

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Ивановский государственный политехнический университет»

А. А. Тувин, Р. В. Шляпугин, Д. А. Пирогов

АВТОМАТИЗИРОВАННЫЙ РАСЧЕТ КУЛАЧКОВО-СТЕРЖНЕВЫХ МЕХАНИЗМОВ

*Рекомендовано учебно-методическим советом ИВГПУ
в качестве учебного пособия для студентов направления подготовки
бакалавров 15.03.02 Технологические машины и оборудование*

Иваново 2018

УДК 531.8(519.677)

Тувин, А. А. Автоматизированный расчет кулачково-стержневых механизмов : учеб. пособие для студ. направления подгот. бакалавров 15.03.02 *Технологические машины и оборудование* / А. А. Тувин, Р. В. Шляпугин, Д. А. Пирогов. – Иваново : ИВГПУ, 2018. – 224 с. : ил.

В учебном пособии изложена методика составления алгоритмов и программ для кинематического и силового анализа стержневых механизмов II класса, представлены подпрограммы расчета кинематических и силовых параметров для групп Ассура II класса 1–5 видов, рассмотрены алгоритмы определения геометрических, кинематических и силовых параметров кулачковых механизмов: плоских с толкателями, совершающими возвратно-вращательное и поступательное движения; пространственных с возвратно-вращательным движением толкателя.

Пособие может быть использовано при курсовом и дипломном проектировании студентами направления подготовки бакалавров 15.03.02 *Технологические машины и оборудование*.

Рецензенты:

кафедра машин и аппаратов химических производств
Ивановского государственного химико-технологического университета
(заведующий кафедрой доктор технических наук, профессор **В. Н. Блиничев**);

доктор физико-математических наук, доцент,
заведующий кафедрой теоретической и прикладной механики
Ивановского государственного энергетического университета
им. В. И. Ленина **Л. Б. Маслов**

Учебное издание

Тувин Александр Алексеевич
Шляпугин Роман Владимирович
Пирогов Дмитрий Андреевич

**АВТОМАТИЗИРОВАННЫЙ РАСЧЕТ
КУЛАЧКОВО-СТЕРЖНЕВЫХ МЕХАНИЗМОВ**

Редактор **О. Р. Ростов**

Подписано в печать 20.02.2018. Формат 60 × 84 ¹/₁₆. Трафаретная печать.
Усл. печ. л. 13,02. Уч.-изд. л. 10,0. Тираж 50 экз.

ФГБОУ ВО «Ивановский государственный политехнический университет»
Издательский центр ДИВТ
153000, г. Иваново, Шереметевский проспект, 21

ISBN 978-5-88954-464-7

© ФГБОУ ВО «ИВГПУ», 2018
© Тувин А. А., Шляпугин Р. В., Пирогов Д. А., 2018

О Г Л А В Л Е Н И Е

Введение	6
РАЗДЕЛ 1. Методика кинематического и кинетостатического анализа плоских рычажных механизмов	9
1.1. Общие замечания по моделированию рычажных механизмов	9
1.2. Математическое моделирование рычажных механизмов	10
1.3. Основные понятия вектора и векторных исчислений	11
1.4. Кинематический анализ рычажных механизмов	13
1.4.1. Математические модели, используемые при анализе рычажных (стержневых) механизмов	13
1.4.2. Кинематический анализ группы Ассура II класса 1 вида	20
1.4.3. Кинематический анализ группы Ассура II класса 2 вида	25
1.4.4. Кинематический анализ группы Ассура II класса 3 вида	30
1.4.5. Кинематический анализ группы Ассура II класса 4 вида	34
1.4.6. Кинематический анализ группы Ассура II класса 5 вида	39
1.4.7. Контрольные вопросы по кинематическому анализу рычажных механизмов	44
1.5. Силовой анализ рычажных механизмов	45
1.5.1. Математическое моделирование силового анализа отдельного звена	47
1.5.2. Силовой анализ группы Ассура II класса 1 вида	49
1.5.3. Силовой анализ группы Ассура II класса 2 вида	53
1.5.4. Силовой анализ группы Ассура II класса 3 вида	55
1.5.5. Силовой анализ группы Ассура II класса 4 вида	58
1.5.6. Силовой анализ группы Ассура II класса 5 вида	61
1.5.7. Силовой анализ входного звена	64
1.5.8. Контрольные вопросы по силовому анализу рычажных механизмов	71

РАЗДЕЛ 2. Математические модели, используемые при анализе кулачковых механизмов	73
2.1. Проектирование профиля плоского кулачка с возвратно-вращательным движением толкателя	73
2.1.1. Расчетная схема механизма	73
2.1.2. Определение входных и постоянных параметров кулачкового механизма	74
2.1.3. Определение координат точек теоретического профиля кулачка	76
2.1.4. Определение радиуса кривизны и координат центра кривизны профиля кулачка в системе XOY	79
2.1.5. Определение углов давления	80
2.2. Проектирование профиля плоского кулачка с возвратно-поступательным движением толкателя	81
2.2.1. Определение координат профиля кулачка	81
2.2.2. Входные параметры при расчете кулачкового механизма на ЭВМ	84
2.3. Проектирование профиля пространственного кулачка с возвратно-вращательным движением толкателя	84
2.3.1. Определение координат профиля пространственного кулачка	84
2.3.2. Определение угла давления и радиуса кривизны теоретического профиля кулачка	86
2.3.3. Входные параметры при расчете пространственного кулачкового механизма на ЭВМ	88
2.4. Законы движения толкателей	88
2.5. Блок-схема программы расчета геометрических, кинематических и динамических параметров кулачка	97
РАЗДЕЛ 3. Основы программирования в Visual Basic	99
3.1. Общие сведения о языке программирования	99
3.2. Настройка приложения	103

3.3. Интерфейс приложения	104
3.4. Форма и объекты	110
3.5. Подпрограмма командного кода	113
3.6. Переменные и массивы	119
РАЗДЕЛ 4. Пример выполнения самостоятельной работы	128
4.1. Пример выполнения кинематического расчета рычажного механизма	128
4.1.1. Исходные данные для расчета механизма	128
4.1.2. Структурный анализ механизма	129
4.1.3. Блок-схема программы кинематического расчета механизма	130
4.1.4. Создание проекта	133
4.2. Пример выполнения кинетостатического расчета рычажного механизма	152
4.2.1. Исходные данные для расчета механизма	152
4.2.2. Структурный анализ механизма	154
4.2.3. Блок-схема программы кинетостатического расчета механизма	154
4.2.4. Создание проекта	157
4.3. Пример расчета механизма с кулачковым приводом	173
4.3.1. Исходные данные для расчета механизма	173
4.3.2. Структурный анализ механизма	175
4.3.3. Блок-схема программы кинетостатического расчета кулачко-рычажного механизма	176
4.3.4. Создание проекта	179
РАЗДЕЛ 5. Задание для самостоятельной работы	209
5.1. Содержание задания	209
5.2. Варианты задания	211
Библиографический список	224

ВВЕДЕНИЕ

Проектирование представляет собой область умственного труда, крайне нуждающуюся в автоматизации, т. к. решаемые здесь задачи сложные, и порой даже большой коллектив конструкторов не может найти действительно лучший вариант проекта.

Основная проблема в этой области состоит сейчас уже не в разработке методов проектирования, рассчитанных на использование их человеком, а в создании таких методов, которые были бы ориентированы на применение ЭВМ.

Автоматизация проектирования заключается в систематическом применении ЭВМ для выполнения проектных операций и процедур при рациональном использовании творческих способностей человека и вычислительных возможностей машины. Автоматизация необходима для выполнения тяжелых рутинных работ, не требующих творческого подхода. Большой эффект приносят разработки математического и программного обеспечения для конструкторского проектирования.

Автоматизация проектирования помогает инженеру быстро проверить гипотезу, оценить и модифицировать ее для следующего испытания в многошаговом процессе оптимизации. Она позволяет снабдить проектировщиков средствами оперативного получения необходимой информации, быстро оценивать различные варианты сочетания производств с учетом их внешних и внутренних связей, технического оснащения, автоматизировать трудоемкие работы по оформлению проектной документации. Все это возможно только на основе современных методов математического моделирования, оптимизации и системного анализа.

Таким образом, автоматизация проектирования заключается в передаче ЭВМ некоторых функций проектирования, выполняемых в настоящее время людьми, а ее основной целью является повышение эффективности, т. е. повышение качества и экономия ресурсов проектирования. В этой связи глав-

ная задача при создании САПР (системы автоматизированного проектирования) – не сокращение сроков разработки и не уменьшение числа специалистов, а улучшение качества проектирования.

Рассмотрим основные этапы решения задач проектирования на ЭВМ.

1. Постановка задачи. На этом этапе по имеющимся исходным данным формируются цель и задачи проектирования, определяются условия, которым должно удовлетворять проектное решение, разрабатывается метод и выбираются средства решения задач.

2. Составление математической модели. По результатам первого этапа разрабатывается вариант математического решения (математической модели) поставленной задачи, соответствующий с достаточной точностью реальному объекту, выделяется перечень исходных данных, определяются начальные и граничные условия. Математическая модель представляет собой систему уравнений, решением которой достигается поставленная цель. В системе используют дифференциальные уравнения кинематики и кинетостатики, трансцендентные уравнения и другие, описывающие реальные физические процессы и состояние исследуемого или проектируемого объекта.

3. Составление алгоритма решения задачи. Для численного решения поставленной задачи на ЭВМ по разработанной математической модели необходимо составить логическую схему вычислительного процесса, т. е. точное, полное и однозначное определение последовательности этого процесса. На этом же этапе при необходимости разрабатывается блок-схема решения задачи. Если предусматривается диалоговый режим вычислений, то в вычислительном процессе планируется точка выхода и входа для связи с пользователем.

4. Составление программ (программирование). Это этап, на котором:

- разрабатываются имена (идентификаторы) и структура используемых данных (типы величин);
- планируется размещение данных на машинных носителях (назначаются устройства ввода-вывода информации);

- составляется блок-схема программы с учетом особенностей выбранного языка программирования, на основании которой разрабатывается программа решения той или иной задачи.

5. Отладка программы. Этап отладки – один из самых ответственных и трудоемких. Вследствие возможных ошибок на всех предыдущих этапах (синтаксических ошибок, логических ошибок в формулировке задачи, опечаток, опечаток и др.) при реализации программы на ЭВМ могут быть либо неверные результаты, либо останов программы. Синтаксические ошибки выявляются в процессе трансляции программы, а все остальные – в счетном режиме: путем сравнения полученных результатов с контрольными данными. С целью упрощения процесса отладки желательно предусмотреть в программе дополнительные операторы для вывода промежуточных результатов. После окончательной отладки программы эти операторы исключаются.

6. Решение задачи на ЭВМ и анализ результатов. После отладки программы выполняется ее непосредственное решение с целью получения результата для всех вариантов исходных данных задачи. На этом этапе происходит собственно счет. Затем проводится обработка (анализ и интерпретация) полученных результатов: требуется правильно расшифровать результаты и дать им нужное толкование.

РАЗДЕЛ 1

МЕТОДИКА КИНЕМАТИЧЕСКОГО И КИНЕТОСТАТИЧЕСКОГО АНАЛИЗА ПЛОСКИХ РЫЧАЖНЫХ МЕХАНИЗМОВ

1.1. Общие замечания по моделированию рычажных механизмов

Известно, что конечным результатом кинематического исследования является определение закономерности изменения кинематических параметров звеньев, их соответствие геометрическим параметрам механизма и кинематике входного звена.

Соответственно задачами кинематического анализа являются:

- определение положений звеньев и траекторий движения, которые описывают отдельные точки звеньев;
- определение скоростей звеньев и скоростей отдельных точек этих звеньев;
- определение ускорений звеньев и ускорений отдельных точек этих звеньев.

Все существующие методы определения кинематических параметров механизмов делятся на *графические*, *графоаналитические* и *аналитические*.

К *графическим* методам можно отнести метод диаграмм, к *графоаналитическим* методам – метод точек Ассура, ложных планов, метод мгновенных центров скоростей и ускорений и др. Однако наиболее распространенным из графоаналитических методов является метод построения планов скоростей и ускорений (метод О. Мора). Все эти методы отличаются простотой и наглядностью, но точность результатов исследований невелика вследствие погрешностей, возникающих при проведении графических построений.

Этого недостатка лишены аналитические методы исследования, но сложность и трудность расчетов значительно возрастает, что вызывает необходимость автоматизации данных расчетов. К основным *аналитическим*

методам можно отнести методы В. А. Зиновьева [1], Ю. Ф. Морошкина [2], П. А. Лебедева [3] и др.

В данном учебном пособии кинематический анализ рычажных механизмов проведен с помощью метода В. А. Зиновьева, основанного на решении уравнений замкнутых векторных контуров. Этот метод является одним из самых простых и наглядных и рассматривается при изучении таких дисциплин, как высшая математика (раздел векторной геометрии), теоретическая механика, теория механизмов и машин и др.

1.2. Математическое моделирование рычажных механизмов

Возможность разложения сложного многозвенного механизма на более простые механизмы и отдельные структурные образования позволяет получать достаточно простые математические модели и этих структурных образований, и самих механизмов.

Предварительно разрабатываются математические модели, описывающие кинематику отдельных структурных образований плоских кулачковых, кулачково-рычажных и рычажных механизмов. После этого составление математической модели исследуемого конкретного механизма будет заключаться лишь в подключении (в соответствии с его структурой) необходимых математических моделей входящих в его состав структурных образований. Такой конструктивный подход позволяет значительно упростить кинематический анализ любой сложности для механизмов широкого класса и свести разработку алгоритма анализа механизма к последовательным обращениям к стандартным процедурам, описывающим кинематику отдельных структурных образований механизма.

Применение ЭВМ при анализе механизмов требует преобразования обычно используемой графической информации в математическую. Такое преобразование можно проводить путем создания разнообразных математических моделей механизмов.

Разработка системы анализа механизмов, ориентированной на применение ЭВМ, требует построения общей модели любого механизма. Общая модель может быть сформирована на основе представления реального звена механизма в виде вектора с постоянными или переменными модулями. Замена звеньев механизма конечными системами векторов приводит к геометрической модели механизма, приближающейся по смыслу к кинематической схеме, т. е. кинематическая схема есть графическая модель механизма.

1.3. Основные понятия вектора и векторных исчислений

Любой вектор \bar{l} можно представить в виде произведения его модуля $l(\varphi_1)$ как функции обобщенной координаты φ_1 и единичного вектора \bar{e} , направление которого определяется углом φ_m , а положение в плоскости – точкой приложения. Соответственно

$$\bar{l} = l(\varphi_1) \cdot \bar{e}. \quad (1.1)$$

Точка приложения вектора \bar{l} (или \bar{e}) определяется координатами в принятой инерциальной прямоугольной системе XOY . Единичные векторы осей координат (орты) обозначаются \bar{i}, \bar{j} .

Вспомним правила действия с векторами [4]. Скалярное произведение двух векторов равно длине одного из них, умноженной на алгебраическую проекцию другого вектора по направлению первого. Соответственно

$$\bar{l} \cdot \bar{m} = l \cdot m \cdot \cos(\bar{l} \wedge \bar{m}). \quad (1.2)$$

Отсюда следует, что скалярное произведение двух единичных векторов равно косинусу угла между ними:

$$\bar{e}_k \cdot \bar{e}_n = \cos(\bar{e}_k \wedge \bar{e}_n); \quad (1.3)$$

$$\bar{e}_k \cdot \bar{e}_n = \cos(\varphi_k - \varphi_n). \quad (1.4)$$

Произведения единичного вектора на орты системы (рис. 1.1)

$$\bar{e}_1 \cdot \bar{i} = \cos \varphi_1; \quad (1.5)$$

$$\bar{e}_1 \cdot \bar{j} = \sin \varphi_1. \quad (1.6)$$

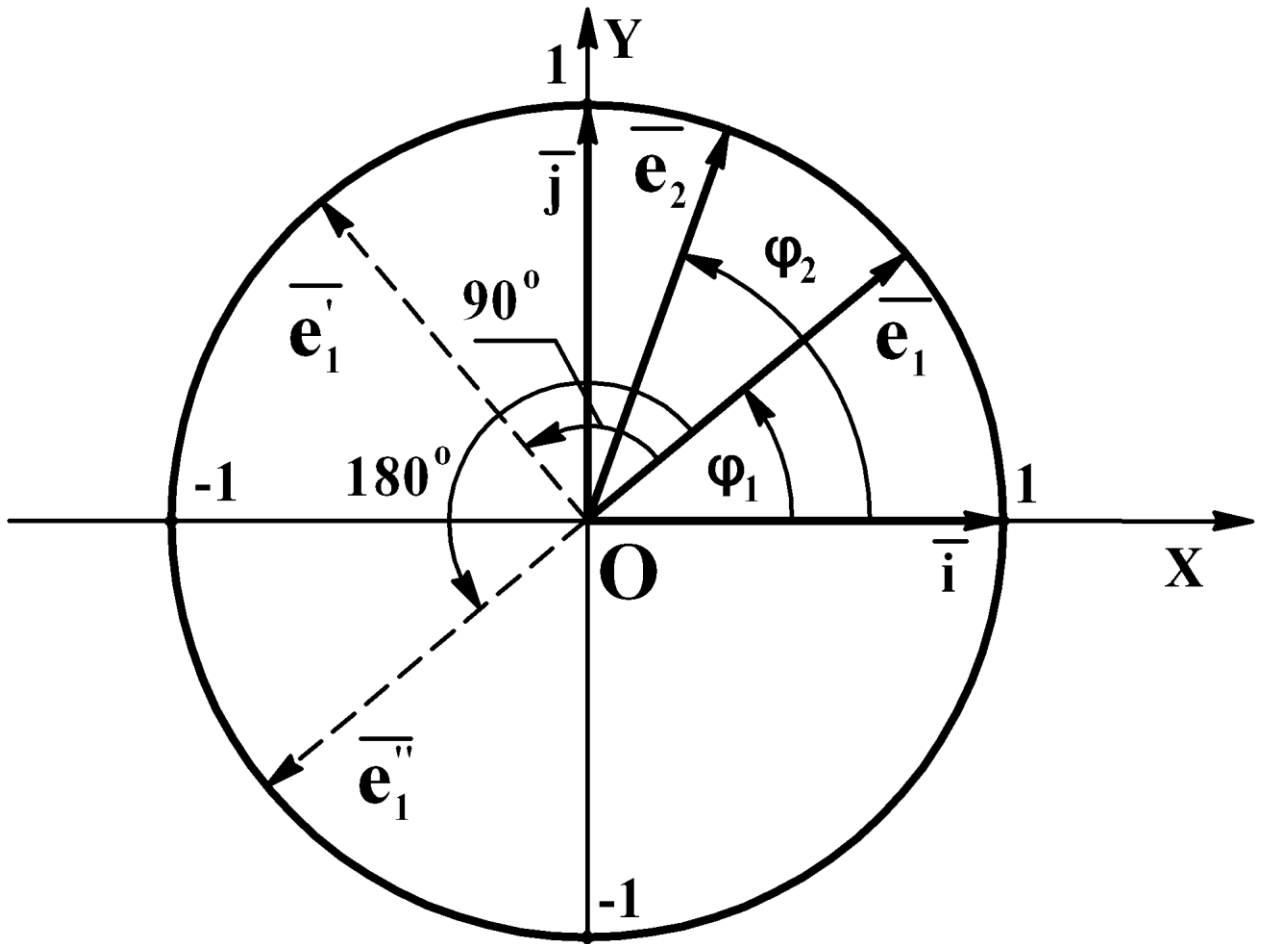


Рис. 1.1. Расчетная схема

Производная единичного вектора по обобщенной координате имеет вид

$$\frac{d\bar{e}_k}{d\varphi_1} = \frac{d\bar{e}_k}{d\varphi_k} \frac{d\varphi_k}{d\varphi_1}. \quad (1.7)$$

Обозначим

$$\frac{d\bar{e}_k}{d\varphi_k} = \bar{e}'_k; \quad (1.8)$$

$$\frac{d\varphi_k}{d\varphi_1} = \varphi'_k, \quad (1.9)$$

где \bar{e}'_k – новый единичный вектор, полученный поворотом орта \bar{e}_k на 90° в положительном направлении (причем в правой системе координат положительное направление вращения считается от оси OX к оси OY , т. е. против часовой стрелки, а в левой – по часовой стрелке (см. рис. 1.1));

φ'_k – аналог угловой скорости вектора.

Вектор \bar{e}'_k координируется углом $\varphi_k + 0,5 \cdot \pi$.

Подстановкой выражений (1.8) и (1.9) в уравнение (1.7) получим

$$\frac{d\bar{e}_k}{d\varphi_1} = \bar{e}'_k \cdot \varphi'_k. \quad (1.10)$$

После второго дифференцирования единичного вектора по обобщенной координате получим выражение

$$\frac{d^2\bar{e}_k}{d\varphi_1^2} = (\varphi'_k)^2 \cdot \bar{e}''_k + \bar{e}'_k \cdot \varphi''_k, \quad (1.11)$$

где \bar{e}''_k – единичный вектор, полученный поворотом орта \bar{e}'_k на 90° или поворотом орта \bar{e}_k на 180° в положительном направлении (см. рис. 1.1);

φ''_k – аналог углового ускорения вектора.

Вектор \bar{e}''_k координируется углом $\varphi_k + \pi$.

1.4. Кинематический анализ рычажных механизмов

1.4.1. Математические модели, используемые при анализе рычажных (стержневых) механизмов

Рассмотрим произвольное звено AB , представленное в виде вектора \bar{l}_m , координирующегося углом φ_m (рис. 1.2). Любую точку C звена AB будем координировать относительно вектора \bar{l} этого звена радиусом-вектором \bar{R}_C , выходящим из точки A приложения вектора \bar{l}_m , и углом β_C , отсчитываемым в положительном направлении. Положительный отсчет углов производится от оси OX принятой правой системы координат в сторону положительной оси OY .

Положение точки C определяется координатами

$$\left. \begin{aligned} X_C &= X_A + R_C \cdot \cos(\varphi_m + \beta_C), \\ Y_C &= Y_A + R_C \cdot \sin(\varphi_m + \beta_C), \end{aligned} \right\} \quad (1.12)$$

где X_C, Y_C, X_A, Y_A – координаты соответствующих точек в системе XOY ;

φ_m – координата вектора \bar{l}_m ;

R_C, β_C – полярные координаты точки C .

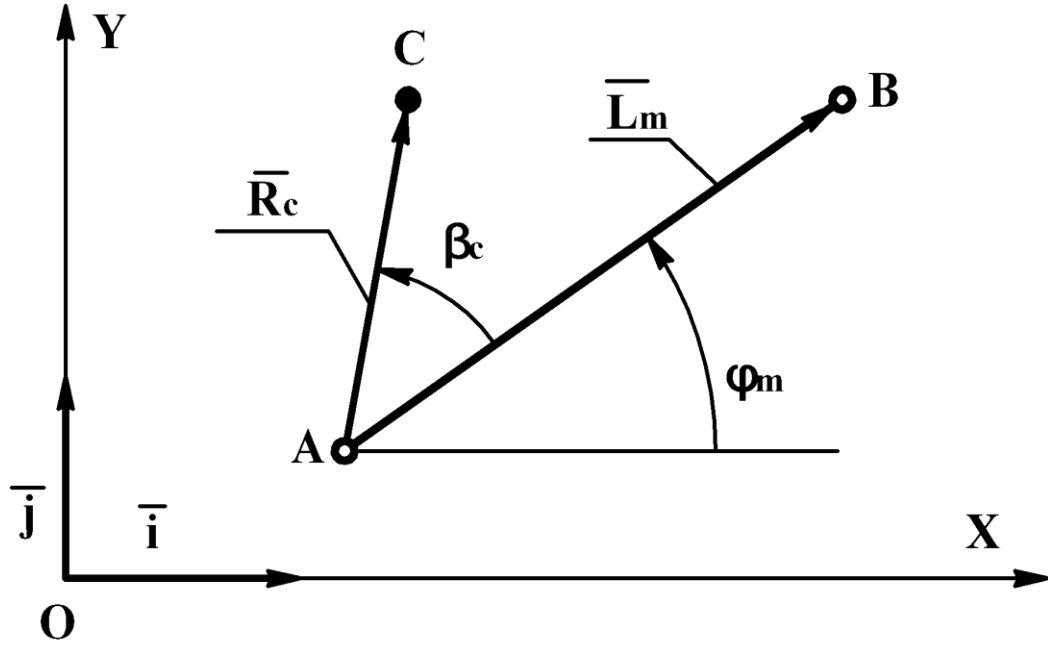


Рис. 1.2. Расчетная схема отдельного звена

При дифференцировании системы (1.12) по обобщенной координате φ_1 при известных функциях $X_A(\varphi_1)$, $Y_A(\varphi_1)$ и $\varphi_m(\varphi_1)$ определяются проекции аналогов линейных скоростей и ускорений точки C:

$$\left. \begin{aligned} X'_C &= X'_A - R_C \cdot \varphi'_m \sin(\varphi_m + \beta_C), \\ Y'_C &= Y'_A + R_C \cdot \varphi'_m \cos(\varphi_m + \beta_C), \end{aligned} \right\} \quad (1.13)$$

$$\left. \begin{aligned} X''_C &= X''_A - R_C (\varphi'_m)^2 \cos(\varphi_m + \beta_C) - R_C \cdot \varphi''_m \cdot \sin(\varphi_m + \beta_C), \\ Y''_C &= Y''_A - R_C (\varphi'_m)^2 \sin(\varphi_m + \beta_C) + R_C \cdot \varphi''_m \cdot \cos(\varphi_m + \beta_C), \end{aligned} \right\} \quad (1.14)$$

где X'_C , Y'_C , X'_A , Y'_A – проекции аналогов линейных скоростей соответствующих точек в системе XOY ;

X''_C , Y''_C , X''_A , Y''_A – проекции аналогов линейных ускорений соответствующих точек в системе XOY ;

φ'_m , φ''_m – аналоги угловых скорости и ускорения звена AB .

Величины модулей абсолютных векторов аналогов линейных скоростей и ускорений искомой точки определяется по формулам

$$V_C = \sqrt{(X'_C)^2 + (Y'_C)^2}; \quad (1.15)$$

$$W_C = \sqrt{(X''_C)^2 + (Y''_C)^2}. \quad (1.16)$$

Направление вектора \bar{V}_C определяется по выражению

$$\varphi_{V_C} = \arctg\left(\frac{Y'_C}{X'_C}\right) + k\pi, \quad (1.17)$$

где

$$k = \begin{cases} 0, & \text{при } X'_C > 0, Y'_C > 0, \\ 1, & \text{при } X'_C < 0, \\ 2, & \text{при } X'_C > 0, Y'_C < 0, \end{cases} \quad (1.18)$$

$$\varphi_{V_C} = 0,5 \cdot \pi \text{ при } X'_C = 0, Y'_C > 0; \quad (1.19)$$

$$\varphi_{V_C} = 1,5 \cdot \pi \text{ при } X'_C = 0, Y'_C < 0. \quad (1.20)$$

При определении угла φ_{w_C} , координирующего вектор \bar{w}_C , используются формулы, аналогичные (1.17)–(1.20), при соответствующей замене индексов.

В задачах анализа возможны случаи, когда вектор задан координатами его начала (X_A, Y_A) и конца (X_B, Y_B) . Тогда модуль вектора \bar{l}_m и угол φ_m вычисляются из выражений

$$\left. \begin{aligned} l_m &= \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2} \\ \varphi_{l_m} &= \arctg\left(\frac{Y_B - Y_A}{X_B - X_A}\right) + k\pi \end{aligned} \right\}. \quad (1.21)$$

При анализе механизмов требуется знать производные векторов \bar{l}_m по обобщенной координате φ_1 :

$$\frac{d}{d\varphi_1}(l_m \bar{e}_m) = l_m \frac{d\bar{e}_m}{d\varphi_1} + \bar{e}_m \frac{dl_m}{d\varphi_1}; \quad (1.22)$$

$$\frac{d^2 \bar{l}_m}{d\varphi_1^2} = l_m \frac{d^2 \bar{e}_m}{d\varphi_1^2} + 2 \frac{dl_m}{d\varphi_1} \frac{d\bar{e}_m}{d\varphi_1} + \bar{e}_m \frac{d^2 l_m}{d\varphi_1^2}. \quad (1.23)$$

Производные единичных векторов \bar{e}_m будут равны:

$$\frac{d\bar{e}_m}{d\varphi_1} = \varphi'_m \bar{e}'_m; \quad (1.24)$$

$$\frac{d^2 \bar{e}_m}{d\varphi_1^2} = \varphi''_m \bar{e}'_m + (\varphi'_m)^2 \bar{e}''_m, \quad (1.25)$$

где
$$\varphi'_m = \frac{d\varphi_m}{d\varphi_1}; \quad (1.26)$$

$$\varphi''_m = \frac{d^2\varphi_m}{d\varphi_1^2}. \quad (1.27)$$

При подстановке (1.26) в (1.22) и (1.27) в (1.23) окончательно имеем

$$\frac{d\bar{l}_m}{d\varphi_1} = l_m \varphi'_m \bar{e}'_m + l'_m \bar{e}_m; \quad (1.28)$$

$$\frac{d^2\bar{l}_m}{d\varphi_1^2} = l_m (\varphi'_m)^2 \bar{e}''_m + l_m \varphi''_m \bar{e}'_m + 2l'_m \varphi'_m \bar{e}'_m + l''_m \bar{e}_m, \quad (1.29)$$

где
$$l'_m = \frac{dl_m}{d\varphi_1}; \quad (1.30)$$

$$l''_m = \frac{d^2l_m}{d\varphi_1^2}. \quad (1.31)$$

В случаях, когда $l_m = const$, имеем $l'_m = l''_m = 0$.

В процессе разработки алгоритмов кинематического и силового анализа механизмов используются скалярные и векторные произведения векторов. На основании векторного исчисления [5] можно записать следующее:

$$\bar{e}_m \cdot \bar{e}_n = \cos(\varphi_m - \varphi_n); \quad (1.32)$$

$$\bar{e}_m \cdot \bar{e}'_n = \cos(\varphi_m - \varphi_n - 0,5 \cdot \pi); \quad (1.33)$$

$$\bar{e}_m \cdot \bar{e}''_n = \sin(\varphi_m - \varphi_n); \quad (1.34)$$

$$\bar{e}_m \cdot \bar{e}'''_n = \cos(\varphi_m - \varphi_n - \pi); \quad (1.35)$$

$$\bar{e}_m \cdot \bar{e}''''_n = -\cos(\varphi_m - \varphi_n); \quad (1.36)$$

$$\bar{e}_m \cdot \bar{e}_m = \cos(\varphi_m - \varphi_m); \quad (1.37)$$

$$\bar{e}_m \cdot \bar{e}_m = 1; \quad (1.38)$$

$$\bar{e}_m \cdot \bar{e}'_m = \cos(\varphi_m - \varphi_m - 0,5 \cdot \pi); \quad (1.39)$$

$$\bar{e}_m \cdot \bar{e}'_m = -1. \quad (1.40)$$

При переходе от аналогов скоростей к действительным значениям используются следующие формулы:

$$\omega_m = \varphi'_m \cdot \omega_1; \quad (1.41)$$

$$V_C = V_C \cdot \omega_1; \quad (1.42)$$

$$V_n = l'_m \cdot \omega_1, \quad (1.43)$$

где V_n – скорость точки звена в абсолютном или относительном поступательном движении;

V_C – скорость точки C в абсолютном движении;

ω_1 – обобщенная угловая скорость.

В общем случае при неравномерном движении входного звена механизма абсолютные угловые ускорения звеньев определяются по формуле

$$\varepsilon_m = \omega_1^2 \cdot \varphi''_m + \varepsilon_1 \cdot \varphi'_m; \quad (1.44)$$

При определении абсолютных линейных ускорений точек используется формула

$$W_C = W_C \cdot \omega_1^2 + V_C \cdot \varepsilon_1, \quad (1.45)$$

или (через проекции) формула

$$W_C = \sqrt{W_{C_X}^2 + W_{C_Y}^2}, \quad (1.46)$$

где

$$W_{C_X}^2 = X''_C \cdot \omega_1^2 + X'_C \cdot \varepsilon_1; \quad (1.47)$$

$$W_{C_Y}^2 = Y''_C \cdot \omega_1^2 + Y'_C \cdot \varepsilon_1. \quad (1.48)$$

$$\text{Если } \varepsilon_1 = 0, \text{ то } W_C = W_C \cdot \omega_1^2. \quad (1.49)$$

Приведенные выше формулы (1.12)–(1.20) при кинематическом анализе стержневого механизма используются неоднократно. С целью сокращения объема основной программы расчета кинематических параметров, времени ее разработки и отладки целесообразно иметь следующие подпрограммы (рис. 1.3):

- *PKT* – для вычисления координат промежуточных точек по формулам (1.12);
- *PMY* – для вычисления модуля вектора и направляющего угла по формулам (1.17)–(1.21);
- *PA* – для вычисления аналогов линейных скоростей и ускорений точек звеньев по формулам (1.13)–(1.20).

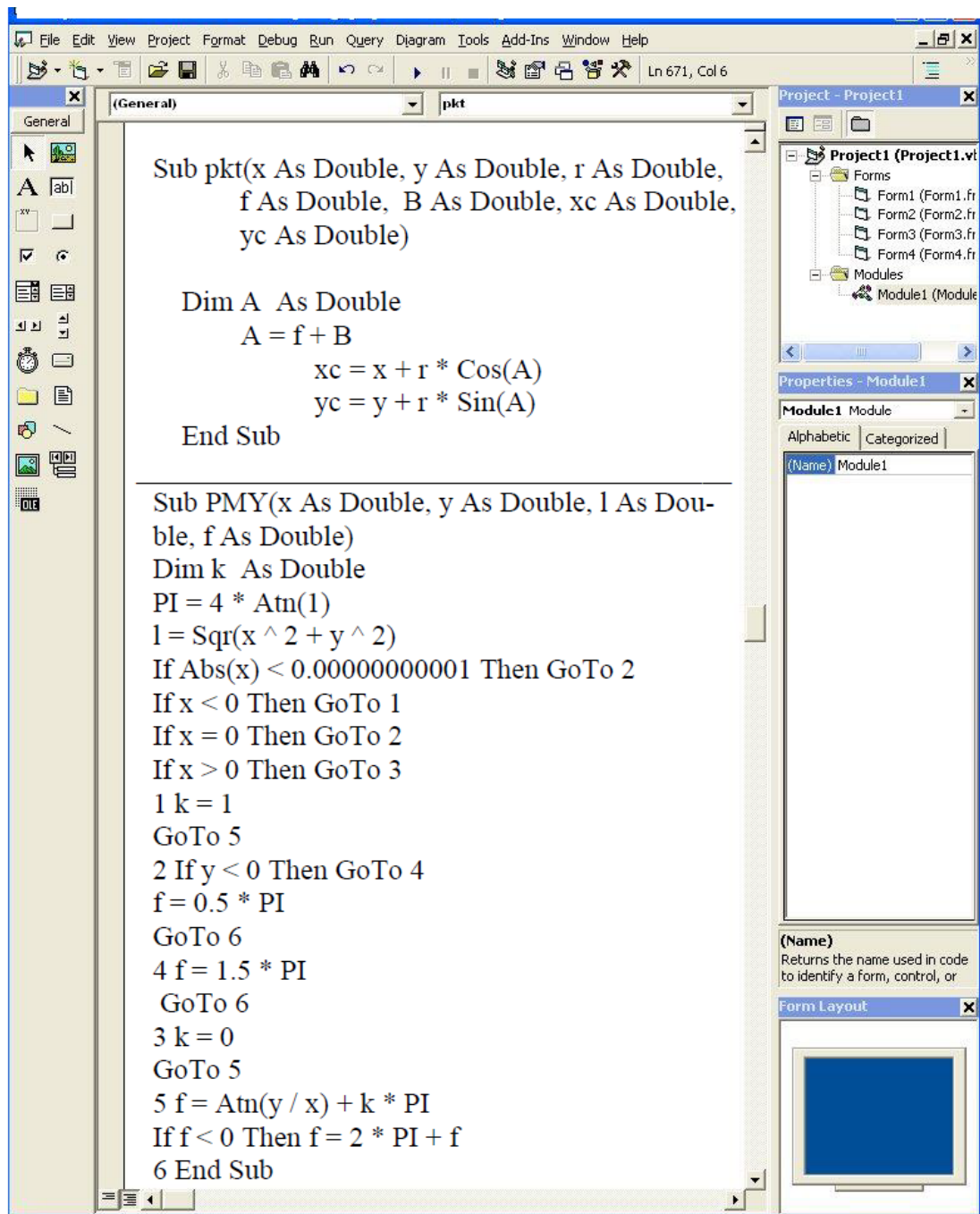


Рис. 1.3. Реализация подпрограмм PKT и PMY
в модуле Modul1(OAP.BAS)

В подпрограмме *PKT* входными величинами являются X_A , Y_A , R_C , φ_m , β_C . В коде подпрограммы они обозначаются соответственно X , Y , R , F , B . Выходные величины X_C и Y_C обозначаются XC и YC .

В подпрограмме *PMY* входные параметры обозначаются следующим образом: X и Y – проекции вектора на оси координат, а выходные параметры таковы: L – модуль вектора, F – его направляющий угол.

В каждом отдельном проекте может возникнуть необходимость множественного применения указанных процедур. С учетом этого для размещения этих и последующих подпрограмм создается специальный модуль *Modul1* (*OAP.BAS*). В частности, реализация подпрограмм *PKT* и *PMY* с использованием языка программирования *Visual Basic* в модуле *Modul1* (*OAP.BAS*) показана на рис. 1.3 (вид остальных подпрограмм, которые также записаны в модуле *Modul1* (*OAP.BAS*), аналогичен).

На рис. 1.3 видно, что имя процедуры располагается после слова *Sub*. В скобках указываются наименования входных и выходных параметров (требуется строгое соблюдение положения каждой переменной и их количества). Код процедуры заключен между словами *Sub* и *End Sub*.

Входные параметры подпрограммы *PA*:

$$X1 = X'_A, Y1 = Y'_A, X2 = X''_A, Y2 = Y''_A, R = R_C, B = \beta_C,$$

$$F = \varphi_m, F1 = \varphi'_m, F2 = \varphi''_m.$$

Выходные параметры:

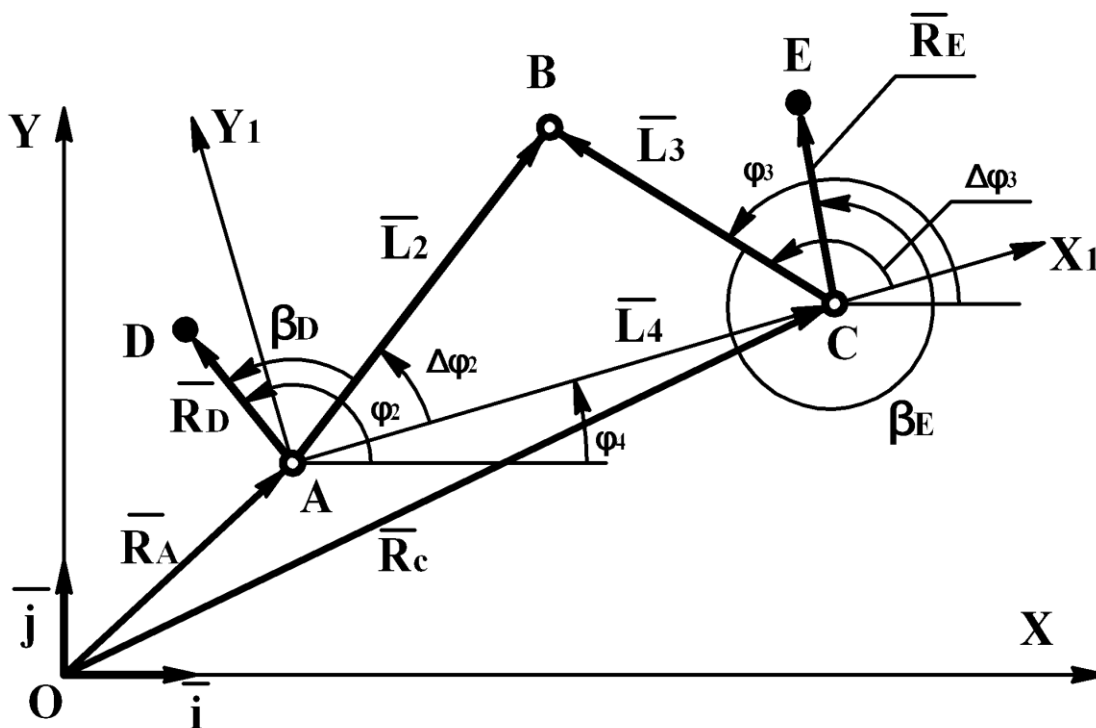
$$X1C = X'_C, Y1C = Y'_C, X2C = X''_C, Y2C = Y''_C.$$

В соответствии с формулами (1.13)–(1.20) подпрограмму *PA* можно представить следующим образом.

OAP.BAS	PA
Sub PA(va As Double, fva As Double, wa As Double, fwa As Double, _ r As Double, B As Double, f As Double, f1 As Double, _ f2 As Double, vc As Double, fvc As Double, wc As Double, _ fwc As Double) Dim A As Double, x1c As Double, y1c As Double, s As Double Dim C As Double, x2c As Double, y2c As Double A = f + B: s = r * Sin(A): C = r * Cos(A) x1c = va * Cos(fva) - f1 * s y1c = va * Sin(fva) + f1 * C x2c = wa * Cos(fwa) - f1 ^ 2 * C - f2 * s y2c = wa * Sin(fwa) - f1 ^ 2 * s + f2 * C Call PMY(x1c, y1c, vc, fvc) Call PMY(x2c, y2c, wc, fwc) End Sub	

Рассмотрим методики кинематического анализа групп Ассура II класса 1–5 видов.

Задача кинематического анализа группы Ассур II класса 1 вида (рис. 1.4) может быть решена, если известны положения, величины и направления аналогов скоростей и ускорений центров свободных элементов кинематических пар, которыми группа присоединяется к механизму при его образовании.



20

Исходными данными анализа являются:

L_2, L_3 – длины звеньев AB и CB ;

X_A, Y_A и X_C, Y_C – координаты точек A и C ;

R_D, β_D и R_E, β_E – полярные координаты точек D и E ;

V_{φ_A} и V_{φ_C} , W_{φ_A} и W_{φ_C} – аналоги линейных скоростей и ускорений точек A и C соответственно;

$\varphi_{V_A}, \varphi_{V_C}, \varphi_{W_A}, \varphi_{W_C}$ – углы, определяющие направления векторов $\bar{V}_{\varphi_A}, \bar{V}_{\varphi_C}, \bar{W}_{\varphi_A}, \bar{W}_{\varphi_C}$ в системе XOY .

Выходные данные (искомые величины):

φ_2 и φ_3 – углы, координирующие векторы \bar{L}_2 и \bar{L}_3 в системе координат XOY ;

φ'_2 и φ'_3 – аналоги угловых скоростей звеньев L_2 и L_3 соответственно;

φ''_2 и φ''_3 – аналоги угловых ускорений звеньев L_2 и L_3 соответственно.

К выходным данным также относим координаты, аналоги линейных скоростей и ускорений среднего шарнира B и заданных точек звеньев.

Введем подвижную систему координат X_1AY_1 , причем ось AX_1 проведем из точки A в точку C .

По отношению к этой системе координат рассмотрим замкнутый контур $ABCA$, уравнение которого имеет вид

$$\bar{L}_4 + \bar{L}_3 - \bar{L}_2 = 0; \quad (1.50)$$

$$L_4 \cdot \bar{e}_4 + L_3 \cdot \bar{e}_3 - L_2 \cdot \bar{e}_2 = 0. \quad (1.51)$$

Контур $ABCA$ будем называть положительным, если он расположен в I четверти подвижной системы координат X_1AY_1 , и отрицательным, если он расположен в IV четверти.

Будем характеризовать:

- параметром $z = +1$ – положительный контур;

- параметром $z = -1$ – отрицательный контур.

Модуль и направление вектора \bar{L}_4 в системе XOY определяются по следующим выражениям:

$$L_4 = \sqrt{(X_C - X_A)^2 + (Y_C - Y_A)^2}; \quad (1.52)$$

$$\varphi_4 = \arctg\left(\frac{Y_C - Y_A}{X_C - X_A}\right) + k\pi. \quad (1.53)$$

При решении уравнения (1.51) относительно вектора \bar{L}_3 находим

$$L_3 \cdot \bar{e}_3 = L_2 \cdot \bar{e}_2 - L_4 \cdot \bar{e}_4. \quad (1.54)$$

После возведения выражения (1.54) в скалярный квадрат и разрешения результата относительно угла $\Delta\varphi_2$ будем иметь

$$\Delta\varphi_2 = \arccos((L_2^2 + L_4^2 - L_3^2)/(2 L_2 \cdot L_4)). \quad (1.55)$$

Аналогично при разрешении уравнения (1.51) относительно \bar{L}_2 найдем

$$\Delta\varphi_3 = \arccos((L_2^2 - L_4^2 - L_3^2)/(2 L_3 \cdot L_4)). \quad (1.56)$$

Методом поворота координат [4] X_1AY_1 на угол φ_4 определяется направление в системе XOY :

$$\text{- вектора } \bar{L}_2: \quad \varphi_2 = \varphi_4 + z \cdot \Delta\varphi_2; \quad (1.57)$$

$$\text{- вектора } \bar{L}_3: \quad \varphi_3 = \varphi_4 + z \cdot \Delta\varphi_3. \quad (1.58)$$

Координаты точек B , D и E определяются по уравнению (1.12) при соответствующей замене индексов.

Для определения аналогов угловых скоростей звеньев L_2 и L_3 продифференцируем уравнение (1.54) по обобщенной координате φ_1 :

$$\frac{d\bar{L}_4}{d\varphi_1} + L_3 \cdot \varphi'_3 \cdot \bar{e}'_3 - L_2 \cdot \varphi'_2 \cdot \bar{e}'_2 = 0. \quad (1.59)$$

С другой стороны, рассмотрим контур $OACO$ и запишем уравнение его замкнутости:

$$\bar{L}_4 = \bar{R}_C - \bar{R}_A; \quad (1.60)$$

$$\frac{d\bar{L}_4}{d\varphi_1} = \bar{V}_{\varphi_C} - \bar{V}_{\varphi_A} = \bar{V}_{\varphi_C} \cdot \bar{e}_{V_C} - \bar{V}_{\varphi_A} \cdot \bar{e}_{V_A}. \quad (1.61)$$

Подстановкой (1.61) в (1.59) получаем

$$V_{\varphi_C} \cdot \bar{e}_{V_C} - V_{\varphi_A} \cdot \bar{e}_{V_A} + L_3 \cdot \varphi'_3 \cdot \bar{e}'_3 - L_2 \cdot \varphi'_2 \cdot \bar{e}'_2 = 0. \quad (1.62)$$

Умножим уравнение (1.62) на единичный вектор \bar{e}_3 :

$$V_{\varphi_C} \cdot \bar{e}_{V_C} \cdot \bar{e}_3 - V_{\varphi_A} \cdot \bar{e}_{V_A} \cdot \bar{e}_3 + L_3 \cdot \varphi'_3 \cdot \bar{e}'_3 \cdot \bar{e}_3 - L_2 \cdot \varphi'_2 \cdot \bar{e}'_2 \cdot \bar{e}_3 = 0. \quad (1.63)$$

После раскрытия скалярных произведений векторов и разрешения результата относительно φ'_2 получим

$$\varphi'_2 = \frac{-V_{\varphi_A} \cos(\varphi_{V_A} - \varphi_3) + V_{\varphi_C} \cos(\varphi_{V_C} - \varphi_3)}{L_2 \sin(\varphi_3 - \varphi_2)}. \quad (1.64)$$

Умножим уравнение (1.62) на единичный вектор \bar{e}_2 и аналогично найдем

$$\varphi'_3 = \frac{-V_{\varphi_A} \cos(\varphi_{V_A} - \varphi_2) + V_{\varphi_C} \cos(\varphi_{V_C} - \varphi_2)}{L_3 \sin(\varphi_3 - \varphi_2)}. \quad (1.65)$$

Результатом дифференцирования уравнения (1.62) будет выражение

$$W_{\varphi_C} \cdot \bar{e}_{W_C} - W_{\varphi_A} \cdot \bar{e}_{W_A} + L_3 \cdot \varphi''_3 \cdot \bar{e}'_3 + L_3 (\varphi'_3)^2 \bar{e}''_3 - \\ - L_2 \cdot \varphi''_2 \cdot \bar{e}'_2 - L_2 (\varphi'_2)^2 \bar{e}''_2 = 0. \quad (1.66)$$

Умножим уравнение (1.66) на \bar{e}_3 , после преобразования имеем

$$\varphi''_2 = \frac{W_{\varphi_C} \cos(\varphi_{W_C} - \varphi_3) - W_{\varphi_A} \cos(\varphi_{W_A} - \varphi_3)}{L_2 \sin(\varphi_3 - \varphi_2)} + \\ + \frac{L_2 (\varphi'_2)^2 \cos(\varphi_3 - \varphi_2) - L_3 (\varphi'_3)^2}{L_2 \sin(\varphi_3 - \varphi_2)}. \quad (1.67)$$

Умножим выражение (1.66) на \bar{e}_2 и после преобразования определим

$$\varphi''_3 = \frac{W_{\varphi_C} \cos(\varphi_{W_C} - \varphi_2) - W_{\varphi_A} \cos(\varphi_{W_A} - \varphi_2)}{L_3 \sin(\varphi_3 - \varphi_2)} + \\ + \frac{L_2 (\varphi'_2)^2 - L_3 (\varphi'_3)^2 \cos(\varphi_3 - \varphi_2)}{L_3 \sin(\varphi_3 - \varphi_2)}. \quad (1.68)$$

Аналоги скоростей и ускорений точек B , D и E находятся по формулам (1.13)–(1.20), т. е. с помощью подпрограммы PA . Вычисление кинематиче-

ских параметров $\varphi_2, \varphi_3, \varphi'_2, \varphi'_3, \varphi''_2, \varphi''_3, V_{\varphi_D}, V_{\varphi_E}, W_{\varphi_D}, W_{\varphi_E}, \varphi_{V_D}, \varphi_{V_E}, \varphi_{W_D}, \varphi_{W_E}$ группы Ассура II класса 1 вида проходит в подпрограмме *PK1*.

Входные величины в подпрограмме *PK1* обозначены следующим образом (j – индекс, определяющий порядковый номер группы Ассура в структурной схеме механизма):

$$L2(j) = L_2; L3(j) = L_3; XA(j) = X_A; YA(j) = Y_A; XC(j) = X_C;$$

$$YC(j) = Y_C; VA(j) = V_{\varphi_A}; VC(j) = V_{\varphi_C}; WA(j) = W_{\varphi_A}; WC(j) = W_{\varphi_C};$$

$$FVA(j) = \varphi_{V_A}; FVC(j) = \varphi_{V_C}; FWA(j) = \varphi_{W_A}; FWC(j) = \varphi_{W_C};$$

$$z(j) = z; RD(j) = R_D; BD(j) = \beta_D; RE(j) = R_E; BE(j) = \beta_E,$$

Выходные параметры:

$$F2 = \varphi_2; F3 = \varphi_3; F12 = \varphi'_2; F13 = \varphi'_3;$$

$$F22 = \varphi''_2; F23 = \varphi''_3;$$

$$VD = V_{\varphi_D}; VE = V_{\varphi_E}; WD = W_{\varphi_D}; WE = W_{\varphi_E};$$

$$FVD = \varphi_{V_D}; FVE = \varphi_{V_E}; FWD = \varphi_{W_D}; FWE = \varphi_{W_E}.$$

OAP.BAS	PK1
Sub pk1(f2 As Double, f3 As Double, f12 As Double, f13 As Double, _ f22 As Double, f23 As Double, vd As Double, ve As Double, _ wd As Double, we As Double, fvd As Double, fve As Double, _ fwd As Double, fwe As Double) Dim l4 As Double, f4 As Double, dx As Double, dy As Double Dim C As Double, A As Double, B As Double, a1 As Double Dim a2 As Double, b1 As Double, vex As Double, vey As Double Dim wex As Double, wey As Double, vdx As Double, vdy As Double Dim wdx As Double, wdy As Double, x As Double, B2 As Double dx = xc(j) - xa(j): dy = yc(j) - ya(j) Call PMY(dx, dy, l4, f4) x = (L2(j) ^ 2 + l4 ^ 2 - l3(j) ^ 2) / (2 * L2(j) * l4) f2 = f4 + z(j) * FNARC(x) x = (L2(j) ^ 2 - l4 ^ 2 - l3(j) ^ 2) / (2 * l3(j) * l4) f3 = f4 + z(j) * FNARC(x):C = f3 - f2: A = Sin(C): B = Cos(C) a1 = L2(j) * A: a2 = l3(j) * A	

```

f12 = (-va(j) * Cos(fva(j) - f3) + vc(j) * Cos(fvc(j) - f3)) / a1
f13 = (-va(j) * Cos(fva(j) - f2) + vc(j) * Cos(fvc(j) - f2)) / a2
b1 = L2(j) * B * f12 ^ 2: B2 = l3(j) * B * f13 ^ 2
f22 = (wc(j) * Cos(fwc(j) - f3) - wa(j) * Cos(fwa(j) - f3) -
- l3(j) * f13 ^ 2 + b1) / a1
f23 = (wc(j) * Cos(fwc(j) - f2) - wa(j) * Cos(fwa(j) - f2) -
+ L2(j) * f12 ^ 2 - B2) / a2
vex = vc(j) * Cos(fvc(j)) - re(j) * f13 * Sin(f3 + be(j))
vey = vc(j) * Sin(fvc(j)) + re(j) * f13 * Cos(f3 + be(j))
Call PMY(vex, vey, ve, fve)
wex = wc(j) * Cos(fwc(j)) - re(j) * f23 * Sin(f3 + be(j)) -
- re(j) * f13 ^ 2 * Cos(f3 + be(j))
wey = wc(j) * Sin(fwc(j)) + re(j) * f23 * Cos(f3 + be(j)) -
- re(j) * f13 ^ 2 * Sin(f3 + be(j))
Call PMY(wex, wey, we, fwe)
vdx = va(j) * Cos(fva(j)) - rd(j) * f12 * Sin(f2 + BD(j))
vdy = va(j) * Sin(fva(j)) + rd(j) * f12 * Cos(f2 + BD(j))
Call PMY(vdx, vdy, vd, fvd)
wdx = wa(j) * Cos(fwa(j)) - rd(j) * f22 * Sin(f2 + BD(j)) -
- rd(j) * f12 ^ 2 * Cos(f2 + BD(j))
wdy = wa(j) * Sin(fwa(j)) + rd(j) * f22 * Cos(f2 + BD(j)) -
- rd(j) * f12 ^ 2 * Sin(f2 + BD(j))
Call PMY(wdx, wdy, wd, fwd)
End Sub

```

1.4.3. Кинематический анализ группы Ассура II класса 2 вида

В общем виде группа Ассура II класса 2 вида показана на рис. 1.5.

В системе координат XOY положение группы будет строго определено, если известны:

- координаты свободного шарнира A звена 2;
- координаты какой-либо точки C направляющей 4;
- угол φ_4 , координирующий положение направляющей 4.

Для полного решения кинематического анализа должны быть известны:

- величины и направления аналогов линейных скоростей и ускорений точек A , C , т. е. V_A , $V_{\varphi A}$, V_C , $V_{\varphi C}$, W_A , $W_{\varphi A}$, W_C , $W_{\varphi C}$;
- аналоги угловых скоростей φ'_4 и ускорений φ''_4 направляющей 4.

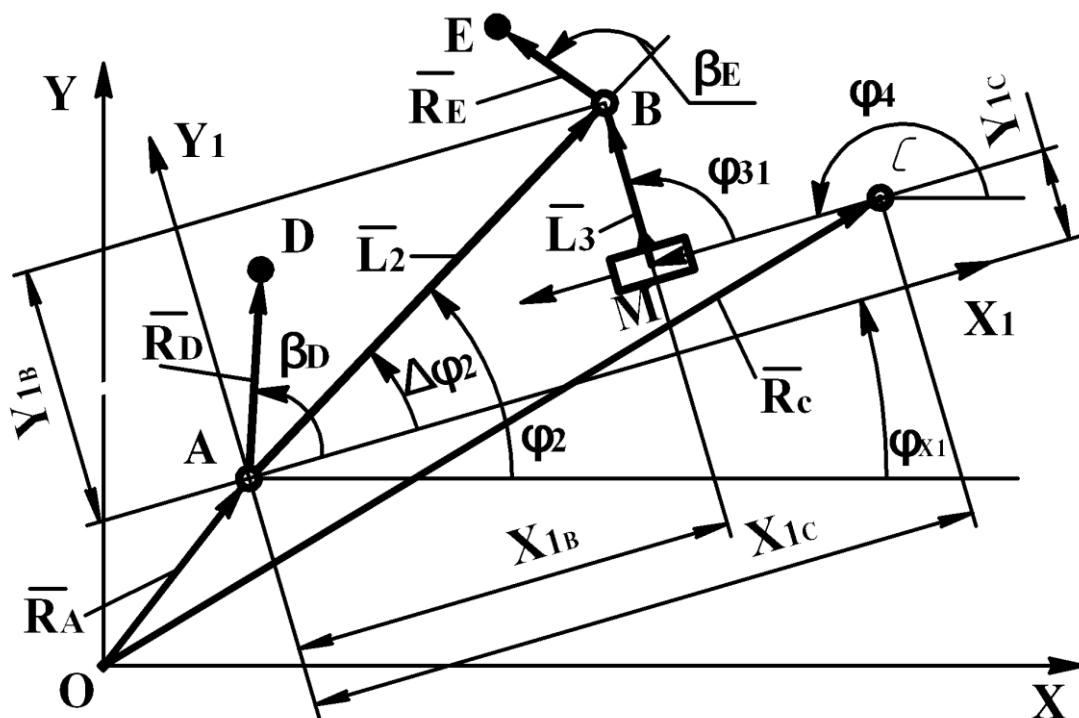


Рис. 1.5. Расчетная схема группы Ассура II класса 2 вида

Введем подвижную систему координат X_1AY_1 , причем ось AX_1 направим параллельно направляющей 4 в сторону ползуна 3. Тогда угол φ_{x_1} , координирующий направление оси AX_1 в системе XOY , при совпадении направлений оси и направляющей (что определяется в каждом конкретном случае) будет известен:

$$\varphi_{x_1} = \varphi_4. \quad (1.69)$$

В случае их противоположных направлений

$$\varphi_{x_1} = \varphi_4 + \pi. \quad (1.70)$$

Направление вектора \bar{L}_3 в системе координат X_1AY_1 определяет угол φ_{31} .

Если вектор \bar{L}_3 направлен:

$$\text{- в сторону оси } AY_1, \text{ то } \varphi_{31} = 0,5 \cdot \pi; \quad (1.71)$$

$$\text{- против оси } AY_1, \text{ то } \varphi_{31} = -0,5 \cdot \pi. \quad (1.72)$$

Методом переноса и поворота осей координат [5] найдем положение точки C в системе X_1AY_1 :

$$X_{1C} = (X_C - X_A) \cos(\varphi_{x_1}) + (Y_C - Y_A) \sin(\varphi_{x_1}); \quad (1.73)$$

$$Y_{1C} = -(X_C - X_A) \sin(\varphi_{x_1}) + (Y_C - Y_A) \cos(\varphi_{x_1}). \quad (1.74)$$

Координаты точки B в системе X_1AY_1

$$\left. \begin{aligned} Y_{1B} &= Y_{1C} + L_3 \sin(\varphi_{31}) \\ X_{1B} &= \sqrt{L_2^2 - Y_{1B}^2} \end{aligned} \right\}. \quad (1.75)$$

В силу принятого направления оси AX_1 имеем $X_{1B} > 0$. Тогда на основании рис. 1.5

$$\bar{L}_M = \bar{X}_{1B} - \bar{X}_{1C}; \quad (1.76)$$

$$L_M \cdot \bar{e}_{M_1} = X_{1B} \cdot \bar{i}_1 - X_{1C} \cdot \bar{i}_1, \quad (1.77)$$

где \bar{i}_1 – орт оси AX_1 ;

\bar{e}_{M_1} – единичный вектор соответственно вектора \bar{L}_M .

