Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕХНОЛОГИИ И ДИЗАЙНА»

КОМПЬЮТЕРНЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ПО КИНЕМАТИКЕ СЛОЖНОГО ДВИЖЕНИЯ ТОЧКИ И ПЛОСКОГО ДВИЖЕНИЯ ТВЕРДОГО ТЕЛА

Методические указания к самостоятельной работе
и выполнению домашних заданий
для студентов СПГУТД, обучающихся по направлениям
150000 – металлургия, машиностроение и металлообработка,
220000 – автоматика и управление,
260000 – технология продовольственных продуктов и потребительских товаров

Составитель
А. Г. Усов

Санкт-Петербург

2013

УДК 531.1(075.8)

УТВЕРЖДЕНО
на заседании кафедры ТПМ
23.01.2013 г.
протокол № 4

Рецензент
профессор кафедры
машиноведения СПГУТД
А. В. Марковец

Оригинал подготовлен составителем
и издан в авторской редакции.

Подписано в печать 11.04. 2013 г. Формат 60 х $84^1/_{16}$.

Усл. печ. л. 1.5. Тираж 100 экз. Заказ 129 /13

Электронный адрес: http://publish.sutd.ru

Отпечатано в типографии ФГБОУВПО «СПГУТД»

191028, С.-Петербург, ул. Моховая, 26

Оглавление

Введение 4
1.Сложное движение точки 4
1.1. Пример решения задачи «вручную» 4
1.2. Компьютерное решение задания 9
2. Плоское движение твердого тела 11
2.1. Содержание задания 11
2.2. Кинематический анализ кривошипно-ползунного механизма (КПМ): постановка задачи 11
2.3. КПМ: геометрический анализ. ……………………………………. 13
2.4. Расчет скоростей и ускорений 15
2.5. Решение задачи на компьютере 15
2.5.1. Ввод исходных данных 15
2.5.2. Анализ исходных данных 16
2.5.3. Коды программ `kriv_polz,kriv_polz1` 20
2.5.4. Пример счета по программе `kriv_polz1` 24
Заключение 25
Список литературы 25

КОМПЬЮТЕРНЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ПО КИНЕМАТИКЕ СЛОЖНОГО ДВИЖЕНИЯ ТОЧКИ И ПЛОСКОГО ДВИЖЕНИЯ ТВЕРДОГО ТЕЛА

Введение

Уважаемые студенты! Механика всегда являлась источником и полем приложения математических идей. Леонардо да Винчи называл ее раем для математики. Задачи механики легко алгоритмизируются и программируются. Хороший инженер сочетает интуитивный подход к решению задачи со строгим выводом решения по известным ему формулам и умозаключениям. Постановка задачи на компьютере зачастую есть не просто проверка «ручного» решения, а создание нового метода исследования, позволяющего изучить поведение исследуемой модели при варьировании тех или иных параметров.

Здесь рассматриваются примеры компьютерного решения задач на сложное движение точки и плоское движение твердого тела. В качестве объектов исследования взяты задачи разделов К.7 и К.3 из известного сборника заданий для курсовых работ по теоретической механике под редакцией А. А. Яблонского [1].

1. Сложное движение точки

1.1. Пример решения задачи «вручную»

В этом задании требуется определить величины (т.е. модули векторов) абсолютной скорости и абсолютного ускорения точки, совершающей сложное движение. Абсолютным движением точки *M* является ее движение относительно неподвижной системы отсчета, связанной с подшипниками, в которые вставлен вал, несущий тело *D*. С этим вращающимся телом свяжем подвижную систему отсчета. Таким образом, относительным движением точки *M* будет ее движение относительно тела *D*.

Абсолютная скорость и абсолютное ускорение определяются с использованием формул, выражающих теоремы сложения [2]: $\vec{v}_a = \vec{v}_r + \vec{v}_e$, $\vec{a}_a = \vec{a}_r + \vec{a}_e + \vec{a}_{cor}$. Вспомним, что здесь $\vec{v}_r$ - относительная (*relatif - фр.)* скорость точки, т.е. скорость ее относительно тела *D,* а $\vec{v}_e$ - переносная (*emporter)* скорость точки M. Это скорость того пункта подвижной системы отсчета (тела *D*) , в котором находится точка M в расчетный момент времени.

Аналогичный смысл имеют относительное и переносное ускорения. Ускорение Кориолиса рассчитывается по формуле: $\vec{a}_{cor} = 2\vec{\omega}_e \times \vec{v}_r$, где $\vec{\omega}_e$ - вектор угловой скорости подвижной системы отсчета (тела *D*), направленный вдоль оси вращения тела. Направление вращения тела показывается на рисунке также и круговой стрелкой, изображаемой как бы в аксонометрии, причем разрыв стрелки происходит за осью.

Величина (модуль) вектора в евклидовом пространстве есть корень квадратный из скалярного произведения вектора на самого себя. Скалярное произведение удобно рассчитывать в декартовой системе координат. При этом проекция вектора на ось равна сумме проекций его составляющих.

При решении задачи «вручную» информация о направлениях векторов хранится на рисунке, а информация об их величинах - в тексте решения. Только после того, как определены и величина, и направление какого-либо вектора, имеет смысл переходить к определению следующего вектора.

Условие задания К.7: точка *M* движется относительно тела *D* (рис. 1) по окружности радиуса *R* согласно уравнению $s_r = s_r(t)$, где s_r - дуговая координата точки М на ее относительной траектории, которая отсчитывается в направлении от точки O_1 до изображенной в условии точки M. В таблице 6 сборника заданий относительная траектория показана сдвоенной линией.

Тело D вращается вокруг неподвижной оси согласно уравнению $\varphi_e = \varphi_e(t)$, где φ_e - угол поворота тела D вокруг оси его вращения Oz, которую направляем так, чтобы ее направление было связано с направление отсчета φ_e правилом правого винта.

Требуется найти величины абсолютной скорости и абсолютного ускорения точки M в момент времени t_1.

<u>Дано</u>: $R = 0.2\,см$; $s_r = 0.4\pi t^2 / 3\,см$; $\varphi_e = 3t - t^2$; $t_1 = 1c$.

<u>Найти</u>: v_A; a_A.

<u>Решение.</u>

Р1. На рис. 1 показываем направление оси z вращения тела D. Представляем себе движение точки M как сложное движение, и при этом считаем движение этой точки относительно тела D относительным, а вместе с телом (пластиной) – переносным. Оба эти движения заданы исходными данными. Тогда мы оказываемся в условиях, описываемых теорией сложного движения: $\vec{v}_a = \vec{v}_r + \vec{v}_e$, $\vec{a}_a = \vec{a}_r + \vec{a}_e + \vec{a}_{cor}$.

Р2. Исследуем относительное движение, «заморозив» мысленно переносное. Относительное движение точки M задано естественным способом: дана траектория – окружность радиуса R, и закон движения $s_r = s_r(t)$. Далее в обозначении координаты s_r опустим индекс «r».

Р2.1. Определяем положение M_1 точки M в подвижной системе отсчета. Рассчитываем координату $s_{r1} = s_r(t_1) = s_1 = 0.4\pi / 3$ см. Это длина дуги OM_1 окружности, которой соответствует центральный угол $\delta = |s_1| / R = 2\pi / 3 = 120°$. Дугу OM_1 откладываем от точки O_1 в сторону положительных значений координаты s, т.к. $s_1 > 0$. Пометим на рисунке углы $\alpha = 30°$ и $\beta = 60°$.

Р2.2. Находим относительную скорость.

Находим производную $\dot{s} = 0{,}8\pi\, t / 3$. Ее значение в момент t_1: $\dot{s}_1 = 0{,}8\pi / 3 = 0.84$.

Величина относительной скорости: $v_r = |\dot{s}_1| = 0.84 см / c$.

Направление: вектор $\vec{v}_r$ направлен по касательной к относительной траектории в точке M_1 в сторону возрастания координаты s, т.к. $\dot{s}_1 > 0$ (рис. 1).

