

Предисловие

Пособие рассчитано на 8 практических занятий, проводимых в компьютерном классе. Занятие 1 является вводным; занятия 2, 3, 7, 8 сопровождают лабораторные работы; на занятиях 4, 5, 6 выполняется часть курсового проекта (силовой расчёт рычажного механизма). В помощь выполняющим курсовой проект приводится также программа геометрического расчёта зубчатого зацепления (прил. 3).

Предполагается, что лабораторные работы выполняются до одноимённой темы практического занятия. Например, кинематические диаграммы строятся сначала на лабораторной работе. При этом применяются графические методы. Эти же диаграммы строятся на практических занятиях, но получаются аналитическими методами с вычислениями и построениями на компьютере.

Система MatLab избрана потому, что ориентирована на инженерные и научные вычисления. Система проста, как ранний Basic. Это, а также обширные пакеты прикладных программ, обусловило её широчайшую популярность среди студентов. Более чем в 70 ведущих университетах мира и таких отечественных вузах, как МАИ, МГТУ, МГУ, МИФИ применяют MatLab в учебной и научной работе. MatLab включает в себя практически все известные на сегодняшний день методы вычислительной математики.

Отличительная особенность пособия состоит в том, что студент видит, как выводятся формулы и как они программируются. Одно из занятий отводится на программирование анимации исследуемого механизма, что при изучении теории механизмов и машин делается пока редко.

Данное учебное пособие предназначено для обучения теории механизмов и машин. MatLab при этом играет служебную роль: он ускоряет расчёты и за одно и то же время позволяет получить гораздо больше информации, чем даёт калькулятор. Например, силовой расчёт рычажного механизма делается для всего кинематического цикла его работы.

Пособие может быть полезно также при выполнении курсовых и дипломных проектов по другим дисциплинам.

Занятие 1. Введение в MatLab

В отличие от универсальных языков программирования, таких как Паскаль, Си и другие, MatLab (матричная лаборатория) является специализированной системой, созданной для выполнения инженерных и научных расчётов, этим и обусловлен её выбор для решения задач. Из огромных возможностей системы будут рассмотрены пока только те, которые потребуются на ближайших занятиях. Остальное будет добавляться по ходу дела.

Запустите MatLab. Для этого дважды щёлкните левой кнопкой мыши на ярлыке MatLab. После этого появится Рабочий стол системы MatLab, содержащий три окна: Comand Window (командное окно), Workspace (рабочая область) и Comand History (история команд) (рис. 1.1).

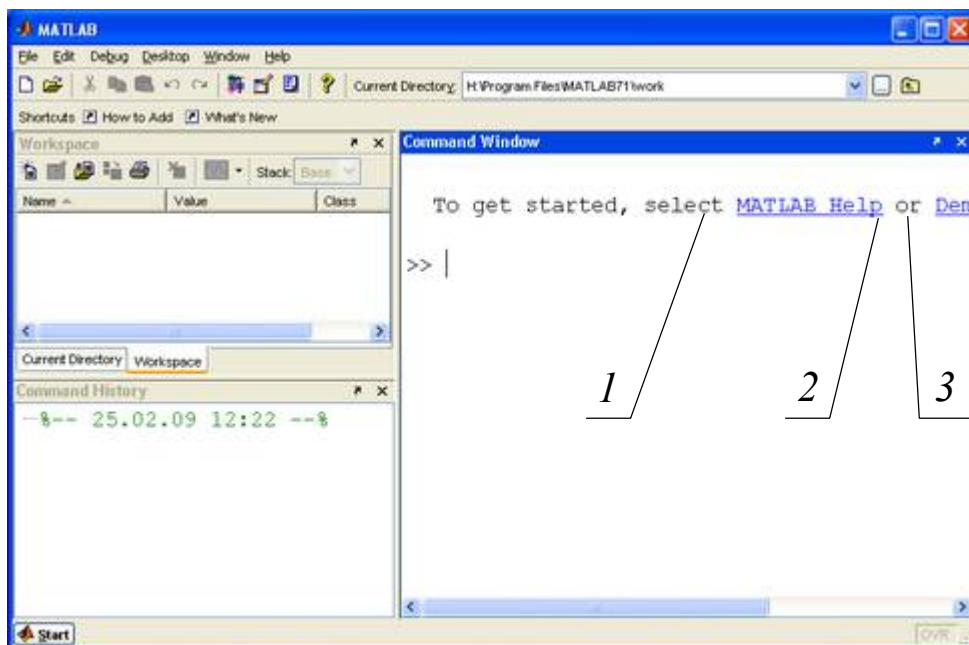


Рис. 1.1

Вместо «щёлкните левой кнопкой мыши на ...» будем говорить далее «щёлкните на ...». Если не сказано «дважды», значит, щёлкнуть надо один раз.

Если что-то будет не так, как на рис. 1.1, щёлкните на Desktop (рабочий стол), Desktop Layout (схема стола), Default (по умолчанию). Проверьте и при необходимости настройте представление информации на экране. Для этого щёлкните на File, Preference, Com-

mand Window. В строке Numeric format выберите short. В строке Numeric display (межстрочный интервал) выберите compact. Щёлкните на крестике возле строки Fonts. Щёлкните на Monospaced (равноширинный), Plain (простой), 14 (пунктов). Щёлкните на ОК.

Теперь вы можете начать вычисления. MatLab позволяет делать это либо в режиме сверхмощного калькулятора – суперкалькулятора, либо в программном режиме.

Работа в режиме суперкалькулятора

Работа в этом режиме ведётся практически так же, как на карманном калькуляторе. Вот несколько примеров, иллюстрирующих этот режим.

В строку приглашения или, иначе, командную строку, обозначенную символом `>>`, введите: $3^2/(1+2)*(6-4)$. Это надо понимать как 3 в квадрате, разделить и т. д. Нажмите на клавишу Enter. В ответ на это на экране появится **ans=6**. На экране результат располагается всегда на строку ниже знака равенства. В данном учебном пособии в целях экономии места всё пишется в одну строку.

В дальнейшем нажатие на клавишу Enter будем обозначать символом `↵`, в результате предложенное выше действие будет выглядеть так: наберите $3^2/(1+2)*(6-4)$ `↵`.

Если в набранной строке вы сделали ошибку и нажали клавишу Enter, то вернуться назад уже невозможно. Чтобы исправить ошибку, используются клавиши `↑` – вверх по списку введенных команд, `↓` – вниз. При этом в строке приглашения последовательно появляются набранные ранее строки. Дойдя до нужной строки, сделайте в ней исправления и нажмите Enter. Не пользуйтесь клавишами `↑`, `↓` для перемещения по экрану: для этого есть клавиши Page Up, Page Down, а также полоса прокрутки.

Слово **ans**, употреблённое выше, зарезервировано системой MatLab для обозначения результатов вычислений по умолчанию и происходит от английского **answer** – ответ.

Кроме **ans** в число зарезервированных слов входит также **pi** – число π (хранится в виде 3,141592653589793). Наберите **b=sin(pi/6)*3** `↵`. После набора получите: **b=1.5000**.

Обозначения синуса и других необходимых нам элементарных функций приведены ниже.

sin(x) , asin(x)	– синус и арксинус x
cos(x) , acos(x)	– косинус и арккосинус x
tan(x) , atan(x)	– тангенс и арктангенс x
sqrt(x)	– корень квадратный из x

Имена функций и команд, встроенных в MatLab, должны набираться непременно малыми буквами. Имена ваших переменных могут быть набраны как большими, так и малыми буквами, однако MatLab их различает. Наберите **D=2; d=3; C=D*d** ↵. После набора получите: **C=6**.

Точка с запятой после **D** и **d** подавляет вывод этих переменных на экран после нажатия на клавишу Enter.

Содержимое списка команд, введённых ранее, называется историей команд. Если этот список вам больше не нужен, последовательно щёлкните на Edit, Clear Comand History. Кроме того, этим же способом вы можете очистить Окно команд и Рабочую область, в последней хранятся ваши переменные. Все эти три области памяти существуют независимо друг от друга, и очистка одной из них не уничтожает содержимое другой.

Все вычисления MatLab производит с двойной (*double*) точностью. Что такое двойная точность, видно на примере числа π . На экран результаты вычислений выводятся с точностью, которая нужна вам. В данном учебном пособии все результаты выведены в формате short – короткий.

MatLab предоставляет исключительно большие возможности по работе с массивами – наборами численных значений какой-либо величины, например аргумента функции.

Примеры задания массивов. Наберите **x=[2 9 5]** ↵, разделяя числа пробелами; или **x=[2,9,5]** ↵, разделяя числа запятыми. В любом случае получите:

x = 2 9 5.

Если массив представляет собой арифметическую прогрессию, то квадратные скобки не обязательны: достаточно воспользоваться двоеточием, разделяющим начальное значение, знаменатель прогрессии и конечное значение. Наберите **x=2:0.5:4** ↵ или **x=[2:0.5 :4]** ↵. В любом случае получите:

x = 2.0000 2.5000 3.0000 3.5000 4.0000 .

Если знаменатель прогрессии равен единице, то указывают лишь начальное и конечное значение массива. Наберите `x=2:4` ↵ или `x=[2:4]` ↵. В обоих случаях получите: `x= 2 3 4`.

Несколько массивов можно группировать в один массив. Наберите `y1=2:4; y2=5:7; y3=[y1 y2]` ↵. После набора получите: `y3= 2 3 4 5 6 7`.

Значение или значения любой вычисленной ранее величины можно вывести на экран без воспроизведения имени этой величины. Наберите `disp(y1)` ↵. После набора получите:

`2 3 4`.

Строчный массив можно превратить (транспонировать) в столбцовый. Транспонирование делает апостроф. Наберите `disp(y1')` ↵. После набора получите:

`2`

`3`

`4`.

Над каждым элементом массива MatLab может производить все элементарные математические операции, порождая массив того же размера. Возможно также поэлементное сложение, вычитание, умножение и деление массива на массив. Одно число – это тоже массив, но только из одного элемента. Примеры поэлементной обработки массивов приведены ниже.

№	Ввод	Результат
1	<code>x = [2 5 4]</code>	<code>x = 2 5 4</code>
2	<code>y1 = x+1</code> или <code>y1 = 1+x</code>	<code>y1 = 3 6 5</code>
3	<code>y2 = x-1</code> или <code>y2 = -1+x</code>	<code>y2 = 1 4 3</code>
4	<code>y3 = x * 2</code> или <code>y3 = 2 * x</code>	<code>y3 = 4 10 8</code>
5	<code>y4 = y3/2</code>	<code>y4 = 2 5 4</code>
6	<code>y5 = sin(x)</code>	<code>y5 = 0.9093 -0.9589 -0.7568</code>
7	<code>y6 = y2.*y3</code>	<code>y6 = 4 40 24</code>
8	<code>y7 = y3./y4</code>	<code>y7 = 2 2 2</code>
9	<code>y8 = x.^2</code>	<code>y8 = 4 25 16</code>
10	<code>y9 = 2./x</code>	<code>y9 = 1.0000 0.4000 0.5000</code>

Обратите особое внимание на строки 7...10 таблицы. Они показывают, что для поэлементного возведения массива в степень (умножения самого на себя), деления числа или массива на массив, а также умножения массива на массив необходимо ставить точку перед знаком названных операций, иначе MatLab воспримет массивы, участвующие в вычислениях, как матрицы, в данном случае строчные, со всеми вытекающими отсюда последствиями. Возведение в квадрат, например, будет воспринято как умножение строчной матрицы на другую такую же. В матричном исчислении это недопустимо, поэтому будет выдано сообщение об ошибке.

Несколько больших массивов выводят на экран, превращая их в столбцы и одновременно объединяя в ещё один массив. Наберите `x=0:pi/6:2*pi ; y1=sin (x) ; y2=cos (2*x)/2; disp ([x' y1' y2'])` ↵. После набора получите:

0	0	0.5000
0.5236	0.5000	0.2500
1.0472	0.8660	-0.2500
1.5708	1.0000	-0.5000
2.0944	0.8660	-0.2500
2.6180	0.5000	0.2500
3.1416	0.0000	0.5000
3.6652	-0.5000	0.2500
.....
5.7596	-0.5000	0.2500
6.2832	-0.0000	0.5000

В первой колонке находится массив **x**, во второй – **y1**, в третьей – **y2**. В каждом массиве 12+1 элементов.

Чтобы по вычисленным значениям построить графики, наберите `plot(x, y1, x, y2), grid` ↵. После набора получите (рис. 1.2).

Команда `plot` рисует кривые. Из-за большого шага аргумента **x** кривые получились ломаными.

Параметрами или аргументами команды `plot` являются **x**, **y1**, **x**, **y2**. Как видим, для каждой кривой указывают имя её аргумента и имя функции. Повторение имени аргумента предусмотрено на случай разных аргументов, например: `plot(x1, y1, x2, y2)`.

Аргументы любой функции и любой команды заключают в круглые скобки и, в отличие от элементов массива, обязательно разделяют запятыми.

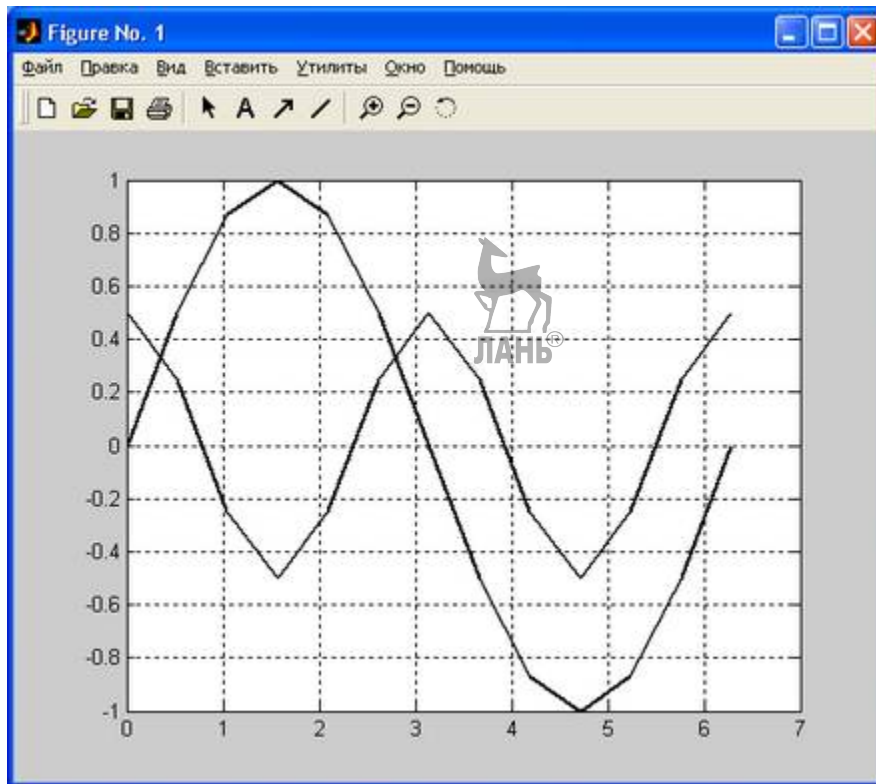


Рис. 1.2

Команда **grid**, следующая после **plot**, наносит сетку. Эта команда – без аргументов.

Режим суперкалькулятора хорош только для разового решения задач с небольшим числом команд. Задачи кинематического анализа, с которых мы начнём, к таковым не относятся. Но мы потратили время не зря: всё, что здесь рассмотрено, применяется и в программном режиме. Возможности программного режима неизмеримо шире.

Программный режим

Составление программы. Щелчком на Edit, Clear ... очистите Command Window, Command History и Workspace.

Сделайте текущей папку, в которой будете сохранять свою программу. Для этого в окне Current Directory (текущая папка) щёлкните справа на значке 3 (см. рис. 1.1). В открывшемся дереве каталогов (папок) последовательно щёлкните на ... Мои документы, MatlabUsers, ОК. Убедитесь, что в окне 1 появилось ... MatlabUsers. В дальнейшем достаточно будет щёлкнуть на значке 2 и выбрать MatlabUsers.

Запустите Editor (редактор-отладчик). Для этого последовательно щёлкните на File, New, M-File. В открывшемся окне редактора (с заголовком Untitled – без названия) можно набирать программу. Строки программы будут нумероваться автоматически. В отличие от окна команд, здесь можно делать исправления в любой ранее набранной строке. Стрелки \uparrow , \downarrow при этом используются по своему обычному назначению – для перемещения по тексту вверх, вниз. Пусть набираемой будет программа вычисления и построения диаграмм, показанных на рис. 1.2.

Чтобы диаграммы получились более гладкими, вычисления произведём с шагом 3° или, иначе, $\pi/60$ рад. Определение числа элементов в массиве x возложим на MatLab. Система делает это с помощью функции **length**.

В таблицу выведем только каждое десятое значение переменных. Таблицу снабдим заголовком «Результаты вычислений», а также подзаголовком с именами выводимых величин. С комментариями, которые располагаются после знака % и могут не вводиться, программа будет иметь следующий вид.

```

1  x=0:pi/60:2*pi;  y1=sin (x); y2=cos (2*x)/2 ;
3  plot(x, y1, x, y2), grid
4  L=length(x); % Длина массива x (число элементов)
5  i=1:10:L; % Индекс каждого 10-го значения переменной
6  num=0:(L-1)/10; % Номера строк таблицы, начиная с нуля
7  disp('.....Результаты вычислений') % Заголовок таблицы
8  disp('.....num.....y1.....y2') % Подзаголовки
9  disp([num' y1(i)' y2(i)']) % Вывод num, y1, y2 столбцами

```

При наборе строк 7, 8 заменяйте точки пробелами. При использовании формата short количество пробелов определяют из расчёта: 10 разрядов (мест) на одну колонку таблицы. Так, например, перед словом num в строке 8 вы видите 7 точек, что в сумме с количеством букв в слове num равно 10.

Сохранение и запуск программы. Набранную вами программу необходимо сохранить. Для этого щёлкните на File, Save As... . В открывшемся окне сохранения в строке Имя файла наберите это имя. Имя файла может содержать только латинские буквы (прописные и строчные), цифры и знаки подчёркивания. Пробелы не допускаются, их заменяют знаками подчёркивания. В начале

имени должна быть буква. Если вы Александр Васильевич Суворов, учитесь в группе МР-021 и выполняете работу «Введение», то наберите MR021_AVS_Introduction.

Пока программа не заработала как следует, т. е. находится в стадии отладки, запускать её можно прямо из окна редактора-отладчика. Для этого щёлкните на Debug, Run. Результат выполнения программы показан на рис. 1.3.

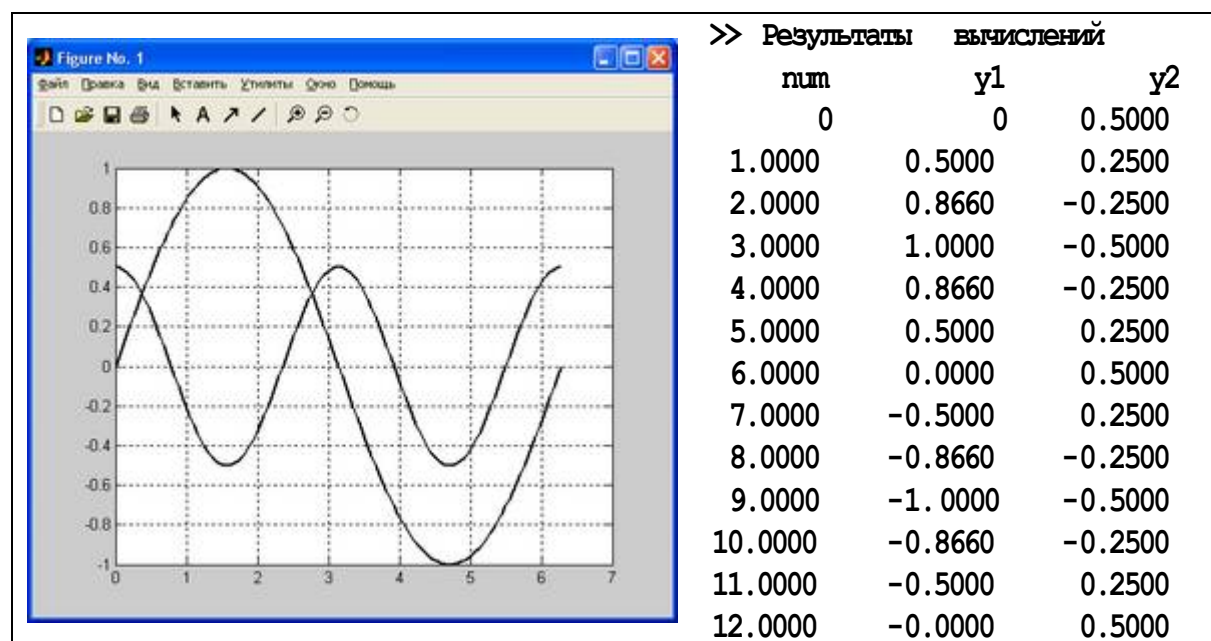


Рис. 1.3

Графики будут расположены в окне Figure № 1, таблица – в командном окне.

Завершение работы. Если преподаватель скажет, что всё верно, закройте все окна и завершите работу с компьютером.

Если вы не успели завершить работу до окончания занятия, то, придя в компьютерный класс после занятий, запустите MatLab. Сделайте текущей папку MatLabUsers, в которой сохранён ваш файл. Для этого щелчком на значке 2 (см. рис. 1.1) попробуйте сначала отыскать её в списке последних открывавшихся папок. Если это не удастся, воспользуйтесь, как и прежде, значком 3.

Сделав папку MatLabUsers текущей, щёлкните на File, Open. В открывшемся списке файлов дважды щёлкните на имени своей программы (MR021_AVS_Introduction). В окне редактора-отладчика продолжите работу с программой.

Занятие 2. Кинематические диаграммы

Цель занятия – освоение одного из аналитических методов кинематического анализа механизмов – метода векторных контуров.

С этой целью предлагается рычажный механизм, для которого на лабораторной работе были построены три кинематические диаграммы – функция положения и две её производные. Задача данного занятия – получить те же диаграммы с помощью компьютера.

Краткие сведения из теории

Как видно по занятию 1, диаграммы (графики) компьютер строит по результатам вычислений. Вычисления делаются по формулам. Формулы могут быть выведены различными методами. На данном занятии применяется метод векторных контуров.

Пусть исследуемым будет кулисный механизм, показанный на рис. 2.1, а. Звено 1 – входное, 3 – выходное; AB_0C – одно из крайних положений механизма. Функцией положения этого механизма является зависимость координаты ψ звена 3 от координаты φ звена 1.

Чтобы вывести эту зависимость, с каждым звеном механизма связывают вектор, выбранный так, чтобы образовался замкнутый векторный контур, кинематически эквивалентный механизму (рис. 2.1, б). Направление векторов на линии действия – произвольное, вектор \vec{l}_1 , например, может быть направлен от точки B к A , а \vec{l}_2 – от C к B .

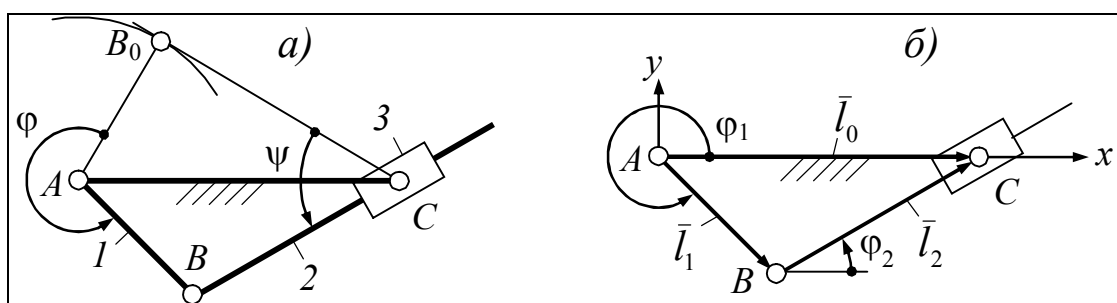


Рис. 2.1

Со стойкой связывают систему координат Axy , расположенную тоже произвольно. Показывают положительные углы векторов относительно оси x , и только x . Положительные углы отсчитывают, как известно, против часовой стрелки.

Составляют уравнение замкнутости векторного контура. Оно представляет собой геометрическую сумму векторов. Этой

сумме придают такую форму, при которой она равна нулю. Сумма равна нулю при условии, что векторы, попутные обходу контура в каком-либо направлении, подставляют в эту сумму со знаком «плюс», а встречные – со знаком «минус». Начиная, например, с вектора \bar{l}_1 и обходя контур против часовой стрелки, получают:

$$\bar{l}_1 + \bar{l}_2 - \bar{l}_0 = 0.$$

По уравнению замкнутости составляют *уравнения проекций* контура на оси x и y . При проецировании на ось x модуль каждого вектора умножают на косинус его угла с осью x , при проецировании на ось y – на синус того же угла. Знаки в уравнениях проекций оставляют такими же, как в уравнении замкнутости. Учитывая, что угол φ_0 вектора \bar{l}_0 равен нулю, получают:

$$x: \quad \begin{cases} l_1 \cos \varphi_1 + l_2 \cos \varphi_2 - l_0 = 0; \end{cases} \quad (2.1)$$

$$y: \quad \begin{cases} l_1 \sin \varphi_1 + l_2 \sin \varphi_2 = 0. \end{cases} \quad (2.2)$$

Уравнения проекций образуют систему. В ней l_0, l_1 предполагаются известными; φ_1 отличается от φ только началом отсчёта и, следовательно, может считаться величиной задаваемой. Неизвестными оказываются l_2 и φ_2 , их выводят методом исключения.

Выгоднее исключать φ_2 . Чтобы это сделать, произведения $l_2 \cos \varphi_2$ и $l_2 \sin \varphi_2$ выносят за знак равенства. Полученное возводят в квадрат и складывают. После преобразований получают l_2 .

Из уравнений (2.1), (2.2) можно вывести как синус, так и косинус от φ_2 . Но поскольку в любой фазе движения механизма вектор \bar{l}_2 находится в правой половине тригонометрического круга, то для определения угла φ_2 достаточно знать только синус. Через него из (2.2) находят φ_2 .

Дифференцируя уравнения (2.1), (2.2) по координате φ_1 , получают систему, содержащую первые производные от l_2 и φ_2 :

$$x': \quad \begin{cases} -l_1 \sin \varphi_1 - l_2' \cos \varphi_2 - l_2 (\sin \varphi_2) \varphi_2' = 0; \end{cases} \quad (2.3)$$

$$y': \quad \begin{cases} l_1 \cos \varphi_1 + l_2' \sin \varphi_2 + l_2 (\cos \varphi_2) \varphi_2' = 0. \end{cases} \quad (2.4)$$

Поворотом осей координат на угол φ_2 получают:

$$\begin{cases} -l_1 \sin(\varphi_1 - \varphi_2) - l_2' = 0; \\ l_1 \cos(\varphi_1 - \varphi_2) + l_2 \varphi_2' = 0. \end{cases}$$

Отсюда выводят l_2' и φ_2' .

Дифференцируя уравнения (2.3), (2.4) по φ_1 , получают систему, содержащую вторые производные от l_2 и φ_2 . Их находят так же, как и первые производные – поворотом системы координат на φ_2 . При дифференцировании придётся столкнуться с произведением трёх переменных. Производную от такого произведения получают по формуле

$$(uvw)' = u'vw + uv'w + uvw'.$$

Производная от $l_2(\cos \varphi_2)\varphi_2'$, например, будет выглядеть так:

$$l_2'(\cos \varphi_2)\varphi_2' - l_2(\sin \varphi_2)(\varphi_2')^2 + l_2(\cos \varphi_2)\varphi_2''.$$

Чтобы сделать сопоставимыми графические и аналитические диаграммы, из треугольника AB_0C (рис. 2.2) находят начальные углы α и β входного и выходного звеньев соответственно. Независимой переменной по-прежнему считают координату φ . Величины, вычисляемые после φ , выстраивают в следующую цепочку: $\varphi_1 \rightarrow \varphi_2 \rightarrow \psi$.

Как видно по рисунку, $\varphi_1 = \varphi + \alpha$. Вычисление φ_2 рассмотрено выше, $\psi = \beta + \varphi_2$. Дифференцируя ψ , находят: $\psi' = \varphi_2'$; $\psi'' = \varphi_2''$. Производные по φ_1 равны производным по φ , что и требуется.

Если контур четырёхугольный (рис. 2.3), то для упрощения задачи вводят вектор \bar{l}_4 , который делит четырёхугольный контур на два треугольных, причём так, что в первый контур – ABD входят векторы \bar{l}_0, \bar{l}_1 с заданным взаимным положением.

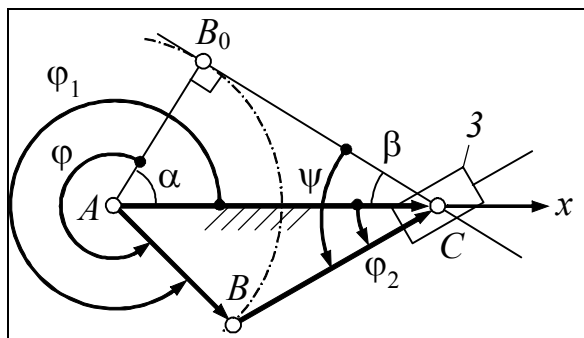


Рис. 2.2

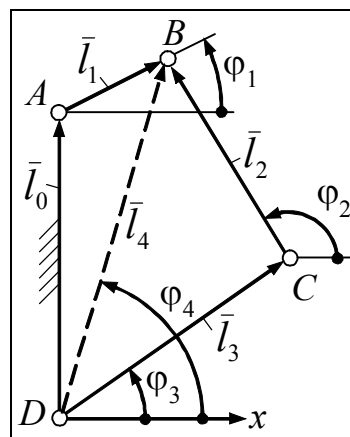


Рис. 2.3

Из уравнений проекций контура ABD находят l_4, φ_4 ; из уравнений проекций контура BCD находят φ_2, φ_3 . Производные полу-

чают дифференцированием уравнений проекций контура $ABCD$. Такова суть метода векторных контуров.

Ввиду сложности задачи, начиная со следующей страницы приведены формулы и программы для всех механизмов, исследуемых на данном занятии.