Умножением выражения (1.77) на орт \bar{i}_1 найдем

$$L_M \cdot \cos(\varphi_{M_1}) = X_{1B} - X_{1C}, \quad (1.78)$$

где φ_{M_1} – угол, координирующий вектор \bar{L}_M в системе координат X_1AY_1 .

$$|L_M| = |X_{1B} - X_{1C}|. \quad (1.79)$$

На основании выражения (1.78)

$$\varphi_{M_1} = \arccos\left(\frac{X_{1B} - X_{1C}}{L_M}\right). \quad (1.80)$$

На основании рис. 1.5

$$\varphi_2 = \varphi_{X_1} + \Delta\varphi_2, \quad (1.81)$$

$$\Delta\varphi_2 = \arctg\left(\frac{Y_{1B}}{X_{1B}}\right). \quad (1.82)$$

Для определения аналогов скоростей запишем векторное уравнение

$$R_A \cdot \bar{e}_{R_A} + L_2 \cdot \bar{e}_2 = R_C \cdot \bar{e}_{R_C} + L_M \cdot \bar{e}_M + L_3 \cdot \bar{e}_3, \quad (1.83)$$

где \bar{e}_M – единичный вектор, координирующийся углом φ_M ;

\bar{e}_3 – единичный вектор, координирующийся углом φ_3 .

$$\varphi_M = \varphi_{M_1} + \varphi_{X_1}; \quad (1.84)$$

$$\varphi_3 = \varphi_{31} + \varphi_{X_1}; \quad (1.85)$$

$$\varphi'_M = \varphi'_4; \quad (1.86)$$

$$\varphi''_M = \varphi''_4; \quad (1.87)$$

$$\cos(\varphi_M - \varphi_3) = 0. \quad (1.88)$$

В результате дифференцирования уравнения (1.83) по обобщенной координате имеем

$$\begin{aligned} V_{\varphi_A} \cdot \bar{e}_{V_A} + L_2 \cdot \varphi_2 \cdot \bar{e}'_2 = V_{\varphi_C} \cdot \bar{e}_{V_C} + L'_M \cdot \bar{e}_M + \\ + L_M \cdot \varphi'_4 \cdot \bar{e}'_M + L_3 \cdot \varphi'_4 \cdot \bar{e}'_3. \end{aligned} \quad (1.89)$$

Умножим уравнение (1.89) на \bar{e}'_M , после преобразования запишем

$$\varphi'_2 = \frac{V_{\varphi_A} \sin(\varphi_M - \varphi_{V_A}) - V_{\varphi_C} \sin(\varphi_M - \varphi_{V_C}) + L_M \varphi'_4}{L_2 \cos(\varphi_M - \varphi_2)}. \quad (1.90)$$

Умножим уравнение (1.89) на \bar{e}_M , преобразуем и получим

$$\begin{aligned} L'_M = V_{\varphi_A} \cdot \cos(\varphi_{V_A} - \varphi_M) - L_2 \cdot \varphi'_2 \cdot \sin(\varphi_2 - \varphi_M) - \\ - V_{\varphi_C} \cdot \cos(\varphi_{V_C} - \varphi_M) + L_3 \cdot \varphi'_4 \cdot \sin(\varphi_3 - \varphi_M). \end{aligned} \quad (1.91)$$

При дифференцировании уравнения (1.83) по обобщенной координате

$$\begin{aligned} W_{\varphi_A} \cdot \bar{e}_{W_A} + L_2 \cdot \bar{e}''_2 \cdot (\varphi'_2)^2 + L_2 \cdot \bar{e}'_2 \cdot \varphi''_2 = W_{\varphi_C} \cdot \bar{e}_{W_C} + \\ + L''_M \cdot \bar{e}_M + 2L'_M \cdot \bar{e}'_M \cdot \varphi'_4 + L_M \cdot \bar{e}'_M \cdot \varphi''_4 + \\ + L'_M \cdot \bar{e}''_M \cdot (\varphi'_4)^2 + L_3 \cdot \bar{e}''_3 \cdot (\varphi'_4)^2 + L_3 \cdot \varphi''_4 \cdot \bar{e}'_3. \end{aligned} \quad (1.92)$$

После умножения выражения (1.92) на \bar{e}'_M и преобразования получим

$$\varphi''_2 = \frac{\left[W_{\varphi_A} \cdot \sin(\varphi_M - \varphi_{W_A}) - L_2 \cdot (\varphi'_2)^2 \cdot \sin(\varphi_M - \varphi_2) - \right. \\ \left. - W_{\varphi_C} \cdot \sin(\varphi_M - \varphi_{W_C}) + 2L'_M \cdot \varphi'_4 + L_M \cdot \varphi''_4 + \right. \\ \left. + L_3 \cdot (\varphi'_4)^2 \cdot \sin(\varphi_M - \varphi_3) \right]}{L_2 \cos(\varphi_M - \varphi_2)}. \quad (1.93)$$

После умножения выражения (1.92) на \bar{e}_M и преобразования получим

$$\begin{aligned} L''_M = W_{\varphi_A} \cdot \cos(\varphi_M - \varphi_{W_A}) - L_2 \cdot (\varphi'_2)^2 \cdot \cos(\varphi_M - \varphi_2) + \\ + L_2 \cdot \varphi''_2 \cdot \sin(\varphi_M - \varphi_2) - W_{\varphi_A} \cdot \cos(\varphi_M - \varphi_{W_C}) + \\ + L'_M \cdot (\varphi'_4)^2 - L_3 \cdot \varphi''_4 \cdot \sin(\varphi_M - \varphi_3). \end{aligned} \quad (1.94)$$

Координаты, аналоги линейных скоростей и ускорений точек B , D и E определяются с помощью подпрограмм PKT , PMY и PA .

Приведенную методику кинематического анализа группы Ассура II класса 2 вида оформим в виде подпрограммы с именем *PK2*.

Входные параметры:

$$\begin{aligned}XA(j) &= X_A; YA(j) = Y_A; XC(j) = X_C; YC(j) = Y_C; \\L2(j) &= L_2; L3(j) = L_3; FX1(j) = \varphi_{X_1}; F31(j) = \varphi_{31}; \\VA(j) &= V_{\varphi_A}; VC(j) = V_{\varphi_C}; FVA(j) = \varphi_{V_A}; FVC(j) = \varphi_{V_C}; \\WA(j) &= W_{\varphi_A}; WC(j) = W_{\varphi_C}; FWA(j) = \varphi_{W_A}; FWC(j) = \varphi_{W_C}; \\F14(j) &= \varphi'_4; F24(j) = \varphi''_4; \\RD(j) &= R_D; BD(j) = \beta_D; RE(j) = R_E; BE(j) = \beta_E.\end{aligned}$$

Выходные параметры:

$$\begin{aligned}LM &= L_M; F2 = \varphi_2; F3 = \varphi_3; \\L1M &= L'_M; F12 = \varphi'_2; L2M = L''_M; F22 = \varphi''_2; \\VD &= V_{\varphi_D}; VE = V_{\varphi_E}; \\WD &= W_{\varphi_D}; WE = W_{\varphi_E}; \\FVD &= \varphi_{V_D}; FVE = \varphi_{V_E}; FWD = \varphi_{W_D}; FWE = \varphi_{W_E}.\end{aligned}$$

OAP.BAS	PK2
Sub pk2(lm As Double, f2 As Double, f3 As Double, l1m As Double, _ f12 As Double, l2m As Double, f22 As Double, vd As Double, _ ve As Double, wd As Double, we As Double, fvd As Double, _ fve As Double, fwd As Double, fwe As Double) Dim vb() As Double, fvb() As Double, arg As Double, Dim vbX As Double, vby As Double, vey As Double, vex As Double Dim vdy As Double, vdx As Double, fwb As Double, wb As Double Dim wbx As Double, wby As Double, wdy As Double, wdx As Double Dim wex As Double, wey As Double, dx As Double, dy As Double Dim x1c As Double, y1c As Double, y1b As Double, x1b As Double Dim x As Double, fm1 As Double, fm As Double, a1 As Double Dim a2 As Double, b1 As Double, B2 As Double, c1 As Double Dim c2 As Double, c3 As Double ReDim fvb(j): ReDim vb(j) dx = xc(j) - xa(j): dy = yc(j) - ya(j): x1c = dx * Cos(fx1(j)) + dy * Sin(fx1(j)) y1c = -dx * Sin(fx1(j)) + dy * Cos(fx1(j)): y1b = y1c + l3(j) * Sin(fx31(j)) x1b = Sqr(L2(j) ^ 2 - y1b ^ 2): lm = Abs(x1b - x1c): x = ((x1b - x1c) / lm)	

```

arg = (x1b - x1c) / lm:fm1 = FNARC(arg)
f2 = fx1(j) + Atn(y1b / x1b):fm = fm1 + fx1(j): f3 = f31(j) + fx1(j)
a1 = fm - fva(j): a2 = fm - fvc(j):b1 = fm - fwa(j): B2 = fm - fwc(j)
c1 = l3(j) * Sin(fm - f3): c2 = L2(j) * Sin(fm - f2):c3 = L2(j) * Cos(fm - f2)
f12 = (va(j) * Sin(a1) - vc(j) * Sin(a2) + lm * f14(j)) / c3
l1m = va(j) * Cos(a1) + c2 * f12 - vc(j) * Cos(a2)
b1 = fm - fwa(j): B2 = fm - fwc(j)
f22 = (wa(j) * Sin(b1) - c2 * f12 ^ 2 - wc(j) * Sin(B2) + _
      2 * l1m * f14(j) + lm * f24(j) + c1 * f14(j) ^ 2) / c3
l2m = wa(j) * Cos(b1) - c3 * f12 ^ 2 + c2 * f22 - _
      wc(j) * Cos(B2) + lm * f14(j) ^ 2 - c1 * f24(j)
vbx = va(j) * Cos(fva(j)) - L2(j) * f12 * Sin(f2)
vby = va(j) * Sin(fva(j)) + L2(j) * f12 * Cos(f2)
Call PMY(vbx, vby, vb(j), fvb(j))
vex = vb(j) * Cos(fvb(j)) - re(j) * f14(j) * Sin(f3 + be(j))
vey = vb(j) * Sin(fvb(j)) + re(j) * f14(j) * Cos(f3 + be(j))
Call PMY(vex, vey, ve, fve)
vdx = va(j) * Cos(fva(j)) - rd(j) * f12 * Sin(f2 + BD(j))
vdy = va(j) * Sin(fva(j)) + rd(j) * f12 * Cos(f2 + BD(j))
Call PMY(vdx, vdy, vd, fvd)
wbx = wa(j) * Cos(fwa(j)) - L2(j) * f22 * Sin(f2) - L2(j) * f12 ^ 2 * Cos(f2)
wby = wa(j) * Sin(fwa(j)) + L2(j) * f22 * Cos(f2) - L2(j) * f12 ^ 2 * Sin(f2)
Call PMY(wbx, wby, wb, fwb)
wdx = wa(j) * Cos(fwa(j)) - rd(j) * f22 * Sin(f2 + BD(j)) - _
      - rd(j) * f12 ^ 2 * Cos(f2 + BD(j))
wdy = wa(j) * Sin(fwa(j)) + rd(j) * f22 * Cos(f2 + BD(j)) - _
      rd(j) * f12 ^ 2 * Sin(f2 + BD(j))
Call PMY(wdx, wdy, wd, fwd)
wex = wb * Cos(fwb) - re(j) * f24(j) * Sin(f3 + be(j)) - _
      re(j) * f14(j) ^ 2 * Cos(f3 + be(j))
wey = wb * Sin(fwb) + re(j) * f24(j) * Cos(f3 + be(j)) - _
      re(j) * f14(j) ^ 2 * Sin(f3 + be(j))
Call PMY(wex, wey, we, fwe)
End Sub

```

1.4.4. Кинематический анализ группы Ассура II класса 3 вида

В общем виде группа Ассура II класса 3 вида показана на рис. 1.6.

Исходными данными для решения задачи кинематики являются:

- координаты точек *A* и *C* в системе *XOY*;

- модуль вектора \bar{L}_3 , направленного в сторону направляющей и перпендикулярного к ней;

- величины и направления аналогов линейных скоростей и ускорений точек A и C ;

- полярные координаты промежуточных точек D и E .

Исходные данные обозначим аналогично принятым в предыдущих группах Ассура.

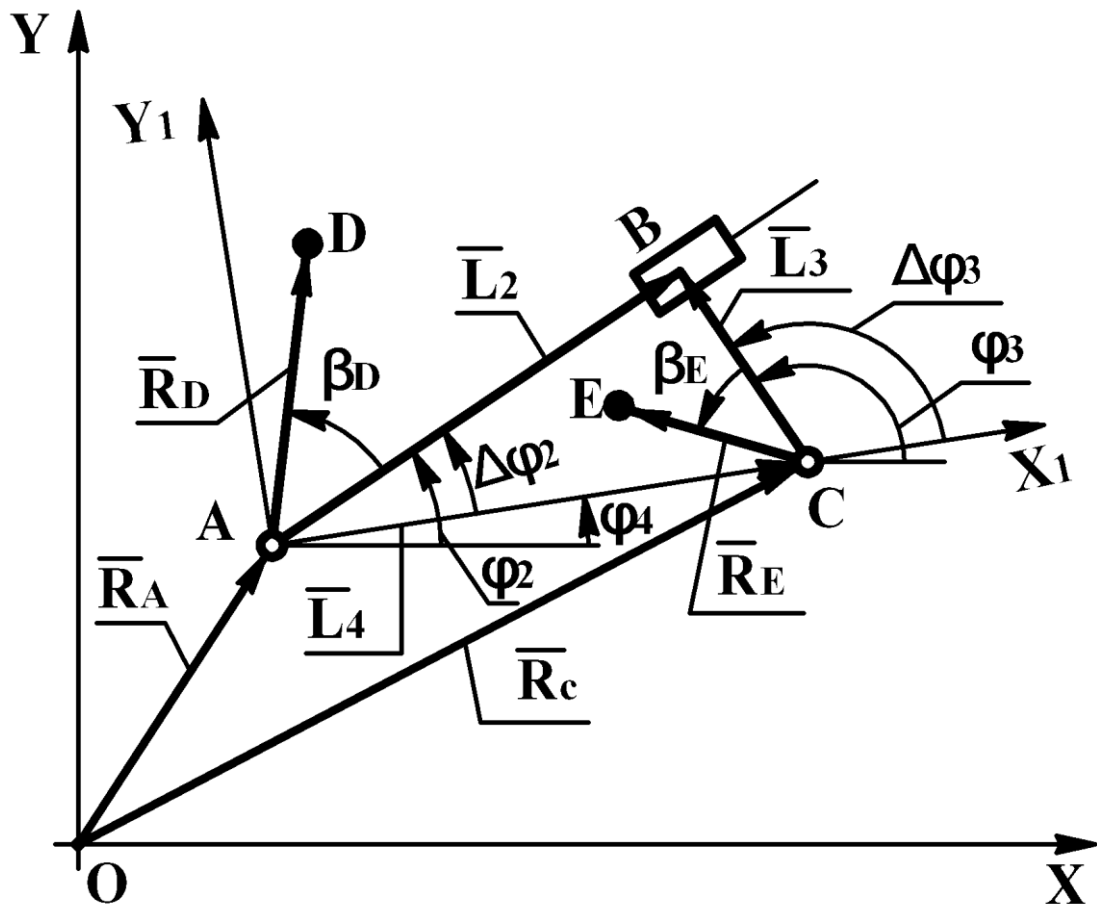


Рис. 1.6. Расчетная схема группы Ассура II класса 3 вида

Выходными параметрами являются:

- угловое перемещение φ_2 звена 2;
- модули и направления аналогов угловых скоростей и ускорений звена 2 (φ_2' , φ_2'');
- угловое перемещение φ_3 звена 3;
- модули аналогов линейных скоростей и ускорений ползуна 3 (L_3' , L_3'') и промежуточных точек D, E ($V_{\varphi D}$, $V_{\varphi E}$, $\varphi_{V D}$, $\varphi_{V E}$, $W_{\varphi D}$, $W_{\varphi E}$, $\varphi_{W D}$, $\varphi_{W E}$).

Введем подвижную систему координат X_1AY_1 .

Начало этой системы принимаем в шарнире A кулисы 2 и положительную ось AX_1 направляем через заданную точку C . В этом случае точка B может располагаться либо в I, либо в IV четверти подвижной системы координат. С помощью подпрограммы *PMY* определяем модуль вектора \bar{L}_4 и его направляющий угол φ_4 .

При рассмотрении контура $ABCA$ запишем

$$L_2 = \sqrt{(L_4)^2 - (L_3)^2}; \quad (1.95)$$

$$\Delta\varphi_2 = \arctg\left(\frac{z \cdot L_3}{L_2}\right). \quad (1.96)$$

Если точка B расположена:

- в I четверти системы координат X_1AY_1 , то $z = +1$;
- в IV четверти, то $z = -1$.

Вектор \bar{L}_2 координируется углом φ_2 , а вектор \bar{L}_3 – углом φ_3 :

$$\varphi_2 = \varphi_4 + \Delta\varphi_2; \quad (1.97)$$

$$\varphi_3 = \varphi_2 + 0,5\pi \cdot z. \quad (1.98)$$

При этом на основании рис. 1.6

$$\varphi'_3 = \varphi'_2; \quad (1.99)$$

$$\varphi''_3 = \varphi''_2. \quad (1.100)$$

Координаты промежуточных точек D и E определяются с помощью подпрограммы *PMY*.

Для определения аналогов скоростей запишем векторное уравнение

$$\bar{R}_A + \bar{L}_2 = \bar{R}_C + \bar{L}_3. \quad (1.101)$$

При дифференцировании уравнения (1.101) по обобщенной координате с учетом равенств (1.99) и (1.100) имеем

$$V_{\varphi_A} \cdot \bar{e}_{V_A} + L_2 \cdot \varphi'_2 \cdot \bar{e}'_2 = V_{\varphi_C} \cdot \bar{e}_{V_C} + L_3 \cdot \varphi'_3 \cdot \bar{e}'_3. \quad (1.102)$$

Умножим уравнение (1.102) на \bar{e}'_2 , преобразуем и найдем

$$\varphi'_2 = \frac{V_{\varphi_A} \sin(\varphi_2 - \varphi_{V_A}) - V_{\varphi_C} \sin(\varphi_2 - \varphi_{V_C})}{L_2}. \quad (1.103)$$

Умножим уравнение (1.102) на \bar{e}_2 , преобразуем и получим

$$L'_2 = -V_{\varphi_A} \cos(\varphi_2 - \varphi_{V_A}) + V_{\varphi_C} \sin(\varphi_2 - \varphi_{V_C}) - z \cdot L_3 \cdot \varphi'_2. \quad (1.104)$$

При дифференцировании уравнения (1.102) имеем

$$\begin{aligned} W_{\varphi_A} \cdot \bar{e}_{W_A} + L_2 \cdot \varphi''_2 \cdot \bar{e}'_2 + L_2(\varphi'_2)^2 \bar{e}''_2 + 2 \cdot L'_2 \cdot \varphi'_2 \cdot \bar{e}'_2 + L''_2 \cdot \bar{e}_2 = \\ = W_{\varphi_C} \cdot \bar{e}_{W_C} + L_3 \cdot \varphi''_3 \cdot \bar{e}'_3 + L_3(\varphi'_3)^2 \bar{e}''_3. \end{aligned} \quad (1.105)$$

После умножения выражения (1.105) на \bar{e}'_2 и преобразования

$$\varphi''_2 = \frac{1}{L_2} \left(W_{\varphi_A} \sin(\varphi_2 - \varphi_{W_A}) - 2 \cdot L'_2 \cdot \varphi'_2 - \right. \\ \left. - W_{\varphi_C} \sin(\varphi_2 - \varphi_{W_C}) - z \cdot L_3(\varphi'_2)^2 \right). \quad (1.106)$$

Умножим выражение (1.105) на \bar{e}_2 , преобразуем и найдем

$$\begin{aligned} L''_2 = -W_{\varphi_A} \cos(\varphi_2 - \varphi_{W_A}) + L_2(\varphi'_2)^2 + \\ + W_{\varphi_C} \sin(\varphi_2 - \varphi_{W_C}) + z \cdot L_3 \cdot \varphi''_2. \end{aligned} \quad (1.107)$$

Аналоги линейных скоростей и ускорений точек D и E определяются с помощью подпрограммы *PMY*.

Определение значений L_2 , φ_2 , φ_3 , L'_2 , φ'_2 , L''_2 и φ''_2 , а также кинематических параметров промежуточных точек D и E оформим в виде подпрограммы с именем *PK3* – подпрограммы кинематического анализа группы Ассура II класса 3 вида.

Входные параметры подпрограммы:

$$L3(j) = L_3; \quad XA(j) = X_A; \quad YA(j) = Y_A; \quad XC(j) = X_C; \quad YC(j) = Y_C;$$

$$VA(j) = V_{\varphi_A}; \quad VC(j) = V_{\varphi_C}; \quad FVA(j) = \varphi_{V_A}; \quad FVC(j) = \varphi_{V_C};$$

$$WA(j) = W_{\varphi_A}; \quad WC(j) = W_{\varphi_C}; \quad FWA(j) = \varphi_{W_A}; \quad FWC(j) = \varphi_{W_C}; \quad Z(j) = z;$$

$$RD(j) = r_D; \quad RD(j) = R_D; \quad BD(j) = \beta_D; \quad RE(j) = R_E; \quad BE(j) = \beta_E.$$

Выходные параметры:

$$L2 = L_2; \quad F2 = \varphi_2; \quad F12 = \varphi'_2; \quad L12 = L'_2; \quad F22 = \varphi''_2; \quad L22 = L''_2;$$

$$F3 = \varphi_3; \quad VD = V_{\varphi_D}; \quad VE = V_{\varphi_E}; \quad WD = W_{\varphi_D}; \quad WE = W_{\varphi_E};$$

$$FVD = \varphi_{V_D}; \quad FVE = \varphi_{V_E}; \quad FWD = \varphi_{W_D}; \quad FWE = \varphi_{W_E}.$$

OAP.BAS	PK3
Sub PK3(Z As Double, L2 As Double, F2 As Double, F3 As Double, F12 As Double, L12 As Double, F22 As Double, L22 As Double, VD As Double, VE As Double, WD As Double, WE As Double, FVD As Double, FVE As Double, FWD As Double, FWE As Double) Call PMY(xc(J) - XA(J), yc(J) - YA(J), l4, f4) L2 = Sqr(l4 ^ 2 - L3(J) ^ 2) F2 = f4 + Atn(Z * L3(J) / L2) F3 = 0.5 * PI * Z + F2 F12 = (va(J) * Sin(F2 - fva(J)) - vc(J) * Sin(F2 - fvc(J))) / L2 L12 = -va(J) * Cos(F2 - fva(J)) + vc(J) * Cos(F2 - fvc(J)) - Z * L3(J) * F12 F22 = (wa(J) * Sin(F2 - fwa(J)) - 2 * L12 * F12 - wc(J) * Sin(F2 - fwc(J)) - Z * L3(J) * F12 ^ 2) / L2 L22 = -wa(J) * Cos(F2 - fwa(J)) + L2 * F12 ^ 2 + wc(J) * Cos(F2 - fwc(J)) - Z * L3(J) * F22 vex = vc(J) * Cos(fvc(J)) - re(J) * F12 * Sin(F3 + be(J)) vey = vc(J) * Sin(fvc(J)) + re(J) * F12 * Cos(F3 + be(J)) Call PMY(vex, vey, VE, FVE) vdx = va(J) * Cos(fva(J)) - rd(J) * F12 * Sin(F2 + BD(J)) vdy = va(J) * Sin(fva(J)) + rd(J) * F12 * Cos(F2 + BD(J)) Call PMY(vdx, vdy, VD, FVD) wex = wc(J) * Cos(fwc(J)) - re(J) * F22 * Sin(F3 + be(J)) - re(J) * _ F12(J) ^ 2 * Cos(F3 + be(J)) wey = wc(J) * Sin(fwc(J)) + re(J) * F22 * Cos(F3 + be(J)) - re(J) * _ F12(J) ^ 2 * Sin(F3 + be(J)) Call PMY(wex, wey, WE, FWE) wdx = wa(J) * Cos(fwa(J)) - rd(J) * F22 * Sin(F2 + BD(J)) - rd(J) * F12 ^ 2 _ * Cos(F2 + BD(J)) wdy = wa(J) * Sin(fwa(J)) + rd(J) * F22 * Cos(F2 + BD(J)) - rd(J) * F12 ^ 2 _ * Sin(F2 + BD(J)) Call PMY(wdx, wdy, WD, FWD) End Sub	

1.4.5. Кинематический анализ группы Ассура II класса 4 вида

В общем виде группа Асура II класса 4 вида показана на рис. 1.7.

Исходными данными для решения задачи кинематики являются:

- координаты $X_A, Y_A, X_C, Y_C, \varphi_1, \varphi_4$ в системе XOY ;
- аналоги линейных скоростей и ускорений $V_{\varphi_A}, V_{\varphi_C}, W_{\varphi_A}, W_{\varphi_C}$;
- углы $\varphi_{V_A}, \varphi_{V_C}, \varphi_{W_A}, \varphi_{W_C}$, определяющие их направления;

- аналоги угловых скоростей и ускорений $\varphi'_1, \varphi''_1, \varphi'_4, \varphi''_4$ направляющих 1 и 4;
- длины звеньев L_2 и L_3 ;
- положения заданных (промежуточных) точек D и E (R_D, β_D и R_E, β_E) относительно звеньев (за начало векторов \bar{r}_1 принимается центр внутренней пары – точка B).

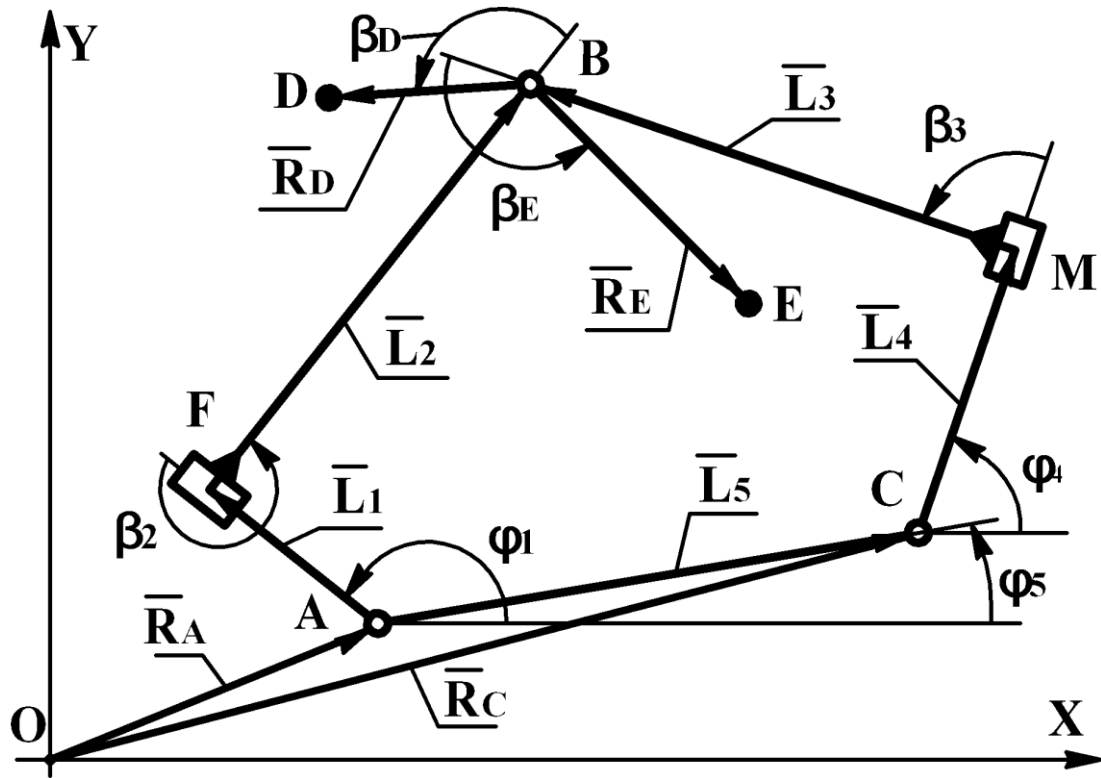


Рис. 1.7. Расчетная схема группы Ассура II класса 4 вида

Рассмотрим контур $ACBA$.

Условие замкнутости этого контура имеет вид

$$L_1 \cdot \bar{e}_1 + L_2 \cdot \bar{e}_2 = L_5 \cdot \bar{e}_5 + L_4 \cdot \bar{e}_4 + L_3 \cdot \bar{e}_3. \quad (1.108)$$

Модуль и направление вектора \bar{L}_5 определяются при расчетах в подпрограмме PMY .

Умножим уравнение (1.108) на единичный вектор \bar{e}'_1 .

При разрешении полученного результата относительно величины L_4 найдем следующее:

$$L_4 = \frac{L_5 \sin(\varphi_1 - \varphi_5) + L_3 \sin(\varphi_1 - \varphi_4 - \beta_3) + L_2 \sin(\beta_2)}{\sin(\varphi_4 - \varphi_1)}. \quad (1.109)$$

При проектировании контура *OCBO* на оси координат получим

$$\left. \begin{aligned} X_B &= X_C + L_4 \sin \varphi_4 + L_3 \sin(\varphi_3 + \beta_3) \\ Y_B &= Y_C + L_4 \sin \varphi_4 + L_3 \sin(\varphi_3 + \beta_3) \end{aligned} \right\}. \quad (1.110)$$

Умножаем (1.108) на \bar{e}'_4 и разрешаем результат относительно L_1 :

$$L_1 = \frac{L_5 \sin(\varphi_4 - \varphi_5) - L_2 \sin(\varphi_4 - \varphi_1 - \beta_2) - L_3 \sin(\beta_3)}{\sin(\varphi_1 - \varphi_4)}. \quad (1.111)$$

Рассмотрим контур *OAFBMSO*.

Условие замкнутости этого контура имеет вид

$$R_A \cdot \bar{e}_{R_A} + L_1 \cdot \bar{e}_1 + L_2 \cdot \bar{e}_2 = R_C \cdot \bar{e}_{R_C} + L_4 \cdot \bar{e}_4 + L_3 \cdot \bar{e}_3. \quad (1.112)$$

При дифференцировании уравнения (1.112) по обобщенной координате

$$\begin{aligned} V_{\varphi_A} \cdot \bar{e}_{V_A} + L'_1 \cdot \bar{e}_1 + L_1 \cdot \varphi'_1 \cdot \bar{e}'_1 + L_2 \cdot \varphi'_2 \cdot \bar{e}'_2 = \\ = V_{\varphi_C} \cdot \bar{e}_{V_C} + L'_4 \cdot \bar{e}_4 + L_4 \cdot \varphi'_4 \cdot \bar{e}'_4 + L_3 \cdot \varphi'_3 \cdot \bar{e}'_3. \end{aligned} \quad (1.113)$$

После умножения уравнения (1.113) на \bar{e}'_1 и с учетом, что звенья L_1 и L_2 перпендикулярны, найдем

$$L'_4 = \frac{\left(V_{\varphi_C} \sin(\varphi_1 - \varphi_{V_C}) - V_{\varphi_A} \cos(\varphi_1 - \varphi_{V_A}) + L_1 \cdot \varphi'_1 - \right.}{\sin(\varphi_4 - \varphi_1)} \left. - L_4 \cdot \varphi'_4 \cos(\varphi_1 - \varphi_4) + L_3 \cdot \varphi'_4 \cos(\varphi_1 - \varphi_4 - \beta_3) \right). \quad (1.114)$$

После умножения уравнения (1.113) на \bar{e}'_4 и с учетом, что звенья L_3 и L_4 перпендикулярны, найдем

$$L'_1 = \frac{\left(V_{\varphi_C} \sin(\varphi_1 - \varphi_{V_C}) - V_{\varphi_A} \cos(\varphi_1 - \varphi_{V_A}) + \varphi'_1 (L_1 \cos(\varphi_1 - \varphi_4)) + \right.}{\sin(\varphi_4 - \varphi_1)} \left. + L_2 \cos(\varphi_4 - \varphi_1 - \beta_2) + L_3 \cdot \varphi'_4 \cos(\varphi_1 - \varphi_4 - \beta_3) - L_4 \cdot \varphi'_4 \right). \quad (1.115)$$

В результате дифференцирования по обобщенной координате уравнения (1.113) имеем

$$\begin{aligned} W_{\varphi_A} \cdot \bar{e}_{W_A} + L''_1 \cdot \bar{e}_1 + 2 \cdot L'_1 \cdot \varphi'_1 \cdot \bar{e}'_1 + L_1 \cdot \varphi''_1 \cdot \bar{e}'_1 + L_1 (\varphi'_1)^2 \bar{e}''_1 + \\ + L_2 (\varphi'_1)^2 \bar{e}''_2 + L_2 \cdot \varphi''_1 \cdot \bar{e}'_2 = W_{\varphi_C} \cdot \bar{e}_{W_C} - L''_4 \cdot \bar{e}_4 + 2 \cdot L'_4 \cdot \varphi'_4 \cdot \bar{e}'_4 + \\ + L_4 \cdot \varphi''_4 \cdot \bar{e}'_4 + L_4 (\varphi'_4)^2 \bar{e}''_4 + L_3 \cdot \varphi''_4 \cdot \bar{e}'_3 + L_3 (\varphi'_4)^2 \bar{e}''_3. \end{aligned} \quad (1.116)$$

После умножения выражения (1.116) на \bar{e}_1' и преобразования

$$L_4'' = \frac{\left(W_{\varphi_A} \cos(\varphi_1 - \varphi_{W_A}) - W_{\varphi_C} \sin(\varphi_1 - \varphi_{W_C}) - \right. \\ \left. - 2 \cdot L_1' \cdot \varphi_1' + L_2 (\varphi_2')^2 \sin(\beta_2) + L_4 (\varphi_4')^2 \sin(\varphi_1 - \varphi_4) + \right. \\ \left. + (2 \cdot L_4' \cdot \varphi_4' + L_4 \cdot \varphi_4'') \cos(\varphi_1 - \varphi_4) + \right. \\ \left. + L_3 \cdot \varphi_4'' \cos(\varphi_1 - \varphi_4 - \beta_3) + L_3 (\varphi_4')^2 \sin(\varphi_1 - \varphi_4 - \beta_3) \right) \\ \sin(\varphi_4 - \varphi_1)}. \quad (1.117)$$

При дифференцировании выражения (1.110) получим

$$\left. \begin{aligned} X_B' &= X_C' + L_4' \cos \varphi_4 - L_4 \cdot \varphi_4' \sin \varphi_4 - L_3 \cdot \varphi_4' \sin(\varphi_4 + \beta_3), \\ Y_B' &= Y_C' + L_4' \sin \varphi_4 - L_4 \cdot \varphi_4' \cos \varphi_4 + L_3 \cdot \varphi_4' \cos(\varphi_4 + \beta_3). \end{aligned} \right\} \quad (1.118)$$

$$\left. \begin{aligned} X_C' &= V_{\varphi_C} \cos(\varphi_{V_C}), \\ Y_C' &= V_{\varphi_C} \sin(\varphi_{V_C}). \end{aligned} \right\} \quad (1.119)$$

При дифференцировании выражения (1.118) будем иметь

$$\left. \begin{aligned} X_B'' &= X_C'' + L_4'' \cos \varphi_4 - 2 \cdot L_4' \cdot \varphi_4' \sin \varphi_4 - L_4 \cdot \varphi_4'' \sin \varphi_4 - \\ &- L_4 (\varphi_4')^2 \cos \varphi_4 - L_3 \cdot \varphi_4'' \sin(\varphi_4 + \beta_3) - L_3 (\varphi_4')^2 \cos(\varphi_4 + \beta_3), \\ Y_B'' &= Y_C'' + L_4'' \sin \varphi_4 + 2 \cdot L_4' \cdot \varphi_4' \cos \varphi_4 + L_4 \cdot \varphi_4'' \cos \varphi_4 - \\ &- L_4 (\varphi_4')^2 \sin \varphi_4 + L_3 \cdot \varphi_4'' \cos(\varphi_4 + \beta_3) - L_3 (\varphi_4')^2 \sin(\varphi_4 + \beta_3). \end{aligned} \right\} \quad (1.120)$$

$$\left. \begin{aligned} X_C'' &= W_{\varphi_C} \cos(\varphi_{W_C}) \\ Y_C'' &= W_{\varphi_C} \sin(\varphi_{W_C}) \end{aligned} \right\}. \quad (1.121)$$

Кинематические параметры точек B , D и E можно определить с помощью подпрограмм PKT , PMY и PA .

Определение значений кинематических параметров точек и звеньев группы Ассура II класса 4 вида осуществим в подпрограмме кинематического анализа с именем $PK4$.

Входные параметры подпрограммы:

$$XA(j) = X_A; YA(j) = Y_A; XC(j) = X_C; YC(j) = Y_C;$$

$$L2(j) = L_2; L3(j) = L_3; VA(j) = V_{\varphi_A}; VC(j) = V_{\varphi_C};$$

$$F1(j) = \varphi_1; F4(j) = \varphi_4; F14(j) = \varphi_4'; F24(j) = \varphi_4''; F11(j) = \varphi_1';$$

$$F21(j) = \varphi_1''; FVA(j) = \varphi_{vA}; FVC(j) = \varphi_{vC}; WA(j) = W_{\varphi_A};$$

$$WC(j) = W_{\varphi C}; FWA(j) = \varphi_{wA}; FWC(j) = \varphi_{wC};$$

$$B2(j) = \beta_2; B3(j) = \beta_3;$$

$$RD(j) = r_D; BD(j) = \beta_D; RE(j) = r_E; BE(j) = \beta_E.$$

Выходные параметры:

$$L1 = L_1; L11 = L'_1; L12 = L''_1; L4 = L_4; L14 = L'_4; L24 = L''_4;$$

$$XB = X_B; YB = Y_B; VD = V_{\varphi D}; VE = V_{\varphi E};$$

$$WD = W_{\varphi D}; WE = W_{\varphi E}; FVD = \varphi_{V_D}; FVE = \varphi_{V_E};$$

$$FWD = \varphi_{W_D}; FWE = \varphi_{W_E}.$$

OAP.BAS	PK4
SUB PK4 (L1 as Double, L11 as Double, L12 as Double, L4 as Double, L14 as Double, L24 as Double, XB as Double, YB as Double, VD as Double, VE as Double, WD as Double, WE as Double, FVD as Double, FVE as Double, FWD as Double, FEW as Double) CALL PMY (XC(J)-XA(J), YC(J)-YA(J), L5, FL5) L4=(L5*SIN(F1(J)-F5)+L3(J)*SIN(F1(J)-F4(J)-B3(J))+L2(J)*SIN(B2(J)))/_ SIN(F1(J)-F4(J)) L1=(L5*SIN(F1(J)-F5)-L2(J)*SIN(F4(J)-F1(J)-B2(J))-L3(J)*SIN(B2(J)))/_ SIN(F1(J)-F1(J)) L14=(VC(J)*SIN(F1(J)-FVC(J))-VA(J)*SIN(F1(J)-FVA(J))+F11(J)*L1-_ F14(J)**L4*COS(F1(J)-F4(J))+L3(J)*F14(J)*COS(F1(J)-F4(J)-B3(J)))/_ SIN(F4(J)-F1(J)) L11=(VC(J)*SIN(F4(J)-FVC(J))-VA(J)*SIN(F4(J)-FVA(J))+F11(J)*_ (L1*COS(F4(J)-F1(J))+L2(J)*COS(F4(J)-F1(J)-B2(J)))-L4*F14(J)/_ SIN(F4(J)-F1(J)) L24=(WA(J)*SIN(F1(J)-FWA(J))-WC(J)*SIN(F1(J)-FWC(J))-2*L11*_ F11(J)-L1*F21(J)+L2(J)*F21^2*SIN(B2(J))+(2*L14*F14(J)+L4*F24(J))*_ COS(F1(J)-F4(J)-B3(J))+L3(J)*F14(J)^2*SIN(F1(J)-F4(J)-B3(J)))/_ SIN(F1(J)-F4(J)) XB = XC(J)+L4*COS(F4(J))+L3(J)*COS(F4(J)+B3(J)) YB = YC(J)+L4*SIN(F4(J))+L3(J)*SIN(F4(J)+B3(J)) X1C = VC(J)*COS(FVC(J)): Y1C = VC(J)*SIN(FVC(J)) X2C = WC(J)*COS(FWC(J)): Y2C = WC(J)*SIN(FWC(J)) X1B=X1C+L14*COS(F4(J))-L4*F14(J)*SIN(F4(J))-L3(J)*F14(J)*_ SIN(F4(J)+ B3(J)) Y1B=Y1C+L14*SIN(F4(J))-L4*F14(J)*COS(F4(J))-L3(J)*F14(J)*_ COS(F4(J)+B3(J)) X2B=X2C+L24*COS(F4(J))-2*L14*F14(J)*SIN(F4(J))-L4*F24(J)*_ SIN(F4(J))-L4*F14(J)^2*COS(F4(J))-L3(J)*F24(J)*SIN(F4(J)+_ B3(J))-L3(J)*F14(J)^2*COS(F4(J)+B3(J))	

```

X2B=X2C+L24*SIN(F4(J))-2*L14*F14(J)*COS(F4(J))-L4*F24(J)*_
COS(F4(J))-L4*F14(J)^2*SIN(F4(J))-L3(J)*F24(J)*COS(F4(J)+_
B3(J))-L3(J)*F14(J)^2*SIN(F4(J)+B3(J))
CALL PMY (X1B, Y1B, VB, FVB):CALL PMY (X2B, Y2B, WB, FWB)
VEX=VB(J)*COS(FVB)-RE(J)*F14(J)*SIN(F4(J)+B3(J)+BE(J))
VEY=VB(J)*SIN(FVB)+RE(J)*F14(J)*COS(F4(J)+B3(J)+BE(J))
CALL PMY (VEX, VEY, VE, FVE)
VDX= VB(J)*COS(FVB)-RD(J)*F12(J)*SIN(F1(J)+B2(J)+BD(J))
VDY=VB(J)*SIN(FVB)+RD(J)*F12(J)*COS(F1(J)+B2(J)+BD(J))
CALL PMY (VDX, VDY, VD, FVD)
WEX=WB*COS(FWB)-RE(J)*F24(J)*SIN(F4(J)+B3(J)+BE(J))-RE(J)* _
F14(J)^2*COS(F4(J)+B3(J)+BE(J))
WEY=WB*SIN(FWB)+RE(J)*F24(J)*COS(F4(J)+B3(J)+BE(J))-RE(J)* _
F14(J)^2*SIN(F4(J)+B3(J)+BE(J))
CALL PMY (WEX, WEY, WE, FWE)
WDX=WB(J)*COS(FWB(J))-RD(J)*F21(J)*SIN(F2(J)+B2(J)+BD(J))-_
RD(J)* F21(J)^2*COS(F1(J)+B2(J)+BD(J))
WDY=WB(J)*SIN(FWB(J))+RD(J)*F21(J)*COS(F2(J)+B2(J)+BD(J))-_
RD(J)* F21(J)^2*SIN(F1(J)+B2(J)+BD(J))
CALL PMY (WDX, WDY, WD, FWD)
END SUB

```

1.4.6. Кинематический анализ группы Ассура II класса 5 вида

В общем виде группа Ассура II класса 5 вида показана на рис. 1.8, на котором A – произвольная точка, принадлежащая направляющей.

Исходными данными для решения задачи кинематики являются:

- угол φ_1 , определяющий направление вектора \bar{L}_1 ;
- координаты X_A, Y_A, X_C, Y_C соответствующих точек;
- аналоги линейных скоростей и ускорений $V_{\varphi A}, V_{\varphi C}, W_{\varphi A}, W_{\varphi C}$;
- их направляющие углы $\varphi_{VA}, \varphi_{VC}, \varphi_{WA}, \varphi_{WC}$;
- аналоги угловых скоростей и ускорений направляющей 1, т. е. φ'_1, φ''_1 ;
- геометрические параметры звеньев 2 и 3, т. е. L_2, β_2 и β_3 ;
- полярные координаты заданных точек D и E , т. е. R_D, β_D и R_E, β_E .

Искомые величинами являются:

- аналоги скоростей и ускорений точек D и E , т. е. $V_{\varphi D}$ и $V_{\varphi E}, W_{\varphi D}$ и $W_{\varphi E}$;
- их направляющие углы φ_{VD} и $\varphi_{VE}, \varphi_{WD}$ и φ_{WE} .

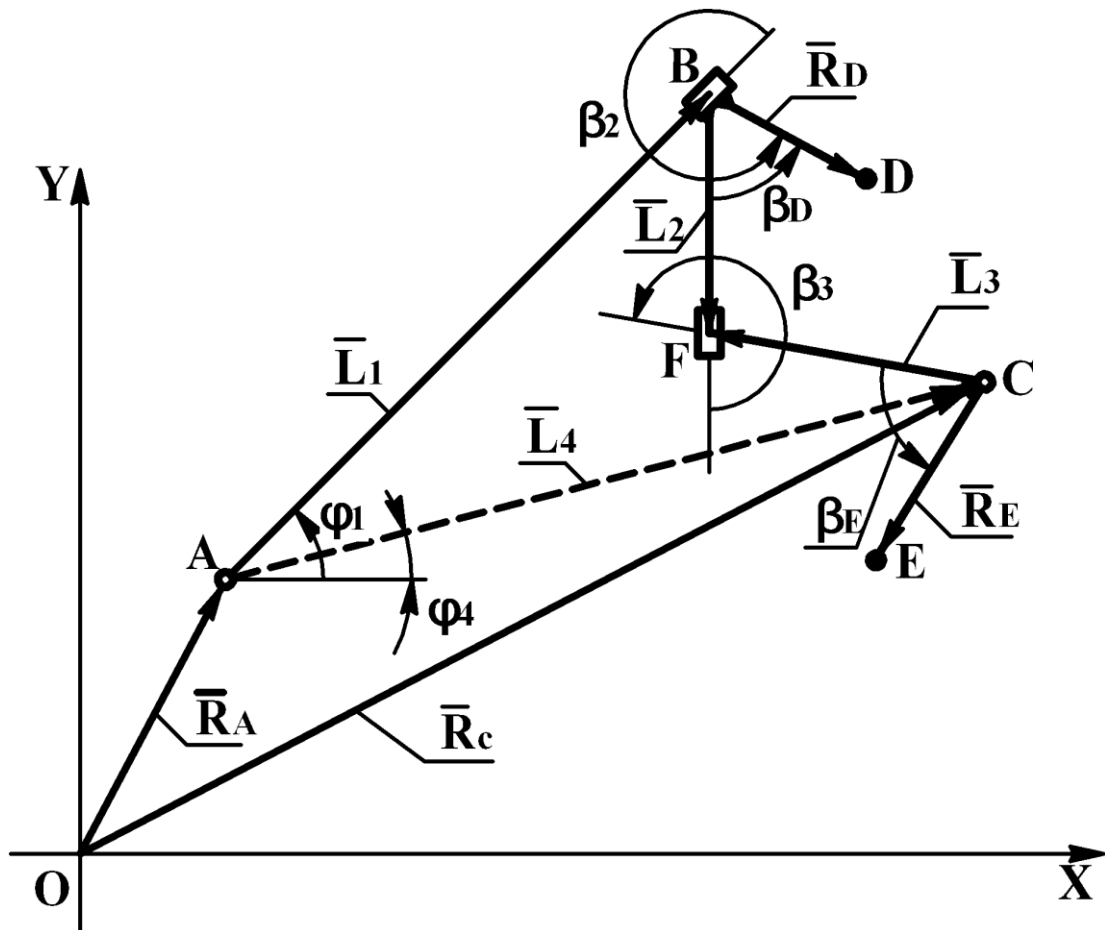


Рис. 1.8. Расчетная схема группы Ассура II класса 5 вида

Векторы \bar{L}_2 и \bar{L}_3 направляем к внутренней паре F и координируем их углами β_2 и β_3 , отсчитываемыми в положительном направлении от осей соответствующих направляющих.

Рассмотрим контур $ACBA$, условие замкнутости которого имеет вид

$$L_1 \cdot \bar{e}_1 + L_2 \cdot \bar{e}_2 = L_4 \cdot \bar{e}_4 + L_3 \cdot \bar{e}_3. \quad (1.122)$$

Модуль и направление вектора \bar{L}_4 определяется с помощью подпрограммы PMY .

Умножим уравнение (1.122) на единичный вектор \bar{e}_1' и при разрешении результата относительно величины L_2 найдем следующее:

$$L_2 = \frac{L_3 \sin(\beta_2 + \beta_3) - L_4 \sin(\varphi_1 - \varphi_4)}{\sin(\beta_2)}. \quad (1.123)$$

Умножим выражение (1.122) на \bar{e}_2' , разрешим результат относительно L_1 и будем иметь

$$L_1 = \frac{L_4 \sin(\varphi_1 + \beta_2 - \varphi_4) - L_3 \sin(\beta_3)}{\sin(\beta_2)}. \quad (1.124)$$

Координаты точки B при необходимости определяются с помощью подпрограммы *PKT*. Для ее применения имеются исходные параметры φ_1 и L_1 .

Рассмотрим контур $OABFCO$. Его уравнение замкнутости имеет вид

$$\bar{R}_A + \bar{L}_1 + \bar{L}_2 = \bar{R}_C + \bar{L}_3. \quad (1.125)$$

Дифференцированием уравнения (1.125) по обобщенной координате получим

$$V_{\varphi_A} \cdot \bar{e}_{V_A} + L_1' \cdot \bar{e}_1 + L_1 \cdot \varphi_1' \cdot \bar{e}_1' + L_2 \cdot \varphi_2' \cdot \bar{e}_2' = V_{\varphi_C} \cdot \bar{e}_{V_C} + L_3 \cdot \varphi_3' \cdot \bar{e}_3'. \quad (1.126)$$

После умножения выражения (1.126) на \bar{e}_1' запишем

$$L_2' = \frac{\left(V_{\varphi_A} \cos(\varphi_1 - \varphi_{V_A}) - V_{\varphi_C} \sin(\varphi_1 - \varphi_{V_C}) + \right.}{\sin \beta_2} \left. \varphi_1' (L_3 \cos(\beta_2 + \beta_3) - L_1 - L_2 \cos \beta_2) \right). \quad (1.127)$$

После умножения выражения (1.127) на \bar{e}_2' получим

$$L_1' = \frac{\left(V_{\varphi_C} \sin(\varphi_1 + \beta_2 - \varphi_{V_C}) - V_{\varphi_A} \sin(\varphi_1 + \beta_2 - \varphi_{V_A}) + \right.}{\sin \beta_2} \left. + \varphi_1' (L_1 \cos \beta_2 + L_2 - L_3 \cos \beta_3) \right). \quad (1.128)$$

При дифференцировании уравнения (1.126) по обобщенной координате

$$W_{\varphi_A} \cdot \bar{e}_{W_A} + L_1 \cdot \varphi_1'' \cdot \bar{e}_1' + 2 \cdot L_1' \cdot \varphi_1' \cdot \bar{e}_1' + L_1'' \cdot \bar{e}_1 + L_1 (\varphi_1')^2 \bar{e}_1'' + L_2'' \cdot \bar{e}_2 + \\ + 2 \cdot L_2' \cdot \varphi_2' \cdot \bar{e}_2' + L_2 \cdot \varphi_2'' \cdot \bar{e}_2' = W_{\varphi_C} \cdot \bar{e}_{W_C} + L_3' \cdot \varphi_3' \cdot \bar{e}_3' + L_3 \cdot (\varphi_3')^2 \bar{e}_3''. \quad (1.129)$$

После умножения выражения (1.29) на \bar{e}_2' и преобразования найдем

$$L_1'' = \frac{1}{\sin \beta_2} \left(W_{\varphi_C} \sin(\varphi_1 + \beta_2 - \varphi_{W_C}) - W_{\varphi_A} \cos(\varphi_1 + \beta_2 - \varphi_{W_A}) + \right. \\ \left. + 2\varphi_1' (L_1' \cos \beta_2 + L_2') + \varphi_1'' (L_2 + L_1 \cos \beta_2 - L_3 \cos \beta_3) + \right. \\ \left. + (\varphi_1')^2 (L_1 \sin \beta_2 + L_3 \sin \beta_3) \right). \quad (1.130)$$

Рассмотрим замкнутый контур $OABDO$. Его уравнение замкнутости имеет вид

$$\bar{R}_A + \bar{L}_1 + \bar{R}_D = \bar{X}_D + \bar{Y}_D. \quad (1.131)$$

Направление вектора \bar{R}_D , входящего в уравнение (1.131), определяется углом φ_D :

$$\varphi_D = \varphi_1 + \beta_2 + \beta_D. \quad (1.132)$$

При дифференцировании по обобщенной координате выражения (1.131) с учетом равенства (1.132) после умножения на орты \bar{i} и \bar{j} осей системы координат получим следующие соответствующие проекции:

$$\left. \begin{aligned} X'_D &= V_A \cos(\varphi_{V_A}) + L'_1 \cos \varphi_1 - L_1 \cdot \varphi'_1 \cdot \sin \varphi_1 - \\ &- R_D \cdot \varphi'_1 \cdot \sin(\varphi_1 + \beta_2 + \beta_D), \\ Y'_D &= V_A \sin(\varphi_{V_A}) + L'_1 \sin \varphi_1 + L_1 \cdot \varphi'_1 \cdot \cos \varphi_1 + \\ &+ R_D \cdot \varphi'_1 \cdot \cos(\varphi_1 + \beta_2 + \beta_D). \end{aligned} \right\} \quad (1.133)$$

Для определения проекций аналога линейного ускорения искомой точки на оси системы координат продифференцируем систему (1.133) по обобщенной координате:

$$\left. \begin{aligned} X''_D &= W\varphi_A \cos(\varphi_{W_A}) + L''_1 \cos \varphi_1 - 2L'_1 \cdot \varphi'_1 \sin \varphi_1 - L_1 \cdot \varphi''_1 \sin \varphi_1 - \\ &- L_1 (\varphi'_1)^2 \cos \varphi_1 - R_D \cdot \varphi''_1 \sin(\varphi_1 + \beta_2 + \beta_D) - \\ &- R_D (\varphi'_1)^2 \cos(\varphi_1 + \beta_2 + \beta_D), \\ Y''_D &= W\varphi_A \sin(\varphi_{W_A}) + L''_1 \sin \varphi_1 + 2L'_1 \cdot \varphi'_1 \cos \varphi_1 + \\ &+ L_1 \cdot \varphi''_1 \cos \varphi_1 - L_1 (\varphi'_1)^2 \sin \varphi_1 + R_D \varphi''_1 \cos(\varphi_1 + \beta_2 + \beta_D) - \\ &- R_D (\varphi'_1)^2 \sin(\varphi_1 + \beta_2 + \beta_D). \end{aligned} \right\} \quad (1.134)$$

Кинематические параметры точек D и E определяются с помощью подпрограмм PKT , PMY и PA .

Определение значений V_{φ_D} , V_{φ_E} , W_{φ_D} , W_{φ_E} , φ_{V_D} , φ_{V_E} , φ_{W_D} , φ_{W_E} произведем в подпрограмме кинематического анализа группы Ассура II класса 5 вида с именем $PK5$.

Входные параметры подпрограммы:

$$XA(j) = X_A; YA(j) = Y_A; XC(j) = X_C; YC(j) = Y_C;$$

$$L3(j) = L_3; VA(j) = V_{\varphi_A}; VC(j) = V_{\varphi_C};$$

$$F1(j) = \varphi_1; F11(j) = \varphi'_1; F12(j) = \varphi''_1;$$

$$FVA(j) = \varphi_{vA}; FVC(j) = \varphi_{vC}; WA(j) = W_{\varphi A}; WC(j) = W_{\varphi C};$$

$$FWA(j) = \varphi_{wA}; FWC(j) = \varphi_{wC};$$

$$B2(j) = \beta_2; B3(j) = \beta_3; RD(j) = r_D; BD(j) = \beta_D;$$

$$RE(j) = r_E; BE(j) = \beta_E.$$

Выходные параметры:

$$VD = V_{\varphi D}; VE = V_{\varphi E}; WD = W_{\varphi D}; WE = W_{\varphi E};$$

$$FVD = \varphi_{V_D}; FVE = \varphi_{V_E}; FWD = \varphi_{W_D}; FWE = \varphi_{W_E}.$$

OAP.BAS	PK5
SUB PK5 (VD as Double, FVD as Double, WD as Double, FWD as Double, VE as Double, FVE as Double, WE as Double, FEW as Double) CALL PMY (XC(J)-XA(J),YC(J)-YA(J), L4, F4) L2=(L3(J)*SIN(B2(J)+B3(J))-L4*SIN(F1(J)-F4))/SIN(B2(J)) L1=(L4*SIN(F1(J)+B2(J)-F4)-L3(J)*SIN(B3(J)))/SIN(B2(J)) L11=(VC(J)*SIN(F1(J)+B2(J)-FVC(J))-VA(J)*SIN(F1(J)+B2(J)-_ FVA(J))+F11(J)*(L1*COS(B2(J))+L2-L3(J)*COS(B3(J)))/SIN(B2(J)) L12=(VA(J)*SIN(F1(J)-FVA(J))-VC(J)*SIN(F1(J)-FVC(J))+_ F11(J)*(L3(J)*COS(B2(J)+B3(J))-L1-L2*COS(B2(J)))/SIN(B2(J)) L21=(WC(J)*SIN(F1(J)+B2(J)-FWC(J))-WA(J)*SIN(F1(J)+B2(J)-_ FWA(J))+2*F11(J)*(L11*COS(B2(J))+L12)+F12(J)*(L2+L1*_ COS(B2(J)-L3(J)*COS(B3(J)))+F11(J)^2*(L1*SIN(B2(J))+_ L3(J)*SIN(B3(J)))/SIN(B2(J)) X1D=VA(J)*COS(FWA(J))+L11*COS(F1(J))-L1*F11(J)*SIN(F1(J))-_ RD*F11(J)*SIN(F1(J)+B2(J)+BD(J)) Y1D=VA(J)*SIN(FWA(J))+L11*SIN(F1(J))+L1*F11(J)*COS(F1(J))+_ RD*F11(J)*COS(F1(J)+B2(J)+BD(J)) X2D=WA(J)*COS(FWA(J))+L21*COS(F1(J))-2*L11*F11(J)*_ SIN(F1(J))-L1*F12(J)*SIN(F1(J))-L1*F11(J)^2*COS(F1(J))-RD(J)*_ F12(J)*SIN(F1(J)+B2(J)+BD(J))-RD(J)*F11(J)^2*COS(F1(J)+B2(J)+BD(J)) X2D=WA(J)*SIN(FWA(J))+L21*SIN(F1(J))+2*L11*F11(J)*_ COS(F1(J))+L1*F12(J)*COS(F1(J))-1*F11(J)^2*SIN(F1(J))+RD(J)*_ *F12(J)*COS(F1(J)+B2(J)+BD(J))-RD(J)*F11(J)^2*_ SIN(F1(J)+B2(J)+BD(J)) CALL PMY (X1D, Y1D, VD, FVD) CALL PMY (X2D, Y2D, WD, FWD) FL3 = F1(J)+B2(J)+B3(J) CALL PMY (XC(J), YC(J), RE(J), FL3, BE(J), XE, YE) CALL PA(VC(J),FVC(J),WC(J),FWC(J),RE(J),BE(J),FL3,F11(J),F12(J),_ VE,FVE,WE, FWE) END SUB	

**1.4.7. Контрольные вопросы по кинематическому анализу
рычажных механизмов**

1. Каково назначение подпрограмм *PKT*, *PMY* и *PA*? Опишите входные и выходные параметры этих подпрограмм.
2. Каково назначение подпрограммы *PK1*? Опишите входные и выходные параметры этой подпрограммы.
3. Что означает понятие «параметр группы Ассура II класса 1 вида»? Как он обозначается? Какие у него бывают численные значения?
4. Каково назначение подпрограммы *PK2*? Опишите входные и выходные параметры этой подпрограммы.
5. Что означают параметры φ_{x_1} и φ_{z1} ? Какие у них бывают численные значения?
6. Каким образом определяются параметры φ_4 , φ'_4 , и φ''_4 ?
7. Каково назначение подпрограммы *PK3*? Опишите входные и выходные параметры этой подпрограммы.
8. Каким образом определяются параметры L_2 и $\Delta\varphi_2$?
9. Что означает параметр z ? Какие возможны значения у параметра z ?
10. Каково назначение подпрограммы *PK4*? Опишите входные и выходные параметры этой подпрограммы.
11. Каким образом определяются параметры φ_1 и φ_4 ?
12. Что означают параметры β_2 и β_3 ?
13. Каково назначение подпрограммы *PK5*? Опишите входные и выходные параметры этой подпрограммы.
14. Каким образом определяются параметры β_2 и L_1 ?
15. Какие промежуточные точки звеньев используются при анализе данной группы? Какими параметрами они задаются?