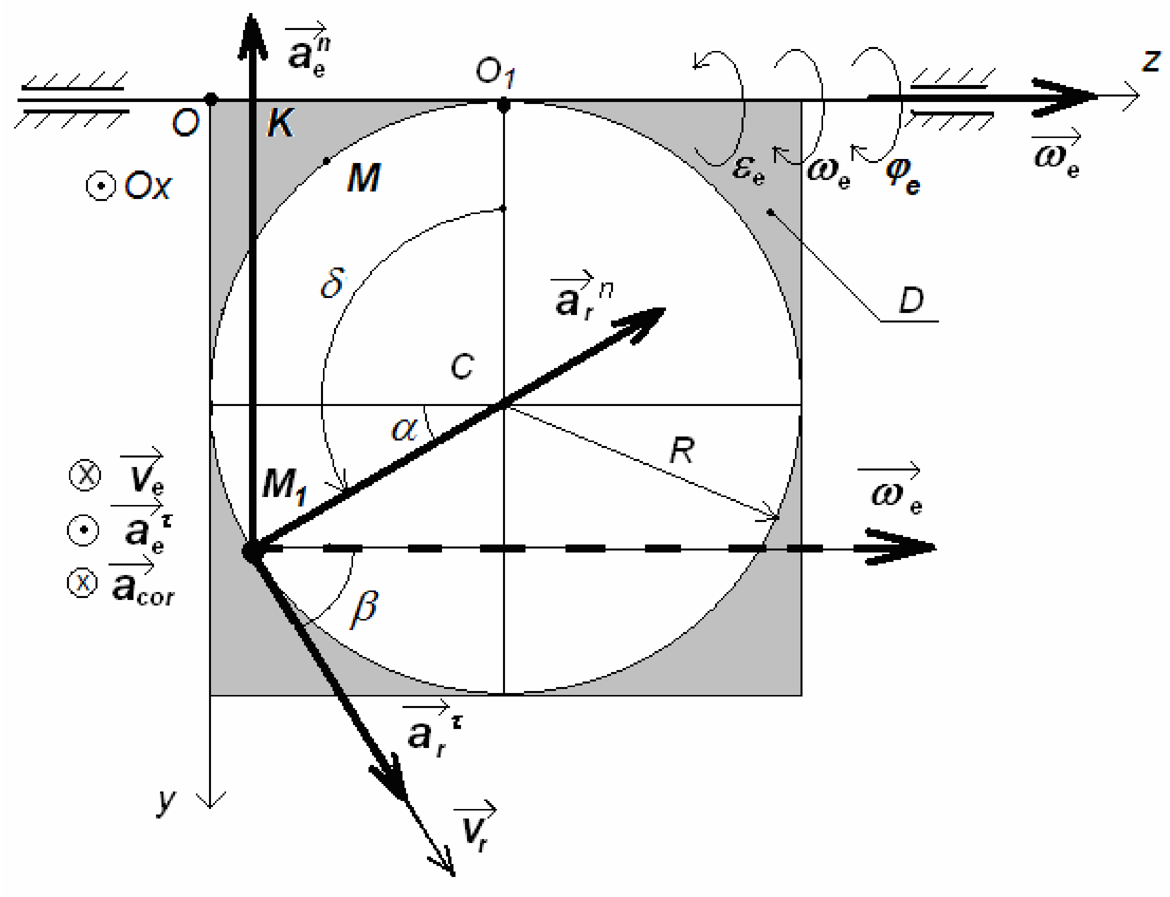


Рис. 1. Составляющие абсолютной скорости и абсолютного ускорения

Р2.3. Относительное ускорение.

При естественном задании движения точки ускорение раскладывают на естественные составляющие: $\vec{a}_r = \vec{a}_r^\tau + \vec{a}_r^n$.

Р2.3.1. Касательное ускорение.

Находим вторую производную $\ddot{s} = 0.8\pi / 3$.

Величина касательного ускорения: $\left|a_r^\tau\right| = |\ddot{s}| = 0.84 см/c^2$.

Направление: вектор $\vec{a}_r^\tau$ направлен по касательной к относительной траектории в сторону возрастания координаты s, т.к. $\ddot{s} > 0$.

Р2.3.2. Нормальное ускорение.

Величина: $a_r^n = \dfrac{v_r^2}{R} = \dfrac{0,84^2}{0,2} = 3.53 см/c^2$.

Направление: вектор $\vec{a}_r^n$ направлен к центру C кривизны относительной траектории.

Для дальнейших расчетов удобнее оставить относительное ускорение разложенным на составляющие.

Р3. Исследуем переносное движение, «заморозив» относительное, т.е. закрепив мысленно точку M_1 на пластине. Движение пластины D – вращательное. Переносные скорость и ускорение – это скорость и ускорение точки вращающегося твердого тела D и зависят от его кинематических параметров.

Р3.1. Исследуем движение тела *D*.

а). Положение тела: будем считать, что плоскость пластины в момент t_1 лежит в плоскости рисунка.

б). Угловая скорость тела.

Находим производную $\dot{\varphi}_e = 3 - 2t = \omega_{ez}$ и ее значение при $t = t_1$: $\dot{\varphi}_{e1} = 1 = \omega_{ez1}$.

Величина угловой скорости : $\omega_e(t_1) = |\dot{\varphi}_{e1}| = 1 рад/c$.

Направление: вектор $\vec{\omega}_e$ направлен так же, как и ось Oz, т.к. его проекция $\omega_{ez1} > 0$. Направление этой угловой скорости указано двумя способами: круговой стрелкой и прямой.

в). Угловое ускорение пластины D.

Величина: $\varepsilon_{ez} = \dot{\omega}_{ez} = -2, \quad \varepsilon_e = \left|\varepsilon_{e,z}\right| = 2$ рад/c^2.

Направление: вектор $\vec{\varepsilon}_e$ направлен против оси Oz, т.к. его проекция ε_{ez} на эту ось отрицательна. Укажем направление углового ускорения соответствующей круговой стрелкой.

Р3.2. Исследуем переносное движение самой точки M_1.

Р3.2.1. Траектория точки M_1 как точки тела *D* – окружность радиуса h, где h - расстояние от точки M_1 до оси вращения. Плоскость, в которой лежит эта окружность, перпендикулярна оси вращения. Опустим из точки перпендикуляр M_1K на ось:

$$h = MK = R + R \sin\alpha = 0.2 + 0.1 = 0.3 \text{ см}.$$

Касательная к обсуждаемой окружности в точке M_1 перпендикулярна плоскости рисунка. Введем ось Oy, как показано на рисунке, и ось Ox, так, чтобы орты $\vec{i}, \vec{j}, \vec{k}$ наших осей образовали правый ортонормированный базис: $\vec{i} \times \vec{j} = \vec{k}$. Тогда ось Ox направлена «на нас».

Р3.2.2. Переносная скорость.

Величина: $v_e = \omega_e h = 0.3 см/c$.

Направление: вектор $\vec{v}_e$ направлен по касательной к окружности радиуса h, т.е. перпендикулярно плоскости тела D «от нас» - кружок с крестовиной на рис. 1.

Р3.2.3. Переносное ускорение.

При вращательном движении твердого тела ускорение его точки ищем в виде суммы касательной (вращательной) и нормальной (центростремительной) составляющих:

$$\vec{a}_e = \vec{a}_e^{\tau} + \vec{a}_e^{n}.$$

Находим касательное ускорение.

Величина: $\left|a_e^\tau\right| = \varepsilon\, h =$ 2× 0,3 = 0,6 $см/с^2$.

Направление: поскольку вектор $\vec{\varepsilon}_e$ направлен против $\vec{\omega}_e$, то и вектор $\vec{a}_e^\tau$ направлен против вектора переносной скорости, т.е. «на нас» - кружок с точкой на рис. 1.

Находим нормальное ускорение.

Величина: $a_e^n = \omega_e^2\, h =$ 1× 0.3 = 0.3 см/c^2.

Направление: это ускорение направлено к центру O окружности радиуса h.

Р4. Ускорение Кориолиса $\vec{a}_{cor} = 2\vec{\omega}_e \times \vec{v}_r$.

Величина: $a_{cor} = 2\,\omega_e\, v_r \sin(\vec{\omega}_e; \vec{v}_r) = 2 \cdot 1 \cdot 0.84 \sin\beta = 1.68 \sin 60° = 1.45$ см/c^2.

Направление: для удобства расчета вектор $\vec{\omega}_e$ перенесен параллельно в точку M и обозначен пунктиром. Вращаем правый винт от первого множителя $\vec{\omega}_e$ в векторном произведении ко второму множителю $\vec{v}_r$ в сторону наименьшего угла β между ними. Движение оси винта происходит перпендикулярно плоскости рисунка «от нас» (кружок с крестовиной).

Р5. Находим кинематические параметры абсолютного движения точки M.

Р5.1. Абсолютная скорость.

Величина: в нашем примере два составляющих вектора абсолютной скорости взаимно ортогональны, так что $v_a = \sqrt{v_r^2 + v_e^2} = \sqrt{0.84^2 + 0.3^2} = 0.89$ см/с.

Направление абсолютной скорости определять в задании не надо.

Р5.2. Абсолютное ускорение.

В итоге предыдущих рассуждений имеем: $\vec{a}_a = \vec{a}_r^\tau + \vec{a}_r^n + \vec{a}_e^\tau + \vec{a}_e^n + \vec{a}_{cor}$. Поскольку слагаемых здесь более двух, сложение выполним аналитически в декартовой системе координат $Oxyz$. Тогда величина (модуль вектора) абсолютного ускорения будет равна $a_a = \sqrt{\vec{a}_a \vec{a}_a} = \sqrt{a_{ax}^2 + a_{ay}^2 + a_{az}^2}$.

Величина абсолютного ускорения:

$$a_{ax} = a_{rx}^\tau + a_{rx}^n + a_{ex}^\tau + a_{ex}^n + a_{cor,x} = a_{cor} - \left|a_e^\tau\right| = 1.45 - 0.6 = 0.85,$$

$$a_{ay} == \left|a_r^\tau\right| \cos\alpha - a_r^n \cos\beta - a_e^n = 0.84 \cos 30° - 3.53 \cos 60° - 0{,}3 = -1.34,$$

$$a_{az} == \left|a_r^\tau\right| \cos\beta + a_r^n \cos\alpha = 0.84 \cos 60° + 3.53 \cos 30° = 3.48,$$

$$a_a = \sqrt{0.85^2 + 1.34^2 + 3.48^2} = 3.82\, см/с^2.$$

Направление вектора $\vec{a}_a$ определять не надо.

Ответ: $v_a = 0.89\, см/с, \quad a_a = 3.82\, см/с^2.$

1.2. Компьютерное решение задания

Во всех условиях задания К.7 из сборника Яблонского [2] тело D совершает или поступательное, или вращательное движение. Отличать эти два случая будем, вводя в программу признак p, принимающий значение 1 при вращательном (ротационном) движении подвижного репера и значение 0 при поступательном (трансляционном) его движении. Рассчитывать скорости и ускорения будем в декартовой системе координат, связанной с телом D, причем ось Oz направлена по оси вращения тела.