Обозначения

Обозначения в формулах	Обозначения в программах
l_0, l_1, l_2 и т. д. – размеры звеньев или длины векторов	L0, L1, L2
$\alpha, \beta, \varphi, \psi$ – угловые координаты звеньев или векторов	alfa, beta, fi, psi
$s', \varphi'_2, \varphi'_3, \psi'$ – первые производные от координат $s, \varphi_2, \varphi_3, \psi$	sv, fi2v, fi3v, psiv
$s'', \varphi''_2, \varphi''_3, \psi''$ – вторые производные от координат $s, \varphi_2, \varphi_3, \psi$	sw, fi2w, fi3w, psiw

Как видно по таблице, окончания **v** и **w** в именах переменных означают первую и вторую производную от этих переменных.

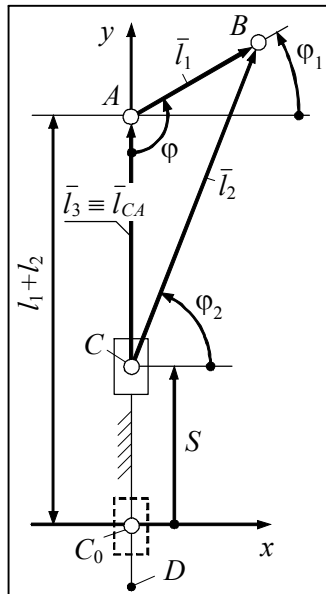
Порядок работы

1. Включите компьютер и запустите MatLab.
2. Щелчком на File, New M-File откройте окно редактора-отладчика. Введите в это окно программу. Сохраните её под вашим именем с добавлением Diagrames и запустите на выполнение.
3. Сопоставьте вид кинематических диаграмм, полученных графическим и аналитическим методами.
4. Если всё правильно, предъявите результаты преподавателю.
5. Скопируйте на «флэшку» вывод формул, программу и кинематические диаграммы. Распечатайте всё это для отчёта.



Формулы и программы

Кривошипно-ползунный механизм



Задача. По размерам l_1, l_2 вывести формулы для вычисления зависимостей $s(\varphi), s'(\varphi), s''(\varphi)$.

Решение. φ – независимая переменная. $\boxed{\varphi_1 = \varphi - \pi/2}$.

Уравнение замкнутости векторного контура ABC :

$$\bar{l}_1 - \bar{l}_2 + \bar{l}_3 = 0.$$

Уравнения проекций контура на оси x и y :

$$x: \begin{cases} l_1 \cos \varphi_1 - l_2 \cos \varphi_2 = 0; \end{cases} \quad (1)$$

$$y: \begin{cases} l_1 \sin \varphi_1 - l_2 \sin \varphi_2 + l_3 = 0. \end{cases} \quad (2)$$

Неизвестными являются φ_2 и l_3 . Из (1) находят:

$$\boxed{\varphi_2 = \arccos \frac{l_1 \cos \varphi_1}{l_2}}. \text{ Из (2): } \boxed{l_3 = -l_1 \sin \varphi_1 + l_2 \sin \varphi_2}.$$

Вывод производных по φ_1 .

l_1 и l_2 – константы, l_3 – величина переменная, с учётом этого производные от уравнений (1), (2) имеют вид:

$$x': \begin{cases} -l_1 \sin \varphi_1 + l_2 (\sin \varphi_2) \varphi_2' = 0; \end{cases} \quad (3)$$

$$y': \begin{cases} l_1 \cos \varphi_1 - l_2 (\cos \varphi_2) \varphi_2' + l_3' = 0. \end{cases} \quad (4)$$

Неизвестными являются φ_2' и l_3' .

Из уравнения (5): $\boxed{\varphi_2' = \frac{l_1 \sin \varphi_1}{l_2 \sin \varphi_2}}.$ Из (6): $\boxed{l_3' = -l_1 \cos \varphi_1 + l_2 (\cos \varphi_2) \varphi_2'}$.

Для определения вторых производных дифференцируют уравнения (3), (4):

$$x'': \begin{cases} -l_1 \cos \varphi_1 + l_2 (\cos \varphi_2) (\varphi_2')^2 + l_2 (\sin \varphi_2) \varphi_2'' = 0; \end{cases} \quad (5)$$

$$y'': \begin{cases} -l_1 \sin \varphi_1 + l_2 (\sin \varphi_2) (\varphi_2')^2 - l_2 (\cos \varphi_2) \varphi_2'' + l_3'' = 0. \end{cases} \quad (6)$$

Неизвестными являются φ_2'' и l_3'' .

Из (5): $\boxed{\varphi_2'' = \frac{l_1 \cos \varphi_1 - l_2 (\cos \varphi_2) (\varphi_2')^2}{l_2 \sin \varphi_2}}.$

Из (6): $\boxed{l_3'' = l_1 \sin \varphi_1 - l_2 (\sin \varphi_2) (\varphi_2')^2 + l_2 (\cos \varphi_2) (\varphi_2'')}$.

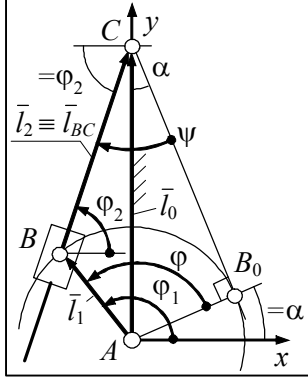
Согласно рисунку, $\boxed{s = l_1 + l_2 - l_3}$. Отсюда: $\boxed{s' = -l_3'}$, $\boxed{s'' = -l_3''}$, что и требуется.

Программа

```
%----- --Постоянные параметры схемы-----
L1=40; L2=120; % Снять с модели вашего механизма
%-----
n=120; % Число шагов кривошипа за оборот
h=2*pi/n; % Шаг кривошипа
fi=0:h:2*pi; % Входная координата
%----- Выходные координаты и их производные -----
fi1=fi-pi/2;
fi2=acos(L1*cos(fi1)/L2);
L3=-L1*sin(fi1)+L2*sin(fi2);
fi2v=L1*sin(fi1)/(L2*sin(fi2));
fi2w=(L1*cos(fi1)-L2*cos(fi2).*fi2v.^2)/(L2*sin(fi2));
s=L1+L2-L3;
sv=L1*cos(fi1)-L2*cos(fi2).*fi2v;
sw=-L1*sin(fi1)+L2*sin(fi2).*fi2v.^2-L2*cos(fi2).*fi2w;
%----- Построение диаграмм -----
plot(fi,s,fi,sv,fi,sw), grid
%----- Построение таблицы -----
i=1:10:n+1; % Индекс каждого 10-го значения переменной
num=0:n/10; % Номера строк в таблице
disp(' Результаты вычислений') % Заголовок таблицы
disp(' num fi s sv sw') % Подзаголовки
disp([num' fi(i)' s(i)' sv(i)' sw(i)']) % Таблица
```



Механизм с возвратно-вращающейся кулисой



Задача. По размерам l_0, l_1 вывести формулы для вычисления зависимостей $\psi(\varphi), \psi'(\varphi), \psi''(\varphi)$.

Решение. Из $\triangle AB_0C$ $\alpha = \arcsin(l_1/l_0)$. φ – независимая переменная, $\varphi_1 = \varphi + \alpha$.

Уравнение замкнутости векторного контура ABC :

$$\bar{l}_1 + \bar{l}_2 - \bar{l}_0 = 0.$$

Уравнения проекций контура на оси x и y :

$$x: l_1 \cos \varphi_1 + l_2 \cos \varphi_2 = 0; \quad (1)$$

$$y: l_1 \sin \varphi_1 + l_2 \sin \varphi_2 - l_0 = 0. \quad (2)$$

Корнями системы (1), (2) являются l_2 и φ_2 . Чтобы найти l_2 , избавляются от φ_2 . Для этого выносят за знак равенства $l_2 \cos \varphi_2$ и $l_2 \sin \varphi_2$. Уравнения, полученные после переноса, возводят в квадрат и складывают друг с другом. После преобразований получают: $l_2 = \sqrt{l_1^2 + l_0^2 - 2l_1l_0 \sin \varphi_1}$. Формула определяет модуль вектора l_2 , поэтому \pm перед радикалом не ставят.

Возвращаясь в систему, из (1) находят второй её корень:

$$\varphi_2 = \arccos [(-l_1 \cos \varphi_1)/l_2].$$

Вывод производных по φ_1 . l_0 и l_1 – константы, l_2 – величина переменная, с учётом этого производные от уравнений (1), (2) имеют вид:

$$x': -l_1 \sin \varphi_1 + l_2' \cos \varphi_2 - l_2 (\sin \varphi_2) \varphi_2' = 0; \quad (3)$$

$$y': l_1 \cos \varphi_1 + l_2' \sin \varphi_2 + l_2 (\cos \varphi_2) \varphi_2' = 0. \quad (4)$$

Корнями системы (3), (4) являются l_2' и φ_2' , выводят оба. Для этого поворачивают систему координат Ax_1y_1 на угол φ_2 . В результате поворота все углы уменьшаются на φ_2 и уравнения (3), (4) принимают вид:

$$-l_1 \sin(\varphi_1 - \varphi_2) + l_2' = 0; \quad (5)$$

$$l_1 \cos(\varphi_1 - \varphi_2) + l_2 \varphi_2' = 0. \quad (6)$$

Из (5) находят: $l_2' = l_1 \sin(\varphi_1 - \varphi_2)$. Из (6): $\varphi_2' = [-l_1 \cos(\varphi_1 - \varphi_2)]/l_2$.

Для определения вторых производных дифференцируют уравнения (3), (4):

$$x'': [-l_1 \cos \varphi_1 + l_2'' \cos \varphi_2 - 2l_2' (\sin \varphi_2) \varphi_2' - l_2 (\cos \varphi_2) (\varphi_2')^2 - l_2 (\sin \varphi_2) \varphi_2'' = 0; \quad (7)$$

$$y'': [-l_1 \sin \varphi_1 + l_2'' \sin \varphi_2 + 2l_2' (\cos \varphi_2) \varphi_2' - l_2 (\sin \varphi_2) (\varphi_2')^2 + l_2 (\cos \varphi_2) \varphi_2'' = 0. \quad (8)$$

Корнями системы (7), (8) являются величины l_2'' и φ_2'' . Требуется только последняя. После поворота координатных осей на угол φ_2 уравнение (8) принимает вид: $-l_1 \sin(\varphi_1 - \varphi_2) + 2l_2' \varphi_2' + l_2 \varphi_2'' = 0$, откуда

$$\varphi_2'' = [l_1 \sin(\varphi_1 - \varphi_2) - 2l_2' \varphi_2'] / l_2.$$

Согласно рисунку, $\psi = \pi/2 + \alpha - \varphi_2$. Отсюда: $\psi' = -\varphi_2'$, $\psi'' = -\varphi_2''$.

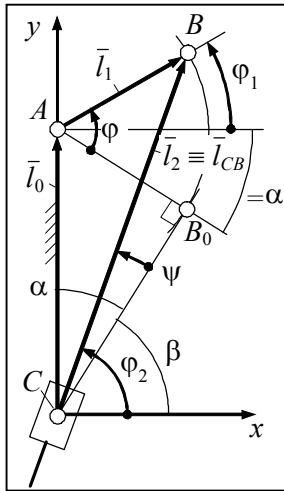
Программа

```

%----- Постоянные параметры схемы -----
L0= 120; L1= 40; % Снять с модели вашего механизма
alfa =asin (L1/L0);
%-----
n=120; % Число шагов кривошипа за оборот
h=2*pi/n; % Шаг кривошипа
fi=0:h:2*pi; % Входная координата
%----- Выходные координаты и их производные -----
fi1=fi+alfa;
L2=sqrt(L0^2+L1^2 -2*L0*L1*sin(fi1)) ;
fi2= acos( -L1*cos(fi1)./L2);
L2v=L1*sin(fi1 -fi2);
fi2v= -L1*cos(fi1 -fi2)./L2;
fi2w=(L1*sin(fi1 -fi2) -2*L2v.*fi2v)./L2;
psi=pi /2+alfa -fi2;
psiv= -fi2v;
psiw= -fi2w;
%----- Построение диаграмм -----
plot(fi, psi, fi, psiv, fi, psiw), grid
%----- Построение таблицы -----
i=1:10:n+1; % Индекс каждого 10-го значения переменной
num=0:n/10; % Номера строк в таблице
disp(' Результаты вычислений') % Заголовок таблицы
disp([' num fi psi psiv psiw']) % Подзаголовки
disp([num' fi(i)' psi(i)' psiv(i)' psiw(i)']) % Таблица

```

Механизм с кулисой плоско-параллельного движения



Задача. По размерам l_0 , l_1 вывести формулы для вычисления зависимостей $\psi(\varphi)$, $\psi'(\varphi)$, $\psi''(\varphi)$.

Решение. Из $\triangle AB_0C$ $\alpha = \arcsin(l_1/l_0)$. φ – независимая переменная, $\varphi_1 = \varphi - \alpha$.

Уравнение замкнутости векторного контура ABC :

$$\vec{l}_1 - \vec{l}_2 + \vec{l}_0 = 0.$$

Уравнения проекций контура на оси x и y :

$$x: \begin{cases} l_1 \cos \varphi_1 - l_2 \cos \varphi_2 = 0; \end{cases} \quad (1)$$

$$y: \begin{cases} l_1 \sin \varphi_1 - l_2 \sin \varphi_2 + l_0 = 0. \end{cases} \quad (2)$$

Корнями системы (1), (2) являются l_2 и φ_2 . Чтобы найти l_2 , исключают φ_2 . Для этого выносят за знак равенства $l_2 \cos \varphi_2$ и $l_2 \sin \varphi_2$. Уравнения, полученные после переноса, возводят в квадрат и складывают друг с другом. После преобразований получают: $l_2 = \sqrt{l_1^2 + l_0^2 + 2l_1l_0 \sin \varphi_1}$. Формула определяет модуль вектора l_2 , поэтому \pm перед радикалом не ставят.

Возвращаясь в систему, из (1) находят второй её корень:

$$\varphi_2 = \arccos[(l_1 \cos \varphi_1)/l_2].$$

Вывод производных по φ_1 . l_0 и l_1 – константы, l_2 – величина переменная, с учётом этого производные от уравнений (1), (2) имеют вид:

$$x': \begin{cases} -l_1 \sin \varphi_1 - l_2' \cos \varphi_2 + l_2 (\sin \varphi_2) \varphi_2' = 0; \end{cases} \quad (3)$$

$$y': \begin{cases} l_1 \cos \varphi_1 - l_2' \sin \varphi_2 - l_2 (\cos \varphi_2) \varphi_2' = 0. \end{cases} \quad (4)$$

Корнями системы (3), (4) являются l_2' и φ_2' , выводят оба. Для этого поворачивают систему координат S_{xy} на угол φ_2 . В результате поворота все углы уменьшаются на φ_2 и уравнения (3), (4) принимают вид:

$$\begin{cases} -l_1 \sin(\varphi_1 - \varphi_2) - l_2' = 0; \end{cases} \quad (5)$$

$$\begin{cases} l_1 \cos(\varphi_1 - \varphi_2) - l_2 \varphi_2' = 0. \end{cases} \quad (6)$$

Из (5) находят: $l_2' = -l_1 \sin(\varphi_1 - \varphi_2)$. Из (6): $\varphi_2' = l_1 [\cos(\varphi_1 - \varphi_2)]/l_2$.

Для определения вторых производных дифференцируют уравнения (3), (4):

$$x'': \begin{cases} -l_1 \cos \varphi_1 - l_2'' \cos \varphi_2 + 2l_2' (\sin \varphi_2) \varphi_2' + l_2 (\cos \varphi_2) (\varphi_2')^2 + l_2 (\sin \varphi_2) \varphi_2'' = 0; \end{cases} \quad (7)$$

$$y'': \begin{cases} -l_1 \sin \varphi_1 - l_2'' \sin \varphi_2 - 2l_2' (\cos \varphi_2) \varphi_2' + l_2 (\sin \varphi_2) (\varphi_2')^2 - l_2 (\cos \varphi_2) \varphi_2'' = 0. \end{cases} \quad (8)$$

Корнями системы (7), (8) являются величины l_2'' и φ_2'' . Требуется только последняя. После поворота системы координат на угол φ_2 уравнение (8) принимает вид: $-l_1 \sin(\varphi_1 - \varphi_2) - 2l_2' \varphi_2' - l_2 \varphi_2'' = 0$, откуда $\varphi_2'' = [-l_1 \sin(\varphi_1 - \varphi_2) - 2l_2' \varphi_2']/l_2$.

Согласно рисунку, $\beta = \pi/2 - \alpha$, $\psi = \varphi_2 - \beta$. Отсюда: $\psi' = \varphi_2'$, $\psi'' = \varphi_2''$.

Программа

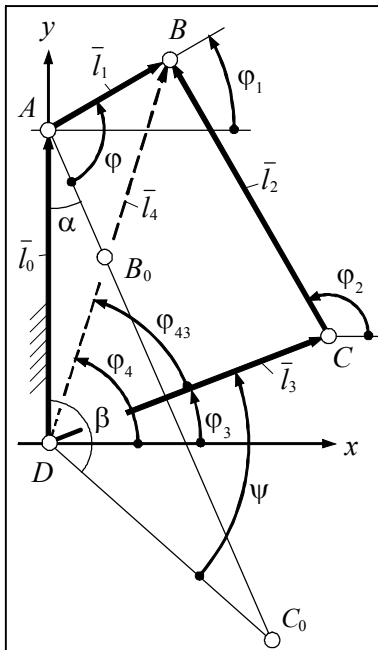
```

%----- Постоянные параметры схемы -----
L0=120; L1=40; % Снять с модели вашего механизма
alfa=asin(L1/L0); beta=pi/2-alfa;
%-----
n=120; % Число шагов кривошипа за оборот
h=2*pi/n; % Шаг кривошипа
fi=0:h:2*pi; % Входная координата
%----- Выходные координаты и их производные -----
fi1=fi-alfa;
L2=sqrt(L0^2+L1^2+2*L0*L1*sin(fi1));
fi2=acos(L1*cos(fi1)/L2);
L2v=-L1*sin(fi1-fi2);
fi2v=L1*cos(fi1-fi2)/L2;
fi2w=(-L1*sin(fi1-fi2)-2*L2v.*fi2v)/L2;
psi=fi2-beta;
psiv=fi2v;
psiw=fi2w;
%----- Построение диаграмм -----
plot(fi, psi, fi, psiv, fi, psiw), grid
%----- Построение таблицы -----
i=1:10:n+1; % Индекс каждого 10-го значения переменной
num=0:n/10; % Номера строк в таблице
disp(' Результаты вычислений') % Заголовок таблицы
disp([' num fi psi psiv psiw']) % Подзаголовки
disp([num' fi(i)' psi(i)' psiv(i)' psiw(i)']) % Таблица

```



Шарнирный четырёхзвенник



Задача. По размерам $l_0 \dots l_3$ вывести формулы для вычисления зависимостей $\psi(\varphi)$, $\psi'(\varphi)$, $\psi''(\varphi)$.

Решение. AB_0C_0D – начальное положение механизма. Обозначая AC_0 через l_s , получают: $l_s = l_1 + l_2$. Из $\triangle AC_0D$ по теореме косинусов находят начальные углы звеньев AB и CD :

$$\alpha = \arccos [(l_0^2 + l_s^2 - l_3^2) / (2 l_0 l_s)];$$

$$\beta = \arccos [(l_0^2 + l_3^2 - l_s^2) / (2 l_0 l_3)].$$

φ – независимая переменная, $\varphi_1 = \varphi + \alpha - \pi/2$.

Контур ABD . Уравнение замкнутости: $\bar{l}_1 - \bar{l}_4 + \bar{l}_0 = 0$.

Уравнения проекций:

$$x: \begin{cases} l_1 \cos \varphi_1 - l_4 \cos \varphi_4 = 0; \end{cases} \quad (1)$$

$$y: \begin{cases} l_1 \sin \varphi_1 - l_4 \sin \varphi_4 + l_0 = 0. \end{cases} \quad (2)$$

Корнями системы (1), (2) являются l_4 и φ_4 . Исключая φ_4 , находят первый корень: $l_4 = \sqrt{l_0^2 + l_1^2 + 2l_0 l_1 \sin \varphi_1}$.

Способ исключения такой же, как при решении системы (1), (2) в предыдущем примере. Возвращаясь в данный пример, из уравнения (1) находят второй корень: $\varphi_4 = \arccos (l_1 \cos \varphi_1 / l_4)$.

Контур BCD . Уравнение замкнутости: $\bar{l}_3 + \bar{l}_2 - \bar{l}_4 = 0$. Уравнения проекций:

$$x: \begin{cases} l_3 \cos \varphi_3 + l_2 \cos \varphi_2 - l_4 \cos \varphi_4 = 0; \end{cases} \quad (3)$$

$$y: \begin{cases} l_3 \sin \varphi_3 + l_2 \sin \varphi_2 - l_4 \sin \varphi_4 = 0. \end{cases} \quad (4)$$

Корнями системы (3), (4) являются φ_3 и φ_2 . Исключая φ_2 , как это делалось выше, получают: $\varphi_{43} = \varphi_4 - \varphi_3 = \arccos [(l_3^2 + l_4^2 - l_2^2) / (2 l_3 l_4)]$. Отсюда $\varphi_3 = \varphi_4 - \varphi_{43}$.

Из уравнения (3) находят второй корень: $\varphi_2 = \arccos [(l_4 \cos \varphi_4 - l_3 \cos \varphi_3) / l_2]$.

Вывод производных по φ_1 . Рассматривают контур $ABCD$. Составляют уравнение замкнутости $\bar{l}_1 - \bar{l}_2 - \bar{l}_3 + \bar{l}_0 = 0$ и уравнения проекций контура:

$$x: \begin{cases} l_1 \cos \varphi_1 - l_2 \cos \varphi_2 - l_3 \cos \varphi_3 = 0; \end{cases} \quad (5)$$

$$y: \begin{cases} l_1 \sin \varphi_1 - l_2 \sin \varphi_2 - l_3 \sin \varphi_3 + l_0 = 0. \end{cases} \quad (6)$$

Дифференцируя уравнения (5) и (6), получают:

$$x': \begin{cases} -l_1 \sin \varphi_1 + l_2 (\sin \varphi_2) \varphi_2' + l_3 (\sin \varphi_3) \varphi_3' = 0; \end{cases} \quad (7)$$

$$y': \begin{cases} l_1 \cos \varphi_1 - l_2 (\cos \varphi_2) \varphi_2' - l_3 (\cos \varphi_3) \varphi_3' = 0. \end{cases} \quad (8)$$

Корнями системы (7), (8) являются φ_2' и φ_3' . После поворота осей координат

сначала на φ_3 , затем на φ_2 все углы уменьшаются на φ_3 , φ_2 ; уравнение (7) принимает следующие две формы: $l_1 \sin(\varphi_1 - \varphi_3) + l_2 \sin(\varphi_2 - \varphi_3) \varphi_2' = 0$

и $-l_1 \sin(\varphi_1 - \varphi_2) + l_3 \sin(\varphi_3 - \varphi_2) \varphi_3' = 0$. Из них выводят:

$$\varphi_2' = \frac{-l_1 \sin(\varphi_1 - \varphi_3)}{l_2 \sin(\varphi_2 - \varphi_3)},$$

$$\varphi_3' = \frac{l_1 \sin(\varphi_1 - \varphi_2)}{l_3 \sin(\varphi_3 - \varphi_2)}.$$

Для вывода вторых производных дифференцируют уравнение (7):

$$x'': -l_1 \cos \varphi_1 + l_2 (\cos \varphi_2)(\varphi_2')^2 + l_2 (\sin \varphi_2)(\varphi_2'')^2 + l_3 (\cos \varphi_3)(\varphi_3')^2 + l_3 (\sin \varphi_3)\varphi_3'' = 0. \quad (9)$$

Уравнение (9) содержит два неизвестных – φ_2'' и φ_3'' . Повернув оси на φ_2 , исклю-

чают ненужное φ_2'' и получают:

$$\varphi_3'' = \frac{l_1 \cos(\varphi_1 - \varphi_2) - l_2 (\varphi_2')^2 - l_3 (\varphi_3')^2 \cos(\varphi_3 - \varphi_2)}{l_3 \sin(\varphi_3 - \varphi_2)}.$$

Согласно рисунку, $\psi = \beta - \pi/2 + \varphi_3$. Отсюда: $\psi' = \varphi_3'$, $\psi'' = \varphi_3''$.

Программа

```
%----- Постоянные параметры схемы -----
L0=110; L1=40; L2=140; L3=100; % Снять с модели механизма
Lsum=L1+L2;
alfa=acos((L0^2+Lsum^2-L3^2)/(2*L0*Lsum));
beta=acos((L0^2+L3^2-Lsum^2)/(2*L0*L3));
n=120; h=2*pi/n; fi=0:h:2*pi; % n - число шагов кривошипа
% за оборот, h - шаг, fi - входная координата
%----- Выходные координаты и их производные -----
fi1=fi+alfa-pi/2; L4=sqrt(L0^2+L1^2+2*L0*L1*sin(fi1));
fi4=acos(L1*cos(fi1)/L4);
fi3=fi4-acos((L3^2+L4.^2-L2^2)/(2*L3*L4));
fi2=acos((L4.*cos(fi4)-L3*cos(fi3))/L2);
fi2v=-L1*sin(fi1-fi3)/(L2*sin(fi2-fi3));
fi3v=L1*sin(fi1-fi2)/(L3*sin(fi3-fi2));
psi=beta-pi/2+fi3;
psiv=fi3v;
psiw=(L1*cos(fi1-fi2)-L2*fi2v.^2-...
L3*fi3v.^2.*cos(fi3-fi2))/(L3*sin(fi3-fi2));
%----- Построение диаграмм -----
plot(fi,psi,fi,psiv,fi,psiw), grid
%----- Построение таблицы -----
i=1:10:n+1; % Индекс каждого 10-го значения переменной
num=0:n/10; % Номера строк в таблице
disp(' Результаты вычислений') % Заголовок таблицы
disp(' num fi psi psiv psiw') % Подзаголовки
disp([num' fi(i)' psi(i)' psiv(i)' psiw(i)']) % Таблица
```

Занятие 3. Планы положений и анимация

Цель занятия – научиться использовать компьютер для построения планов положений и анимации рычажных механизмов. С этой целью предлагается: вывести формулы и составить по ним программу построения 12 планов положений; программу построения планов положений преобразовать в программу анимации.

Элементарные построения в системе MatLab

Кинематические схемы рычажных механизмов, рассмотренных на предыдущих занятиях, состоят из отрезков прямых и ломаных линий, а также окружностей и прямоугольников.

Прямые и ломаные линии строят с помощью функции **line (...)**, встроенной в MatLab. В скобках указывают аргументы функции. При построении отрезка аргументами являются массивы, состоящие из абсцисс и ординат начала и конца этого отрезка. На первом месте располагают массив абсцисс, на втором массив ординат. Аргументы, как всегда, разделяют запятыми.

У п р а ж н е н и е 1. Требуется построить отрезок AB с координатами: $x_a = 0,5$; $x_b = 3,6$; $y_a = 0,8$; $y_b = 2,4$. В командном окне наберите: **line ([0.5 3.6], [0.8 2.4]), grid ↵**. После набора получите (рис. 3.1).

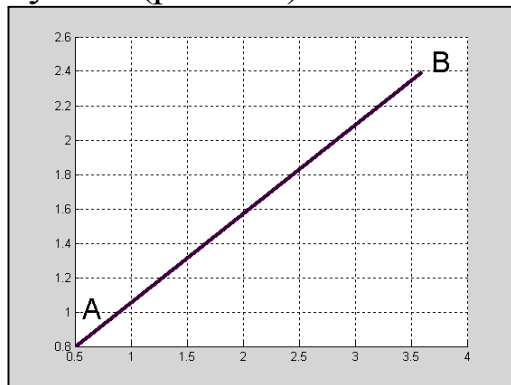


Рис. 3.1

Букв A и B на вашем рисунке не будет, иной будет и толщина линии. Эти изменения сделаны средствами ручного редактирования рисунка, хотя их можно и запрограммировать.

При построении ломаной аргументами являются массивы, состоящие из абсцисс и ординат ломаной от её начала до конца. Как и в случае отрезка, на первом месте располагают массив абсцисс, на втором массив ординат. Чтобы новый рисунок не накладывался на построенный ранее, перед выполнением следующего упражнения закройте окно со старым рисунком.

У п р а ж н е н и е 2. Требуется построить ломаную $ABCDE$ с абсциссами 0, 1, 2, 3, 4 и ординатами 0, 4, 1, 3, 2.

Наберите: `line([0 1 2 3 4],[0 4 1 3 2]), grid ↵`. После набора получите (рис. 3.2).

С помощью функции **line** можно построить и замкнутую ломаную, т. е. многоугольник. Перечисляя координаты, многоугольник обходят полностью: с какой точки начали, той и заканчивают. В результате координаты первой точки указывают дважды – в начале и конце массива.

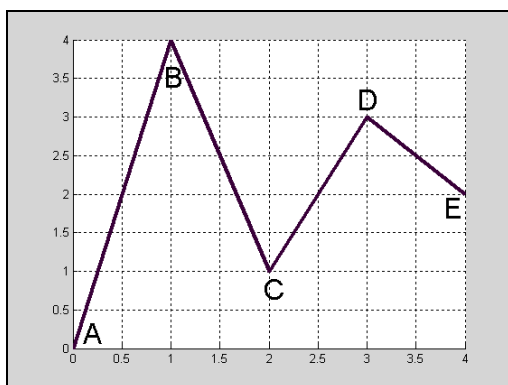


Рис. 3.2

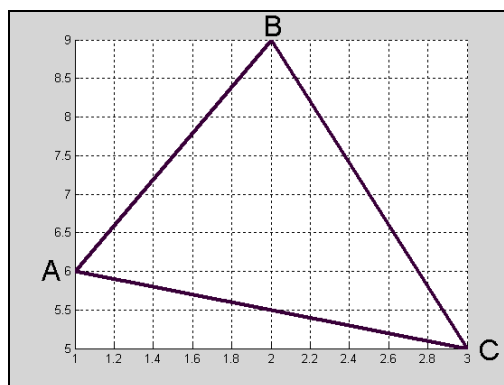


Рис. 3.3

У п р а ж н е н и е 3. Требуется построить треугольник *ABC* с абсциссами 1, 2, 3, 1 и ординатами 6, 9, 5, 6. Судя по цифрам, $x_a=1$, $y_a=6$. Наберите: `line([1 2 3 1], [6 9 5 6]), grid ↵`. После набора получите (рис. 3.3).