1.5. Силовой анализ рычажных механизмов

Движение звеньев механизма, входящего в состав машинного агрегата, происходит под действием различных по своей природе сил, которые обуславливают определенные перемещения, скорости и ускорения звеньев. Это движущие силы, силы сопротивления, силы тяжести и др. Характер их действия может быть разным: некоторые из них постоянны, другие зависят от положения звеньев механизма или от их скорости [5].

Своим действием приложенные силы сообщают механизму тот или иной закон движения. Кинематические характеристики – перемещение, скорость, ускорение, время срабатывания, коэффициент неравномерности и др. – определяются посредством решения уравнения движения. Выбор способа решения уравнения движения зависит от характера действия заданных сил и от передаточных свойств механизма. При этом размеры, массы и моменты инерции масс звеньев должны быть известными. Однако распространена и обратная задача, когда заданы кинематические характеристики режима движения машины и необходимо найти массы, моменты инерции, а следовательно, и размеры звеньев, при которых механизм, нагруженный заданными силами, двигался бы в требуемом режиме.

Установление общих зависимостей между силами, действующими на звенья механизма, с одной стороны, и параметрами кинематики (обобщенными координатами, скоростями и ускорениями) этого механизма, с другой, составляет главную цель динамического анализа механизмов и машин.

Динамический анализ механизма состоит из двух основных задач [6]:

- первая задача – определение внешних неизвестных сил, действующих на звенья механизма, а также усилий и реакций, возникающих в кинематических парах при заданном законе движения механизма. На решении этой задачи основывается силовой, или кинетостатический, расчет механизмов;

- вторая задача – определение закона движения механизма под действием приложенных к нему сил.

В данном подразделе рассматриваются способы решения первой (прямой) задачи динамического анализа механизма машинного агрегата. При этом необходимо подчеркнуть, что все звенья механизма являются абсолютно жесткими. В этой связи следует отметить, что в классической теории механизмов и машин рассмотрены механизмы с жесткими звеньями, обладающие одной степенью свободы. Такие механизмы имеют преимущественное распространение и в настоящее время. Основные уравнения движения этих механизмов в конечной и дифференциальной форме вытекают из теоремы об изменении кинетической энергии. Эта теорема наряду с принципом Даламбера и дает возможность решать широкий круг задач, которые ставит перед теорией механизмов и машин современное машиностроение. Однако в современной технике получают все большее распространение сложные многоприводные механизмы с несколькими степенями свободы. Кроме того, необходимый в ряде случаев расчет механизмов с одной степенью свободы, обусловленный (в частности) быстроходностью современных машин, требует учета упругих свойств звеньев этих механизмов и зазоров в их кинематических парах. В таких случаях механизм с одной степенью свободы превращается в механическую систему, обладающую несколькими степенями свободы.

Все существующие методы определения силовых (кинетостатических) параметров механизмов делятся на графоаналитические (метод построения планов сил) и аналитические. Решение задач силового анализа связано с большим объемом вычислительного процесса, поэтому автоматизация данных расчетов позволит значительно сократить как время самого расчета, так и сроки процесса проектирования механизма в целом.

В данном подразделе рассматривается силовой анализ групп Ассура II класса пяти видов, приводятся подпрограммы определения главного вектора сил и главного момента сил, вычисления касательных составляющих реакций и силового анализа групп Ассура. Использование данных подпрограмм силового расчета значительно сократит время на составление основной программы силового анализа любого рычажного механизма.

1.5.1. Математическое моделирование силового анализа отдельного звена

В общем случае на каждое звено группы (рис. 1.9) могут действовать n внешних сил \bar{P}_i , суммарный внешний момент \bar{M}_B , сила тяжести \bar{G} , момент \bar{M}_{II} сил инерции и проекции \bar{P}_{IIx} , \bar{P}_{IIy} силы инерции. Точки приложения внешних сил \bar{P}_i задаются полярными координатами R_i и β_i относительно вектора \bar{L} звена.

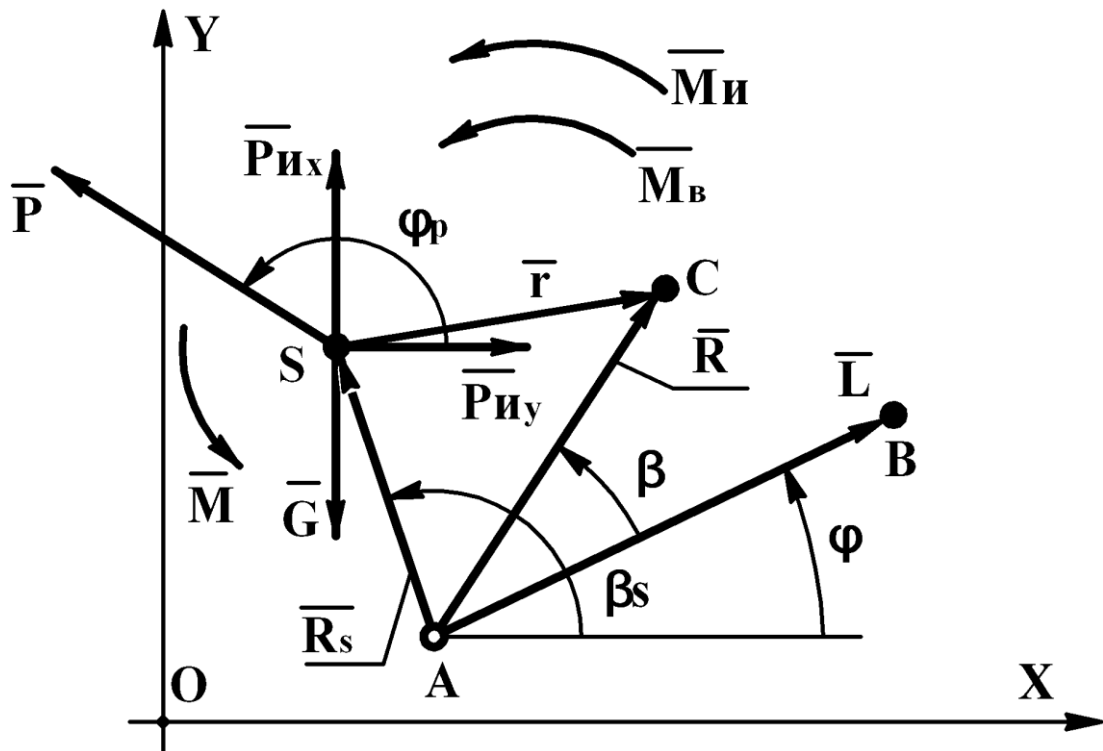


Рис. 1.9. Расчетная схема силового анализа отдельного звена

С целью упрощения силового расчета целесообразно систему сил и моментов, действующих на звено, заменить главным вектором сил \bar{P} и главным моментом \bar{M} , приняв за точку приведения этой системы сил точку S – центр масс звена, определяемый полярными координатами R_S и β_S .

Для определения величин \bar{P} и \bar{M} запишем следующие выражения:

$$M \cdot \bar{k} = M_{II} \cdot \bar{k} + M_B \cdot \bar{k} + \sum_i^n (\bar{r}_i \times \bar{P}_i); \quad (1.135)$$

$$M_{II} = -J_S \cdot \varphi'' \cdot \omega^2, \quad (1.136)$$

где J_S – момент инерции массы звена относительно оси, проходящей через точку центра масс;

φ'' – аналог углового ускорения звена;

ω – обобщенная скорость;

$$\bar{P} = \bar{G} + \bar{P}_{H_X} + \bar{P}_{H_Y} + \sum_i^m \bar{P}_i . \quad (1.137)$$

В выражении (1.137)

$$\left. \begin{aligned} P_{H_X} &= x_S'' m \omega^2, \\ P_{H_Y} &= y_S'' m \omega^2, \end{aligned} \right\} \quad (1.138)$$

где x_S'' , y_S'' – проекции аналога линейного ускорения центра масс S звена;

m – масса звена.

На основании рис. 1.9

$$\bar{r}_i = \bar{R}_i - \bar{R}_S . \quad (1.139)$$

Проекцией уравнения (1.135) с учетом выражений (1.136), (1.139) получим

$$\begin{aligned} M &= -J_S \cdot \varphi'' \cdot \omega^2 + M_B + \\ &+ \sum_i^n (R_i \cdot P_i \sin(\varphi_{P_i} - \varphi - \beta_i) - R_S \cdot P_i \sin(\varphi_{P_i} - \varphi - \beta_S)) . \end{aligned} \quad (1.140)$$

Проекцией уравнения (1.137) на оси координат с учетом выражения (1.138) получим

$$\left. \begin{aligned} P_X &= -x_S'' \cdot m \cdot \omega^2 + \sum_i P_i \cos(\varphi_{P_i}), \\ P_Y &= -G - y_S'' \cdot m \cdot \omega^2 + \sum_i P_i \sin(\varphi_{P_i}). \end{aligned} \right\} \quad (1.141)$$

С помощью подпрограммы *PMY* находим модуль вектора \bar{P} и его направляющий угол φ_P .

Таким образом, целесообразно разработать подпрограмму определения главного вектора сил и главного момента сил, предназначенную для вычислений по формулам (1.140) и (1.141). Назовем ее *PBC*.

Входные параметры:

$$N = n; F = \varphi; F2 = \varphi''; WS = W_S; FWS = \varphi_{W_S};$$

$$MB = M_B; RS = R_S; BS = \beta_S; JS = J_S; MS = m.$$

Выходные параметры:

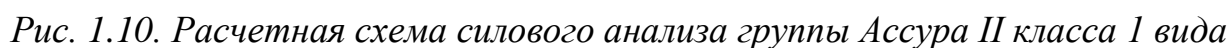
$$Q = P; FQ = \varphi_P; M = m.$$

OAP.BAS	PBC
<pre> Sub PBC(N As Double, f As Double, f2 As Double, ws As Double, _ fws As Double, mb As Double, rs As Double, bs As Double, _ js As Double, ms As Double, q As Double, fq As Double, _ m As Double) Dim x As Double, y As Double, z As Double, A As Double Dim d As Double, C As Double, j As Integer A = -js * f2 * OM ^ 2 + mb d = -ws * Cos(fws) * ms C = -ms * 9.81 - ws * Sin(fws) * ms * OM ^ 2 x = 0: y = 0: z = 0 For j = 1 To N x = x + qp(j) * Cos(fp(j)) y = y + qp(j) * Sin(fp(j)) z = z + r(j) * qp(j) * Sin(fp(j) - f - b3(j)) - rs * qp(j) * Sin(fp(j) - f - bs) Next j x = x + d y = y + C m = z + A Call PMY(x, y, q, fq) End Sub </pre>	

1.5.2. Силовой анализ группы Ассура II класса 1 вида

Рассмотрим группу Ассура II класса 1 вида, к которой приложены внешняя сила \bar{P}_B , главные векторы сил \bar{P}_2 , \bar{P}_3 и главные моменты \bar{M}_2 , \bar{M}_3 (рис. 1.10). Точки D и E – центры масс звеньев 2 и 3. Под действием сил \bar{P}_2 , \bar{P}_3 и моментов \bar{M}_2 , \bar{M}_3 в шарнирах A и C будут возникать реакции \bar{R}_A^n и \bar{R}_A^τ , \bar{R}_C^n и \bar{R}_C^τ .

Нормальные составляющие \bar{R}_A^n и \bar{R}_C^n направим по векторам \bar{L}_2 и \bar{L}_3 соответственно, а направление касательных \bar{R}_A^τ и \bar{R}_C^τ определим поворотом нормальных составляющих на 90° в принятом положительном направлении.


$$\sum M_{B_7} = \bar{M}_2 + \bar{r}_2 \times \bar{P}_2 + \bar{r}_A \times \bar{R}_A^\tau. \quad (1.142)$$
$$\bar{r}_2 = \bar{R}_2 - \bar{L}_2; \quad (1.143)$$

$$\bar{r}_A = -\bar{L}_2. \quad (1.144)$$

$$\overline{M}_2 + \overline{R}_2 \times \overline{P}_2 - \overline{L}_2 \times \overline{P}_2 - \overline{L}_2 \times \overline{R}_A^\tau = 0. \quad (1.145)$$
$$M_2 \cdot \bar{k} + R_2 \cdot P_2(\bar{e}_{R_2} \times \bar{e}_{P_2}) - L_2 \cdot P_2(\bar{e}_2 \times \bar{e}_{P_2}) - \bar{L}_2 \cdot \bar{R}_A^\tau(\bar{e}_2 \times \bar{e}_2) = 0. \quad (1.146)$$
$$R_A^\tau = \frac{M_2 + R_2 \cdot P_2 \sin(\varphi_{P_2} - \varphi_2 - \beta_2) - L_2 \cdot P_2 \sin(\varphi_{P_2} - \varphi_2)}{L_2}. \quad (1.147)$$

50

входящих в нее величин. Величины \bar{R}_A^n и \bar{R}_C^n определяются из векторного уравнения

$$\begin{aligned} \sum \bar{P}_i = & R_A^n \cdot \bar{e}_2 + R_A^\tau \cdot \bar{e}_2' + P_2 \cdot \bar{e}_{P_2} + P_3 \cdot \bar{e}_{P_3} + \\ & + R_C^\tau \cdot \bar{e}_3' + R_C^n \cdot \bar{e}_3 + P_B \cdot \bar{e}_{P_B} = 0. \end{aligned} \quad (1.148)$$

После умножения выражения (1.148) на \bar{e}_3' , раскрытия скалярных произведений векторов и разрешения результата относительно R_A^n получим

$$R_A^n = \frac{1}{\sin(\varphi_3 - \varphi_2)} \left(R_a^\tau \cos(\varphi_3 - \varphi_2) - P_2 \sin(\varphi_3 - \varphi_{P_2}) - \right. \\ \left. - P_3 \sin(\varphi_3 - \varphi_{P_3}) + R_C^\tau - P_B \sin(\varphi_3 - \varphi_{P_B}) \right). \quad (1.149)$$

После умножения выражения (1.148) на \bar{e}_2' аналогично имеем

$$R_C^n = \frac{1}{\sin(\varphi_2 - \varphi_3)} \left(R_C^\tau \cos(\varphi_2 - \varphi_3) - P_2 \sin(\varphi_2 - \varphi_{P_2}) - \right. \\ \left. - P_3 \sin(\varphi_2 - \varphi_{P_3}) + R_a^\tau - P_B \sin(\varphi_2 - \varphi_{P_B}) \right). \quad (1.150)$$

Анализ уравнений (1.149) и (1.150) указывает на их единую структуру, что можно учесть при разработке программы вычислений.

При рассмотрении равновесия звена 2 имеем

$$\bar{R}_B = -\bar{R}_A^n - \bar{R}_A^\tau - \bar{P}_2 - \bar{P}_B. \quad (1.151)$$

Умножением уравнения (1.151) скалярно на орты \bar{i} и \bar{j} находим проекции реакции \bar{R}_B на оси координат XOY :

$$\left. \begin{aligned} R_{BX} &= -R_A^n \cos(\varphi_2) + R_A^\tau \sin(\varphi_2) - P_2 \cos(\varphi_{P_2}) - P_B \cos(\varphi_{P_B}), \\ R_{BY} &= -R_A^n \sin(\varphi_2) - R_A^\tau \cos(\varphi_2) - P_2 \sin(\varphi_{P_2}) - P_B \sin(\varphi_{P_B}). \end{aligned} \right\} \quad (1.152)$$

Для определения полных реакций в шарнирах \bar{R}_A , \bar{R}_B и \bar{R}_C и их направлений φ_{R_A} , φ_{R_B} и φ_{R_C} можно воспользоваться подпрограммой *PMY*.

Целесообразно также разработать подпрограмму-функцию *FNPRТ* для вычисления касательных составляющих реакций \bar{R}_A^τ и \bar{R}_C^τ по формуле (1.147) и подпрограмму *PCI* для определения реакций R_A , R_B , R_C и их координирующих углов φ_{R_A} , φ_{R_B} и φ_{R_C} .

Входные параметры подпрограммы-функции *FNPR*T:

$M = M_2$ (или M_3); $R = R_2$ (R_3); $L = L_2$ (L_3); $P = P_2$ (P_3);

$F = \varphi_2$ (φ_3); $FP = \varphi_{P_2}$ (φ_{P_3}); $B = \beta_2$ (β_3);

$RAT = R_A^\tau$; $RCT = R_C^\tau$; $PB = P_B$;

$P2 = P_2$; $P3 = P_3$; $F2 = \varphi_2$; $F3 = \varphi_3$; $FPB = \varphi_{P_B}$; $FP2 = \varphi_{P_2}$; $FP3 = \varphi_{P_3}$.

OAP.BAS	FNPR
Function FNPR (m As Double, r As Double, l As Double, qp As Double, _ f As Double, fp As Double, B As Double) Dim A As Double, prtt As Double A = fp - f prtt = (m + r * qp * Sin(A - B) - l * qp * Sin(A)) / l FNPR = prtt End Function	

Входные параметры подпрограммы *PC*1:

$RAT = R_A^\tau$; $RCT = R_C^\tau$; $PB = P_B$;

$P2 = P_2$; $P3 = P_3$; $F2 = \varphi_2$; $F3 = \varphi_3$;

$FPB = \varphi_{P_B}$; $FP2 = \varphi_{P_2}$; $FP3 = \varphi_{P_3}$.

Выходные параметры:

$RA = R_A$; $RB = R_B$; $RC = R_C$; $FRA = \varphi_{R_A}$; $FRB = \varphi_{R_B}$; $FRC = \varphi_{R_C}$.

OAP.BAS	PC1
Sub PC1(rat As Double, rct As Double, pb As Double, pp2 As Double, _ p3 As Double, fpb As Double, f2 As Double, f3 As Double, _ fp2 As Double, fp3 As Double, ra As Double, rc As Double, _ rb As Double, fra As Double, frc As Double, frb As Double) Dim A As Double, C As Double, s As Double, ran As Double Dim rcn As Double, rbx As Double, rby As Double A = f3 - f2: C = Cos(A): s = Sin(A) ran = (rat * C - pp2 * Sin(f3 - fp2) - p3 * Sin(f3 - fp3) _ + rct - pb * Sin(f3 - fpb)) / s rcn = (rct * C - pp2 * Sin(f2 - fp2) - p3 * Sin(f2 - fp3) _ + rat - pb * Sin(f2 - fpb)) / (-s) rbx = -ran * Cos(f2) + rat * Sin(f2) - pp2 * Cos(fp2) - pb * Cos(fpb) rby = -ran * Sin(f2) - rat * Cos(f2) - pp2 * Sin(fp2) - pb * Sin(fpb) Call PMY(ran, rat, ra, fra) : fra = fra + f2 Call PMY(rcn, rct, rc, frc) : frc = frc + f3 Call PMY(rbx, rby, rb, frb) End Sub	

1.5.3. Силовой анализ группы Ассура II класса 2 вида

Схема сил, действующих на группу Ассура II класса 2 вида, показана на рис. 1.11.

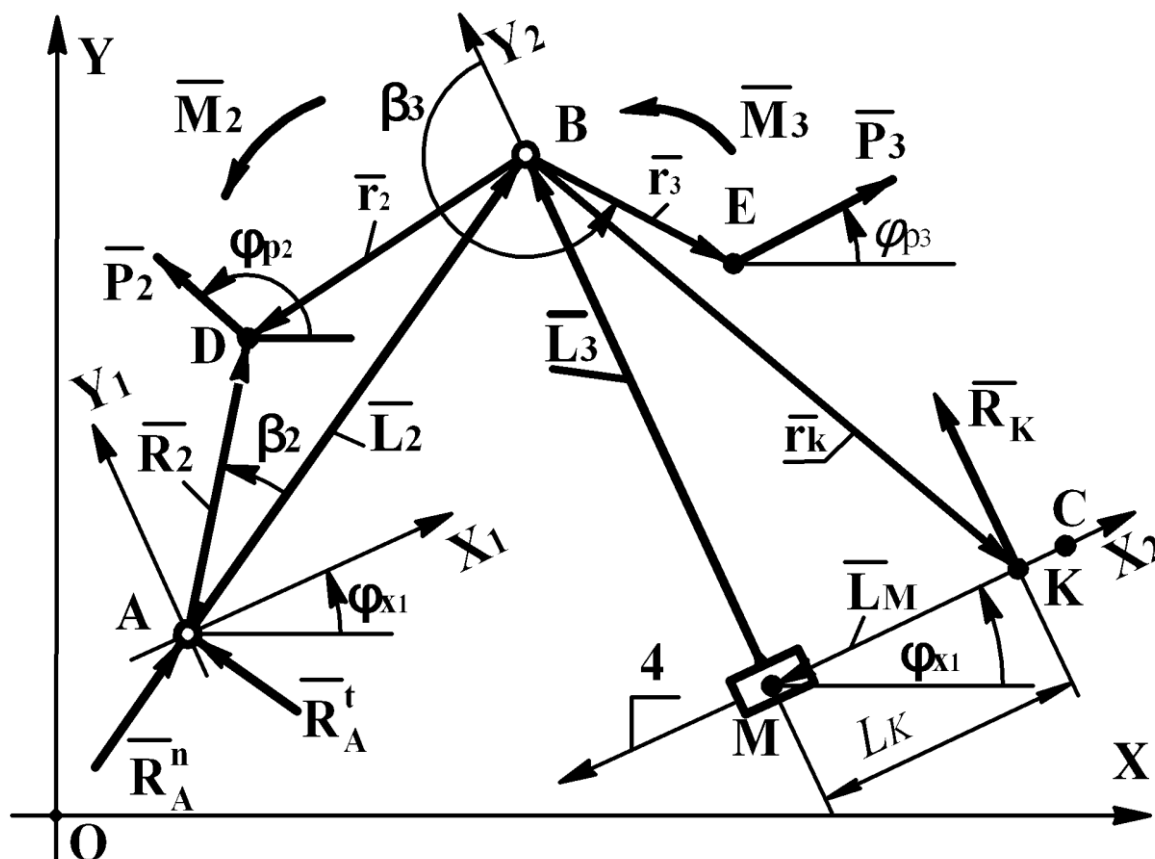


Рис. 1.11. Расчетная схема силового анализа группы Ассура II класса 2 вида

В отличие от группы Ассура II класса 1 вида в данном случае точка приложения реакции, действующей на звено 3 со стороны направляющей 4, неизвестна, но известно направление линии действия этой реакции.

Перенесем подвижную систему координат X_1AY_1 в точку M . Точку K приложения реакции \bar{R}_3^r расположим произвольно на положительной оси MX_2 новой подвижной системы координат X_2MY_2 . Реакцию \bar{R}_3^n направим по оси MY_2 .

Сравнительный анализ рис. 1.11 и 1.10 показывает, что методика определения всех реакций остается неизменной с учетом $R_3^r = 0$. Поэтому вели-

чины составляющих реакций и полные реакции определяются по тем же формулам и подпрограммам, что и в подразд. 1.5.2.

Обозначим реакцию \bar{R}_C^n через \bar{R}_K . Для определения точки приложения реакции \bar{R}_k составим следующее уравнение моментов относительно точки B от всех сил, действующих на звено 3:

$$\sum M_{B_3} = M_3 \cdot \bar{k} + r_3 \cdot P_3 (\bar{e}_{r_3} \times \bar{e}_{P_3}) + r_k \cdot R_3^n (\bar{e}_{r_k} \times \bar{j}_2) = 0. \quad (1.153)$$

Согласно рис. 1.11

$$\bar{r}_k = \bar{L}_k - \bar{L}_3. \quad (1.154)$$

Выражение (1.154) можно представить в виде

$$\bar{r}_k = L_k \cdot \bar{i}_2 - \bar{L}_3 \cdot \bar{j}_2. \quad (1.155)$$

Подстановкой формулы (1.155) в выражение (1.153) и раскрытием векторных произведений получим

$$L_k = \frac{-M_3 + r_3 \cdot R_3 \cos(\varphi_{X_1} - \varphi_{P_3} + \beta_3)}{R_k}. \quad (1.156)$$

Величина угла β_3 подставляется в уравнение (1.156) со своим знаком. Отрицательное значение L_k покажет, что точка K располагается на отрицательной полуоси MX_2 .

Таким образом, целесообразно иметь подпрограмму расчета реакции в кинематических парах A , B и K , их направляющих углов и точки приложения реакции \bar{R}_k . Подпрограмму расчета силовых параметров группы Ассура II класса 2 вида назовем $PC2$.

Входные параметры подпрограммы:

$$RAT = R_A^\tau; P2 = P_2; P3 = P_3; F2 = \varphi_2; F3 = \varphi_3; FP2 = \varphi_{P_2}; FP3 = \varphi_{P_3};$$

$$M3 = M_3; R3 = r_3; B3 = \beta_3; FX1(j) = \varphi_{X_1}.$$

Выходные параметры:

$$RA = R_A; RB = R_B; RK = R_K; FRA = \varphi_{R_A};$$

$$FRB = \varphi_{R_B}; LK = L_K.$$

OAP.BAS	PC2
<pre> Sub PC2(rat As Double, p2 As Double, p3 As Double, pb As Double, _ f2 As Double, f3 As Double, fp2 As Double, fp3 As Double, fpb As Double, _ m3 As Double, ra As Double, rb As Double, rk As Double, fra As Double, _ frb As Double, lk As Double) Dim j As Integer, A As Double, frk As Double, rkt As Double, C As Double Dim rbx As Double, rby As Double, s As Double, ran As Double Dim rkn As Double, rct As Double A = f3 - f2: rkt = 0: C = Cos(A): s = Sin(A) ran = (rat * C - p2 * Sin(f3 - fp2) - p3 * Sin(f3 - fp3) - pb * Sin(f3 - fpb)) / s rkn = (rct * C - p2 * Sin(f2 - fp2) - p3 * Sin(f2 - fp3) - pb * Sin(f2 - fpb)) / (-s) rbx = -ran * Cos(f2) + rat * Sin(f2) - p2 * Cos(fp2) - pb * Cos(fpb) rby = -ran * Sin(f2) - rat * Cos(f2) - p2 * Sin(fp2) - pb * Sin(fpb) Call PMY(ran, rat, ra, fra): fra = fra + f2: Call PMY(rkn, rkt, rk, frk) frk = frk + f3: Call PMY(rbx, rby, rb, frb) lk = (-m3 + r4(j) * p3 * Cos(b4(j) - fp3 - fx1(j))) / rkn End Sub </pre>	

1.5.4. Силовой анализ группы Ассура II класса 3 вида

Схема сил, действующих на группу Ассура II класса 3 вида, показана на рис. 1.12. Схема сил, действующих на звено 2 группы Ассура II класса 3 вида, представлена на рис. 1.13.

Неизвестную реакцию в точке A разложим на составляющие R_{AX_1} и R_{AY_1} , действующие по осям подвижной системы координат X_1AY_1 .

Реакцию в точке C разложим на составляющие R_{CX} и R_{CY} , действующие вдоль осей OX и OY .

Рассмотрим равновесие всей группы Ассура II класса 3 вида под действием приложенных сил и моментов:

$$\begin{aligned} \sum \bar{M}_C = & M_3 \cdot \bar{k} + M_2 \cdot \bar{k} + R_3 \cdot P_3 (\bar{e}_3 \cdot \bar{e}_{P_3}) + \\ & + R_{CD} \cdot P_2 (\bar{e}_{R_{CD}} \cdot \bar{e}_{P_2}) - L_4 \cdot R_{AY_1} (\bar{e}_4 \cdot \bar{e}'_4); \end{aligned} \quad (1.157)$$

$$\begin{aligned} \sum \bar{P}_i = & R_{AY_1} \cdot \bar{e}'_4 + R_{AX_1} \cdot \bar{e}_4 + P_2 \cdot \bar{e}_{P_2} + \\ & + P_3 \cdot \bar{e}_{P_3} + R_{XC} \cdot \bar{i} + R_{YC} \cdot \bar{j} = 0. \end{aligned} \quad (1.158)$$

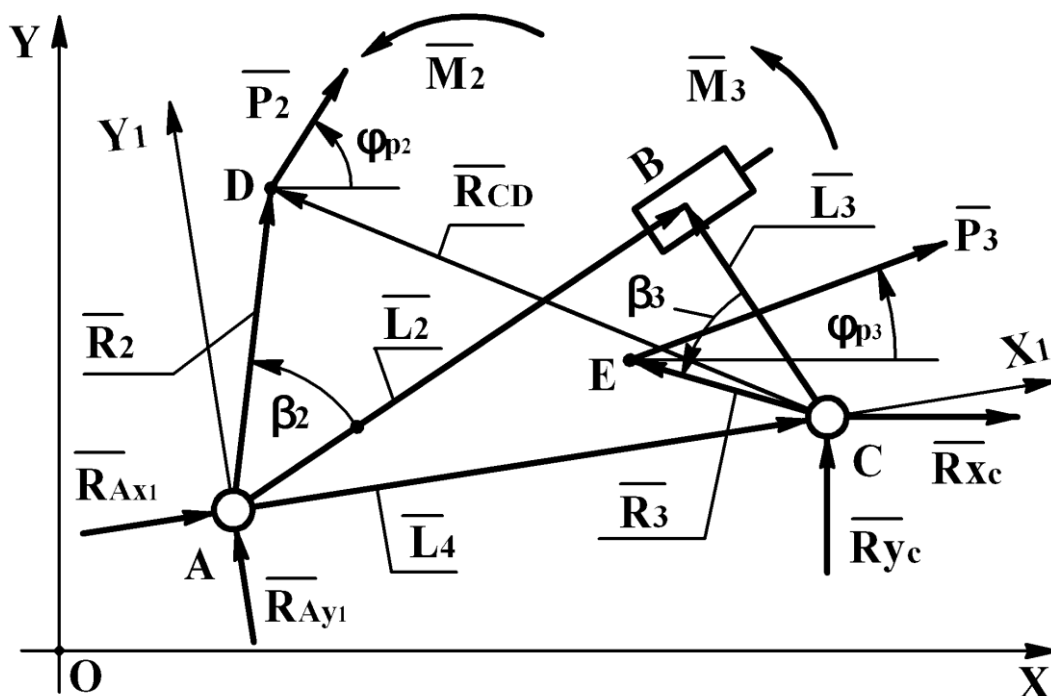


Рис. 1.12. Схема сил, действующих на группу Ассур II класса 3 вида

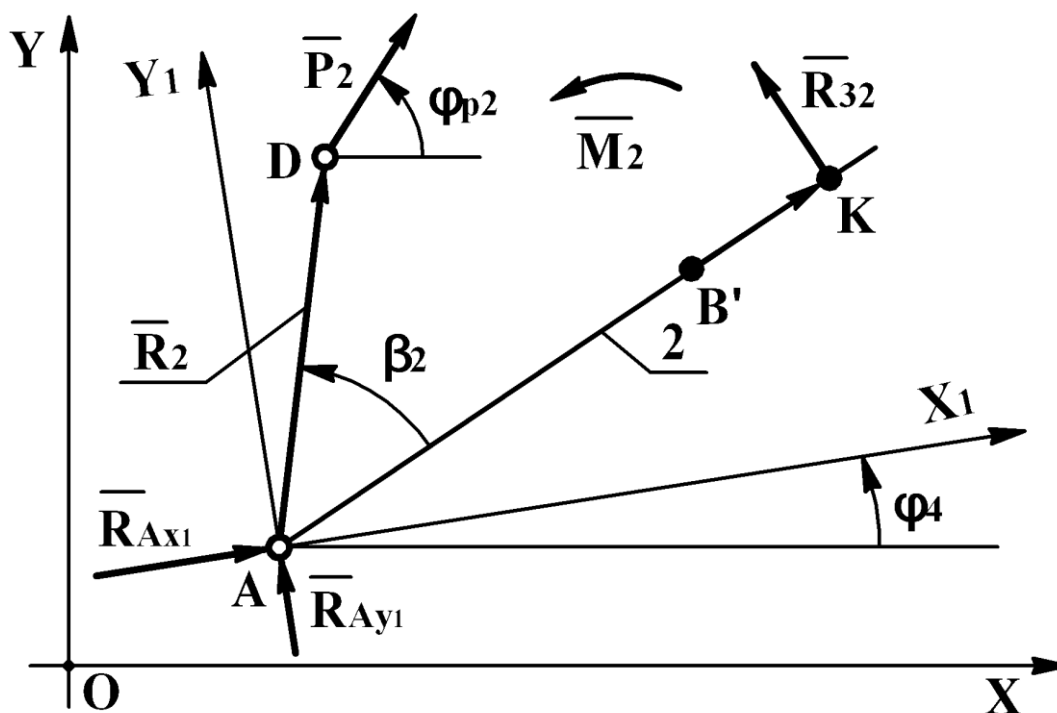


Рис. 1.13. Схема сил, действующих на звено 2 группы Ассур II класса 3 вида

Для определения реакций \bar{R}_{Ax1} и \bar{R}_{32} рассмотрим равновесие звена 2 (см. рис. 1.13):

$$\sum \bar{P}_i = R_{Ax1} \cdot \bar{e}_4 + R_{Ay1} \cdot \bar{e}'_4 + P_2 \cdot \bar{e}_{P2} + P_{32} \cdot \bar{e}'_2 = 0. \quad (1.159)$$

Умножим выражение (1.159) на \bar{e}_2 , затем на \bar{e}'_2 .

В результате после преобразований

$$R_{AX_1} = (R_{AY_1} \sin \varphi_4 - \varphi_2 + P_2 \cos(\varphi_2 - \varphi_{P2}))/\cos(\varphi_2 - \varphi_4); \quad (1.160)$$

$$R_{32} = R_{AX_1} \sin(\varphi_2 - \varphi_4) + R_{AY_1} \cos(\varphi_2 - \varphi_4) + P_2 \sin(\varphi_2 - \varphi_{P2}). \quad (1.161)$$

Составим следующее уравнение моментов относительно точки A от всех сил, действующих на звено 2:

$$M_2 \cdot \bar{k} + R_2 \cdot P_2 (\bar{e}_{R_2} \times \bar{e}_{P_2}) + \bar{L}_{AK} \cdot P_{32} (\bar{e}_2 \times \bar{e}'_2) = 0. \quad (1.162)$$

Расстояние, определяющее положение точки приложения реакции \bar{R}_{32} , рассчитывается по выражению

$$L_{AK} = -\frac{1}{R_{32}} (M_2 + R_2 \cdot P_2 \sin(\varphi_{P_2} - \varphi_{R_2})). \quad (1.163)$$

Для определения реакций \bar{R}_{CX} и \bar{R}_{CY} умножим выражение (1.158) на \bar{i} , затем на \bar{j} . После преобразований получим

$$\left. \begin{aligned} R_{XC} &= R_{AY_1} \sin \varphi_4 - R_{AX_1} \cos \varphi_4 - P_2 \cos(\varphi_{P_2}) - P_3 \cos(\varphi_{P_3}), \\ R_{YC} &= -R_{AY_1} \cos \varphi_4 - R_{AX_1} \sin \varphi_4 - P_2 \sin(\varphi_{P_2}) - P_3 \sin(\varphi_{P_3}). \end{aligned} \right\} \quad (1.164)$$

Полные реакции \bar{R}_A и \bar{R}_C , их направляющие углы φ_{R_A} и φ_{R_C} определяются с помощью подпрограммы PMY .

Направляющий угол в системе координат XOY

$$\varphi_{R_A} = \varphi_4 + \varphi_{RA_1}. \quad (1.165)$$

где φ_{RA_1} – направляющий угол в системе координат X_1AY_1 .

Подпрограмме силового анализа группы Ассура II класса 3 вида присвоим имя $PC3$.

Входные параметры:

$$R2 = R_2; R3 = R_3; L4 = L_4; F2 = \varphi_2; F3 = \varphi_3; FX1 = \varphi_4;$$

$$B2 = \beta_2; B3 = \beta_3; M2 = M_2; M3 = M_3; P2 = P_2; P3 = P_3;$$

$$FP2 = \varphi_{P_2}; FP3 = \varphi_{P_3}.$$

Выходные параметры подпрограммы:

$$RA = R_A; FRA = \varphi_{R_A}; R32 = R_{32}; RC = R_C; FRC = \varphi_{R_C}; LAK = L_{AK}.$$

```

SUB PC3(L4 As Double, F2 As Double, F3 As Double, M2 As Double, _
M3 As Double, P2 As Double, P3 As Double, FP2 As Double, _
FP3 As Double, RA As Double, FRA As Double, R32 As Double, _
RC As Double, FRC As Double, LAK As Double)
DEF FNC(R, F, U):C=R*COS(F-U):END DEF
DEF FNC(R, F, U):C=R*SIN(F-U): END DEF
RAY=(M2+M3+FNS(R3(J)*P3,FP3,F3+B3(J))+FNS(R2(J)*P2,FP2,F2+_
B2(J))-FNS(L4*P2,FP2,FX1(J)))/L4:RAX=(FN(RAY,FX1(J),F2)_
+FNC(P2,F2,FP2))/FNC(1,F2,FX1(J)):R32= FNS(RAX, F2, FX1(J))-_
FNC(RAY,F2,FX1(J))+FNS(P2,F2,FP2)
LAK= -(M2+FNS(R2(J)*P2,FP2,F2+B2(J)))/R32
RXC=FNS(RAY,FX1(J),0)-FNC(RAX,FX1(J),0)-FNC(P2,FP2,0)-FNC(P3,FP3,0)
RYC=-FNC(RAY,FX1(J),0)-FNS(RAX,FX1(J),0)-FNS(P2,FP2,0)-FNC(P3,FP3,0)
CALL PMY(RAX,RAY,RA,FRA): FRA=FRA+FX1(J)
CALL PMY(RXC, RYC, RC, FRC)
END SUB

```

1.5.5. Силовой анализ группы Ассура II класса 4 вида

Схема сил, действующих на группу Ассура II класса 4 вида, показана на рис. 1.14.

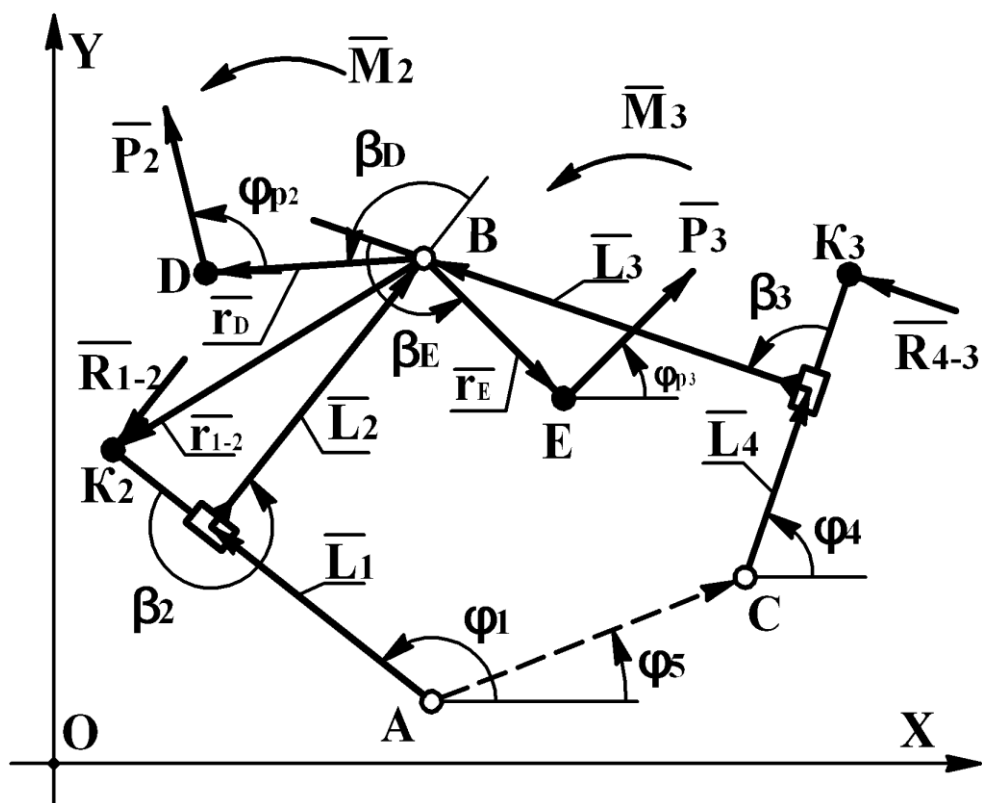


Рис. 1.14. Схема сил, действующих на группу Ассура II класса 4 вида

Неизвестными являются реакции \bar{R}_{1-2} , \bar{R}_{4-3} , а также реакция \bar{R}_{3-2} во внутреннем шарнире B .

Рассмотрим равновесие всей группы Ассура II класса 4 вида под действием приложенных сил и моментов:

$$\sum \bar{P}_i = \bar{R}_{1-2} + \bar{P}_2 + \bar{P}_3 + \bar{P}_{4-3} = 0. \quad (1.166)$$

Умножим уравнение (1.166) на \bar{e}_1 :

$$P_2 \cdot \bar{e}_{P_2} \cdot \bar{e}_1 + P_3 \cdot \bar{e}_{P_3} \cdot \bar{e}_1 + R_{4-3} \cdot \bar{e}'_4 \cdot \bar{e}_1 = 0. \quad (1.167)$$

Раскроем скалярные произведения векторов:

$$P_2 \cos(\varphi_1 - \varphi_{P_2}) + P_3 \cos(\varphi_1 - \varphi_{P_3}) + R_{4-3} \sin(\varphi_1 - \varphi_4) = 0. \quad (1.168)$$

Из уравнения (1.168)

$$R_{4-3} = \frac{P_2 \cos(\varphi_1 - \varphi_{P_2}) + P_3 \cos(\varphi_1 - \varphi_{P_3})}{\sin(\varphi_4 - \varphi_1)}. \quad (1.169)$$

Умножим уравнение (1.166) на \bar{e}_4 и раскроем скалярные произведения векторов:

$$R_{1-2} = \frac{P_2 \cos(\varphi_4 - \varphi_{P_2}) + P_3 \cos(\varphi_4 - \varphi_{P_3})}{\sin(\varphi_1 - \varphi_4)}. \quad (1.170)$$

Для определения положения точки K_2 приложения реакции \bar{R}_{1-2} составим следующее уравнение моментов от всех сил, действующих на звено 2:

$$\bar{r}_{1-2} \times \bar{R}_{1-2} + \bar{r}_D \times \bar{P}_2 + \bar{M}_2 = 0. \quad (1.171)$$

На основании рис. 1.14

$$\bar{L}_{AB} = \bar{L}_1 + \bar{L}_2; \quad (1.172)$$

$$\bar{L}_{AK_2} = \bar{r}_{1-2} + \bar{L}_2 + \bar{L}_1 = 0; \quad (1.173)$$

$$\bar{r}_{1-2} = \bar{L}_{AK_2} - \bar{L}_2 - \bar{L}_1 = 0. \quad (1.174)$$

Подстановкой выражения (1.174) в уравнение (1.171) получим

$$\bar{L}_{AK_2} \times \bar{R}_{1-2} - \bar{L}_2 \times \bar{R}_{1-2} - \bar{L}_1 \times \bar{R}_{1-2} + \bar{r}_D \times \bar{P}_2 + \bar{M}_2 = 0. \quad (1.175)$$

Раскроем векторные произведения в выражении (1.175) и разрешим его относительно L_{AK_2} :

$$L_{AK_2} = \frac{1}{R_{1-2}} (L_2 \cdot R_{1-2} \cos \beta_2) + L_1 \cdot R_{1-2} - r_D \cdot P_2 \sin(\varphi_{P_2} - \varphi_1 - \beta_2 - \beta_D) - M_2. \quad (1.176)$$

Аналогично для точки K_3

$$L_{AK_3} = \frac{1}{R_{4-3}} (L_3 \cdot R_{4-3} \cos \beta_3) + L_4 \cdot R_{4-3} - r_E \cdot P_3 \sin(\varphi_{P_3} - \varphi_3 - \beta_3 - \beta_E) - M_3. \quad (1.177)$$

Для определения реакций \bar{R}_{3-2} составим уравнение равновесия звена 2:

$$\bar{R}_{1-2} + \bar{P}_2 + \bar{R}_{3-2} = 0. \quad (1.178)$$

Из выражения (1.178)

$$\bar{R}_{3-2} = -\bar{R}_{1-2} - \bar{P}_2. \quad (1.179)$$

Последовательно умножим выражение (1.179) на орты \bar{i} и \bar{j} системы координат:

$$\left. \begin{aligned} R_{3-2}^X &= -R_{1-2} \cos(\varphi_1 + 0,5\pi) - P_2 \cos \varphi_{P_2} = R_{1-2} \sin \varphi_1 - P_2 \cos \varphi_{P_2}, \\ R_{3-2}^Y &= -R_{1-2} \sin(\varphi_1 + 0,5\pi) - P_2 \sin \varphi_{P_2} = -R_{1-2} \cos \varphi_1 - P_2 \sin \varphi_{P_2}. \end{aligned} \right\} \quad (1.180)$$

Полная реакция \bar{R}_{3-2} и ее направляющий угол $\varphi_{R_{3-2}}$ определяются с помощью подпрограммы *PMY*. Подпрограмме силового анализа группы Ассура II класса 4 вида присвоим имя *PC4*.

Входные параметры подпрограммы:

$$L1 = L_1; L2 = L_2; L3 = L_3; L4 = L_4;$$

$$FX1 = \varphi_1; FX4 = \varphi_4; B2 = \beta_2; B3 = \beta_3; M2 = M_2; M3 = M_3;$$

$$P2 = P_2; P3 = P_3; FP2 = \varphi_{P_2}; FP3 = \varphi_{P_3};$$

$$RD = r_D; BD = \beta_D; RE = r_E; BE = \beta_E.$$

Выходные параметры:

$$R12 = R_{1-2}; R43 = R_{4-3}; LAK2 = L_{AK_2}; LCK3 = L_{CK_3};$$

$$R32 = R_{3-2}; FR32 = \varphi_{R_{3-2}}.$$

Рассмотрим равновесие звена 2:

$$\sum \bar{P}_i = \bar{R}_{1-2} + \bar{P}_2 + \bar{P}_{3-2} = 0. \quad (1.181)$$

Умножим уравнение (1.181) на вектор \bar{e}_1 :

$$P_2 \cos(\varphi_2 - \varphi_{P_2}) + R_{3-2} \cos(0,5 \cdot \pi + \varphi_1 + \beta_2 - \varphi_1) = 0. \quad (1.182)$$

После преобразования уравнения (1.182) имеем

$$P_2 \cos(\varphi_2 - \varphi_{P_2}) + R_{3-2} \sin \beta_2 = 0. \quad (1.183)$$

Тогда реакция \bar{R}_{3-2} определится следующим образом:

$$R_{3-2} = \frac{P_2 \cos(\varphi_2 - \varphi_{P_2})}{\sin \beta_2}. \quad (1.184)$$

На основании уравнения (1.181)

$$R_{1-2} = \frac{-P_2 \cos(\varphi_1 + \beta_2 - \varphi_{P_2})}{\sin \beta_2}. \quad (1.185)$$

Рассмотрим равновесие звена 3:

$$\sum \bar{P}_i = -\bar{R}_{3-2} + \bar{P}_3 + \bar{R}_C^X + \bar{R}_C^Y = 0. \quad (1.186)$$

Перепишем выражение (1.186) в виде

$$\bar{R}_{3-2} - \bar{P}_3 = \bar{R}_C^X + \bar{R}_C^Y. \quad (1.187)$$

Умножим уравнение (1.187) на вектор \bar{i} , затем на вектор \bar{j} :

$$\left. \begin{aligned} R_C^X &= -R_{3-2} \cos(0,5 \cdot \pi + \varphi_1 + \beta_2) - P_3 \cos(\varphi_{P_3}); \\ R_C^Y &= R_{3-2} \sin(0,5 \cdot \pi + \varphi_1 + \beta_2) - P_3 \sin(\varphi_{P_3}). \end{aligned} \right\} \quad (1.188)$$

После преобразований в системе (1.188)

$$\left. \begin{aligned} R_C^X &= -R_{3-2} \sin(\varphi_1 + \beta_2) - P_3 \cos(\varphi_{P_3}); \\ R_C^Y &= R_{3-2} \cos(\varphi_1 + \beta_2) - P_3 \sin(\varphi_{P_3}). \end{aligned} \right\} \quad (1.189)$$

Полная величина реакции \bar{R}_C и направляющий угол φ_{R_C} определяются с помощью подпрограммы *РМУ*.

Для определения положения точки K_3 (точки приложения реакции \bar{R}_{3-2}) составим следующее уравнение моментов относительно точки C от всех сил, действующих на звено 3:

$$-\bar{r}_{K_3} \cdot \bar{R}_{3-2} + \bar{r}_E \cdot \bar{P}_3 + \bar{M}_3 = 0. \quad (1.190)$$

На основании рис. 1.15

$$\bar{r}_{K_3} = \bar{L}_{BK_3} + \bar{L}_{CB}; \quad (1.191)$$

$$\bar{L}_{CB} = \bar{L}_3 - \bar{L}_2. \quad (1.192)$$

Подстановкой выражения (1.192) в выражение (1.191) получим

$$\bar{r}_{K_3} = \bar{L}_{BK_3} + \bar{L}_3 - \bar{L}_2. \quad (1.193)$$

Перепишем выражение (1.190) с учетом выражения (1.193):

$$-\bar{L}_{BK_3} \cdot \bar{R}_{3-2} - \bar{L}_3 \cdot \bar{R}_{3-2} + \bar{L}_2 \cdot \bar{R}_{3-2} + \bar{r}_E \cdot \bar{P}_3 + \bar{M}_3 = 0. \quad (1.194)$$

Умножим скалярно выражение (1.194) на вектор \bar{k} и получим

$$\bar{L}_{BK_3} = \frac{1}{R_{3-2}} \left[-L_3 \cdot R_{3-2} \cos(\beta_3) + L_2 \cdot R_{3-2} + r_E \cdot P_3 \sin(\varphi_{P_3} - \varphi_1 - (\beta_2 + \beta_3 + \beta_E)) + M_3 \right]. \quad (1.195)$$

Рассмотрим равновесие звена 2 и составим следующее уравнение моментов от всех сил, действующих на это звено, относительно точки A :

$$\bar{L}_{AK_2} \cdot \bar{R}_{1-2} + \bar{L}_1 \cdot \bar{P}_2 + \bar{r}_D \cdot \bar{P}_2 + \bar{M}_2 + \bar{L}_1 \cdot \bar{R}_{3-2} + \bar{L}_{BK_3} \cdot \bar{R}_{3-2} = 0. \quad (1.196)$$

Расстояние L_{AK_2} до точки приложения реакции \bar{R}_{1-2}

$$L_{AK_2} = \frac{1}{R_{1-2}} \left[-L_1 \cdot P_2 \sin(\varphi_{P_2} - \varphi_1) + r_D \cdot P_2 \sin(\varphi_{P_2} - \varphi_1 - \beta_2 - \beta_D) + M_2 + L_1 \cdot R_{3-2} \cos(\beta_2) + L_{BK_3} \cdot R_{3-2} \right]. \quad (1.197)$$

Подпрограмме силового анализа группы Ассур II класса 5 вида присвоим имя $PC5$.

Входные параметры подпрограммы:

$$L1 = L_1; L2 = L_2; L3 = L_3; FX1 = \varphi_1; BD = \beta_D;$$

$$P2 = P_2; P3 = P_3; FP2 = \varphi_{P_2}; FP3 = \varphi_{P_3};$$

$$M2 = M_2; M3 = M_3; B2 = \beta_2;$$

$$B3 = \beta_3; RE = r_E; BE = \beta_E.$$

Выходные параметры:

$$R12 = R_{1-2}; R32 = R_{3-2}; LAK2 = L_{AK_2}; LBK3 = L_{BK_3}; RC = R_C.$$

OAP.BAS	PC5
SUB PC5(L1 As Double, L2 As Double, L3 As Double, B2 As Double, _ B3 As Double, M2 As Double, M3 As Double, P2 As Double, _ P3 As Double, FP2 As Double, FP3 As Double, R12 As Double, _ R32 As Double, LAK2 As Double, LBK3 As Double, RC As Double) R32=(P2*COS(FX1(J)-FP2/SIN(B2):R12=(-P2*COS(FX1(J)+B2-FP2/_ SIN(B2):LBK3=(-L3*R32*COS(B3)+L2*R32-RE(J)*P3*SIN(FP3-FX1(J)-_ B2-B3(J))+M3)/R32:LAK2=(-L1*P2*SIN(FP2-X1(J))+RD(J)*P2*_ SIN(FP2-FX1(J)-B2-BD(J))+M2+L1*R32*COS(B2)+LBK3*R32)/R12 RCX=-R32*SIN(F1+B2)-P3*COS(FP3):RCY=R32*COS(F1+B2)-_ P3*SIN(FP3):CALL PMY(RCX, RCY, RC, FRC) END SUB	

1.5.7. Силовой анализ входного звена

В стержневых механизмах входное звено (звено 1) может быть выполнено в виде кривошипа или толкателя кулачка, т. е. звено \bar{L}_1 (рис. 1.16) может совершать вращательное движение с постоянной угловой скоростью (кривошип) или возвратно-вращательное движение (толкатель).

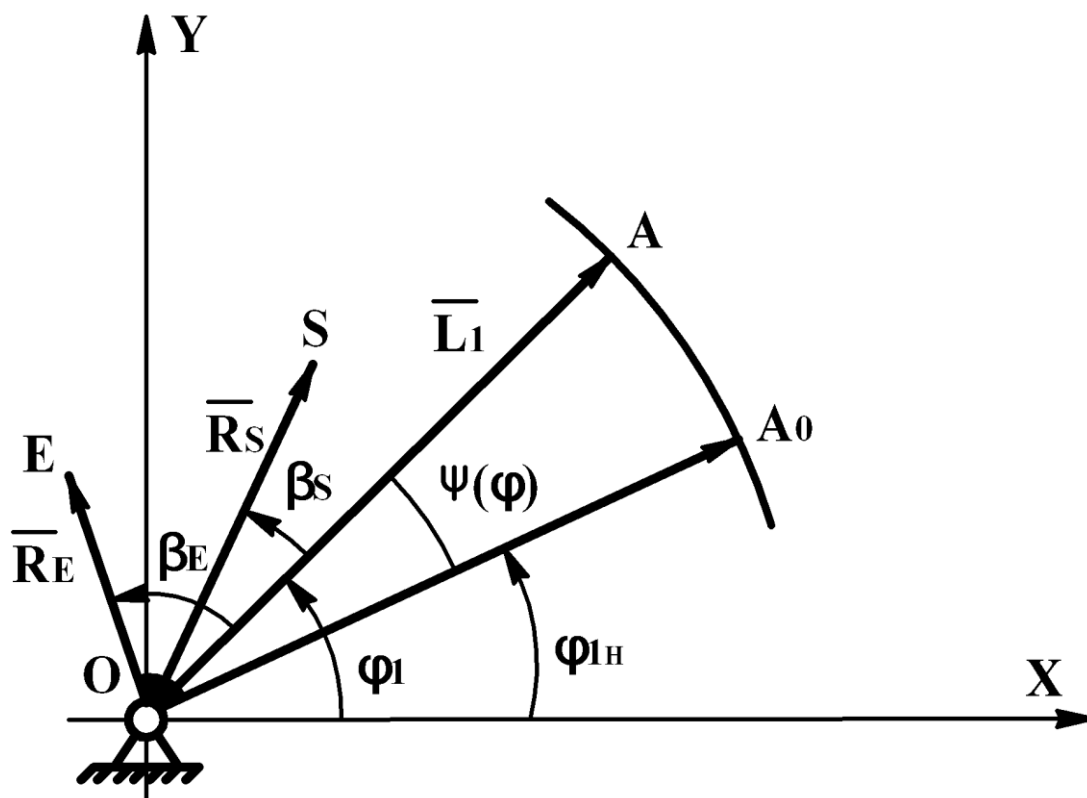


Рис. 1.16. Расчетная схема кинематики входного звена

Для кривошипа текущая угловая координата φ_1 является обобщенной координатой всего механизма. Кривошип – это звено, совершающее полный оборот относительно неподвижной оси вращения. Тогда закон движения кривошипа можно представить в виде выражения

$$\varphi_1 = \varphi_{1N} + (i - 1)\Delta\varphi, \quad (1.198)$$

где φ_{1N} – значение угловой координаты в начальный момент времени (обычно крайнее положение механизма);

$\Delta\varphi$ – шаг расчета;

$i = 1, 2, 3, \dots, n$ (где n – число расчетных точек).

Число расчетных точек n зависит от максимального хода звена и шага расчета.

Если шаг расчета задан:

- в градусах, то

$$n = \frac{360^\circ}{\Delta\varphi} + 1; \quad (1.199)$$

- в радианах, то

$$n = \frac{2 \cdot \pi}{\Delta\varphi} + 1. \quad (1.200)$$

При постоянной величине угловой скорости кривошипа:

- аналог угловой скорости

$$\varphi'_1 = \frac{d\varphi_1}{d\varphi_1} = 1; \quad (1.201)$$

- аналог углового ускорения

$$\varphi''_1 = \frac{d^2\varphi_1}{d\varphi_1^2} = 0. \quad (1.202)$$

Если звено I является толкателем, то обобщенной координатой будет угол φ поворота кулачка. Координата φ_1 определяется из выражения

$$\varphi_1 = \varphi_{1N} + \Psi(\varphi), \quad (1.203)$$

где $\Psi(\varphi)$ – функция положения (закон движения) толкателя.

Аналоги угловых скоростей и ускорений толкателя

$$\varphi_1' = \Psi'(\varphi); \quad (1.204)$$

$$\varphi_1'' = \Psi''(\varphi), \quad (1.205)$$

где $\Psi'(\varphi)$, $\Psi''(\varphi)$, – аналоги скорости и ускорения толкателя соответственно.

Число расчетных точек зависит от формы кулачка:

$$n = N_1 + N_2, \quad (1.206)$$

где N_1 – число расчетных точек на фазе удаления;

N_2 – число расчетных точек на фазе сближения.

$$N_1 = \varphi_{y\partial} / \Delta\varphi_1 + 1; \quad (1.207)$$

$$N_2 = \varphi_{c\partial} / \Delta\varphi_2 + 1, \quad (1.208)$$

где $\varphi_{y\partial}$, $\varphi_{c\partial}$ – соответственно фазовые углы удаления и сближения;

$\Delta\varphi_1$, $\Delta\varphi_2$ – шаги вычислений на соответствующих фазах кулачка.

Координаты, аналоги линейных скоростей и ускорений точки A и промежуточных точек S и E входного звена \bar{L}_1 (рис. 1.16) определяются с помощью подпрограмм PKT и PA . Подпрограмма PBC применяется для определения главного вектора сил \bar{P}_1 и главного момента сил \bar{M}_1 , действующих на звено I

К входному звену (звену I) в зависимости от конструкции привода приложены либо уравнивающий момент \bar{M}_{YP} (привод посредством муфты), либо уравнивающая сила \bar{P}_{YP} (привод с помощью зубчатого зацепления или кулачковый привод). Схема сил, действующих на входное звено (звено I), показана на рис. 1.17.