Программу составим как программу-процедуру под названием `check_k7`. Она предназначена только для проверки правильности окончательного ответа: вычисления величин абсолютной скорости и абсолютного ускорения. Рассмотрим текст (листинг, или, как еще выражаются, коды программы) в терминах Матлаб'а [3].

```
%Процедура check_k7: rонтроль счета в задании К.7
%(Яблонский) методом сложн. дв-я. r(abs)=rc + r(rel)
%Подвижный базис, связанный с телом D, движется поступ.
%или вращается вокруг неподв. оси Oz по закону fi=fi(t)
%Надо построить выражения для относительных координат
%в виде зависимостей от натур. параметра s, который,
%в свою очередь, задается функцией s=s(t)
clear all;
syms t s fie xc yc zc xr yr zr;
char namestud;
novar = input('Задание К.7. Номер варианта = ');
namestud = input('Фамилия, группа   ','s');
s = input('Задаем закон относит. движения: натур. парам. s(t) = ');
xr = input('Вводим относительную абсциссу точки М: xr(s) = ');
yr = input('Вводим относительную ординату точки М: yr(s) = ');
zr = input('Вводим относительную аппликату точки М: zr(s) = ');
t1 = input('Вводим расчетный момент t1 = ');
pr = input('Введи вид перенос дв-я: 1 -вращат., 0 -поступ. pr=');
if pr
   fie = input('Вводим угол поворота fie(t) = ');
   dfie = diff(fie);
   om = [0 0 dfie];
   eps = diff(om);
   vc = [0 0 0];%или vc = zeros(1,3);
   ac = [0 0 0];
else
   xc = input('Вводим абсолютную абсциссу полюса С: xc(t) = ');
   yc = input('Вводим абсолютную ординату полюса С: yc(t) = ');
   zc = input('Вводим абсолютную аппликату полюса С: zc(t) = ');
   rc = [xc yc zc];%Рад-вект. точки С в неподв. базисе
   vc = diff(rc);%скорость полюса
   ac = diff(vc);%ускорение полюса
   om = [0 0 0];
   eps = [0 0 0];
end
r = [xr yr zr];%Рад-вект. точки М в подвижном базисе
vr = diff(r);%вектор относит. скорости
ar = diff(vr);%вектор относит. ускорения
t = t1;%******************   count    ******************************
vc1 = subs(vc);
ac1 = subs(ac);
r1 = subs(r);
vr1 = subs(vr);
ar1 = subs(ar);
om1 = subs(om);%вектор omega.emporter(t1)
eps1 = subs(eps);
verot1 = cross(om1,r1);%переносная скорость ротации
```

```
vabs1 = norm(vc1+verot1+vr1);
acor1 = 2*cross(om1,vr1);
aabs1 = norm(ac1+cross(eps1,r1)+cross(om1,verot1)+ar1+acor1);
disp('СЧЕТ: величина абс. скорости при t=t1, см/с, vabs1 = '); disp(vabs1);
disp('Величина абс. ускорения при t=t1, см/с^2, aabs1 = '); disp(aabs1);
```

Программа коротка, поскольку расчеты в основном производятся в векторной форме и используются символьные вычисления. Описан тип только тех символьных величин, которые вводятся. Величинам, которые получаются из вводных, символьный тип присваивается в программе автоматически.

Просчитаем предыдущий пример. Главная подготовительная работа - создание выражений $x_r(s), y_r(s), z_r(s)$, описывающих зависимости относительных декартовых координат от натурального параметра. Радиус-вектор точки M в локальном репере $Oxyz$ представляем в виде $\vec{r}\,'_M = \overrightarrow{OC} + \overrightarrow{CM}$. Записывая последнее равенство в проекциях на оси, получаем:

$x_r = 0$,

$$z_r = R - R\sin\delta = R - R\sin\frac{s}{R} = 0.2 - 0.2\sin\frac{s}{0.2},$$

$$y_r = R - R\cos\delta = 0.2 - 0.2\cos\frac{s}{0.2}.$$

С дисплея командного окна копируем протокол диалога и счета:

```
>> check_k7
Задание K.7. Номер варианта = 1
Фамилия, группа   Ivanoff 1-MD-45
Задаем закон относит. движения: натур. парам. s(t) = 0.4*pi*t*t/3
Вводим относительную абсциссу точки M: xr(s) = 0
Вводим относительную ординату точки M: yr(s) = 0.2-0.2*cos(s/0.2)
Вводим относительную аппликату точки M: zr(s) = 0.2-0.2*sin(s/0.2)
Вводим расчетный момент t1 = 1
Введи вид перенос дв-я: 1 -вращат., 0 -поступ. pr=1
Вводим угол поворота fie(t) = 3*t-t*t
Величина абс. скорости при t=t1, см/с, vabs1 =
    0.8899
Величина абс. ускорения при t=t1, см/с^2, aabs1 =
    3.8011
```

Студенту предлагается самому создать графический объект, дублирующий рис. 1. Для изображения прямоугольных областей используем матлабовские функции `patch` или `fill`. Круг на плоскости удобно задать функцией `rectangle`, но в пространстве лучше создать круг как множество треугольных секторов, вводимых функцией `patch`. Надо создать анимацию, изображающую движение тела D и движение точки M по телу D. Изображение атрибутов тела D, векторов скоростей и ускорений, рассчитываемых в локальном базисе, надо представлять в абсолютной системе координат $OXYZ$. Пусть ось OZ совпадает с осью Oz и пусть в начальный момент плоскость Oyz совпадает с плоскостью OYZ. Задаемся шагом по времени Δt и количеством шагов n. В программу вводим цикл, состоящий из n шагов. В теле цикла выполняем расчеты, начинающиеся со строки, комментированной как `count`. Перерасчет проекций векторов при повороте тела D делается с помощью введения матрицы поворота вокруг оси OZ. Так, столбец

$r'_M = \begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix}$ относительных координат точки M преобразуется в столбец абсолютных координат $r_M = \begin{pmatrix} x_M \\ y_M \\ z_M \end{pmatrix}$ по формуле $r_M = A_z r'_M$, где $A_z = \begin{pmatrix} \cos\varphi_e & \sin\varphi_e & 0 \\ -\sin\varphi_e & \cos\varphi_e & 0 \\ 0 & 0 & 1 \end{pmatrix}$. Создание матрицы поворота рассматривается в следующем разделе.

2. Плоское движение твердого тела

2.1. Содержание задания

Основная тема задания К.3 в задачнике А. А. Яблонского - кинематика плоскопараллельного движения твердого тела. Применение компьютерных технологий позволяет расширить условие задания и исследовать работоспособность механизма, проследить за его работой в течение некоторого времени.

Кинематические схемы условий задания К.3 имеют 4 вида: 1) кривошипно-ползунный механизм (КПМ), 2) эллипсограф, 3) планетарный ил дифференциальный механизм (ДМ), 4) катящееся без скольжения колесо. Наиболее трудоемкая программа – для расчета КПМ. Расчет эллипсографа сводится к расчету диады «шатун-ползун» (диады ВВП), которая входит в состав КПМ. Расчет планетарного механизма с одной степенью свободы есть частный случай расчета ДМ с двумя степенями свободы. Он выполняется с привлечением формулы Виллиса. Абсолютную угловую скорость сателлита, определенную по формуле Виллиса, следует проинтегрировать для исследования положения механизма и продифференцировать для расчета ускорений.

2.2. Кинематический анализ кривошипно-ползунного механизма (КПМ): постановка задачи

Требуется выполнить кинематический анализ кривошипно-ползунного механизма, а именно: определить положения звеньев механизма, их угловые скорости и угловые ускорения, а также скорости и ускорения отдельных точек звеньев в заданные моменты времени.

Кривошипно-ползунный механизм (рис. 2, а) состоит из 4 звеньев: одного неподвижного звена, называемого стойкой, и трех подвижных звеньев. Стойка, или звено №0 – это станина, суппорт, основание механизма, изображаемое на рис. косой штриховкой. Звено №1 – кривошип OA, образующий со стойкой вращательную пару, т.е. звено OA может совершать относительно стойки вращательное движение вокруг оси Oz, проходящей через точку O и направленной перпендикулярно плоскости рисунка. Звено №2 - шатун – образует с кривошипом и с ползуном также вращательные пары. Шатун относительно стойки совершает плоскопараллельное движение. Звено №3 – ползун – относительно стойки движется поступательно. Со стойкой связываем абсолютную систему координат $Oxyz$.

Ведущим звеном будем считать кривошип, поскольку его движение задано. По умолчанию, предполагаем, что в реальной машине к этому звену подводится мощность от внешнего ее источника (электродвигателя ЭД), и приложен вращающий момент $T_{OA,z}$ (torque). Вектор этого момента направлен вдоль оси вращения. Подводимая мощность N есть скалярное произведение векторов вращающего момента и угловой скорости кривошипа:

$$N = \vec{T}_{OA} \cdot \vec{\omega}_{OA} = T_{OA,z}\omega_{OA,z},$$

поскольку $\omega_{OA,x} = \omega_{OA,y} = 0$. Во время разгона или установившегося «рабочего» движения $N > 0$, а во время, скажем, рекуперативного торможения (когда ЭД работает в режиме генератора) $N < 0$.

Наш механизм имеет одну степень свободы. Положение механизма удобно задавать обобщенной координатой φ - углом поворота кривошипа OA. Угол φ будем отсчитывать от неподвижной оси Ox против часовой стрелки. Аналогичное правило знаков (правило ориентации) применяем по отношению ко всем углам и угловым кинематическим параметрам звеньев. Углы считаем положительными, когда они отсчитываются против часовой стрелки. Угловые скорости и угловые ускорения звеньев будем считать положительными, если они направлены против часовой стрелки. Последнее условие есть условие положительности проекций векторов угловых скоростей и угловых ускорений на ось Oz, направленную на нас. Правило часовой стрелки есть редуцированное (упрощенное, сведенное к плоской ситуации) правило правого винта: вращение рукоятки воображаемого правого «буравчика» против часовой стрелки вызывает движение его острия в ту сторону, куда направлена ось Oz.