Закрашенные многоугольники. Многоугольник на рис. 3.3 получился прозрачным: видны линии сетки. Нам потребуются только непрозрачные многоугольники. Для их построения используется функция **patch (...)**. Буквально, **patch** – лоскут, заплатка.

Обязательными аргументами функции **patch** являются массивы координат, указывающих углы многоугольника, а также цвет заливки (закраски). Как и в функции **line**, сначала перечисляют абсциссы, затем ординаты вершин многоугольника, но не от первой до первой, а от первой до последней.

Из цветов заливки нам понадобится только белый. Белый цвет (по-английски *white*) обозначают буквой **w**, заключаемой в апострофы. Из необязательных аргументов укажем **LineWidth**, определяющий толщину линии контура многоугольника. Имя этого аргумента также заключают в апострофы. Толщину указывают после имени аргумента и выражают в пунктах (пт), как в текстовом редакторе Word. Если толщину линии не указать, она получится равной 0,5 пт.

Упражнение 4. Требуется построить треугольник с размерами, взятыми из предыдущего упражнения, но с белой заливкой и толщиной линии 3 пт. Наберите:

`patch([1 2 3],[6 9 5],'w','LineWidth', 3),grid` ↵. После набора получите (рис. 3.4).

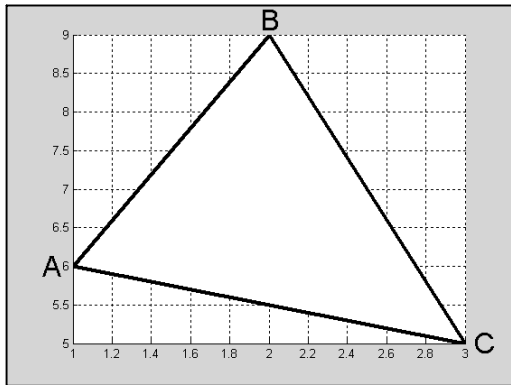


Рис. 3.4

MatLab не имеет функций, с помощью которых строятся окружности с заданным положением центра, а также прямоугольники с заданным углом наклона. Эти функции автору данного пособия пришлось создавать самому. Функции, созданные в системе MatLab пользователем и предназначенные для использования при любых обозначениях входных и вы-

ходных параметров этих функций, называются *файл-функциями*.

Построение окружностей. Файл-функция построения окружности приведена ниже. Текст этой функции набирать не нужно.

```
1 function circle (r, xc, yc)
2 % Рисует белую окружность радиуса r
3 % с координатами xc, yc центра окружности
4 h=pi/12 ; fi = h:h:2*pi ;
5 x = xc+r*cos(fi) ;
6 y = yc+r*sin(fi) ;
7 patch(x,y,'w','LineWidth',1)
```

Любая файл-функция начинается с ключевого слова **function**. Далее следует имя функции, назначаемое пользователем. В данном случае это — **circle**. После имени функции указывают её аргументы или, иначе, входные параметры — **r**, **xc**, **yc**. Как видно по комментариям (строки 2, 3), это — радиус и координаты центра окружности. После комментариев располагается тело функции. Этим телом является программа, которая по входным параметрам строит многоугольник с двадцатью четырьмя, в данном случае, гранями. На экране такой многоугольник видится как окружность. Число граней определяет шаг **h**.

Файл-функция **circle** будет находиться в папке MatLabUsers, которую, как и прежде, вы сделаете текущей. Имя

файла, в котором хранится файл-функция, – это имя самой файл-функции. Текст функции приведён для ознакомления, а также на случай, если по какой-то причине указанной файл-функции не окажется или она будет повреждена, и тогда, и только тогда, вам придётся набрать её самим.

Файл-функцию вызывают в основную программу (*файл-сценарий*) так же, как любую встроенную функцию, т. е. по её имени. В скобках указывают численные значения или обозначения аргументов, перечисляя их в том порядке, как они записаны в строке 1. Обозначения аргументов могут отличаться от принятых в строке 1, поэтому первые, принятые в строке 1, называются формальными, а принятые нами называются фактическими.

У п р а ж н е н и е 5. Требуется построить окружность радиуса $r = 3$ с координатами центра: $x_A = 1, y_A = 2$. При обращении к функции **circle** замените формальные параметры фактическими. Значения этих параметров определите заранее: **R = 3; Xa = 1; Ya = 2; ↵**. После этого наберите: **circle (R, Xa, Ya), grid ↵** или **circle (3, 1, 2), grid ↵**. После набора получите (рис. 3.5).

Окружность получилась некруглой. Чтобы этого не было, необходимо потребовать равенства масштабов по осям координат. Наберите: **axis equal ↵**. Окружность стала правильной (рис. 3.6).

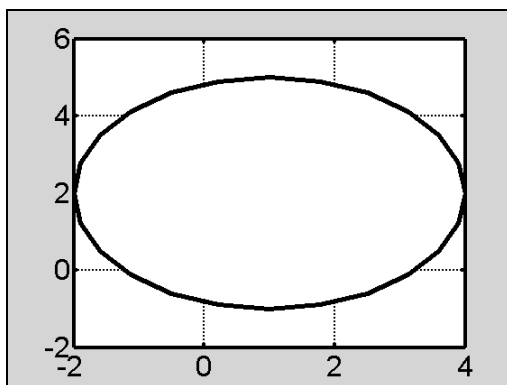


Рис. 3.5

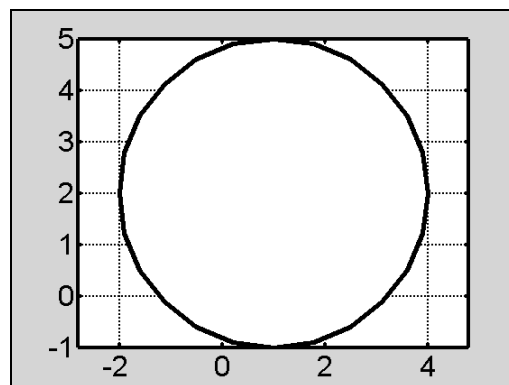


Рис. 3.6

Обратите внимание на то, что координатное окно формируется автоматически, причём получается не квадратным и, сохраняя пропорцию (примерно 3 на 4), плотно – по вертикали или горизонтали – охватывает заключённую в нём фигуру.

Построение прямоугольников. Прямоугольниками на кинематических схемах изображают кулисные камни и ползуны. Нет

смысла изображать эти звенья прямоугольниками разных размеров, поэтому строить прямоугольники будет одна и та же файл-функция **polzun**. Она также должна находиться в папке **MatLabUsers**, как и функция **circle**, и набирать её не следует. Вот эта функция.

```

1 function polzun (k,d,x,y,fi)
2 % Рисует прямоугольник со сторонами k, d (k<d)
3 % с координатами центра - x, y
4 % с длинной стороной (d), расположенной
5 % под углом fi рад относительно оси x
6 a=k/2; b=d/2; c=cos(fi); s=sin(fi);
7 x1=-b*c-a*s+x; y1=-b*s+a*c+y;
8 x2=-b*c+a*s+x; y2=-b*s-a*c+y;
9 x3=+b*c+a*s+x; y3=+b*s-a*c+y;
10 x4=+b*c-a*s+x; y4=+b*s+a*c+y;
11 patch([x1 x2 x3 x4],[y1 y2 y3 y4],'w','LineWidth',1)

```

В строках 7...10 функции **polzun** вычисляются координаты каждого из четырёх углов прямоугольника.

У п р а ж н е н и е 6. Требуется построить горизонтальный и наклонный прямоугольники со сторонами $k=0,5$, $d=1$, с координатами центра $x=1$, $y=1$ и углом наклона $\varphi=30^\circ$ для второго прямоугольника.

При обращении к функции **polzun** формальные параметры замените их численными значениями. Для построения горизонтального прямоугольника в окне команд наберите: **polzun (0.5,1,1,1,0)**, **axis equal**, **grid** ↵. После набора получите (рис. 3.7).

Закрыв окно с рис. 3.7, наберите **polzun (0.5,1,1,1, pi/6)**, **axis equal**, **grid** ↵. После набора получите (рис. 3.8).

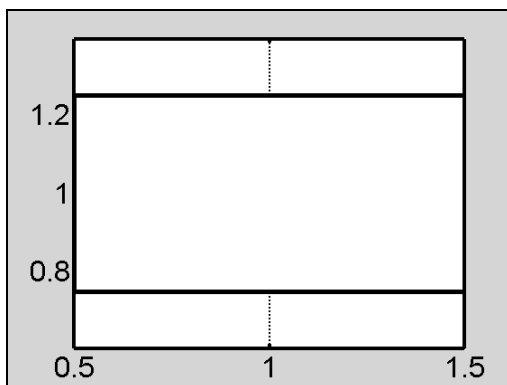


Рис. 3.7

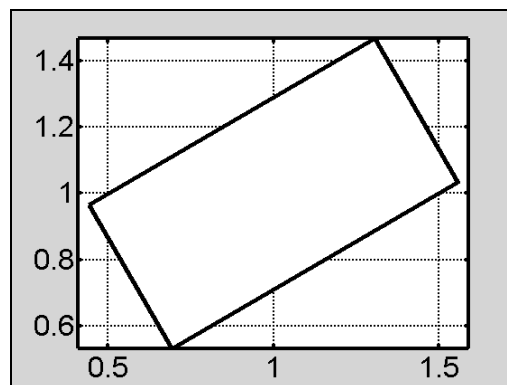


Рис. 3.8

Планы положений механизма

Планов положений будет 12, как в кинематических диаграммах, построенных на лабораторной работе графическим методом. Планы положений строят по угловым и линейным координатам механизма. Определение угловых координат рассмотрено в разделе «Занятие 2. Кинематические диаграммы».

Определение линейных координат. Пусть дан шарнирный четырёхзвенник $ABCD$ (рис. 3.9). Известны размеры звеньев. По координате φ_1 найдены координаты φ_2 , φ_3 . Требуется определить координаты точки C .

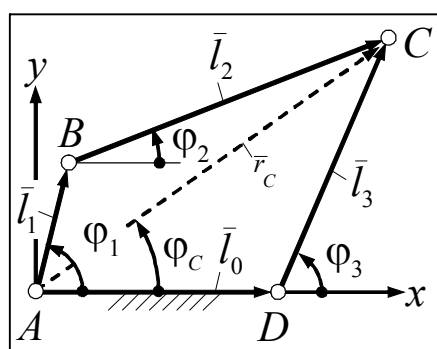


Рис. 3.9

Чтобы решить задачу, вводят радиус-вектор \vec{r}_C точки C . Радиус-вектор представляют в виде геометрической суммы каких-либо существующих векторов. В данном случае имеют смысл следующие два представления вектора \vec{r}_C :

$$\vec{r}_C = \vec{l}_1 + \vec{l}_2 \quad \text{и} \quad \vec{r}_C = \vec{l}_0 + \vec{l}_3.$$

Из первого представления получают:

$$r_C \cos \varphi_C = l_1 \cos \varphi_1 + l_2 \cos \varphi_2;$$

$$r_C \sin \varphi_C = l_1 \sin \varphi_1 + l_2 \sin \varphi_2.$$

Поскольку $r_C \cos \varphi_C = x_C$, а $r_C \sin \varphi_C = y_C$, то сразу пишут: $x_C = \dots$, $y_C = \dots$. Второе представление радиус-вектора \vec{r}_C даёт более простые выражения координат: $x_C = l_0 + l_3 \cos \varphi_3$; $y_C = l_3 \sin \varphi_3$.

Программирование координат неподвижных шарниров. В программах предыдущего занятия угловые координаты звеньев вычислялись путём поэлементной обработки массивов. Координаты, относящиеся к подвижным шарнирам, естественно вычислять этим же методом. Чтобы применить поэлементную обработку к координатам неподвижных шарниров, их превращают тоже в массивы. Превращение позволяют сделать функции `zeros(1,n)` и `ones(1,n)`. При аргументах, указанных в скобках, эти функции формируют массивы из n нулевых (`zeros`) и n единичных (`ones`) элементов, расположенных в одну строку. На число строк указывает первый аргумент – цифра 1.

Программирование планов положений. Помимо размеров звеньев, известных по предыдущему занятию, в программе должен быть задан радиус r окружности, изображающей шарниры, а также размеры короткой K_s и длинной L_s сторон кулисного камня или ползуна, если эти звенья имеются в механизме. Размеры задают исходя из пропорций, принятых в литературе. Возьмите эти размеры из примеров, приведённых ниже.

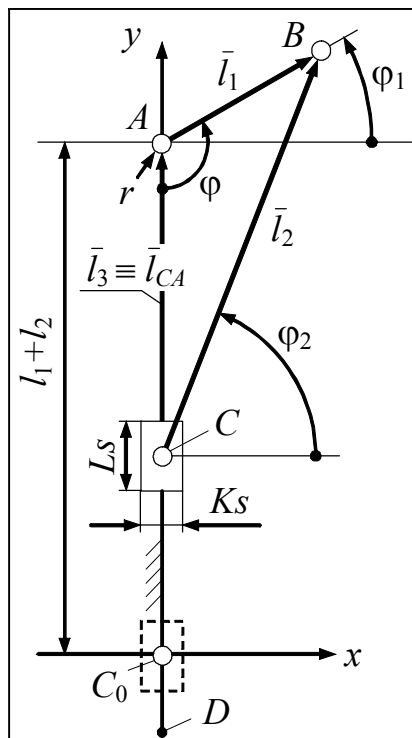
Планы положений строят в цикле **for ... end**. На каждом шаге цикла строится какое-то одно положение механизма, начиная с первого. В отличие от кинематических диаграмм, нулевое положение строить не требуется, т. к. его замещает 12-е положение.

Порядок построения должен быть таким, чтобы элементы схемы накрывали друг друга в последовательности, показанной на наших рисунках. Например, на схеме кривошипно-ползунного механизма (см. ниже) ползун изображён поверх своей направляющей, а шатун – поверх ползуна, следовательно, строить необходимо в последовательности: направляющая, ползун, шатун.

Порядок построения планов положений

1. Выведите формулы координат всех шарниров и точки D , лежащей на конце направляющей, если направляющая есть.
2. Откройте файл **Diagrams**, в котором сохранена программа построения кинематических диаграмм вашего механизма.
3. Сделайте копию файла, для чего сохраните его под вашим именем с дополнением **Planes**.
4. В файле **Planes** удалите строки, начиная с производных.
5. Внесите в урезанную программу изменения и дополнения, руководствуясь примерами, приведёнными ниже. Новые элементы программ в примерах выделены жирным шрифтом.
6. Не выходя из редактора-отладчика, запустите программу на выполнение. Результат предъявите преподавателю.
7. Если всё правильно, перепишите в тетрадь размеры осей графического окна. Размеры перечисляйте в последовательности: x_{\min} , x_{\max} , y_{\min} , y_{\max} .
8. Сделайте копию файла **Planes**, для чего сохраните его под вашим именем с дополнением **Animation**.
9. Превращение планов положений в анимацию рассмотрено после примеров, приведённых ниже.

Примеры построения планов положений Кривошипно-ползунный механизм



Координаты шарниров и точки D :

$$x_a = 0, \quad y_a = l_1 + l_2;$$

$$x_b = l_1 \cos \varphi_1, \quad y_b = y_a + l_1 \sin \varphi_1;$$

$$x_c = 0, \quad y_c = l_1 + l_2 - l_3;$$

$$x_d = 0, \quad y_d = -L_s.$$

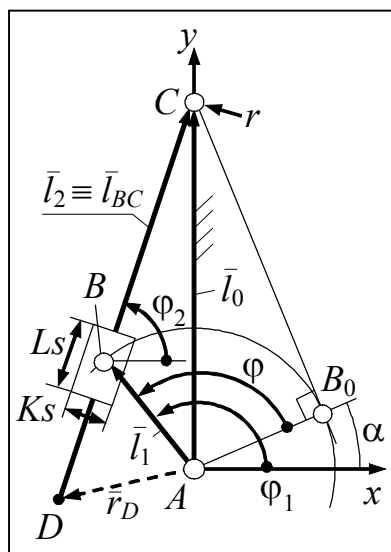
Программа

```
%--- Постоянные параметры схемы ---
L1=40; L2=120; r=0.06*L1;
Ks=4*r; Ls=2*Ks;
%-----
n=12; % Число планов положений
h=2*pi/n; % Шаг кривошипа
fi=h:h:2*pi; % Входная координата
%-- Выходные параметры и координаты --
fi1=fi-pi/2;

fi2=acos(L1*cos(fi1)/L2);
L3=-L1*sin(fi1)+L2*sin(fi2);
z=zeros(1,n); % Вспомогательный массив из n нулей
xa=z; ya=(L1+L2)*ones(1,n); % ones - массив из n единиц
xd=z; yd=-Ls*ones(1,n);
xb=L1*cos(fi1); yb=L1*sin(fi1)+L1+L2;
xc=z; yc=L1+L2-L3;

%----- Построение планов положений -----
for k=1:n % Начало цикла построения схемы
    line([xa,xd],[ya,yd]) % Стойка AD
    polzun(Ks,Ls,xc(k),yc(k),pi/2) % Ползун
    line([xa(k) xb(k) xc(k)],[ya(k) yb(k) yc(k)]) % Ломаная ABC
    circle(r,xa(k),ya(k)) % Шарнир A
    circle(r,xb(k),yb(k)) % Шарнир B
    circle(r,xc(k),yc(k)) % Шарнир C
end % Конец цикла построения схемы
axis equal % Установка равных масштабов по осям координат.
clear % Очистка рабочей области
```

Механизм с кулисой возвратно-вращательного движения



Координаты шарниров A, B, C очевидны (см. программу). Для определения координат точки D введём радиус-вектор $\bar{r}_D = \bar{l}_0 - \bar{l}_{DC}$. Проекции радиус-вектора дают:

$$\boxed{x_d = -l_{dc} \cos \varphi_2}, \quad \boxed{y_d = l_0 - l_{dc} \sin \varphi_2}.$$

Программа

--- Постоянные параметры схемы ---

```
L0=120; L1=40;  alfa=asin (L1/ L0);
```

```
r=0.04* L1;
```

$$K_s = 4 \cdot r; \quad L_s = 2 \cdot K_s; \quad L_{dc} = L_0 + L_1 + L_s;$$

```
n=12; h=2*pi/n; % Число планов пол о-
```

жений и шаг кривошипа.

```
fi=h:h:2*p i; % Входная координата.
```

%----- Прочие параметры и координаты -----

```
fi1=fi+alfa;
```

$$L2 = \sqrt{L1^2 + L0^2 - 2 * L1 * L0 * \sin(f11)};$$

```
fi2=acos( -L1*cos(fi1)./L2);
```

```
z=zeros(1, n); % Вспомогательный массив из n нулей
```

xa=z ; ya=z ; xc=z ;

```
ys=L0*ones (1, n); % ones - массив из n единиц
```

```
xb=L1* cos (fi 1);  yb=L1* sin (fi 1);
```

```
xd=-Ldc*cos (fi 2); yd= L0-Ldc*sin (fi 2);
```

----- Построение планов положений -----

```
for k=1:n % Начало цикла построения схемы
```

```
line ([xa (k) xc (k) xd (k)], ...
```

[ya (k) yc (k) yd (k)]) % Ломаная ACD

polzun (Ks ,Ls ,xb (k) ,yb (k) ,fi 2(k)) % Кулисный камень

```
line([xa(k)    xb(k)], [ya(k)    yb(k)]) % Кривошип АВ
```

```
circle(r,xa(k),ya(k)) % Шарик А
```

```
circle(r,xb(k),yb(k)) % Шарнир В
```

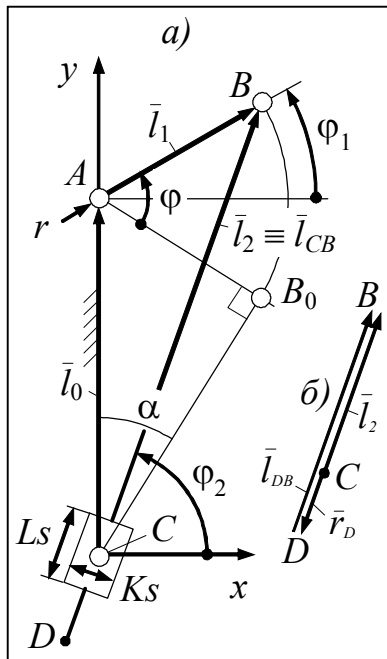
```
circle(r,xc(k),yc(k)) % Шарнир С
```

```
end % Конец цикла
```

axis equal % Установка равных масштабов по осям коорд.

clear % Очистка рабочей области

Механизм с кулисой плоско-параллельного движения



Координаты шарниров A, B, C (вид a) очевидны (см. программу). Для определения координат точки D введём радиус-вектор $\vec{r}_D = \vec{l}_2 - \vec{l}_{DB}$ (вид b).

Проекции радиус-вектора дают:

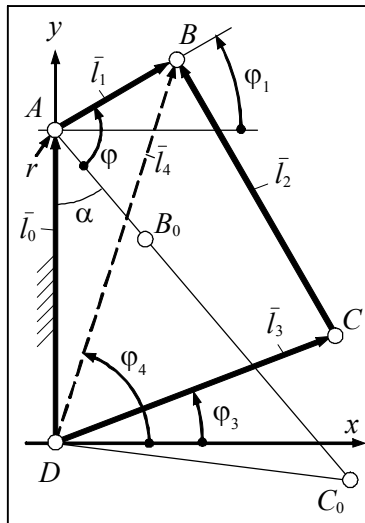
$$x_d = (l_2 - l_{db}) \cos \varphi_2, \quad y_d = (l_2 - l_{db}) \sin \varphi_2.$$

Программа

```
%----- Постоянные параметры схемы -----
L0=120; L1=40; alfa=asin(L1/L0);
r=0.06*L1;
Ks=4*r; Ls=2*Ks; Ldb=L0+L1+ Ls;
n=12; % Число планов положений
h=2*pi/n; % Шаг кривошипа

fi=h:h:2*pi; % Входная координата
% ----- Прочие параметры и координаты -----
fi1=fi+alfa;
L2=sqrt(L1^2+L0^2+2*L1*L0*sin(fi1));
fi2=acos(L1*cos(fi1)./L2);
z=zeros(1,n); % Вспомогательный массив из n нулей
xa=z; xc=z; yc=z;
ya=L0*ones(1,n); % ones - массив из n единиц
xb=L1*cos(fi1); yb=L0+L1*sin(fi1);
xd=(L2 -Ldb).*cos(fi2); yd=(L2 -Ldb).*sin(fi2);
%----- Построение планов положений -----
for k=1:n % Начало цикла построения схемы
    line([xa(k) xb(k) xd(k)], ...
         [ya(k) yb(k) yd(k)]) % Ломаная ABD
    polzun(Ks,Ls,xc(k),yc(k),fi2(k)) % Кулисный камень
    line([xa(k) xc(k)], [ya(k) yc(k)]) % Стойка
    circle(r,xa(k),ya(k)) % Шарнир A
    circle(r,xb(k),yb(k)) % Шарнир B
    circle(r,xc(k),yc(k)) % Шарнир C
end % Конец цикла
axis equal % Установка равных масштабов по осям коорд.
clear % Очистка рабочей области
```

Шарнирный четырёхзвенник



Координаты шарниров A, B, C, D очевидны (см. программу).

Программа

```
%----- Постоянные параметры схемы -----
L0=110; L1=40; L2=140; L3=100; Ls=L1+L2;
alfa=acos((L0^2+Ls^2-L3^2)/(2*L0*Ls));
r=0.07*L1;
n=12; % Число планов положений
h=2*pi/n; % Шаг кривошипа
fi=h:h:2*pi; % Входная координата
%----- Прочие параметры и координаты -----

fi1=fi+alfa-pi/2;
L4=sqrt(L0^2+L1^2+2*L0*L1*sin(fi1));
fi4=acos(L1*cos(fi1)/L4);
fi3=fi4-acos((L3^2+L4^2-L2^2)/(2*L3*L4));
z=zeros(1,n); % Вспомогательный массив из n нулей
xa=z; ya=L0*ones(1,n); % ones - массив из n единиц
xb=L1*cos(fi1); yb=L0+L1*sin(fi1);
xc=L3*cos(fi3); yc=L3*sin(fi3);
xd=z; yd=z;

%----- Построение планов положений -----
for k=1:n % Начало цикла построения схемы
    line([xa(k) xb(k) xc(k) xd(k) xa(k)], ...
        [ya(k) yb(k) yc(k) yd(k) ya(k)]) % 4-угольник ABCD
    % Шарниры - A, B, C, D
    circle(r, xa(k), ya(k)), circle(r, xb(k), yb(k))
    circle(r, xc(k), yc(k)), circle(r, xd(k), yd(k))
end % Конец цикла
axis equal % Установка равных масштабов по осям коорд.
clear % Очистка рабочей области
```


Анимация

Анимация состоит в показе планов положений по отдельности. Для этого командой **getframe** (захват кадра) каждый план заносят в некоторый массив. Затем с помощью команды **movie** планы показывают.

Чтобы каждый план строился в окне одного и того же размера и шарниры оставались круглыми, размеры окна указывают принудительно. Это делают с помощью функции **axis**, аргументами которой является массив из размеров x_{\min} , x_{\max} , y_{\min} , y_{\max} , снятых с окна планов положений. Размеры этого окна сформировались исходя из равенства масштабов по осям, поэтому команда **axis equal** не требуется и её удаляют.

Чтобы планы положений не накапливались, первой командой после **for k=1:n** должна быть **plot**, а не **line**. Это объясняется тем, что **plot** очищает окно, а **line** этого не делает.

Превращение планов положений в анимацию

1. В файле **Animation** замените **n=12** на **n=24**. Комментарий **Построение планов положений** замените на **Анимация**.
2. Удалите строки **axis equal** и **clear**.
3. Измените программу по аналогии с примером, приведённым ниже.
4. Запустите анимацию, результат предьявите преподавателю.

Анимация кривошипно-ползунного механизма

```
%----- Постоянные параметры схемы -----
L1=40; L2=120; r=0.06*L1;
... Прочие строки программы.....
for k=1:n % Начало цикла построения планов положений
    plot([xa,xd],[ya,yd]) % line заменено на plot
    Прочие строки цикла, -
    остаются без изменений

    .....
    axis([-100 158 -50 153]) % Размеры осей
    M(k)=getframe; % Захват кадра в массив M
end
movie(M,5,24) % Показать M 5 раз с частотой 24 кадра /с
```

Занятие 4. Силовой расчёт рычажного механизма Формирование массивов внешней силы и сил инерции

Цель данного и двух последующих занятий состоит в освоении методики компьютерного силового расчёта рычажных механизмов. Полный пример силового расчёта дан в прил. 2.

Задача силового расчёта в целом

Кривошип I механизма (см. своё задание) с известными размерами, массами и моментами инерции звеньев вращается с заданной скоростью ω_1 . Механизм находится под действием двух внешних сил. Одна из сил представляет собой вектор F , другая – момент M . В большинстве заданий M – это движущий момент, а F – сила полезного сопротивления, в остальных заданиях – наоборот. Для силы F дано максимальное значение F_{\max} и график $f(s)$ относительного значения f в функции от относительной координаты s точки приложения. Смысл относительных значений поясняют формулы:

$$s = S / S_{\max}; \quad f = F / F_{\max}.$$

По исходным данным требуется определить реакции связей во всех кинематических парах механизма, а также момент M . Всё это нужно сделать для 24-х равноотстоящих положений кривошипа AB , начиная от так называемого стартового положения, при котором $s = 0$. Результаты расчёта должны быть представлены в виде годографов реакций и графика момента M .

Допущения при расчёте: звенья абсолютно твёрдые, трения и сил тяжести нет; массы и моменты инерции, не указанные в задании, пренебрежимо малы либо не требуются.

Задача данного занятия состоит в формировании массива силы F , а также массивов сил инерции. Поскольку $F = f F_{\max}$, то всё сводится к определению f . Создание массива f называется далее табуляцией внешней силы.

Табуляция внешней силы

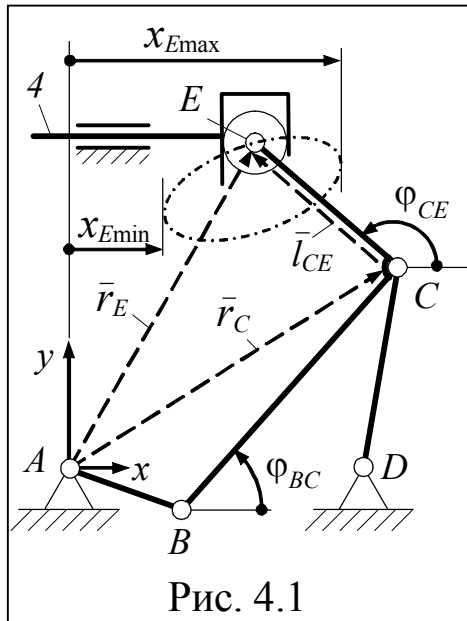
Каждой координате φ_1 кривошипа соответствует своя координата s и своя сила f , заданная графиком $f(s)$, поэтому, чтобы воспользоваться графиком, необходимо сначала сформировать массив значений s .

Во всех заданиях, за исключением № 12, механизмы отличаются от исследованных на занятиях 1...3 наличием рычага, на конец которого через ролик действует сила F , поэтому формирование массива s

начинается с вывода координат центра ролика, а в задании 12 – координат шарнира C .

Координаты центра ролика

Пример. К шатуну BC шарнирного четырёхзвенника $ABCD$ с известными углами звеньев и координатами всех шарниров прикреплён под прямым углом рычаг CE известной длины. Требуется определить координаты точки E (рис. 4.1).



Решая задачу методом векторных контуров, вводят радиус-векторы \vec{r}_E , \vec{r}_C и вектор \vec{l}_{CE} . Первый радиус-вектор направлен в точку с искомыми координатами, второй – в ближайшую точку с известными координатами, вектор \vec{l}_{CE} характеризует длину и ориентацию вновь введённого элемента схемы – рычага CE . Как видно по рисунку, $\vec{r}_E = \vec{r}_C + \vec{l}_{CE}$. Проекции вектора \vec{r}_E – это искомые координаты точки E :

$$x_E = x_C + l_{CE} \cos \varphi_{CE}, \quad y_E = y_C + l_{CE} \sin \varphi_{CE},$$

где $\varphi_{CE} = \varphi_{BC} + \pi/2$.

Помимо текущих координат, подобных x_E , y_E , необходимо знать координаты выходного звена 4 в его крайних положениях. В данном примере это координаты x_{Emin} и x_{Emax} .