Для определения уравнивающего момента составим уравнение моментов относительно оси O всех сил, действующих на входное звено:

$$\bar{k} \cdot M_{YP} + M_1 \cdot \bar{k} + R_1 \cdot P_1 (\bar{e}_{R_1} \times \bar{e}_{P_1}) = 0. \quad (1.209)$$

Раскроем векторное произведение векторов в выражении (1.209), помножим результат на орт \bar{k} оси OZ и после преобразования получим

$$M_{YP} = -M_1 - R_1 \cdot P_1 \sin(\varphi_{P_1} - \varphi_{R_1}). \quad (1.210)$$

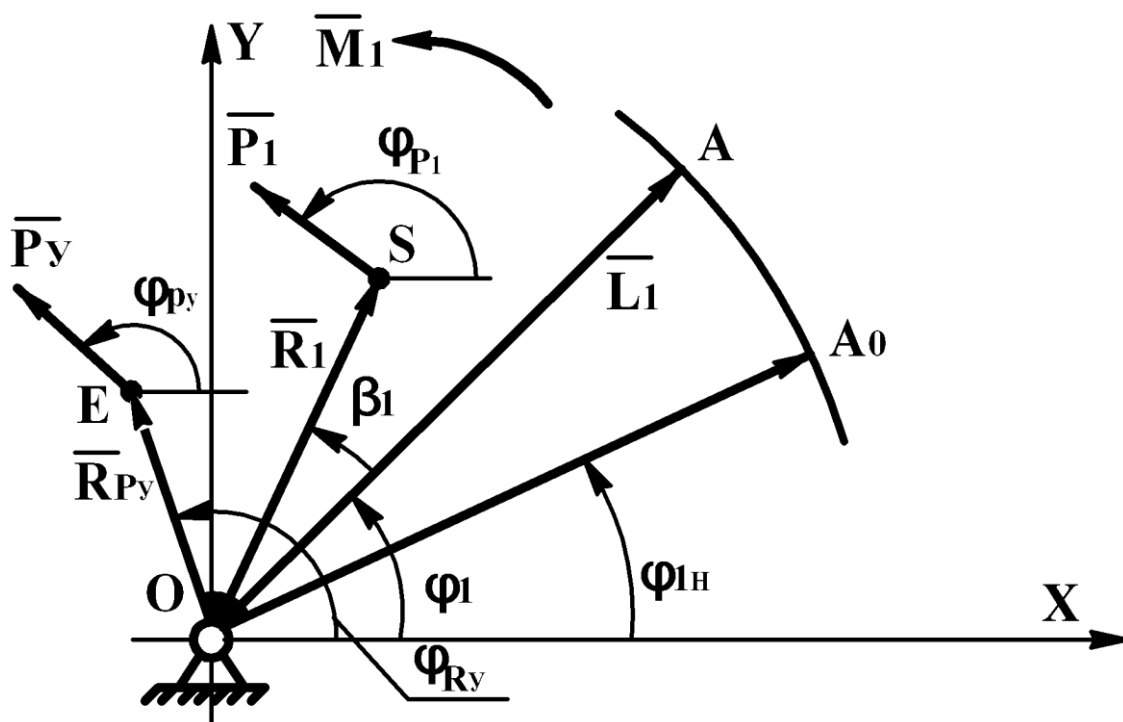


Рис. 1.17. Схема сил, действующих на входное звено (звено 1)

Уравновешивающую силу можно выразить следующим образом:

$$P_{YP} = \frac{M_{YP}}{R_{PY} \sin(\varphi_{PY} - \varphi_{R_{PY}})}. \quad (1.211)$$

Точка приложения и линия действия силы \bar{P}_{YP} определяются в каждом конкретном случае исходя из конструктивных особенностей привода. Для кривошипа точка приложения – шарнир A , а ее направление – перпендикуляр к \bar{L}_1 . Реакцию \bar{R}_O в опоре O разложим на составляющие R_{OX} и R_{OY} , направленные вдоль осей системы координат. Уравнение равновесия звена 1 под действием приложенных сил

$$\bar{P}_{YP} + \bar{P}_1 + \bar{R}_{OX} + \bar{R}_{OY} = 0. \quad (1.212)$$

Умножим выражение (1.212) на вектор \bar{i} , затем на вектор \bar{j} :

$$\left. \begin{aligned} R_{OX} &= -P_{YP} \cos(\varphi_{PY}) - P_1 \cos(\varphi_{P_1}); \\ R_{OY} &= -P_{YP} \sin(\varphi_{PY}) - P_1 \sin(\varphi_{P_1}). \end{aligned} \right\} \quad (1.213)$$

Полная реакция \bar{R}_O и ее направляющий угол φ_{R_O} определяются с использованием подпрограммы *PMY*. На этом методика силового анализа входного звена завершена.

Для вспомогательных расчетов также требуется ряд подпрограмм.

Подпрограмма-функция *Function FNARC* – для возвращения арккосинуса аргумента. Входным параметром является *arg* – аргумент функции.

OAP.BAS	FNARC
Function FNARC(arg As Double) Dim alfa As Double alfa = Atn(Sqr(1 - arg ^ 2) / arg): If arg < 0 Then alfa = alfa + PI FNARC = alfa End Function	

Подпрограмма-функция *Function FNasn* – для возвращения арксинуса аргумента. Входным параметром является *arg* – аргумент функции.

OAP.BAS	FNasn
Function FNasn(arg As Double) Dim alfa As Double alfa = Atn(arg / Sqr(1 - arg ^ 2)) If arg < 0 Then alfa = 2 * PI + alfa FNasn = alfa End Function	

Подпрограмма *GRAF* – для построения графика одной функции из массива данных. Входные параметры: *y1()* – массив данных рассматриваемой переменной; *fo* – имя формы, на которой располагается окно для изображения графика; *color* – цвет линии графика; *poyasnenie* – текст пояснения графика.

OAP.BAS	GRAF
Sub graf(y1() As Double, fo As Form, color As Integer, poyasnenie As String) Dim MMAX As Double, mmin As Double, goriz1 As Double, k As Double Dim goriz2 As Double, goriz3 As Double, goriz4 As Double, K1 As Double Dim K2 As Double, Picture1 As String Rem ТОЛЩИНА ЛИНИЙ fo.Picture1.DrawWidth = 1 Rem ОПРЕДЕЛЕНИЕ МАКСИМУМОВ И МИНИМУМОВ ФУНКЦИЙ MMAX = 0: mmin = 0: mmin = y1(LBound(y1)) For i = LBound(y1) To UBound(y1) If y1(i) <= mmin Then mmin = y1(i) Next MMAX = y1(LBound(y1))	


```

For i = LBound(y1) To UBound(y1)
If y1(i) > MMAХ Then MMAХ = y1(i)
Next
Rem ЧТОБЫ НЕ ИСЧЕЗАЛ НОЛЬ
If mmin > 0 And MMAХ > 0 Then mmin = 0
If mmin < 0 And MMAХ < 0 Then MMAХ = 0
Rem ГОРИЗОНТАЛЬНАЯ РАЗМЕТКА
fo.Picture1.Scale (LBound(y1), fo.Picture1.Height)-(UBound(y1), 0)
goriz1 = fo.Picture1.Height / 4:goriz2 = fo.Picture1.Height / 2
goriz3 = 3 * fo.Picture1.Height / 4
fo.Picture1.Line (LBound(y1), goriz1)-(UBound(y1), goriz1)
fo.Picture1.Line (LBound(y1), goriz2)-(UBound(y1), goriz2)
fo.Picture1.Line (LBound(y1), goriz3)-(UBound(y1), goriz3)
Rem РАЗМЕРЫ ПОЛЯ ДЛЯ ВЫВОДА ГРАФИКОВ
fo.Picture1.Scale (LBound(y1), MMAХ)-(UBound(y1), mmin)
Rem ВЕРТИКАЛЬНАЯ РАЗМЕТКА
fo.Picture1.Line (0.5 * UBound(y1), mmin)-(0.5 * UBound(y1), MMAХ)
fo.Picture1.Line (0.25 * UBound(y1), mmin)-(0.25 * UBound(y1), MMAХ)
fo.Picture1.Line (0.75 * UBound(y1), mmin)-(0.75 * UBound(y1), MMAХ)
Rem Ось ОХ
68 fo.Picture1.DrawWidth = 3
fo.Picture1.Line (LBound(y1), 0)-(UBound(y1), 0), QBColor(0)
Rem ВЫВОД ГРАФИКОВ ФУНКЦИЙ
fo.Picture1.DrawWidth = 3
For K1 = LBound(y1) + 1 To UBound(y1) Step 0.1
K2 = K1 + 1
If K2 > UBound(y1) Then K2 = UBound(y1)
fo.Picture1.Line (K1, y1(K1))-(K2, y1(K2)), QBColor(color)
Next
Rem ЭКСТРЕМУМЫ ФУНКЦИИ
fo.CurrentX = fo.Picture1.Left - 300
fo.CurrentY = fo.Picture1.Top + MMAХ * (fo.Picture1.Height) / _
Abs(MMAХ) + Abs(mmin))
fo.Print "0":fo.CurrentX = fo.Picture1.Left - 300
fo.CurrentY = fo.Picture1.Top - 300:fo.Print Format(MMAХ, "000.000")
fo.CurrentX = fo.Picture1.Left - 300
fo.CurrentY = fo.Picture1.Top + fo.Picture1.Height + 300
fo.Print Format(mmin, "000.000")
End Sub

```

Подпрограмма *GRAF3* – для построения графика двух или трех функций из массивов данных. Входные параметры: *y1()*, *y2()*, *y3()* – массивы данных рассматриваемых переменных; *fo* – имя формы, на которой располагается ок-

но для изображения графика; *color* – цвет линии графика; *poyasnenie* – текст пояснения графиков.

OAP.BAS	GRAF3
<pre> Sub graf3(y1() As Double, y2() As Double, y3() As Double, fo As Form, _ color As Integer, poyasnenie As String) Dim MMAX As Double, mmin As Double, goriz1 As Double, k As Double Dim goriz2 As Double, goriz3 As Double, goriz4 As Double, i As Double Dim K2 As Double, K1 As Double, Picture1 As String Rem ТОЛЩИНА ЛИНИЙ fo.Picture1.DrawWidth = 1 Rem ОПРЕДЕЛЕНИЕ МАКСИМУМОВ И МИНИМУМОВ ФУНКЦИЙ MMAX = 0: mmin = 0: mmin = y1(LBound(y1)) For i = LBound(y1) To UBound(y1) If y1(i) <= mmin Then mmin = y1(i) Next For i = LBound(y2) To UBound(y2) If y2(i) <= mmin Then mmin = y2(i) Next For i = LBound(y3) To UBound(y3) If y3(i) <= mmin Then mmin = y3(i) Next MMAX = y1(LBound(y1)) For i = LBound(y1) To UBound(y1) If y1(i) > MMAX Then MMAX = y1(i) Next For i = LBound(y2) To UBound(y2) If y2(i) > MMAX Then MMAX = y2(i) Next For i = LBound(y3) To UBound(y3) If y3(i) > MMAX Then MMAX = y3(i) Next Rem УСЛОВИЕ СОХРАНЕНИЯ НУЛЯ If mmin > 0 And MMAX > 0 Then mmin = 0 If mmin < 0 And MMAX < 0 Then MMAX = 0 Rem ГОРИЗОНТАЛЬНАЯ РАЗМЕТКА fo.Picture1.Scale (LBound(y1), fo.Picture1.Height)-(UBound(y1), 0) goriz1 = fo.Picture1.Height / 4: goriz2 = fo.Picture1.Height / 2 goriz3 = 3 * fo.Picture1.Height / 4 fo.Picture1.Line (LBound(y1), goriz1)-(UBound(y1), goriz1) fo.Picture1.Line (LBound(y1), goriz2)-(UBound(y1), goriz2) fo.Picture1.Line (LBound(y1), goriz3)-(UBound(y1), goriz3) Rem РАЗМЕРЫ ПОЛЯ ДЛЯ ВЫВОДА ГРАФИКОВ fo.Picture1.Scale (LBound(y1), MMAX)-(UBound(y1), mmin) </pre>	

```

Rem ВЕРТИКАЛЬНАЯ РАЗМЕТКА
fo.Picture1.Line (0.5 * UBound(y1), mmin)-(0.5 * UBound(y1), MMAX)
fo.Picture1.Line (0.25 * UBound(y1), mmin)-(0.25 * UBound(y1), MMAX)
fo.Picture1.Line (0.75 * UBound(y1), mmin)-(0.75 * UBound(y1), MMAX)
Rem Ось ОХ
67 fo.Picture1.DrawWidth = 3
For k = LBound(y1) To UBound(y1) Step 0.1
fo.Picture1.PSet (k, 0), QBColor(0)
Next
Rem ВЫВОД ГРАФИКОВ ФУНКЦИЙ
fo.Picture1.DrawWidth = 3
For K1 = LBound(y1) + 1 To UBound(y1) Step 0.1
K2 = K1 + 1:If K2 > UBound(y1) Then K2 = UBound(y1)
fo.Picture1.Line (K1, y1(K1))-(K2, y1(K2)), QBColor(color)
fo.Picture1.Line (K1, y2(K1))-(K2, y2(K2)), QBColor(color + 1)
fo.Picture1.Line (K1, y3(K1))-(K2, y3(K2)), QBColor(color + 2)
Next
Rem ЭКСТРЕМУМЫ ФУНКЦИИ
fo.CurrentX = fo.Picture1.Left - 300
fo.CurrentY = fo.Picture1.Top + MMAX * (fo.Picture1.Height)/_
Abs(MMAX) + Abs(mmin))
fo.Print "0":fo.CurrentX = fo.Picture1.Left - 300
fo.CurrentY = fo.Picture1.Top - 300:fo.Print Format(MMAX, "000.000")
fo.CurrentX = fo.Picture1.Left - 300
fo.CurrentY = fo.Picture1.Top + fo.Picture1.Height + 300
fo.Print Format(mmin, "000.000")
End Sub

```

1.5.8. Контрольные вопросы по силовому анализу рычажных механизмов

1. В чем состоит задача силового расчета рычажного механизма?
2. Каково назначение подпрограмм *PKT*, *PA*, *PMY*? Опишите входные и выходные параметры этих подпрограмм.
3. Каково назначение подпрограммы *PBC*? Опишите входные и выходные параметры этой подпрограммы.
4. Объясните понятия главного вектора сил и момента сил.
5. Каково назначение подпрограммы-функции *FNPR*? Опишите входные и выходные параметры этой подпрограммы-функции.

6. Каково назначение подпрограммы *PC1*? Опишите входные и выходные параметры этой подпрограммы.
7. Каким образом определяется величина силы \bar{P}_B ?
8. Каково назначение подпрограммы *PC2*? Опишите входные и выходные параметры этой подпрограммы.
9. В чем разница между подпрограммами *PC1* и *PC2*?
10. Каково назначение подпрограммы *PC3*? Опишите входные и выходные параметры этой подпрограммы.
11. Как определяется точка приложения реакции \bar{R}_{32} ?
12. Каково назначение подпрограммы *PC4*? Опишите входные и выходные параметры этой подпрограммы.
13. Как определяется точка приложения реакций R_{1-2} и R_{4-3} ?
14. Каково назначение подпрограммы *PC5*? Опишите входные и выходные параметры этой подпрограммы.
15. Какие силовые факторы действуют на входное звено рычажного механизма?
16. В чем заключается задача силового анализа входного звена?
17. Перечислите исходные данные для силового анализа входного звена.

РАЗДЕЛ 2

МАТЕМАТИЧЕСКИЕ МОДЕЛИ, ИСПОЛЬЗУЕМЫЕ ПРИ АНАЛИЗЕ КУЛАЧКОВЫХ МЕХАНИЗМОВ

Кулачковые механизмы широко применяются в основных исполнительных механизмах машин текстильной и легкой промышленности: в батанных и рапирных; зевообразования ткацких станков; съемного гребня в чесальной машине; мотальных в прядильной машине; перемещения материала, отклонения иглы в швейных машинах и др. Для точного воспроизведения заданного закона движения толкателя [6] требуется точное проектирование профилей кулачков, что достигается только аналитическим методом. В зависимости от технологии изготовления необходимо знать координаты точек профиля кулачка в полярной и прямоугольной системах координат.

Рассмотрим проектирование профиля кулачка для наиболее применяемых в текстильном и швейном оборудовании плоских кулачковых механизмов с толкателями, совершающими возвратно-вращательное (качательное) движение, и с толкателями, совершающими возвратно-поступательное движение, а также пространственных кулачковых механизмов с возвратно-вращательным движением толкателя.

2.1. Проектирование профиля плоского кулачка с возвратно-вращательным движением толкателя

2.1.1. Расчетная схема механизма

На рис. 2.1 показана схема кулачкового механизма с качающимся толкателем. Для унификации расчета схему кулачкового механизма представляют в правой прямоугольной системе координат следующим образом:

- начало координат принимают в центре вращения кулачка;
- центр вращения толкателя – на положительной полуоси OX ;

- толкатель устанавливают в крайнее положение, соответствующее началу фазы удаления, с расположением центра ролика в первом квадранте системы координат. При таком расположении механизма возможно вращение кулачка: 1) по ходу часовой стрелки; 2) против хода часовой стрелки.

Для дальнейших расчетов введем параметр Z , определяющий направление вращения кулачка: в случае 1) $Z = 1$, в случае 2) $Z = -1$.

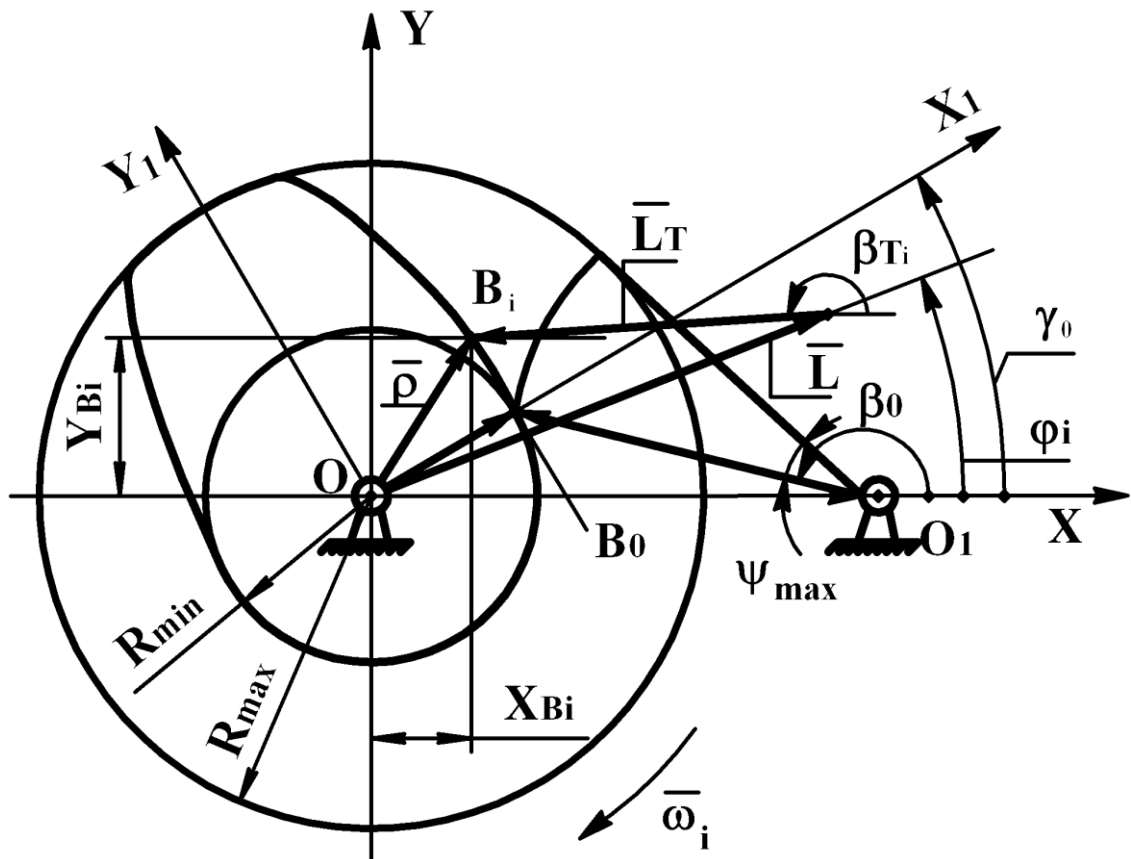


Рис. 2.1. Расчетная схема кулачкового механизма с качающимся толкателем

2.1.2. Определение входных и постоянных параметров кулачкового механизма

Для решения задачи проектирования профиля кулачка необходимы следующие входные параметры: L_1 – межосевое расстояние OO_1 ; L_T – длина толкателя BO_1 ; R_{\min} – минимальный радиус кулачка; $\varphi_{y\partial}$, $\varphi_{\partial.ct}$, $\varphi_{c\partial}$, $\varphi_{\partial.ct}$ – фазовые углы поворота кулачка, град; ψ_{\max} – максимальный ход толкателя, рад; $\psi = \psi(\varphi)$ – законы движения толкателя в фазах удаления и сближения;

ψ' , ψ'' – законы изменения аналогов угловых скоростей и ускорений толкателя в фазе удаления и сближения; Z – параметр, задающий направления вращения; $\Delta\varphi$ – шаг расчета, град; C – параметр, определяющий наличие ($C = 1$) или отсутствие ($C = 0$) контркулачка. В некоторых случаях вместо параметра ψ_{\max} задается R_{\max} – максимальный радиус кулачка, тогда ψ_{\max} рассчитывается.

Определим расчетные постоянные параметры кулачкового механизма. Рассмотрим начальный момент времени (см. рис. 2.1), когда толкатель касается профиля кулачка в точке B_0 . В этой точке радиус профиля кулачка соответствует минимальной величине R_{\min} .

Начальное положение вектора \bar{L}_T координируется углом β_0 . Из векторного треугольника OB_0O_1 (см. рис. 2.1) получим

$$\bar{R}_{\min} = \bar{L} + \bar{L}_T. \quad (2.1)$$

Возведем выражение (2.1) в скалярный квадрат и, разрешая относительно параметра β_0 , получим

$$\beta_0 = \arccos\left(\frac{R_{\min}^2 - L^2 - L_T^2}{2 \cdot L \cdot L_T}\right). \quad (2.2)$$

Если задана величина R_{\max} , необходимо рассчитать значение ψ_{\max} (см. рис. 2.1) по формуле

$$\psi_{\max} = \beta_0 - \beta_{\max}. \quad (2.3)$$

Предварительно следует определить угол β_{\max} , координирующий положение толкателя в момент времени, когда толкатель касается точки профиля кулачка максимального радиуса:

$$\beta_{\max} = \arccos\left(\frac{R_{\max}^2 - L^2 - L_T^2}{2 \cdot L \cdot L_T}\right). \quad (2.4)$$

При наличии контркулачка (рис. 2.2) необходимо определить величины θ_T – угла между толкателями кулачка и контркулачка и θ_{cd} – угла сдвига между осями, от которых отсчитываются координаты профиля кулачка и контркулачка:

$$\theta_T = 2 \cdot \pi - 2 \cdot \beta_0 + \psi_{\max}; \quad (2.5)$$

$$\theta_{cd} = \gamma_0 + \gamma_{\max}. \quad (2.6)$$

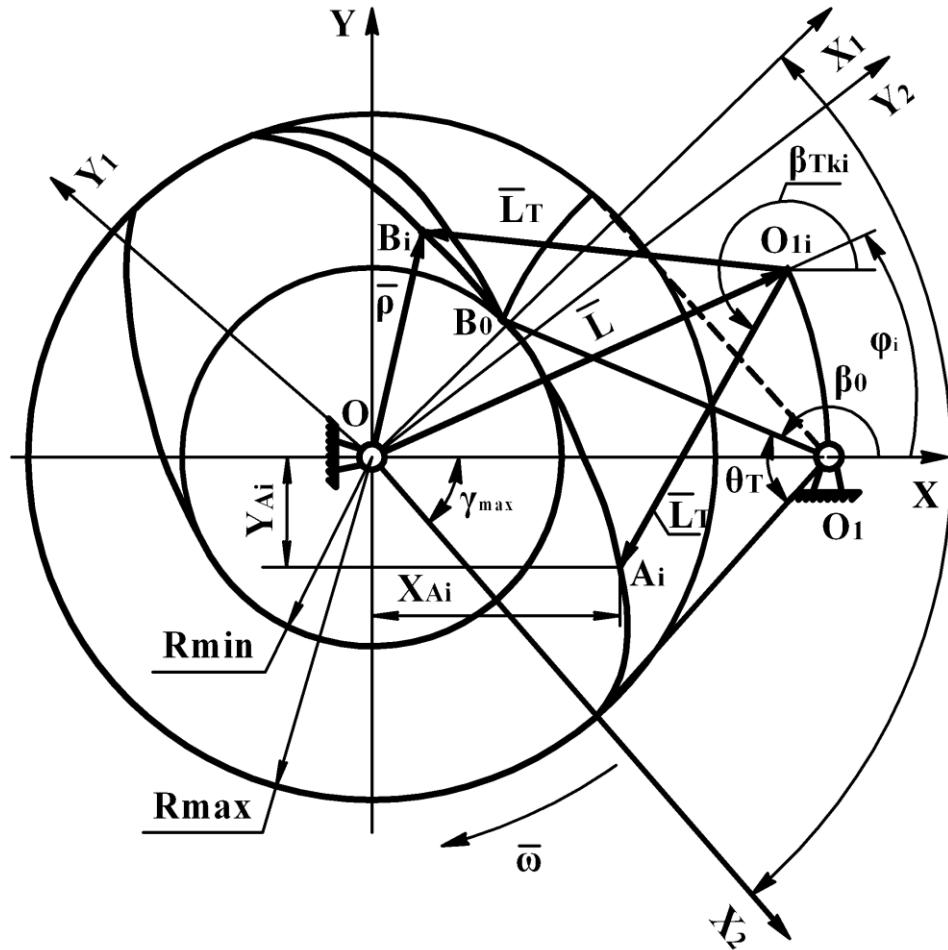


Рис. 2.2. Расчетная схема контркулачка

В выражении (2.6) $\gamma_0 = \arctg\left(\frac{l_T \sin \beta_0}{L + l_T \cos \beta_0}\right);$ (2.7)

$$\gamma_{\max} = \arctg\left(\frac{l_T \sin \beta_{\max}}{L + l_T \cos \beta_{\max}}\right). \quad (2.8)$$

2.1.3. Определение координат точек теоретического профиля кулачка

Для определения координат произвольной точки B_i профиля кулачка воспользуемся методом обращения движения [5].

Поворачиваем векторный треугольник OO_1B_0 (см. рис. 2.1) против хода часовой стрелки (при $Z = 1$) на некоторый угол φ_i , тогда точка B толкателя скользит по профилю B_0B_i и поворачивает толкатель на некоторый угол ψ_i .

Таким образом, направление вектора \bar{L}_T будет определяться углом β_{T_i} :

$$\beta_{T_i} = \beta_0 - \psi_i + Z \cdot \varphi_i. \quad (2.9)$$

Вектор \bar{L} будет определяться углом φ_i , а вектор $\bar{\rho}$ – некоторым углом α_{Ti} .

Уравнение замкнутого контура OB_iO_1O имеет вид

$$\bar{\rho} = \bar{L} + \bar{L}_T. \quad (2.10)$$

Проекцией на оси OX и OY уравнения (2.10) получим

$$\left. \begin{aligned} X_{Bi} &= L \cos(Z \cdot \varphi_i) + L_T \cos \beta_{Ti}, \\ Y_{Bi} &= L \sin(Z \cdot \varphi_i) + L_T \sin \beta_{Ti}. \end{aligned} \right\} \quad (2.11)$$

Система уравнений (2.11) определяет прямоугольные координаты профиля кулачка в системе координат XOY .

На практике удобно использовать систему координат X_1OY_1 , повернутую на угол γ_0 по отношению к системе координат XOY .

Методом преобразования координат

$$\left. \begin{aligned} X_{1i} &= X_{Bi} \cos \gamma_0 + Y_{Bi} \sin \gamma_0, \\ Y_{1i} &= -X_{Bi} \sin \gamma_0 + Y_{Bi} \cos \gamma_0. \end{aligned} \right\} \quad (2.12)$$

В случае полярной системы координат

$$\left. \begin{aligned} \rho_{1i} &= \sqrt{X_{1i}^2 + Y_{1i}^2}, \\ \alpha_{1i} &= \arccos\left(\frac{X_{1i}}{\rho_{1i}}\right). \end{aligned} \right\} \quad (2.13)$$

Если в системе (2.13) выполняется условие

$$Y_{1i} < 0, \quad (2.14)$$

$$\text{то следует считать} \quad \alpha_{1i} = 2 \cdot \pi - \alpha_{1i}. \quad (2.15)$$

При наличии контркулачка его толкатель (см. рис. 2.2) в начале движения будет касаться профиля в точке A_0 .

При обращении движения на угол φ_i , когда точка B толкателя кулачка будет находиться в точке B_i , точка A толкателя контркулачка будет находиться в точке A_i .

Направление вектора $\overline{O_{1i}A_i}$, равного по модулю L_T , будет определяться углом

$$\beta_{tki} = \beta_{ti} + \theta_T. \quad (2.16)$$

На основании выражений (2.11)

$$\left. \begin{aligned} X_{Ai} &= L \cos(Z \cdot \varphi_i) + L_T \cos \beta_{Tki}, \\ Y_{Ai} &= L \sin(Z \cdot \varphi_i) + L_T \sin \beta_{Tki}. \end{aligned} \right\} \quad (2.17)$$

В системе координат X_2OY_2 , повернутой относительно системы XOY на угол γ_{\max} , уравнения для определения координат будут иметь вид

$$\left. \begin{aligned} X_{2i} &= X_{Ai} \cos \gamma_{\max} + Y_{Ai} \sin \gamma_{\max}, \\ Y_{2i} &= X_{Ai} \sin \gamma_{\max} + Y_{Ai} \cos \gamma_{\max}. \end{aligned} \right\} \quad (2.18)$$

В случае полярной системы координат для контркулачка

$$\left. \begin{aligned} \rho_{2i} &= \sqrt{X_{2i}^2 + Y_{2i}^2}, \\ \alpha_{2i} &= \arccos\left(\frac{X_{2i}}{\rho_{2i}}\right). \end{aligned} \right\} \quad (2.19)$$

Если в системе (2.19) выполняется условие

$$Y_{2i} < 0, \quad (2.20)$$

то следует считать $\alpha_{2i} = 2 \cdot \pi - \alpha_{2i}$. (2.21)

Для расчета координат теоретического профиля кулачка и контркулачка с возвратно-вращательным движением толкателя в декартовой и полярной системах координат по приведенной методике предназначена подпрограмма (процедура) *KOORDPROF1*, работающая в среде *Visual Basic*.

Входные параметры подпрограммы: $B0 = \beta_0$; $F9 = \varphi_i$; $T0 = \theta_T$; $G0 = \gamma_0$; $S = \psi_i$.

Выходные параметры: $X0 = X_{1i}$; $Y0 = Y_{1i}$; $R0 = \rho_{1i}$; $AlfaR0 = \alpha_{1i}$.

Код программы (процедуры) *KOORDPROF1*:

KULAK.BAS	KOORDPROF1
Sub KOORDPROF1(B0 As Double, F9 As Double, T0 As Double, _ G0 As Double, S As Double, X0 As Double, Y0 As Double, R0 As Double, _ AlfaR0 As Double) ' ПОДПРОГРАММА РАСЧЕТА КООРДИНАТ ' КРИВОЛИНЕЙНОГО ПРОФИЛЯ Dim arg As Double, B As Double B = B0 - S + RAD * F9 + T0: x = L * Cos(F9 * RAD) + L1 * Cos(B) y = L * Sin(F9 * RAD) + L1 * Sin(B) X0 = x * Cos(G0) + y * Sin(G0): Y0 = -x * Sin(G0) + y * Cos(G0) R0 = Sqr(X0 ^ 2 + Y0 ^ 2) : arg = X0 / R0	

```

If Abs(arg) > 1 Then arg = 1
A9 = DEG * FNARC(arg)
If arg < 0 Then A9 = A9 + 180
If Y0 > 0 Then GoTo 5810
AlfaR0 = 360 - A9
5810 T1 = Int(A9): M1 = (A9 - T1) * 60: T2 = Int(M1)
M2 = (M1 - T2) * 60: t3 = Int(M2)
End Sub

```

Поскольку проектирование кулачковых механизмов – достаточно сложная задача, для ее решения требуется большое количество подпрограмм. Они размещены в отдельном модуле *KULAK.BAS*.

2.1.4. Определение радиуса кривизны и координат центра кривизны профиля кулачка в системе XOY

Системы уравнений (2.11) и (2.18) представляют собой параметрические уравнения плоских кривых. В нашем случае параметром является угол поворота кулачка φ .

Согласно теории о плоских кривых [7], радиус кривизны профиля кулачка определяется по формуле

$$R_{кр} = \frac{\sqrt{((X'_{Bi})^2 + (Y'_{Bi})^2)^3}}{X'_{Bi}Y''_{Bi} - X''_{Bi}Y'_{Bi}}. \quad (2.22)$$

Найдем производные X'_{Bi} , Y'_{Bi} , X''_{Bi} , Y''_{Bi} , для чего продифференцируем систему (2.11) по обобщенной координате φ :

$$\left. \begin{aligned} X'_{Bi} &= -Z \cdot L \sin(z\varphi_i) + L_T \frac{d\beta_{Ti}}{d\varphi} \sin \beta_{Tki}, \\ Y'_{Bi} &= ZL \cos(z\varphi_i) + L_T \frac{d\beta_{Ti}}{d\varphi} \cos \beta_{Tki}. \end{aligned} \right\} \quad (2.23)$$

Выполним подстановку:

$$\frac{d\beta_{Ti}}{d\varphi} = \beta'_{Ti}. \quad (2.24)$$

Тогда на основании выражения (2.9) перепишем выражение (2.24):

$$\beta'_{Ti} = -\psi'_i + Z. \quad (2.25)$$

Дифференцированием выражения (2.25) по обобщенной координате φ получим

$$\beta''_{Ti} = -\psi''_i. \quad (2.26)$$

Дифференцированием уравнений (2.23) определим

$$\left. \begin{aligned} X''_{Bi} &= L \cos(Z \cdot \varphi_i) - (\beta'_{Ti})^2 L_T \cos \beta_{Ti} - \beta''_{Ti} \cdot L_T \sin \beta_{Ti}, \\ Y''_{Bi} &= -L \sin(Z \cdot \varphi_i) - (\beta'_{Ti})^2 L_T \sin \beta_{Ti} + \beta''_{Ti} \cdot L_T \cos \beta_{Ti}. \end{aligned} \right\} \quad (2.27)$$

Координаты центра кривизны определяются по формулам

$$\left. \begin{aligned} X_{ki} &= X_{Bi} - Y_{Bi} \frac{(X'_{Bi})^2 + (Y'_{Bi})^2}{X'_{Bi} \cdot Y''_{Bi} - X''_{Bi} \cdot Y'_{Bi}}, \\ Y_{ki} &= Y_{Bi} + X'_{Bi} \frac{(X'_{Bi})^2 + (Y'_{Bi})^2}{X'_{Bi} \cdot Y''_{Bi} - X''_{Bi} \cdot Y'_{Bi}}. \end{aligned} \right\} \quad (2.28)$$

2.1.5. Определение углов давления

Угол давления кулачка определяется из выражений

$$\alpha_{\partial i} = \beta_{Ti} - \alpha_{ni}; \quad (2.29)$$

$$\alpha_{ni} = \arccos \frac{X'_{Bi}}{\sqrt{(X'_{Bi})^2 + (Y'_{Bi})^2}}. \quad (2.30)$$

$$\text{Если в формуле (2.30) выполняется условие } Y'_{Bi} < 0, \quad (2.31)$$

$$\text{то следует считать} \quad \alpha_{ni} = 2\pi - \alpha_{ni}. \quad (2.32)$$

Для расчета радиуса кривизны, координат центра кривизны и угла давления кулачка с возвратно-вращательным движением толкателя предназначена подпрограмма *UGLIDAVL1*.

Входные параметры подпрограммы:

$$B0 = \beta_0; F9 = \varphi_i; T0 = \theta_T; S = \psi_i; S1 = \psi'_i; S2 = \psi''_i.$$

Выходные параметры:

$$r = R_{kp}; x3 = X_{ki}; y3 = Y_{ki}; d0 = \alpha_{\partial i}.$$

KULAK.BAS	UGLIDAVL1
Sub UGLIDAVL1(B0 As Double, F9 As Double, T0 As Double, _ S As Double, S1 As Double, S2 As Double, r As Double, X3 As Double, _ Y3 As Double, D0 As Double) 'ПОДПРОГРАММА РАСЧЕТА РАДИУСА КРИВИЗНЫ И УГЛОВ 'ДАВЛЕНИЯ КУЛАЧКОВОГО МЕХАНИЗМА 'С ВОЗВРАТНО-ВРАЩАТЕЛЬНЫМ ТОЛКАТЕЛЕМ B = B0 - S + RAD * F9 + T0 x = L * Cos(RAD * F9) + L1 * Cos(B) y = L * Sin(RAD * F9) + L1 * Sin(B) X1 = -ZZ * L * Sin(RAD * F9) - L1 * (-S1 + ZZ) * Sin(B) y1 = ZZ * L * Cos(RAD * F9) + L1 * (-S1 + ZZ) * Cos(B) X2 = -L * Cos(RAD * F9) - L1 * (-S1 + ZZ) ^ 2 * Cos(B) + S2 * L1 * Sin(B) y2 = -L * Sin(RAD * F9) - L1 * (-S1 + ZZ) ^ 2 * Sin(B) - S2 * L1 * Cos(B) d = (X1 ^ 2 + y1 ^ 2) / (X1 * y2 - X2 * y1) X3 = x - y1 * d: Y3 = y + X1 * d: r = Sqr((x - X3) ^ 2 + (y - Y3) ^ 2) arg2 = X1 / (Sqr(X1 ^ 2 + y1 ^ 2)): A0 = Atn(Sqr(1 - arg2 ^ 2) / arg2) If arg2 < 0 Then A0 = A0 + PI If y1 < 0 Then A0 = 2 * PI - A0 If B < 0 Then B = 2 * PI + B D0 = B - A0 + 0.5 * PI * (1 - ZZ) If T0 > 0 Then D0 = B - A0 - 0.5 * PI * (1 + ZZ) If (Abs(D0)) > 1.5 * PI Then D0 = -Sgn(D0) * (2 * PI - Abs(D0)) End Sub	

2.2. Проектирование профиля плоского кулачка с возвратно-поступательным движением толкателя

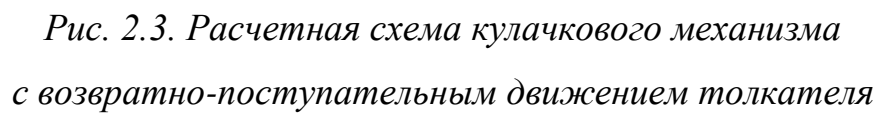
2.2.1. Определение координат профиля кулачка

На рис. 2.3 показана схема кулачкового механизма. Возьмем на профиле произвольную точку B . Когда точка B подойдет к толкателю, кулачок повернется на угол φ , а толкатель переместится на величину S_B . В подвижной системе координат X_1OY_1 , связанной с поворотом кулачка, координаты точки B

$$\left. \begin{aligned} X_B &= E; \\ Y_B &= B_0 + S_B, \end{aligned} \right\} \quad (2.33)$$

где S_B – закон движения толкателя.

Из рис. 2.3
$$B_0 = \sqrt{R_{\min}^2 - E^2}. \quad (2.34)$$


$$\left. \begin{aligned} X_B &= X_{1B} \cos(Z \cdot \varphi) - Y_{1B} \sin(Z \cdot \varphi), \\ Y_B &= X_{1B} \sin(Z \cdot \varphi) + Y_{1B} \cos(Z \cdot \varphi). \end{aligned} \right\} \quad (2.35)$$
$$\gamma_0 = \arctg(\beta_0/E); \quad (2.36)$$

$$\gamma_{\max} = \arctg((\beta_0 + S_{\max})/E); \quad (2.37)$$

$$\gamma_{\max} = \arctg(Y_B / X_B); \quad (2.38)$$

$$\rho = \sqrt{X_B^2 + Y_B^2}; \quad (2.39)$$

$$\alpha = \varphi + Z(\gamma - \gamma_0). \quad (2.40)$$

Для определения радиуса кривизны профиля имеем

$$X'_B = X''_B = 0; \quad (2.41)$$

$$Y'_B = S'_B; \quad (2.42)$$

$$Y''_B = S''_B. \quad (2.43)$$

Для дальнейших расчетов используются уравнения (2.22), (2.23), (2.27) и (2.28), которые показаны в подразд. 2.1.4. При этом в отличие от подразд. 2.1.5 для определения угла давления применяется формула

$$\operatorname{tg} \alpha_0 = \frac{S'_B + E}{S_B + B_0}. \quad (2.44)$$

При проектировании различных кулачков стоят одинаковые задачи, и для расчета указанных параметров кулачка с возвратно-поступательным движением толкателя предназначены аналогичные подпрограммы.

1. Подпрограмма *KORDPROF2*

Входные параметры подпрограммы: $B_0 = \beta_0$; $S = \psi_i$; $F = \varphi_i$.

Выходные параметры: $xb = X_B$; $yb = Y_B$; $RB = \rho$; $AlfaRB = \alpha$.

KULAK.BAS	KORDPROF2
Sub KORDPROF2(B0 As Double, S As Double, F As Double, _ XB As Double, YB As Double, RB As Double, AlfaRB As Double) 'ПОДПРОГРАММА РАСЧЕТА КООРДИНАТ ТОЧКИ ПРОФИЛЯ 'КУЛАЧКА КУЛАЧКОВОГО МЕХАНИЗМА 'С ПОСТУПАТЕЛЬНО ДВИЖУЩИМСЯ ТОЛКАТЕЛЕМ XB = e * Cos(F) - (B0 + S) * Sin(F) YB = e * Sin(F) + (B0 + S) * Cos(F) rb = Sqr(XB ^ 2 + YB ^ 2) If e = 0 And B0 > 0 Then G0 = PI / 2 Else G0 = Atn(B0 / e) If XB = 0 And YB > 0 Then G = PI / 2: GoTo 2500 If XB = 0 And YB < 0 Then G = -PI / 2: GoTo 2500 G = Atn(YB / XB) 2500 AB = (i - 1) * F0 + Z * (G - G0): A9 = AB: T1 = Int(A9) End Sub	

2. Подпрограмма UGLIDAVL2

Входные параметры подпрограммы: $B0 = \beta_0$; $S = \psi_i$; $S1 = \psi'_i$; $S2 = \psi''_i$;

$F = \varphi_i$.

Выходные параметры: $R0 = R_{kp}$; $X0 = X_{ki}$; $Y0 = Y_{ki}$; $d10 = \alpha_{di}$.

KULAK.BAS	UGLIDAVL2
Sub UGLIDAVL2(B0 As Double, S As Double, S1 As Double, _ S2 As Double, RO As Double, XO As Double, YO As Double, _ d1o As Double) $XBB = e * \cos(F) - (B0 + S) * \sin(F)$; $YBB = e * \sin(F) + (B0 + S) * \cos(F)$ $X1BB = -S1 * \sin(F) - Z * YBB$; $Y1BB = S1 * \cos(F) + Z * XBB$ $X2BB = -S2 * \sin(F) - 2 * Z * Y1BB + XBB$; $Y2BB = S2 * \cos(F) + 2 * Z * X1BB + YBB$ $y1 = Y1BB / X1BB$; $y2 = (X1BB * Y2BB - X2BB * Y1BB) / (X1BB ^ 3)$ $XO = XBB - ((y1 * (1 + y1 ^ 2)) / y2)$; $YO = YBB + ((1 + y1 ^ 2) / y2)$ $RO = \text{Sqr}((XBB - XO) ^ 2 + (YBB - YO) ^ 2)$; $d1o = \text{Atn}((S1 + e) / (S + B0))$ If (S1 + e) = 0 Then d1o = 0 End Sub	

2.2.2. Входные параметры при расчете кулачкового механизма на ЭВМ

Для проведения указанных выше расчетов кулачкового механизма с возвратно-поступательным движением толкателя необходимо знать следующие входные параметры: R_{\min} – минимальный радиус кулачка, м; S_{\max} – максимальный ход толкателя, м; E – эксцентриситет, м; $S = S(\varphi)$ – закон движения толкателя; Z – направление вращения кулачка; H – шаг расчета, град; $\varphi_{y\partial}$, $\varphi_{\partial.cm}$, $\varphi_{c\partial}$, $\varphi_{\partial.cm}$ – фазовые углы поворота кулачка, град.

2.3. Проектирование профиля пространственного кулачка с возвратно-вращательным движением толкателя

2.3.1. Определение координат профиля пространственного кулачка

Расчетная схема пространственного кулачкового механизма представлена на рис. 2.4.

В начальный момент толкатель с горизонтальной осью составляет угол β_0 , величина которого определяется по формуле

$$\beta_0 = \arccos((L_B - L_A)/L_T). \quad (2.45)$$

Примем подвижную систему координат XO_1Y , у которой ось O_1Y совпадает с осью вращения кулачка.

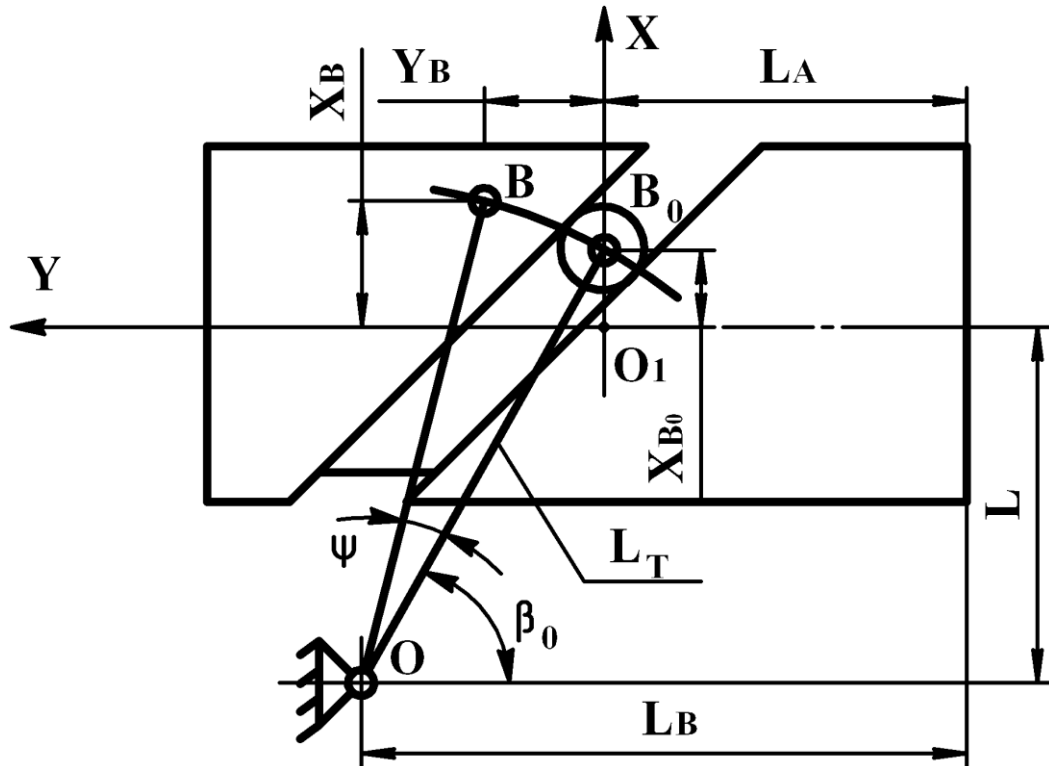


Рис. 2.4. Расчетная схема пространственного кулачкового механизма:

L – межосевое расстояние; L_B – расстояние от торца кулачка до центра вращения толкателя;

L_T – длина толкателя; L_A – расстояние от торца кулачка до центра ролика,

когда он находится в крайнем правом положении

В начальный момент координаты оси B ролика

$$\left. \begin{aligned} Y_{B_0} &= 0, \\ X_{B_0} &= L_T \sin \beta_0 - L. \end{aligned} \right\} \quad (2.46)$$

Допустим, что закон движения как функция угла поворота кулачка известен, тогда при повороте кулачка на угол φ толкатель повернется на некоторый угол ψ .

Координаты точки B толкателя:

- в системе XO_1Y

$$\left. \begin{aligned} X_B &= L_T \sin(\beta_0 + \psi) - L \\ Y_B &= L_T \cos \beta_0 = L_T \cos(\beta_0 + \psi) \end{aligned} \right\}; \quad (2.47)$$

- в полярной системе

$$\rho_B = \sqrt{(X_B)^2 + (Y_B)^2}; \quad (2.48)$$

$$\alpha_B = \arctg\left(\frac{X_B}{\rho_B}\right), \quad (2.49)$$

где ρ_B – радиус-вектор;

α_B – угол, координирующий радиус-вектор.

2.3.2. Определение угла давления и радиуса кривизны теоретического профиля кулачка

Угол давления в пространственном кулачке определяется по формуле

$$\operatorname{tg} \alpha_D = \frac{L_T \psi' / R_{cp} + \cos(\beta_0 + \psi)}{\sin(\beta_0 + \psi)}. \quad (2.50)$$

Координаты центра кривизны и радиус кривизны профиля определяются по формулам (2.22), (2.23), (2.27) и (2.28) с учетом

$$\left. \begin{aligned} X_B &= x_B - x_{B_0} + R_{cp} \cdot \varphi, \\ Y_B &= y_B. \end{aligned} \right\} \quad (2.51)$$

$$\left. \begin{aligned} X'_B &= x'_B + R_{cp}, \\ Y'_B &= y'_B. \end{aligned} \right\} \quad (2.52)$$

$$\left. \begin{aligned} X''_B &= x''_B, \\ Y''_B &= y''_B. \end{aligned} \right\} \quad (2.53)$$

Дважды проведем дифференцирование по φ системы (2.47):

$$\left. \begin{aligned} x'_B &= L_T \cdot \psi' \cos(\beta_0 + \psi), \\ y'_B &= L_T \cdot \psi' \sin(\beta_0 + \psi). \end{aligned} \right\} \quad (2.54)$$

$$\left. \begin{aligned} x''_B &= L_T \cdot \psi'' \cos(\beta_0 + \psi) - L_T (\psi')^2 \sin(\beta_0 + \psi), \\ y''_B &= L_T \cdot \psi'' \sin(\beta_0 + \psi) + L_T (\psi')^2 \cos(\beta_0 + \psi). \end{aligned} \right\} \quad (2.55)$$

Поскольку данная методика предназначена для решения задачи, аналогичной расчету кулачкового привода возвратно-качательного типа, то составим две подпрограммы.

1. *KOORDPROF3* – для расчета координат профиля пространственного кулачка.

Входные параметры подпрограммы: $B0 = \beta_0$; $S = \psi_i$.

Выходные параметры: $XB = X_B$; $YB = Y_B$; $RB = \rho_B$; $AlfaRB = \alpha_B$.

KULAK.BAS	KOORDPROF3
Sub KOORDPROF3(B0 As Double, S As Double, XB As Double, _ YB As Double, rb As Double, AB As Double) Dim arg As Double 3830 XB = L1 * Sin(B0 + S) - L: YB = L1 * Cos(B0) - L1 * Cos(B0 + S) rb = Sqr(XB ^ 2 + YB ^ 2):arg = XB / rb : AB = DEG * FNARC(arg) If arg < 0 Then AB = AB + 180 If YB > 0 Then GoTo 3880 AB = 360 - AB 3880 End Sub	

2. *UGLIDAVL3* – для расчета радиуса кривизны и угла давления пространственного кулачка.

Входные параметры подпрограммы: $B0 = \beta_0$; $S = \psi_i$; $S1 = \psi'_i$; $S2 = \psi''_i$.

Выходные параметры: $R0 = R_{kp}$; $X0 = X_B$; $Y0 = Y_B$; $d10 = \alpha_\partial$.

KULAK.BAS	UGLIDAVL3
Sub UGLIDAVL3(B0 As Double, S As Double, S1 As Double, _ S2 As Double, RO As Double, XO As Double, YO As Double, _ d1o As Double) XB = L1 * Sin(B0 + S) - L: YB = L1 * Cos(B0) - L1 * Cos(B0 + S) XBB = XB - XB0 + rs * F: YBB = YB X1BB = L1 * S1 * Cos(B0 + S) + rs Y1BB = L1 * S1 * Sin(B0 + S) X2BB = L1 * S2 * Cos(B0 + S) - L1 * S1 ^ 2 * Sin(B0 + S) Y2BB = L1 * S2 * Sin(B0 + S) + L1 * S1 ^ 2 * Cos(B0 + S) y1 = Y1BB / X1BB y2 = (X1BB * Y2BB - X2BB * Y1BB) / (X1BB ^ 3) XO = XBB - ((y1 * (1 + y1 ^ 2)) / y2) YO = YBB + ((1 + y1 ^ 2) / y2) RO = Sqr((XBB - XO) ^ 2 + (YBB - YO) ^ 2) d1o = Atn((L1 * S1 / rs + Cos(B0 + S)) / Sin(B0 + S)) End Sub	

2.3.3. Входные параметры при расчете пространственного кулачкового механизма на ЭВМ

Для проведения указанных выше расчетов пространственного кулачкового механизма необходимо знать следующие входные параметры: L – межосевое расстояние, м; L_T – длина толкателя, м; L_B – расстояние от правого торца кулачка до центра вращения толкателя, м; R_{cp} – средний радиус кулачка, м; ψ_{\max} – максимальный угол перемещения толкателя, град; $\psi = \psi(\varphi)$ – закон движения толкателя; $\varphi_{y\partial}$, $\varphi_{\partial.ct}$, $\varphi_{c\bar{b}}$, $\varphi_{\bar{b}.ct}$ – фазовые углы поворота кулачка, град; H – шаг расчета, град.

2.4. Законы движения толкателей

Выбор закона движения толкателя предоставляется самому постановщику задачи по проектированию данного кулачкового механизма. В [6] законы движения толкателя, их уравнения в большинстве случаев задаются только для фазы удаления. Законы движения толкателя на фазе удаления и на фазе сближения могут быть разными как по форме, так и по продолжительности фаз. Зная закон движения на фазе удаления как функцию безразмерного параметра, представляющего собой отношение текущего угла поворота кулачка к фазовому углу кулачка, можно всегда найти уравнение для фазы сближения.

Расчет на ЭВМ фазы сближения удобнее начинать с конца фазы. При этом получим

$$\psi_{c\bar{b}}(j) = \psi_{y\partial}(n + 1 - j), \quad (2.56)$$

где $j = 1, 2, 3, \dots, n$.

$$\text{Если } j = 1, \text{ то } \psi_{c\bar{b}}(1) = \psi_{y\partial}(n). \quad (2.57)$$

$$\text{Если } j = n, \text{ то } \psi_{c\bar{b}}(n) = \psi_{y\partial}(1). \quad (2.58)$$

Такой подход позволяет в программе расчета кулачков использовать уравнения законов движения толкателей только для фазы удаления и значи-

тельно уменьшить объем программы. Поскольку рассмотренные параметры кулачка являются функциями от величины φ_i , все расчеты необходимо проводить в цикле изменения этого параметра.

Установим границы требуемых циклов:

- число расчетных точек для фазы удаления N_1 или сближения N_3

$$N_{1,3} = \frac{\varphi_{фаз}}{\Delta\varphi} + 1, \quad (2.59)$$

где $\varphi_{фаз}$ – одна из фаз движения толкателя;

- число расчетных точек для фазы дальнего стояния

$$N_2 = \frac{\varphi_{д.ст}}{\Delta\varphi}; \quad (2.60)$$

- число расчетных точек для фаз удаления, дальнего стояния и сближения

$$N_3 = 2 \cdot N_1 + N_2 - 1; \quad (2.61)$$

- общее число расчетных точек

$$N_4 = \frac{360^\circ}{\Delta\varphi} + 1. \quad (2.62)$$

Коэффициенты пропорциональности [6] аналогов скоростей и ускорений соответственно

$$A_1 = \frac{180 \cdot \psi_{\max}}{\pi \cdot \varphi_{фаз}}; \quad (2.63)$$

$$A_2 = \frac{180^2 \cdot \psi_{\max}}{(\pi \cdot \varphi_{фаз})^2}. \quad (2.64)$$

Для расчета основных геометрических и силовых параметров кулачкового механизма с возвратно-вращательным движением толкателя необходимо иметь отдельную подпрограмму расчета постоянных величин с именем *CONSTVEL* (входными параметрами являются исходные данные).

Выходные параметры:

$Mr2 = R_{\max}$; $Mb0 = \beta_0$; $Mg0 = \gamma_0$; $Mg1 = \gamma_{\max}$; $MT = \theta_T$; $MT1 = \theta_{cd}$;

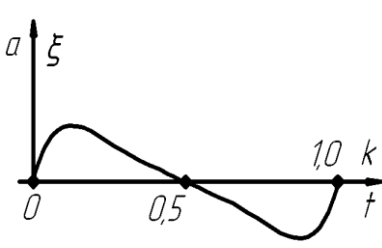
$MA1 = A_1$; $MA2 = A_2$; $MN1 = N_1$; $MN3 = N_3$.

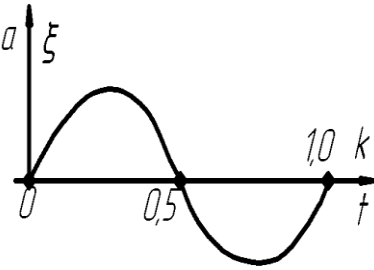
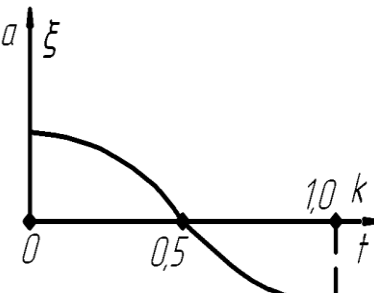
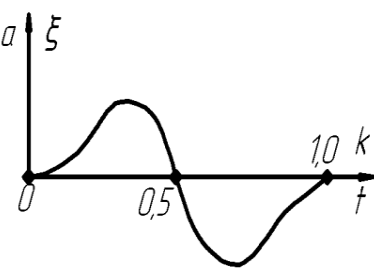
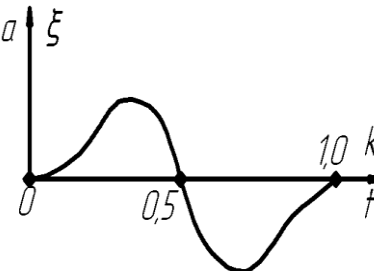
KULAK.BAS	CONSTVEL
Sub CONSTVEL (Mr2 As Double, Mb0 As Double, Mg0 As Double, _ Mg1 As Double, MT As Double, MT1 As Double, MA1 As Double, _ MA2 As Double, MN1 As Integer, MN3 As Integer) Dim argg As Double MM = RAD * M:argg = (R1^2-L^2-L1^2)/(2*L*L1) Mb0 = FNARC(argg) If MM > 0 Then GoTo 6530 argg = (R2^2-L^2-L1^2)/(2*L*L1): b1 = Atn(Sqr(1-argg^2)/argg) If argg < 0 Then b1 = b1 + PI MM = Mb0 - b1: GoTo 6540 6530 Mr2=Sqr(L^2 +L1^2+2*L*L1*Cos(Mb0-MM)):b1 = Mb0 - MM 6540 Mg0 = Atn(L1*Sin(Mb0)/(L + L1*Cos(Mb0))) Mg1 = Atn(L1*Sin(b1)/(L +L1*Cos(b1))) MT = DEG *(2*PI-2*Mb0 + MM) MT1 = DEG * (Mg0 + Mg1) 6670 MA1=MM/(RAD*F1): MA2 = MM/((RAD*F1))^2 MN1 = Int(F1/F0 + 1.01): MN3 = Int(F3/F0 +1.01) MK2 = MN1 + Int(F2/F0 +1.01) MK3 = MN1 + Int(F3/F0 +1.01) +MK2 End Sub	

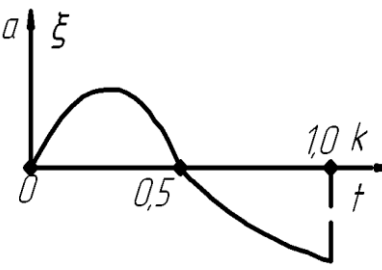
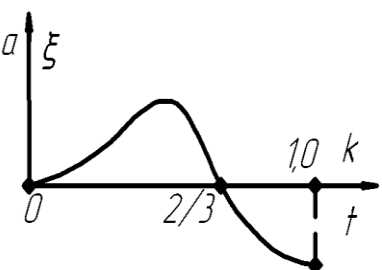
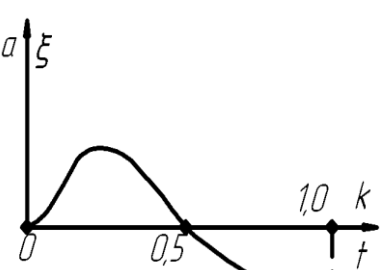
В настоящем пособии рассматриваются десять основных законов движения толкателя [6] (табл. 2.1). Все эти законы отражены в наборе соответствующих подпрограмм модуля *KULAK.BAS*.

