. sol7, {t, 0, tk}, Background -> RGBColor[1, 1, 1], AxesLabel -> "φ1'[t]", PlotStyle -> {RGBColor[0, 0, 0], Thickness[.01]}, DisplayFunction -> Identity]; ris111 = Plot[φ1''[t] /. sol7, {t, 0, tk}, Background -> RGBColor[1, 1, 1], AxesLabel -> "φ1''[t]", PlotStyle -> {RGBColor[0, 0, 0], Thickness[.01]}, DisplayFunction -> Identity];
```

```

ris2 = Plot[φ2[t] /. sol7, {t, 0, tk}, Background -> RGBColor[1,
1, 1], AxesLabel -> "φ2[t]", PlotStyle -> {RGBColor[0, 0, 0], Thick-
ness[.01]}, DisplayFunction -> Identity];
ris22 = Plot[φ2'[t] /. sol7, {t, 0, tk}, Background ->
RGBColor[1, 1, 1], AxesLabel -> "φ2'[t]", PlotStyle -> {RGBColor[0, 0,
0], Thickness[.01]}, DisplayFunction -> Identity];
ris222 = Plot[φ2''[t] /. sol7, {t, 0, tk}, Background ->
RGBColor[1, 1, 1], AxesLabel -> "φ2''[t]", PlotStyle -> {RGBColor[0,
0, 0], Thickness[.01]}, DisplayFunction -> Identity];
...
ris9p = Plot[φ9p[t] /. sol7, {t, 0, tk}, Background ->
RGBColor[1, 1, 1], AxesLabel -> "φ11[t]", PlotStyle -> {RGBColor[0, 0,
0], Thickness[.01]}, DisplayFunction -> Identity];
ris99p = Plot[φ9p'[t] /. sol7, {t, 0, tk}, Background ->
RGBColor[1, 1, 1], AxesLabel -> "φ11'[t]", PlotStyle -> {RGBColor[0,
0, 0], Thickness[.01]}, DisplayFunction -> Identity];
ris999p = Plot[φ9p''[t] /. sol7, {t, 0, tk}, Background ->
RGBColor[1, 1, 1], AxesLabel -> "φ11''[t]", PlotStyle -> {RGBColor[0,
0, 0], Thickness[.01]}, DisplayFunction -> Identity];

```

Для непосредственного вывода графиков на экран, используется следующая функция:

```

Show[GraphicsArray[{ris1, ris11, ris111}]];
Show[GraphicsArray[{ris2, ris22, ris222}]];
...
Show[GraphicsArray[{ris9p, ris99p, ris999p}]];

```

В результате, получаем следующие графики (рис. 2), для примера приведены зависимости координаты  $a$ ) и ее производных  $b$ ),  $в$ ):

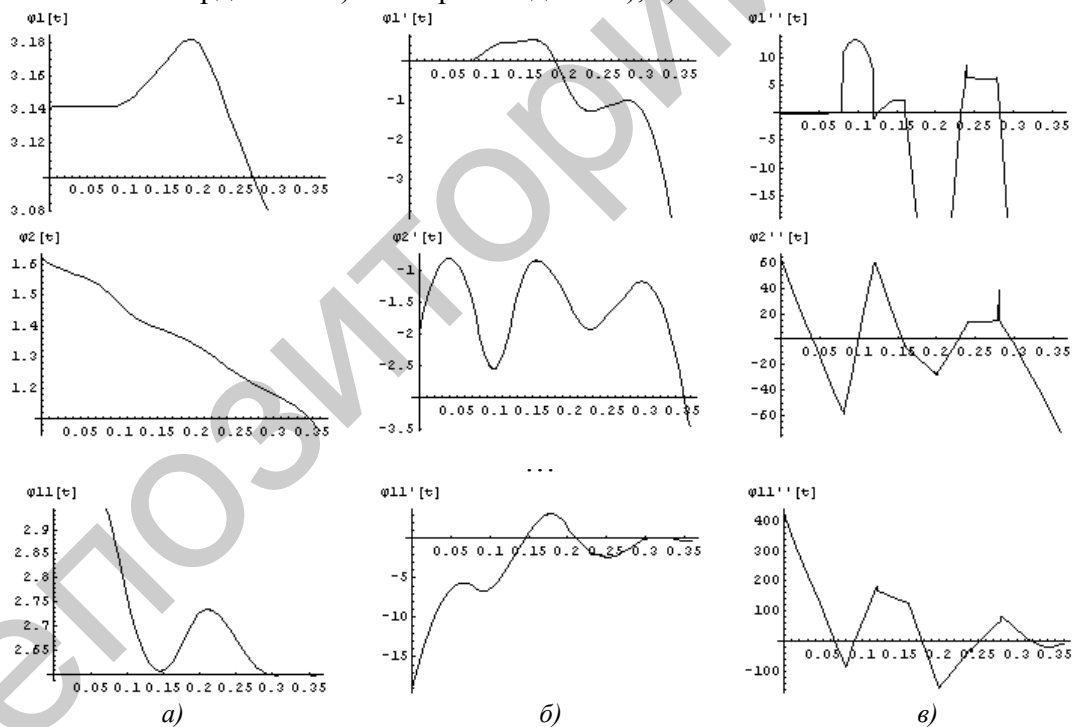


Рис. 2

Однако, из них можно получить и анимационную визуальную информацию.

Для этого зададим в явном виде:

```

fi1 = φ1 /. sol7[[1]];
fi2 = φ2 /. sol7[[1]];
...
fi9p = φ9p /. sol7[[1]];

```

**Замечание:** вообще говоря, анимационно визуализировать таким способом можно не только решение дифференциальных уравнений для антропоморфного робота, но и любое движение, заданное интерполяционными функциями, например, если экспериментально определить углы поворота звеньев человека при движении, затем по этой экспериментальной информации построить интерполяционные функции. Разница будет состоять только в значениях соответствующих констант, которые необходимо будет определить для человека и в том, что функции  $fi_1$  и т.д. должны будут задаваться по-другому, например, так:

```
fi1 = φ1i;
fi2 = φ2i;
...
fi9p = φ9pi;
```

вся остальная методика, описываемая ниже, будет оставаться прежней.

Затем задаем углы поворота звеньев в виде функций времени в виде отложенного задания функции пользователем:

```
Fi1[t_] := fi1[tk/2 + Sign[t]*(-tk/2 + Abs[t] -
tk*IntegerPart[Abs[t]/tk]);
Fi2[t_] := fi2[tk/2 + Sign[t]*(-tk/2 + Abs[t] -
tk*IntegerPart[Abs[t]/tk]);
...
Fi9p[t_] := fi9p[tk/2 + Sign[t]*(-tk/2 + Abs[t] -
tk*IntegerPart[Abs[t]/tk]);
```

Определим еще одну вспомогательную функцию пользователя, необходимую для более построения изображения антропоморфного робота:

```
PS7[t_] := If[t > 0, IntegerPart[t/tk], IntegerPart[t/tk] - 1];
```

Вычислим длину робота по горизонтали, необходимую для задания более компактного задания команд построения изображения:

```
Lk7 = Abs[l1[0]*Cos[φ10] + l2[0]*Cos[φ20] + l3[0]*Cos[φ30] -
l1[0]*Cos[φ60] - l2[0]*Cos[φ50] - l3[0]*Cos[φ40]] /. ss
```

Задаем функцию определения координат антропоморфного робота:

```
robotzwet7[fi1_, fi2_, fi3_, fi4_, fi5_, fi6_, fi7_, fi8_, fi9_,
fi8p_, fi9p_, a_, b_, t_] := {RGBColor[Mod[Abs[PS7[t] - 1], 2], 0,
Mod[Abs[PS7[t]], 2]], Thickness[.02], Line[{{Lk7*PS7[t] + 0, 0},
{Lk7*PS7[t] + l1[t]*Cos[fi1], l1[t]*Sin[fi1]}, {Lk7*PS7[t] +
l1[t]*Cos[fi1] + l2[t]*Cos[fi2], l1[t]*Sin[fi1] + l2[t]*Sin[fi2]},
{Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] + l3[t]*Cos[fi3],
l1[t]*Sin[fi1] + l2[t]*Sin[fi2] + l3[t]*Sin[fi3]}]},
RGBColor[Mod[Abs[PS7[t]], 2], 0, Mod[Abs[PS7[t] - 1], 2]],
Thickness[.02], Line[{{Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] +
l3[t]*Cos[fi3], l1[t]*Sin[fi1] + l2[t]*Sin[fi2] + l3[t]*Sin[fi3]},
{Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] + l3[t]*Cos[fi3] -
l4[t]*Cos[fi4], l1[t]*Sin[fi1] + l2[t]*Sin[fi2] + l3[t]*Sin[fi3] -
l4[t]*Sin[fi4]}, {Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] +
l3[t]*Cos[fi3] - l4[t]*Cos[fi4] - l5[t]*Cos[fi5], l1[t]*Sin[fi1] +
l2[t]*Sin[fi2] + l3[t]*Sin[fi3] - l4[t]*Sin[fi4] - l5[t]*Sin[fi5]},
{Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] + l3[t]*Cos[fi3] -
l4[t]*Cos[fi4] - l5[t]*Cos[fi5] - l6[t]*Cos[fi6], l1[t]*Sin[fi1] +
l2[t]*Sin[fi2] + l3[t]*Sin[fi3] - l4[t]*Sin[fi4] - l5[t]*Sin[fi5] -
l6[t]*Sin[fi6]}]},
RGBColor[0, 1, 0], Thickness[.05], Line[{{Lk7*PS7[t] +
l1[t]*Cos[fi1] + l2[t]*Cos[fi2] + l3[t]*Cos[fi3] + l7[t]*Cos[fi7],
l1[t]*Sin[fi1] + l2[t]*Sin[fi2] + l3[t]*Sin[fi3] + l7[t]*Sin[fi7]},
{Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] + l3[t]*Cos[fi3],
l1[t]*Sin[fi1] + l2[t]*Sin[fi2] + l3[t]*Sin[fi3]}]},
RGBColor[1, 0.8, 0.2], Thickness[.02], Line[{{Lk7*PS7[t] +
l1[t]*Cos[fi1] + l2[t]*Cos[fi2] + l3[t]*Cos[fi3] + l7[t]*n7R*Cos[fi7]
- l8[t]*Cos[fi8] - l9[t]*Cos[fi9], l1[t]*Sin[fi1] + l2[t]*Sin[fi2] +
l3[t]*Sin[fi3] + l7[t]*n7R*Sin[fi7] - l8[t]*Sin[fi8] -
```