По способу определения таких координат все задания делятся на две группы. В первой группе (задания 2, 3, 5, 6, 7, 10, 12) эти координаты могут быть найдены принципиально точно, во второй (задания 1, 4, 8, 9, 11) приближённо. Для сравнения точное и приближённое решения показаны ниже на примере одного и того же механизма (рис. 4.2, а).

Точное решение

Ползун 3 достигает крайних положений, когда кривошип AB и шатун BC либо вытягиваются в одну прямую – ABC_0 (рис. 4.2, б), либо складываются в одну прямую – BAC_t (рис. 4.2, в). На этом основании из рис. 4.2, б, в последовательно получают:

$$y_{Cmax} = l_{DC_0} = \sqrt{(l_{BC} + l_{AB})^2 - l_{DA}^2}; \quad y_{Cmin} = l_{DC_t} = \sqrt{(l_{BC} - l_{AB})^2 - l_{DA}^2}.$$

Для дальнейшего решения задачи табуляции выводят:

$S_{\max} = y_{C_{\max}} - y_{C_{\min}}$; $\varphi_{1s} = \pi - \alpha = \pi - \arccos(l_{AD} / (l_{AB} + l_{BC}))$ (см. рис. 4.2, б);
 $s = S / S_{\max}$. По результатам вычислений строят график координаты s (см. ниже файл-сценарий **tabulation 1**).

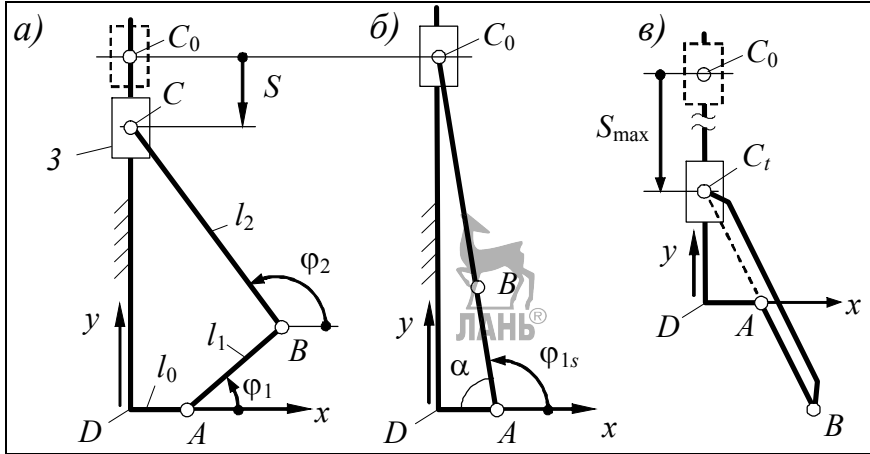


Рис. 4.2

Файл-сценарий точного решения – **tabulation 1**

```

1 L0=0.015; L1=0.04; L2=3.5*L1; % Размеры звеньев
2 fils=pi -acos(L0/(L1+L2));
3 n=24; h=2*pi/n; % Число положений и шаг вычислений
4 fi1=fil+(h:h:2*pi);
5 fi2=acos((-L1*cos(fi1) -L0)/L2);
6 yc=L1*sin(fi1)+L2*sin(fi2);
7 ycMax=sqrt((L2+L1)^2 -L0^2);
8 ycMin=sqrt((L2 -L1)^2 -L0^2);
9 Smax=ycMax -ycMin; s=(ycMax -yc)/Smax;
10 figure % Открытие окна для графика number (s)
11 number=1:n; % Номера положений
12 plot(s, number, 'Marker', '+'), grid % График
13 xlabel('s'), ylabel('number') % Обозначения осей

```

Результат выполнения этого сценария показан на рис. 4.3.

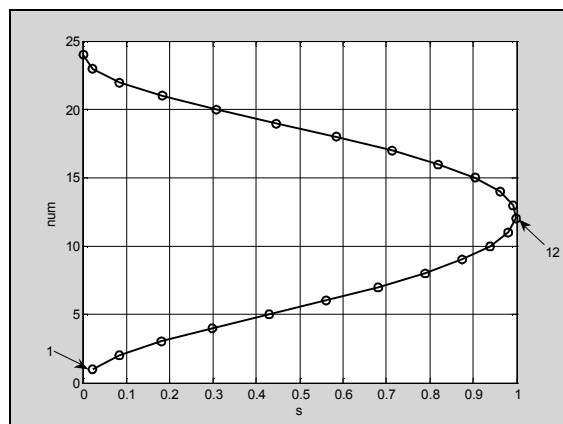


Рис. 4.3

Маркеры пронумерованы инструментами ручного редактирования.

Приближённое решение

Координату y_c (см. рис. 4.2, а) вычисляют в этом случае с достаточно мелким шагом, например $2\pi/360$. С помощью функций **max(...)**, **min(...)** находят экстремальные значения координаты. Например, **ycMax=max(yc)** означает: переменной **ycMax** присвоить максимальное значение координаты **yc**.

Помимо экстремальных координат, находят их индексы. Это делают с помощью функции **find(...)** – найти. В скобках указывают условие поиска. Например, **iMax=find(yc==ycMax)** означает: найти индекс того элемента в массиве **yc**, который равен **ycMax**, и присвоить этот индекс переменной **iMax**. Коротко говоря, это означает: найти индекс максимального элемента в массиве **yc** и присвоить этот индекс переменной **iMax**. Определяют также начальную или, иначе, стартовую координату **fi1s** кривошипа.

Все эти вычисления делают с помощью отдельной программы, называемой файл-функцией. Имя файл-функции и имя файла, в котором её сохраняют, должно быть одинаковым. В примере, приводимом ниже, это имя – **extreme**. Ваша файл-функция и файл хранения должны иметь более сложное имя, состоящее, как и прежде, из названия группы, ваших инициалов и поясняющего слова, например, того же **extreme**. Matlab автоматически предложит вам правильное имя, вам останется только согласиться с ним.

Файл-функция **extreme**

```

1 function [ycMax, ycMin, fi1s]=extreme(n, L0, L1, L2)
2 h=2*pi/n; fi1=h:h:2*pi;
3 fi2=acos((-L1*cos(fi1) -L0)/L2);
4 yc=L1*sin(fi1)+L2*sin(fi2);
5 ycMax=max(yc); ycMin=min(yc);
6 iMax=find(yc==ycMax);
7 fi1s=fi1(iMax);

```

Слово **function** в первой строке говорит о том, что далее идёт файл-функция. Слева от знака равенства в квадратных скобках приведены величины, для вычисления которых создана функ-



ция. Справа – имя функции. В скобках[®] после имени – входные величины, называемые также входными параметрами или аргументами функции.

Файл-функцию вызывают из основной программы, называемой файл-сценарием. За основу файла-сценария имеет смысл принять файл **planes** предыдущего занятия. Для этого сохраните его под вашим именем с дополнением **tabulation**.

В примере, приведённом ниже, сценарий называется **tabulation 2**. В нём показано только то, что необходимо для определения координаты s .

Файл-сценарий приближённого решения – **tabulation 2**

```

1 L0=0.015; L1=0.04; L2=3.5*L1; % Размеры по рис. 4.2
2 n=360; % Число шагов для нахождения крайних положений
3 [ycMax, ycMin, fi1s]=extreme(n, L0, L1, L2);
4 n=24; h=2*pi/n; % Число положений и шаг вычислений
5 fi1=fi1s+(h:h:2*pi);
6 fi2=acos((-L1*cos(fi1) -L0)/L2);
7 yc=L1*sin(fi1)+L2*sin(fi2);
8 Smax=ycMax -ycMin;
9 s=(ycMax -yc)/Smax;
10 figure % Открытие окна для графика number (s)
11 number=1:n; % Номера положений
12 plot(s, number, 'Marker', '+'), grid % График
13 xlabel('s'), ylabel('number') % Обозначения осей

```

В строке 3 сценарий **tabulation 2** обращается к файл-функции **extreme**, которая, как видно по тексту этой функции, вычисляет **fi1**, **fi2**, **yc** для 360 положений механизма, т. е. с интервалом в 1° . В рабочее пространство файл-функция выводит только **ycMax**, **ycMin**, **fi1s**. Эти три числа берёт из рабочего пространства сценарий **tabulation 2** и использует у себя в строках 5 и 8. Огромное количество значений **fi1**, **fi2**, **yc** (360×3), вычисленных функцией **extreme**, автоматически стирается, освобождая память. Сценарий **tabulation 2** даёт такой же график (см. рис. 4.3), как и **tabulation 1**, но применять **tabulation 2** стоит только тогда, когда невозможно или трудоёмко получить точное решение.

График **number (s)** (см. рис. 4.3) и график внешней силы **f(s)**, взятый из задания (в папке MatLabUsers), копируют в отдельный документ Word (рис. 4.4), сохраняемый под вашим именем в папке Мои документы.

Чтобы свободно перемещать графики, им задают обтекание За текстом. Растяжением-сжатием графики выравнивают по ширине. Инструментами рисования редактора Word маркеры проецируют на график силы. В точках пересечения проецирующих линий с графиком считывают значения силы и заносят их в массив **f**, располагаемый после строки 13 (**xlabel ('s') , ...**).

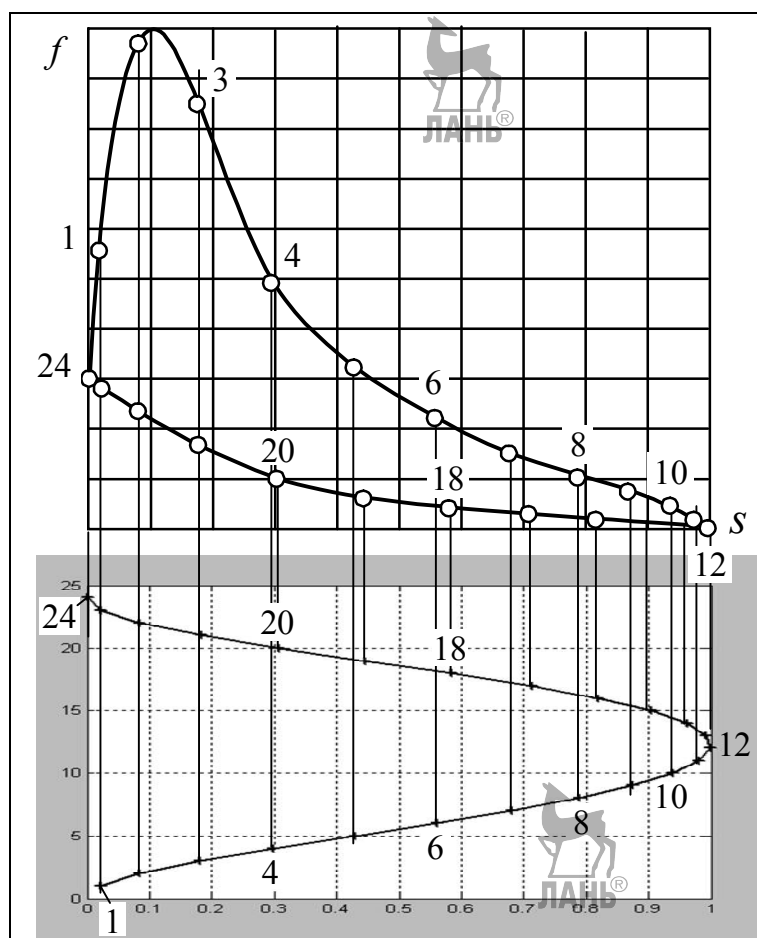


Рис. 4.4

В обсуждаемом примере массив содержит следующие числа:
 $f = [0.58 \ 0.98 \ 0.84 \ 0.49 \ 0.33 \ 0.23 \ 0.16 \ 0.11 \ 0.07 \ 0.05 \ 0.02 \ 0 \dots$
 $0 \ 0.005 \ 0.01 \ 0.02 \ 0.03 \ 0.04 \ 0.06 \ 0.09 \ 0.17 \ 0.23 \ 0.28 \ 0.3]; .$

Правильность массива **f** проверяют построением фактического графика **f(s)**:

```
figure % Открытие окна под график f(s)
plot(s, f, '-') % График, маркированный точками
axis equal, grid
```

По относительным значениям силы вычисляют абсолютные значения: $F=f \cdot F_{\max}$;.

Для проверки правильности определения координат строят обновлённые – по сравнению с предыдущим занятием – планы положений. При этом перед циклом **for ... end** построения планов добавляют команду **figure** .

Порядок решения задачи табуляции

1. Получите задание на силовой расчёт.

2. Выполните предложенное ниже.

В заданиях 2, 3, 5, 6, 7, 10, 12	В заданиях 1, 4, 8, 9, 11
<p>а) Руководствуясь материалами занятия 3 и вашим заданием, выведите формулу стартового значения φ_{1s} координаты φ_1.</p> <p>б) Выведите координаты центра ролика, входящие в формулу $s = \dots$, приведённую в задании.</p> <p>в) Откройте файл planes и сохраните его под вашим именем с добавлением tabulation .</p> <p>г) Обновите размеры звеньев.</p> <p>д) Добавьте формулы, выведенные в п. а) и б).</p> <p>е) Скорректируйте программу, подобно tabulation 1.</p>	<p>а) Руководствуясь материалами занятия 3 и вашим заданием, выведите формулы координат центра ролика.</p> <p>б) Откройте файл planes и сохраните его под вашим именем с добавлением tabulation .</p> <p>в) Обновите размеры звеньев.</p> <p>г) Добавьте формулы, выведенные в п. а).</p> <p>д) Составьте программы, подобные extreme и tabulation 2.</p>

3. Перед циклом построения планов положений (**for...end**) добавьте команду **figure** , а в цикл вставьте команду **line (...)** построения рычага, несущего ролик.

4. Запустите программу. Если всё правильно, появится график координаты s , как на рис. 4.3, и обновлённые планы положений.

5. В документ Word скопируйте графики силы f и координаты s .

6. Выводите графики по ширине, как показано на рис. 4.4.

7. Снимите значения силы и занесите их в массив **f**.

8. Проверьте правильность массива построением графика $f(s)$.

9. Создайте массив абсолютных значений силы F .

Определение сил инерции

В общем случае силы инерции звена i сводятся к главному вектору \bar{I}_i и главному моменту \bar{M}_i . При аналитических расчётах главный вектор раскладывают на составляющие по осям неподвижной системы координат. Составляющие прикладывают к центру масс звена и изображают так, чтобы их проекции на оси были положительными. Главный момент представляют своей проекцией M_i и изображают круговой стрелкой, имеющей положительное направление (рис. 4.5).

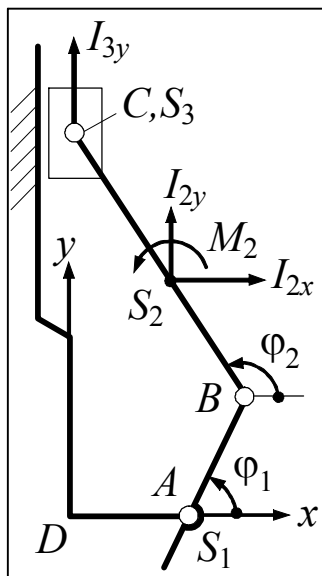


Рис. 4.5

На предыдущих занятиях производные находили аналитически, на данном занятии это будет сделано численно, причём центральными разностями. Согласно этому методу, вторую производную в точке дифференцирования C , называемой центральной (рис. 4.6), вычисляют по формуле

$$y'' = \frac{y(x_c + \Delta x) + y(x_c - \Delta x) - 2y(x_c)}{2\Delta x}, \quad (4.4)$$

где в числителе последовательно представлены значения функции в точках R , L и C .

Ниже приведена программа дифференцирования и вычисления сил инерции звеньев механизма, изображённого на рис. 4.5. При этом предполагается, что массивы переменных **fi1**, **fi2**, **xb**, **yb**, **yc**,

При плоской системе сил и равномерном вращении кривошипа I , как принято в заданиях, проекции главного вектора и главного момента на оси неподвижной системы координат вычисляют по формулам:

$$I_{ix} = -m_i \omega_1^2 x_{S_i}''; \quad (4.1)$$

$$I_{iy} = -m_i \omega_1^2 y_{S_i}''; \quad (4.2)$$

$$M_i = -J_{S_i} \omega_1^2 \varphi_i''. \quad (4.3)$$

Как следует из этих формул, для вычисления сил инерции достаточно знать только вторую производную от соответствующей координаты.

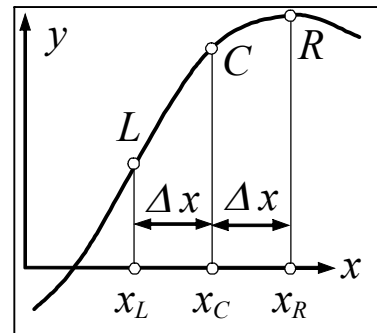


Рис. 4.6

xs2, **ys2** в точке дифференцирования *C* уже созданы, т. е. существуют формулы **fi1=...**; , **fi2=...**; , и т. д., и остаётся добавить:

```

1 dfi=0.02; % Приращение угла fi1 (≈1,15 °)
2 fi1L=fi1-dfi; fi1R=fi1+dfi; % Левые и прав. знач. аргумента
   % Левые и прав. знач. функций:
3 fi2L=acos((-L1*cos(fi1L)-L0)/L2);
4 fi2R=acos((-L1*cos(fi1R)-L0)/L2);
5 xbL=L0+L1*cos(fi1L);   xbR=L0+L1*cos(fi1R);
6 ybL=L1*sin(fi1L);   ybR=L1*sin(fi1R);
7 ycL=ybL+L2*sin(fi2L);   ycR=ybR+L2*sin(fi2R);
8 xs2L=xbL+Lbs2*cos(fi2L);   xs2R=xbR+Lbs2*cos(fi2R);
9 ys2L=ybL+Lbs2*sin(fi2L);   ys2R=ybR+Lbs2*sin(fi2R);
10 xs2w=(xs2R+xs2L-2*xs2)/dfi^2; % Вторые производные
11 ys2w=(ys2R+ys2L-2*ys2)/dfi^2;
12 fi2w=(fi2R+fi2L-2*fi2)/dfi^2;
13 ycw=(ycR+ycL-2*yc)/dfi^2;
14 clear fi1L fi1R xbL xbR % Удаление отработавших величин
15 clear ybL ybR ycL ycR fi2L fi2R xs2L ys2L ys2R
16 omq=omega^2;
16 I2x=-m2*omq*xs2w; I2y=-m2*omq*ys2w;
17 M2=-J2*omq*fi2w; I3y=-m3*omq*ycw;
18 figure, plot(I2x,I2y), grid % Годограф вектора I2

```

Порядок определения сил инерции

1. Переименуйте файл **tabulation** в **forceAnalis** .
2. Завершите ввод исходных данных, содержащихся в задании.
3. Добавьте формулы координат центров масс S_i .
4. Вычислите силы инерции.

Файл-сценарий, созданный на данном этапе, будет выглядеть примерно так, как показано в строках 1...66 прил. 2.

Занятие 5. Силовой расчёт рычажного механизма

Составление и решение уравнений равновесия

Для решения этой задачи механизм раскладывают на отдельные звенья. К каждому звену прикладывают внешние силы, реакции связей и силы инерции. Все силы представляют своими положительными проекциями. Согласно принципу Даламбера, приложение сил инерции приводит звенья в состояние равновесия. Из совместного решения уравнений равновесия находят реакции и внешний момент M , приложенный к кривошипу.

Предпочтительны уравнения равновесия в форме суммы проекций и суммы моментов: $\sum X_i = 0$; $\sum Y_i = 0$; $\sum M_i = 0$.

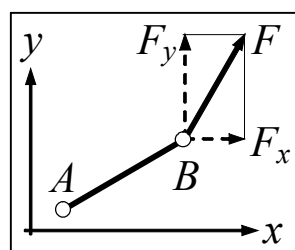


Рис. 5.1

Момент какой-либо силы F относительно некоторой точки A звена AB (рис. 5.1) выражают через проекции силы по формуле

$$M_A(F) = F_y(x_B - x_A) - F_x(y_B - y_A). \quad (5.1)$$

Вид формулы не зависит от взаимного положения точек A и B . Полезно запомнить, что на первом месте в скобках всегда стоит координата точки приложения силы. Кроме того, перед F_y всегда ставится «плюс», а перед F_x «минус».

Пример 1. Требуется составить и решить уравнения равновесия звеньев двигателя, изображённого на рис. 5.2, а.

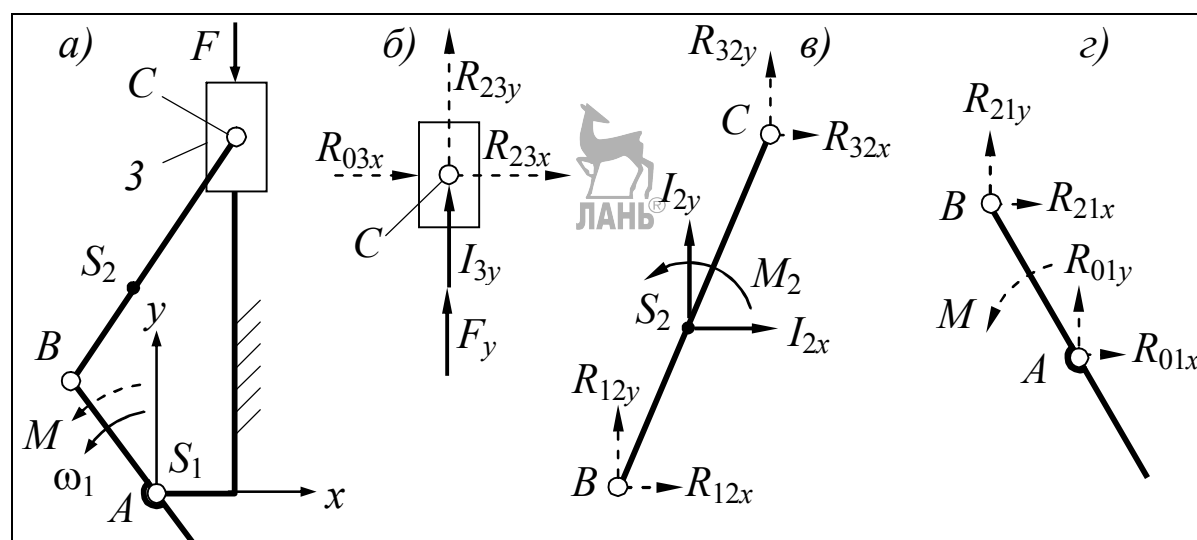


Рис. 5.2

Составление уравнений равновесия. Согласно рекомендациям, приведённым выше, механизм разложен на отдельные звенья (рис. 5.2, б, в, г); все силы и моменты представлены своими поло-

жительными проекциями. Для моментов положительным является направление против часовой стрелки. Для чёткого разграничения известных и искомых величин последние выделены пунктиром. Из равновесия звеньев получены следующие 8 уравнений (табл. 5.1).

Таблица 5.1

№ зв.	Вид	№	Уравнения равновесия звеньев
3	$\Sigma X=0$	1	$R_{23x} + R_{03x} = 0$
	$\Sigma Y=0$	2	$R_{23y} + F_y + I_{3y} = 0$
2	$\Sigma X=0$	3	$R_{32x} + R_{12x} + I_{2x} = 0$
	$\Sigma Y=0$	4	$R_{32y} + R_{12y} + I_{2y} = 0$
	$\Sigma M_B=0$	5	$R_{32y}(x_c-x_b) - R_{32x}(y_c-y_b) + I_{2y}(x_{S_2}-x_B) - I_{2x}(y_{S_2}-y_B) + M_2 = 0$
1	$\Sigma X=0$	6	$R_{21x} + R_{01x} = 0$
	$\Sigma Y=0$	7	$R_{21y} + R_{01y} = 0$
	$\Sigma M_A=0$	8	$R_{21y}(x_B-x_A) - R_{21x}(y_B-y_A) + M = 0$

Представление уравнений в матричной форме. Уравнения равновесия представляют собой систему вида

[illegible]

В матричной форме эту систему записывают как $Ax = b$, где A – квадратная матрица из коэффициентов при неизвестных, x – столбец неизвестных, b – столбец свободных членов. Пусть столбец x соответствует табл. 5.2.

Таблица 5.2

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
R_{01x}	R_{01y}	R_{12x}	R_{12y}	R_{23x}	R_{23y}	R_{03x}	M

Порядок присваиваний в таблице – произвольный. В данном случае реакции расположены в порядке возрастания номеров индексов: 0-1, 1-2, 2-3, 0-3.

В таблицу не вошли реакции с обратным чередованием индексов – 1-0, 2-1, 3-2, 3-0: они будут учтены в матрице A на основании равенств действия и противодействия, например: $R_{10} = -R_{01}$.

На основании табл. 5.1, 5.2 и равенств вида $R_{ji} = -R_{ij}$ получают следующие матрицы A и b :

A=	0	0	0	0	1	0	1	0	1
	0	0	0	0	0	1	0	0	2
	0	0	1	0	-1	0	0	0	3
	0	0	0	1	0	-1	0	0	4
	0	0	0	0	y_c-y_b	$-x_c+x_b$	0	0	5
	1	0	-1	0	0	0	0	0	6
	0	1	0	-1	0	0	0	0	7
	0	0	y_b	x_b	0	0	0	1	8
$x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8$									
$R_{01x} \quad R_{01y} \quad R_{12x} \quad R_{12y} \quad R_{23x} \quad R_{23y} \quad R_{03x} \quad M$									

0
$F-I_{3y}$
$-I_{2x}$
$-I_{2y}$
$-I_{2y}(x_{S_2}-x_B)+I_{2x}(y_{S_2}-y_B)-M_2$
0
0
0

```

b=[0; F(i) -I3y(i); -I2x(i); -I2y(i);...
    -I2y(i)*(xs2(i) -xb(i))+ ...
    I2x(i)*(ys2(i) -yb(i)) -M2(i); 0; 0; 0];
x=A\b; % Вычисление корней системы
% Заполнение массивов R01x, R01y и т. д. корнями
системы
R01x(i)=x(1); R01y(i)=x(2);
R12x(i)=x(3); R12y(i)=x(4);
R23x(i)=x(5); R23y(i)=x(6);
R03x(i)=x(7); M(i)=x(8);
end

```

Проверка силового расчёта

Расчёт сделан для установившегося движения машины. При таком движении работа силы F и работа момента M за цикл (в наших заданиях – за один оборот кривошипа) должны быть равны по модулю. Это и есть критерий правильности.

Работа силы F за цикл пропорциональна площади, ограниченной графиком $f(s)$. Работа момента M равна его среднему значению, умноженному на угол поворота звена приложения, т. е. на 2π . Допустимая погрешность расчёта $\delta \leq 2\%$.

Программа проверки

```

AF=polyarea (s,f)*Fmax*Smax; % Работа силы F
% polyarea (s,f) - площадь многоугольника f(s)
Mm=mean (M); % Среднее значение момента M
AM=Mm*2*pi % Работа момента M
delta =abs ((AF+AM)/AF)*100 % Ошибка в процентах, выводится
в окно команд. abs - абсолютное значение. AF и AM
на самом деле вычитаются, т. к. имеют разные знаки

```

Мощность двигателя

В результате силового расчёта определяют также мощность N , развиваемую двигателем. При расчёте машин-орудий, т. е. не двигателей, определяют потребляемую мощность. Как развиваемую, так и потребляемую мощность определяют по программе $N=abs (Mm) * \omega$.

Пример 2. Механизм с кулисой (рис. 5.3)

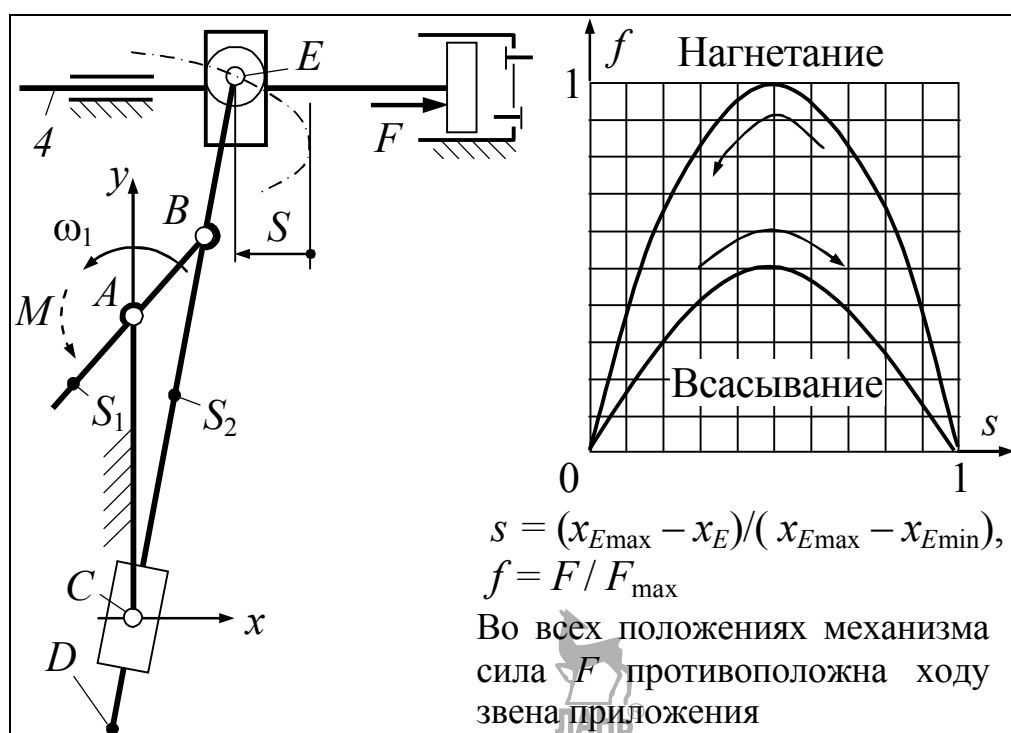


Рис. 5.3

Для упрощения задачи к звеньям, подобным 4, прикладывают только горизонтальные силы (рис. 5.4, б). При табуляции следует учитывать, что при всасывании $F_x > 0$, при нагнетании $F_x < 0$.