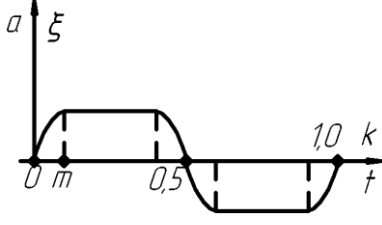
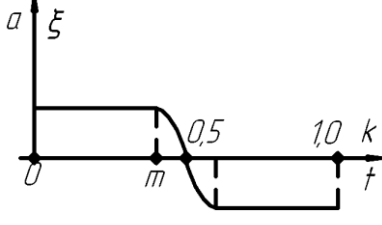
Таблица 2.1

Законы движения толкателя

№	Наименование закона и график ускорения	Пределы для φ_i	Коэффициенты пути ξ , скорости δ , ускорения ζ
1	Закон Шуна 	От 0 до $\varphi_{\text{фаз}}$	$\xi = 10k^3 - 15k^4 + 6k^5$ $\delta = 30(k^2 - 2k^3 + k^4)$ $\xi = 60(k - 3k^2 + 2k^3),$ <p>где $k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$</p>

1	2	3	4
2	<p>Синусоидальное ускорение</p> 	<p>От 0 до $\varphi_{\text{фаз}}$</p>	$\xi = k - \frac{\sin(2\pi k)}{2\pi}$ $\delta = 1 - \cos(2\pi k)$ $\xi = 2\pi \sin(2\pi k),$ <p>где $k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$</p>
3	<p>Косинусоидальное ускорение</p> 	<p>От 0 до $\varphi_{\text{фаз}}$</p>	$\xi = \frac{1 - \cos(\pi k)}{2}$ $\delta = \frac{\pi \sin(\pi k)}{2}$ $\xi = \frac{\pi^2 \cos(\pi k)}{2},$ <p>где $k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$</p>
4	<p>Степенной закон (4-5-6-7)</p> 	<p>От 0 до $\varphi_{\text{фаз}}$</p>	$\xi = 35k^4 - 84k^5 + 70k^6 - 20k^7$ $\delta = 140k^3(1 - 3k + 3k^2 - k^3)$ $\xi = 420k^2(1 - 4k + 5k^2 - 2k^3),$ <p>где $k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$</p>
5	<p>Степенной закон (5-6-7-8-9)</p> 	<p>От 0 до $\varphi_{\text{фаз}}$</p>	$\xi = 125k^5 - 420k^6 + 540k^7 - 315k^8 + 70k^9$ $\xi = 630k^4(1 - 4k + 65k^2 - 4k^3 + k^4)$ $\xi = 2520k^3(1 - 5k + 9k^2 - 7k^3 + 2k^4),$ <p>где $k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$</p>

1	2	3	4
6	<p>Закон Стоддарта</p> 	От 0 до $\varphi_{\text{фаз}}$	$\xi = 6,098k^3 - 20,78k^5 + 26,731k^6 - 13,61k^7 + 2,561k^8$ $\delta = 18,293k^2 - 103,902k^4 + 160,389k^5 - 953268k^6 + 20,488k^7$ $\xi = 36,585k - 415,608k^3 + 801,947k^4 - 571,605k^5 + 143,413k^6,$ <p>где $k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$</p>
7	<p>Двойной гармонический закон</p> 	От 0 до $\varphi_{\text{фаз}}$	$\xi = 0,5(1 - \cos(\pi k)) - \frac{(1 - \cos((2\pi k)))}{8}$ $\delta = 0,5\pi(\sin(\pi k) - 0,5\sin(2\pi k))$ $\xi = 0,5\pi^2(\cos(\pi k) - \cos(2\pi k)),$ <p>где $k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$.</p>
8	<p>Степенной закон (2-6-10-14-18)</p> 	От 0 до $\varphi_{\text{фаз}}$	$\xi = \frac{1}{128}(128 - 315(1-k)^2 + 420(1-k)^6 - 378(1-k)^{10} + 180(1-k)^{14} - 315(1-k)^{18})$ $\delta = \frac{1}{64}(315(1-k) - 1260(1-k)^5 + 1890(1-k)^9 - 1260(1-k)^9 - 1260(1-k)^{13} + 315(1-k)^{17})$ $\xi = \frac{5}{64}(63 + 1260(1-k)^4 - 3402(1-k)^8 + 3276(1-k)^{12} - 1071(1-k)^{16}),$ <p>где $k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$</p>

1	2	3	4
9	<p>Закон Неклютина для движения с выстоями</p> 	<p>$0 \dots m$</p> <p>$m \dots 0,5 - m$</p> <p>$0,5 - m \dots 0,5$</p>	$\varsigma = M_1 \cdot \frac{2 \cdot m}{\pi} \left[\frac{2 \cdot m \cdot k}{\pi} \sin \left(\frac{\pi \cdot k}{2 \cdot m} \right) \right]$ $\delta = M_1 \cdot \frac{2 \cdot m}{\pi} \left[1 - \cos \left(\frac{\pi \cdot k}{2 \cdot m} \right) \right]$ $\xi = M_1 \cdot \sin \left(\frac{\pi \cdot k}{2 \cdot m} \right), \text{ где } k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$ $M_1 = \frac{4 \cdot \pi}{\pi - 4 \cdot m \cdot \pi + 8 \cdot m}$ $\varsigma = M_1 \left[\frac{(\pi^2 - 8)m^2}{2 \cdot m} + \frac{(2 - \pi)m \cdot k}{\pi} + \frac{k^2}{2} \right]$ $\delta = M_1 \cdot \frac{(2 - \pi)m}{\pi} + k; \xi = M_1$ $\varsigma = M_1 \left[0,5m - \frac{1}{8} + k \left(0,5 - \frac{2m}{\pi} (\pi - 1) \right) - \left(\frac{2 \cdot m}{\pi} \right)^2 \sin \left(\frac{(0,5 - k)\pi}{2 \cdot m} \right) \right]$ $\delta = M_1 \left[0,5 - \frac{2m(\pi - 4)}{\pi} + \frac{2m}{\pi} \cos \left(\frac{0,5 - k}{2m} \pi \right) \right]$ $\xi = M_1 \sin \left(\frac{(0,5 - k)\pi}{2 \cdot m} \right)$
10	<p>Закон Неклютина для движения без выстоев</p> 	<p>$0 \dots m$</p> <p>$m \dots 0,5$</p>	$\varsigma = M_1 \cdot 0,5 \cdot k^2; \delta = M_1 \cdot k; \xi = M_1$ $\varsigma = M_1 \left[m \cdot k - 0,5 \cdot m^2 + 4 \left(\frac{0,5 - m}{\pi} \right)^2 (1 - \sin(M_2)) \right]$ $\delta = M_1 \left[m + \frac{1 - 2 \cdot m}{\pi} \cos(M_2) \right]$ $\xi = M_1 \cdot \sin(M_2), \text{ где } k = \frac{\varphi_i}{\varphi_{\text{фаз}}}$ $M_1 = \frac{1}{0,184(m - m^2) + 0,204}$ $M_2 = \frac{0,5 \cdot \pi(0,5 - k)}{0,5 - m}$

Для расчета коэффициентов пути, скорости и ускорения толкателя с учетом тех или иных законов движения толкателя предназначены следующие подпрограммы: 1) по закону Шуна – подпрограмма *LOWSUNA*; 2) закону синусоидального ускорения – *LOWSIN*; 3) закону косинусоидального ускорения – *LOWCOS*; 4) степенному закону 7-й степени (4-5-6-7) – *LOWSEVEN*; 5) степенному закону 9-й степени (5-6-7-8-9) – *LOWEIGHTEEN*; 6) закону Стодarta – *LOWSTODART*; 7) двойному гармоническому закону – *LOWDOUBLEGARM*; 8) степенному закону 18-й степени (2-6-10-14-18) – *LOWEIGHTEEN*; 9) закону Неклютина с выстоями – *LOWNECLWITH*; 10) закону Неклютина без выстоев – *LOWNECL*.

Выходные параметры подпрограмм: $pp() = \varsigma$; $pp1() = \delta$; $pp2() = \xi$.

KULAK.BAS	LOWSUNA
Sub LOWSUNA(pp() As Double, pp1() As Double, pp2() As Double) 620 'ЗАКОН ШУНА For i = 1 To N1 $f = (i - 1) * F0$: $k = f / F1$: $P(i) = 10 * k^3 - 15 * k^4 + 6 * k^5$ $P1(i) = 30 * (k^2 - 2 * k^3 + k^4)$: $P2(i) = 60 * (k - 3 * k^2 + 2 * k^3)$ Next i End Sub	

KULAK.BAS	LOWSIN
Sub LOWSIN(pp() As Double, pp1() As Double, pp2() As Double) REM ЗАКОН СИНУСА For i = 1 To N1 $f = (i - 1) * F0$: $k = f / F1$: $p(i) = k - \sin(2 * \pi * k) / (2 * \pi)$ $p1(i) = 1 - \cos(2 * \pi * k)$: $p2(i) = 2 * \pi * \sin(2 * \pi * k)$ Next i End Sub	

KULAK.BAS	LOWCOS
Sub LOWCOS(pp() As Double, pp1() As Double, pp2() As Double) REM ЗАКОН КОСИНУСА For i = 1 To N1 $f = (i - 1) * F0$: $k = f / F1$: $P(i) = (1 - \cos(\pi * k)) * 0.5$ $P1(i) = \pi / 2 * \sin(\pi * k)$: $P2(i) = \pi^2 / 2 * \cos(\pi * k)$ Next i End Sub	

KULAK.BAS	LOWSEVEN
Sub LOWSEVEN(pp() As Double, pp1() As Double, pp2() As Double) 'ЗАКОН 7-Й СТЕПЕНИ For i = 1 To N1 f = (i - 1) * F0: k = f / F1 $P(i) = 35 * k^4 - 84 * k^5 + 70 * k^6 - 20 * k^7$ $P1(i) = 140 * k^3 * (1 - 3 * k + 3 * k^2 - k^3)$ $P2(i) = 420 * k^2 * (1 - 4 * k + 5 * k^2 - 2 * k^3)$ Next i End Sub	

KULAK.BAS	LOWNINE
Sub LOWNINE(pp() As Double, pp1() As Double, pp2() As Double) For i = 1 To N1 'ЗАКОН 9-Й СТЕПЕНИ F = (i - 1) * F0: k = F / F1 $P(i) = 126 * k^5 - 420 * k^6 + 540 * k^7 - 315 * k^8 + 70 * k^6$ $P1(i) = 630 * k^4 * (1 - 4 * k + 6 * k^2 - 4 * k^3 + k^4)$ $P2(i) = 2520 * k^3 * (1 - 5 * k + 9 * k^2 - 7 * k^3 + 2 * k^4)$ Next i End Sub	

KULAK.BAS	LOWSTODART
Sub LOWSTODART(pp() As Double, pp1() As Double, pp2() As Double) 'ЗАКОН СТОДАРТА For i = 1 To N1 f = (i - 1) * F0: k = f / F1 $P(i) = 6.098 * k^3 - 20.78 * k^5 + 26.731 * k^6 - 13.61 * k^7 + 2.561 * k^8$ $P1(i) = 18.293 * k^2 - 103.902 * k^4 + 160.389 * k^5 - 95.268 * k^6 + 20.488 * k^7$ $P2(i) = 36.585 * k - 415.608 * k^3 + 801.947 * k^4 - 571.605 * k^5 + 143.413 * k^6$ Next i End Sub	

KULAK.BAS	LOWDOUBLEGARM
Sub LOWDOUBLEGARM(pp() As Double, pp1() As Double, pp2() As Double) 540 'ЗАКОН ДВОЙНОЙ ГАРМОНИЧЕСКИЙ For i = 1 To N1 F = (i - 1) * F0: k = F / F1 $P(i) = 0.5 * (1 - \cos(\pi * k)) - 0.125 * (1 - \cos(2 * \pi * k))$ $P1(i) = 0.5 * \pi * (\sin(\pi * k) - 0.5 * \sin(2 * \pi * k))$ $P2(i) = 0.5 * \pi^2 * (\cos(\pi * k) - \cos(2 * \pi * k))$ Next i End Sub	

KULAK.BAS	LOWEIGHTEEN
Sub LOWEIGHTEEN(pp() As Double, pp1() As Double, pp2() As Double) 1370 'ЗАКОН 18-Й СТЕПЕНИ For i = 1 To N1 F = (i - 1) * F0: k = F / F1 P(i) = 1 / 128 * (128 - 315 * (1 - k) ^ 2 + 420 * (1 - k) ^ 6 - 378 * (1 - k) ^ 10 + 180 * (1 - k) ^ 14 - 35 * (1 - k) ^ 18) P1(i) = 1 / 64 * (315 * (1 - k) - 1260 * (1 - k) ^ 5 + 1890 * (1 - k) ^ 9 - 1260 * (1 - k) ^ 13 + 315 * (1 - k) ^ 17) P2(i) = 5 / 64 * (-63 + 1260 * (1 - k) ^ 4 - 3402 * (1 - k) ^ 8 + 3276 * (1 - k) ^ 12 - 1071 * (1 - k) ^ 16) Next i End Sub	

KULAK.BAS	LOWNECL
Sub LOWNECL(pp() As Double, pp1() As Double, pp2() As Double) 'ЗАКОН НЕКЛЮТИНА БЕЗ ВЫСТОЯ i = 1: j = N1 720 f = (i - 1) * F0: k = f / F1 If k > N Then GoTo 780 P2(i) = 1 / (0.184 * (N - N ^ 2) + 0.204) P1(i) = P2(i) * k: P(i) = P2(i) * (k ^ 2) / 2 GoTo 820 780 M1 = 1 / (0.184 * (N - N ^ 2) + 0.204) M2 = 1 - Sin((0.5 * PI) * (0.5 - k) / (0.5 - N)) P(i) = M1 * (N * k - (0.5 * N ^ 2) + 4 * ((0.5 - N) / PI) ^ 2 * M2) P1(i) = M1 * (N + ((1 - 2 * N) / PI) * Cos((0.5 * PI) * (0.5 - k) / (0.5 - N))) P2(i) = M1 * Sin((PI / 2) * (0.5 - k) / (0.5 - N)) 820 P(j) = 1 - P(i) P1(j) = P1(i) P2(j) = -P2(i) i = i + 1: j = j - 1 If i <= n4 Then GoTo 720 End Sub	

KULAK.BAS	LOWNECLWITH
Sub LOWNECLWITH(pp() As Double, pp1() As Double, pp2() As Double) 880 'ЗАКОН НЕКЛЮТИНА С ВЫСТОЕМ i = 1: j = N1 900 f = (i - 1) * F0 k = f / F1 M1 = 4 * PI / (PI - 4 * N * PI + 8 * N) If k > N Then GoTo 970 P(i) = M1 * (2 * N / PI) * (k - (2 * N / PI) * Sin(k * PI / (2 * N))) P1(i) = M1 * (2 * N / PI) * (1 - Cos(k * PI / (2 * N)))	

```

P2(i) = M1 * Sin(k * PI / (2 * N)):GoTo 1060
970 If k > (0.5 - N) Then GoTo 1020
P(i) = M1 * (((PI ^ 2 - 8) * N ^ 2) / (2 * PI ^ 2) + ((2 - PI) / PI) * (N * k) +_
(k ^ 2) * 0.5)
P1(i) = M1 * (((2 - PI) / PI) * N + k)
P2(i) = M1
GoTo 1060
1020 M2 = ((0.5 - k) * PI / (2 * N))
P(i) = M1 * (-0.125 + 0.5 * N + (0.5 - (2 * N / PI) * (PI - 1)) * k -_
((2 * N / PI) ^ 2) * Sin(M2))
P1(i) = M1 * (0.5 - (2 * N * (PI - 1) / PI) + ((2 * N) / PI) * Cos(M2))
P2(i) = M1 * Sin((0.5 - k) / (2 * N) * PI)
1060 P(j) = 1 - P(i)
P1(j) = P1(i):P2(j) = -P2(i)
i = i + 1:j = j - 1
If i <= n4 Then GoTo 900
Next i
End Sub

```

2.5. Блок-схема программы расчета геометрических, кинематических и динамических параметров кулачка

На рис. 2.5 представлена укрупненная блок-схема полного расчета параметров кулачковых приводов. Блок-схема включает в себя расчет перемещения, аналогов скорости и ускорения толкателя с помощью подпрограмм законов движения; расчет координат теоретического профиля кулачка, а также радиуса кривизны профиля кулачка, координат центра кривизны и углов давления кулачка с помощью подпрограмм, выбираемых в зависимости от типа кулачка. Для расчета контркулачка применяется аналогичный алгоритм, за исключением кинематики толкателя. В зависимости от поставленной задачи некоторые блоки расчета могут быть исключены.

Программа расчета составляется так, чтобы проводить анализ механизма при изменении угла поворота кулачка за один цикл работы.

Алгоритм применяется дважды: для фаз удаления и сближения, поскольку законы перемещения, скорости и ускорения толкателя на этих фазах могут отличаться.

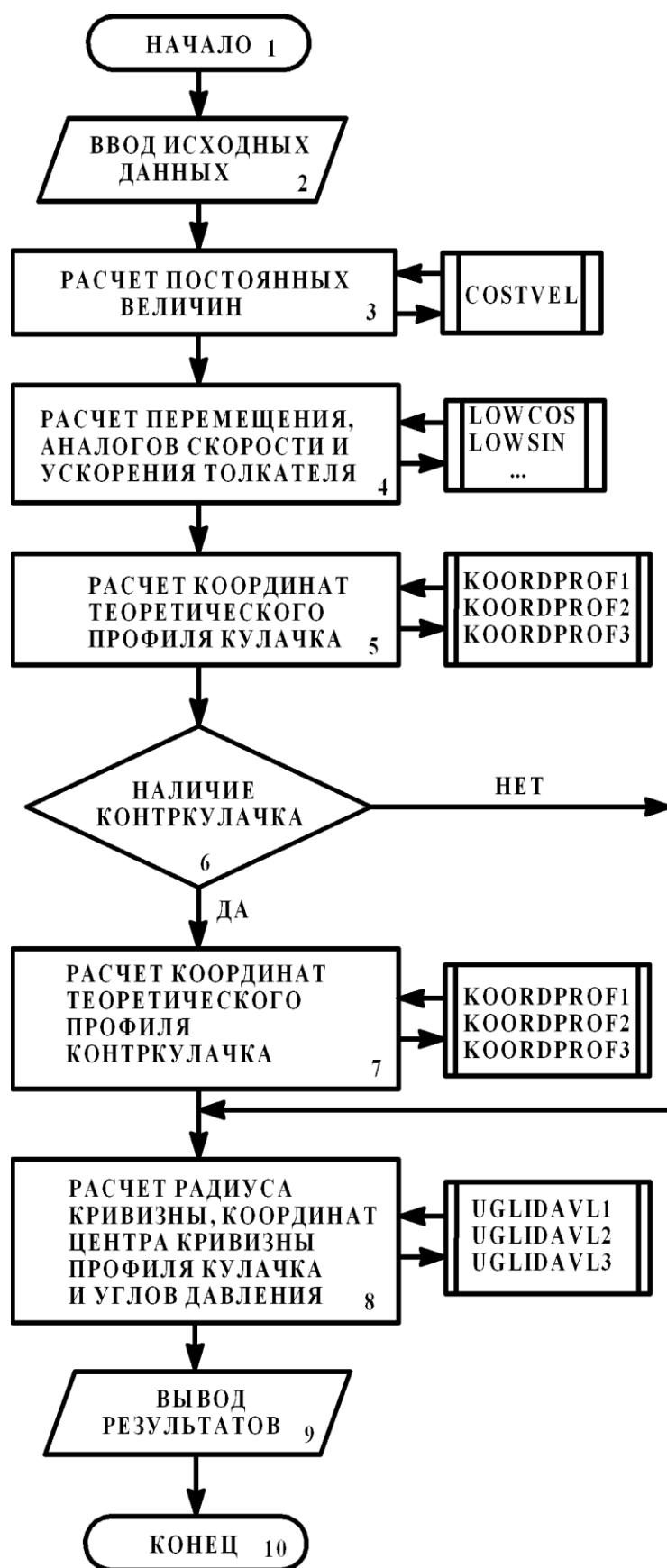


Рис. 2.5. Блок-схема полного расчета параметров кулачковых приводов

Подпрограммы расчета необходимых параметров выбираются в зависимости от типа кулачкового привода.

Это же относится к расчету постоянных величин. При этом перемещение, аналогии скорости и ускорения толкателя рассчитываются в зависимости от выбранного закона.

При расчете контркулачка применяются те же подпрограммы.

Если поставлена задача кинематического расчета всего механизма с кулачковым приводом, то кинематические параметры входного звена (толкателя) будут известны для любого применяемого закона движения.

В этой связи потребуется использование подпрограмм модуля *KULAK.BAS* совместно с подпрограммами модуля *Modul1 (OAP.BAS)*.

Примеры работы с подпрограммами приводятся в разд. 4.

РАЗДЕЛ 3

ОСНОВЫ ПРОГРАММИРОВАНИЯ В VISUAL BASIC

3.1. Общие сведения о языке программирования

Visual Basic (VB) – это система программирования, предназначенная для написания программ, работающих под управлением операционной системы *Windows*. В *Visual Basic* можно разрабатывать очень сложные приложения практически для любой области современных компьютерных технологий: бизнес-приложения, игры, мультимедиа, базы данных [8].

Причины широкой популярности *Visual Basic* в том, что написание компьютерных программ доступно широкому кругу пользователей благодаря применению графического интерфейса. В визуальной среде программирования можно быстро создать дизайн диалоговых окон, с которыми будет работать пользователь, разместив и упорядочив их элементы.

При написании программ традиционным способом (без использования графического интерфейса) система взаимодействия с пользователем создается на основе команд языка программирования. Очевидно, что начинающим гораздо проще освоить метод визуального программирования.

Одним из основных преимуществ языка *Visual Basic* является возможность очень быстрого создания работоспособных приложений.

В *Visual Basic* реализована модель объектно-ориентированного событийно управляемого программирования.

Основным понятием в объектно-ориентированном программировании является «объект». *Объект* – программная конструкция, объединяющая наборы данных, программы для их обработки и интерфейс для взаимодействия с внешней средой.

Видимые, или визуальные, объекты *Visual Basic* используются для создания средства общения пользователя и приложения, которое принято называть *пользовательским интерфейсом*.

Большинство объектов служат для организации ввода данных в программу, вывода данных на печатающие устройства или экран и для изменения хода работы программы. Каждый из объектов передает в приложение сообщение о произошедшем событии: нажатии клавиши, щелчке мыши, т. е. о действии оператора. В связи с этим их еще называют *элементами управления*. Это *командные кнопки, клавиши, текстовые поля, рамки и окна* для рисунков и проч. Объекты обладают *свойствами*, могут использовать *методы* и реагируют на *события*.

При многообразии средств разработки интерфейса необходимо соблюдать основные принципы его создания.

1. Естественность (интуитивность). Работа с программой не должна вызывать у пользователя сложностей в поиске необходимых директив (элементов интерфейса) для управления процессом выполнения поставленной задачи.

2. Непротиворечивость. Если в процессе работы с программой пользователь использовал некоторые приемы работы с некоторой частью программы, то и в другой части программы приемы работы должны быть идентичны.

3. Неизбыточность. Пользователь должен вводить только минимальную информацию, необходимую для работы.

4. Непосредственный доступ к системе помощи. В процессе работы требуется, чтобы система обеспечивала пользователя необходимыми инструкциями. Система помощи отвечает ряду основных аспектов: качеству и количеству обеспечиваемых команд; характеру сообщений об ошибках и подтверждения того, что система делает. Сообщения об ошибках должны быть полезны и понятны пользователю.

5. Гибкость. Для неопытных пользователей интерфейс может быть организован как иерархическая структура меню, а для опытных пользователей – как команды, комбинации нажатий клавиш и параметры.

Приложение, создаваемое в среде *Visual Basic*, называется *проектом*.

Программный проект – это совокупность частей, составляющих будущее *WINDOWS*-приложение. Любой программный проект должен обязательно со-

стоять из *экранных форм* (хотя бы одной) и *программных модулей* (хотя бы одного). Каждый проект хранится в отдельном файле с расширением *vbp*.

Экранная форма – это графическое представление *WINDOWS*-приложения вместе с содержанием этого окна.

Содержание окна включает в себя совокупность:

- а) свойств окна с их значениями;
- б) объектов, находящихся в окне;
- в) свойств этих объектов с их значениями.

В *Visual Basic* экранная форма хранится в отдельном файле с расширением *frm*.

Программный модуль – это хранящийся в отдельном файле программный код (текст некоторой программы). Он может использоваться при решении чаще всего одной, а иногда и нескольких задач. Имя этого файла имеет расширение *bas*.

Программный код проекта существует не сам по себе, он привязан к отдельным объектам экранной формы. Часть кода, которая относится только к одному объекту, в свою очередь может состоять из нескольких фрагментов – *процедур*.

В *Visual Basic* программный код почти всегда привязывается к какому-либо *событию* (например, щелчок мыши по какому-либо объекту экранной формы, загрузка новой экранной формы, перемещение указателя мыши вдоль полосы прокрутки, нажатие какой-либо клавиши), которое является сигналом к началу работы программы.

Сначала проектируется экранная форма, затем устанавливаются события, которые будут происходить в работающем приложении, и только потом программируются действия, связанные с этими событиями. Большинство процедур, из которых состоит программный код *Visual Basic*, привязано к событиям и называется *процедурами-событиями*.

Особым видом объектов являются формы и элементы управления. *Элементы управления* (ЭУ) позволяют инициировать определенные

события и тем самым обеспечивают управление программой. *Формы* являются окнами, или контейнерами, содержащими другие ЭУ и обеспечивающими обмен информацией между компонентами проекта, между проектом и другими приложениями, между проектом и пользователем.

Каждый объект характеризуется параметрами, которые можно разделить на три категории: *события, методы и свойства*.

События связаны с определенными действиями пользователя и могут вызывать код *Visual Basic*.

Методы – это рабочие операторы объекта, программные процедуры.

Свойства отвечают за внешний вид и поведение объекта.

Границы между свойствами и методами в определенной степени являются размытыми. Основное различие между ними заключается в том, что со свойствами можно работать как во время разработки проекта, так и во время его выполнения, а методы доступны только при выполнении проекта.

Свойства и методы называются *интерфейсом* объекта.

Объекты объединяются в *классы*. К одному классу принадлежат объекты с одинаковым набором свойств, методов и событий.

Манипулировать объектами можно:

- 1) изменяя свойства объекта;
- 2) заставляя объект выполнять специфические задания с помощью активизации методов, ассоциированных с этим объектом.

Оба способа манипуляции объектами часто ассоциируются с *наступлением* некоторого пользовательского (программного) или системного *события*, т. е. действия или ситуации, связанной с объектом.

Таким образом:

- *программные коды* содержатся в процедурах и функциях, объединяемых в *модули*;
- *функции и процедуры* соответствуют *методам и свойствам* объекта;
- *формы, классы и модули* являются контейнерами для других элементов управления и объектов.

3.2. Настройка приложения

После запуска *Visual Basic* на экране появляется диалоговое окно (рис. 3.1), в котором предлагается выбор типа создаваемого приложения. Из этого же окна можно загрузить уже существующий проект. Перед работой с *VB* его необходимо настроить. Для этого следует запустить *VB*, зайти в меню *Tools>Options* (рис. 3.2) и поставить галочку *Require Variable Declaration* (для устранения ошибок при автоматическом определении переменных). Далее на вкладке *Editor Format* (рис. 3.3) в списке *Font* нужно указать *Courier New Cyr* (для отображения кириллицы).

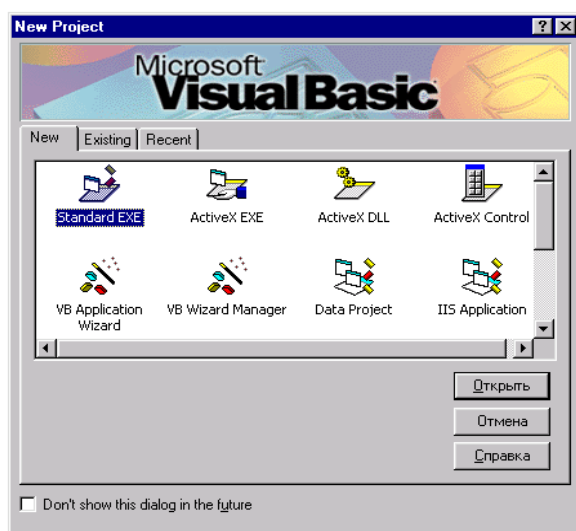


Рис. 3.1. Выбор приложения

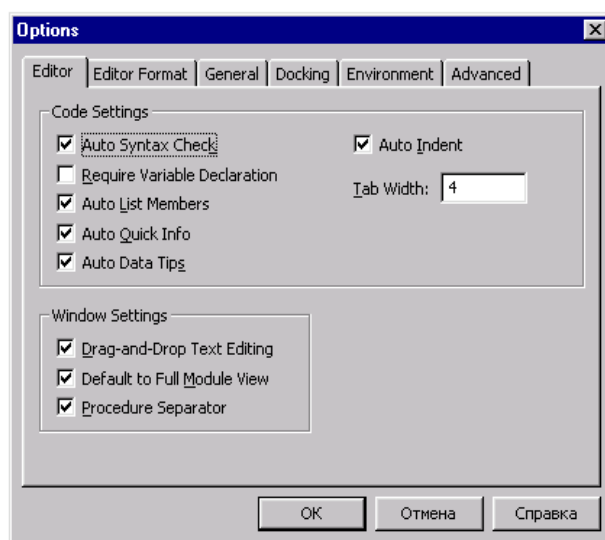


Рис. 3.2. Установка настроек

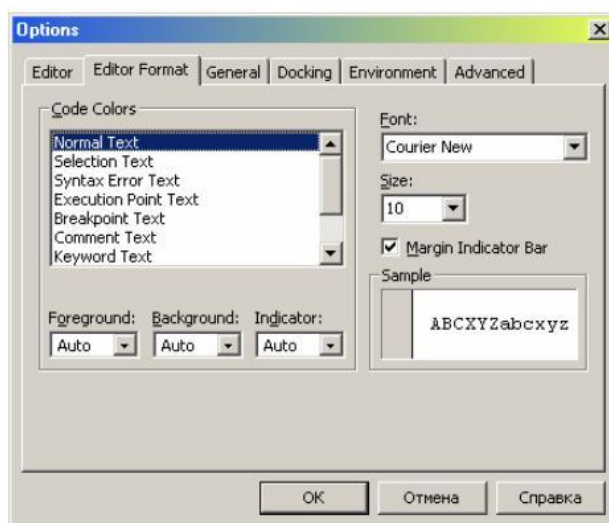


Рис. 3.3. Настройка приложения

Рекомендуется установить ярко-синий цвет зарезервированных слов: в списке *Code Colors Keyword Text* (поле *Foreground*) указать ярко-синий цвет (седьмой снизу).

3.3. Интерфейс приложения

Графический интерфейс среды разработки приложения показан на рис. 3.4.

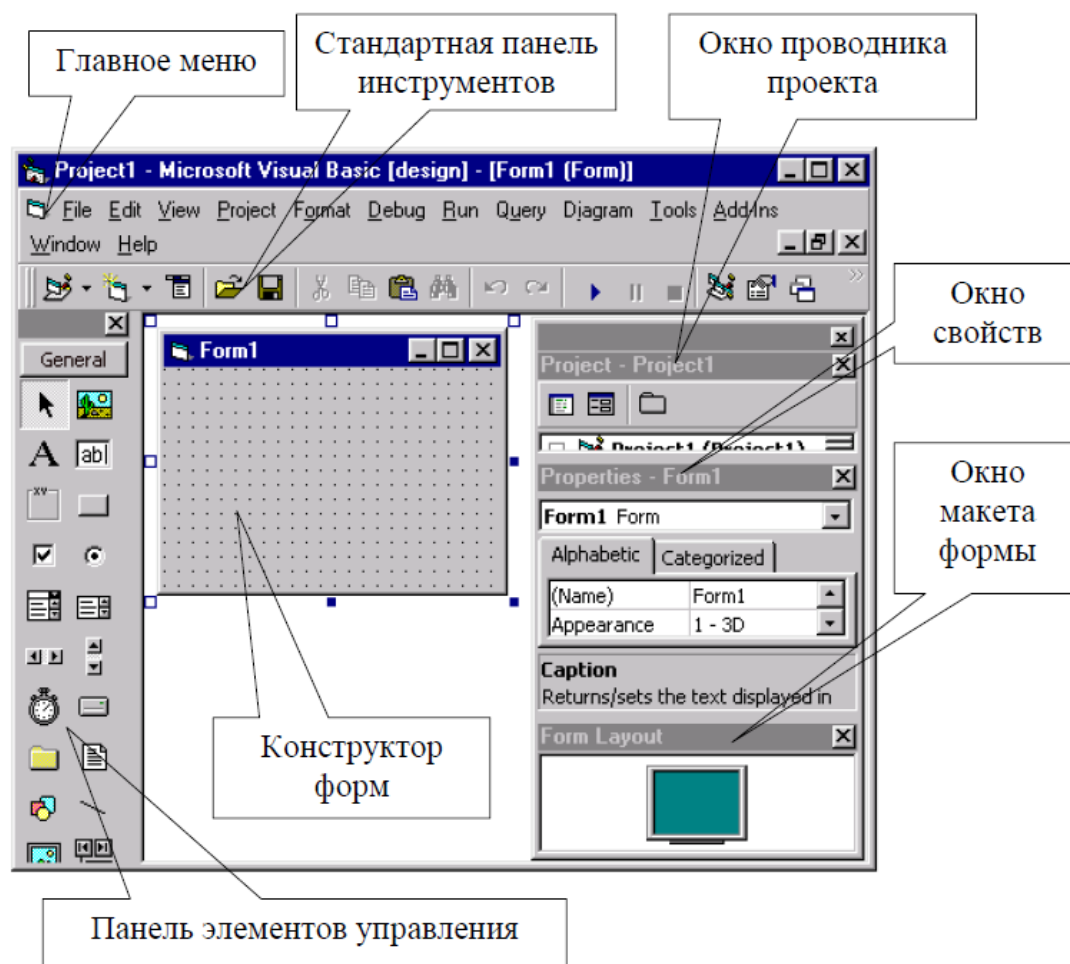


Рис. 3.4. Графический интерфейс среды разработки приложения

Строка меню (рис. 3.5) состоит из заголовков меню, которые содержат все команды, необходимые при работе с *Visual Basic*.



Рис. 3.5. Меню приложения

Меню *File*, *View*, *Edit*, *Window* и *Help* являются характерными для *Windows* и приложений *Windows*, но имеют и ряд особенностей.

Меню *File* (Файл) содержит команды управления файлами проекта.

Меню *Edit* (Правка) применяется при редактировании текста программ.

Меню *View* (Вид) позволяет изменять конфигурацию интегрированной среды. В меню *View* имеется команда *Toolbars* (Панели инструментов). Эта команда вызывает диалоговое окно со списком дополнительных панелей инструментов, каждая из которых может быть помещена в интегрированную среду.

Меню *Project* (Проект) содержит команды управления компонентами проекта и используется на этапе конструирования.

Меню *Format* (Формат) позволяет изменять размеры и взаимное расположение объектов, размещенных в окне форм.

Меню *Debug* (Отладка) применяется при внесении исправлений в код программы и при контроле над ходом выполнения процедур.

Меню *Tools* (Инструменты) содержит инструменты для автоматизации написания текстов программ.

Меню *Add-Ins* (Подключить) позволяет использовать дополнительные ресурсы.

Меню *Help* (Помощь) содержит файл справки.

Панель инструментов *Toolbar* значительно ускоряет и упрощает доступ к командам меню *Visual Basic*.

Стандартная панель инструментов показана на рис. 3.6.



Рис. 3.6. Стандартная панель инструментов

Назначение элементов стандартной панели инструментов согласно нумерации на рис. 3.6 указано в табл. 3.1.

Таблица 3.1

Элементы стандартной панели инструментов

№	Название	Назначение
1	<i>Add Standard EXE Project</i>	Добавляет стандартный <i>exe</i> -проект
2	<i>Add Form</i>	Добавляет форму в проект
3	<i>Menu Editor</i>	Вызывает редактор меню
4	<i>Open Project</i>	Открывает проект
5	<i>Save Project</i>	Сохраняет проект
6	<i>Cut</i>	Вырезает информацию в буфер обмена
7	<i>Copy</i>	Копирует информацию в буфер обмена
8	<i>Paste</i>	Вставляет информацию из буфера обмена
9	<i>Find</i>	Осуществляет поиск информации по контексту
10	<i>Can't Undo</i>	Отменяет предыдущее действие
11	<i>Can't Redo</i>	Восстанавливает отмененное действие
12	<i>Start</i>	Запускает программу на выполнение
13	<i>End</i>	Прекращает выполнение программы
14	<i>Break</i>	Прерывает выполнение программы
15	<i>Project Explorer</i>	Открывает окно проводника проектов
16	<i>Properties Window</i>	Открывает окно свойств
17	<i>Form Layout Window</i>	Открывает окно макета формы
18	<i>Object Browser</i>	Открывает окно браузера объектов
19	<i>Toolbox</i>	Открывает панель элементов управления
20	<i>Data View Window</i>	Открывает окно просмотра данных
21	<i>Visual Component Manager</i>	Открывает окно управления визуальными компонентами

Панель элементов управления (рис. 3.7) содержит основные элементы управления проектом, назначение которых согласно их нумерации поясняется в табл. 3.2.

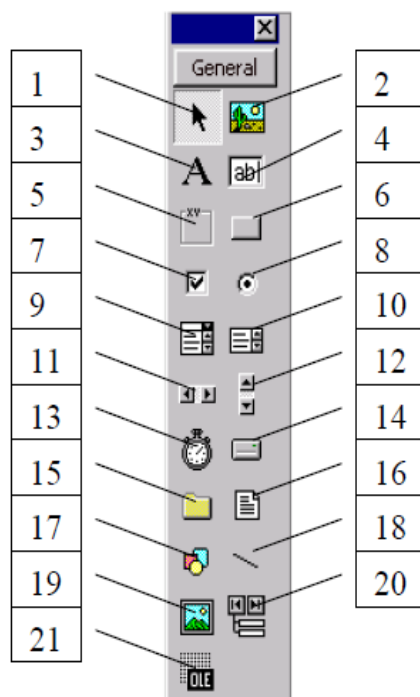


Рис. 3.7. Панель элементов управления

Таблица 3.2

Элементы управления

№	Название	Назначение
1	2	3
1	<i>Pointer</i> (Указатель)	Используется для позиционирования маркера (указателя) мыши
2	<i>PictureBox</i> (Графическое окно)	Размещает в форме графическое окно, предназначенное для объединения элементов группы, вывода в него графических изображений, а также текста графических элементов и анимации
3	<i>Label</i> (Метка)	Размещает в форме объекты, предназначенные для создания текстовой информации, надписей и примечаний

1	2	3
4	<i>TextBox</i> (Текстовое поле)	Размещает в форме текстовое поле, предназначенное для ввода текстовой информации, чисел и дат
5	<i>Frame</i> (Рамка)	Создает в форме рамку с заголовком для группировки объектов в логическую группу
6	<i>CommandButton</i> (Кнопка управления)	Размещает в форме кнопку управления для инициации действий, выполнения команд, запуска программ
7	<i>CheckBox</i> (Флажок)	Размещает в форме флажок, предназначенный для формирования условий выполнения программ или каких-либо настроек, работающих по принципу «да – нет»
8	<i>OptionButton</i> (Переключатель)	Создает в форме переключатели для выбора режима работы или настроек выполнения программы
9	<i>ComboBox</i> (Поле со списком)	Создает в форме объект, содержащий одновременно поле ввода и раскрывающийся список
10	<i>ListBox</i> (Список)	Создает в форме список для выбора одного или нескольких значений из предлагаемого списка
11	<i>HScrollBar</i> (Горизонтальная полоса прокрутки)	Размещает в форме горизонтальную полосу прокрутки, используемую в качестве ползунка для выбора значений и заданного диапазона

1	2	3
12	<i>VScrollBar</i> (Вертикальная полоса прокрутки)	Размещает в форме вертикальную полосу прокрутки
13	<i>Timer</i> (Таймер)	Размещает в форме таймер
14	<i>DriveListBox</i> (Список устройств)	Создает в форме список устройств
15	<i>DirListBox</i> (Список папок)	Создает в форме древовидный список папок
16	<i>FileListBox</i> (Список файлов)	Создает в форме список файлов
17	<i>Shape</i> (Очертание)	Создает в форме геометрические фигуры: прямоугольник, квадрат, круг, эллипс, прямоугольник и квадрат со скругленными углами
18	<i>Line</i> (Линия)	Создает линии
19	<i>Image</i> (Изображение)	Создает в форме поля, предназначенные для отображения графических изображений
20	<i>Data</i> (Данные)	Создает элемент управления данными в базе данных для перемещения по записям и отображения результата навигации
21	<i>Pointer</i> (Указатель)	Используется для позиционирования маркера (указателя) мыши

Окно *Properties* (Свойства) (рис. 3.8) предназначено для отображения и настройки свойств формы, а также размещенных в ней объектов (размеры, расположение, цвет и т. д.).

Окно *Project group* (проводника проекта) (рис. 3.9) позволяет просматривать состав и свойства выбранного проекта, перемещаться между проектами,

если их открыто сразу несколько, копировать необходимые объекты из окна одного проекта в другое, как это осуществляется в проводнике системы *Windows*.

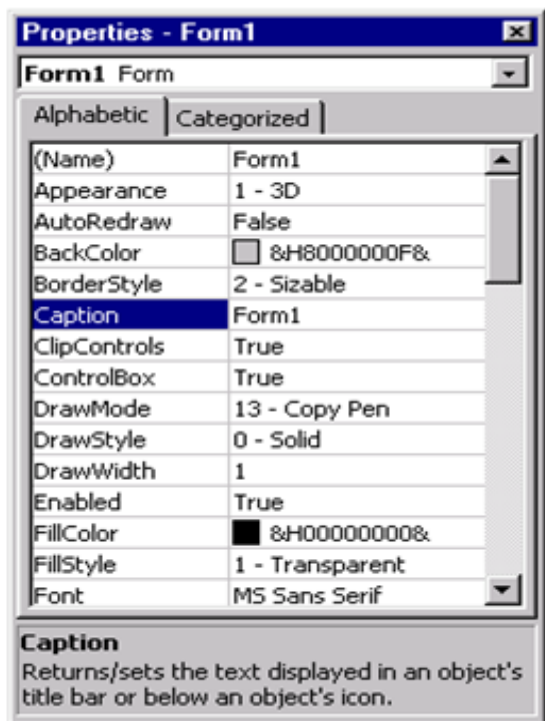


Рис. 3.8. Окно свойств
активного элемента

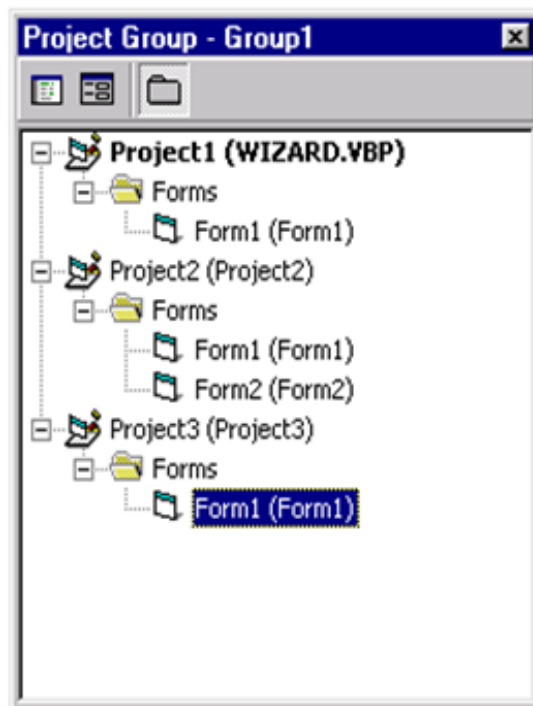


Рис. 3.9. Окно проводника проекта

3.4. Форма и объекты

Для создания проекта требуемой формы используется окно *конструктора форм* проекта (рис. 3.10).

Поскольку создаваемый проект должен состоять по меньшей мере из одной формы, окно формы появляется автоматически при открытии оболочки, а последующие формы при их необходимости добавляются в проект с помощью команды из меню *FILE* → *ADD FILE*.

С помощью мыши компьютера можно поместить на форму объекты управления (выбрав их на панели элементов), менять размеры формы и помещенных на нее объектов (или программным способом в коде процедуры).

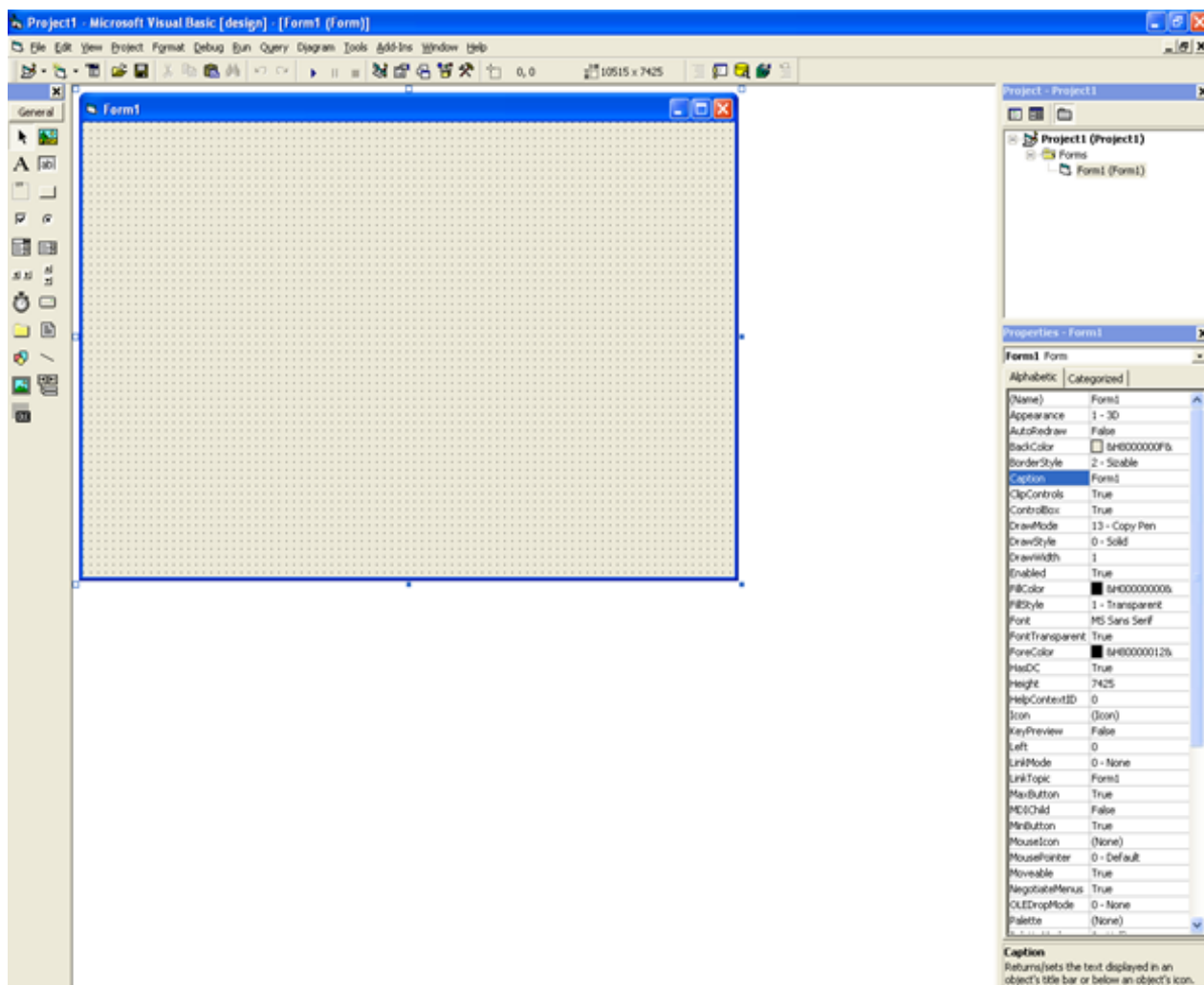


Рис. 3.10. Общий вид окна конструктора форм проекта

Разместим на форме проекта несколько объектов. Для этого с помощью мыши выберем нужный объект на панели элементов управления (рис. 3.11) и поместим его на форме, задав положение и размеры. Все свойства объекта отображаются на панели свойств (рис. 3.12).

Выделяем объект *Text1* и назначаем его основные свойства (рис. 3.12, а):

- *height* – высота;
- *left* – расстояние до левого края формы;
- *top* – расстояние до верхнего края формы;
- *width* – ширина (все величины в пикселях);
- *name* – имя объекта;
- *font* – шрифт текста на объекте (выбирается из всплывающего меню после нажатия данной строчки);

- *text* – текст, отображаемый на элементе (дальше пустым для записи данных при выполнении программы).

То же повторим для следующих объектов:

- *Label1* (добавляя свойство *Caption* (заголовок – текст, отображаемый на объекте), знак «+» и *Alignment* (расположение текста) посередине (рис. 3.12, б));
- *Text2* (рис. 3.12, в);
- *CommandButton1* (добавляя свойство *Caption*, знак «=» (рис. 3.12, г));
- *List1* (рис. 3.12, д).

При необходимости имеется возможность задать дополнительные свойства объектов: цвет фона, видимые или невидимые границы и т. п.

Все указанные действия можно заменить выполнением программного кода. Редактирование кода происходит в специальной области. Чтобы открыть эту область, необходимо выполнить действия *View* → *Code*.

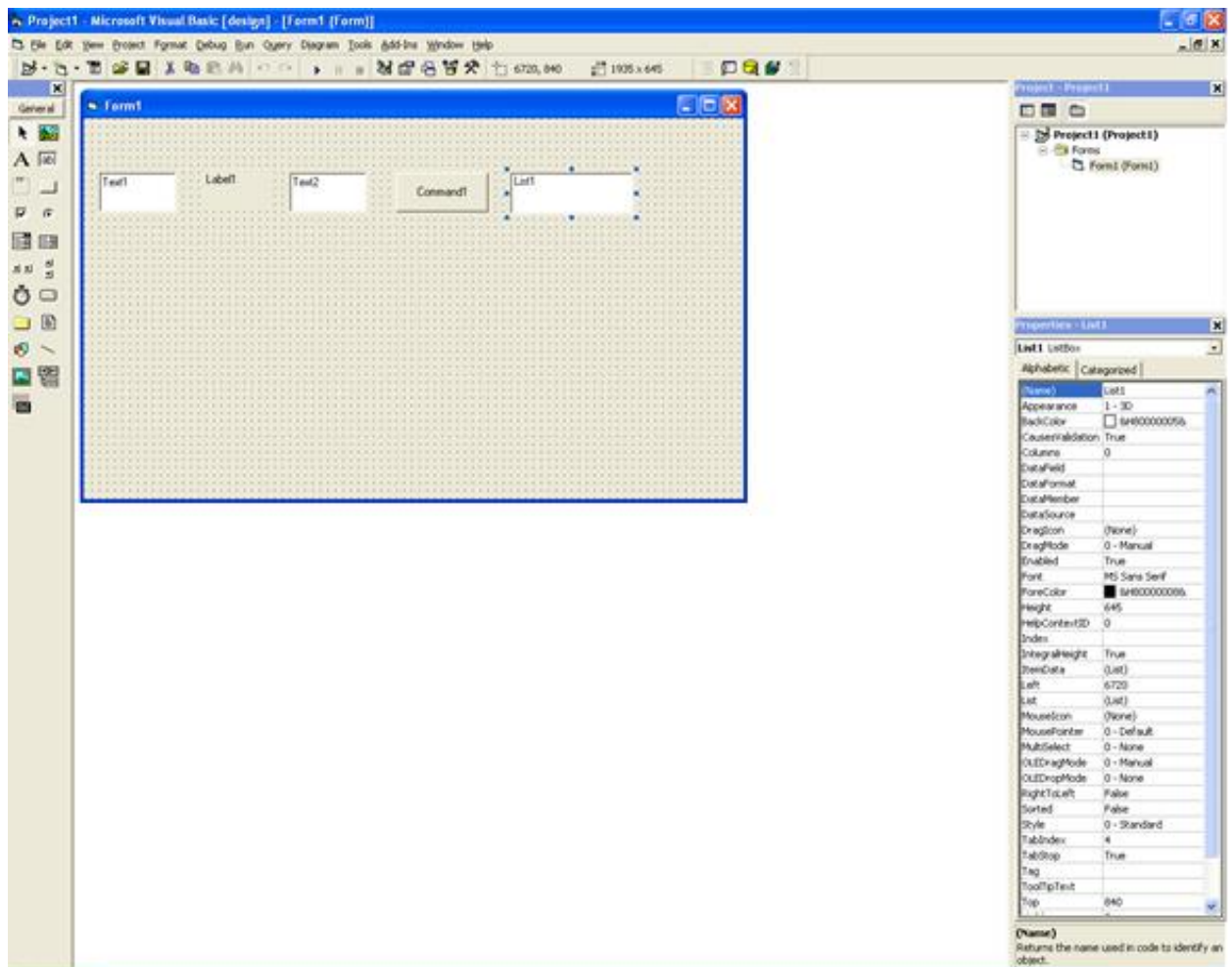
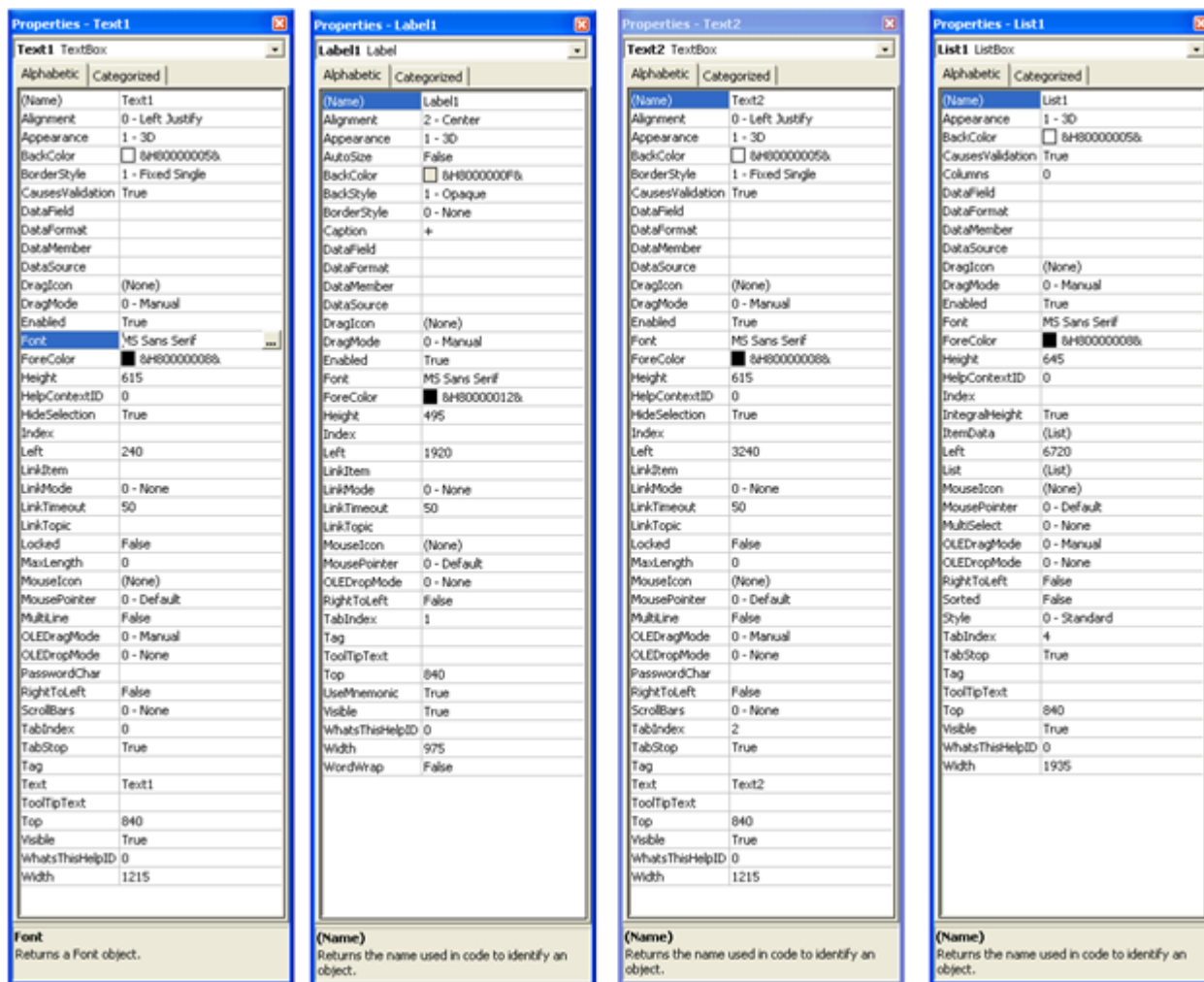


Рис. 3.11. Размещение элементов управления на форме проекта



а)

б)

в)

г)

Рис. 3.12. Свойства элементов управления

3.5. Подпрограмма командного кода

Кроме размещения визуальных объектов на форме проекта, нужно написать *код программы*, который будет манипулировать элементами управления и производить вычисления.

Выполнение проекта происходит по следующему принципу.

После запуска появляется стартовая форма, которую нужно объявить заранее (по умолчанию *Form1*).

Пользователь производит действия, предполагаемые данной формой (ввод данных, выбор, подтверждение), путем воздействия на объекты формы, в результате чего происходит реакция на них путем выполнения *программ-*

ного кода, хранящегося в подпрограмме этого события. Код подпрограммы (процедуры) размещается в модуле (самостоятельно задает ее название) или в имеющихся для этого вкладках различных событий, соответствующих объектам, помещенных на форме (рис. 3.13).

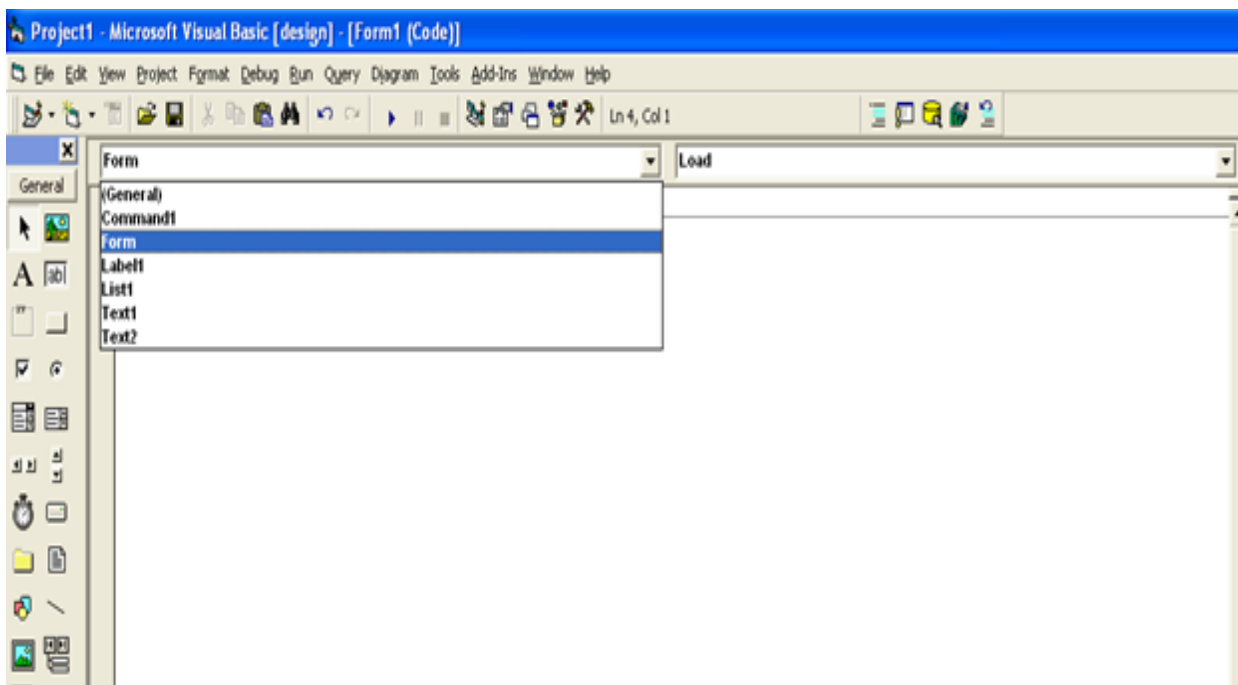


Рис. 3.13. Возможности размещения программного кода

Здесь во всплывающем слева меню представлены объекты, размещенные на форме, показанной на рис. 3.11. В правом меню – события, которые могут произойти с каждым из этих элементов (например, *Load* (загрузка) элемента *Form1*).