Согласуем условие нашей расширенной задачи с условием К.3 задачника Яблонского. Движение кривошипа будем считать равнопеременным, происходящим согласно закону

$$\varphi = \varphi_0 + \dot{\varphi}_0 t + \ddot{\varphi}\frac{t^2}{2} = \varphi_0 + \omega_{OA,z}(0)\cdot t + \varepsilon_{OA,z}\frac{t^2}{2}. \qquad (1)$$

Здесь $\varphi_0, \omega_{OA,z}(0)$ - начальные значения угла поворота φ и угловой скорости ω_{OA} звена OA, заданные в условии К.3 и доопределенные с учетом правила знаков, $\varepsilon_{OA,z} = const$ - заданное угловое ускорение звена OA, которое считаем постоянным. Начальным положением механизма считаем то положение, которое изображено на рисунке к заданию К.3.

Дополнительно к условию К.3 задается шаг по времени Δt и числом шагов n. Надо выполнить кинематический анализ механизма в $n+1$ его положении, начиная с начального. Иначе говоря, для каждого расчетного значения времени $t_k = \Delta t(k-1)$, где $k = 1,\ldots n+1$, надо определить положения звеньев, их угловые скорости и угловые ускорения, а также скорости и ускорения точек A, B и некоторой точки M шатуна. Положение точки M задается ее локальными координатами в подвижном базисе, связанном с шатуном.

Решение задачи включает в себя множества значений (числовые массивы) параметров, определяющих положения, скорости и ускорения в расчетные моменты времени. Для каждого значения t_k выполняется одна и та же процедура.

1) Сначала, используя формулу (1), находим величины $\varphi(t_k)$ и $\omega_{OA,z} = \dot{\varphi}(t_k)$ (величина $\varepsilon_{OA,z}$ известна из условия).

2) Производим геометрический анализ механизма и определяем положения его звеньев.

3) Рассчитываем угловые скорости звеньев и скорости их точек A, B, M.

4) Рассчитываем угловые ускорения звеньев и ускорения точек.

Расчеты должны сопровождаться графическими иллюстрациями. В конце решения подводим итоги расчетам и делаем выводы. При этом могут быть построены различные графики, выражающие зависимости между переменными величинами.

2.3. КПМ: геометрический анализ.

Конфигурация механизма при заданном значении угла φ определяется заданными постоянными параметрами механизма. Это длины звеньев $l_1 = OA$, $l_2 = AB$, параметры направляющей ползуна x_D, y_D, ψ, признак p_{sb} варианта сборки присоединенной к кривошипу диады 2-3 (шатун-ползун) и локальные координаты AM, α_M точки M. Всего 8 параметров.

Положение направляющей ползуна в абсолютном базисе задаем координатами x_D, y_D какую-нибудь ее точки D и углом ψ наклона направляющей к оси Ox. Придаем «направление направляющей» и связываем с ней неподвижный базис Dx_3y_3, как показано на рис. 2, а. Этот базис должен быть правым согласно «декларации об ориентации». Это означает, что поворот от оси Dx_3 к оси Dy_3 в сторону наименьшего угла 90° между ними совершается против часовой стрелки. Угол ψ отсчитываем от оси Ox к оси Dx_3 против часовой стрелки.

Систему координат, связанную со звеном № 1 (кривошипом) будем называть $СК_1$, систему координат, связанную со звеном № 2 , будем называть $СК_2$ и т.д. Абсолютная система $СК_0$ связанна с полюсом O, $СК_{03}$ – система координат Dx_3y_3.

Признаку p_{sb} сборки диады 2-3 дадим значение $p_{sb} = +1$, если проекция вектора $\overrightarrow{AB}$ на ось Dx_3 положительна, и $p_{sb} = -1$, если наоборот. Два значения признака сборки соответствуют двум возможным положениям B и B' ползуна при одном и том же положении шарнира A (рис. 4, а). Переход механизма из одной сборки в другую означал бы, что механизм проходит через особое положение, когда отрезок AB перпендикулярен прямой Dx_3. Движение ползуна из этого положения не детерминировано: ползун может начать двигаться в одну или в другую сторону по направляющей. Такой ситуации конструкторы стараются обычно избегать. Более того, шатун должен быть отклонен от особого положения на достаточно большой угол, чтобы его не заклинило силами трения в паре стойка-ползун.

Запишем условие сохранения признака сборки:

$$l_2 > l_1 + h, \qquad (2)$$

где h - расстояние от точки O до направляющей (рис. 2, б).

Рассмотрим проблему заклинивания ползуна. Благодаря приложенному вращающему моменту T_{OA} на ползун в точке B со стороны шатуна действует сила, пропорциональная величине вращающего момента. Примем гипотезу о том, что превалирующая составляющая этой силы $\vec{F}$ направлена вдоль прямой AB (предположим, что шатун выполняет при передаче усилий роль стержня: он или сжат, или растянут). Угол $\mu = \angle A_2B_2\breve{D}$ между направлением движущей силы и направлением скорости точки ее приложения называют углом передачи (передачи усилия).

Вспомним известную школьную задачу о страгивании тела, опирающегося на шероховатую поверхность. Пусть f (friction) - коэффициент трения скольжения тела по опоре, $\vec{F}$ - сила, которая прижимает и одновременно стремится сдвинуть тело вдоль опорной поверхности (рис. 3). Когда тело начинает движение вправо из состояния покоя, то проекция его ускорения удовлетворяет условию

$$a_x > 0, \qquad (3)$$

и при этом сила трения равна

$$F_{mp} = fN. \qquad (4)$$

Запишем второй закон Ньютона:

$$m\vec{a} = \vec{F} + \vec{F}_{mp} + \vec{N} + m\vec{g}\ .$$

Пренебрежем силой тяжести по сравнению с другими силами и запишем последнее уравнение в проекциях на оси координат:

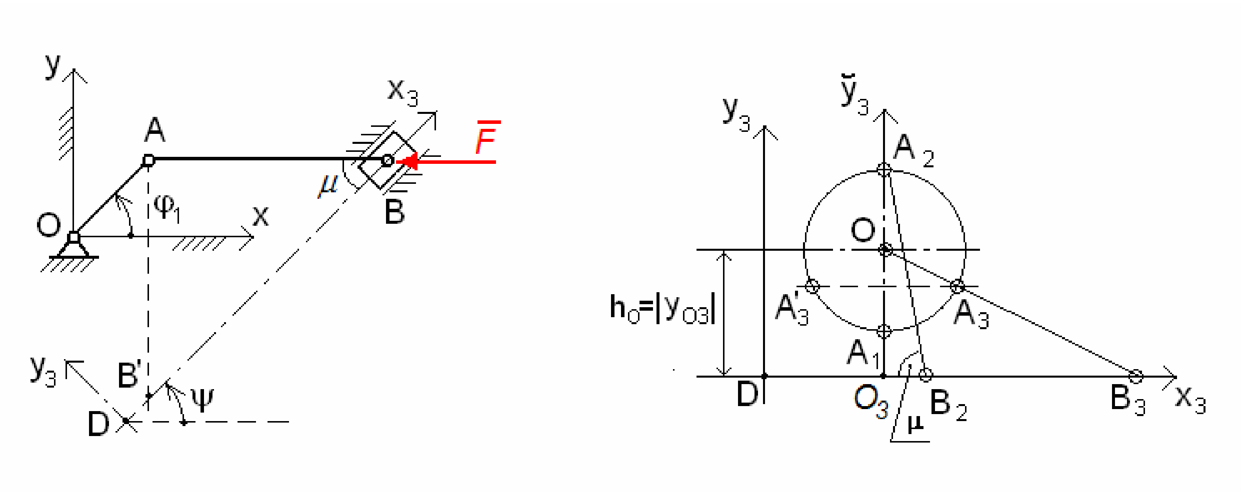
$$ma_x = F\sin\alpha - F_{mp}, \qquad N = F\cos\alpha\ . \tag{5}$$

Подставляя (4) и (5) в (3), получаем условие страгивания тела:

$$tg\alpha > f\ , \text{ или } \alpha = \frac{\pi}{2} - \mu > arctg\, f\ .$$

Величина $arctg\, f$ называется углом трения. Последнее неравенство является еще одним (дополнительным) условием, накладываемым на звенья, т.к. максимальный угол μ выражается через постоянные параметры.

Заметим, что более «естественной» для нашего механизма является система координат $O_3 x_3 \breve{y}_3$ (рис. 2, б). Если решать задачу в этой СК, то следовало бы задать только 3 постоянных параметра: l_1, l_2 и y_{O3} - ординату точки O в СК $O_3 x_3 \breve{y}_3$ при условии, что проекция $\overrightarrow{AB}$ на ось $O_3 x_3$ всегда положительна.



а б

Рис. 2. Кинематическая схема кривошипно-ползунного механизма (а). К расчету крайних положений механизма (б).

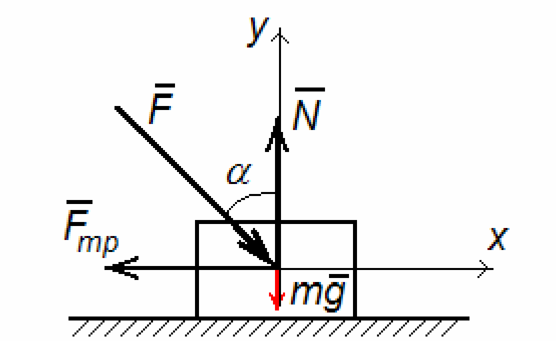
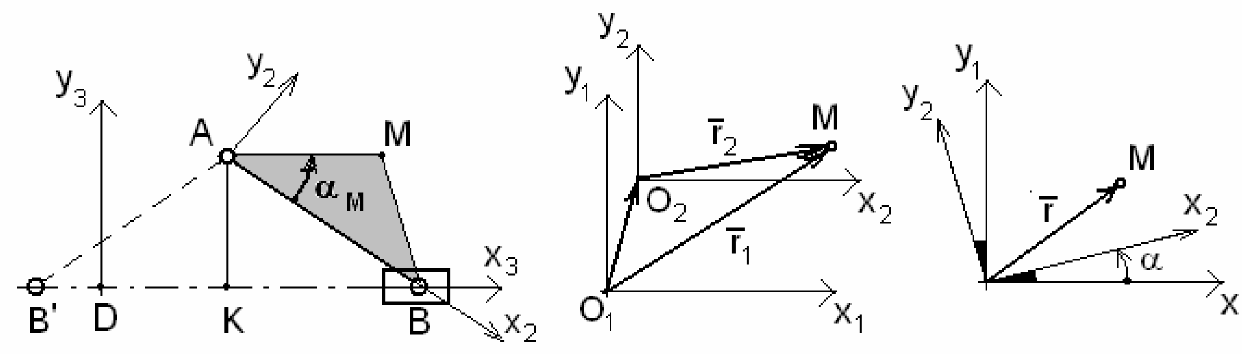


Рис. 3. К задаче о страгивании тела с шероховатой поверхности.