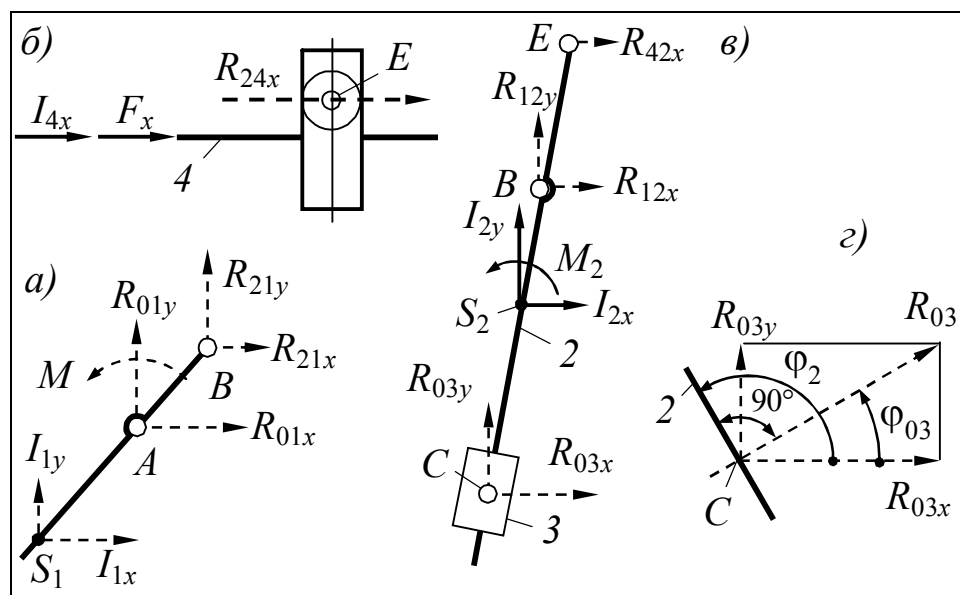


Рис. 5.4

При наших допущениях реакция R_{03} всегда перпендикулярна кулисе 2 (рис. 5.4, г), поэтому $R_{03x} = R_{03} \cos \varphi_{03}$, $R_{03y} = R_{03} \sin \varphi_{03}$, где $\varphi_{03} = \varphi_2 - \pi/2$. Две неизвестные – R_{03x} и R_{03y} – сводятся в данном случае к одной – R_{03} . При $R_{03} > 0$ её направление такое, как на рисунке, при $R_{03} < 0$ противоположно.

Уравнения равновесия звеньев по рис. 5.4.

Зв.	Вид	Уравнение
1	$\Sigma X=0$	$R_{21x}+R_{01x}+I_{1x}=0$
	$\Sigma Y=0$	$R_{21y}+R_{01y}+I_{1y}=0$
	$\Sigma M_A=0$	$R_{21y}(x_B-x_A)-R_{21x}(y_B-y_A)+I_{1y}(x_{S_1}-x_A)-I_{1x}(y_{S_1}-y_A)+M=0$
2	$\Sigma X=0$	$R_{42x}+R_{12x}+I_{2x}+R_{03}\cos \varphi_{03}=0$
	$\Sigma Y=0$	$R_{12y}+I_{2y}+R_{03}\sin \varphi_{03}=0$
	$\Sigma M_C=0$	$I_{2y}(x_{S_2}-x_C)-I_{2x}(y_{S_2}-y_C)+R_{12y}(x_B-x_C)-R_{12x}(y_B-y_C)-R_{42x}(y_E-y_C)+M_2=0$
4	$\Sigma X=0$	$I_{4x}+F_x+R_{24x}=0$

Обозначения	Исходные	R_{01x}	R_{01y}	R_{12x}	R_{12y}	R_{24x}	R_{03}	M
	Новые	x_1	x_2	x_3	x_4	x_5	x_6	x_7

Уравнения равновесия после переобозначений и переноса известных величин в правую часть.

1	$\Sigma X=0$	$-x_3+x_1=-I_{1x}$
	$\Sigma Y=0$	$-x_4+x_2=-I_{1y}$
	$\Sigma M_A=0$	$-x_4(x_B-x_A)+x_3(y_B-y_A)+x_7=-I_{1y}(x_{S_1}-x_A)+I_{1x}(y_{S_1}-y_A)$
2	$\Sigma X=0$	$-x_5+x_3+x_6\cos \varphi_{03}=-I_{2x}$
	$\Sigma Y=0$	$x_4+x_6\sin \varphi_{03}=-I_{2y}$
	$\Sigma M_C=0$	$x_4(x_B-x_C)-x_3(y_B-y_C)+x_5(y_E-y_C)=-I_{2y}(x_{S_2}-x_C)+I_{2x}(y_{S_2}-y_C)-M_2$
4	$\Sigma X=0$	$x_5=-I_{4x}-F_x$

Матрицы A и b

$A=$	1	0	-1	0	0	0	0
	0	1	0	-1	0	0	0
	1	0	y_B-y_A	x_A-x_B	0	0	1
	0	0	1	0	-1	$\cos \varphi_{03}$	0
	0	0	0	1	0	$\sin \varphi_{03}$	0
	0	0	y_C-y_B	x_B-x_C	y_E-y_C	0	0
	0	0	0	0	1	0	0
$x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7$							

$b=$	$-I_{1x}$
	$-I_{1y}$
	$-I_{1y}(x_{S_1}-x_A)+I_{1x}(y_{S_1}-y_A)$
	$-I_{2x}$
	$-I_{2y}$
	$-I_{2y}(x_{S_2}-x_C)+I_{2x}(y_{S_2}-y_C)-M_2$
	$-I_{4x}-F_x$

Далее задача решается, как в предыдущем примере.

Занятие 6. Силовой расчёт рычажного механизма

Построение годографов

Это занятие – заключительное, из трёх, отводимых на силовой расчёт механизма. Задача занятия – вывод результатов расчёта в виде годографов и графиков.

Годографы реакций и сил инерции. Годографы каких-либо сил – это траектории, которые описывают относительно звена концы векторов этих сил, выходя из точки приложения, например из центра масс или шарнира. Годографы строят выборочно, но обязательно среди них должны быть:

- годографы сил, приложенных к шатуну или кулисе (в зависимости от вида механизма);
- годограф реакции этого же звена на кривошип;
- для кривошипно-ползунных механизмов должен быть график давления ползуна на направляющую.

Чтобы построить годограф какой-либо силы, например реакции \bar{R}_{12} (рис. 6.1), приложенной к подвижному звену BC , её проекции R_{12x} ,

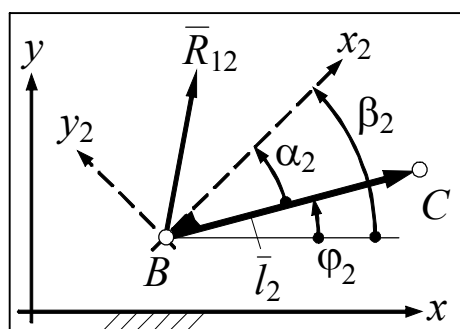


Рис. 6.1

R_{12y} на оси x , y неподвижной системы координат преобразуют в проекции R_{12x_2} , R_{12y_2} на оси x_2 , y_2 подвижной системы координат, неизменно связанной со звеном BC . Угол α_2 оси x_2 относительно вектора \bar{l}_2 , связанного со звеном BC , выбирают

произвольно, но лучше так, чтобы ось x_2 оказалась расположенной вдоль или попе-

рёк звена.

Полагая, что угол φ_2 вектора \bar{l}_2 относительно оси x известен, вычисляют угол β_2 оси x_2 относительно x : $\beta_2 = \varphi_2 + \alpha_2$. Правило знаков для всех углов прежнее: положительные углы отсчитывают против часовой стрелки. Проекции преобразуют по формулам:

$$R_{12x_2} = R_{12x} \cos \beta_2 + R_{12y} \sin \beta_2; \quad (6.1)$$

$$R_{12y_2} = -R_{12x} \sin \beta_2 + R_{12y} \cos \beta_2. \quad (6.2)$$

После вычислений строят годограф. Следующий фрагмент программы иллюстрирует построение годографов всех сил, приложенных к звену 2 (рис. 6.2, в).

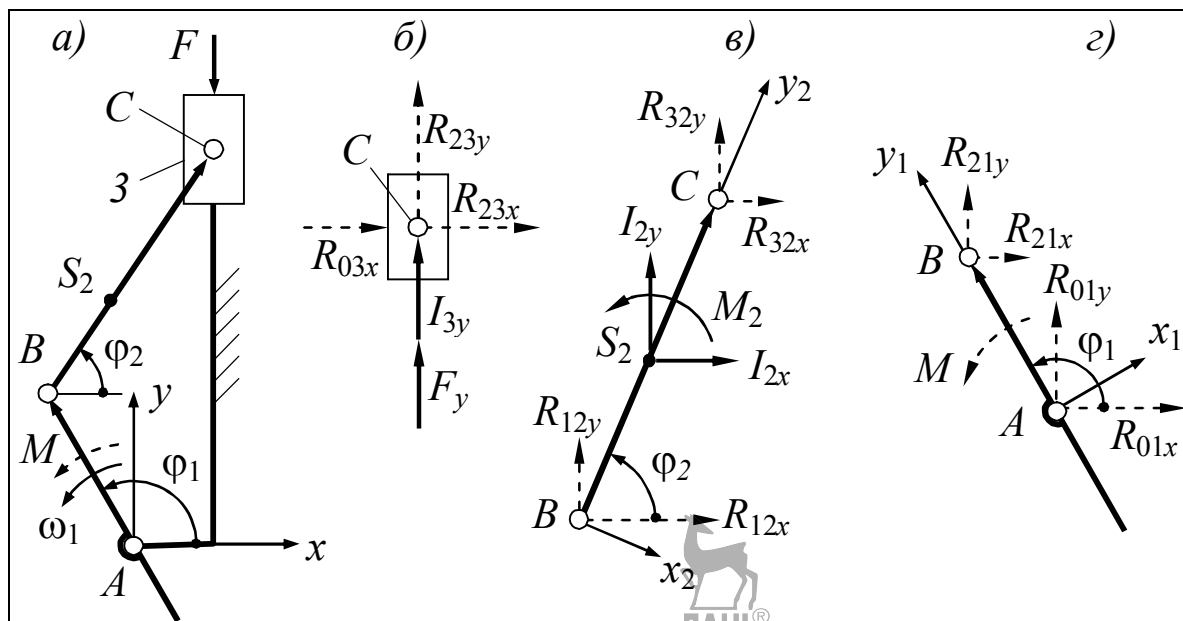


Рис. 6.2

```

74. % --- Преобразование R12, R32, I2 из сист. ху в x2y2
75. bet2=fi2 -pi/2; % Угол оси x2 относительно х
76. R12x2=R12x.*cos(bet2)+R12y.*sin(bet2);
77. R12y2= -R12x.*sin(bet2)+R12y.*cos(bet2);
78. R32x=-R23x; R32y=-R23y; % Замена действия прот и-
    воздействием
79. R32x2=R32x.*cos(bet2)+R32y.*sin(bet2);
80. R32y2= -R32x.*sin(bet2)+R32y.*cos(bet2);
81. I2x2=I2x.*cos(bet2)+I2y.*sin(bet2);
82. I2y2= -I2x.*sin(bet2)+I2y.*cos(bet2);
83. % Построение годографов R12, R32, I2 из единого центра
84. figure
85. plot(R12x2,R12y2,'.--',R32x2, R32y2,'.-',...
86. I2x2, I2y2, '.-'), axis equal, grid
87. title('Gdografs relatively x2y2 from one center')

```

На этом этапе годографы выглядят, как показано на рис. 6.3. Центры всех трёх годографов – точки B , S_2 и C (см. рис. 6.2, в) находятся в начале координат. Размеры графического окна слишком малы, чтобы разнести эти точки по вертикали (по оси y_2) и в то же время избежать пересечения годографов, поэтому годографы сдвигают параллельно себе наискосок, оставляя годограф силы инерции на месте. Для этого, глядя на рис. 6.3, намечают новые координаты точек B и C :

```

88. Bx= -6000; By= -4000; Cx=6000; Cy=3000;

```

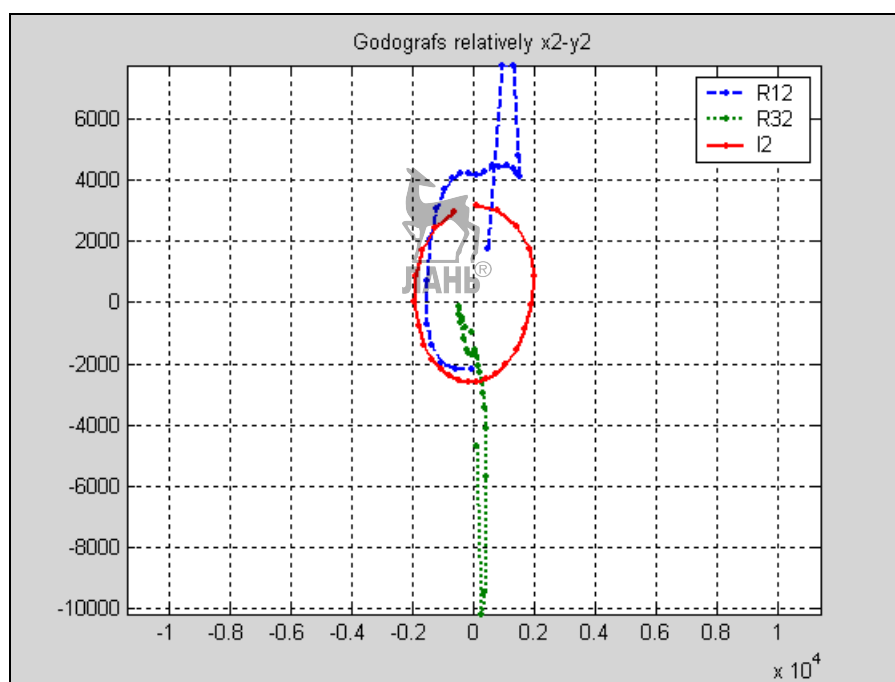


Рис. 6.3

Координаты являются единицами сил. Прибавляя их к существующим значениям, смещают годографы:

89. figure

90. plot (R12x2+Bx, R12y2+By, '-.', I2x2, I2y2, '-.',

91. R32x2+Cx, R32y2+Cy, '-.', ...

Bx, By, 'ok', 0, 0, 'ok', Cx, Cy, 'ok') % Смещённые годографы
и метки точек B, S2, C

92. text(Bx, By, 'B'), text(0, 0, 'S2')

93. text(Cx, Cy, 'C') % Обозначения точек B, S2, C

% --- Нумерация точек 1 и 24 на годографах ---

94. text(R12x2(24)+Bx, R12y2(24)+By, '24'),

95. text(R12x2(1)+Bx, R12y2(1)+By, '1')

96. text(I2x2(24), I2y2(24), '24'),

97. text(I2x2(1), I2y2(1), '1')

98. text(R32x2(24)+Cx, R32y2(24)+Cy, '24')

99. text(R32x2(1)+Cx, R32y2(1)+Cy, '1')

100. grid, legend('R12', 'R32', 'I2', 0)

101. title('Godografs relatively x2y2')

Результат выполнения программы показан на рис. 6.4. Некоторые элементы этого и следующего рисунка отредактированы вручную инструментами графического окна.

Чтобы воспользоваться годографом, надо провести вектор из его центра в любую маркированную точку. Например, вектор $R_{12}(18)$ – это реакция R_{12} в положении 18 механизма. Номер по-

ложения установлен отсчёт от точки 24 в конце годографа. На годографе, расположенном в центре графического окна, показан главный вектор I_2 сил инерции звена 2 в положении 4.

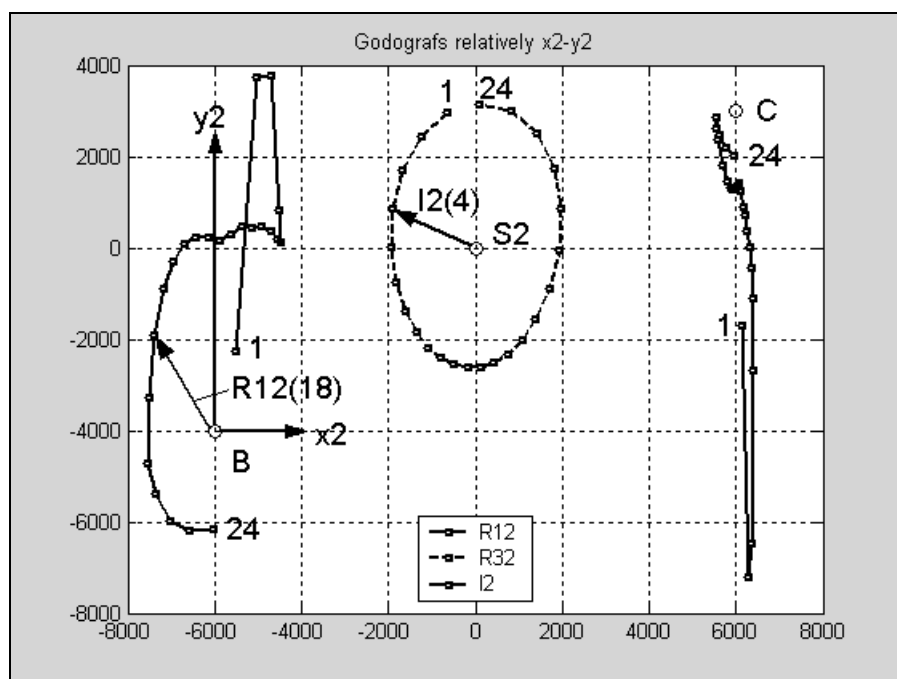


Рис. 6.4

Годографы проще строить по отдельности, однако при этом теряется возможность сопоставления сил из-за разных масштабов в каждом окне. Ниже приведена программа построения одиночного годографа – реакции R_{21} в системе x_1y_1 (см. рис. 6.2, з).

```
% ----- Преобразование R21 из сист. ху в x1y1 -----
102. bet1=fi 1-pi/2; % УГОЛ ОСИ x1 относительно х
103. R21x=-R12x ; R21y=-R12 y;
104. R21x1=R 21x.*cos(bet1) +R21y.*sin(bet1) ;
105. R21y1= -R21x.*sin(bet1) +R21y.*cos(bet1) ;
106. % Построение годограф а R21 в системе x1y1
107. figure
108. plot([R21x1(24) R21x1], ...
109. [R21y1(24) R21y1],'. -', 0,0,'ok') , grid
110. text (0,0,'B'), text(R21x1(24), R21y1(24),'24')
111. text(R21x1(1), R21y1(1),'1')
112. title('Godograf R21 relatively x1-y1')
```

Результат выполнения программы показан на рис. 6.5.

В отличие от предыдущих годографов, этот замкнут. Замыкание достигнуто (см. строки 108, 109) объединением 24-го эле-

мента массива **R21** со всем массивом. Замыкание делать не обязательно.

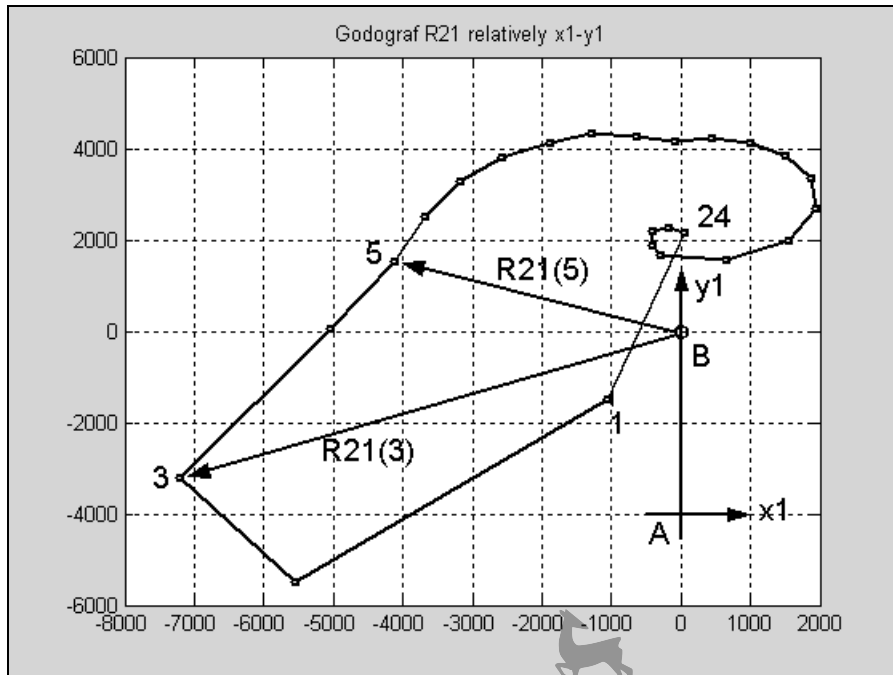


Рис. 6.5

График давления ползуна на направляющую – это график реакции R_{30x} в функции от относительной координаты s ползуна. Начало отсчёта этой координаты находится в верхней мёртвой точке ползуна, поэтому будет применено «матричное» расположение осей координат – с началом вверху.

Программа построения графика реакции R_{30x}

```
113. ti=12; % terminal index – индекс переменной в конце такта рас-
    ширения, установлен по разметке хода ползуна (см. рис. 4.3)
114. R30x=-R03x;
115. figure
116. plot(R30x(1: ti),s(1: ti),'. -',...
117. R30x( ti:end),s( ti:end),'* -'), grid % (ti:end) означает :
    от ti до конца массива
118. axis ij % "Матричное" располож. осей (начало коорд. – вверху)
119. legend('expansion','compression', 0)
120. xlabel('R30'), ylabel('s')
121. text(R30x(1), s(1), '1'), text(R30x(12), s(12), '12')
122. text(R30x(24), s(24), '24') % Здесь программно пронумерованы
    точки 1, 12, 24.
```

Результат выполнения программы показан на рис. 6.6. Вертикальная прямая, проходящая через ноль, изображает ось направляющей ползуна или, иначе, ось цилиндра.

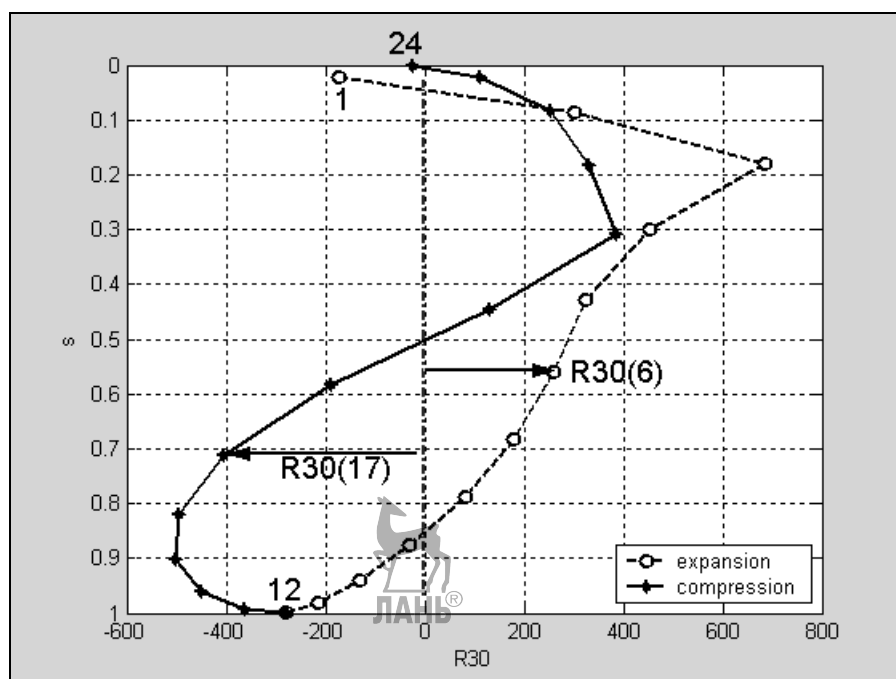


Рис. 6.6

По графику видно, например, что, проходя положение 6, поршень движется вниз и давит на стенку цилиндра вправо с силой 250 Н, а, проходя положение 17, движется вверх и давит влево с силой 400 Н.

Читая годографы, полезно держать перед глазами пронумерованные планы положений механизма. При нумерации любого объекта достаточно запрограммировать 2-3 базовых номера, остальные могут быть добавлены «вручную» при редактировании рисунка в графическом окне. Именно так создавались многие из приведённых выше рисунков. В некоторых случаях об этом делались предупреждения.

Порядок работы

1. Свяжите со звеном 1 систему координат x_1y_1 .
2. Определите угол α и вычислите программно β (см. рис. 6.1).
3. По формулам, подобным (6.1), (6.2), преобразуйте векторы программируемого годографа из системы $xу$ в x_1y_1 .
4. Введите в программу строки 107...112 и запустите её.
5. Инструментами открывшегося графического окна с изображением годографа отредактируйте его по образцу рис. 6.5.

Занятие 7. Кинематический анализ манипуляторов

Цель занятия – практическое освоение ещё одного (после векторных контуров) аналитического метода кинематического анализа механизмов – метода преобразования координат.

С этой целью для предложенного манипулятора требуется определить координаты конечной точки переносимой детали. Координаты выводятся методом преобразования координат. Вычисления производятся на компьютере. Координаты проверяются на модели манипулятора.

Краткие сведения из теории

Пусть поставленную задачу требуется решить для манипулятора, исходное положение которого изображено на рис. 7.1.

Из этого положения звено 2 повернулось относительно звена 1 на угол φ_{21} , а звено 3 повернулось относительно 2 на угол φ_{32} . Для достигнутого при этом положения требуется определить координаты точки E относительно системы координат $Ax_0y_0z_0$, связанной со стойкой 0.

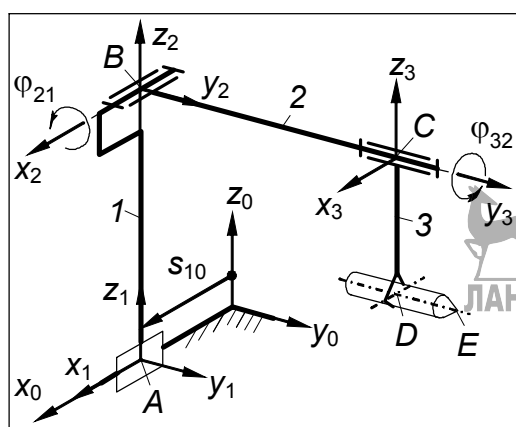


Рис. 7.1

Для решения задачи с каждым звеном механизма связывают систему координат, расположенную произвольно. Так со звеном 0 связана уже упомянутая система $Ax_0y_0z_0$, со звеном 1 связана система $Bx_1y_1z_1$, со звеном 2 – система $Cx_2y_2z_2$.

Координаты точки E последовательно преобразуют из системы 3 (системы $Cx_3y_3z_3$) в 2, из 2

в 1, из 1 в 0. В матричной форме преобразование имеет вид

$$R_{E_0} = T_{10} T_{21} T_{32} R_{E_3}, \quad (7.1)$$

где R_{E_0} и R_{E_3} – матрицы искоемых и заданных координат; T_{10} , T_{21} , T_{32} – матрицы преобразования координат.

$$R_{E_0} = \begin{bmatrix} x_{E_0} \\ y_{E_0} \\ z_{E_0} \\ 1 \end{bmatrix} \quad R_{E_3} = \begin{bmatrix} x_{E_3} \\ y_{E_3} \\ z_{E_3} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ L_{DE} \\ -L_{CD} \\ 1 \end{bmatrix}$$

В матрицах типа T_{ji} первая цифра индекса указывает, откуда ведётся преобразование, вторая – куда.

В общем случае матрица преобразования T_{ji} имеет вид

	x_j	y_j	z_j	
x_i	$\cos(x_i \wedge x_j)$	$\cos(x_i \wedge y_j)$	$\cos(x_i \wedge z_j)$	a
y_i	$\cos(y_i \wedge x_j)$	$\cos(y_i \wedge y_j)$	$\cos(y_i \wedge z_j)$	b
z_i	$\cos(z_i \wedge x_j)$	$\cos(z_i \wedge y_j)$	$\cos(z_i \wedge z_j)$	c
	0	0	0	1

(a, b, c – координаты начала системы j относительно i).

Для облегчения составления матрицы сверху располагают обозначения осей системы, из которой ведётся преобразование, сбоку – в которую. Под знаком косинуса указывают угол между теми осями координат, на пересечении которых находится рассматриваемая ячейка матрицы.

Четвёртая строка в матрицах типа T и R введена искусственно для обеспечения перемножения матриц (см. лекции по ТММ).

Чтобы определить угол между осями, систему j переносят параллельно самой себе до совмещения с i и поворачивают, если поворот предусмотрен, на небольшой положительный угол, независимо от знака угла по заданию. Угол считается положительным, если поворот, наблюдаемый из конца в начало оси поворота, происходит против часовой стрелки (рис. 7.2).

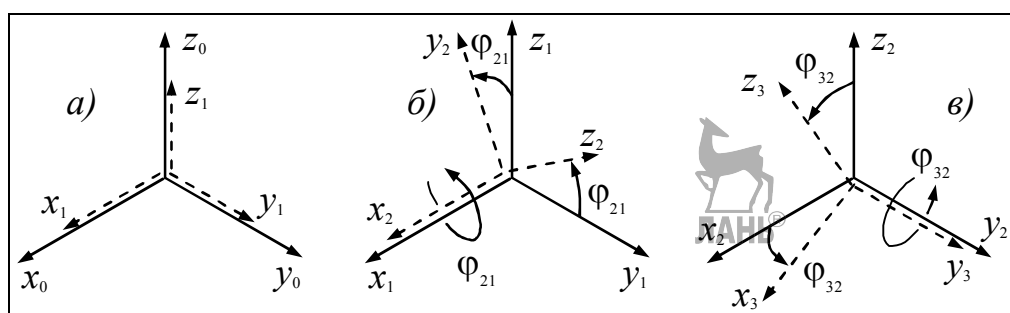


Рис. 7.2

Так, например, угол между осями y_1, y_2 (рис. 7.2, б) равен $\varphi_{21} + 90^\circ$, $\cos(\varphi_{21} + 90^\circ) = -\sin \varphi_{21}$. В матрицу T_{21} заносят $-\sin \varphi_{21}$.

Порядок работы

1. По табл. 7.1 срисуйте схему манипулятора вместе с осями.
2. Составьте матрицы и проверьте их по табл. 7.2.
3. Вычислите координаты точки E по программе, приведённой в табл. 7.3.
4. Вычисленные координаты проверьте на модели манипулятора.