Код начинается после названия подпрограммы (*Sub ИМЯ*) и заканчивается обозначением ее завершения (*End Sub*) (см. рис. 1.3).

Модуль любого типа может содержать не более 65534 строк кода. Строка кода может содержать до 1023 символов.

Символ продолжения строки (перенос строки) содержит символ «пробел», сопровождаемый символом подчеркивания _ (одна логическая строка кода не может содержать более 25 символов переноса).

Комментарий имеет две формы записи: или самостоятельный (не выполняемый) оператор, начинающийся словом *Rem*, за которым может следо-

вать произвольный текст, или знак апострофа ' , который может стоять как в начале строки – перед находящимся текстом, так и после выполняемого оператора или объявления.

В *Visual Basic* каждый оператор записывается с новой строки, однако можно в одну строку поместить два оператора и более, если разделить их двоеточиями.

При описании синтаксиса используются следующие соглашения:

[A1] – необязательный параметр;

A1 | A2 – выбор одного из альтернативных значений;

... – три точки, следующие одна за другой, обозначают, что предшествующий им элемент строки может повторяться.

Каждый элемент данных или объект должен иметь *уникальное имя*:

- имена должны начинаться с литеры;
- не могут содержать внедренные точки или символы объявления типа данных;
- могут содержать не более 255 символов;
- имена элементов управления, форм, классов и модулей не должны превышать 40 символов.

В качестве имен нельзя использовать ключевые слова *Visual Basic*. Однако формы и элементы управления могут быть поименованы ключевыми словами, но обратиться к ним обычным способом не удастся: при обращении такое имя надо заключить в квадратные скобки.

Среди *процедур* различают процедуры *обработки событий* и процедуры *общего назначения*. Имя процедуры общего назначения составляется произвольно (с учетом общих правил построения имени).

Имя процедуры обработки события должно состоять из имени объекта и имени события, разделенных символом подчеркивания `_`. Например, для щелчка Click на командной кнопке *Command1* имя процедуры *Command1_Click*.

Для доступа к свойствам объекта указываются имя объекта и имя свойства, разделенные точкой: *Command1.Name*.

Аналогично для доступа к методам: *Command1.Move*.

При выполнении арифметических расчетов можно применять встроенные функции языка программирования *Visual Basic* [9] (табл. 3.3).

Таблица 3.3

Встроенные функции языка программирования Visual Basic

Функция	Возвращает/действие
<i>Abs(X)</i>	Возвращает абсолютное значение X
<i>Atn(X)</i>	Возвращает арктангенс X как угол, в радианах
<i>Cos(X)</i>	Косинус угла X
<i>Exp(N)</i>	Возвращает константу e , возведенную в степень N : $e \approx 2,718282$
<i>Fix(X)</i>	Возвращает целую часть X Не округляет число, а отбрасывает любую дробную часть
<i>Int(X)</i>	Возвращает целую часть X Не округляет число, а отбрасывает любую дробную часть
<i>Log(X)</i>	Возвращает натуральный логарифм X
<i>Rnd(N)</i>	Возвращает случайное число Используется <i>Rnd</i> совместно с <i>Randomize</i> (инициализация генератора случайных чисел); аргумент необязателен
<i>Sgn(X)</i>	Возвращает знак числа: – 1, если X отрицательно + 1, если X положительно 0, если $X = 0$
<i>Sin(X)</i>	Возвращает синус угла X
<i>Sqr(X)</i>	Возвращает корень квадратный из X
<i>Tan(X)</i>	Возвращает тангенс угла X

Рассмотрим принцип создания кода на примере вышеупомянутого проекта (см. рис. 3.11). Как сказано ранее, свойства элементов можно задать программным способом. Для этого используем событие *Form_Load* (загрузка формы), которое произойдет первым автоматически при запус-

ке программы. После двойного клика на форме открывается область редактирования кода нужного события *Form_Load* (рис. 3.14).

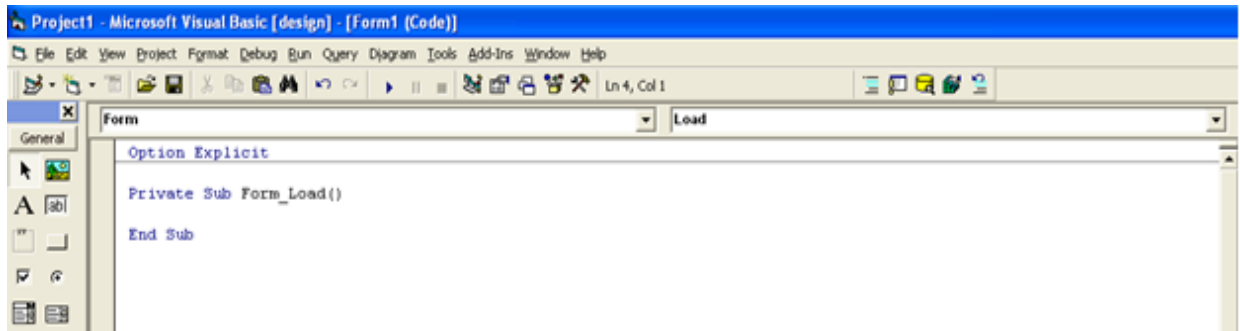


Рис. 3.14. Область редактирования кода

Запишем код, заключив его между строчками *Private Sub Form_Load()* и *End Sub*:

```
Private Sub Form_Load()
```

```
'Свойства элемента Форма1
```

```
Form1.Top = 2500 'Расстояние до верхней границы экрана
```

```
Form1.Left = 2500 'Расстояние до левой границы экрана
```

```
Form1.Height = 6405 'Высота объекта
```

```
Form1.Width = 10515 'Ширина объекта
```

```
'Свойства элемента Текст1
```

```
Text1.Top = 840 'Расстояние до верхней границы формы
```

```
Text1.Left = 240 'Расстояние до левой границы формы
```

```
Text1.Height = 615 'Высота объекта
```

```
Text1.Width = 1215 'Ширина объекта
```

```
Text1.FontSize = 24 'Кегль шрифта
```

```
'Свойства элемента Текст2
```

```
Text2.Top = 840 'Расстояние до верхней границы формы
```

```
Text2.Left = 3240 'Расстояние до левой границы формы
```

```
Text2.Height = 615 'Высота объекта
```

```
Text2.Width = 1215 'Ширина объекта
```

```
Text2.FontSize = 24 'Кегль шрифта
```

```
'Свойства элемента Метка1
```

```
Label1.Top = 840 'Расстояние до верхней границы формы
```

```
Label1.Left = 1920 'Расстояние до левой границы формы
```

```
Label1.Height = 495 'Высота объекта
```

```
Label1.Width = 975 'Ширина объекта
```

```
Label1.FontSize = 24 'Кегль шрифта
```

```
Label1.Caption = "+" 'Надпись на объекте
```

```
'Свойства элемента Командная Кнопка1
Command1.Top = 840 'Расстояние до верхней границы формы
Command1.Left = 4920 'Расстояние до левой границы формы
Command1.Height = 615 'Высота объекта
Command1.Width = 1455 'Ширина объекта
Command1.FontSize = 24 'Кегль шрифта
Command1.Caption = "=" 'Надпись на объекте
```

```
'Свойства элемента Список1
List1.Top = 840 'Расстояние до верхней границы формы
List1.Left = 6720 'Расстояние до левой границы формы
List1.Height = 645 'Высота объекта
List1.Width = 1215 'Ширина объекта
List1.FontSize = 24 'Кегль шрифта
End Sub
```

На этом разработка интерфейса приложения завершена.

Далее требуется составить код подпрограммы расчета, для которого предназначено приложение. Этот код будет активироваться после события нажатия командной кнопки («=»).

Чтобы открыть область редактирования кода, дважды нажмем командную кнопку (рис. 1.15). Код записывается между строками *Private Sub Command1_Click()* и *End Sub*.

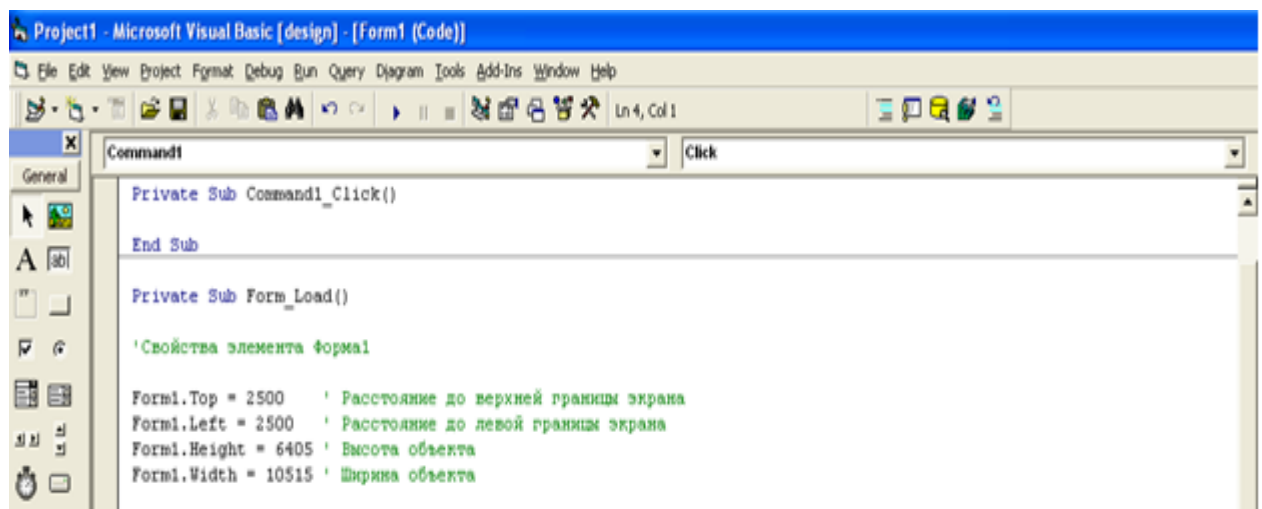


Рис. 3.15. Область редактирования программного кода командной кнопки

Этот код должен выполнять арифметическое действие между числами, занесенными в ячейки *Text1* и *Text2*.

Таким образом, в коде должно происходить присвоение значения первой переменной из ячейки *Text1*, а второй переменной – из ячейки *Text2*. Третья переменная должна быть равна сумме первых двух переменных. Результат должен отражаться в области *List1*. Код подпрограммы:

```
Private Sub Command1_Click()
A = Val(Text1.Text) 'Присвоение переменной A значения из ячейки Text1
B = Val(Text2.Text) 'Присвоение переменной B значения из ячейки Text2
C = A + B           'Расчет переменной C
List1.AddItem Format$(C, "##.##") 'Вывод значения переменной C в формате
                                'два знака перед запятой и два после
End Sub
```

3.6. Переменные и массивы

Во всех языках программирования высокого уровня программный код состоит из *переменных, выражений, операторов, управляющих структур, функций, классов и объектов*. Все данные проекта хранятся в области памяти, зарезервированной для соответствующей переменной (константы). В отличие от констант, содержание переменных может изменяться при выполнении программы. Значения переменных во время выполнения программы компьютер хранит в оперативной памяти: предварительно под каждую переменную отводит в памяти место, размер которого зависит от типа переменной. Типы переменных и констант языка программирования указаны в табл. 3.4.

Таблица 3.4

Типы переменных и констант языка программирования

Наименование	Тип	Длина	Описание
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
<i>Byte</i>	Байт	1 байт	Положительные целые числа в диапазоне 0...255
<i>Integer</i>	Целое	2 байта	Целочисленные значения в диапазоне – 32768...+ 32766

1	2	3	4
<i>Long</i>	Длинное целое	4 байта	Целочисленные значения в диапазоне – 2147483648...+ 2147483647
<i>Single</i>	С плавающей точкой одинарной точности	4 байта	Численные значения с плавающей точкой одинарной точности, т. е. с мантиссой, округленной до 10-значных цифр, в диапазоне – 1.401298E – 45...+ 3.402823E + 38
<i>Double</i>	С плавающей точкой двойной точности	8 байт	Численные значения с плавающей точкой двойной точности в диапазоне – 1.7E + 308...+ 1.7E+308
<i>Boolean</i>	Логический	2 байта	Только два значения: <i>True</i> (Истина) или <i>False</i> (Ложь). По умолчанию переменной присваивается значение <i>False</i> (Ложь)
<i>Variant</i>	Данные любого типа		Присваивается по умолчанию неопределенным переменным, за исключением строк фиксированной длины и пользовательских типов данных
<i>String</i>	Строка переменной длины	2*N	Строки, содержащие до 20000000000 алфавитно-цифровых символов
<i>String *N</i>	Строка постоянной длины N		

1	2	3	4
<i>Date</i>	Даты и время	8 байт	Значения даты и времени. Дата может быть в диапазоне от 1 января 100 года до 31 декабря 9999 года
<i>Object</i>			Особый тип данных, содержащий указатель (ссылку на адрес в памяти) на объект любого типа
<i>Currency</i>	Денежный	8 байт	Для хранения и выполнения операций с числами, имеющими 4 знака после десятичной точки. Обеспечивает правильное округление результатов денежной операции

В *Visual Basic* используется *явное* и *неявное* объявление переменных.

При неявном объявлении переменной она имеет тип *Variant*. Неявное объявление переменной удобно, но если в имени при последующем наборе окажется ошибка, то *Visual Basic* может ошибочно создать новую переменную или выдаст сообщение об ошибке исполнения программы.

Явное объявление переменной имеет следующие преимущества:

- ускорение выполнения кода;
- уменьшение количества ошибок в результате неправильного написания имени переменной;
- улучшенные читабельность и понятность кода. Доступный перечень объявлений всех переменных в начале модуля или процедуры позволяет легко определить, какие переменные используются в этом модуле или процедуре;
- возможность нормализовать выделение прописными буквами в имени переменной: за основу всегда берется объявленное имя;
- повышение эффективности кода, т. к. в этом случае типизированная переменная занимает только память, необходимую для этого типа;

- облегчение обнаружения некоторых ошибок программирования, в частности недопустимого смешивания типов данных. При этом визуализируются ошибки исполнения, которые не возникли бы, если бы переменные были нетипизированными.

Таким образом, предпочтительнее использовать явное объявление переменных, т. е. установить такой режим трансляции программы, при котором допускается только явное объявление переменных.

Синтаксис объявления *константы* следующий:

[Public|Private] Const ИМЯ_КОНСТАНТЫ [As ТИП] = ВЫРАЖЕНИЕ

Использование кириллицы в имени не допускается.

Синтаксис объявления *переменной* следующий:

Dim [Shared] |ReDim |Static|Private|Public ИМЯ_ПЕРЕМЕННОЙ As КЛАСС |ТИП (где *Shared* – указание на то, что переменную можно использовать во всех функциях и процедурах модуля).

ИМЯ_ПЕРЕМЕННОЙ имеет максимальную длину до 255 символов и образуется по тем же правилам, что и имя константы.

As ТИП присутствует в объявлении, если в имени переменной последний символ не является идентификатором типа (например, *Dim k As Integer*, но *Dim c%*).

КЛАСС применяется для объектных переменных.

ТИП используется для обычных переменных.

В одном операторе объявления может присутствовать несколько одно-типных переменных, имена которых разделяются запятыми:

Dim i As Integer, k As Integer.

В одном операторе объявления можно объединять несколько объявлений переменных разных типов:

Dim i As Integer, s As Double.

Объявлять переменные можно в разных местах:

- внутри процедуры (или функции): значение переменной будет доступно только в коде этой процедуры (или функции);

- в разделе *General Declarations* (Глобальных Объявлений) модуля и формы: переменные будут доступны в любом месте кода модуля и формы;

- в данном месте: переменные могут быть объявлены с помощью зарезервированных слов *Private (Dim)* и *Public (Private (Dim))* будут доступны только в коде модуля и формы, *Public* доступны везде).

Операторы *Dim*, *Private*, *Static*, *Public* определяют область действия переменной.

По области определения различают следующие виды переменных:

- а) *локальные* – объявляются внутри процедуры или функции и доступны только внутри такой подпрограммы;

- б) *контейнерные* – объявляются с ключевым словом *Dim* в секции (*General Declarations*) контейнера (форма, модуль или класс) и доступны только в соответствующем текущем контейнере;

- в) *глобальные* – объявляются с ключевым словом *Public* в секции контейнера и доступны во всем проекте.

Локальные переменные при выходе из процедуры удаляются из памяти и при новом вызове процедуры инициализируются заново. Если объявить переменную как *статическую*, то при выходе из процедуры ее содержимое сохраняется (хотя сама переменная оказывается недоступной), при новом же вызове процедуры переменная получает значение, которое она имела при последнем выходе из данной процедуры.

Период существования переменной означает, как долго переменная является доступной. При выходе из процедуры переменная удаляется. Переменные уровня модуля доступны все время работы приложения. При очередном вызове этой процедуры переменная заново инициализируется. Переменные со словом *Static* в процедурах и функциях при повторном вызове этой процедуры или функции не будут заново инициализироваться.

В программировании часто используется особый тип переменных – *массивы*. Массив (вектор) – это набор однотипных переменных, объединенных одним именем и доступных через это имя и порядковый номер переменной в

наборе. Элементы массива обладают непрерывной нумерацией определенного диапазона.

Различают *статические* и *динамические* массивы.

Границы статических массивов устанавливаются на этапе разработки и могут изменяться только в новой версии программы. Динамические массивы изменяют свои границы в ходе выполнения программы.

В общем случае индексирование элементов начинается с 0, однако использование оператора *Option Base 1* позволяет установить индексацию с 1. Данный оператор указывается в секции (*General Declarations*) контейнера. Границы индексации можно установить в опции *РАЗМЕРНОСТЬ*.

Объявления массива отличаются от обычной переменной только индексом, указанным в скобках. Индекс показывает размерность массива. Массивы можно делать многомерными:

Dim ИМЯ_МАССИВА ({A TO} B, { C TO } D) As ТИП

(где A – номер первого индекса; B – номер последнего индекса).

Если первый индекс 0, можно задать одно число – последний индекс.

Динамические массивы – это такие массивы, размерность которых может меняться в ходе работы программы:

Dim ИМЯ_МАССИВА () As ТИП

В отличие от массивов статичных размеров, когда обращаться к элементам можно сразу после его объявления, к элементам динамического массива сразу обращаться нельзя, т. к. они еще не инициализированы.

Новую размерность массива нужно указать с помощью оператора *ReDim*:

ReDim ИМЯ_МАССИВА (A) As ТИП

Если в коде программы требуется узнать текущие размеры массива, то можно использовать встроенные функции *Visual Basic*:

LBound – возвращает нижнюю границу массива;

UBound – возвращает верхнюю границу массива.

Шаблон объявления массивов имеет вид

Static | Public | Dim [Shared] ПЕРЕМЕННАЯ [РАЗМЕРНОСТЬ]

(где РАЗМЕРНОСТЬ указывает диапазон изменения индекса массива, индекс является переменной величиной и используется для указания конкретного элемента массива).

РАЗМЕРНОСТЬ имеет следующий вид:

$$m \text{ To } n,$$

где m – начальное значение индекса;

n – конечное значение индекса.

Если при обработке массива значение индекса выходит из указанного в объявлении диапазона, выдается сообщение об ошибке.

В данном параметре можно через запятую указать диапазон изменения для нескольких индексов (многомерные массивы).

Необходимо подчеркнуть, что в *Visual Basic* параметры можно передавать двумя способами: 1) по значению и 2) по ссылке.

Первый способ передает в процедуру (или функцию) только значение передаваемой переменной, и внутри функции изменить значение переменной нельзя.

Второй способ передает в функцию ссылку на передаваемую переменную, и ее значение легко может быть изменено в теле функции.

По умолчанию *Visual Basic* передает параметры по ссылке. Чтобы параметры передавались по значению перед именем переменной необходимо поставить слово *ByVal* (*By Value*):

```
Public Function CalcDiscremenant (_ByVal a As Double, _ByVal b As Double, _ByVal c As Double) As Double
```

Это повышает наглядность. Соответственно становится понятно, что значения передаваемых переменных внутри функции меняться не будут.

Для выполнения приложения (см. рис. 3.11) также требуется объявление использующихся переменных.

На рис. 3.16 показано сообщение об ошибке при выполнении программы и выделен элемент кода, где она произошла.

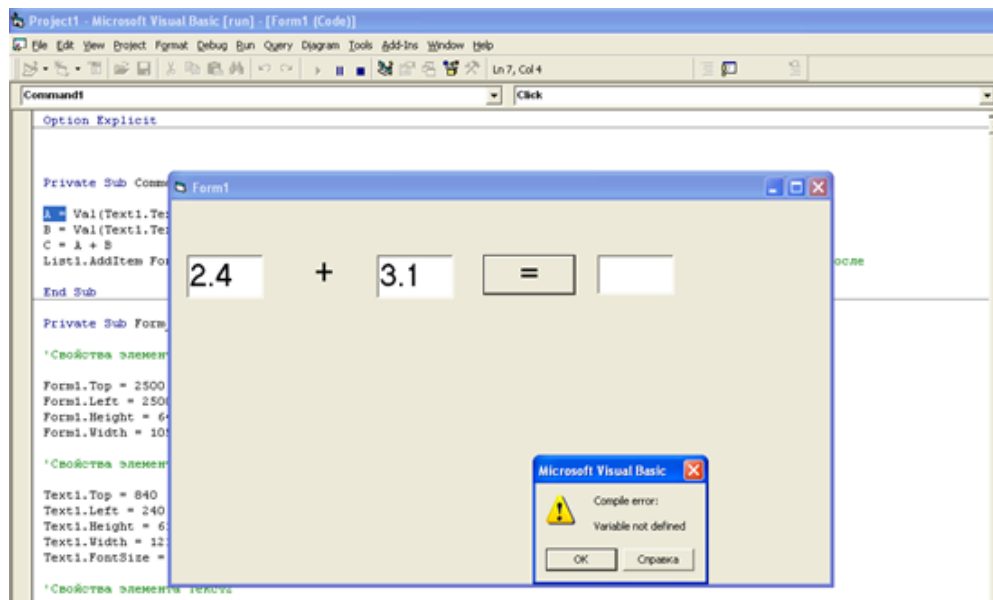


Рис. 3.16. Ошибка выполнения приложения

Для объявления имеющихся переменных A , B и C в области *Declarations* применяется следующий код:

Option Explicit 'Запись условия явного объявления переменных
 Dim A As Double, B As Double, C As Double 'Объявление переменных с
 'двойной точностью

Для запуска приложения выполним команды *Run* → *Start* (или *F5*). Тогда произойдет событие загрузки стартовой формы *Form1* (рис. 3.17).

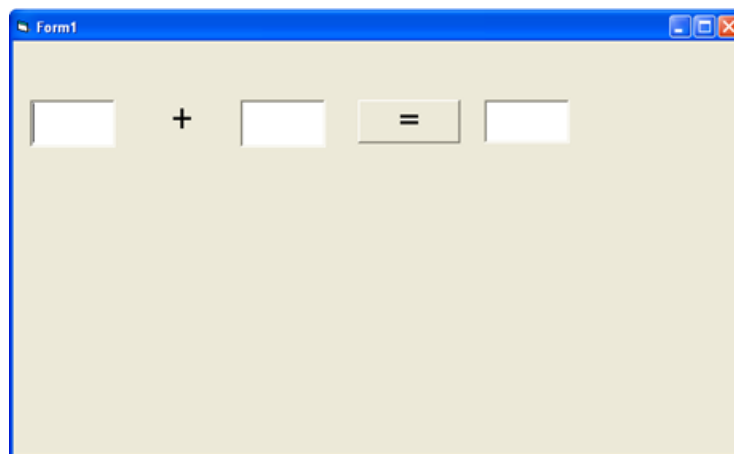


Рис. 3.17. Приложение в режиме пуска

Чтобы произошел ввод исходных данных, нужно в пустые ячейки (см. рис. 3.17) внести требуемые (или произвольные) числа (рис. 3.18). Для получения результата расчетов требуется нажать командную кнопку «=».

Активация данной командной кнопки запускает событие *Command1_Click()*. С событием *Command1_Click()* связан исполнительный код упомянутой процедуры (см. с. 119). После выполнения исполнительного кода в предусмотренной области формы *Form1* появляется ответ (рис. 3.19).

Для завершения работы требуется выполнить команды *Run* \rightarrow *End*.

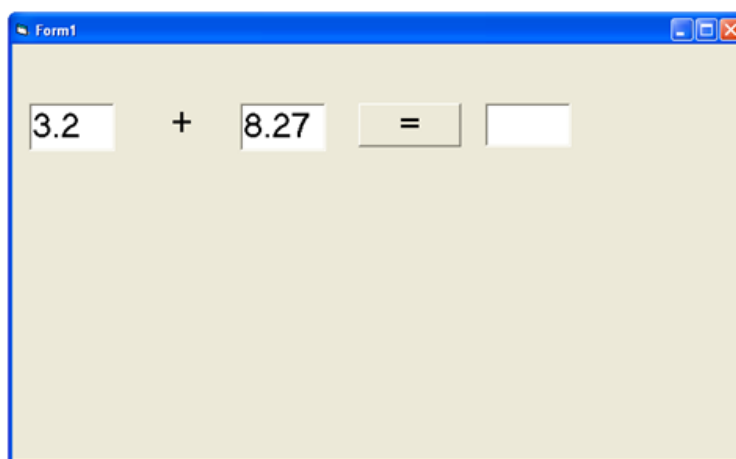


Рис. 3.18. Ввод исходных данных

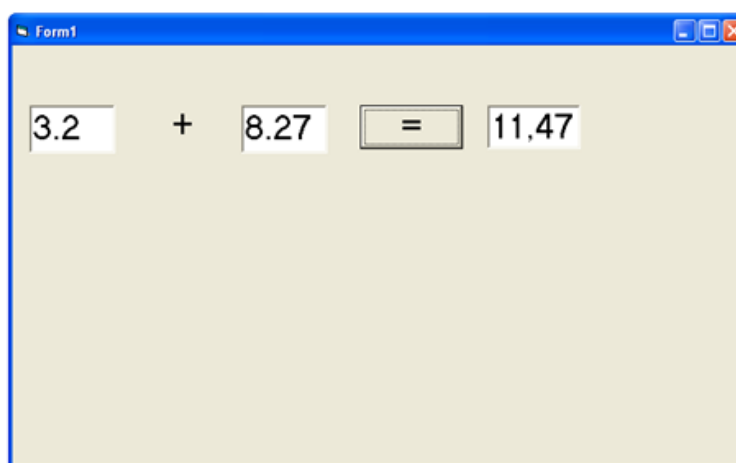


Рис. 3.19. Получение результата расчета

В связи с тем, что более сложные приложения будут состоять из нескольких форм и модулей, при разработке рекомендуется для каждого проекта создавать отдельную папку хранения всех файлов. На основе завершенной программы можно создать приложение, которое будет работать в среде *Windows*. Для этого необходимо в меню *FILE* выбрать команду *MAKE EXE* и записать название приложения.

РАЗДЕЛ 4

ПРИМЕР ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

4.1. Пример выполнения кинематического расчета рычажного механизма

4.1.1. Исходные данные для расчета механизма

Рассмотрим пример разработки программы кинематического анализа плоского рычажного механизма II класса (рис. 4.1).

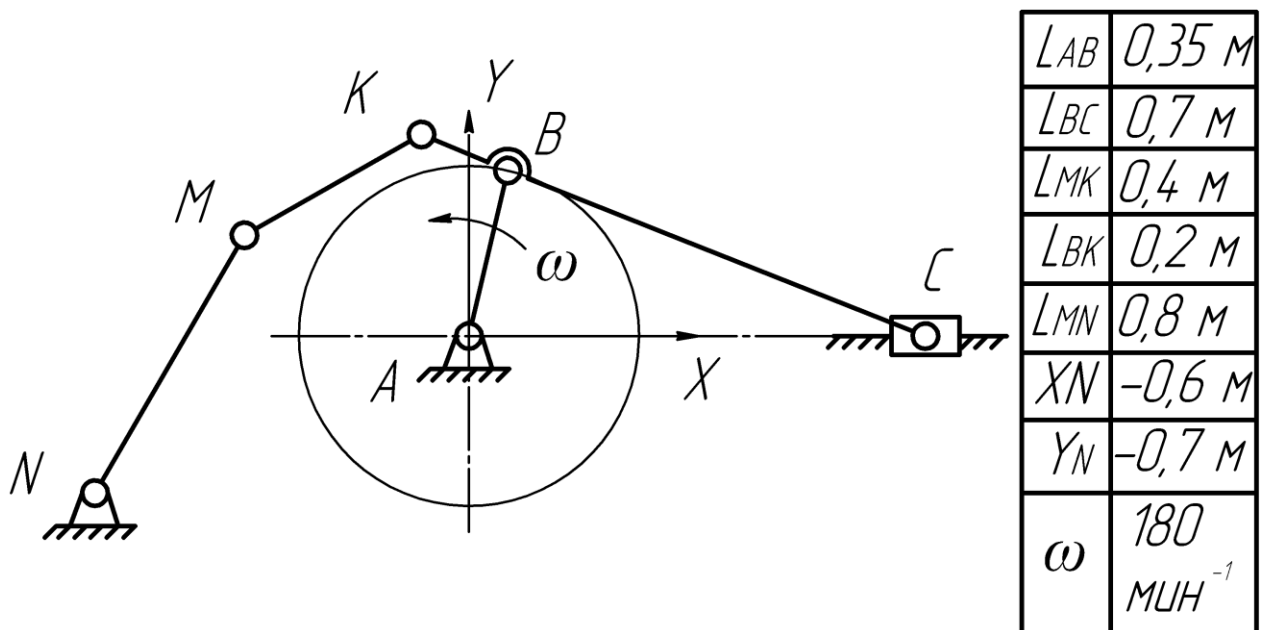


Рис. 4.1. Исходные данные
плоского рычажного механизма II класса

Первым этапом решения любой инженерной задачи является постановка ее цели. Цель данной задачи – определение координат, скоростей и ускорений звеньев и точек рассматриваемого механизма в зависимости от обобщенной координаты – угла поворота входного звена.

Вначале определяется список достаточных исходных данных: геометрические размеры звеньев механизма и закон движения входного звена.

4.1.2. Структурный анализ механизма

Следующий этап решения задачи заключается в разработке математической модели. Исследовать стержневые рычажные механизмы II класса предлагается методом замкнутых векторных контуров (см. разд. 1). Этот метод основывается на исследовании отдельных структурных групп механизма.

При этом нужно помнить, что исследование структурной группы может быть проведено только после рассмотрения предыдущей группы, а соответственно, требуется провести структурный анализ схемы механизма. Обозначим номера звеньев механизма начиная с входного (рис. 4.2).

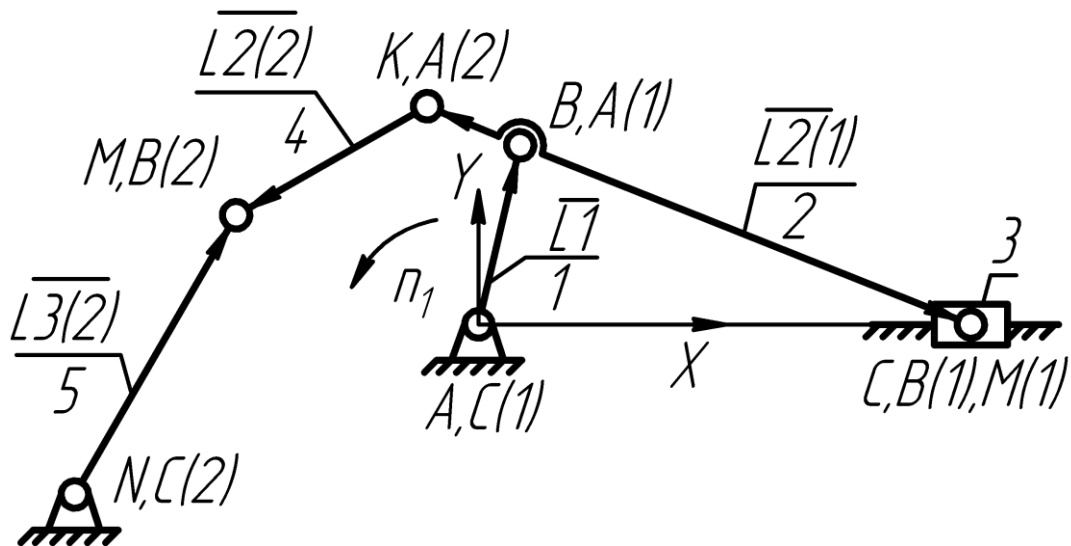


Рис. 4.2. Расчетная схема кинематического анализа

Ограничимся записью структурной формулы строения механизма:

$$I(0,1) \rightarrow II(2,3) \rightarrow II(4,5).$$

$$j = 1; \quad j = 2.$$

Структурный анализ показывает, что механизм состоит из первичного механизма – входного звена (кривошип AB), группы Ассура 2–3 II класса 2 вида, группы Ассура 4–5 II класса 1 вида.

В порядке присоединения к входному звену группа Ассура 2–3 имеет номер $j = 1$, а группа Ассура 4–5 имеет номер $j = 2$.

4.1.3. Блок-схема программы кинематического расчета механизма

Для численного решения задачи на ЭВМ по разработанной математической модели необходимо составить логическую схему вычислительного процесса, т. е. точно, полно и однозначно определить последовательность этого процесса. На этом этапе (при необходимости) разрабатывается блок-схема программы кинематического расчета (рис. 4.3). Блок-схема отражает последовательность этапов выполнения программы кинематического расчета рассматриваемого механизма.

Преимущество алгоритма программы заключается в использовании рассмотренных ранее подпрограмм, организованных в виде процедур, помещенных в модуль *Modul1 (OAP.BAS)*, хранящийся на жестком диске ЭВМ.

Расчет ведется в соответствии со структурной формулой механизма в прямом порядке: начиная с первичного механизма.

Кинематические параметры механизма I класса рассчитываются с применением подпрограмм *PKT* и *PA*. Для кинематического анализа группы Ассура 2–3 II класса 2 вида необходимы подпрограммы *PK2*, *PA* и *PKT*. Кинематический расчет группы Ассура 4–5 II класса 1 вида проводится с помощью подпрограмм *PK1*, *PA* и *PKT*.

В соответствии с тем, что группа Ассура 2–3 имеет номер $j = 1$ (в порядке присоединения к входному звену), группа Ассура 4–5 имеет номер $j = 2$ (см. подразд. 4.1.2), каждая группа получает новые обозначения векторов и точек звеньев (см. рис. 4.2). Эти новые обозначения необходимы для отличия структурных групп друг от друга при применении единого кода подпрограмм для совпадающих по классификации групп (т. е. для одинаковой математической модели, а значит, и подпрограммы).

Как сказано ранее, исполнительный код подпрограмм, необходимых для расчета кинематических параметров групп Ассура, содержится в специальном модуле *Modul1 (OAP.BAS)* (**путь к этому файлу, хранящемуся на жестком диске компьютера, необходимо узнать у преподавателя**).

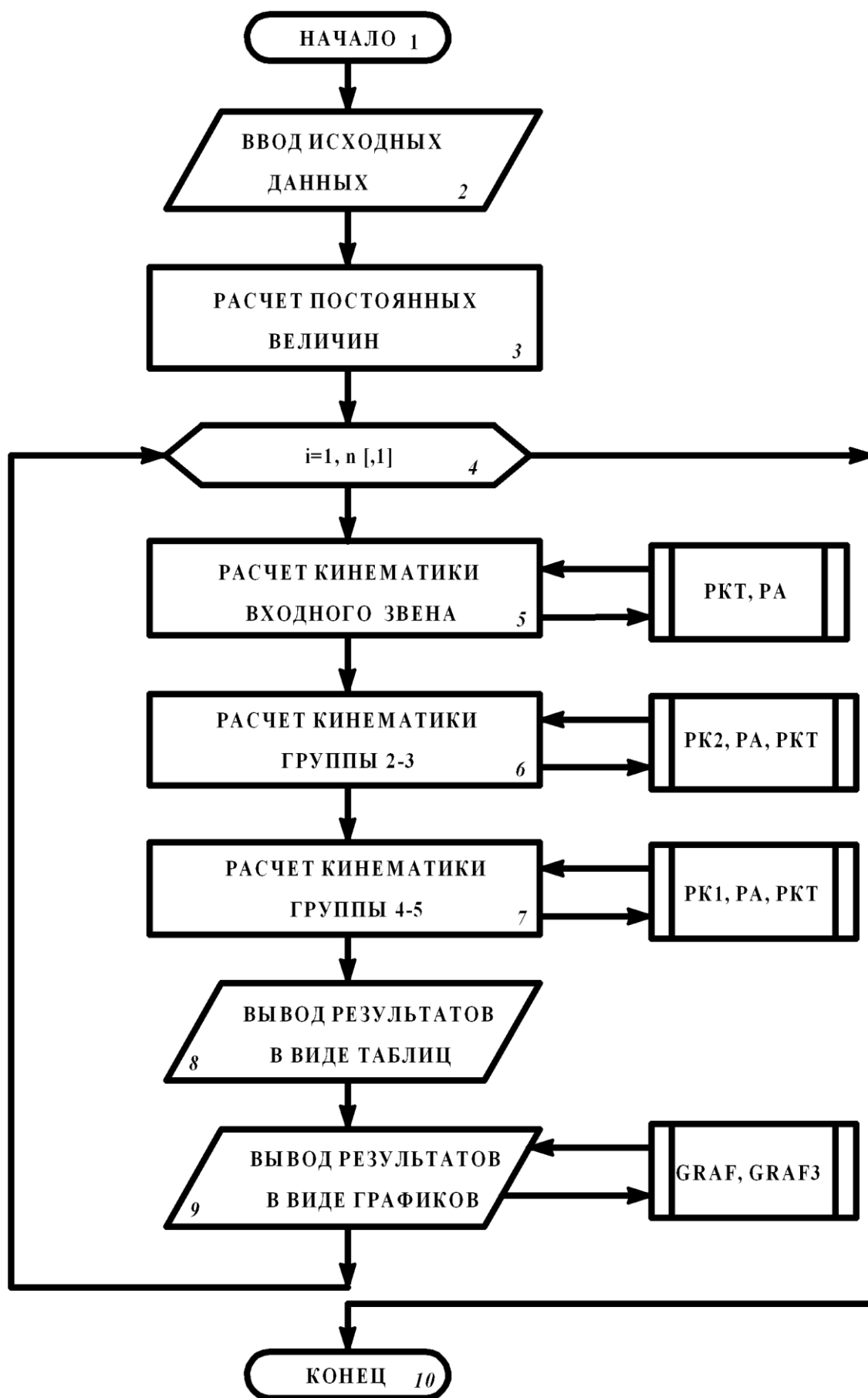


Рис. 4.3. Блок-схема программы кинематического расчета

Блок 3 «Расчет постоянных величин» (см. рис. 4.3) отражает необходимость определить некоторые входные параметры для всей программы: число π , коэффициент перевода угловых величин из радиан в градусы и наоборот, число расчетных точек цикла, угловую скорость вращения входного звена и угловую координату входного звена в начальный момент расчетов (начальный угол звена I). (В конкретном случае могут понадобиться дополнительные параметры.)

Для применения встроенных тригонометрических функций необходимо определить число π по формуле

$$\pi = 4 \cdot \arctg(1). \quad (4.1)$$

Все аргументы тригонометрических функций необходимо подставлять в радиальной мере. Для этого помножим их на коэффициент

$$RAD = \frac{\pi}{180}. \quad (4.2)$$

Для обратного действия имеется коэффициент

$$DEG = \frac{1}{RAD}. \quad (4.3)$$

Формула для перевода величины угловой скорости из об/мин в рад/с

$$OM = \frac{\pi \cdot n}{30}, \quad (4.4)$$

где OM – угловая скорость, рад/с;

n – угловая скорость, об/мин.

Число расчетных точек зависит от шага расчета:

$$N = \frac{360}{h} + 1, \quad (4.5)$$

где 360 – выраженный в градусах путь входного звена за один цикл движения механизма;

h – задаваемый пользователем шаг расчета, град.

Начальный угол поворота входного звена $F1n$ определяется для каждого конкретного механизма из условия нахождения выходного звена в одном из крайних положений. Для рассматриваемого механизма одно из крайних по-

ложений соответствует $F1n = 0^\circ$, когда кривошип совпадает с горизонтальной осью OX , а звено 3 находится на максимальном удалении от вала входного звена.

4.1.4. Создание проекта

На этапе составления программы для VB-проекта разрабатываются имена (идентификаторы) и структура используемых данных (типы величин), назначаются устройства ввода-вывода информации.

Для выполнения блока 1 (см. рис. 4.3) вначале необходимо:

- запустить оболочку VB;
- в меню *FILE* выбрать команду *NEW PROJEKT*: появятся окно формы *Form1* (по умолчанию) и панель проекта *Projekt1* с указанием единственной формы *FORM1.frm* (рис. 4.4).

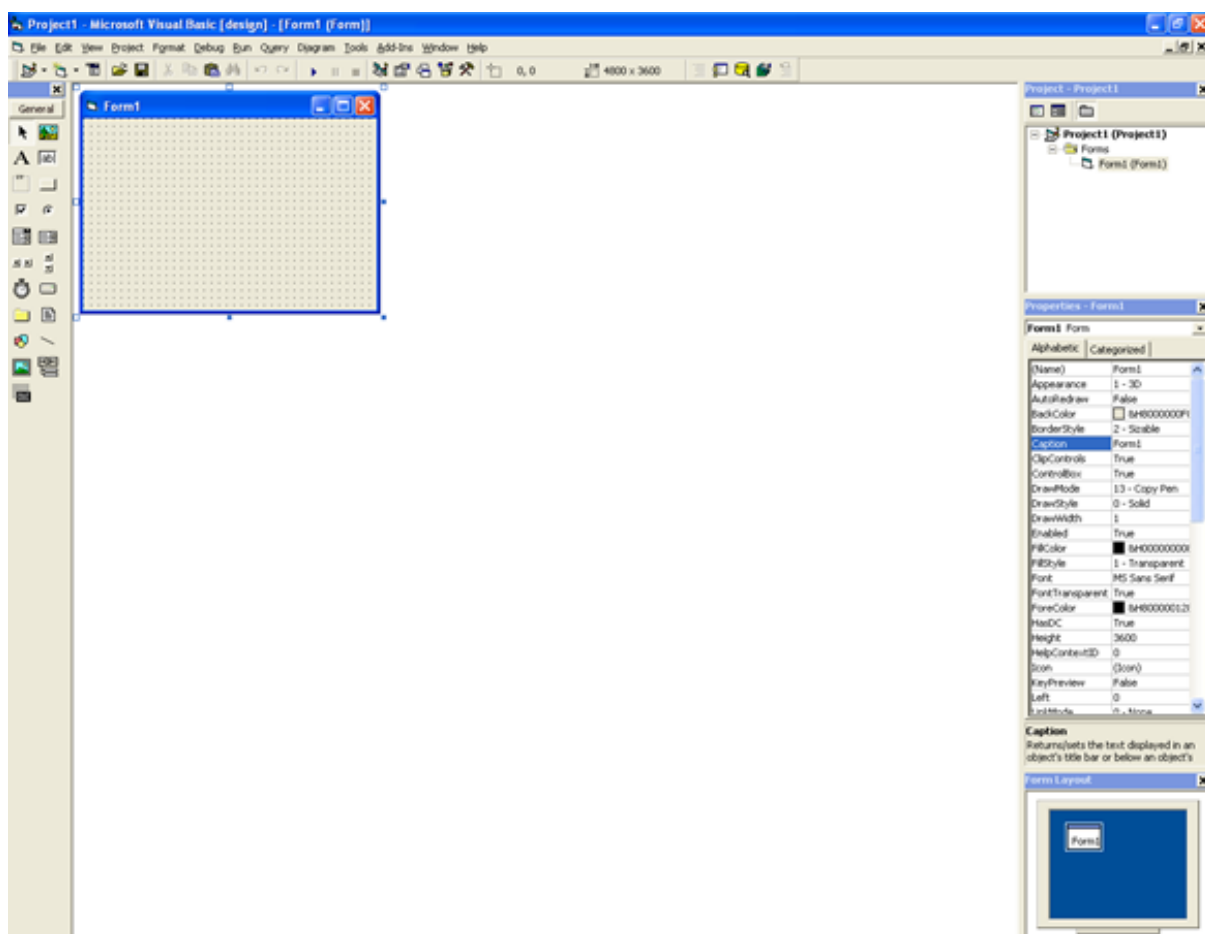


Рис. 4.4. Макет первоначальной формы проекта

Добавим в проект модуль *Modul1 (OAP.BAS)*. В меню *PROJEKT* выберем команду *ADD FILE...* и найдем требуемый файл на жестком диске. Дадим название проекту (*FILE – SAVE PROJEKT AS...*) и форме (*FILE – SAVE FILE AS...*) (рис. 4.5).

Для выполнения блока 2 (см. рис. 4.3) расположим на *Form1* (рис. 4.6) с помощью мыши следующие объекты управления, имеющиеся на панели инструментов (см. рис. 3.7): *Lable1*, 2, ..., 9 (метки 1, 2, ..., 9); *Text1*, 2, ..., 8 (ячейки 1, 2, ..., 8); *CommandButton1* (командная кнопка 1); *PictureBox1* (рисунок 1).

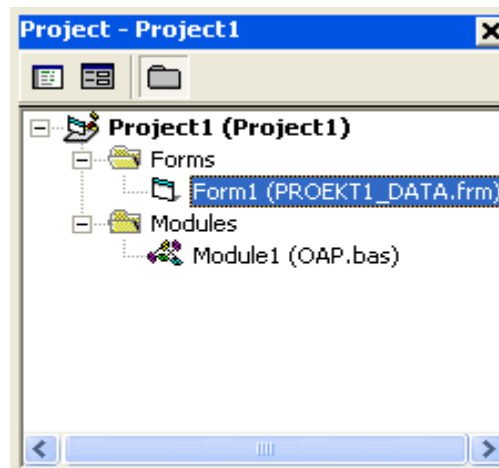


Рис. 4.5. Дерево проекта

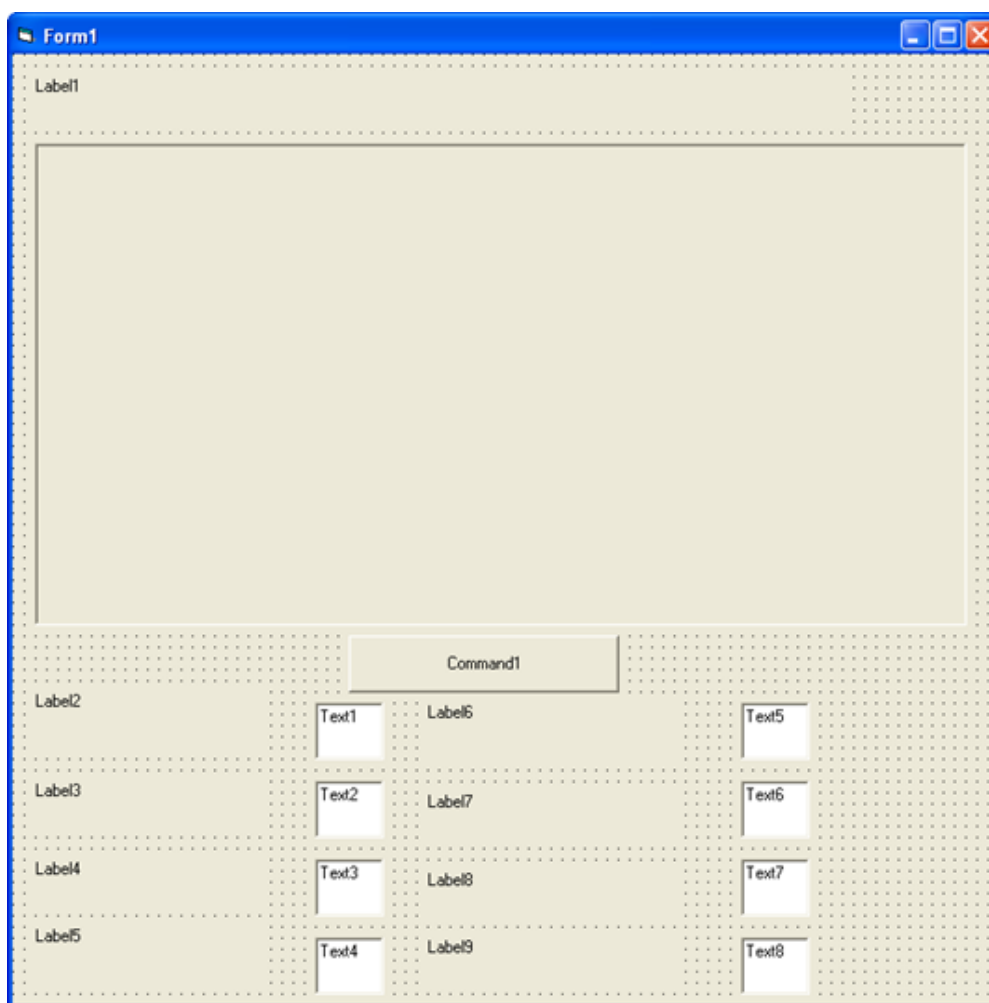


Рис. 4.6. Интерфейс формы *Form1*

Назначение объектов формы:

- метка *Label1* – название работы;
- ячейки *Text1, 2, ..., 8* – ввод значений исходных данных;
- метки *Label2, ..., 9* – пояснение для ячеек *Text1, 2, ..., 8*;
- *CommandButton1* (командная кнопка 1) – подтверждение внесенных исходных данных;
- *PictureBox1* (рисунок 1) – область для размещения схемы механизма.

Предварительно созданное изображение схемы механизма в область *PictureBox1* можно загрузить из буфера (рис. 4.7).

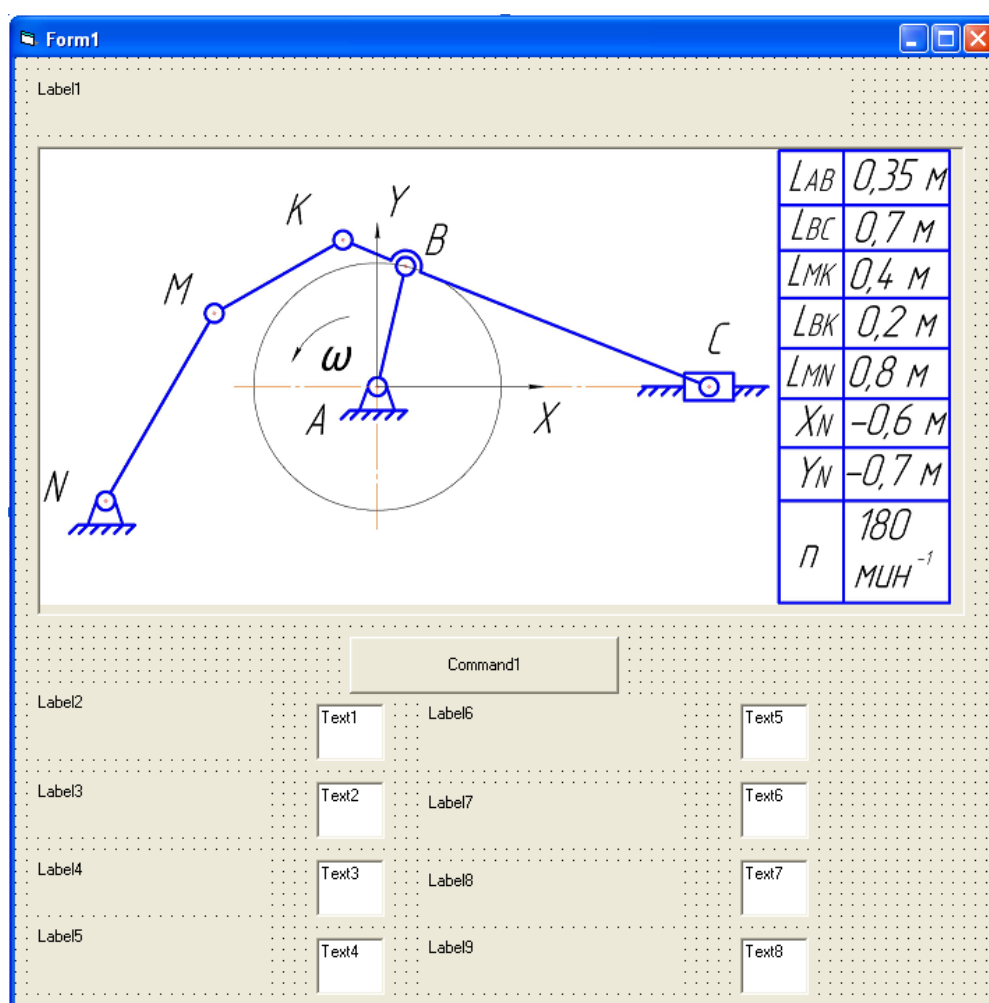


Рис. 4.7. Размещение графического изображения на форме *Form1*

Свойства этих объектов можно задать:

- с помощью меню свойств (рис. 4.8) (см. подразд. 3.4);
- программным методом при реализации события *Form_Load()* (загрузка формы).

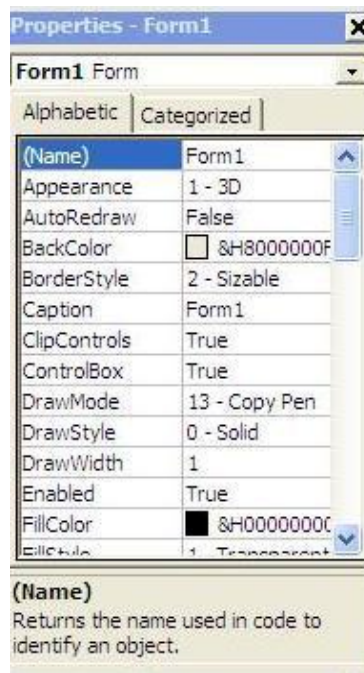


Рис. 4.8. Меню свойств формы Form1

После двойного клика на самой форме осуществляется переход в область редактирования кода.

Исполнительный код процедуры (пояснения см. в подразд. 3.5).

```
Private Sub Form_Load() 'Процедура при реализации события Загрузка формы
Form1.Caption = "ЗАДАНИЕ" 'Заголовок формы
Form1.Left = 1500 'Расстояние между внутренним левым краем объекта и
'левым краем объекта, внутри которого он расположен
Form1.Top = 1200 'Расстояние от внутреннего верхнего края объекта до
'верхнего края объекта, внутри которого он расположен
Form1.Width = 10665 'Ширина объекта
Form1.Height = 10710 'Высота объекта
Form1.BackColor = &HFF& 'Цвет фона объекта
Label1.FontSize = 20 'Размер шрифта
Label1.Caption = "КИНЕМАТИЧЕСКИЙ АНАЛИЗ МЕХАНИЗМА" 'Надпись
Label1.AutoSize = -1 'Изменение размера объекта под его содержимое
Label1.Left = 240: Label1.Height = 495 : Label1.Top = 240 : Picture1.Left = 240
Picture1.AutoSize = -1 : Picture1.Top = 940
Command1.Caption = "ИСХОДНЫЕ ДАННЫЕ"
Command1.FontSize = 16 : Command1.Height = 615
Command1.Left = 3600 : Command1.Top = 6100
Command1.Width = 4000 : Command1.TabIndex = 2
Label2.FontSize = 16 : Label2.Caption = "AB= в м"
Label2.Left = 240 : Label2.Height = 600
Label2.Top = 6800 : Label2.Alignment = 2
Label2.Width = 2415
```

```

Label3.FontSize = 16 : Label3.Caption = "BC= в м"
Label3.Left = Label2.Left : Label3.Height = 600
Label3.Top = Label2.Top + 800 : Label3.Alignment = 2
Label4.FontSize = 16 : Label4.Caption = "BK= в м"
Label4.Left = Label2.Left : Label4.Height = 600
Label4.Top = Label3.Top + 800
Label4.Alignment = 2 'Выравнивание текста по центру объекта
Label5.FontSize = 16 : Label5.Caption = "KM= в м"
Label5.Left = Label2.Left : Label5.Height = 600
Label5.Top = Label4.Top + 800 : Label5.Alignment = 2
Text1.FontSize = 16 : Text1.Left = 3240
Text1.Height = 600 : Text1.Top = Label2.Top
Text1.Text = "0.35" 'Присвоение числового значения
Text2.FontSize = 16 : Text2.Text = "0.7"
Text2.Left = Text1.Left : Text2.Height = 600
Text2.Top = Label3.Top
Text3.FontSize = 16 : Text3.Text = "0.2"
Text3.Left = Text1.Left : Text3.Height = 600
Text3.Top = Label4.Top : Text4.FontSize = 16
Text4.Text = "0.45" : Text4.Left = Text1.Left
Text4.Height = 600 : Text4.Top = Label5.Top
Label6.FontSize = 16 : Label6.Caption = "MN= в м"
Label6.Left = 4440 : Label6.Height = 600
Label6.Top = Label2.Top : Label6.Alignment = 2
Label7.FontSize = 16 : Label7.Caption = "XN= в м" : Label7.Left = Label6.Left
Label7.Height = 600 : Label7.Top = Label3.Top : Label7.Alignment = 2
Label8.FontSize = 16 : Label8.Caption = "YN= в м" : Label8.Left = Label6.Left
Label8.Height = 600 : Label8.Top = Label4.Top : Label8.Alignment = 2
Label9.FontSize = 16 : Label9.Caption = "w= в м" : Label9.Left = Label6.Left
Label9.Height = 600 : Label9.Top = Label5.Top : Label9.Alignment = 2
Text5.FontSize = 16 : Text5.Text = "0.8" : Text5.Left = 7800
Text5.Height = 600 : Text5.Top = Label2.Top
Text6.FontSize = 16 : Text6.Text = "-0.6" : Text6.Left = Text5.Left
Text6.Height = 600 : Text6.Top = Label3.Top
Text7.FontSize = 16 : Text7.Text = "-0.7" : Text7.Left = Text5.Left
Text7.Height = 600 : Text7.Top = Label4.Top : Text8.FontSize = 16
Text8.Text = "180" : Text8.Left = Text5.Left : Text8.Height = 600
Text8.Top = Label5.Top
End Sub 'Завершение подпрограммы

```

Выполнение этого кода произойдет после загрузки стартовой формы *Form1*, условие которой – запуск приложения (*RUN – START/F5*). По виду формы понятно, что она предназначена для ввода исходных данных (рис. 4.9).

ЗАДАНИЕ

КИНЕМАТИЧЕСКИЙ АНАЛИЗ МЕХАНИЗМА



L_{AB}	0,35 м
L_{BC}	0,7 м
L_{MK}	0,4 м
L_{BK}	0,2 м
L_{MN}	0,8 м
X_N	-0,6 м
Y_N	-0,7 м
ω	180 мин^{-1}

ИСХОДНЫЕ ДАННЫЕ

AB= в м		MN= в м	
BC= в м		XN= в м	
BK= в м		YN= в м	
KM= в м		w= в м	

Рис. 4.9. Пуск проекта

ЗАДАНИЕ

КИНЕМАТИЧЕСКИЙ АНАЛИЗ МЕХАНИЗМА



L_{AB}	0,35 м
L_{BC}	0,7 м
L_{MK}	0,4 м
L_{BK}	0,2 м
L_{MN}	0,8 м
X_N	-0,6 м
Y_N	-0,7 м
ω	180 мин^{-1}

ИСХОДНЫЕ ДАННЫЕ

AB= в м	0.35	MN= в м	0.8
BC= в м	0.7	XN= в м	-0.6
BK= в м	0.2	YN= в м	-0.7
KM= в м	0.45	w= в м	180

Рис. 4.10. Ввод исходных данных

После ознакомления с механизмом в текстовые ячейки нужно записать числовые значения указанных параметров (рис. 4.10) и подтвердить эти действия нажатием командной кнопки *ИСХОДНЫЕ ДАННЫЕ*.