Постоянные параметры α_M, AM, задающие на шатуне некоторую точку M, изображены на рис. 4, а.



а б в

Рис. 4. Положение точки M на шатуне и две сборки механизма (а). Сдвиг (б) и поворот (в) системы координат.

2.4. Расчет скоростей и ускорений

Скорость точки плоской фигуры [2] считаем по формуле $\vec{v}_A = \vec{v}_C + \vec{v}_{AC}$, где скорость $\vec{v}_{AC}$ находится по формуле Эйлера $\vec{v}_{AC} = \vec{\omega} \times \overrightarrow{CA}$. Можно для этого создать в Матлабе функцию, например:

```
vA = function(vC,om,CA)
vA = vC+cross(om,CA);
end
```

Входные и выходные аргументы такой функции должны быть трехмерными векторами. Можно в число выходных аргументов включить $\vec{v}_{BA}$, поскольку ее полезно запомнить для расчета нормального ускорения $\vec{a}^n_{BA} = \vec{\omega} \times \vec{v}_{BA}$.

Ускорение точки плоской фигуры находим по формуле $\vec{a}_B = \vec{a}_A + \vec{a}_{BA}$, где $\vec{a}_{BA} = \vec{\varepsilon} \times \overrightarrow{AB} + \vec{\omega} \times \vec{v}_{BA}$. Наиболее краткая программа-функция для расчета ускорения оперирует с трехмерными векторами. В приводимом ниже варианте программы под именем `kriv_polz1` вызываются функции расчета двумерных векторов:

```
function vA = vpf(vC,rCA,om);%Расчет скорости точки плоской фигуры
   for k = 1:2
       vA(k,1) = vC(k,1) + (2*k-3)*om*rCA(3-k,1);
   end

function aA = apf(aC,rCA,om,eps);%Расчет ускорения точки плоской фигуры
   for k = 1:2
       aA(k,1) = aC(k,1) + (2*k-3)*eps*rCA(3-k,1) - om*om*rCA(k,1);
   end
```

2.5. Решение задачи на компьютере

2.5.1. Ввод исходных данных

Кинематический анализ механизма выполним путем создания m-файла процедуры `kriv_polz1`. Такую же процедуру предполагается использовать и в задаче синтеза механизма. Механизм в таком случае удобно характеризовать минимальным набором имён. Постоянные параметры механизма зададим вектором $X_{zv} = [l_1\ l_2\ x_D\ y_D\ \psi\ p_{sb}\ AM\ \alpha_M]$. Введем вектор дополнительных постоянных параметров $X_{par} = [h_p\ mv\ ma\ pv\ pa]$. Здесь h_p - высота прямоугольника, изображающего ползун (ширина прямоугольника пусть равна $2h_p$), mv, ma - масштабы стрелок, изображающих скорости и ускорения точек, pv, pa - признаки потребности в расчетах скоростей и ускорений точек механизма (если $pv = 0$, то расчет скоростей не производится). Пусть анализ требуется провести в одном положении $\varphi = \varphi_1$ механизма - зададим вектор переменных параметров $X_{\mathrm{var}} = [\varphi_1 \ \omega_1 \ \varepsilon_1]$. Для ввода исходных данных можно организовать головную программу, которая формирует исходные данные и вызывает процедуру `kriv_polz1`, а можно просто задавать векторы в командном окне. Например, набрать

```
Xzv = [6.25 12. 2.5 -5.4  pi/6 -1 1.34 0.12*pi];<enter>
Xpar = [0.5 1. 1. 1 1]; <enter>
Xvar = [0. 10. -2.5]; <enter>
```

Это векторы-строки. Элементы отделяем пробелами, или запятыми, или запятыми с пробелами. Можно хранить исходные данные в бинарном или текстовом файле и извлекать их оттуда.

Углы вводим в радианах. Если углы вводить в градусах, то их в программе при вызове функций «`cos`» и «`sin`» следует перевести в радианы, умножая на «pi» и деля на 180.

2.5.2. Анализ исходных данных

Выясним, может ли существовать механизм при введенных постоянных параметрах. Ведь если направляющая ползуна Рассмотрим ближайшее к направляющей Dx_3 положение A_1 шарнира A. Механизм не существует, если в этом положении шатун не «достает» до направляющей, а если и достает в точке O_3, то движение невозможно. Для существования механизма необходимо условие

$$l_2 > h - l_1, \qquad (6)$$

иначе имеем разрыв кинематической цепи. Заметим, что разрыв цепи может произойти и в некотором диапазоне изменения угла φ, когда расстояние от точки A до направляющей окажется короче шатуна.

Это расстояние, как и величину h, будем искать как модуль ординаты точки O в системе координат Dx_3y_3 (СК$_3$). Рассчитаем координаты точки O в СК$_3$, используя матричные преобразования, которые очень быстро выполняются в Матлабе. Пусть матрица-столбец (вектор-столбец) абсолютных координат точки O есть $X_O = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. СК$_3$ сдвинута относительно СК$_0$ на вектор $\overrightarrow{OD}$ и повернута на угол ψ вокруг оси Dz_3 против часовой стрелки. Координаты вектора $\overrightarrow{OD}$ представим в виде столбца $X_D = \begin{pmatrix} x_D \\ y_D \end{pmatrix}$. Найдем вектор $X_{O,3}$ координат точки O в СК$_3$.

Вспомним правила преобразования координат точки при переходе от одной СК к другой.

А) Пусть СК2 $O_2x_2y_2$ сдвинута (перенесена параллельно) относительно СК1 на вектор $\overrightarrow{O_1O_2}$ (рис. 4, б). Создадим вектор сдвига (shift) S - вектор-столбец координат точки O_2 в базисе СК1: $S=\begin{pmatrix} x_{O_2} \\ y_{O_2} \end{pmatrix}$. Обозначим: $X_1=\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ вектор-столбец координат точки M в СК1, $X_2=\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ - вектор-столбец координат этой точки в СК2. Из рис. 4, б следует, что $\vec{r}_1=\overrightarrow{O_1O_2}+\vec{r}_2$, откуда следует $\vec{r}_2=\vec{r}_1-\overrightarrow{O_1O_2}$. В матричной форме последняя операция выглядит следующим образом:

$$X_2=X_1-S. \tag{7}$$

Формула (7) – формула пересчета координат точки M из СК1в СК2. Формула обратного пересчета из СК2 в СК1:

$$X_1=X_2+S. \tag{8}$$

Б) Пусть СК3 повернута относительно СК2 на угол α (рис. 4, в) вокруг оси Oz. Один и тот же радиус-вектор точки M представим в двух базисах:

$$x_2\vec{i}_2+y_2\vec{j}_2=x_3\vec{i}_3+y_3\vec{j}_3. \tag{9}$$

Глядя на рис. 4, в, составим таблицу результатов скалярного умножения ортов (единичных векторов осей) одной системы координат на орты другой системы. Скалярное произведение ортов есть косинус угла между ними, т.к. модули их равны 1. Таблица 1 есть таблица направляющих косинусов:

Таблица 1. Направляющие косинусы углов между координатными осями

	i_2	j_2
i_3	cosα	sinα
j_3	-sinα	cosα

Умножив обе части равенства (9) скалярно на $\vec{i}_3$, получим

$$x_3=x_2\cos\alpha+y_2\sin\alpha,$$

а умножив на $\vec{j}_3$, получим

$$y_3=-x_2\sin\alpha+y_2\cos\alpha.$$

Систему двух последних равенств можно записать (проверьте это!) в матричной форме:

$$\begin{pmatrix} x_3 \\ y_3 \end{pmatrix}=\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}\cdot\begin{pmatrix} x_2 \\ y_2 \end{pmatrix},$$

или

$$X_3=RX_2, \tag{10}$$

где $R=\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}$- матрица направляющих косинусов (DCM – direction cos. matrix) или матрица поворота от СК2 к СК3. Формула (10) служит для перерасчета координат точки M из СК2 в СК3. Подставив (7) в (10), получим матричную формулу преобразования координат из СК1 в СК3:

$$X_3=R(X_1-S). \tag{11}$$

Чтобы получить обратное преобразование, надо решить векторное уравнение (1) относительно X_1. Для этого надо сделать коэффициент при X_1 равным единичной матрице. Умножаем обе части уравнения (1) на матрицу R^{-1}, обратную матрице R. Матрицу

R^{-1}можно получить так же, как была получена матрица R (умножить части уравнения (9) скалярно на $\vec{i}_2$, затем на $\vec{j}_2$) или по известным правилам расчета матрицы, обратной к заданной. При этом важно отметить, что матрица DCM обладает примечательными свойствами, в частности, ее определитель равен единице. Так что матрица R^{-1} есть транспонированная (повернутая вокруг главной диагонали) матрица R:

$$R^{-1} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}.$$

Иначе говоря, матрица R^{-1} выражает идею поворота от СК3 к СК2 вокруг оси Oz на угол ($-\alpha$). Итак, обратное к (1) преобразование имеет вид:

$$X_1 = R^{-1} X_3 + S. \qquad (12)$$

Пеобразования векторов (координат точек) при сдвигах и поворотах («движениях») базисов в пространстве выполняются по формулам, аналогичным (11) и (12), только векторы сдвига и матрицы поворота должны быть трехмерными. Трехмерная матрица A_z поворота вокруг оси Oz будет иметь следующий вид: $A_z = \begin{pmatrix} \cos\alpha_z & \sin\alpha_z & 0 \\ -\sin\alpha_z & \cos\alpha_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$. Матрицы поворотов вокруг оси Ox на угол α_x и вокруг оси Oy на угол α_y выглядят так:

$$A_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_x & \sin\alpha_x \\ 0 & -\sin\alpha_x & \cos\alpha_x \end{pmatrix}, \; A_y = \begin{pmatrix} \cos\alpha_y & 0 & -\sin\alpha_y \\ 0 & 1 & 0 \\ \sin\alpha_x & 0 & \cos\alpha_x \end{pmatrix}.$$

Чтобы найти расстояние h для формулы (6), создаем матрицу $R_{03} = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix}$ и по формуле (11) рассчитываем координаты точки O в системе Dx_3y_3:

$$X_{O,3} = R_{03}(X_O - X_D).$$

Тогда $h = \left| X_{O,3}(2) \right|$.