Таблица 7.1

Схемы манипуляторов

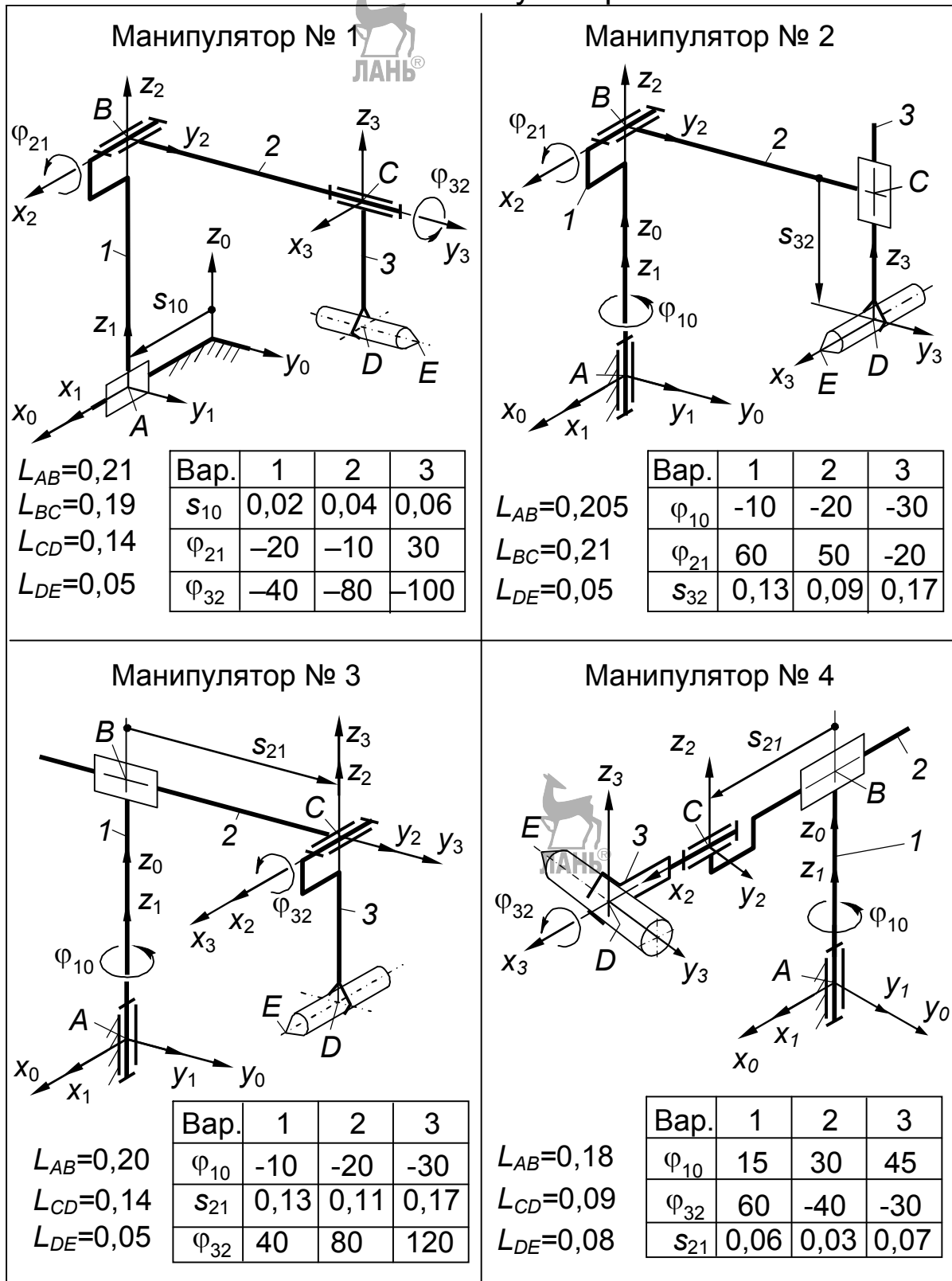


Таблица 7.2

Матрицы преобразования координат

<p>Манипулятор № 1</p> $R_{E_0} = \begin{bmatrix} x_{E_0} \\ y_{E_0} \\ z_{E_0} \\ 1 \end{bmatrix} \quad R_{E_3} = \begin{bmatrix} 0 \\ L_{DE} \\ -L_{CD} \\ 1 \end{bmatrix}$ $T_{10} = \begin{bmatrix} 1 & 0 & 0 & s_{10} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $T_{21} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi_{21} & -\sin \varphi_{21} & 0 \\ 0 & \sin \varphi_{21} & \cos \varphi_{21} & L_{AB} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $T_{32} = \begin{bmatrix} \cos \varphi_{32} & 0 & \sin \varphi_{32} & 0 \\ 0 & 1 & 0 & L_{BC} \\ -\sin \varphi_{32} & 0 & \cos \varphi_{32} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $R_{E_0} = T_{10} T_{21} T_{32} R_{E_3}$	<p>Манипулятор № 2</p> $R_{E_0} = \begin{bmatrix} x_{E_0} \\ y_{E_0} \\ z_{E_0} \\ 1 \end{bmatrix} \quad R_{E_3} = \begin{bmatrix} L_{DE} \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $T_{10} = \begin{bmatrix} \cos \varphi_{10} & -\sin \varphi_{10} & 0 & 0 \\ \sin \varphi_{10} & \cos \varphi_{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $T_{21} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi_{21} & -\sin \varphi_{21} & 0 \\ 0 & \sin \varphi_{21} & \cos \varphi_{21} & L_{AB} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $T_{32} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_{BC} \\ 0 & 0 & 1 & -s_{32} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $R_{E_0} = T_{10} T_{21} T_{32} R_{E_3}$
<p>Манипулятор № 3</p> $R_{E_0} = \begin{bmatrix} x_{E_0} \\ y_{E_0} \\ z_{E_0} \\ 1 \end{bmatrix} \quad R_{E_3} = \begin{bmatrix} L_{DE} \\ 0 \\ -L_{CD} \\ 1 \end{bmatrix}$ $T_{10} = \begin{bmatrix} \cos \varphi_{10} & -\sin \varphi_{10} & 0 & 0 \\ \sin \varphi_{10} & \cos \varphi_{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $T_{21} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & s_{21} \\ 0 & 0 & 1 & L_{AB} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $T_{32} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi_{32} & -\sin \varphi_{32} & 0 \\ 0 & \sin \varphi_{32} & \cos \varphi_{32} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $R_{E_0} = T_{10} T_{21} T_{32} R_{E_3}$	<p>Манипулятор № 4</p> $R_{E_0} = \begin{bmatrix} x_{E_0} \\ y_{E_0} \\ z_{E_0} \\ 1 \end{bmatrix} \quad R_{E_3} = \begin{bmatrix} 0 \\ -L_{DE} \\ 0 \\ 1 \end{bmatrix}$ $T_{10} = \begin{bmatrix} \cos \varphi_{10} & -\sin \varphi_{10} & 0 & 0 \\ \sin \varphi_{10} & \cos \varphi_{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $T_{21} = \begin{bmatrix} 1 & 0 & 0 & s_{21} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_{AB} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $T_{32} = \begin{bmatrix} 1 & 0 & 0 & L_{CD} \\ 0 & \cos \varphi_{32} & -\sin \varphi_{32} & 0 \\ 0 & \sin \varphi_{32} & \cos \varphi_{32} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $R_{E_0} = T_{10} T_{21} T_{32} R_{E_3}$

Программы вычисления координат точки *E**Манипулятор № 1*

```

Lab=0.21; Lbc=0.19; Lcd=0.14; Lde=0.05;
s10=0.02; fi21=-20/180*pi; fi32=-40/180*pi;
Re3=[0 Lde -Lcd 1]';
T10=[1 0 0 s10
      0 1 0 0
      0 0 1 0
      0 0 0 1];
T21=[1 0 0 0
      0 cos(fi21) -sin(fi21) 0
      0 sin(fi21) cos(fi21) Lab
      0 0 0 1];
T32=[cos(fi32) 0 sin(fi32) 0
      0 1 0 Lbc
      -sin(fi32) 0 cos(fi32) 0
      0 0 0 1];
Re0=T10*T21*T32*Re3

```

Манипулятор № 2

```

Lab=0.205; Lbc=0.21; Lde=0.05;
fi10=-10/180*pi; fi21=60/180*pi; s32=0.13;
Re3=[Lde 0 0 1]';
T10=[cos(fi10) -sin(fi10) 0 0
      sin(fi10) cos(fi10) 0 0
      0 0 1 0
      0 0 0 1];
T21=[1 0 0 0
      0 cos(fi21) -sin(fi21) 0
      0 sin(fi21) cos(fi21) Lab
      0 0 0 1];
T32=[1 0 0 0
      0 1 0 Lbc
      0 0 1 -s32
      0 0 0 1];
Re0=T10*T21*T32*Re3

```

Таблица 7.3 (продолжение)

Манипулятор № 3

```

Lab=0.2; Lcd=0.14; Lde=0.05;
fi10=-10/180*pi; s21=0.13; fi32=40/180*pi;
Re3=[Lde 0 -Lcd 1]';
T10=[cos(fi10) -sin(fi10) 0 0
      sin(fi10)  cos(fi10) 0 0
      0           0       1 0
      0           0       0 1];
T21=[1 0 0 0
      0 1 0 s21
      0 0 1 Lab
      0 0 0 1];
T32=[1 0 0 0
      0 cos(fi32) -sin(fi32) 0
      0 sin(fi32)  cos(fi32) 0
      0 0 0 1];
Re0=T10*T21*T32*Re3

```

Манипулятор № 4

```

Lab=0.18; Lcd=0.09; Lde=0.08;
fi10=-10/180*pi; s21=0.1; fi32=30/180*pi;
Re3=[0 -Lde 0 1]';
T10=[cos(fi10) -sin(fi10) 0 0
      sin(fi10)  cos(fi10) 0 0
      0           0       1 0
      0           0       0 1];
T21=[1 0 0 s21
      0 1 0 0
      0 0 1 Lab
      0 0 0 1];
T32=[1 0 0 Lcd
      0 cos(fi32) -sin(fi32) 0
      0 sin(fi32)  cos(fi32) 0
      0 0 0 1];
Re0=T10*T21*T32*Re3

```

Занятие 8. Кинематический анализ кардана

Цель занятия – практическое освоение кинематического анализа механизмов методом преобразования координат на примере замкнутой цепи звеньев.

С этой целью требуется: вывести формулы и построить график функции положения карданного шарнира, вычисления и график сделать на компьютере; компьютерный график сопоставить с экспериментальным, полученным на лабораторной работе «Экспериментальная кинематика кардана».

Вывод формул

На лабораторной работе экспериментально устанавливалась зависимость угла φ_{31} от угла φ_{10} при заданном угле φ_{40} (рис. 8.1, а).

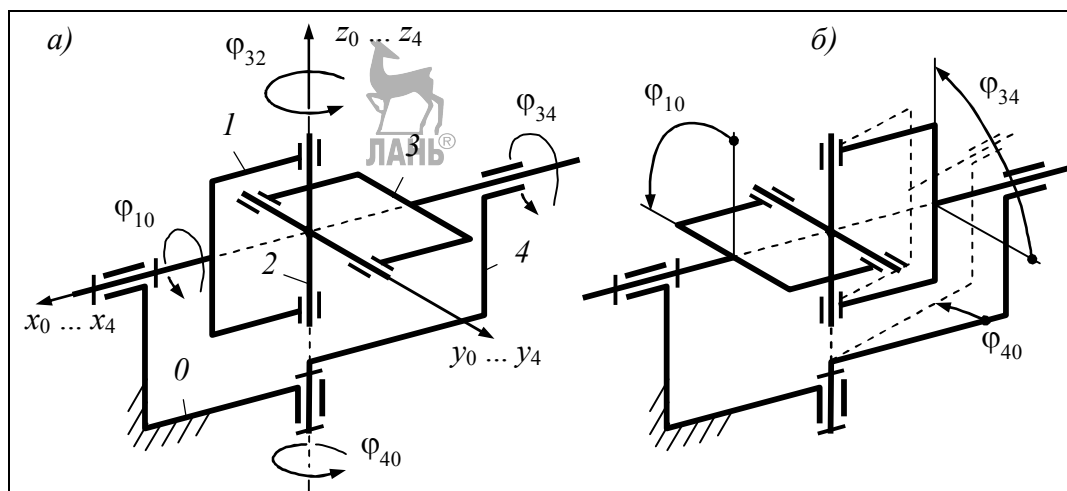


Рис. 8.1

На данном занятии нужно получить эту же зависимость, но аналитически. Как и в случае незамкнутой цепи звеньев (Занятие 7), для решения задачи с каждым звеном механизма связывают систему координат. В карданном шарнире выгодно разместить начала всех систем в центре крестовины 2. При этом в исходном положении, показанном на рисунке, одноимённые оси координат совпадают.

На звене, удалённом от стойки по возможности одинаково, например на крестовине, выбирают произвольную безымянную точку. Координаты этой точки последовательно преобразуют из системы 2 в систему 0 двумя путями: один раз, идя через звено 1, другой – через звено 3. При этом получают следующие уравнения преобразования:

$$R_0 = T_{10} T_{21} R_2 \text{ и } R_0 = T_{40} T_{34} T_{23} R_2.$$

Из равенства левых частей уравнений вытекает равенство правых, откуда, в свою очередь, следует:

$$T_{10} T_{21} = T_{40} T_{34} T_{23}. \quad (8.1)$$

При совпадающих началах всех систем координат элементы a , b , c матриц преобразования равны нулю, четвёртая строка в них становится ненужной и матрицы принимают следующий вид:

$$\begin{array}{c}
 T_{10} \\
 \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & \cos \varphi_{10} & -\sin \varphi_{10} \\ \hline 0 & \sin \varphi_{10} & \cos \varphi_{10} \\ \hline \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 T_{21} \\
 \begin{array}{|c|c|c|} \hline \cos \varphi_{21} & -\sin \varphi_{21} & 0 \\ \hline \sin \varphi_{21} & \cos \varphi_{21} & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array}
 \end{array}$$

$$\begin{array}{c}
 T_{40} \\
 \begin{array}{|c|c|c|} \hline \cos \varphi_{40} & -\sin \varphi_{40} & 0 \\ \hline \sin \varphi_{40} & \cos \varphi_{40} & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 T_{34} \\
 \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & \cos \varphi_{34} & -\sin \varphi_{34} \\ \hline 0 & \sin \varphi_{34} & \cos \varphi_{34} \\ \hline \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 T_{23} \\
 \begin{array}{|c|c|c|} \hline \cos \varphi_{23} & 0 & \sin \varphi_{23} \\ \hline 0 & 1 & 0 \\ \hline -\sin \varphi_{23} & 0 & \cos \varphi_{23} \\ \hline \end{array}
 \end{array}$$

После перемножения в соответствии с (8.1) получают:

$$\begin{array}{|c|c|c|} \hline T_{10} T_{21} \\ \hline \cos \varphi_{21} & -\sin \varphi_{21} & 0 \\ \hline \cos \varphi_{10} \cdot \cos \varphi_{21} & \cos \varphi_{10} \cdot (-\sin \varphi_{21}) & -\sin \varphi_{10} \\ \hline \sin \varphi_{10} \cdot \cos \varphi_{21} & \sin \varphi_{10} \cdot (-\sin \varphi_{21}) & \cos \varphi_{10} \\ \hline \end{array}
 =
 \begin{array}{|c|c|c|} \hline T_{40} T_{34} T_{23} \\ \hline \cos \varphi_{40} \cdot \cos \varphi_{23} - \sin \varphi_{40} \cdot \sin \varphi_{34} \cdot \sin \varphi_{23} & -\sin \varphi_{40} \cdot \cos \varphi_{34} & \cos \varphi_{40} \cdot \sin \varphi_{23} + \sin \varphi_{40} \cdot \sin \varphi_{34} \cdot \cos \varphi_{23} \\ \hline \sin \varphi_{40} \cdot \cos \varphi_{23} + \cos \varphi_{40} \cdot \sin \varphi_{34} \cdot \sin \varphi_{23} & \cos \varphi_{40} \cdot \cos \varphi_{34} & \sin \varphi_{40} \cdot \sin \varphi_{23} - \cos \varphi_{40} \cdot \sin \varphi_{34} \cdot \cos \varphi_{23} \\ \hline -\cos \varphi_{34} \cdot \sin \varphi_{23} & \sin \varphi_{34} & \cos \varphi_{34} \cdot \cos \varphi_{23} \\ \hline \end{array}$$

Приравнивая одноимённые ячейки матриц, получают девять уравнений. По условию задачи было бы достаточно одного, содержащего, во-первых, искомый угол φ_{34} , во-вторых, задаваемые углы φ_{40} , φ_{10} . Перебор всех возможных уравнений показывает, что такого нет. Есть два уравнения (см. выделенные элементы матриц), содержащие нужные углы и один ненужный – φ_{21} :

$$\begin{aligned}
 \sin \varphi_{10} \cos \varphi_{21} &= \sin \varphi_{34}, \\
 \cos \varphi_{10} \cos \varphi_{21} &= \cos \varphi_{40} \cos \varphi_{34}.
 \end{aligned}$$

Поделив первое уравнение на второе, избавляются от φ_{21} и получают: $\operatorname{tg} \varphi_{10} \cos \varphi_{40} = \operatorname{tg} \varphi_{34}$. Отсюда

$$\varphi_{34} = \operatorname{arctg} (\operatorname{tg} \varphi_{10} \cos \varphi_{40}). \quad (8.2)$$



Формула (8.2) даёт правильный результат при угле φ_{10} , лежащем в диапазоне от $-\pi/2$ до $+\pi/2$, т. е. для полуоборота звена I . Этого диапазона достаточно для анализа, т. к. на втором полуобороте закон изменения угла φ_{34} повторяется, это видно по лабораторной работе. Угол звена 3 относительно I (угол забега-ния – отставания)

$$\varphi_{31} = \varphi_{34} - \varphi_{10}. \quad (8.3)$$

Именно для этого угла требуется построить график.

Программа вычислений и построения графика $\varphi_{31}(\varphi_{10})$

```
1. fi40=70*pi/180;
2. n=60; h=pi/n;
3. fi10= -pi/2:h:pi/2;
4. fi34(n+1)=pi/2; fi34(1)= -pi/2;
5. fi34(2:n)=atan(tan(fi10(2:n))*cos(fi40));
6. fi31=fi34 -fi10;
7. fi10G=fi10/pi*180; fi31G=fi 31/pi*180;
8. plot(fi10G,fi31G), grid
9. xlabel('fi10, degree', 'fontSize', 12)
10. ylabel(' fi31, degree', 'fontSize', 12)
11. fi31Gmax=max(fi31G)
```

Комментарии к программе

В строке 1 задаётся угол φ_{40} . Он должен быть таким, как в лабораторной работе «Экспериментальная кинематика кардана». В данном примере $\varphi_{40} = 70^\circ$.

В строке 2 задаётся число интервалов n , на которые будет разбит угол φ_{10} . В этой же строке вычисляется шаг h изменения угла φ_{10} .

В строке 3 задаётся массив значений угла φ_{10} .

В строке 4 задаётся значение последнего и первого элементов массива φ_{34} . Число элементов в массиве на единицу больше числа интервалов, поэтому последний элемент имеет номер $n+1$.

Значения крайних элементов задаются сразу потому, что при $\varphi_{10} = \pm \pi/2$ $\operatorname{tg} \varphi_{10} = \pm \infty$ и вычисление по формуле (8.2) невозможно. Задать значения крайних элементов помогает рис. 8.1, б, где звено I повернуто на $+\pi/2$.

По рисунку видно, что звено 3 повернулось на такой же угол. Поворот звена 4 в этом особом положении не влияет на угол φ_{34} . То же самое будет и при повороте звена 1 на $-\pi/2$.

В строке 5 вычисляются значения всех прочих элементов массива φ_{34} . Это делается по формуле (8.2).

В строке 6 вычисляются значения всех элементов массива φ_{31} .

В строке 7 углы φ_{10} и φ_{31} переводятся в градусы.

В строке 8 программируется построение графика. График снабжается сеткой, на это указывает слово **grid**.

В строках 9,10 команды **xlabel** и **ylabel** создают подписи под осями графика. Аргумент **fontSize,12** указывает, что размер шрифта в подписи равен 12 пунктам.

В строке 11 отыскивается максимальный элемент массива φ_{31} . Если всё правильно, то в окно команд будет выведено: $\varphi_{31\max} = 29,3577^\circ$; график будет иметь вид, показанный на рис. 8.2.

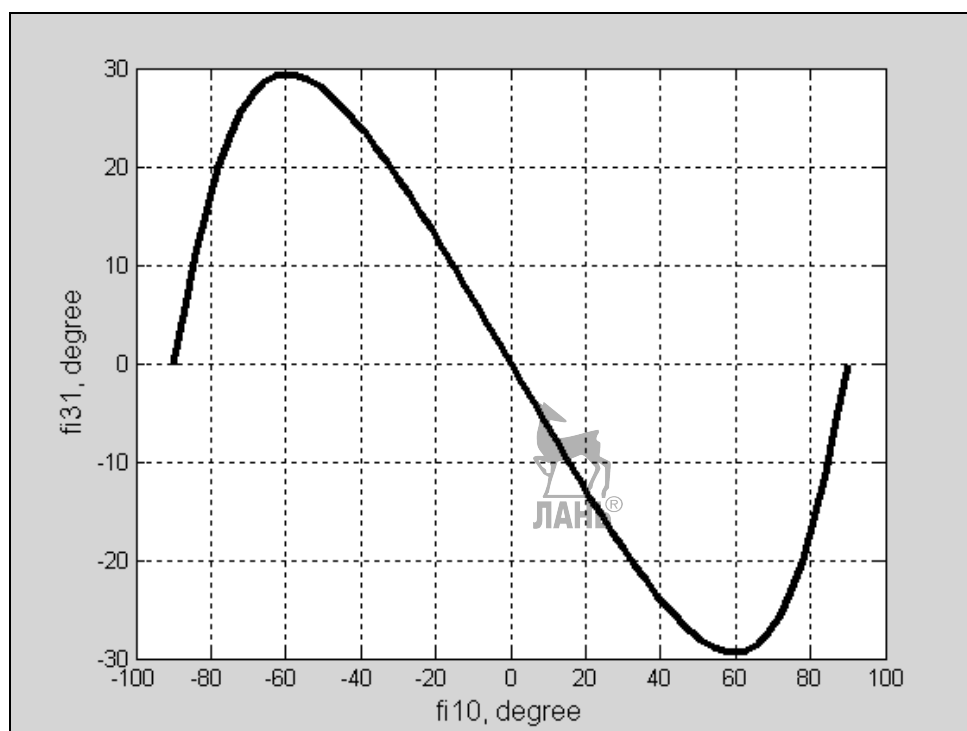


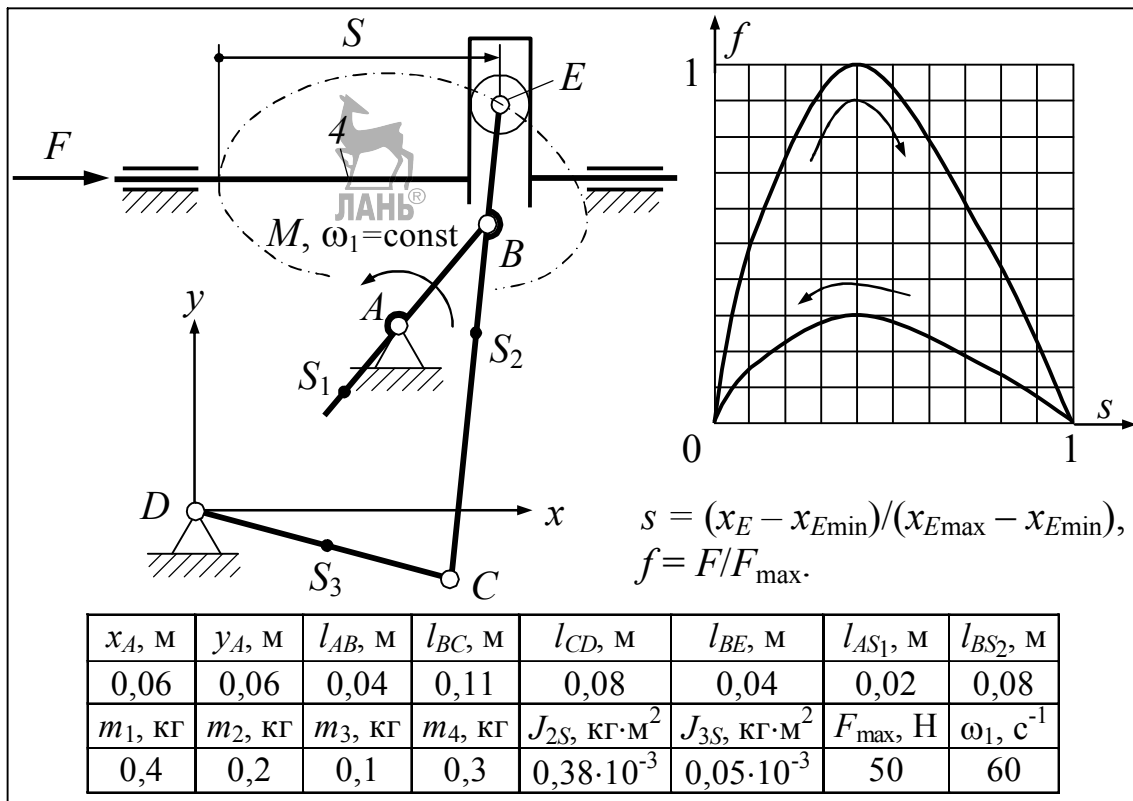
Рис. 8.2

Порядок работы

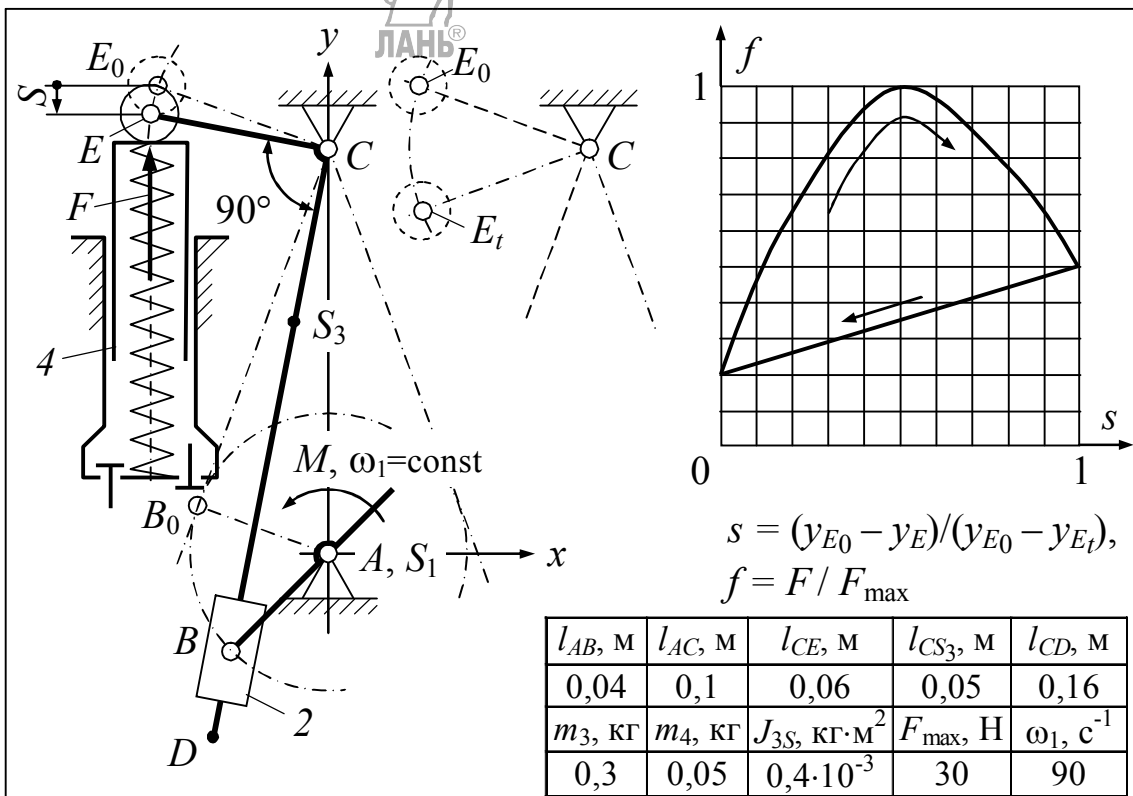
1. Введите программу, подставив в неё своё значение φ_{40} .
2. Сопоставьте компьютерный и экспериментальный графики качественно и количественно (ориентируясь на $\varphi_{31\max}$).
3. Предъявите результат преподавателю.