(*Примечание:* поскольку при отладке программы необходимы ее многочисленные пуски, с целью экономии времени рекомендуется сразу сохранить значения ячеек *Text1*, 2, ..., 8.)

В процедуре события *Command1_Click()* (нажатия командной кнопки) происходит присвоение соответствующим переменным числовых значений из текстовых ячеек. После чего следует обеспечить скрывание текущей формы (код записан в *Form1*) и загрузку следующей формы *Form2*.

После двойного клика по объекту *CommandButton1* осуществляется переход в область редактирования кода.

Исполнительный код этих действий следующий:

```
Private Sub Command1_Click() 'Процедура при реализации события
                              'нажатие командной кнопки
AB = Val(Text1) 'Присвоение переменной
BC = Val(Text2) 'значения соответствующей ячейки
BK = Val(Text3) : MK = Val(Text4) : MN = Val(Text5) : XN = Val(Text6)
YN = Val(Text7) : W1 = Val(Text8) : j = 2
Me.Hide 'Скрыть текущую форму
Form2.Show 'Показать форму Form2
End Sub
```

В приведенном коде появляются переменные *AB*, *BK* и т. д. Для успешной работы программы требуется объявить их типы и области действия.

Поскольку проект состоит из нескольких форм, а указанные переменные являются исходными данными, которые могут потребоваться в любом месте кода, их необходимо сделать видимыми из любой области программы, т. е. объявить их в качестве глобальных переменных. Для этого нужно при добавлении в проект нового модуля *Module2* в его вкладке *Declaration* (рис. 4.11) записать следующий код:

```
Option Explicit 'Появляется по умолчанию
Global AB As Double, BC As Double, BK As Double, MK As Double
Global MN As Double, XN As Double, YN As Double, W1 As Double
```



Рис. 4.11. Расположение исполнительного кода

Для выполнения блоков 3–8 (см. рис. 4.3) понадобится форма *Form2* (рис. 4.12). Для создания и сохранения формы *Form2* необходимо выполнить следующие действия: *ПРОЕКТ – ADD FORM*; *ФАЙЛ – SAVE FILE AS...*

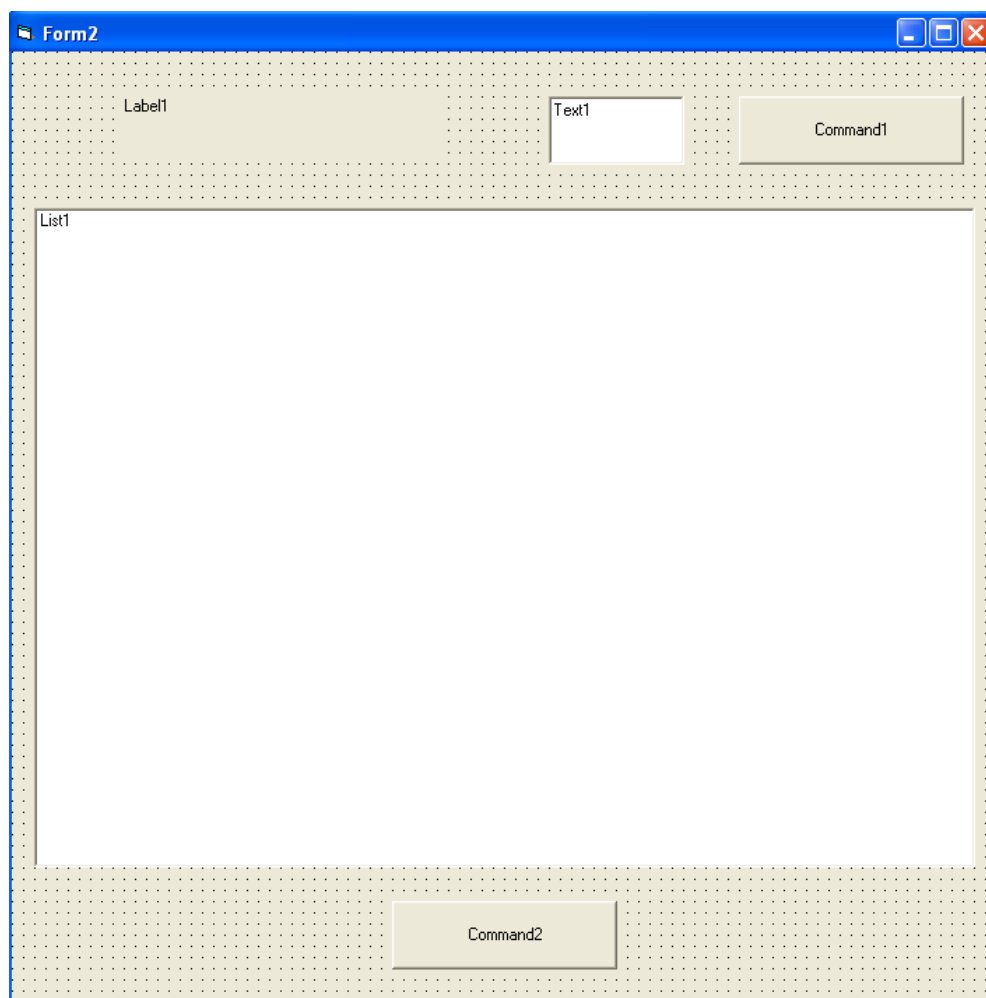


Рис. 4.12. Интерфейс формы *Form2*

На форме *Form2* следует расположить объекты (см. рис. 4.12):

- *Text1* (ячейка 1) – ячейка для ввода значения шага расчета;
- *Lable1* (метка 1) – пояснение для ячейки *Text1*;
- *List1* (область спискового окна) – область для вывода результатов в табличном виде;
- *CommandButton1* (командная кнопка 1) – сигнал к выполнению кода программы кинематического расчета механизма;
- *CommandButton2* (командная кнопка 2) – сигнал о переходе к другой форме, предназначенной для вывода графиков результатов расчетов.

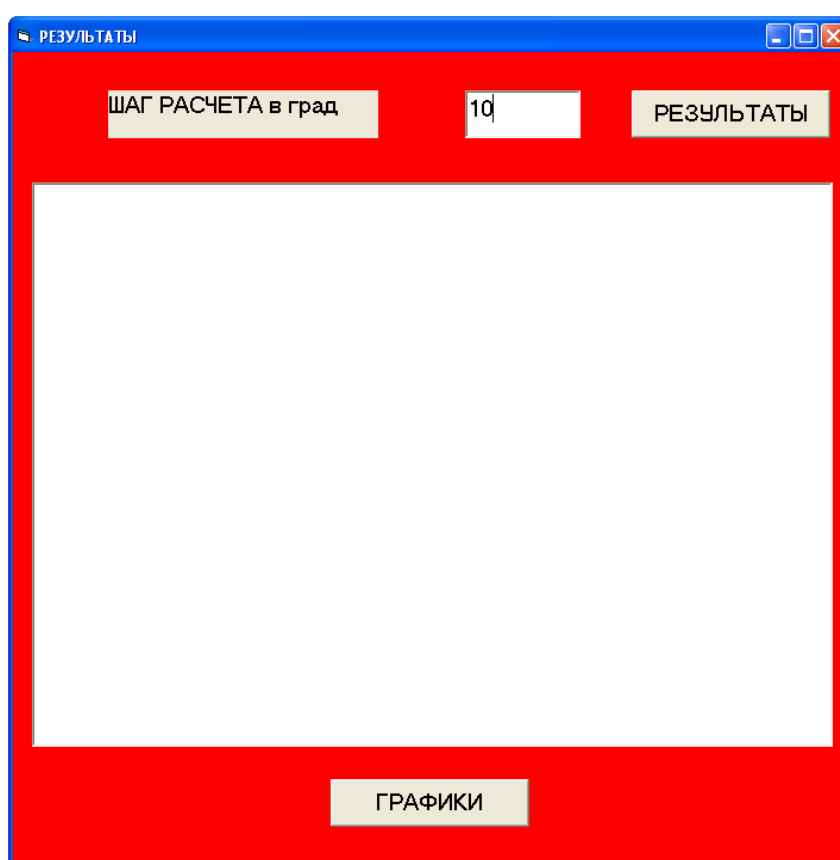


Рис. 4.13. Ввод данных шага расчета

Параметры объектов управления для всех форм задаются аналогично. В данном случае используется программный метод. Исполнительный код для события *Form_Load()*:

```
Private Sub Form_Load()  
'Процедура при реализации события Загрузка формы  
Form2.Caption = "РЕЗУЛЬТАТЫ" : Form2.Left = 1500  
Form2.Top = 1200 : Form2.Width = 10650
```

```

Form2.Height = 10700 : Form2.BackColor = &HFF&
Label1.FontSize = 16 : Label1.Caption = "ШАГ РАСЧЕТА в град"
Label1.Left = 1200 : Label1.Height = 600
Label1.Top = 480 : Label1.Width = 3400
Label1.TabIndex = 1 : Label1.WordWrap = -1
Text1.FontSize = 16 : Text1.Text = "1"
Text1.Left = 5700 : Text1.Height = Label1.Height
Text1.Top = Label1.Top
Command1.FontSize = 16
Command1.Caption = "РЕЗУЛЬТАТЫ"
Command1.Height = Label1.Height
Command1.Left = 7800
Command1.Top = Label1.Top
Command1.Width = 2500
Command2.FontSize = 16
Command2.Caption = "ГРАФИКИ"
Command2.Height = Label1.Height
Command2.Left = 4000
Command2.Top = 9120
Command2.Width = 2500
List1.Left = 240 : List1.Top = 1640
End Sub

```

После загрузки *Form2* следует занести значение шага расчета в предназначенную ячейку (см. рис. 4.13). При нажатии кнопки *РЕЗУЛЬТАТЫ* произойдет выполнение программы в соответствии с блок-схемой (блоки 3–8).

Исполнительный код подпрограммы следующий:

```

Sub Command1_Click () Rem Код события – активации CommandButton1
List1.Clear Rem очистка объекта List1
Rem формирование шапки таблицы
znak1 = "      F1      F2      F4      F5      Sc      Vc      Ac"
znak2 = "      град      град      град      град      м      м/с      м/с^2"
znak3 = "===== "
Rem Вывод шапки таблицы в области List1
List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak1, "")
List1.AddItem Format$(znak2, "")
List1.AddItem Format$(znak3, "")
Rem Преобразование динамических массивов
J = 2 Rem Размерность динамических массивов
ReDim L2(J), l3(J), xa(J), ya(J), xc(J), yc(J), z(J) va(J), fva(J), wa(J)
ReDim fwa(J), vc(J), fvc(J), wc(J), fwc(J), rd(J), BD(J)
Rem Ввод исходных данных
L1 = AB: L2(1) = BC: L3(1) = 0: L2(2) = МК: L3(2) = MN : XC(1) = 0

```

```

YC(1) = 0: XC(2) = XN: YC(2) = YN : Z(2) = -1 : h = Val(Text1)
Rem расчет постоянных величин
PI = 4 * Atn(1) Rem Число ПИ
rad = PI / 180 Rem коэффициент перевода градусов в радианы
deg = 1 / rad Rem коэффициент перевода радиан в градусы
OM = PI * W1 / 30 Rem перевод угловой скорости звена 1 из мин-1 в рад/с
n = 360 / h + 1 Rem определение числа расчетных точек
f1n = 0 Rem определение начального положения звена 1
Rem условие получения результата за один оборот
If f1n > 2 * PI Then f1n = f1n - 2 * PI
Rem Перевод угловых величин из градусов в радианы
fx31(1) = 90 * RAD
Rem Подготовка массивов переменных, предназначенных для вывода
ReDim hm(n), fungr1(n), fungr2(n), fungr3(n), fungr4(n), fungr5(n), y(n), fun1(n)
Rem Кинематический расчет механизма
For i = 1 To n Step 1 ' Открытие цикла с шагом 1
Rem Кинематический расчет звена 1
Rem задание изменение угла поворота звена 1 через переменную цикла
f1 = f1n + (i-1)*h* rad : F11 = 1: F12 = 0 ' аналоги скорости и ускорения звена 1
Call pkt(0, 0, L1, f1, 0, xa(1), ya(1)) Rem координаты точки A(1)
Rem аналоги скорости и ускорения точки A(1)
Call pa(0, 0, 0, 0, L1, 0, f1, F11, F12, va(1), fva(1), wa(1), fwa(1))
Rem Кинематический расчет группы 2-3
j = 1 Rem ввод порядкового номера структурной группы 2-3
Call pk2(Lc, fBC, fBB, L1c, F1BC, L2c, f2BC, 0, 0, 0, 0, 0, 0, 0, 0)
Call pkt(xa(1), ya(1), BK, fBC, PI, xa(2), ya(2)) Rem координаты точки A(2)
Rem аналоги скорости и ускорения точки A(2)
Call pa(va(1), fva(1), wa(1), fwa(1), BK, PI, fBC, F1BC, f2BC, va(2), fva(2), wa(2), fwa(2))
Rem Кинематический расчет группы 4-5
j = 2 Rem ввод порядкового номера структурной группы 4-5
Call pk1(fmk, fmn, f1mk, f1mn, f2mk, f2mn, 0, 0, 0, 0, 0, 0, 0, 0)
Rem Формирование массивов результатов с переводом величин в градусы
fungr1(i) = f1 * deg : fungr2(i) = fBC * deg: fungr3(i) = Fmk * DEG
fun1(i) = Lc : fungr4(i) = Fmn * DEG : fungr6(i) = L1c : fungr7(i) = L2c
Rem Расчет перемещения ползуна
fungr5(i) = fun1(i) - fun1(1)
100 Next ' Конец цикла
Rem Вывод результатов в элемент List1 с заданным шагом h
For i = 1 To N Step h
List1.FontSize = 14 Rem Кегль текста таблицы
List1.AddItem Format$(fungr1(i), " 000.0") + " " + Format(fungr2(i), "000.0")_
+ " " + Format(fungr3(i), "000.0") + " " + Format(fungr4(i), "000.0") + " " + _
Format(fungr5(i), "0.000") + " " + Format(fungr6(i), " 0.000") + " " + _
Format(fungr7(i), " 0.000")
Next
400 End Sub Rem Конец подпрограммы

```

Для успешной работы программы требуется объявление типа и области действия использующихся переменных. Глобальные переменные располагаются в разделе *Declaration Module2*.

Rem Переменные формы 1

Global AB As Double Rem Длина звена AB

Global BC As Double Rem Длина звена BC

Global BK As Double Rem Длина звена BK

Global MK As Double Rem Длина звена MK

Global MN As Double Rem Длина звена MN

Global XN As Double Rem Координата опоры N

Global YN As Double Rem Координата опоры N

Global W1 As Double Rem Частота вращения звена AB

Rem Переменные формы 2

Global fungr1() As Double Rem Массив для вывода результатов

Global fungr2() As Double Rem Массив для вывода результатов

Global fungr3() As Double Rem Массив для вывода результатов

Global fungr4() As Double Rem Массив для вывода результатов

Global fungr5() As Double Rem Массив для вывода результатов

Global fungr6() As Double Rem Массив для вывода результатов

Global fungr7() As Double Rem Массив для вывода результатов

Локальные переменные располагаются в разделе *Declaration Form2*.

Option Explicit

Dim znak1 As String 'Строка заголовки таблицы

Dim znak2 As String 'Строка заголовки таблицы

Dim znak3 As String'

Dim N As Double 'Число расчетных точек

Dim h As Double 'Шаг расчета

Dim i As Integer 'Параметр цикла

Dim OM As Integer 'Угловая скорость вращения звена 1 в радианах

Dim L1 As Double 'Длина звена 1

Dim arg As Double 'Вспомогательный угол

Dim RAD As Double 'Коэффициент перевода угла из радиан в градусы

Dim DEG As Double 'Коэффициент перевода угла из градусов в радианы

Dim f1 As Double 'Угловая координата звена 1

Dim f1n As Double 'Угловая координата звена 1 в начальный момент расчета

Dim F11 As Double 'Аналог угловой скорости звена 1

Dim F12 As Double 'Аналог углового ускорения звена 1

Dim Fbc As Double 'Угловая координата звена BC

Dim F1bc As Double 'Аналог угловой скорости звена BC

Dim F2bc As Double 'Аналог углового ускорения звена BC

Dim Fbb As Double 'Угловая координата звена L3(1)

Dim Fmk As Double 'Угловая координата звена MK

Dim Fmn As Double 'Угловая координата звена MN
 Dim F1mk As Double 'Аналог угловой скорости звена МК
 Dim F1mn As Double 'Аналог угловой скорости звена MN
 Dim F2mk As Double 'Аналог углового ускорения звена МК
 Dim F2mn As Double 'Аналог углового ускорения звена MN
 Dim Lc As Double 'Линейная координата точки С
 Dim L1c As Double 'Аналог линейной скорости точки С
 Dim L2c As Double 'Аналог линейного ускорения скорости точки С
 Dim Nazvgr1 As String 'Подпись названия графика

В результате выполнения приведенного кода в области *List1* появятся итоги расчетов кинематических параметров механизма (рис. 4.14). (Список кинематических параметров зависит от схемы механизма, цели расчета и т. п. и может варьироваться согласно представлениям автора работы.)

F1 град	F2 град	F4 град	F5 град	Sc м	Vc м/с	Ac м/с ²
000,0	000,0	174,8	067,8	0,000	00,0	-0000,5
010,0	-005,0	184,5	068,2	-0,008	-00,1	-0000,5
020,0	-009,8	194,2	068,3	-0,031	-00,2	-0000,5
030,0	-014,5	203,9	068,1	-0,069	-00,3	-0000,4
040,0	-018,7	213,6	067,7	-0,119	-00,3	-0000,3
050,0	-022,5	223,2	067,0	-0,178	-00,4	-0000,2
060,0	-025,7	232,4	066,4	-0,244	-00,4	-0000,1
070,0	-028,0	239,2	067,0	-0,312	-00,4	0000,0
080,0	-029,5	238,5	071,7	-0,380	-00,4	0000,1
090,0	-030,0	234,8	077,9	-0,444	-00,4	0000,2
100,0	-029,5	230,5	084,4	-0,502	-00,3	0000,2
110,0	-028,0	225,9	090,7	-0,552	-00,3	0000,3
120,0	-025,7	221,0	096,8	-0,594	-00,2	0000,3
130,0	-022,5	215,9	102,6	-0,628	-00,2	0000,2
140,0	-018,7	210,4	107,9	-0,655	-00,1	0000,2

Рис. 4.14. Итоги расчетов кинематических параметров механизма:

F1, F2, F4 и F5 – углы поворота, град, звеньев *AB*, *BC*, *KM* и *MN*;

S_C , V_C и A_C – перемещение, м, скорость, м/с, ускорение, м/с², точки *C* звена 3

Теперь программу можно остановить (*RUN – END*) либо просмотреть результаты расчетов в виде графиков, нажав командную кнопку *ГРАФИКИ*

(см. рис. 4.14): загружается форма *Form3* (рис. 4.15), на которую выводятся результаты расчетов в виде зависимостей тех или иных параметров от угла поворота входного звена. Для получения графика зависимости кинематических параметров механизма от изменения обобщенной координаты – угла поворота входного звена применяется подпрограмма *graf*.

При нажатии кнопки *ГРАФИКИ* выполняется следующий код:

```
Private Sub Command2_Click()  
Me.Hide "Закрытие текущей формы"  
Form3.Show "Загрузка следующей формы"  
Rem Название графика  
Nazvgr1 = "УГОЛ ПОВОРОТА ВХОДНОГО ЗВЕНА"  
Rem Вывод названия графика в поле Label1  
Form3.Label1.Caption = Nazvgr1  
Rem Построения графика  
Call graf(fungr1(), Form3, 2, Nazvgr1)  
End Sub
```

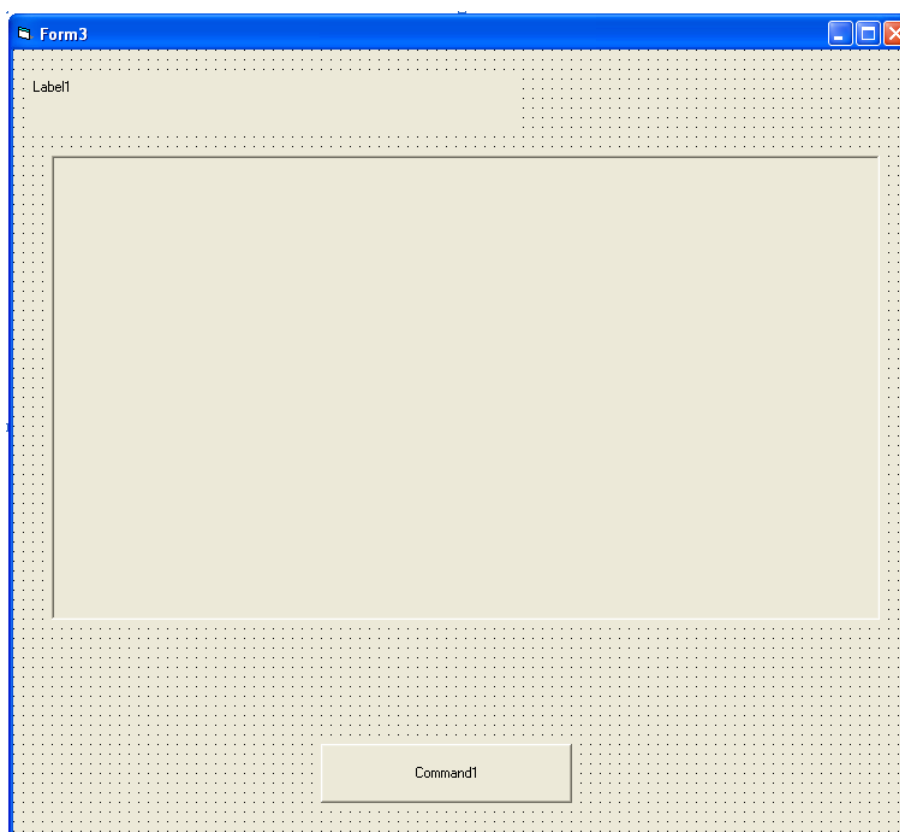


Рис. 4.15. Интерфейс формы *Form3*

Для создания и сохранения формы *Form3* необходимо выполнить последовательность команд *ПРОЕКТ – ADD FORM; FILE – SAVE FILE AS...*

На форме *Form3* расположены элементы:

- *Lable1* (метка 1) – для вывода названия графика;
- *CommandButton1* (командная кнопка 1) – для активации события вывода графиков результатов;
- *PictureBox1* (рисунок 1) – для формирования области, в которой размещаются графики результатов расчетов.

Параметры этих объектов задаются при загрузке формы *Form3*.

Исполнительный код события *Form_Load()*:

```
Private Sub Form_Load()  
Form3.Caption = "ВЫВОД ГРАФИКОВ"  
Form3.Left = Form2.Left  
Form3.Top = 1200  
Form3.Width = Form2.Width  
Form3.Height = Form2.Height  
Label1.FontSize = 16 : Label1.AutoSize = 1  
Label1.Left = 240 : Label1.Height = 500  
Label1.Top = 240  
Command1.FontSize = 16  
Command1.Caption = "СЛЕДУЮЩИЙ"  
Command1.Height = Label1.Height  
Command1.Left = 4000  
Command1.Top = 9120  
Command1.Width = 2500  
Picture1.Top = 1640  
Picture1.Height = 7000  
Picture1.Width = 10000  
Picture1.BorderStyle = 1  
Picture1.Left=480  
End Sub
```

График № 1 (рис. 4.16) появится одновременно с загрузкой формы *Form3*. На форму *Form3* выведены название графика, ноль функции, минимальное и максимальное значения и масштабная сетка. Ось ординат представляет собой изменение угла поворота входного звена за цикл движения, начиная от крайнего положения. В качестве иллюстрации результатов расчетов даны перемещение, скорость и ускорение ползуна 3 (см. рис. 4.2, 4.17–4.19). Выбор кинематических параметров зависит от конкретного задания.

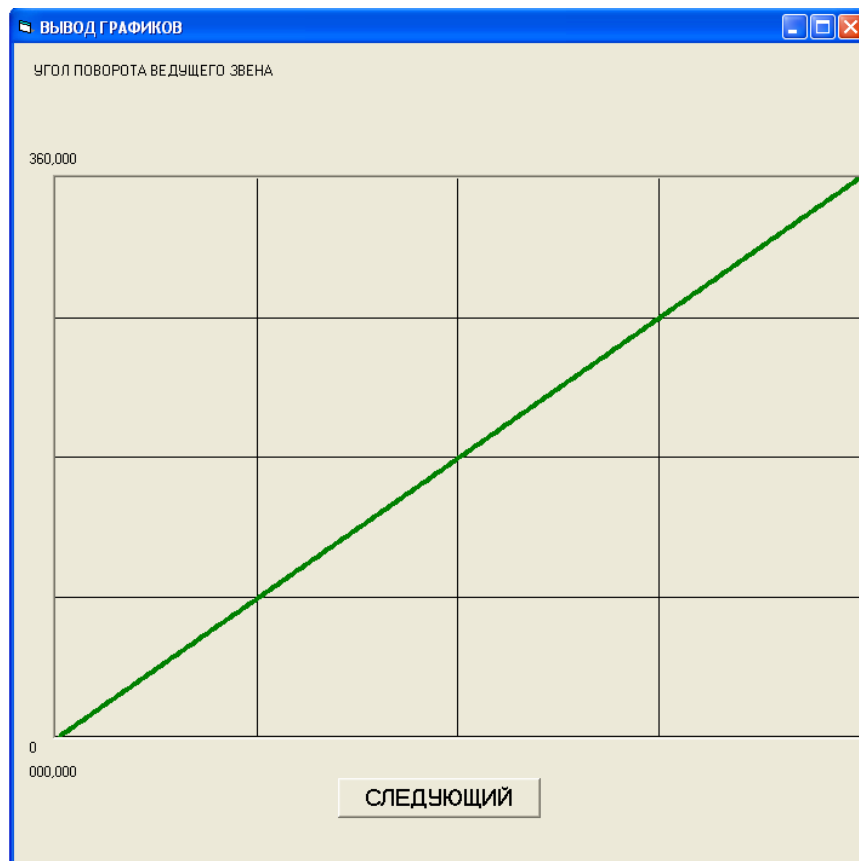


Рис. 4.16. Вывод графика № 1

При активации кнопки *СЛЕДУЮЩИЙ* происходит очистка области и вывод следующего графика.

Код процедуры следующий:

```
Private Sub Command1_Click()
Form3.Cls 'Очистка формы
ngr = ngr + 1 'Указание номера графика
If ngr = 1 Then 'Начало условия
Form3.Picture1.Cls ' Очистка объекта Picture1
Nazvgr2 = "ПЕРЕМЕЩЕНИЕ ПОЛЗУНА" 'Название графика
Form3.Label1.Caption = Nazvgr2 'Запись названия графика в Label1
Call graf(fungr5(), Form3, 1, Nazvgr2) 'Подпрограмма построения графика
End If 'Конец условия
If ngr = 2 Then
Form3.Picture1.Cls
Nazvgr3 = "СКОРОСТЬ ПОЛЗУНА"
Form3.Label1.Caption = Nazvgr3
Call graf(fungr6(), Form3, 2, Nazvgr3)
End If
If ngr = 3 Then
Form3.Picture1.Cls
```



```

Nazvgr4 = "УСКОРЕНИЕ ПОЛЗУНА"
Form3.Label1.Caption = Nazvgr4
Call graf(fungr7(), Form3, 3, Nazvgr4)
End If
If ngr = 4 Then
Form3.Picture1.Cls
Nazvgr5 = "КИНЕМАТИКА ПОЛЗУНА"
Form3.Label1.Caption = Nazvgr5
Call graf3(fungr5(), fungr6(), fungr7(), Form3, 4, Nazvgr5)
End If
End Sub

```

Для указанного кода объявляются следующие локальные переменные в разделе *Declaration Form3*:

```

Option Explicit
Rem Переменные названий графиков
Dim ngr As Double, Nazvgr2 As String, Nazvgr3 As String
Dim Nazvgr4 As String, Nazvgr5 As String

```

В рассматриваемом случае при первом нажатии командной кнопки происходит очистка области и вывод графика перемещения ползуна 3 (рис. 4.17).

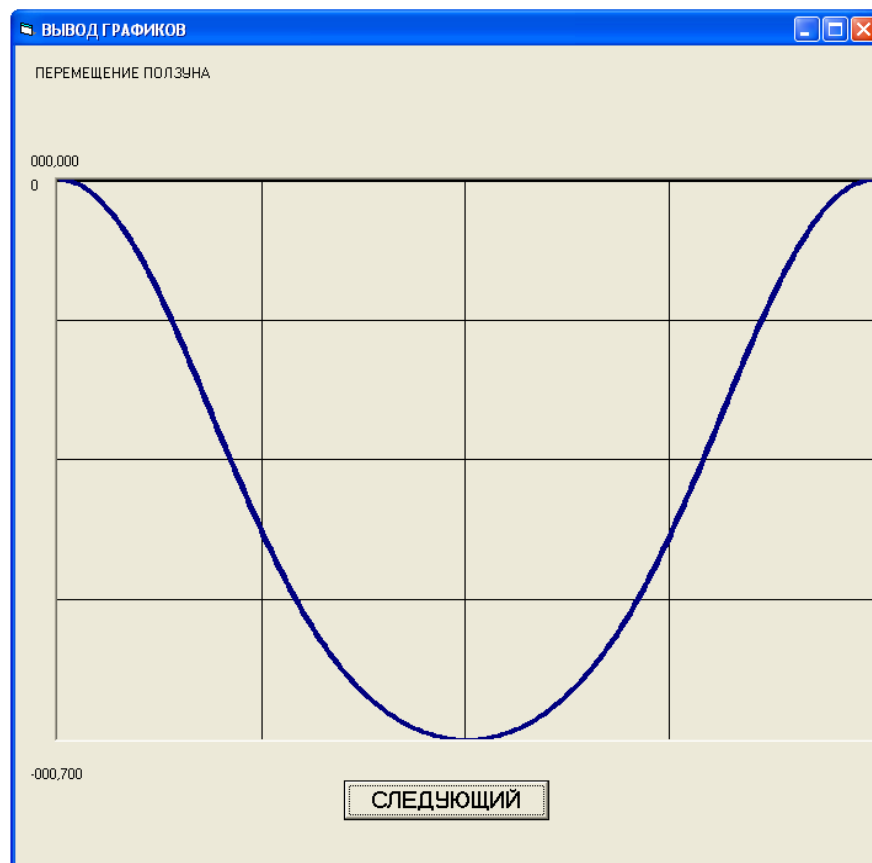


Рис. 4.17. Вывод графика № 2

Отрицательные значения перемещения ползуна $З$ (см. также рис. 4.2) связаны с тем, что при начале движения из крайнего правого положения выходное звено движется против направления оси ординат, а по завершении цикла возвращается в исходную точку.

При второй активации командной кнопки *СЛЕДУЮЩИЙ* произойдет построение графика скорости ползуна $З$ (см. рис. 4.18) с изменением цвета графика. График скорости ползуна имеет экстремум на участке движения справа налево и слева направо, а также пересекает ось ординат приблизительно в середине движения, что соответствует моменту остановки в крайнем положении и смене направления движения. При третьей активации командной кнопки *СЛЕДУЮЩИЙ* произойдет построение графика ускорения ползуна $З$ (рис. 4.19). Четвертая активация командной кнопки *СЛЕДУЮЩИЙ* представит графики аналогов скорости и ускорения ползуна $З$ (см. рис. 4.20).

Для получения графика зависимости кинематических параметров механизма от изменения обобщенной координаты – угла поворота входного звена – применяется подпрограмма *graf*.

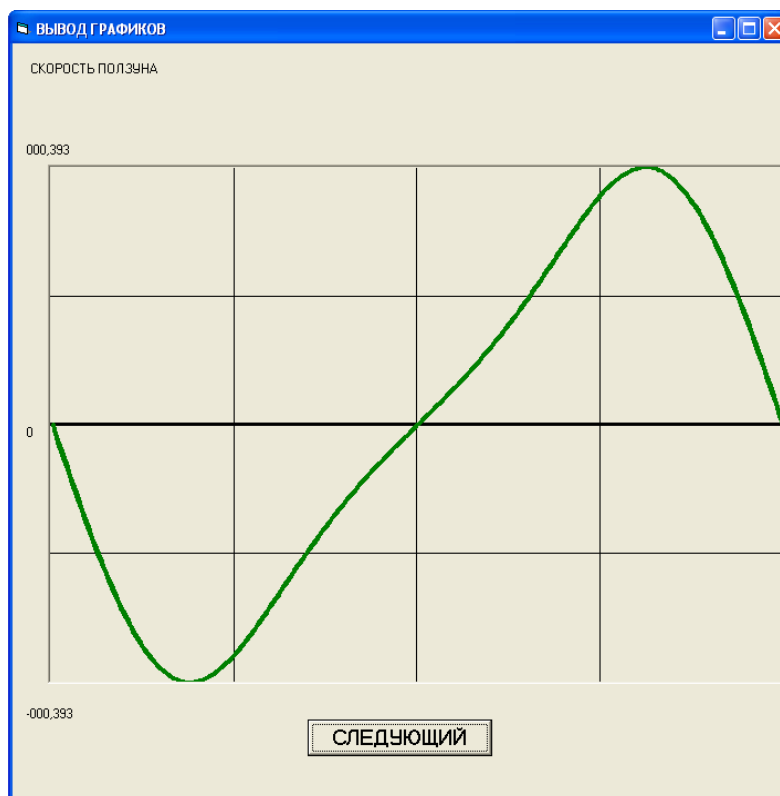


Рис. 4.18. Вывод графика № 3

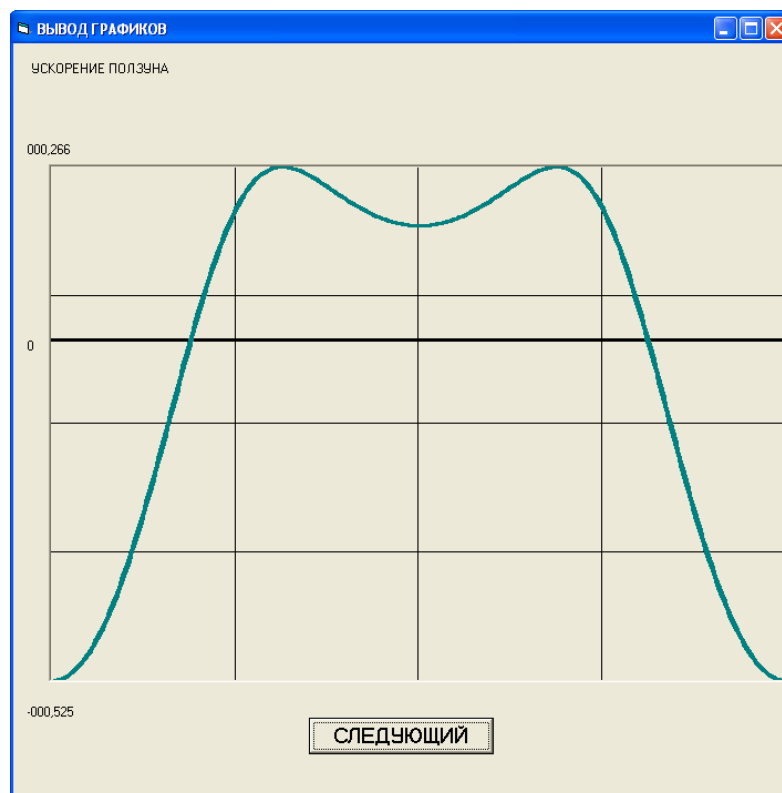


Рис. 4.19. Вывод графика № 4



Рис. 4.20. Вывод графиков № 5

Поскольку графики перемещения, скорости и ускорения ползуна 3 (рис. 4.17–4.19) лучше показать на одном листе, применяют подпрограмму *graf3*.

При этом в связи с тем, что значения перемещения и ускорения могут различаться в десятки раз, при выводе графиков на одном листе некоторые значения могут стать нечитаемыми (близки к нулю).

С выводом итогов расчетов (см. рис. 4.14) или указанных графиков (см. рис. 4.16–4.20) проект будет закончен (*RUN – END*).

Поскольку язык программирования дает пользователю широкие возможности для дизайна и оформления проекта, изложенный выше материал – лишь ориентир при выполнении самостоятельной работы, где допускаются индивидуально-авторские решения. Также имеется возможность создания приложения, которое будет работать в среде *Windows*. Для этого необходимо в меню *FILE* выбрать команду *MAKE EXE* и записать название приложения. **Для оформления отчета по работе требуется распечатка проекта и результатов расчетов.**

4.2. Пример выполнения кинетостатического расчета рычажного механизма

4.2.1. Исходные данные для расчета механизма

Рассмотрим пример кинетостатического расчета рычажного механизма (рис. 4.21).

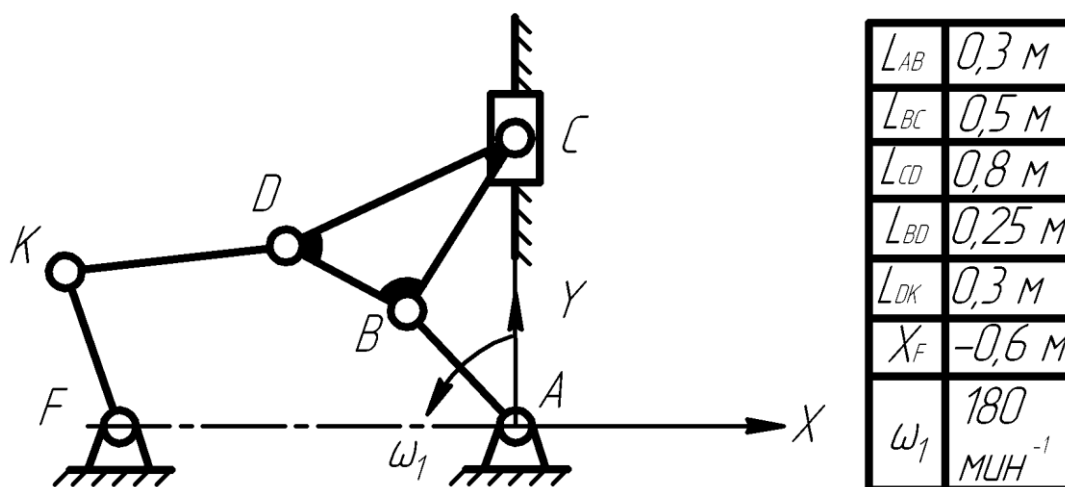


Рис. 4.21. Исходные данные для кинетостатического расчета рычажного механизма

Задача исследования – определение реакций в кинематических парах механизма и уравнивающего момента, приложенного к входному звену.

Для выполнения силового анализа исходные данные должны содержать, кроме геометрических размеров звеньев (как в подразд. 4.1), их массы, моменты инерции масс, полярные координаты точек центров масс относительно центра кинематической пары, точки приложения, направления и величины внешних сил. *(Поскольку в задании (см. подразд. 5.2) массово-инерционные характеристики звеньев не заданы, их требуется определить самостоятельно в соответствии с указаниями подразд. 5.1.)*

Обозначим звенья механизма номерами (рис. 4.22): AB – 1; BCD – 2; ползун C – 3; DK – 4; KF – 5. Укажем положения центров масс звеньев S_1, S_2, S_3, S_4, S_5 и приложим силу полезного сопротивления P_{nc} к звену 3.

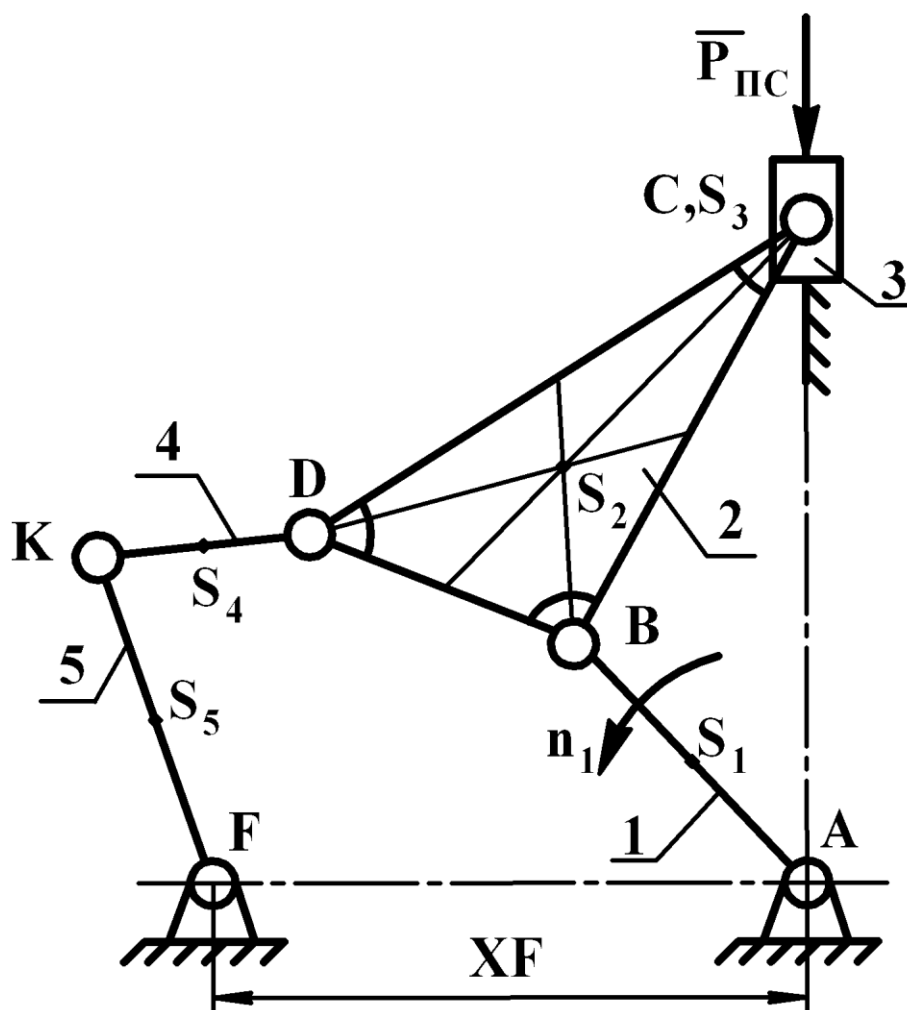


Рис. 4.22. Кинематическая схема рычажного механизма

Для проведения кинетостатического расчета необходимо найти кинематические параметры звеньев механизма и их центров масс. Поэтому силовому предшествует кинематический расчет (см. подразд. 4.1).

4.2.2. Структурный анализ механизма

Ограничимся записью структурной формулы механизма:

$$I(0,1) \rightarrow II(2,3) \rightarrow II(4,5).$$

$$J = 1; \quad J = 2.$$

В соответствии со структурой механизма составим расчетную схему кинетостатического анализа (рис. 4.23).

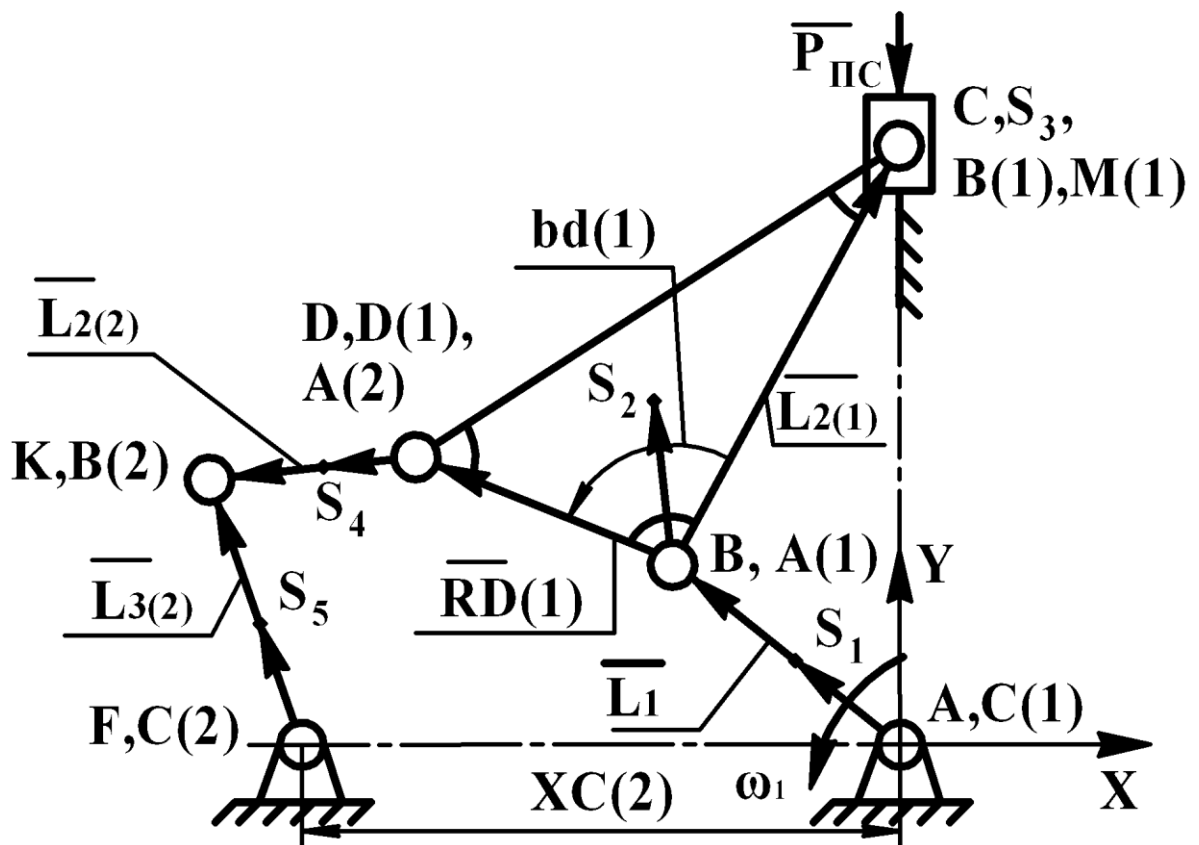


Рис. 4.23. Расчетная схема кинетостатического анализа

4.2.3. Блок-схема программы кинетостатического расчета механизма

После проведения структурного анализа механизма можем составить блок-схему программы его кинетостатического расчета (рис. 4.24).

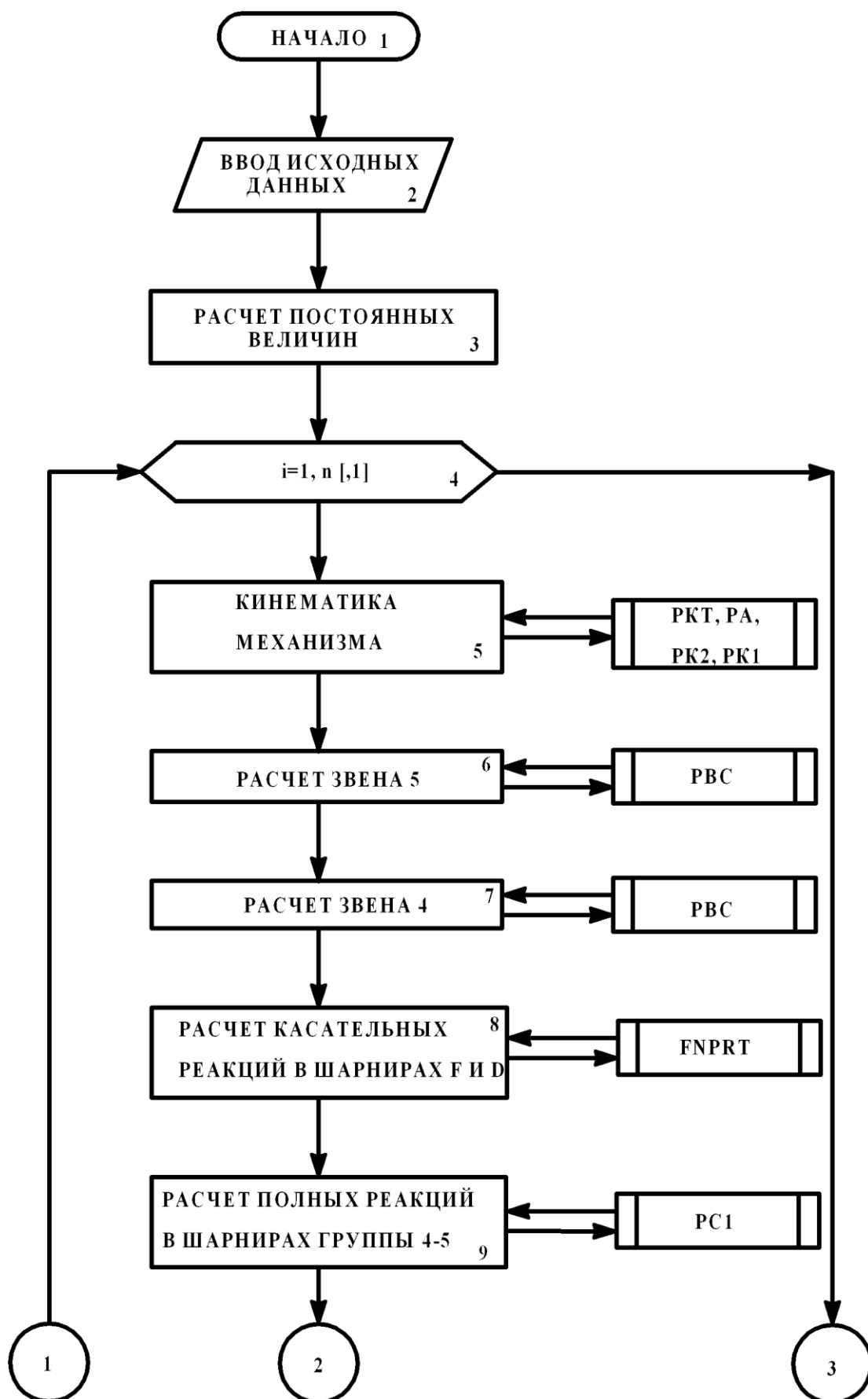


Рис. 4.24. Блок-схема программы кинетостатического расчета
(см. также с. 156)

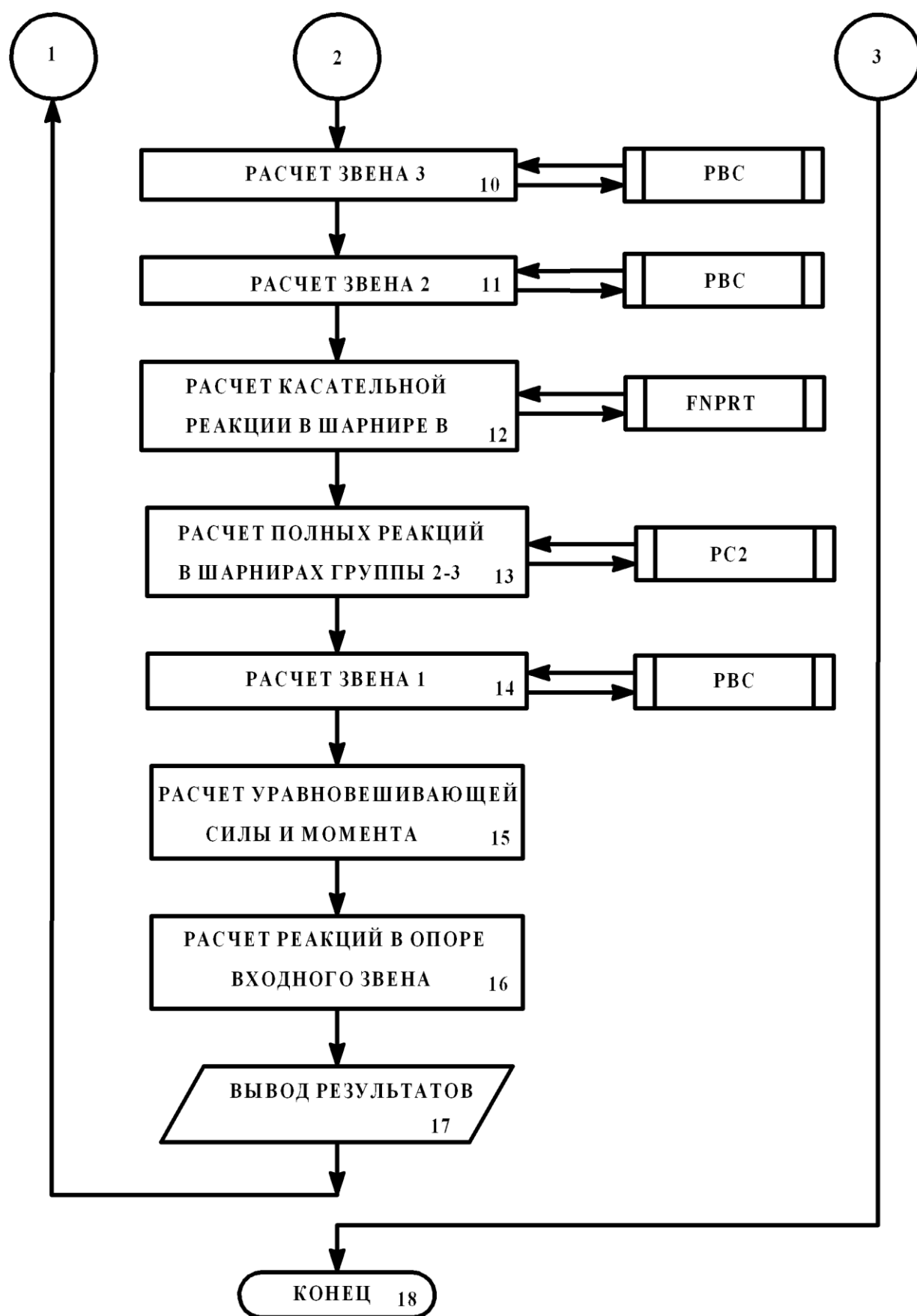


Рис. 4.24. Окончание

Кинематический расчет механизма производится в прямом порядке, т. е. начинается с первичного механизма, а кинетостатический – в обратном порядке, т. е. начинается с наиболее удаленной от входного звена группы.

Алгоритм кинематического анализа (блок 5) (см. рис. 4.24) покажем укрупненно (подробнее см. подразд. 4.1.3). Здесь дополнительно требуется расчет кинематических параметров точек центров масс звеньев. Алгоритм кинетостатического расчета разделен по структурным группам.

Кинетостатический расчет группы Ассура II класса 1 вида II (4, 5) проводится с помощью подпрограммы *PC1*. Необходимо найти ее входные параметры. С этой целью для каждого звена группы (5 (блок 6) и 4 (блок 7)) вызывают подпрограмму *PBC*, которая рассчитывает главный вектор сил, его координирующий угол и главный момент сил. Затем для группы Ассура II класса 1 вида II (4, 5) вызывают подпрограмму-функцию *FNPRT* (блок 8) для расчета тангенциальных составляющих реакций во вращательных внешних кинематических парах группы (см. рис. 4.24).

При рассмотрении группы Ассура II класса 2 вида II (2, 3) применяется подпрограмма *PC2*. Для ее корректной работы необходимо определить главные векторы сил, их координирующие углы и главные моменты сил, приложенные к звеньям 2 и 3. Это осуществляется с помощью подпрограмм *PBC* (блоки 12 и 13) (см. рис. 4.24). При этом на звено 3 действует внешняя сила $P_{ПС}$, а на звено 2 – реакция в шарнире $A(2)$, направленная в противоположенную сторону.

4.2.4. Создание проекта

Проект состоит:

- из стартовой формы *Form1* (*PR2_DATA.frm*) – для ввода исходных данных;
- формы *Form2* (*PR2_SIL_RASCH.frm*) – для расчета механизма и вывода результатов в виде таблиц;
- формы *Form3* (*PR2_GRAF.frm*) – для размещения графиков результатов расчетов.

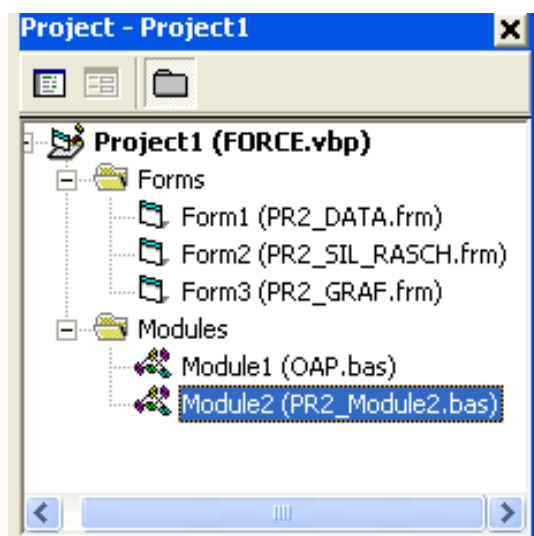


Рис. 4.25. Окно проекта

Используемые в проекте подпрограммы находятся в модуле *Modul1* (*OAP.BAS*). Для объявления глобальных переменных, задействованных в составлении программы, используется модуль *Modul2* (*PR2_Module2.BAS*) (рис. 4.25).

При выполнении блоков 1 (см. рис. 4.24) повторяются действия, приведенные в предыдущем примере,

т. е. ввод исходных данных производится в форму *Form1* (рис. 4.26).

L_{AB}	0,3 м
L_{BC}	0,5 м
L_{CD}	0,8 м
L_{BD}	0,25 м
L_{DK}	0,3 м
X_F	-0,6 м
n_1	180 мин ⁻¹

Рис. 4.26. Интерфейс формы *Form1*

На форме *Form1* размещены элементы:

- *PictureBox1* – область для отображения схемы механизма;
- *Text1*, 2, ..., 22 – ячейки для ввода исходных данных;
- *Lable1*, 2, ..., 23 – поля для пояснений исходных данных;
- *CommandButton1* – командная кнопка для присвоения данных, занесенных в ячейки, указанным переменным.

Расчет и вывод результатов кинетостатических параметров рассматриваемого механизма в соответствии с блок-схемой (см. рис. 4.24) проводится в форме *Form2* (рис. 4.27).

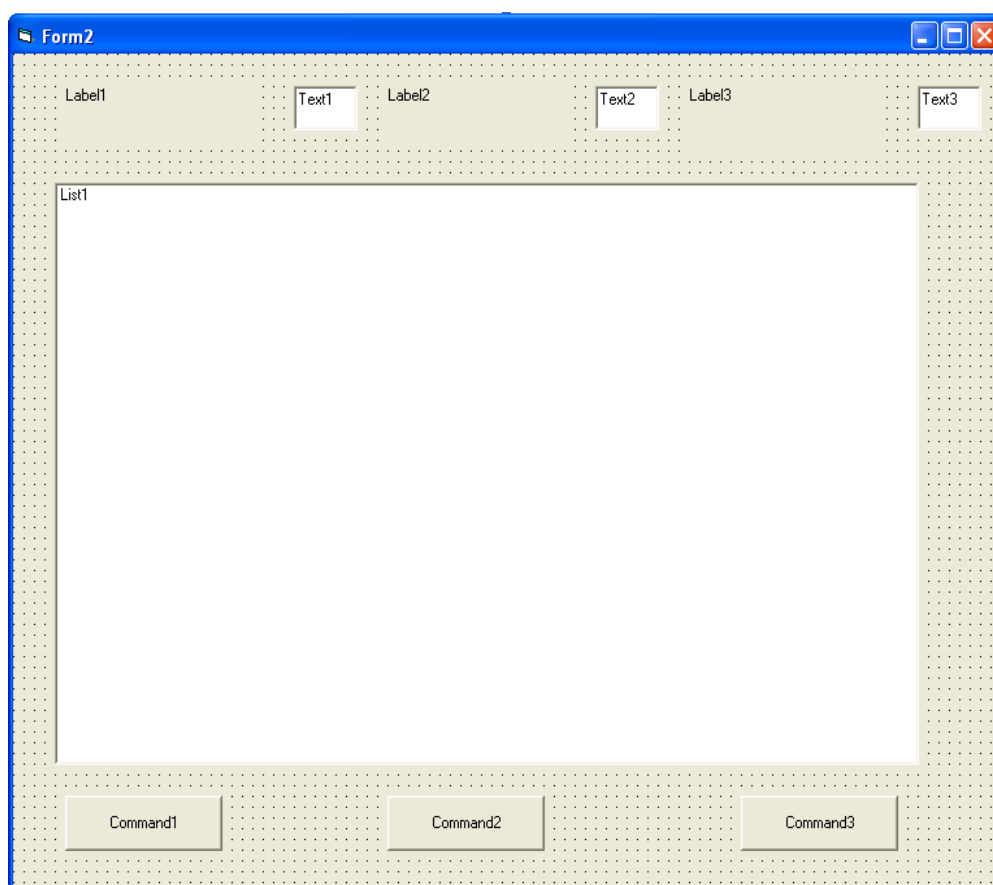


Рис. 4.27. Интерфейс формы *Form2*

На форме *Form2* размещены элементы:

- *Text1*, 2, 3 – ячейки для ввода значений внешней силы полезного сопротивления $P_{ПС}$, шага расчета и шага вывода результатов соответственно;
- *Lable1*, 2, 3 – метки для пояснений содержания элементов *Text1*, 2, 3;
- *List1* – область для вывода таблиц результатов расчетов;

- *CommandButton1* – командная кнопка для выполнения процедуры расчета и вывода таблицы результатов кинематического анализа;
- *CommandButton2* – командная кнопка для выполнения процедуры вывода таблицы результатов кинетостатического анализа;
- *CommandButton3* – командная кнопка для загрузки следующей формы, предназначенной для вывода результатов расчетов в виде графиков.