```

19[t]*Sin[fi9]], {Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] +
13[t]*Cos[fi3] + 17[t]*n7R*Cos[fi7] - l8[t]*Cos[fi8], l1[t]*Sin[fi1] +
12[t]*Sin[fi2] + 13[t]*Sin[fi3] + 17[t]*n7R*Sin[fi7] -
18[t]*Sin[fi8]], {Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] +
13[t]*Cos[fi3] + 17[t]*n7R*Cos[fi7], l1[t]*Sin[fi1] + l2[t]*Sin[fi2] +
13[t]*Sin[fi3] + 17[t]*n7R*Sin[fi7]}}},
  RGBColor[0.8, 0.5, 0.9], Thickness[.02], Line[{{Lk7*PS7[t] +
11[t]*Cos[fi1] + l2[t]*Cos[fi2] + l3[t]*Cos[fi3] + 17[t]*n7R*Cos[fi7]
- l8p[t]*Cos[fi8p] - l9p[t]*Cos[fi9p], l1[t]*Sin[fi1] + l2[t]*Sin[fi2]
+ l3[t]*Sin[fi3] + 17[t]*n7R*Sin[fi7] - l8p[t]*Sin[fi8p] -
19p[t]*Sin[fi9p]}, {Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] +
13[t]*Cos[fi3] + 17[t]*n7R*Cos[fi7] - l8p[t]*Cos[fi8p], l1[t]*Sin[fi1]
+ l2[t]*Sin[fi2] + l3[t]*Sin[fi3] + 17[t]*n7R*Sin[fi7] -
18p[t]*Sin[fi8p]}, {Lk7*PS7[t] + l1[t]*Cos[fi1] + l2[t]*Cos[fi2] +
13[t]*Cos[fi3] + 17[t]*n7R*Cos[fi7], l1[t]*Sin[fi1] + l2[t]*Sin[fi2] +
13[t]*Sin[fi3] + 17[t]*n7R*Sin[fi7]}}}] /. ss

```

**Задаем функцию формирования изображения антропоморфного робота:**

```

RBF7[fi1_, fi2_, fi3_, fi4_, fi5_, fi6_, fi7_, fi8_, fi9_,
fi8p_, fi9p_, a_, b_, t_] := Graphics[{{robotzvet7[fi1, fi2, fi3, fi4,
fi5, fi6, fi7, fi8, fi9, fi8p, fi9p, a, b, t]}}, PlotRange -> {{-1,
2}, {-1, 2}}, AspectRatio -> 1, GridLines -> Automatic,
DisplayFunction -> Identity]

```

**Строим начальное положение антропоморфного робота:**

```

Show[RBF7[Fi1[0], Fi2[0], Fi3[0], Fi4[0], Fi5[0], Fi6[0],
Fi7[0], Fi8[0], Fi9[0], Fi8p[0], Fi9p[0], 0.2, 0.18, 0.0], Display-
Function -> $DisplayFunction];

```

Иллюстрация работы этой команды приведена на рис. 3.

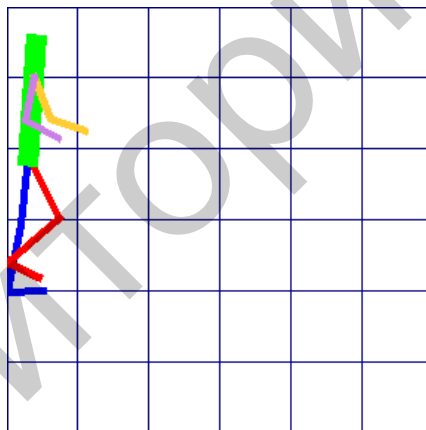


Рис. 3.

**Задаем функцию подготавливающую анимацию антропоморфного робота:**

```

mis7F[t_] := RBF7[Fi1[t], Fi2[t], Fi3[t], Fi4[t], Fi5[t], Fi6[t],
Fi7[t], Fi8[t], Fi9[t], Fi8p[t], Fi9p[t], 0.2, 0.18, t]

```

**Вводим функцию, формирующую n кадров анимации движения:**

```

Res7[n_] := Table[Show[mis7F[tk/n*i], DisplayFunction -> $Dis-
playFunction], {i, n}]

```

**Выводим тридцать кадров анимации командой:**

```
Res7[30]
```

Результаты работы описанной методики иллюстрируются рисунком 3, на котором приведено четыре кадра анимации ходьбы антропоморфного робота.

Экспорт анимации с разными параметрами в распространенный видео формат «.gif» осуществляется следующими командами:

```

SetDirectory["D:\\Robot"] rbtmovie = Res7[30];
Export["RobotDeform.gif", rbtmovie, ConversionOptions -> {"Loop"
-> True}];

```