Задания на силовой расчёт

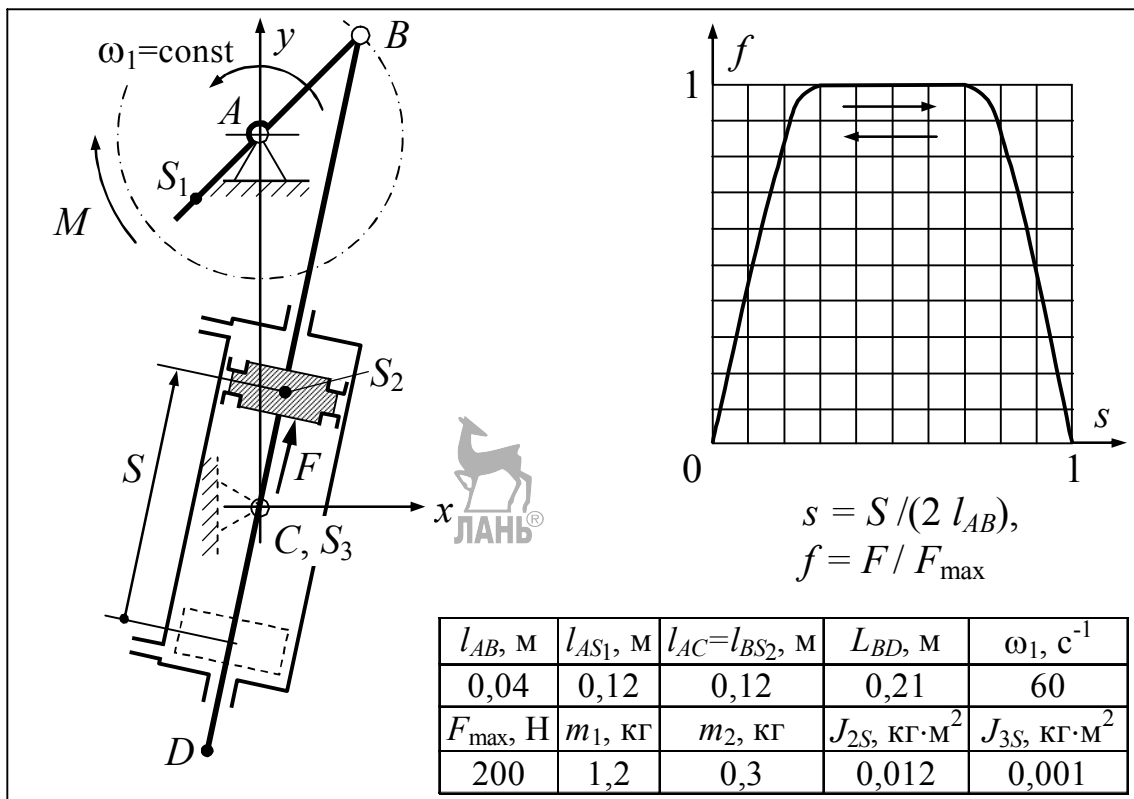
Задание 1



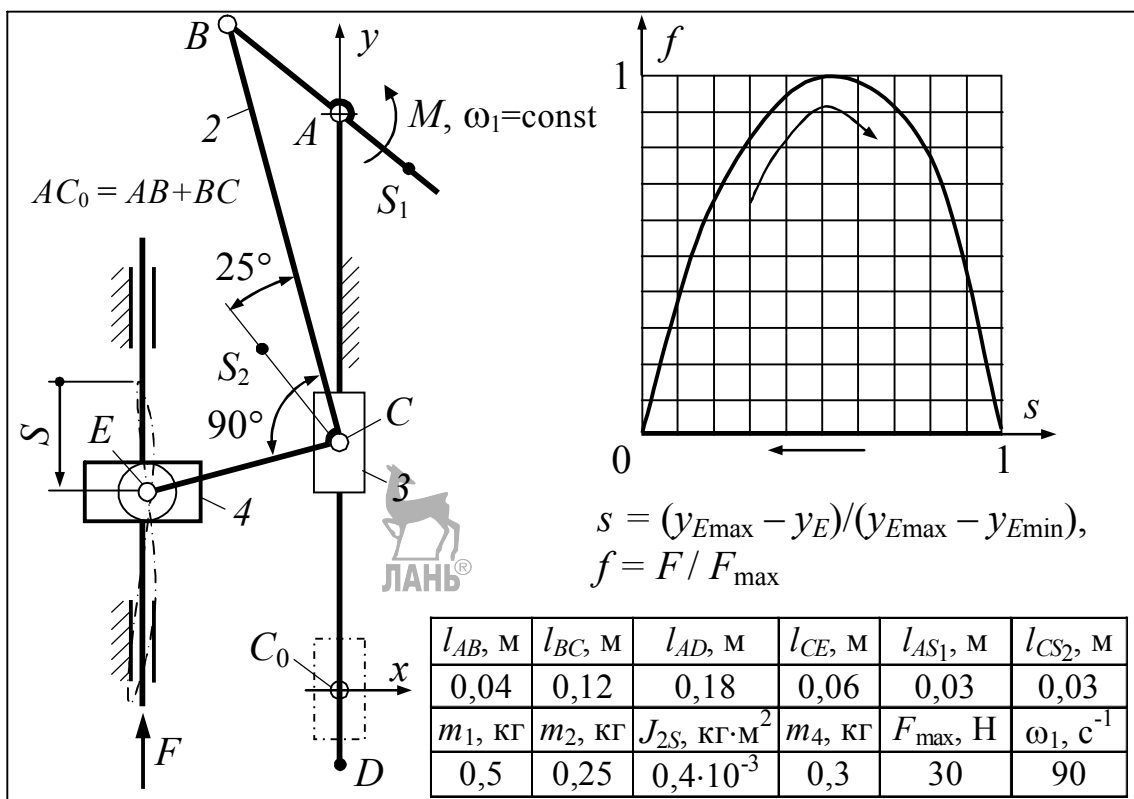
Задание 2



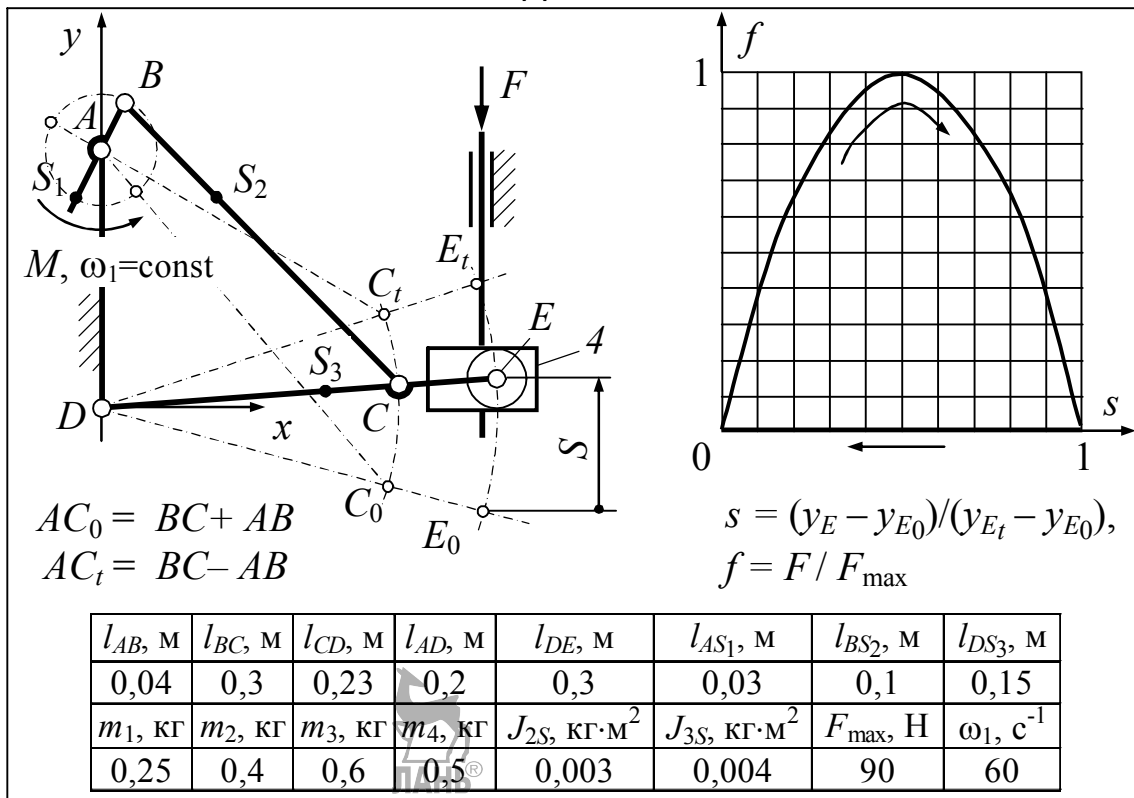
Задание 3



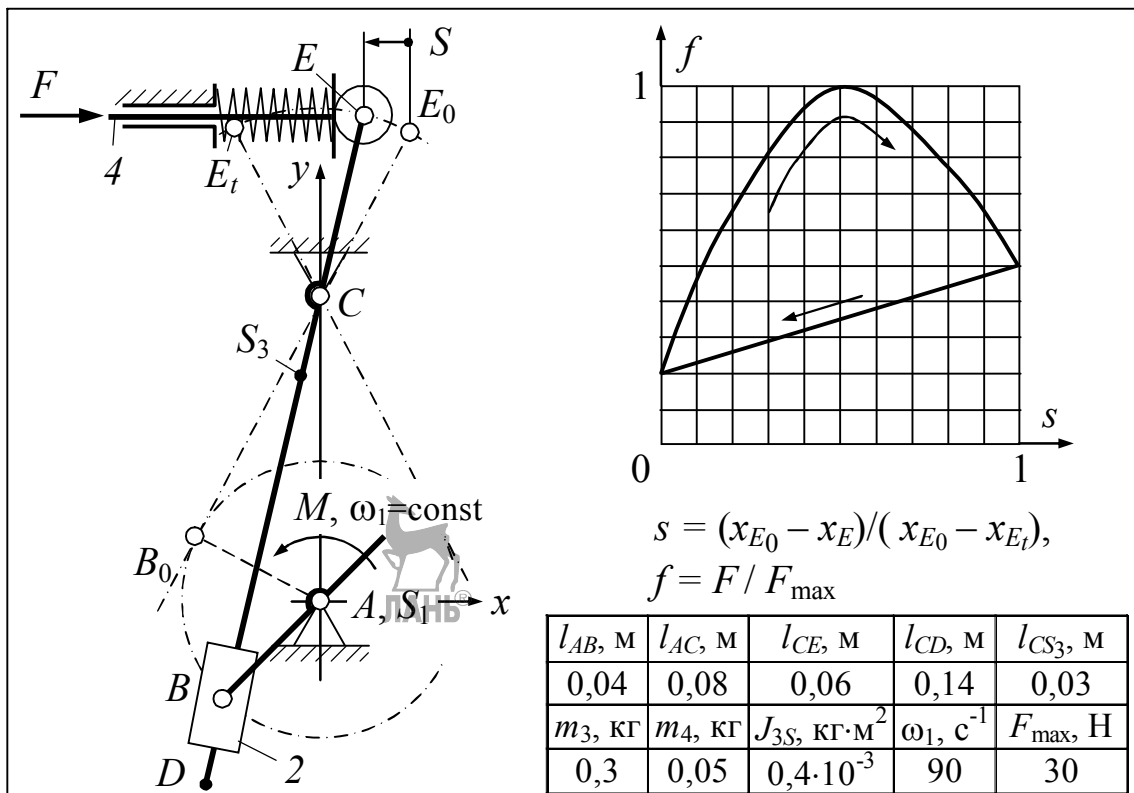
Задание 4



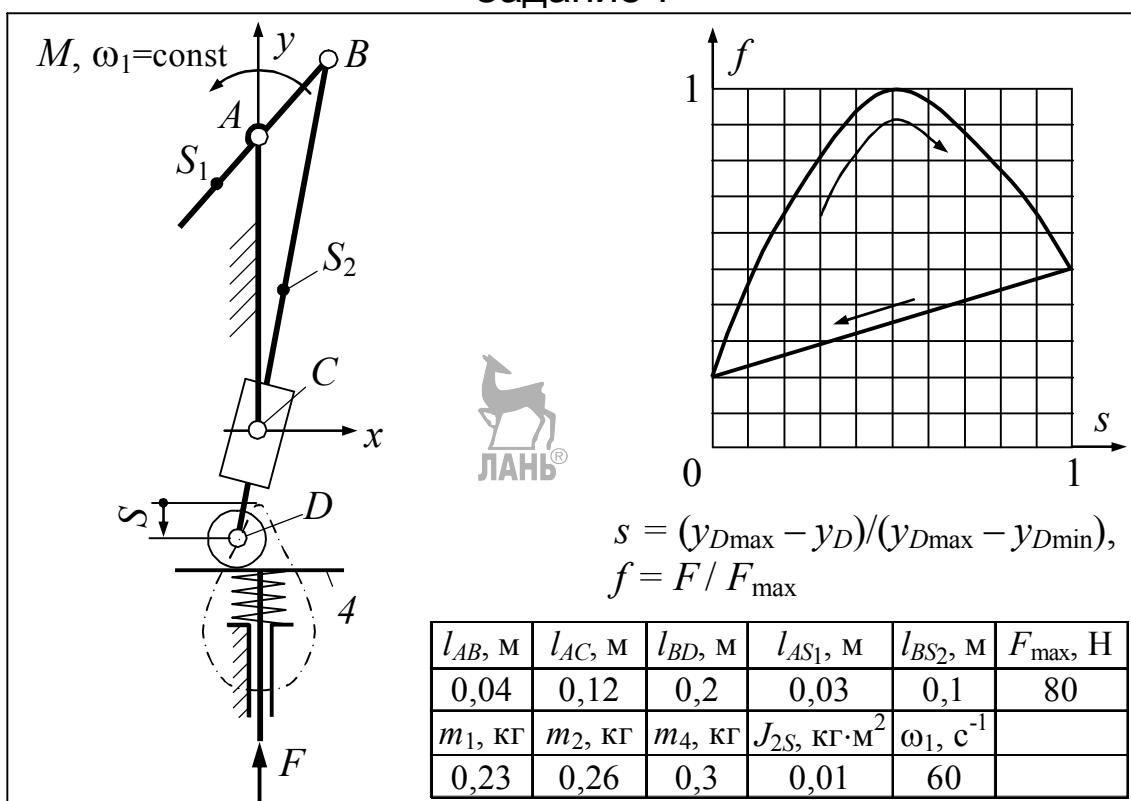
Задание 5



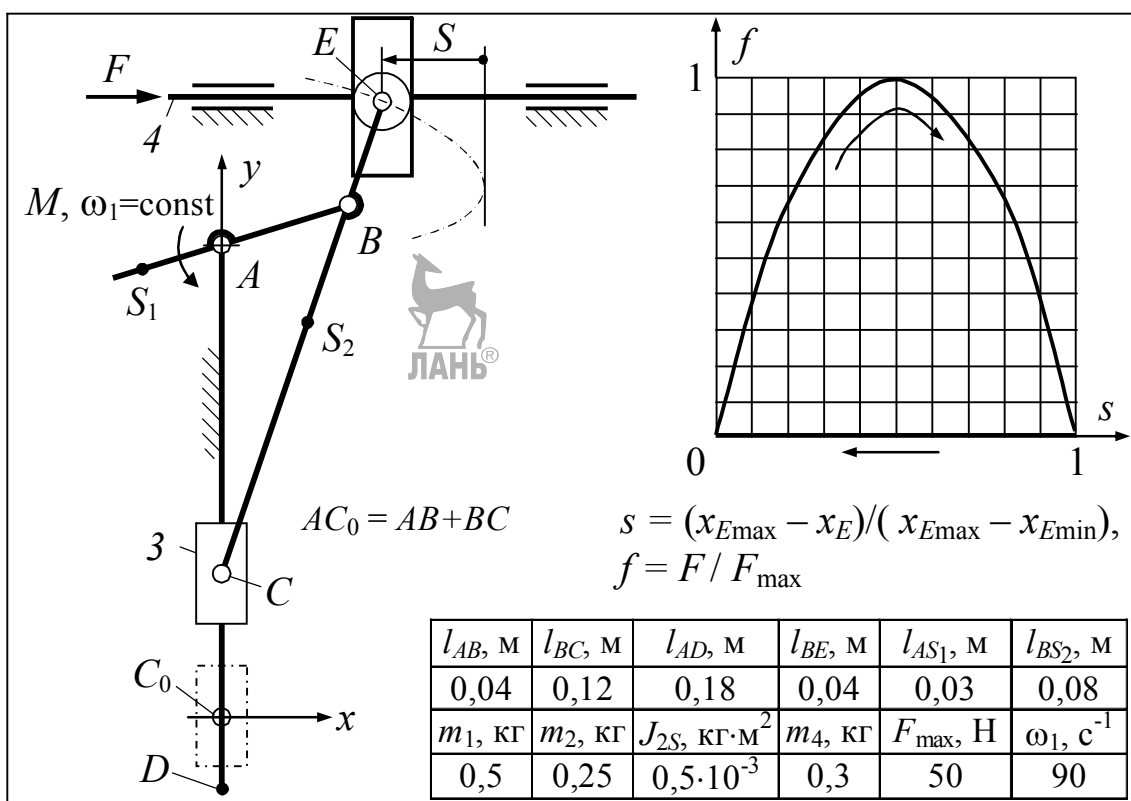
Задание 6



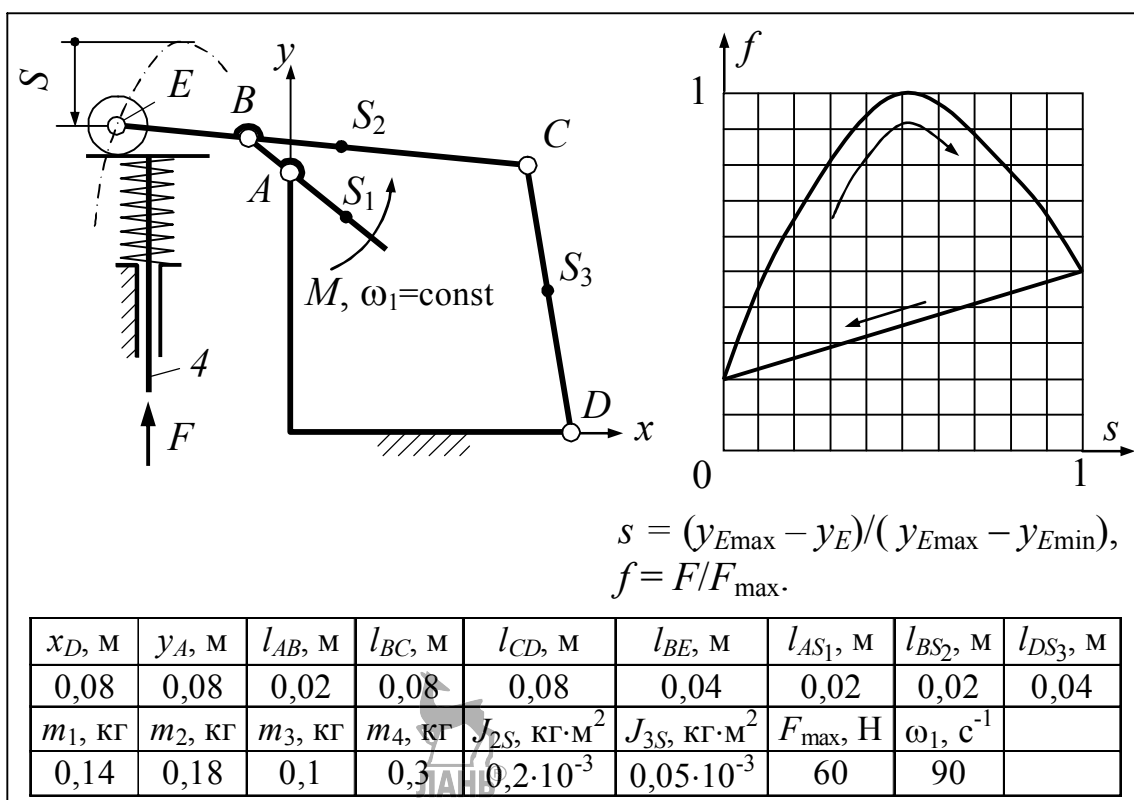
Задание 7



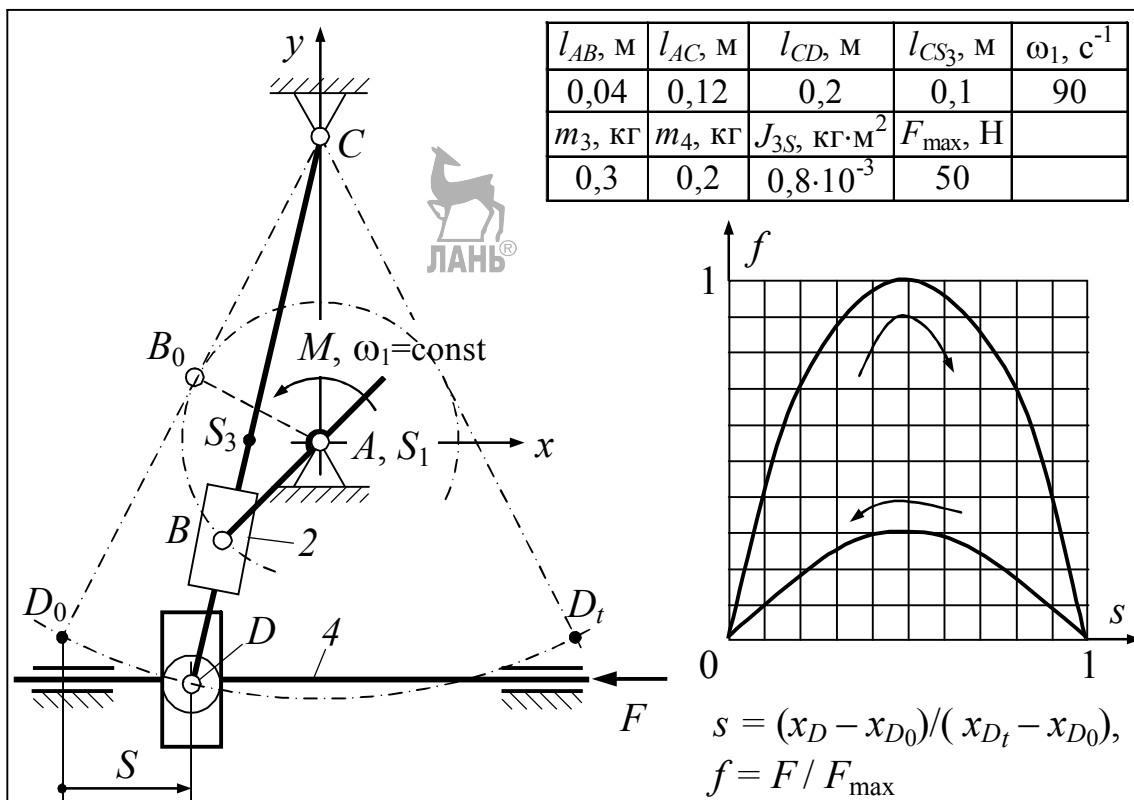
Задание 8



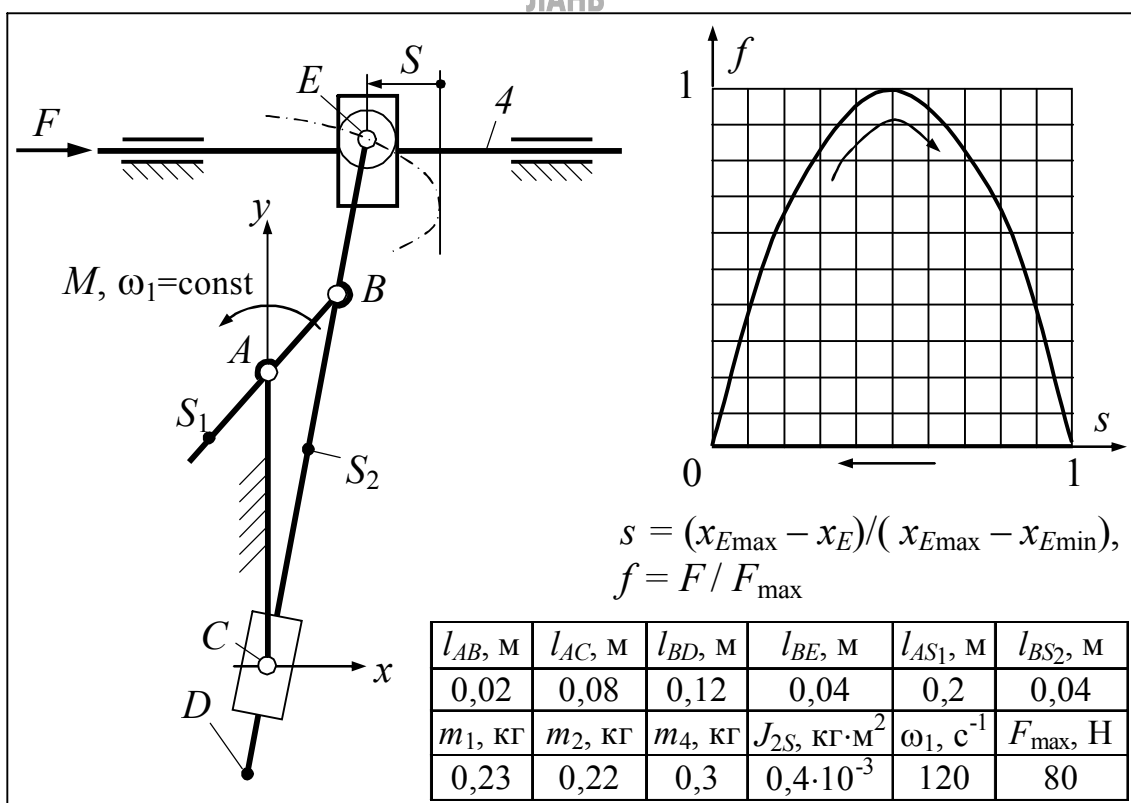
Задание 9



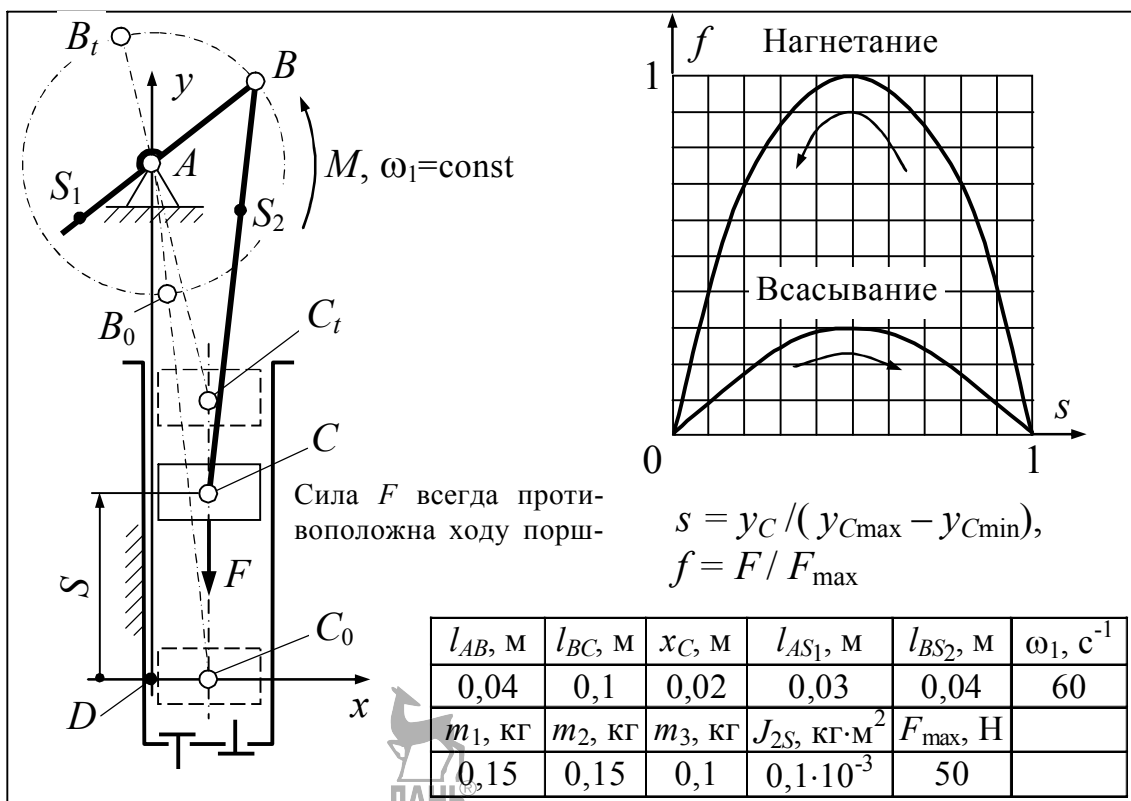
Задание 10



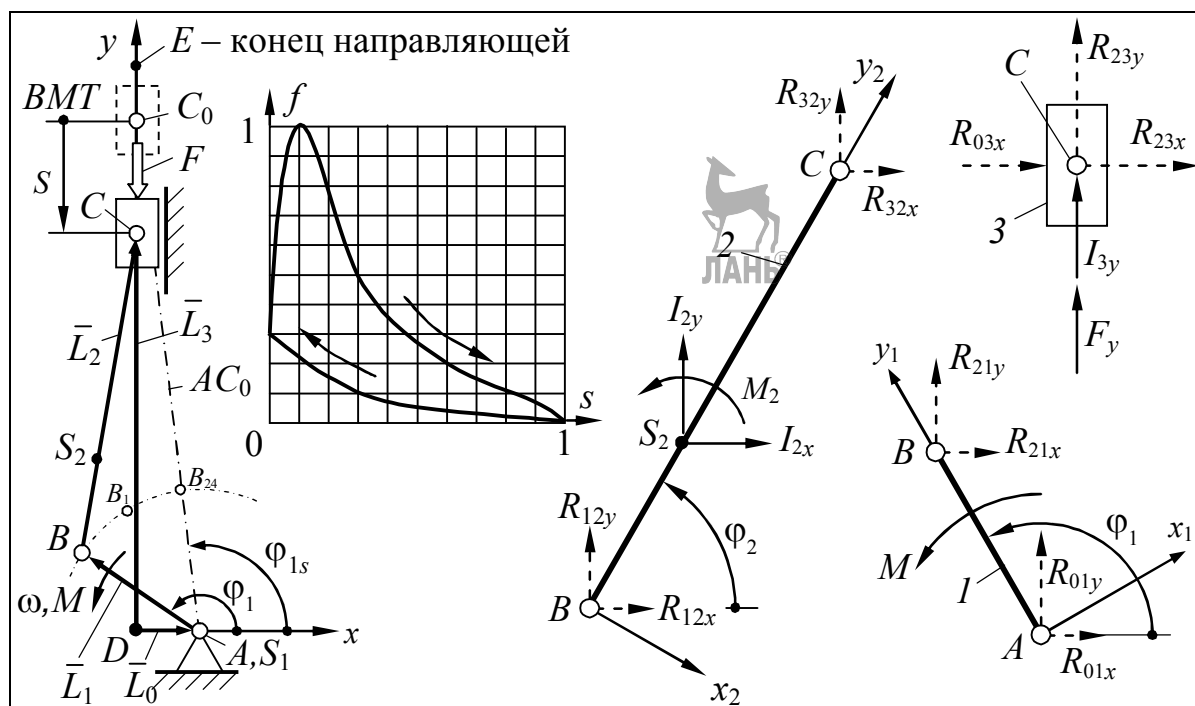
Задание 11



Задание 12



Силовой расчёт двухтактного ДВС



Механизм, индикаторная диаграмма и силы, действующие на звенья механизма

№ зв.	Вид	№	Уравнение
3	$\Sigma X=0$	1	$R_{23x} + R_{03x} = 0$
	$\Sigma Y=0$	2	$R_{23y} + F_y + I_{3y} = 0$
2	$\Sigma X=0$	3	$R_{32x} + R_{12x} + I_{2x} = 0$
	$\Sigma Y=0$	4	$R_{32y} + R_{12y} + I_{2y} = 0$
	$\Sigma M_B=0$	5	$R_{32y}(x_c - x_b) - R_{32x}(y_c - y_b) + I_{2y}(x_{S_2} - x_B) - I_{2x}(y_{S_2} - y_B) + M_2 = 0$
1	$\Sigma X=0$	6	$R_{21x} + R_{01x} = 0$
	$\Sigma Y=0$	7	$R_{21y} + R_{01y} = 0$
	$\Sigma M_A=0$	8	$R_{21y}(x_B - x_A) - R_{21x} y_B + M = 0$

Обозначения

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
R_{01x}	R_{01y}	R_{12x}	R_{12y}	R_{23x}	R_{23y}	R_{03x}	M



$$A =$$

0	0	0	0	1	0	1	0
0	0	0	0	0	1	0	0
0	0	1	0	-1	0	0	0
0	0	0	1	0	-1	0	0
0	0	0	0	$y_c - y_b$	$-x_c + x_b$	0	0
1	0	-1	0	0	0	0	0
0	1	0	-1	0	0	0	0
0	0	y_b	$x_a - x_b$	0	0	0	1

$$B =$$

0
$F - I_{3y}$
$-I_{2x}$
$-I_{2y}$
$-I_{2y}(x_{S2} - x_B) + I_{2x}(y_{S2} - y_B) - M_2$
0
0
0

A – матрица коэффициентов при $x_1 \dots x_8$, B – матрица свободных членов уравнений равновесия. Решение уравнений равновесия в матричной форме: $x = A \setminus B$.