Начнем составлять программу анализа `kriv_polz1` в окне `Editor`.

```
%Procedure kriv_polz - Кривошипно ползунный механизм: кинематич. Анализ
%Xzv = [l1 l2 xD yD psi prsb AM alfABAM]-Вектор величин звеньев
l1 = Xzv(1); l2 = Xzv(2); XD = [Xzv(3); Xzv(4)]; psb = Xzv(6);
cpsi = cos(Xzv(5));
spsi = sin(Xzv(5));
R03 = [cpsi spsi; -spsi cpsi];
U30 = [cpsi -spsi; spsi cpsi];
XO = [0.; 0.];%вектор-столбец абс. координат точки O
XO3 = R03*(XO-XD);%персчет координат точки O в СК3 (сдвиг, затем поворотот)
ho = abs(XO3(2,1));%расстояние от точки O до направляющей ползуна
   if l2<=ho-l1
      error('Механизм не существует')%см. рис. и рассуждение
   end%if
   if l2<=ho+l1
      warning('Звено 1 - коромысло; возможен разрыв цепи')
  end%if
```

В ситуации «error» на дисплей выводится сообщение в апострофах и работа программы прерывается. В ситуации «warning» после вывода сообщения выполнение программы продолжается. Продолжим «распаковку» исходных данных: Если с величинами предстоит еще работать, переобозначим их. Предполагая ось $Ax_2 = AB$, рассчитываем координаты точки M шатуна в СК2:

```
XM2 = [Xzv(8)*cos(Xzv(7)); Xzv(8)*sin(Xzv(7))];%коорд. Т М в СК2
h = Xpar(1);%Толщина ползуна в высраиваемой фигуре
mv = Xpar(3); ma = Xpar(4); pv = Xpar(5); pa = Xpar(6);%Масштабы
b = 2*h;%Длина ползуна
sz = l1+l2+b+1.;%протяженность экрана по x,y
figure, axis equal, axis([-sz, sz, -sz, sz]),
fill([0. -1. 1.],[0. -1. -1.],[1 1 1]), hold on; %задавшись вектором
%абсцисс и вектором ординат вершин, рисуем белый треугольник (стойку)
%Вектор [1 1 1]- вектор белого цвета (в RGB). Черный цвет [0 0 0].
%масштабы осей одинаковы, оси простираются в границах [-sz, sz]
%объявляем открывшуюся фигуру открытой и для дальнейших построений
```

Рассмотрим некоторые приемы построения изображения механизма. Направляющую ползуна изобразим в виде отрезка прямой. проходящей через наиболее удаленные положения ползуна B_3 (рис. 2, б) и B_3'. Строим матрицу столбцов-координат точек B_3, B_3' в СК03 и в СК0. Множитель 3-2*k формирует знак «+» при k=1 и «-» при k=-1.

```
 for k = 1:2
     XBextr(:,k) = [XO3(1,1)+(3-2*k)*sqrt((l1+l2)*(l1+l2)-ho*ho); 0];
     %коорд в СК30 точки B3
     XBextr(:,k) = U30*XBextr(:,k) + XD;% персчет этих координат в СК0
 end%for k
plot(XBextr(1,:),XBextr(2,:),'-k^');%рисуем сплошной (-) черной (k) линией
%направляющую ползуна; крайние точки D и B(extr)помечаем треугольными
%маркерами
axis equal, axis([-sz, sz, -sz, sz]), hold on;
fi1 = Xvar(1);%Начинаем строить  механизм в положении fi1
XA = [l1*cos(fi1); l1*sin(fi1)];%абс. координаты точки A
XA3 = R03*(XA-XD);%координаты точки A в СК30
ll = l2*l2 - XA3(2,1)*XA3(2,1);%квадрат расстояния от основания
%перпендикуляра, опущ. из т.А на направл. ползуна, до т.В
   if ll<0
       error('Разрыв цепи')%Счет прекращается!
   end%if
xb3 = XA3(1,1) + psb*sqrt(ll);%абсцисса т.В в СК3
XB3 = [xb3; 0];%вектор коодинат т.В в СК3
Xp3 = [xb3-b xb3+b xb3+b xb3-b; -h -h h h];%вектор координат вершин
%прямоугольника, изображающего ползун, в СК3
XB = U30*XB3 + XD;%вектор абс. координат т.В
   for m = 1:4
       Xp(:,m) = U30*Xp3(:,m) + XD;%пересчет координат вершин ползуна в СК0
   end%m
XAB = XB - XA;%Вектор AB в абс. СК0
cfi2 = XAB(1)/l2; sfi2 = XAB(2)/l2;%cos и sin угла fi2, образованного
%вектором AB с осью Ox
Ur20 = [cfi2 -sfi2; sfi2 cfi2];%матрица перехода (поворота) от СК2, связан-
%ной со звеном 2 (ось Ax2 направлена от  А к В), к абс. СК0
XM = XA + Ur20*XM2;%Абс. координаты точки М шатуна в k-ом положении
XABM = [XA(1) XB(1) XM];%Вектор абсцисс вершин треуг-ка ABM (шатуна)
YABM = [XA(2) XB(2) XM];%Вектор ординат его вершин
fill(XABM,YABM,[1 0.8 1]); axis equal, axis([-sz, sz, -sz, sz]), hold on;
%Нарисовали розовый треугольник ABM
patch(Xp(1,:),Xp(2,:),[1 1 1],'LineWidth',2); %рисуем белый ползун как
```

```
%"заплатку" (patch)
axis equal, axis([-sz, sz, -sz, sz]), hold on;
X = [XO(1,1) XA(1,1) XB(1,1)];%вектор-строка абсцисс точек O,A,B
Y = [XO(2,1) XA(2,1) XB(2,1)];%их ординаты
plot(X,Y,'-ro','LineWidth',3,'MarkerEdgeColor','k','MarkerFaceColor','w','MarkerSize',8);
%Нарисовали сплошную красную (r) поманую OAB толщиной 3 мм, изображающую
%отрезки OA и AB. Шарниры изобразили как кружочки-маркеры размером 8 пикс
%граница кружка черная, внутренность (Face) - белая
axis equal, axis([-sz, sz, -sz, sz]), hold on; (И т.д.)
```

2.5.3. Коды программ

Ниже приведены тексты двух вариантов программ: `kriv_polz` и `kriv_polz1`. Исходные данные вводятся в них по-разному. С помощью команды `F(k) = getframe` создаются кадры анимационного фильма, демонстрирующего движение механизма. Добавлены звуковые сигналы.