```
Export["RobotDeform1.gif", rbtmovie, ConversionOptions ->
{"Loop" -> False}, AnimationDisplayTime -> 1];
```

Таким образом, описана методика численного решения системы дифференциальных уравнений движения модели антропоморфного робота и его анимационная визуализация.

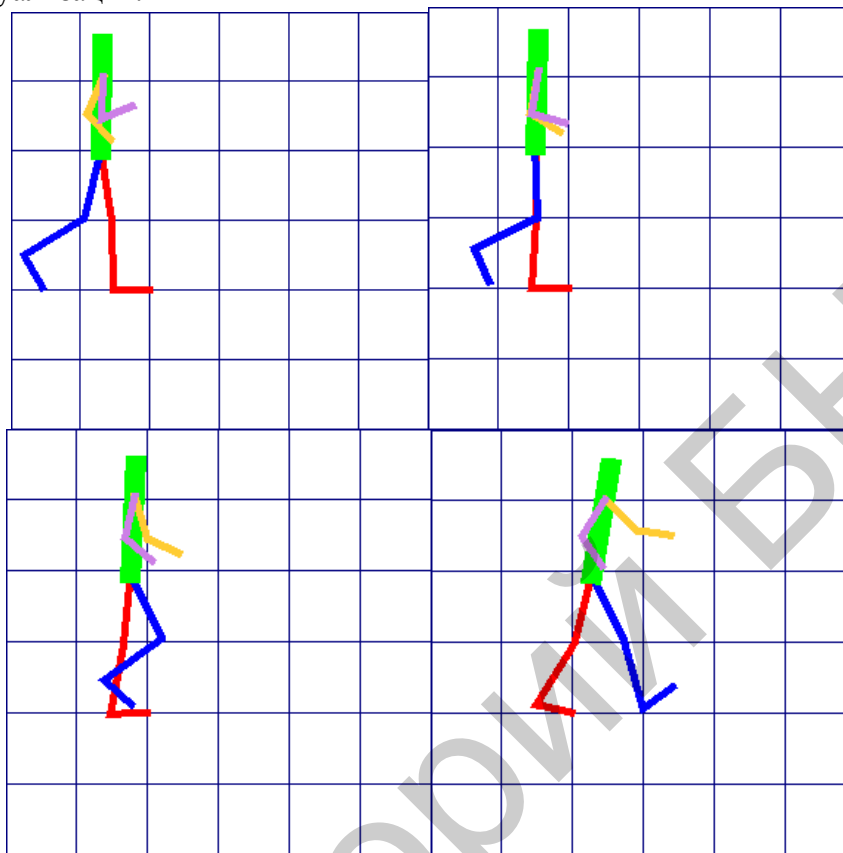


Рис. 4

Описанная методика является достаточно универсальной, и может применяться для решения систем дифференциальных уравнений динамики разнообразных объектов, с большим или меньшим количеством степеней свободы. Анимационная визуализация также может применяться для визуализации движений различных объектов, но при этом она требует изменения текстов функций, определяющих непосредственно геометрическую структуру антропоморфного робота. Необходимо будет задавать соответствующие новой модели геометрические связи. Но сам подход при этом может оставаться прежним.

#### ЛИТЕРАТУРА

1. Борисов А.В. Моделирование опорно-двигательного аппарата человека и применение полученных результатов для разработки модели антропоморфного робота. – М.: Спутник +, 2009. – 212 с.
2. Дьяконов В.П. Mathematica 5.1/5.2/6. Программирование и математические вычисления. – М.: ДМК-Пресс, 2008. – 576 с.
3. Чигарев А.В., Кравчук А.И., Кравчук А.С. Основы системы Mathematica 6.0. Задачи и решения. – Минск, 2008. – 136 с.
4. Чигарев А.В., Михасев Г.И. Борисов А.В. Биомеханика. – Минск: Изд-во Гревцова, 2010. – 284 с.