Программа

Файл-функция **extreme**

```

1 function [ycMax , ycMin , fi1s]=extreme (n,L0,L1,L2)
2 h=2*pi/n; fi1=h:h:2*pi;
3 fi2=acos((-L1*cos(fi1)-L0)/L2);
4 yc=L1*sin(fi1)+L2*sin(fi2);
5 ycMax= max(yc); ycMin=min(yc);
6 iMax=find(yc==ycMax); % Индекс максимального значения yc
7 fi1s=fi1(iMax); % Стартовое значение координаты fi1

```

Файл-сценарий (основная программа)

```

1 L0=0.015; L1=0.04; L2=3.5*L1; lbs2=L2/3; % Размеры
2 m2=0.8; m3=0.6; J2=m2*L2^2/12; % Массы и момент инерции
3 Fmax=12560; % Максимальная внешняя сила
4 omega=300; % Угловая скорость кривошипа
5 r=0.06*L1; % Радиус окружности, изображающей шарнир
6 Ks=4*r; Ls=2*Ks; % Стороны ползуна
7 %----- -Вычисление координат -----
8 n=360; % Число положений механизма при определении ycMax, ycMin
9 [ycMax, ycMin, fi1s]=extreme(n,L0,L1,L2); % Экстремальные
значения координат точки C
10 n=24; % Число положений, подлежащих силовому расчёту
11 h=2*pi/n; % Шаг кривошипа (15 град.)
12 fi1=fi1s+(h:h:2*pi);

```



```

13 fi2=acos(( -L1*cos(fi1) -L0)/L2);
14 z=zeros(1,n);
15 xa=L0*ones(1,n); ya=z;
16 xb=L0+L1*cos(fi1); yb=L1*sin(fi1);
17 xs2=xb+Lbs2*cos(fi2); ys2=yb+Lbs2*sin(fi2);
18 xc=z; yc=yb+L2*sin(fi2);
19 xd=z; yd=z;
20 xe=z; ye=(ycMax+Ls)* ones(1,n);
21 %----- ---- Построение графика num(s) -----
22 Smax=ycMax -ycMin; % Ход ползуна
23 s=(ycMax -yc)/Smax; % Относительная координата ползуна
24 num=1:n; % Номер положения механизма
25 figure
26 plot(s, num, 'o-'), grid % График num(s)
27 xlabel('s'), ylabel('num')
28 % ---- Табуляция силы f-----
29 f=[0.58 0.98 0.87 0.51 0.33 0.23 0.16 0.11 0.07 0.05 0.02 0 ...
30 0 0 0 0.005 0.02 0.04 0.06 0.09 0.17 0.23 0.28 0.3];
31 % f определяют в документе Word по графикам f(s) и num(s)
32 F=f*Fmax; % Абсолютная сила
33 figure
34 plot([s(n) s],[f(n) f],'o-') % Фактическая диаграмма f(s)
35 axis equal, grid, title('Diagram f(s)')
36 % ----- --- Планы положений механизма -----
37 figure
38 for k=2:2:n % Построение каждого 2-го положения механизма
39 line([xa(k),xd(k),xe],[ya(k),yd(k),ye]) % Стойка
40 polzun(Ks,Ls,xc(k),yc(k),pi/2) % Ползун
41 line([xa(k) xb(k) xc(k)],[ya(k) yb(k) yc(k)]) % Крив ошип
и шат ун
42 circle(r,xa(k),ya(k)), circle(r,xb(k),yb(k))
43 circle(r,xc(k),yc(k)) % Шарниры А, В, С
44 plot(xs2(k), ys2(k), '.k') % Изображение центра масс точкой
45 hold on % Удержание нарисованного от стирания
46 end, axis equal
47 %----- Дифференцирование -----
48 dfi=0.02; % Приращение угла fi1 (1,146 градуса)
49 fi1L=fi1 -dfi; fi1R=fi1+dfi; % Левые L и правые R знач. fi1
50 fi2L=acos(( -L1*cos(fi1L) -L0)/L2);
51 fi2R=acos(( -L1*cos(fi1R) -L0)/L2);
52 xbL=L0+L1*cos(fi1L); xbR=L0+L1*cos(fi1R);
53 ybL=L1*sin(fi1L); ybR=L1*sin(fi1R);
54 ycL=ybL+L2*sin(fi2L); ycR=ybR+L2*sin(fi2R);
55 xs2L=xbL+Lbs2*cos(fi2L); xs2R=xbR+Lbs2*cos(fi2R);

```

```

56 ys2L=ybL+Lbs2*sin(fi2L);      ys2R=ybR+Lbs2*sin(fi2R);
57 % ----- Вторые производные от координат -----
58 xs2w=(xs2R+xs2L -2*xs2)/dfi^2;
59 ys2w=(ys2R+ys2L -2*ys2)/dfi^2;
60 fi2w=(fi2R+fi2L -2*fi2)/dfi^2;
61 ycw=(ycR+ycL -2*yc)/dfi^2;
62 clear fi1L fi1R xbL xbR ybL ybR % Удаление отработавших величин
63 clear ycL ycR fi2L fi2R xs2L xs2R ys2L ys2R
64 % ----- Силы инерции -----
65 I2x= -m2*omega^2*xs2w;      I2y= -m2*omega^2*ys2w;
66 M2= -J2*omega^2*fi2w;      I3y= -m3*omega^2*ycw;
67 % ----- Формирование матриц А и В -----
68 for i=1:n
69     A=[0, 0, 0, 0, 1, 0, 1, 0
70         0, 0, 0, 0, 0, 0, 1, 0, 0
71         0, 0, 1, 0, -1, 0, 0, 0, 0
72         0, 0, 0, 1, 0, -1, 0, 0, 0
73         0, 0, 0, 0, yc(i) -yb(i), xb(i) -xc(i), 0, 0
74         1, 0, -1, 0, 0, 0, 0, 0, 0
75         0, 1, 0, -1, 0, 0, 0, 0, 0
76         0, 0, yb(i), xa(i) -xb(i), 0, 0, 0, 1];
77     B=[0;F(i) -I3y(i); -I2x(i); -I2y(i); -I2y(i)*(xs2(i) -xb(i))+ ...
78         I2x(i)*(ys2(i) -yb(i)) -M2(i);0;0;0];
79     x=A \B; % Вычисление корней системы
80 % Заполнение массивов R01x, R01y и т.д. корнями системы
81 R01x(i)=x(1);      R01y(i)=x(2);      R12x(i)=x(3);      R12y(i)=x(4);
82 R23x(i)=x(5);      R23y(i)=x(6);      R03x(i)=x(7);      M(i)=x(8);
83 end
84 % Преобразование R12, R32, I2 из системы ху в систему х2у2
85 bet2=fi2 -pi/2; % Угол оси х2 относительно х
86 R12x2=R12x.*cos(bet2)+R12y.*sin(bet2);
87 R12y2= -R12x.*sin(bet2)+R12y.*cos(bet2);
88 R32x= -R23x; R32y= -R23y;
89 R32x2=R32x.*cos(bet2)+R32y.*sin(bet2);
90 R32y2= -R32x.*sin(bet2)+R32y.*cos(bet2);
91 I2x2=I2x.*cos(bet2)+I2y.*sin(bet2);
92 I2y2= -I2x.*sin(bet2)+I2y.*cos(bet2);
93 % ----- План сил звена -----
94 pn=4; % Номер проверяемого положения механизма
95 % Абсциссы xver и ординаты yver вершин плана сил
96 xver(1)=R12x2(pn);      xver(2)=xver(1)+I2x2(pn);
97 xver(3)=xver(2)+R32x2(pn);
98 yver(1)=R12y2(pn);      yver(2)=yver(1)+I2y2(pn);
99 yver(3)=yver(2)+R32y2(pn);

```

```

100 figure
101 plot([0 xver], [0 yver]), axis equal, grid % План сил
102 title(['forceplane' ' ' num2str(pn) ', ' 'link 2'])
103 text(xver(1), yver(1), 'R12') % Обозначение вектора на его конце
104 text(xver(2), yver(2), 'I2')
105 text(xver(3), yver(3), 'R32')
106 % Годографы R12, R32, I2 предварительно (из начала координат)
107 figure
108 plot(R12x 2,R12y2,'o -', R32x2,R32y2,'o -', I2x2,I2y2,'o -')
109 axis equal, grid
110 title('Godografs relatively x2-y2 from one center')
111 ----- % Смещение точек В и С по осям х и у -----
112 Bx= -6000; By= -4000; Cx=6000; Cy=3000; %
113 % ----- --Разнесённые годографы -----
114 figure
115 plot(R12x2+Bx, R12y2+By,'o -',I2x2,I2y2,'o -',...
116 R32x2+Cx, R32y2+Cy,'o -', Bx,By,'ok' ...
117 0,0,'ok', Cx,Cy,'ok')
118 % Обозначения шарниров и центра масс звена 2
119 text (Bx,By,'B'), text (0,0,'S2'), text (Cx,Cy,'C')
120 % Нумерация точек 1 и 24 на годографах
121 text (R12x2(24)+Bx, R12y2(24)+By, '24')
122 text (R12x2(1)+Bx, R12y2(1)+By, '1')
123 text (I2x2(24), I2y2(24), '24')
124 text (I2x2(1), I2y2(1), '1')
125 text (R32x2(24)+Cx, R32y2(24)+Cy, '24')
126 text (R32x2(1)+Cx, R32y2(1)+Cy, '1')
127 grid, legend('R12','I2','R32',0) % 0-авт. выбор места легенды
128 title('Godografs link 2')
129 % ----- Преобразование R21 из системы ху в x1y1 -----
130 bet1=fil -pi/2; % Угол оси x1 относительно х
131 R21x= -R12x; R21y= -R12y;
132 R21x1=R21x.*cos(bet1)+R21y.*sin(bet1);
133 R21y1= -R21x.*sin(bet1)+R21y.*cos(bet1);
134 %----- Годограф R21 в системе x1y1 -----
135 figure
136 plot([R21x1(24) R21y1], [R21y1(24) R21y1], 'o-', 0, 0,'ok')
137 grid
138 text(0,0,'B'), text(R21x1(24), R21y1(24), '24')
139 text(R21x1(1), R21y1(1), '1')
140 title('Godograf R21 relatively x1-y1')
141 % ----- График реакции -----
142 ti=12; % Индекс переменной в конце такта расширения
143 % устанoвлен по разметке хода ползуна

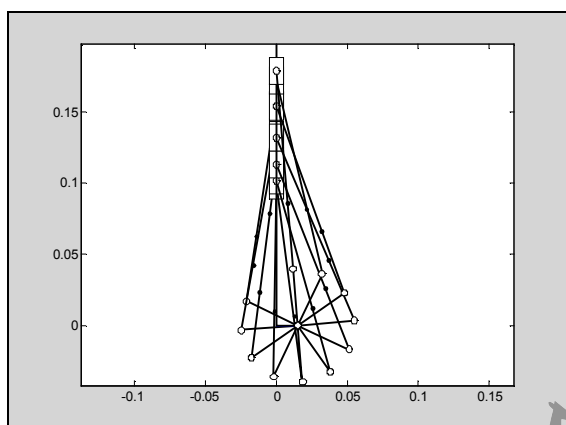
```

```

144 R30x= -R03x;
145 figure
146 plot(R30x(1:ti),s(1:ti),'o - ', R30x(ti:end),s(ti:end),'o - ')
147 grid, axis ij % "Матричное" расположение осей координат
148 % (начало координат - вверху)
149 legend('expansion','compression', 4) % 4 - легенда справа внизу
150 xlabel('R30'), ylabel('s')
151 text(R30x(1),s(1),'1'),text(R30x(12),s(12),'12')
152 text(R30x(24),s(24),'24')
153 figure
154 plot(0:n,[M(n) M],'o-'), grid % График момента M
155 xlabel('num'), ylabel('M')
156 %--- Проверка силового расчёта через работу внешних сил ----
157 AF=polyarea(s,f)*Fmax*Smax % Работа силы F
158 % polyarea - площадь под кривой f(s)
159 Mm=mean(M); % Среднее значение момента M
160 AM=Mm*2*pi % Работа момента M
161 delta=abs((AF+AM)/AF)*100 % Ошибка в процентах
162 N=abs(Mm)*omega % Мощность двигателя

```

Результаты



Планы положений механизма

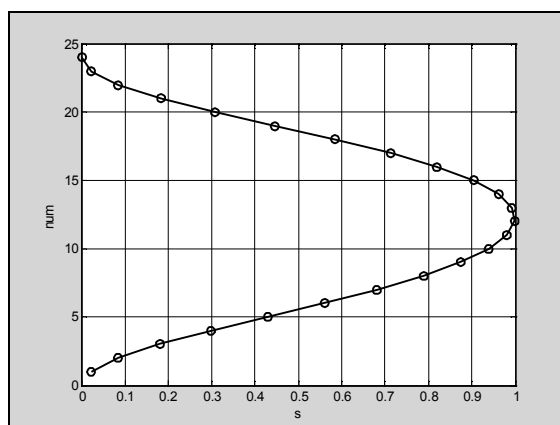
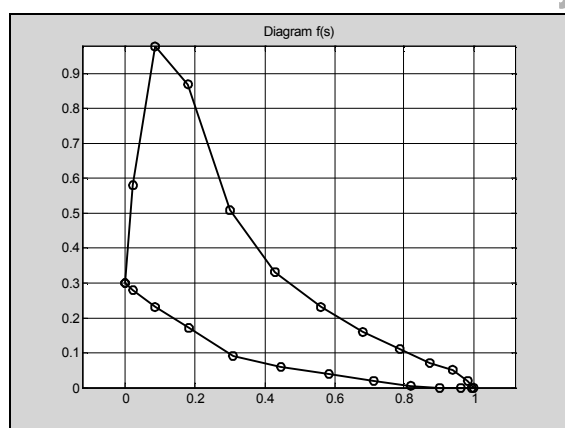
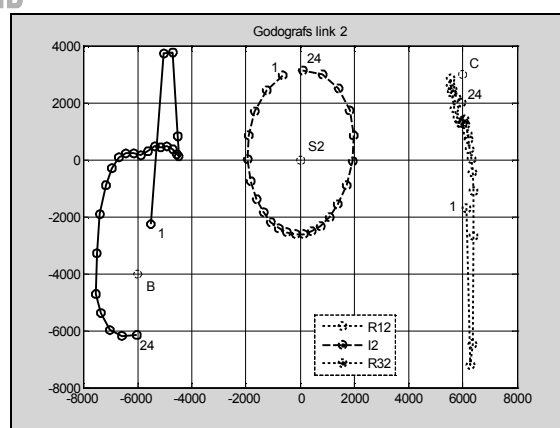


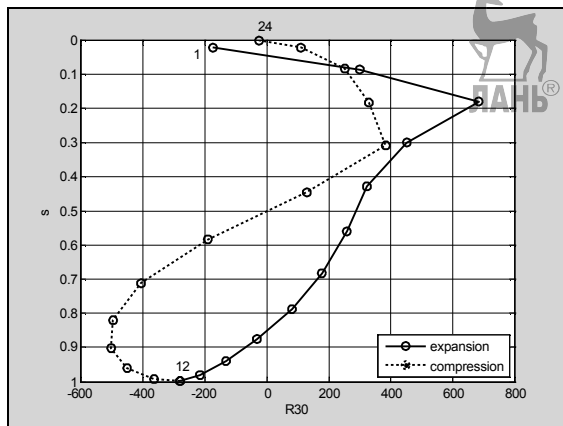
График движения ползуна



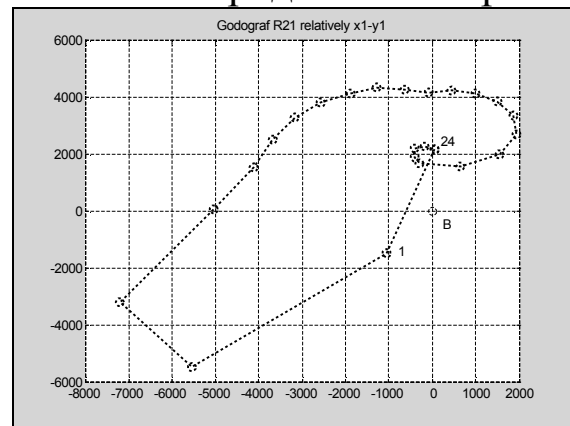
Фактическая внешняя сила



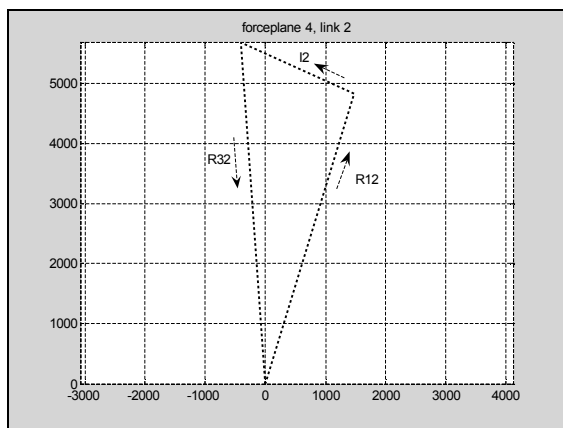
Силы, приложенные к шатуну



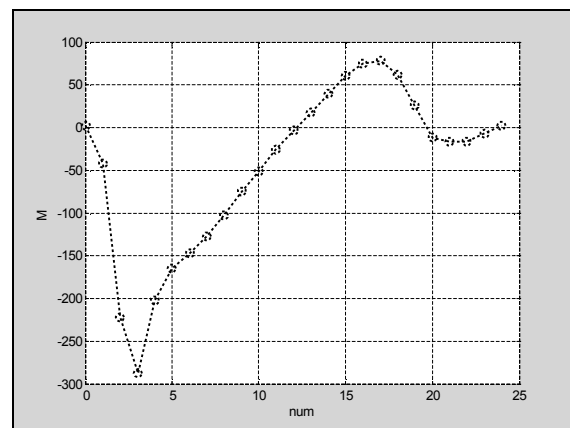
Реакция поршня на цилиндр



Реакция шатуна на кривошип



План сил шатуна



Внешний момент на кривошипе

Результаты, выведенные в окно команд:

1. Мощность $N = 14252,5783$ Вт.
 2. Работа силы F за оборот кривошипа $AF = 293,9636$ Дж.
 3. Работа момента M за тот же период $AM = -298,5053$ Дж.
- Ошибка $\delta = 1,54$ %, что допустимо.



Программы геометрического расчёта зацепления (для выполнения курсового проекта)

1. Расчёт при незаданном межцентровом расстоянии

Введите эту файл-функцию в окно редактора-отладчика и сохраните под именем **ProfAng** .

```
function y=ProfAng (x)
% Вычисление функции  $y = \tan x - x - \text{inv\_al\_w}$ 
% где inv_al_w - глобальная переменная
% Обращение : y=ProfAng(x)
global inv_al_w % Объявление глобальной переменной
y=tan(x) -x-inv_al_w;
```

Введите этот файл-сценарий в окно редактора-отладчика и сохраните под именем **zac** .

```
% Расчёт зацепления при незаданном межцентровом расстоянии и
% Обращение: zac
global inv_al_w ; % Повторное объявление глобальной переменной
al=20/180*pi; % Задание угла профиля исходного контура (ИК)
inv_al=tan(al) -al; % Вычисление инволюты угла профиля ИК
disp('Исходные данные') % Заголовок
% Программирование ввода исходных данных
m=input('m='); z4=input('z4='); z5=input('z5=');
x4=input('x4='); x5=input('x5=');
inv_al_w=2*(x4+x5)*tan(al)/(z4+z5)+inv_al; % Вычисление ин-
% вольуты угла зацепления
al_w=fzero('ProfAng', 0.3); % Вычисление угла зацепления
al_w_g=al_w/pi*180; % Перевод угла зацепления в градусы
aw=m*(z4+z5)*cos(al)/(2*cos(al_w)); % Межцентровое расстояние
% Вычисление радиусов
r4=m*z4/2; r5=m*z5/2; % делительных окружностей
rb4=r4*cos(al); rb5=r5*cos(al); % основных окружностей
rf4=r4+(x4-1.25)*m; rf5=r5+(x5-1.25)*m; % окружностей впадин
ra4=aw-rf5-0.25*m; ra5=aw-rf4-0.25*m; % окружностей вершин
% Вычисление шага по дуге и по хорде
p=pi*m; ph4=2*r4*sin(p/r4/2); ph5=2*r5*sin(p/r5/2);
% Вычисление толщин зубьев
s4=(pi/2+2*x4*tan(al))*m; s5=(pi/2+2*x5*tan(al))*m; % по дуге
```

```

sh4=2*r4*sin(s4/r4/2);    sh5=2*r5*sin(s5/r5/ 2); % по хорде
% Вычисление углов профиля при вершине
al_a4=acos(rb4/ra4);    al_a5=acos(rb5/ra5);
al_a4g=al_a4/pi*180;    al_a5g=al_a5/pi*180 ;
% Вычисление коэффициента перекрытия
epsilon=(z4*(tan(al_a4) -tan(al_w))+z5*(tan(al_a5) -
tan(al_w)))/(2*pi);
% Вывод результатов на дисплей
disp('Результаты') ;
disp(['inv_al_w=',num2str(inv_al_w),' ','al_w=',num2str(al_w_g)]);
disp(['aw=',num2str(aw)]) ;
disp(['r4=',num2str(r4),' ','r5=',num2str(r5)]);
disp(['rb4=',num2str(rb4),' ','rb5=',num2str(rb5)]);
disp(['rf4=',num2str(rf4),' ','rf5=',num2str(rf5)]);
disp(['ra4=',num2str(ra4),' ','ra5=',num2str(ra5)]);
disp(['p=',num2str(p)]);
disp('Шаг по хорде')
disp(['ph4=',num2str(ph4),' ','ph5=',num2str(ph5)]);
disp('Толщина по хорде')
disp(['sh4=',num2str(sh4),' ','sh5=',num2str(sh5)]);
disp(['al_a4=',num2str(al_a4),' ','al_a5=',num2str(al_a5)]);
disp(['epsilon=',num2str(epsilon)]);
M=input('Увеличение M=');
disp('Размеры после увеличения')
disp(['aw=',num2str(aw*M)])
disp(['r4=',num2str(r4*M),' ','r5=',num2str(r5*M)]);
disp(['rb4=',num2str(rb4*M),' ','rb5=',num2str(rb5*M)]);
disp(['rf4=',num2str(rf4*M),' ','rf5=',num2str(rf5*M)]);
disp(['ra4=',num2str(ra4*M),' ','ra5=',num2str(ra5*M)]);
disp(['sh4=',num2str(sh4*M),' ','sh5=',num2str(sh5*M)]);
disp(['ph4=',num2str(ph4*M),' ','ph5=',num2str(ph5*M)]);

```

По блокирующим контурам [5, с. 114 или 6, с. 35] выберите коэффициенты смещения x_4 , x_5 . Запустите программу из окна редактора-отладчика или из окна команд. В первом случае для этого нажмите клавишу **F5**, во втором – введите **zac** ↵ в окно команд. Далее следуйте запросам программы, первым будет: **m=**. В ответ на это введите численное значение модуля и нажмите **Enter**. И так далее.

2. Расчёт при заданном межцентровом расстоянии

```

% Расчёт зацепления при заданном межцентровом расстоянии и
% Обращение: zasa
disp('Исходные данные') % Заголовок
% Программирование ввода исходных данных
m=input('m='); z4=input('z4='); z5=input('z5=');
aw=input('aw=');
% Вычисления
al=20/180*pi; % Перевод угла профиля исходного контура в радианы
al_w=acos(m*(z4+z5)*cos(al)/2/aw); al_wg=al_w/pi*180; % Угол заце-
пления в радианах и градусах
inv_al_w=tan(al_w) -al_w; % Инволюта угла зацепления
inv_al=tan(al) -al; % Инволюта угла профиля исходного контура
x_sum=(z4+z5)*(inv_al_w -inv_al)/2/tan(al); % Сумма коэффи-
циентов смещения
disp('Результаты') % Заголовок
disp(['al_w=',num2str(al_wg)]); % Вывод угла зацепления в градусах
disp(['x_sum=',num2str(x_sum)]); % Вывод суммы коэффициентов смещения
disp('Ввод') % Заголовок
x4=input('x4='); % Ввод x4, выбранного по блокирующему кон-
туру [5, с. 25, 114 или 6, с. 35], исходя из x_sum=...
disp('Результаты') % Заголовок
x5=x_sum -x4; %Вычисление коэффициента смещения x5
disp(['x5=',num2str(x5)]); % Вывод x5
% Вычисление радиусов
r4=m*z4/2; r5=m*z5/2; % делит -х окружностей
rb4=r4*cos(al); rb5=r5*cos(al); % основных окружностей
rf4=r4+(x4 -1.25)*m; rf5=r5+(x5 -1.25)*m; % окружностей впадин
ra4=aw -rf5 -0.25*m; ra5=aw -rf4 -0.25*m; % окружностей вершин
% Вычисление шага по дуге и по хорде
p=pi*m; ph4=2*r4*sin(p/r4/2); ph5=2*r5*sin(p/r5/2);
% Вычисление толщин зубьев
s4=(pi/2+2*x4*tan(al))*m; s5=(pi/2+2*x5*tan(al))*m; % по дуге
sh4=2*r4*sin(s4/r4/2); sh5=2*r5*sin(s5/r5/2); % по хорде
% Вычисление углов профиля при вершине
al_a4=acos(rb4/ra4); al_a5=acos(rb5/ra5);
al_a4g=al_a4/pi*180; al_a5g=al_a5/pi*180;
% Вычисление коэфф-та перекрытия

```



```

epsilon=(z4*(tan(al_a4) -tan(al_w))+z5*(tan(al_a5) - ...
tan(al_w)))/(2*pi);
% Вывод радиусов и прочих параметров зацепления
disp(['r4=', num2str(r4), ' ', 'r5=', num2str(r5)]);
disp(['rb4=', num2str(rb4), ' ', 'rb5=', num2str(rb5)]);
disp(['rf4=', num2str(rf4), ' ', 'rf5=', num2str(rf5)]);
disp(['ra4=', num2str(ra4), ' ', 'ra5=', num2str(ra5)]);
disp(['p=', num2str(p)]);
disp(' Шаг по хорде')
disp(['ph4=', num2str(ph4), ' ', 'ph5=', num2str(ph5)]);
disp('Толщина по хорде')
disp(['sh4=', num2str(sh4), ' ', 'sh5=', num2str(sh5)]);
disp(['al_a4=', num2str(al_a4), ' ', 'al_a5=', num2str(al_a5)]);
disp(['epsilon=', num2str(epsilon)]);
M=input('Увеличение M=');
disp('Размеры после увеличения')
disp(['aw=', num2str(aw*M)]);
disp(['r4=', num2str(r4*M), ' ', 'r5=', num2str(r5*M)]);
disp(['rb4=', num2str(rb4*M), ' ', 'rb5=', num2str(rb5*M)]);
disp(['rf4=', num2str(rf4*M), ' ', 'rf5=', num2str(rf5*M)]);
disp(['ra4=', num2str(ra4*M), ' ', 'ra5=', num2str(ra5*M)]);
disp(['sh4=', num2str(sh4*M), ' ', 'sh5=', num2str(sh5*M)]);
disp(['ph4=', num2str(ph4*M), ' ', 'ph5=', num2str(ph5*M)]);

```

Список рекомендуемой литературы

1. Ануфриев, И. Е. Самоучитель MatLab 5.3/6.x. – СПб. : БХВ-Петербург, 2002. – 736 с. : ил.
2. Мартынов, Н. Н. Введение в MatLab 6. – М. : КУДИЦ-ОБРАЗ, 2002. – 352 с.
3. Потёмкин, В. Г. Введение в MatLab. – М. : Диалог-МИФИ, 2000. – 247 с.
4. Ермак, В. Н. Лекции по теории механизмов и машин : учеб. пособие / Кузбас. гос. техн. ун-т. – Кемерово, 1999. – 218 с.
5. Ермак, В. Н. Теория механизмов и машин (курсовое проектирование) : учеб. пособие / В. Н. Ермак, Н. П. Курышкин ; ГУ КузГТУ. – Кемерово, 2004. – 181 с.
6. Справочник по геометрическому расчёту эвольвентных зубчатых и червячных передач / И. А. Болотовский, В. И. Безруков, О. Ф. Васильева [и др.] ; под ред. И. А. Болотовского. – М. : Машиностроение, 1986. – 448 с.

Оглавление

Предисловие	3
Занятие 1. Введение в MatLab	4
Занятие 2. Кинематические диаграммы	12
Занятие 3. Планы положений и анимация	24
Занятие 4. Силовой расчёт рычажного механизма. Формирование массивов внешней силы и силы инерции	36
Занятие 5. Силовой расчёт рычажного механизма. Составление и решение уравнений равновесия	45
Занятие 6. Силовой расчёт рычажного механизма. Построение годографов	51
Занятие 7. Кинематический анализ манипуляторов	57
Занятие 8. Кинематический анализ кардана	63
Приложение 1. Задания на силовой расчёт	67
Приложение 2. Силовой расчёт двухтактного ДВС	73
Приложение 3. Программы геометрического расчёта зацепления	80
Список рекомендуемой литературы	84