kriv_polz

```
%Procedure kriv_polz - Кривошипно ползунный механизм: расчет и изображение
%положений звеньев, а также траекторий, скоростей и ускорениий точек
%Абсолютная система координат (СК0) Oxy связана со стойкой. Ось Ox
%горизонтальна. Все системы координат - правые. Координаты точек,
%скорости и ускорения в итоге вычисляются отн. абс. системы координат
%Вводим исходные данные из векторов Xzv, Xpar
%Xzv = [l1 l2 xD yD psi prsb AM alfABAM]-Вектор величин звеньев
l1 = Xzv(1); l2 = Xzv(2); XD = [Xzv(3); Xzv(4)]; psb = Xzv(6);
cpsi = cos(Xzv(5));
spsi = sin(Xzv(5));
R03 = [cpsi spsi; -spsi cpsi];
U30 = [cpsi -spsi; spsi cpsi];
XM2 = [Xzv(8)*cos(Xzv(7)); Xzv(8)*sin(Xzv(7))];
%Xpar = [npoz hpolz mv ma pv pa fi10 omega1 epsilon1=0]-Вектор доп.парам-ов
n = Xpar(1);%Число расчетных положений npoz
h = Xpar(2);%Толщина ползуна на рис.
mv = Xpar(3); ma = Xpar(4); pv = Xpar(5); pa = Xpar(6);%Масштабы изобра-
%жаемых векторов и признаки надобности в расчете скоростей и ускорений.
%Движение механизма считаем установившимся: omega1=const, epsilon1=0.
fi10 = Xpar(7); om1 = Xpar(8); eps1 = Xpar(9);
%Начинаем расчет
b = 2*h;%Длина ползуна
sz = l1+l2+b+1.;%протяженность экрана по x,y
XO = [0.; 0.];%вектор-столбец абс. координат точки O
XO3 = R03*(XO-XD);%персчет координат точки O в СК3 (сдвиг, затем повортот)
ho = abs(XO3(2,1));%расстояние от точки O до направляющей ползуна -
%модуль ординаты yO в СК3
   if l2<=ho-l1
      error('Механизм не существует')%см. рис. и рассуждение
   end%if
   if l2<=ho+l1
      warning('Звено 1 не кривошип, а коромысло'))%если мы хотим иметь
      %именно кривошип, надо вместо warning написать error; тогда
      %дальнейшие расчеты не выполняются
   end%if
XBex3 = [XO3(1,1)+sqrt((l1+l2)*(l1+l2)-ho*ho); 0];%координаты экстремаль-
%ной точки B(ex) ползуна (когда кривошип и шатун коллинеарны) в СК3
XBex = U30*XBex3 + XD;% персчет этих координат в СК0
Xguid = [XD(1,1) XBex(1,1)];%вектор-строка абсолютных абсцисс двух точек,
%через которые проходит направляющая ползуна: D и B(ex)
Yguid = [XD(2,1) XBex(2,1)];%вектор-строка ординат этих точек
fi1 = fi10; dfi = 2*pi/n; %назначаем начальное значение угла fi1 поворота
%кривошипа и шаг (fi1 отсчитываем от оси Ox к вектору OA против часовой
%стрелки
friq = 0.3; dfriq = 0.5/n;%начальная частота звукового сигнала и шаг
for k = 1:n+1%расчет положений механизма
XA = [l1*cos(fi1); l1*sin(fi1)];%абс. координаты точки A
XA3 = R03*(XA-XD);%координаты точки A в СК3
ll = l2*l2 - XA3(2,1)*XA3(2,1);%квадрат расстояния от основания
%перпендикуляра, опущ. из т.A на направл. ползуна, до т.B
```

```
   if ll<0
       warning('Разрыв цепи')%Если все-таки не обязательно кривошип,
       %то пропускаем дальнейший расчет механизма в данном положении,
       %когда имеет место разрыв кинематической цепи
       continue%управление передаем следующему номеру k
   end%if
xb3 = XA3(1,1) + psb*sqrt(ll);%абсцисса т.В в СК3
XB3 = [xb3; 0];%вектор коодинат т.В в СК3
Xp3 = [xb3-b xb3+b xb3+b xb3-b; -h -h h h];%вектор координат вершин
%прямоугольника, изображающего ползун, в СК3
XB = U30*XB3 + XD;%вектор абс. координат т.В
   for m = 1:4
       Xp(:,m) = U30*Xp3(:,m) + XD;%пересчет координат вершин ползуна в СК0
   end%m
note=(0:friq:90)*pi;%создаем спектр звукового сигнала
X = [XO(1,1) XA(1,1) XB(1,1)];%вектор-строка абсцисс точек O,A,B
Y = [XO(2,1) XA(2,1) XB(2,1)];%их ординаты
fill([0. -1. 1.],[0. -1. -1.],[1 1 1]);%задавшись вектором абсцисс и векто-
%ром ординат вершин, рисуем белый треугольник (стойку)
%Вектор [1 1 1]- вектор белого цвета (в RGB). Черный цвет [0 0 0].
axis equal, axis([-sz, sz, -sz, sz]), hold on;%масштабы осей одинаковы,
%простираются оси на рис. в границах [-sz, sz],
%объявляем появившуюся фигуру открытой и для дальнейших построений
plot(Xguid,Yguid,'-k^');%рисуем сплошной (-) черной (k) линией направляющую
%ползуна; крайние точки D и B(extr)помечаем треугольными маркерами
axis equal, axis([-sz, sz, -sz, sz]), hold on;
XAB = XB - XA;%Вектор AB в абс. СК0
cfi2 = XAB(1)/l2; sfi2 = XAB(2)/l2;%cos и sin угла fi2, образованного
%вектором AB с осью Ox
Ur20 = [cfi2 -sfi2; sfi2 cfi2];%матрица перехода (поворота) от СК2, связан-
%ной со звенм 2 (ось Ax2 направлена от  A к B), к абс. СК0
XM(:,k) = XA + Ur20*XM2;%Абс. координаты точки M шатуна в k-ом положении
Xfil = [XA(1) XB(1) XM(1,k)];%Вектор абсцисс вершин треуг-ка ABM (шатуна)
Yfil = [XA(2) XB(2) XM(2,k)];%Вектор ординат его вершин
fill(Xfil,Yfil,[1 0.8 1]); axis equal, axis([-sz, sz, -sz, sz]), hold on;
%Нарисовали розовый треугольник ABM
patch(Xp(1,:),Xp(2,:),[1 1 1],'LineWidth',2); %рисуем белый ползун как
%"заплатку" (patch)
axis equal, axis([-sz, sz, -sz, sz]), hold on;
plot(X,Y,'ro','LineWidth',3,'MarkerEdgeColor','k','MarkerFaceColor',...
      'w','MarkerSize',8);
%Нарисовали сплошную красную (r) поманую OAB толщиной 3 мм, изображающую
%отрезки OA и AB. Шарниры изобразили как кружочки-маркеры размером 8 пикс
%граница кружка черная, внутренность (Face) - белая
axis equal, axis([-sz, sz, -sz, sz]), hold on;
sound([sin(note)])%выводим гармонический звуковой сигнал
plot(XM(1,1:k),XM(2,1:k),'-go');%рисуем сплошной зеленой (g) линией часть
%траекторииточки M от первого до k-го ее положения, помечая эти положения
%зелеными же кружочками
axis equal, axis([-sz, sz, -sz, sz]), hold on;
  if pv%т.е. если признак расчета скорсти не равен нулю (в логич. выражении
      %число 0 читается как логический нуль ("нет"), другие числа - как
      %логичексая 1 ("да").
      vA = vpf([0; 0],XA,om1);%Применяем созданную нами функцию vpf расчета
      %скорости точки плоской фигуры для вычисл-ия абс. проекций вектора vA
      %XAB3 = XB3 - XA3;%Вектор AB в СК3, связанной с направляющей ползуна
      %vA3 = R03*vA;%Вектор vA в СК3. Относительно метрики звеньев скорости-
      %векторы свободные; сдвига на вектор XD делать не надо!
      %om2 = -vA3(2)/XAB3(1);%Согласно условию связи, прекция вектора vB на
      %ось Dy3 равна нулю. Применим функцию vpf для расчета vB в СК3.
      %Приравняв к нулю выражение для vB(2) в теле функции  vpf, которое и
      %будет прекцией vB на ось Dy3, находим om2 -проекцию вектроа omega2
      %на ось Oz.
      vB = vpf(vA,XAB,om2);%Зная om2, находим абс. прекции вектра vB
      quiver(XA(1),XA(2),vA(1)*mv,vA(2)*mv, '-g','LineWidth',2);%Строим
      %стрелку, изображающую вектор vA. XA(1),XA(2)- коодинаты начала
      %стрелки, XA(2),vA(1)*mv,vA(2)*mv -проекции стрелки-вектора,
      %'-g','LineWidth',2 -стрелка зеленая толщиной 2 мм.
      axis equal, axis([-sz, sz, -sz, sz]), hold on;
      quiver(XB(1),XB(2),vB(1)*mv,vB(2)*mv, '-g','LineWidth',2);%Строим vB
      axis equal, axis([-sz, sz, -sz, sz]), hold on;
```

```
  end%ifpv
  if pv&&pa%Если хотим рассчитать ускорения, надо сначала рассчитать
      %скорости (угловые скорости звеньев)
      aA = apf([0; 0],XA,om1,eps1);%aA в абс. СК0
      aA3 = R03*aA;%aA в СК3
      eps2 = (om2*om2*XAB3(2) - aA3(2))/XAB3(1);%записываем формулу расчета
      %aB3 в СК3 (см. тело нашей функции apf); проекция aB на ось Dy3 равна
      %нулю. Из этого условия получаем формулу для eps2.
      aB = apf(aA,XAB,om2,eps2);%Зная eps2, находим абс. прекции вектра aB
      quiver(XA(1),XA(2),aA(1)*ma,aA(2)*ma, '-y','LineWidth',2);
      axis equal, axis([-sz, sz, -sz, sz]), hold on;
      quiver(XB(1),XB(2),aB(1)*ma,aB(2)*ma, '-y','LineWidth',2);
      axis equal, axis([-sz, sz, -sz, sz]);
  end%ifpv&&pa
hold off;%закрываем фигуру для дальнейших добавлений в нее
F(k) = getframe;%Запоминаем k-ую фигуру как отдельный кадр (frame) фильма
fi1 = fi1 + dfi;%Следующее значение угла fi1
friq = friq + dfriq;%Следующая частота звука
end%k
%Если потом захотим 5 раз подряд просмотреть фильм, изображающий движение
%механизма, наберем команду movie(F,5)<Enter>
%Чтобы рассмотреть отдельно кадр №3, наберем movie(F(3))
```

kriv_polz1

```
%Procedure kriv_polz1 - Анализ кривошипно ползунного механизма
%Ведущее звено вращается равнопеременно или равномерно
%Исходные данные вводятся интерактивно
novar = input('Введи номер варианта = ');
l1 = input('Введи l1 = OA = ');
l2 = input('Введи l2 = AB = ');
xD = input('Введи xD = ');
yD = input('Введи yD = ');
XD = [xD; yD];
psb = input('Введи признак сборки диады ВВП: psb = ');
AM = input('Введи AM = ');
prizgrad = input('Введи признак задания углов в градусах prizgrad = ');
psi = input('Введи psi = ');
fiABAM = input('Введи угол от AB к AM fiABAM = ');
if prizgrad
   psi = psi*pi/180;
   fiABAM = fiABAM*pi/180;
end%if prizgrad
cpsi = cos(psi);
spsi = sin(psi);
R03 = [cpsi spsi; -spsi cpsi];
U30 = [cpsi -spsi; spsi cpsi];
XM2 = [AM*cos(fiABAM); AM*sin(fiABAM)];
n = input('Введи число расчетных положений n = ');
h = input('Введи толщину ползуна в изображаемой фигуре h = ');
mv = input('Введи масштаб изображения вектора скорости mv = ');
ma = input('Введи масштаб изображения вектора ускорения ma = ');
pv = input('Введи признак необходимости расчета скоростей pv = ');
pa = input('Введи признак необходимости расчета ускорений pa = ');
fi10 = input('Введи начал. угол поворота звена 1: fi10=fiOA,0 = ');
if prizgrad
   fi10 = fi10*pi/180;
end%if prizgrad
om10 = input('Введи начал. угл. скор. звена 1: om10=omOA,0 = ');
eps1 = input('Введи постоянное угловое ускорение зв. 1: eps1=epsOA = ');
if eps1~=0
   dt = input('Введи шаг по времени dt = ');
else dfi = 2*pi/n;
end%if eps1~=0
%Начинаем расчет
b = 2*h;%Длина ползуна
sz = l1+l2+b+1.;%протяженность экрана по x,y
XO = [0.; 0.];
XO3 = R03*(XO-XD);
ho = abs(XO3(2,1));
   if l2<=ho-l1
```

```
      error('Механизм не существует')
   end%if
   if l2<=ho+l1
      warning('Звено 1 не кривошип, а коромысло')
   end%if
  for k = 1:2
     XBextr(:,k) = [XO3(1,1)+(3-2*k)*sqrt((l1+l2)*(l1+l2)-ho*ho); 0];
     %коорд в СК30 точки B3
     XBextr(:,k) = U30*XBextr(:,k) + XD;% персчет этих координат в СК0
  end%for k
fi1 = fi10; om1 = om10; friq = 0.3; dfriq = 0.5/n;
  if eps1~=0
     t = 0.;
  end%if eps1~=0
for k = 1:n+1
K = k%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
XA = [l1*cos(fi1); l1*sin(fi1)];
XA3 = R03*(XA-XD);
ll = l2*l2 - XA3(2,1)*XA3(2,1);
   if ll<0
       warning('Разрыв цепи')
       continue
   end%if
xb3 = XA3(1,1) + psb*sqrt(ll);
XB3 = [xb3; 0];
Xp3 = [xb3-b xb3+b xb3+b xb3-b; -h -h h h];
XB = U30*XB3 + XD;
   for m = 1:4
       Xp(:,m) = U30*Xp3(:,m) + XD;
   end%m
note=(0:friq:90)*pi;
X = [XO(1,1) XA(1,1) XB(1,1)];
Y = [XO(2,1) XA(2,1) XB(2,1)];
patch([0. -h h],[0. -h -h],[1 1 1],'LineWidth',2);
axis equal, axis([-sz, sz, -sz, sz]), hold on;
plot(XBextr(1,:),XBextr(2,:),'-k^');%рисуем сплошной (-) черной (k) линией
%направляющую ползуна; крайние точки D и B(extr)помечаем треугольными
%маркерами
axis equal, axis([-sz, sz, -sz, sz]), hold on;
XAB = XB - XA;
cfi2 = XAB(1)/l2; sfi2 = XAB(2)/l2;
Ur20 = [cfi2 -sfi2; sfi2 cfi2];
XM(:,k) = XA + Ur20*XM2;
Xfil = [XA(1) XB(1) XM(1,k)];
Yfil = [XA(2) XB(2) XM(2,k)];
fill(Xfil,Yfil,[1 0.8 1]); axis equal, axis([-sz, sz, -sz, sz]), hold on;
patch(Xp(1,:),Xp(2,:),[1 1 1],'LineWidth',2);
axis equal, axis([-sz, sz, -sz, sz]), hold on;
plot(X,Y,'-
ko','LineWidth',3,'MarkerEdgeColor','k','MarkerFaceColor','w','MarkerSize',8);
axis equal, axis([-sz, sz, -sz, sz]), hold on;
sound([sin(note)])
plot(XM(1,1:k),XM(2,1:k),'-go');
axis equal, axis([-sz, sz, -sz, sz]), hold on;
  if pv
      vA = vpf([0; 0],XA,om1);
      XAB3 = XB3 - XA3;
      vA3 = R03*vA;
      om2 = -vA3(2)/XAB3(1)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      vB = vpf(vA,XAB,om2)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      quiver(XA(1),XA(2),vA(1)*mv,vA(2)*mv, '-g','LineWidth',2);
      axis equal, axis([-sz, sz, -sz, sz]), hold on;
      quiver(XB(1),XB(2),vB(1)*mv,vB(2)*mv, '-g','LineWidth',2);
      axis equal, axis([-sz, sz, -sz, sz]), hold on;
      XAM = XM(:,k) - XA;
      vM = vpf(vA,XAM,om2)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      quiver(XM(1),XM(2),vM(1)*mv,vM(2)*mv, '-g','LineWidth',2);
      axis equal, axis([-sz, sz, -sz, sz]), hold on;
  end%ifpv
  if pv&&pa
```

```
        aA = apf([0; 0],XA,om1,eps1);
        aA3 = R03*aA;
        eps2 = (om2*om2*XAB3(2) - aA3(2))/XAB3(1)%%%%%%%%%%%%%%%%%%%%
        aB = apf(aA,XAB,om2,eps2)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        quiver(XA(1),XA(2),aA(1)*ma,aA(2)*ma, '-r','LineWidth',2);
        axis equal, axis([-sz, sz, -sz, sz]), hold on;
        quiver(XB(1),XB(2),aB(1)*ma,aB(2)*ma, '-r','LineWidth',2);
        axis equal, axis([-sz, sz, -sz, sz]);
        aM = apf(aA,XAM,om2,eps2)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        quiver(XM(1),XM(2),aM(1)*ma,aM(2)*ma, '-r','LineWidth',2);
        axis equal, axis([-sz, sz, -sz, sz]), hold off;
    end%ifpv&&pa
F(k) = getframe;
if eps1~=0
   t = t+dt;
   om1 = om10 + eps1*t;
   fi1 = fi10 + om10*t + 0.5*eps1*t*t;
else
   fi1 = fi1 + dfi;
end%if eps1~=0
friq = friq + dfriq;
end%k
```

2.5.4. Пример счета по программе `kriv_polz1`

```
>> kriv_polz1
Введи номер варианта = 1
Введи l1 = OA = 10
Введи l2 = AB = 40
Введи xD = 0
Введи yD = 0
Введи признак сборки диады ВВП: psb = 1
Введи AM = 20
Введи признак задания углов в градусах prizgrad = 1
Введи psi = 0
Введи угол от AB к AM fiABAM = 45
Введи число расчетных положений n = 5
Введи толщину ползуна в изображаемой фигуре h = 3
Введи масштаб изображения вектора скорости mv = 1
Введи масштаб изображения вектора ускорения ma = 1
Введи признак необходимости расчета скоростей pv = 1
Введи признак необходимости расчета ускорений pa = 1
Введи начал. угол поворота звена 1: fi10=fiOA,0 = 0
Введи начал. угл. скор. звена 1: om10=omOA,0 = 1
Введи постоянное угловое ускорение зв. 1: eps1=epsOA = 2
Введи шаг по времени dt = 0.04
```

шаг K = 1	...	**шаг K = 6**
om2 = -0.2500		om2 = -0.3406
vB = 0 0		vB = -4.1374 0
vM = 3.5355 6.4645		vM = 1.1938 8.5046
eps2 = -0.5000		eps2 = -0.3768
aB = -12.5000 0		aB = -29.3192 -0.0000
aM = -3.8128 12.0450		aM = -20.5251 7.5925

Фильм содержит 6 кадров, последний из них изображен на рис. 5.

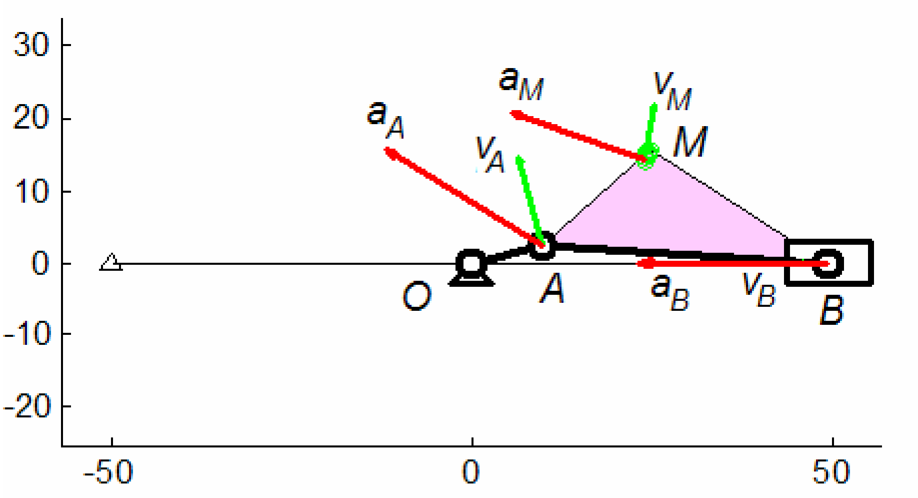


Рис.5. Результат счета по программе `kriv_polz1`. Шестой фрейм из массива F.

Заключение. С помощью упомянутых здесь программ можно проверять решения задач кинематики точки и твердого тела и производить исследования их движений. Программы должны усовершенствоваться студентами в направлении их оптимизации по кодировке, быстродействию и трансляции в иные программные модули. Обширным остается поле графических интерпретаций.

Список литературы

1. Сборник заданий для курсовых работ по теоретической механике : учебное пособие для студентов высших технических учебных заведений / [А.А. Яблонский и др.] ; Под общ. ред. А. А. Яблонского. - Изд. 17-е, стер. - Москва: Интеграл-Пресс, 2010. - 382 с.
2. Яблонский, А. А. Курс теоретической механики/ А. А. Яблонский. Учебник для студентов высших учебных заведений, обучающихся по техническим специальностям. - 15-е изд., стер. - Москва: КноРус, 2010 . - 603 с.
3. Matlab : официальный учебный курс Кембриджского университета / Brian R. Hunt и др. – М.: Изд-во ТРИУМФ, 2008. - 352 с.