Для иллюстрации в виде графиков зависимостей различных параметров механизма от угла поворота вала входного звена предусмотрена форма *Form3* (рис. 4.28).

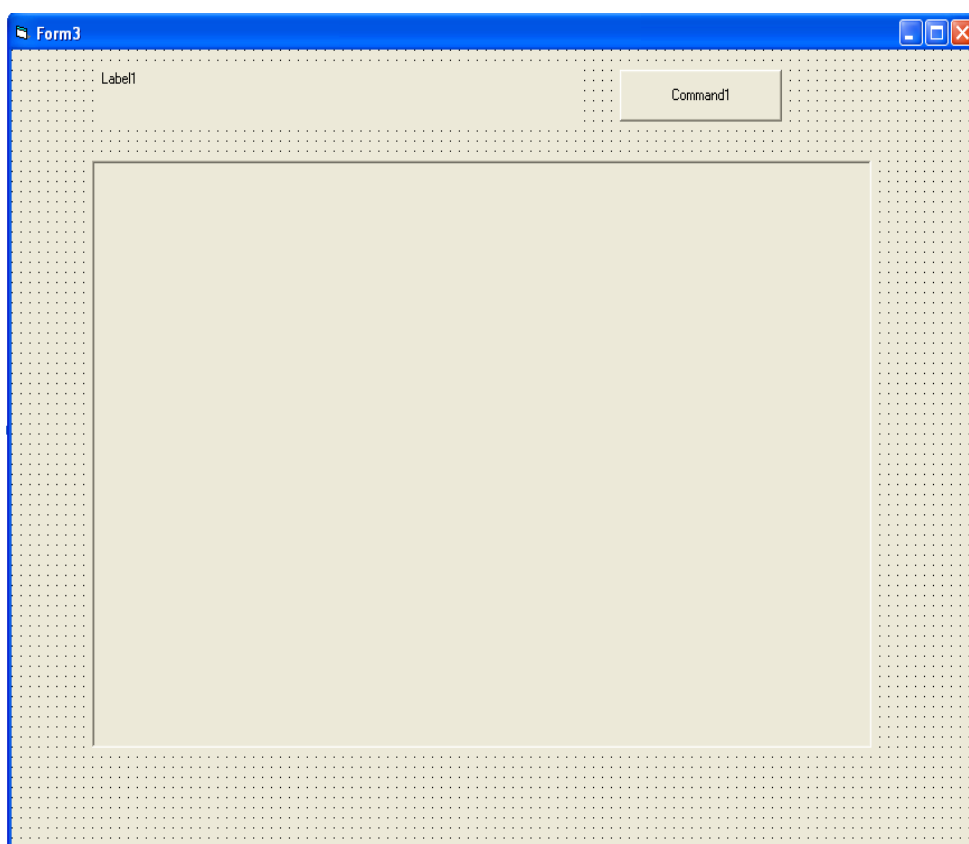


Рис. 4.28. Интерфейс формы *Form3*

На форме *Form3* размещены элементы:

- *Label1* – метка для названия показанного графика;
- *PictureBox1* – область для самих графиков;
- *CommandButton1* – командная кнопка для смены графиков.

При запуске проекта (*RUN – START / F5*) появится стартовая форма (рис. 4.29), в поля которой необходимо занести значения параметров (*жела-*

тельно сохранить эти числовые значения, чтобы не заносить их при каждом из многочисленных запусков программы):

- AB, BC, BD, CD, DK, KF – длины соответствующих звеньев, м;
- XF – координата опоры F относительно начала координат, м;
- W – угловая скорость вращения кривошипа, мин^{-1} ;
- $AS1, BS2, DS4, FS5$ – радиус-векторы от соответствующего шарнира звена до его центра масс, м;
- $MS1, MS2, MS3, MS4, MS5$ – массы соответствующих звеньев, кг;
- $JS1, JS2, JS4, JS5$ – моменты инерции масс соответствующих звеньев относительно оси, проходящей через центр масс, $\text{кг}\cdot\text{м}^2$;
- $BBS2$ – угол радиуса-вектора центра масс звена 2 (точка $S2$) относительно вектора ($L2(1)$) звена 2, град.

КИНЕМАТИЧЕСКАЯ СХЕМА МЕХАНИЗМА

AB, м

BC, м

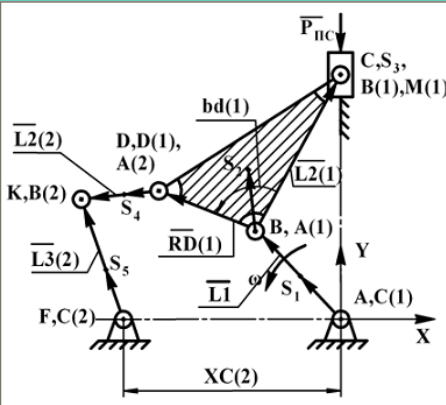
BD, м

CD, м

DK, м

KF, м

XF, м



$l_{AB}, \text{мм}$	$l_{BC}, \text{мм}$
300	600
$l_{DK}, \text{мм}$	$l_{KF}, \text{мм}$
300	400
$l_{CD}, \text{мм}$	$l_{BD}, \text{мм}$
600	250
$l_{AF}, \text{мм}$	$n, \text{об/мин}$
600	210

ИСХОДНЫЕ ДАННЫЕ

AS1, м

BS2, м

DS4, м

JS5, $\text{кг}\cdot\text{м}^2$

JS1, $\text{кг}\cdot\text{м}^2$

FS5, м

MS1, кг

MS2, кг

BBS2, град

JS2, $\text{кг}\cdot\text{м}^2$

MS3, кг

MS4, кг

MS5, кг

W1, 1/мин

JS4, $\text{кг}\cdot\text{м}^2$

Рис. 4.29. Пуск приложения

Для этого разработан код события *Form_Load*, где задаются параметры формы и размещенных на ней элементов.

Код процедуры *Form_Load* следующий:

```
Private Sub Form_Load()  
Form1.Top = 1300:Form1.Left = 700:Form1.BackColor = QBColor(11)  
Form1.Caption = "ЗАДАНИЕ"  
Label1.Caption = "КИНЕМАТИЧЕСКАЯ СХЕМА МЕХАНИЗМА"  
Label1.FontSize = 16 : Label1.Alignment = 2 : Label2.FontSize = 16  
Label2.Caption = "AB, м" : Label3.FontSize = 16 : Label3.Caption = "BC, м"  
Label4.FontSize = 16 : Label4.Caption = "BD, м" : Label5.FontSize = 16  
Label5.Caption = "CD, м" : Label6.FontSize = 16 : Label6.Caption = "DK, м"  
Label7.FontSize = 16 : Label7.Caption = "KF, м" : Label8.FontSize = 16  
Label8.Caption = "XF, м" : Label9.FontSize = 16 : Label9.Caption = "AS1, м"  
Label10.FontSize = 16 : Label10.Caption = "BS2, м" : Label11.FontSize = 16  
Label11.Caption = "DS4, м" : Label12.FontSize = 16 : Label12.Caption = "FS5, м"  
Label13.FontSize = 16 : Label13.Caption = "MS1, кг" : Label14.FontSize = 16  
Label14.Caption = "MS2, кг" : Label15.FontSize = 16  
Label15.Caption = "MS3, кг" : Label16.FontSize = 16  
Label16.Caption = "MS4, кг":Label17.FontSize =16:Label17.Caption = "MS5, кг"  
Label18.FontSize = 16 : Label18.Caption = "JS1, кгм^2" : Label19.FontSize = 16  
Label19.Caption = "JS2, кгм^2" : Label20.FontSize = 16  
Label20.Caption = "JS4, кгм^2" : Label21.FontSize = 16  
Label21.Caption = "JS5, кгм^2" : Label22.FontSize = 16  
Label22.Caption = "BBS2, град" : Label23.FontSize = 16  
Label23.Caption = "W1, 1/мин" : Command1.FontSize = 16  
Command1.Caption = "ИСХОДНЫЕ ДАННЫЕ"  
Text1.FontSize = 16:Text1.Text = "0.3":Text2.FontSize = 16 : Text2.Text = "0.6"  
Text3.FontSize = 16:Text3.Text = "0.25":Text4.FontSize = 16 : Text4.Text = "0.8"  
Text5.FontSize = 16:Text5.Text = "0.3" : Text6.FontSize = 16 : Text6.Text = "0.4"  
Text7.FontSize = 16:Text7.Text = "-0.6":Text8.FontSize = 16:Text8.Text = "0.15"  
Text9.FontSize=16:Text9.Text = "0.4":Text10.FontSize = 16:Text10.Text = "0.15"  
Text11.FontSize=16:Text11.Text="0.3":Text12.FontSize=16:Text12.Text = "0.45"  
Text13.FontSize =16:Text13.Text ="23.25":Text14.FontSize=16:Text14.Text="5"  
Text15.FontSize=16:Text15.Tex = " 0.45":Text16.FontSize=16:Text16.Text="0.9"  
Text17.FontSize=16:Text17.Text="0.0033":Text18.FontSize=16  
Text18.Text="4.07":Text19.FontSize=16:Text19.Text="0.0375"  
Text20.FontSize=16:Text20.Text="0.027":Text21.FontSize=16  
Text21.Text="25" : Text22.FontSize = 16 : Text22.Text = "180"  
End Sub
```

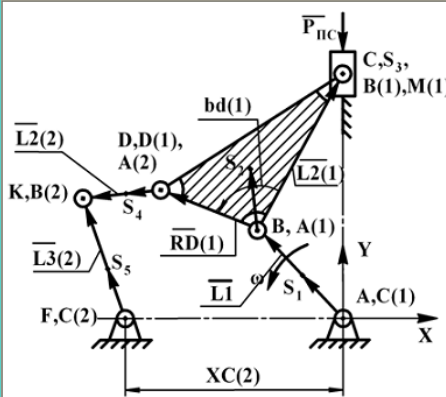
Итог выполнения процедуры ввода исходных данных *Form_Load* показан на рис. 4.30. При нажатии командной кнопки *ИСХОДНЫЕ ДАННЫЕ* происходит подтверждение введенных значений, скрытие текущей формы и загрузка следующей. Код данного события следующий:

```

Private Sub Command1_Click()
'Присвоение переменным значений из соответствующих ячеек
AB = Val(Text1): BC = Val(Text2) : LBD = Val(Text3)
CD = Val(Text4) : DK = Val(Text5) : KF = Val(Text6)
XF = Val(Text7) : W1 = Val(Text8) : AS1 = Val(Text9)
BS2 = Val(Text10) : DS4 = Val(Text11) : FS5 = Val(Text12)
BBS2 = Val(Text13) : MS1 = Val(Text14) : MS2 = Val(Text15)
MS3 = Val(Text16) : MS4 = Val(Text17) : MS5 = Val(Text18)
JS1 = Val(Text19) : JS2 = Val(Text20) : JS4 = Val(Text21)
JS5 = Val(Text22)
'Скрыть текущую форму
Me.Hide
'Загрузить указанную форму
Form2.Show
End Sub

```

КИНЕМАТИЧЕСКАЯ СХЕМА МЕХАНИЗМА



$l_{AB}, \text{ мм}$	$l_{BC}, \text{ мм}$
300	600
$l_{DK}, \text{ мм}$	$l_{KF}, \text{ мм}$
300	400
$l_{CD}, \text{ мм}$	$l_{BD}, \text{ мм}$
600	250
$l_{AF}, \text{ мм}$	$n, \text{ об/мин}$
600	210

ИСХОДНЫЕ ДАННЫЕ

AB, м	0.3	BC, м	0.6	BD, м	0.25	CD, м	0.8	DK, м	0.3	KF, м	0.4	XF, м	-0.6																
AS1, м	0.15	FS5, м	0.3	MS3, кг	5	BS2, м	0.4	MS1, кг	0.45	MS4, кг	0.45	DS4, м	0.15	MS2, кг	23.25	MS5, кг	0.9	JS5, кгм ²	0.027	BBS2, град	25	W1, 1/мин	180	JS1, кгм ²	0.0033	JS2, кгм ²	4.07	JS4, кгм ²	0.0375

Рис. 4.30. Итог выполнения процедуры ввода исходных данных

На основании кода имеющихся процедур получим список переменных, требующих объявления. Так как исходные данные могут потребоваться в любом месте приложения, сделаем их глобальными и поместим в соответствующем разделе *Module2*.

Global AB As Double, AS1 As Double, LBD As Double, BC As Double
 Global CD As Double, DK As Double, XF As Double, KF As Double
 Global W1 As Double, BS2 As Double, DS4 As Double, FS5 As Double
 Global BBS2 As Double, MS1 As Double, MS2 As Double, MS3 As Double
 Global MS4 As Double, MS5 As Double, JS1 As Double, JS2 As Double
 Global JS4 As Double, JS5 As Double

После выполнения указанной процедуры текущая форма скрывается, а на экране появляется следующая форма (*Form2*) с заголовком «Результаты расчета». Для точной прорисовки графиков (за счет получения массивов данных большего размера) зададимся минимальным шагом расчета – 1. Шаг вывода результатов в табличном виде для удобства просмотра таблицы прием крупнее – 10 (рис. 4.31).

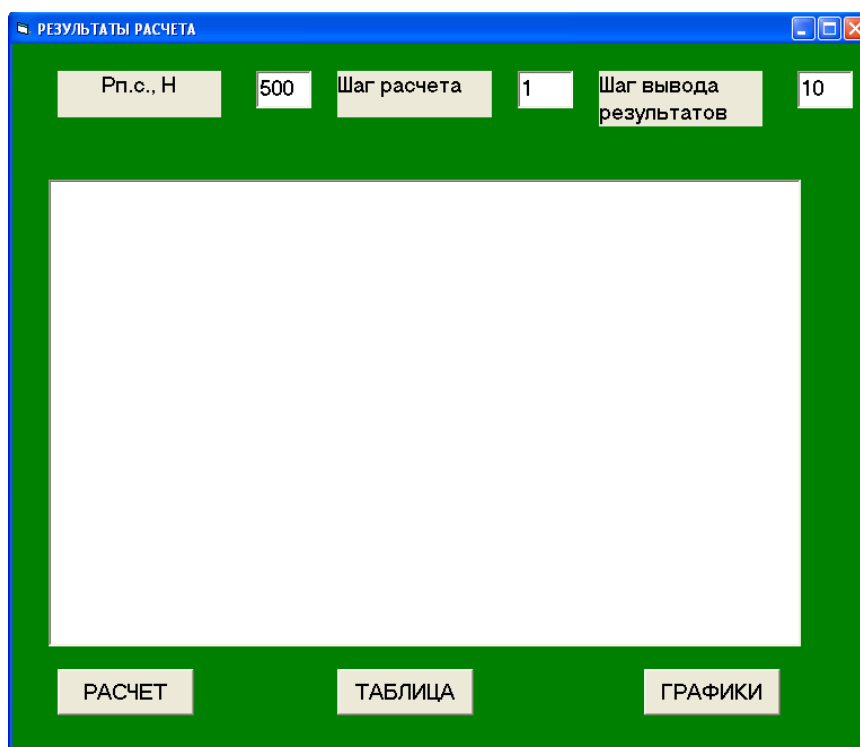


Рис. 4.31. Загрузка формы «Результаты расчета»

При загрузке формы результатов расчета выполняется следующая процедура:

```
Private Sub Form_Load()  
Form2.Caption="РЕЗУЛЬТАТЫ РАСЧЕТА" : BackColor=QBColor(2)  
Form2.Top = 1300 : Form2.Left = 700 : List1.Top = 1800 : List1.Height = 6500  
List1.Width = 10000 : List1.FontSize = 16: Label1.FontSize = 16  
Label1.Alignment = 2 : Label1.Caption = "Рп.с., Н" : Label2.FontSize = 16
```



```

Label2.Caption = "Шаг расчета" : Label3.Caption = "Шаг вывода результатов"
Label3.FontSize = 16 : Text1.FontSize = 16 : Text1.Text = 500 : Text2.Text = 1
Text2.FontSize = 16 : Text3.FontSize = 16 : Text3.Text = 10
Command1.FontSize = 16 : Command1.Caption = "РАСЧЕТ"
Command2.FontSize = 16 : Command2.Caption = "ТАБЛИЦА"
Command3.FontSize = 16 : Command3.Caption = "ГРАФИКИ"
End Sub

```

После ввода требуемых параметров и нажатия командной кнопки *РАСЧЕТ* происходит выполнение командного кода, в котором реализован алгоритм кинематического и кинетостатического анализа (блоки 3–16, см. рис. 4.24) заданного механизма с помощью рассматриваемых подпрограмм. Код подпрограммы следующий:

```

Private Sub Command1_Click()
Rem Исходные данные
j = 2 : L1 = AB: L2(1) = BC: l3(1) = 0: L2(2) = DK: l3(2) = KF: xc(2) = XF
fx1(1) = 90 : f31(1) = 90: z(2) = -1: PPC = Val(Text1): h = Val(Text2)
H1 = Val(Text3)
Rem Нулевые параметры
XC(1) = 0: YC(1) = 0 : YC(2) = 0 : r4(1) = 0: b4(1) = 0
Rem Расчет постоянных величин
PI = 4 * Atn(1): RAD = PI / 180: DEG = 1 / RAD : OM = PI * W1 / 30
BBS2 = BBS2 * RAD : fx31(1) = fx31(1) * RAD: fx1(1) = fx1(1) * RAD
UgolDBC=FNARC(CD^2-BC^2-LBD^2)/(2*BC*LBD)-PI : N = 360 / h + 1
Rem Значения угловой координаты входного звена 1 в момент начала расчета
f1n = PI / 2
If f1n > 2 * PI Then f1n = f1n - 2 * PI
Rem Переобозначение массивов
ReDim L2(j), l3(j), xa(j), ya(j), va(j), fva(j), wa(j), fwa(j), xc(j), yc(j), vc(j), fvc(j)
ReDim wc(j), fwc(j) BD(j), rd(j), z(j), hm(N + 1), fungr1(N), fungr2(N)
ReDim fungr3(N), fungr4(N), fungr5(N), fungr6(N), fungr7(N), fungr8(N)
ReDim fungr9(N), fungr10(N), fungr11(N), ReDim hm(N), fun1(N)
For i = 1 To N Step h Rem Создание цикла с шагом h
Rem Кинематический анализ механизма
Rem Кинематика звена 1
Rem Угловые параметры звена 1
f1=f1n+(i-1)*h*RAD: F11=1: f12 = 0
Rem Координаты точки A(1)
Call pkt(0, 0, L1, f1, 0, xa(1), ya(1))
Rem Аналоги скорости и ускорения точки A(1)
Call PA(0, 0, 0, 0, L1, 0, f1, F11, f12, va(1), fva(1), wa(1), fwa(1))
Rem Координаты точки S1
Call pkt(0, 0, AS1, f1, 0, XS1, YS1)

```

```

Rem Аналоги скорости и ускорения точки S1
Call PA(0, 0, 0, 0, AS1, 0, f1, F11, f12, VS1, FVS1, WS1, FWS1)
Rem Кинематика группы 2-3
j = 1 Rem Номер группы
Call pk2(Lc, FBC, FCC, L1c, F1BC, L2c, F2BC, 0, 0, 0, WS3, 0, 0, 0, FWS3)
Rem Координаты точки A(2)
Call pkt(xa(1), ya(1), LBD, FBC, UgoIDBC, xa(2), ya(2))
Rem Аналоги скорости и ускорения точки A(2)
Call PA(va(1), fva(1), wa(1), fwa(1), LBD, UgoIDBC, FBC, F1BC, F2BC, _
va(2), _fva(2), wa(2), fwa(2))
Rem Координаты точки S2
Call pkt(xa(1), ya(1), BS2, FBC, BBS2, XS2, YS2)
Rem Аналоги скорости и ускорения точки S2
Call PA(va(1), fva(1), wa(1), fwa(1), BS2, BBS2, FBC, F1BC, F2BC, VS2, _
FVS2, WS2, FWS2)
Rem Кинематика группы 4-5
j = 2 Rem Номер группы
Call pk1(FDK, FKF, F1DK, F1KF, F2DK, F2KF, 0, 0, 0, 0, 0, 0, 0, 0)
Rem Координаты точки S4
Call pkt(xa(2), ya(2), DS4, FDK, 0, XS4, YS4)
Rem Аналоги скорости и ускорения точки S4
Call PA(va(2), fva(2), wa(2), fwa(2), DS4, 0, FDK, F1DK, F2DK, VS4, _
FVS4, WS4, FWS4)
Rem Координаты точки S5
Call pkt(XF, 0, FS5, FKF, 0, XS5, YS5)
Rem Аналоги скорости и ускорения точки S5
Call PA(0, 0, 0, 0, FS5, 0, FKF, F1KF, F2KF, VS5, FVS5, WS5, FWS5)
Rem Силовой анализ механизма
Rem Силовой анализ группы 4-5
Rem Внешние силы, действующие на звено 5
qp(1) = 0: r(1) = 0: fp(1) = 0: B(1) = 0
Rem Главный вектор и момент сил звена 5
Call PBC(1, FKF, F2KF, WS5, FWS5, 0, FS5, 0, JS5, MS5, P5, FP5, M5)
Rem Внешние силы, действующие на звено 4
qp(1) = 0: r(1) = 0: fp(1) = 0: B(1) = 0
Rem Главный вектор и момент сил звена 4
Call PBC(1, FDK, F2DK, WS4, FWS4, 0, DS4, 0, JS4, MS4, P4, FP4, M4)
Rem Касательная реакция в шарнире C(2)
rct(2) = FNPRT(M5, FS5, l3(2), P5, FKF, FP5, 0)
Rem Касательная реакция в шарнире A(2)
rat(2) = FNPRT(M4, DS4, L2(2), P4, FDK, FP4, 0)
Rem Силовой анализ группы 4-5
Call PC1(rat(2), rct(2), 0, P4, P5, 0, FDK, FKF, FP4, FP5, RA2, RC2, RB2, _
FRA2, FRC2, FRB2)
Rem Внешние силы, действующие на звено 3
qp(1) = PPC: r(1) = 0: fp(1) = 2 * PI / 3: B(1) = 0

```

```

Rem Главный вектор и момент сил звена 3
Call PBC(1, FCC, 0, WS3, FWS3, 0, 0, 0, 0, MS3, p3, fp3, m3)
Rem Внешние силы, действующие на звено 2
qp(1) = RA2: r(1) = LBD: fp(1) = FRA2 + PI: B(1) = UgoIDBC
Rem Главный вектор и момент сил звена 2
Call PBC(1, FBC, F2BC, WS2, FWS2, 0, BS2, BBS2, JS2, MS2, p2, fp2, M2)
Rem Силовой анализ группы 2-3
rct(1) = 0 Rem Касательная реакция в шарнире C(1)
Rem Касательная реакция в шарнире A(1)
rat(1) = FNPRT(M4, BS2, L2(1), p2, FBC, fp2, 0)
Call PC2(rat(1), p2, p3, 0, FBC, FCC, fp2, fp3, 0, m3, RA1, RB1, RK1, _
FRA1, FRB1, lk)
Rem Силовой анализ входного звена 1
Rem Внешние силы, действующие на звено 1
qp(1) = RA1: r(1) = L1: fp(1) = FRA1 + PI: B(1) = 0
Rem Главный вектор и момент сил звена 1
Call PBC(1, f1, f12, WS1, FWS1, 0, AS1, 0, JS1, MS1, P1, FP1, M1)
Rem Расчет уравновешивающего момента
MUR1 = -M1 - P1 * AS1 * Sin(FP1 - f1): PUR1 = MUR1 / L1
Rem Расчет реакции в опоре кривошипа
ROX = -PUR1 * Cos(f1 + PI / 2) - P1 * Cos(FP1)
ROY = -PUR1 * Sin(f1 + PI / 2) - P1 * Sin(FP1)
Call PMY(ROX, ROY, RO, FRO)
If f1 > 2 * PI Then f1 = f1 - 2 * PI
Rem Подготовка массивов для вывода результатов
fungr1(i) = f1 * DEG: fungr2(i) = FBC * DEG: fungr3(i) = FDK * DEG
fungr4(i) = FKF * DEG: fungr5(i) = RC2: fungr6(i) = RK1: fungr7(i) = RO
fungr8(i) = MUR1: fungr9(i) = Lc: fungr10(i) = L1c: fungr11(i) = L2c
100 Next i
Rem Подготовка таблицы
List1.Clear Rem Очистка области List1
znak1 = "УГЛОВЫЕ КООРДИНАТЫ ЗВЕНЬЕВ"
znak2 = "      F1      F2      F3      F4"
znak3 = "      град      град      град      град"
znak4 = "===== "
List1.AddItem Format$(znak1, "")
List1.AddItem Format$(znak4, "")
List1.AddItem Format$(znak2, "")
List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak4, "")
Rem Вывод результатов кинематического анализа
For i = 1 To N Step H1 Rem Цикл для печати результатов с шагом H1
List1.AddItem Format$(fungr1(i), "      000.0") + "" + Format(fungr2(i), "
000.0") + "" + Format(fungr3(i), "      000.0") + Format(fungr4(i), "      000.0")
Next i
End Sub

```

На основании данного кода получим список переменных.

Глобальные переменные объявляются в модуле *Module2*:

```
Global fungr1() As Double, fungr2() As Double, fungr3() As Double
Global fungr4() As Double, fungr5() As Double, fungr6() As Double
Global fungr7() As Double, fungr8() As Double, fungr9() As Double
Global fungr10() As Double, fungr11() As Double, i As Integer
Global N As Integer, j As Integer, PI As Double, OM As Double
```

Локальные переменные объявляются в форме *Form2*:

```
Dim znak1 As String, znak2 As String, znak3 As String, znak4 As String
Dim znak5 As String, Nazv1 As String, L1 As Double, n As Double
Dim h As Double, H1 As Double, arg As Double, RAD As Double
Dim DEG As Double, f1 As Double, F11 As Double, f12 As Double
Dim FBC As Double, F1BC As Double, F2BC As Double, FCC As Double
Dim FDK As Double, FKF As Double, F1DK As Double, F1KF As Double
Dim F2DK As Double, F2KF As Double, L2c As Double, L1c As Double
Dim Lc As Double, fln As Double, PPC As Double, MUR1 As Double
Dim PUR1 As Double, r(1), B(1) As Double, rat(5) As Double, rct(5) As Double
Dim ra(), rb(), rk(), fra(), frb(), rc(), frc(), RRA1(), RRB1() As Double
Dim FRB2 As Double, FRC2 As Double, FRA2 As Double, RB2 As Double
Dim RA2 As Double, RC2 As Double, FRB1 As Double, FRA1 As Double
Dim RB1 As Double, RA1 As Double, RK1 As Double, UgolDBC As Double
Dim YS1 As Double, XS1 As Double, YS2 As Double, XS2 As Double
Dim YS4 As Double, XS4 As Double, YS5 As Double, XS5 As Double
Dim FWS1 As Double, WS1 As Double, FVS1 As Double, VS1 As Double
Dim FWS2 As Double, WS2 As Double, FVS2 As Double, VS2 As Double
Dim FWS3 As Double, WS3 As Double, fvs3 As Double, vs3 As Double
Dim FWS4 As Double, WS4 As Double, FVS4 As Double, VS4 As Double
Dim FWS5 As Double, WS5 As Double, FVS5 As Double, VS5 As Double
Dim BBS2 As Double, BS2 As Double, DS4 As Double, M5 As Double
Dim FP5 As Double, P5 As Double, M4 As Double, FP4 As Double
Dim P4 As Double, m3 As Double, fp3 As Double, p3 As Double, M2 As Double
Dim fp2 As Double, p2 As Double, M1 As Double, FP1 As Double, P1 As Double
Dim FRO As Double, RO As Double, ROX As Double, ROY As Double
```

В результате выполнения процедуры в пустом поле *List1* появится таблица кинематических параметров – угловых координат звеньев (рис. 4.32).

Активация командной кнопки *ТАБЛИЦА* выводит параметры кинетостатического расчета реакций в кинематических парах и уравновешивающего момента (рис. 4.33) за счет выполнения процедуры со следующим кодом:

```

Private Sub Command2_Click()
List1.Clear Rem Очистка области List1
Rem ПОДГОТОВКА ТАБЛИЦЫ
Rem Название таблицы
znak1 = "РЕАКЦИИ В КИНЕМАТИЧЕСКИХ ПАРАХ "
znak2 = "И УРАВНОВЕШИВАЮЩИЙ МОМЕНТ"
Rem Шапка таблицы
znak3 = "    F1      RA1      RA2      RO      MUR"
znak4 = "    град    Н      Н      Н      Нм"
znak5 = "=====
List1.AddItem Format$(znak1, ""):List1.AddItem Format$(znak2, "")
List1.AddItem Format$(znak5, ""):List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak4, ""):List1.AddItem Format$(znak5, "")
Rem ВЫВОД РЕЗУЛЬТАТОВ КИНЕТОСТАТИЧЕСКОГО РАСЧЕТА
For i = 1 To N Step H1 Rem Цикл для печати результатов с шагом H1
List1.AddItem Format$(fungr1(i), " 000.0") + "      " _
+ Format(fungr5(i) "000000.0") + "      " _
+ Format(fungr6(i), "000000.0") + "      " _
+ Format(fungr7(i), "000000.0") + "      " _
+ Format(fungr8(i), "00.000")
Next i
End Sub

```

РЕЗУЛЬТАТЫ РАСЧЕТА

Рп.с., Н: 500 Шаг расчета: 1 Шаг вывода результатов: 10

УГЛОВЫЕ КООРДИНАТЫ ЗВЕНЬЕВ

F1 град	F2 град	F3 град	F4 град
090,0	090,0	171,5	081,6
100,0	085,0	173,9	088,5
110,0	080,2	175,4	094,9
120,0	075,5	175,7	100,5
130,0	071,3	174,6	105,5
140,0	067,5	171,7	109,4
150,0	064,3	166,8	112,0
160,0	062,0	159,3	112,7
170,0	060,5	148,2	110,2
180,0	060,0	131,8	102,4
190,0	060,5	107,6	086,2
200,0	062,0	077,1	061,1

РАСЧЕТ ТАБЛИЦА ГРАФИКИ

Рис. 4.32. Результаты расчета угловых координат звеньев

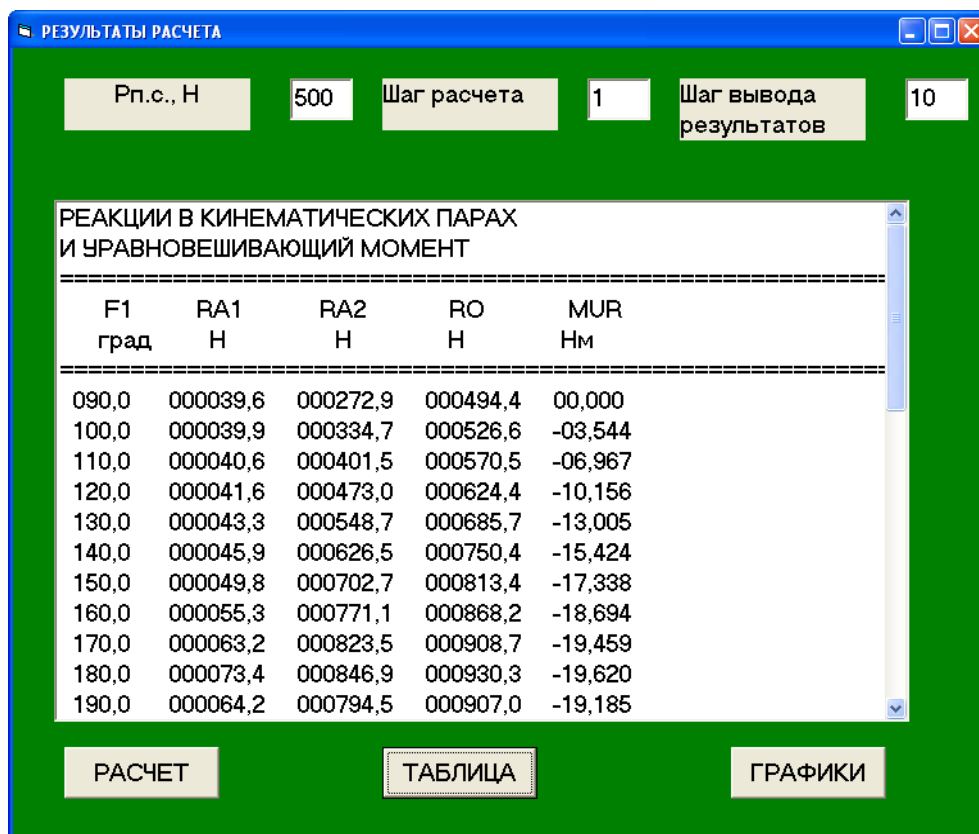


Рис. 4.33. Результаты расчета реакций в кинематических парах и уравновешивающего момента

Нажав командную кнопку *ГРАФИКИ*, можно просмотреть результаты в виде ряда графиков (№ 1–5), выведенных в форме *Form3* (рис. 4.34–4.38).

Код процедуры:

```
Private Sub Command3_Click()
Me.Hide Rem Выгрузка текущей формы
Form3.Show Rem Загрузка следующей формы
Nazv1 = "КИНЕМАТИКА ПОЛЗУНА" Rem Название графика
Form3.Label1.Caption = Nazv1 Rem Запись названия графика в метке1
Call graf3(fungr9(), fungr10(), fungr11(), Form3, 2, Nazv1) 'Построение графика
End Sub
```

Смена графиков происходит после каждого нажатия кнопки *СЛЕДУЮЩИЙ*. Исполнительный код данной процедуры:

```
Private Sub Command1_Click()
Rem счетчик номера графика
ngr = ngr + 1
Rem Условия вывода графика №2
If ngr = 1 Then
```

```

Form3.Cls Rem Очистка формы
Form3.Picture1.Cls Rem Очистка области Picture1
Nazv2 = "РЕАКЦИИ В ШАРНИРЕ F, Н"
Rem Вывод названия графика в поле Label1
Form3.Label1.Caption = Nazv2
Rem Построение графика с помощью подпрограммы
Call graf(fungr5(), Form3, 3, Nazv2)
End If Rem Завершение условия вывода графика №2
Rem Условия вывода графика №3
If ngr = 2 Then
Form3.Cls
Form3.Picture1.Cls
Nazv3 = "РЕАКЦИИ В ШАРНИРЕ C, Н"
Form3.Label1.Caption = Nazv3
Call graf(fungr6(), Form3, 4, Nazv3)
End If
Rem Условия вывода графика №4
If ngr = 3 Then
Form3.Cls
Form3.Picture1.Cls
Nazv4 = "РЕАКЦИИ В ШАРНИРЕ O, Н"
Form3.Label1.Caption = Nazv4
Call graf(fungr7(), Form3, 5, Nazv4)
End If
Rem Условия вывода графика №5
If ngr = 4 Then
Form3.Cls
Form3.Picture1.Cls
Nazv5 = "УРАВНОВЕШИВАЮЩИЙ МОМЕНТ, Нм"
Form3.Label1.Caption = Nazv5
Call graf(fungr8(), Form3, 6, Nazv5)
End If
End Sub

```

В коде имеются только локальные переменные, которые помещаются в форме *Form3*:

```

Dim ngr As Double, Nazv1 As String, Nazv2 As String
Dim Nazv3 As String, Nazv4 As String, Nazv5 As String

```

Организация ввода-вывода параметров механизма может быть изменена в соответствии с конкретным заданием, целями проекта и эстетическими представлениями автора проекта. Желательно создать приложение в виде исполняемого файла, как это показано в предыдущем примере.

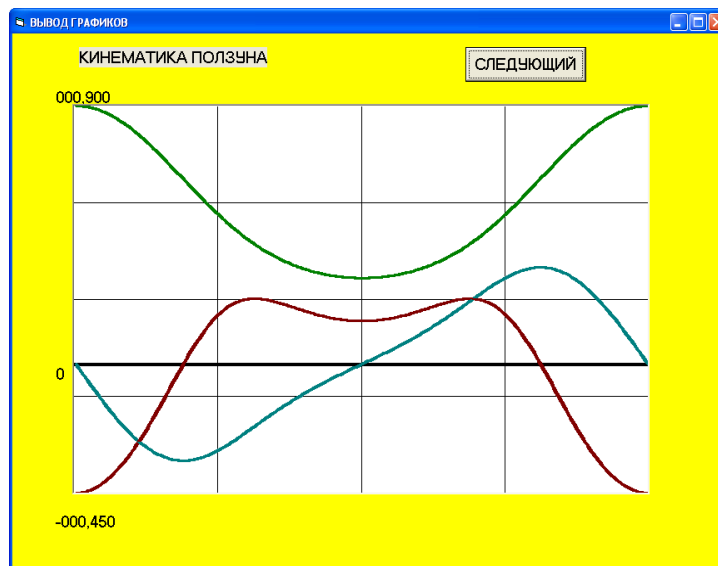


Рис. 4.34. Вывод графика № 1

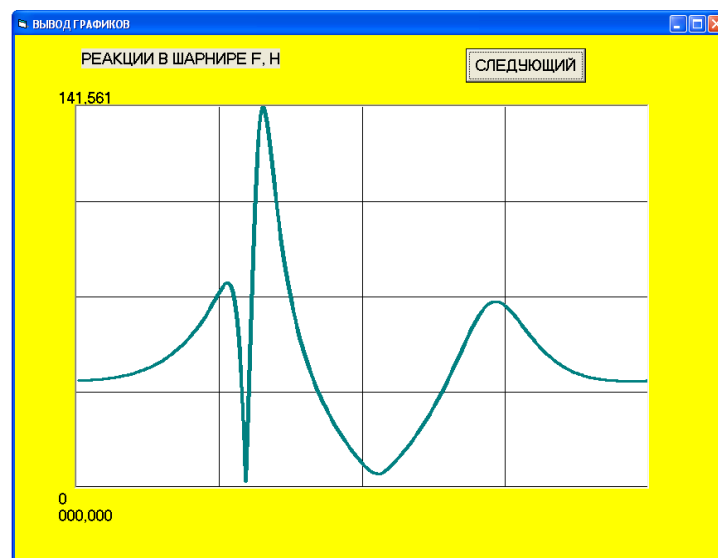


Рис. 4.35. Вывод графика № 2

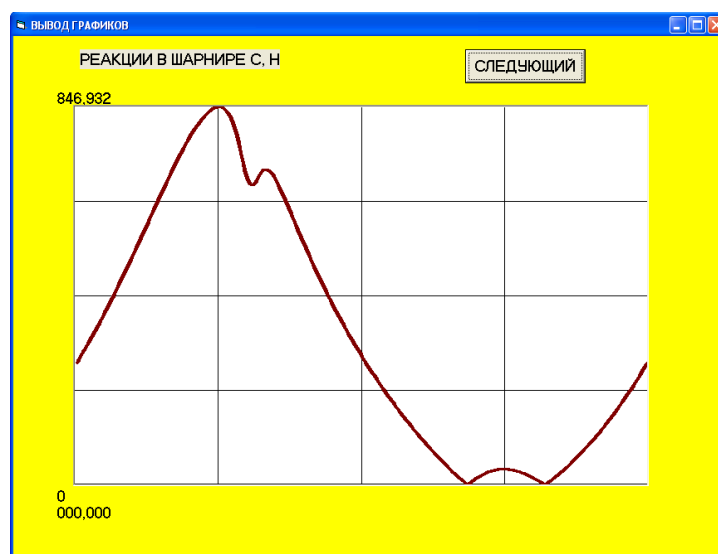


Рис. 4.36. Вывод графика № 3

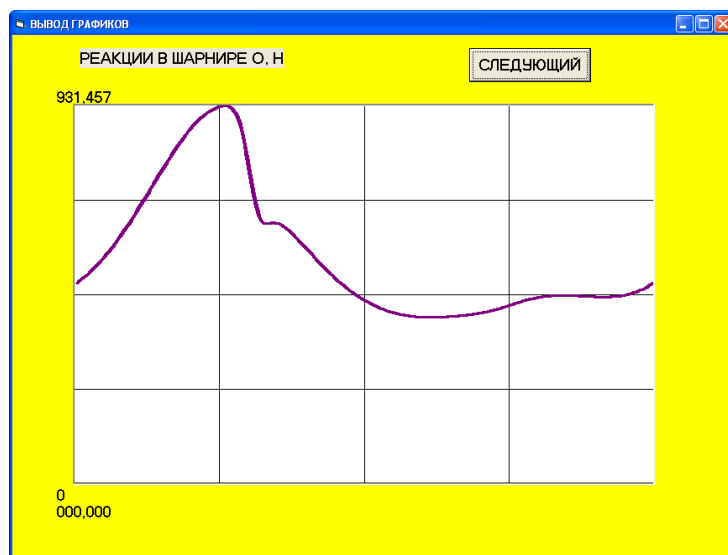


Рис. 4.37. Вывод графика № 4

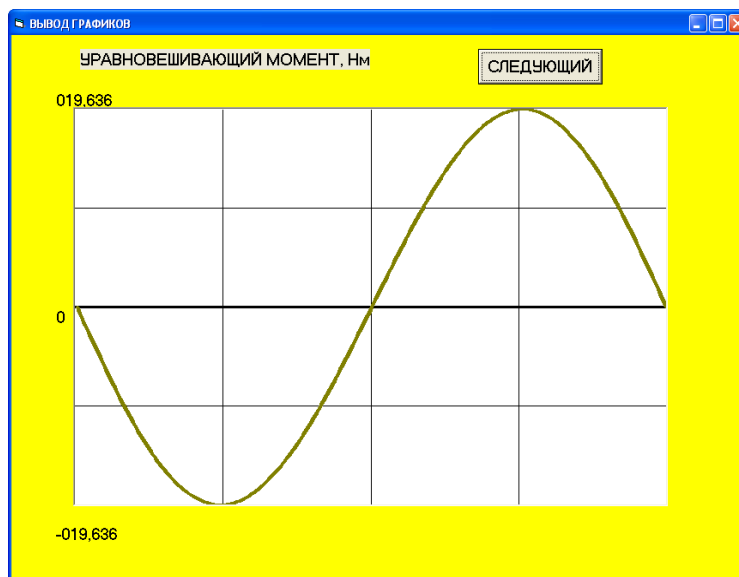


Рис. 4.38. Вывод графика № 5

На этом проект завершен (*RUN – END*).

4.3. Пример расчета механизма с кулачковым приводом

4.3.1. Исходные данные для расчета механизма

Рассмотрим пример разработки проекта кинетостатического анализа плоского кулачкового-рычажного механизма II класса (рис. 4.39). Для выполнения силового анализа должны быть известны следующие исходные данные: размеры звеньев; массы звеньев; моменты инерции масс звеньев; поляр-

ные координаты точек центров масс относительно центра кинематической пары; направления и величины внешних сил; полярные координаты точек их приложения (табл. 4.1).

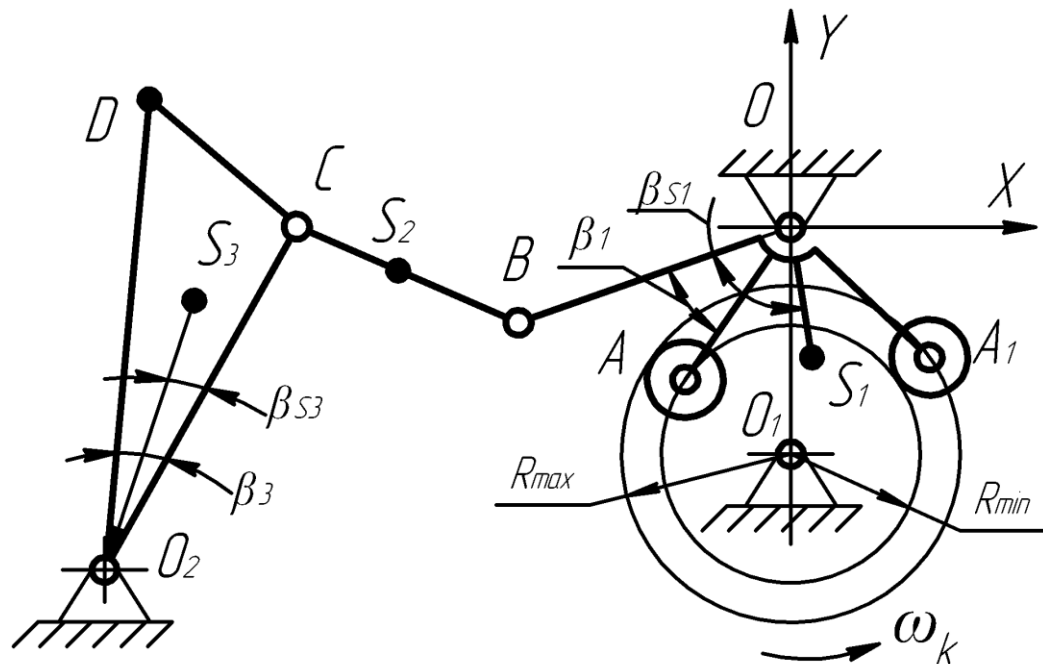


Рис. 4.39. Кинематическая схема
плоского кулачкового-рычажного механизма II класса

Таблица 4.1

Исходные данные

X_{O_1} , мм	Y_{O_1} , мм	X_{O_2} , мм	Y_{O_2} , мм	R_{\min} , мм	Ψ_{\max}
0	175	-435	-440	120	$\pi/6$
O_1A , мм	O_1B , мм	BC , мм	O_2C , мм	O_2D , мм	O_1S_1 , мм
90	220	180	630	770	110
BS_2 , мм	O_2S_3 , мм	β_1 , град	β_{S_1} , град	β_3 , град	β_{S_3} , град
110	900	22,5	22,5	8,1	10
$\varphi_{уд}$, град	$\varphi_{см.д}$, град	$\varphi_{сб}$, град	$\varphi_{см.б}$, град	m_1 , кг	m_2 , кг
60	0	60	240	15,3	1,08
m_3 , кг	J_{S_1} , кг·м ²	J_{S_2} , кг·м ²	J_{S_3} , кг·м ²	P , Н	ω_k , мин ⁻¹
31,5	0,146	0,005	8,44	500	100

Для проведения кинетостатического расчета необходимо найти кинематические параметры звеньев механизма и их центров масс. Поэтому силовому анализу предшествует кинематический анализ. Методики проведения данных расчетов показаны в подразд. 4.1 и 4.2.

4.3.2. Структурный анализ механизма

Кулачок – это тело переменного радиуса кривизны относительно оси вращения, профиль которого выполнен по определенному закону. По заданному закону получим перемещение, скорость и ускорение толкателя в зависимости от угла поворота кулачка. При проектировании профиля кулачка по заданному закону движения используются подпрограммы отдельного модуля *KULAK.BAS* (см. разд. 2). Обозначим звенья механизма номерами: O_1B – 1; BC – 2; O_2CD – 3 (рис. 4.40). Таким образом, входным звеном механизма становится толкатель O_1B и кинематическая схема преобразовывается.

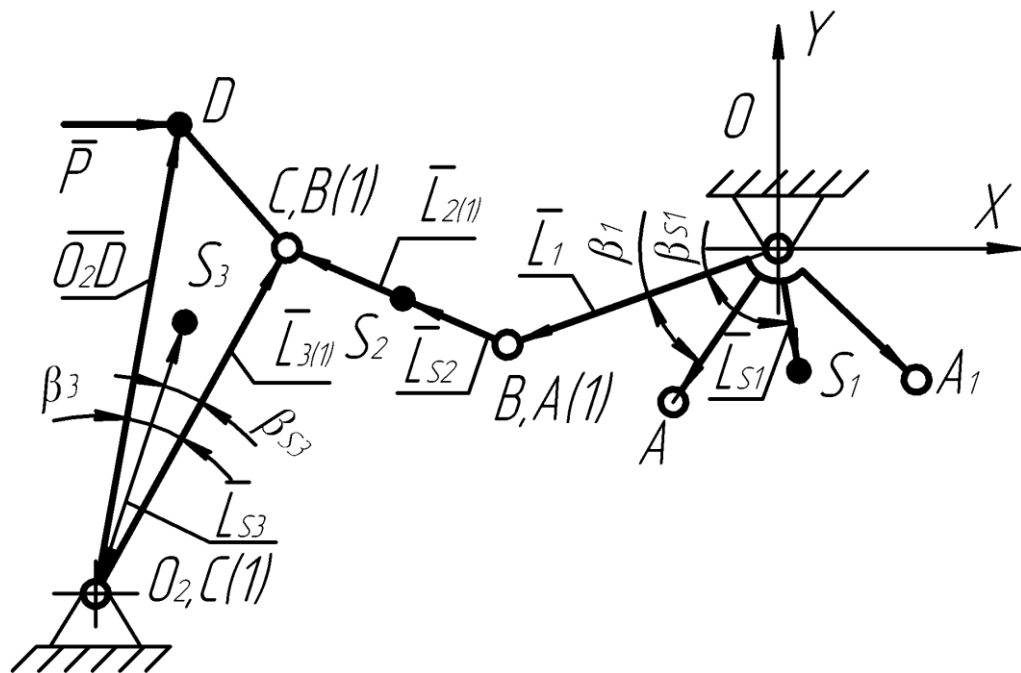


Рис. 4.40. Расчетная схема механизма

Ограничимся записью формулы строения механизма

$$I(0,1) \rightarrow II(2,3);$$

$$J = 1.$$

Структурный анализ показывает, что механизм состоит:

- из первичного механизма – входного звена (толкателя O_1B), кинематические параметры которого рассчитываются с применением подпрограмм PKT и PA ;

- группы Ассура 2–3 II класса 1 вида (подпрограммы PKI , PA и PKT).

Группа Ассура 2–3 имеет номер $J = 1$ (в порядке присоединения к входному звену). В соответствии с этим делением каждая группа получает новые обозначения векторов и точек звеньев (см. рис. 4.40). Силовой расчет группы Ассура 2–3 выполняется с помощью подпрограммы PCI и вспомогательных подпрограмм (что относится и к расчету входного звена) PBC и $FNPRT$.

4.3.3. Блок-схема программы кинетостатического расчета кулачково-рычажного механизма

После проведения структурного анализа механизма можно составить блок-схему программы его кинетостатического расчета (рис. 4.41). Она состоит из двух укрупненных частей:

- проектирование и расчет кулачка;
- расчет рычажного механизма (см. рис. 4.40) (с подобной методикой расчета уже ознакомились ранее).

Проектирование кулачка заключается в выполнении расчетов:

- постоянных величин;
- перемещения, аналогов скорости и ускорения толкателя по выбранному закону;
- координат теоретического профиля кулачка;
- радиуса кривизны профиля и угла давления кулачка;
- координат теоретического профиля контркулачка;
- радиуса кривизны профиля и угла давления контркулачка.

Используемые подпрограммы показаны на блок-схеме (рис. 4.41).

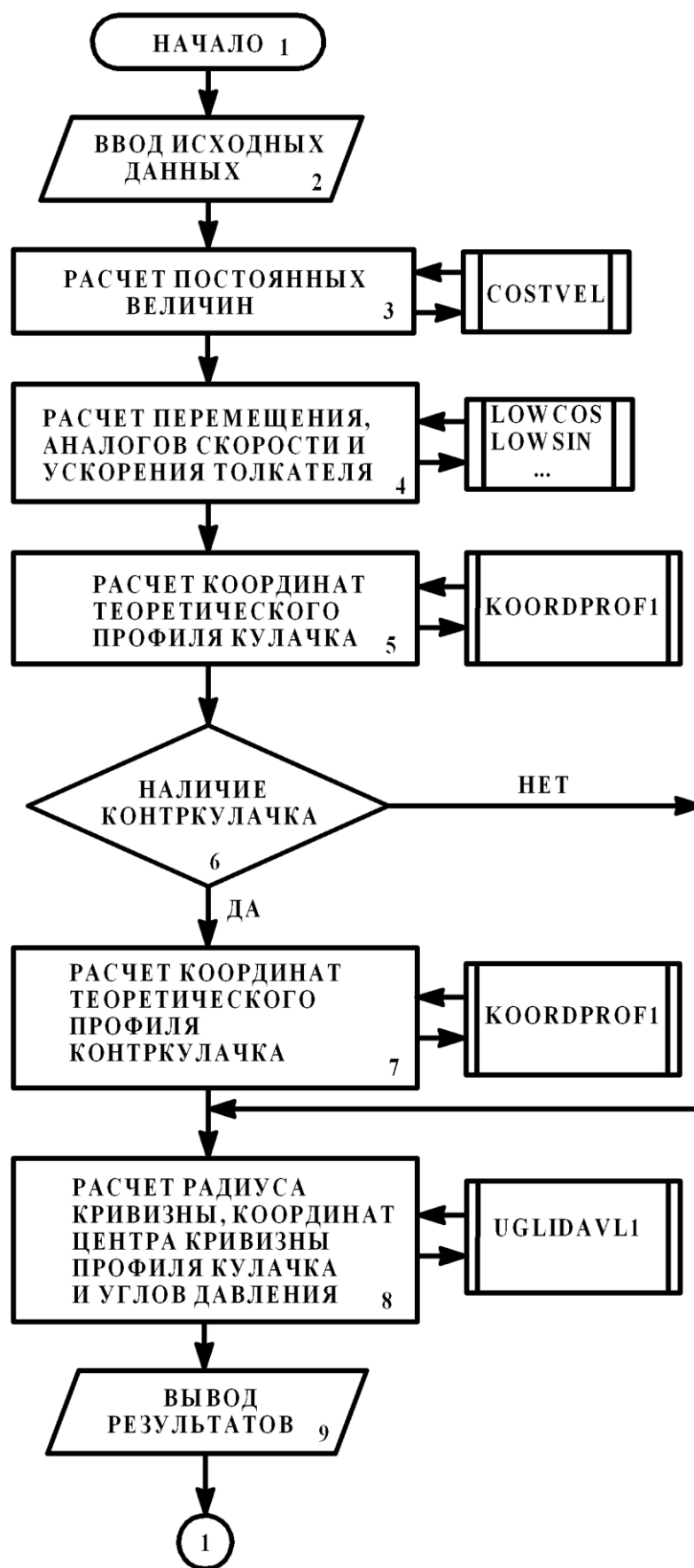


Рис. 4.41. Блок-схема программы расчета кулачково-рычажного механизма
(см. также с. 178)

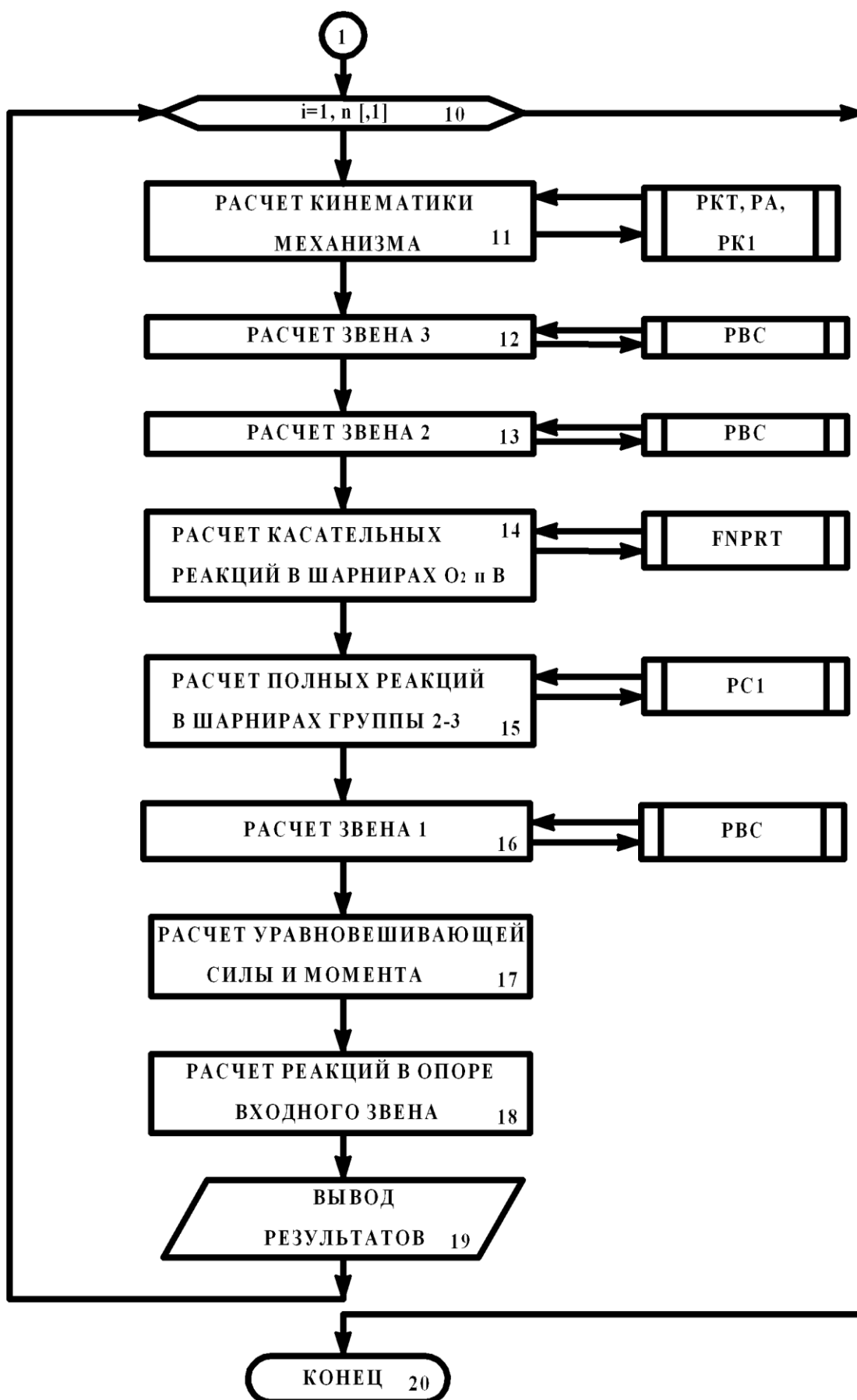


Рис. 4.41. Окончание

4.3.4. Создание проекта

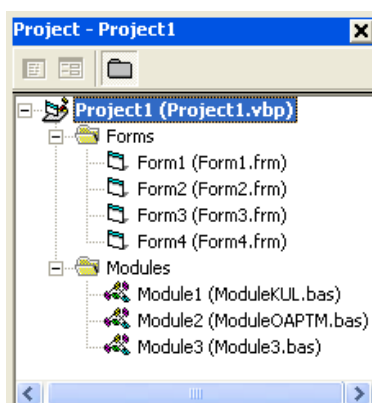


Рис. 4.42. Дерево проекта

Блок 1 отражает создание нового проекта, содержащего два модуля подпрограмм (для расчета параметров кулачка и второй – для расчета параметров рычажного механизма), третий модуль – для объявления переменных и четыре формы (рис. 4.42).

Блок 2 и блок 3 выполним с помощью формы *Form1* (рис. 4.43).

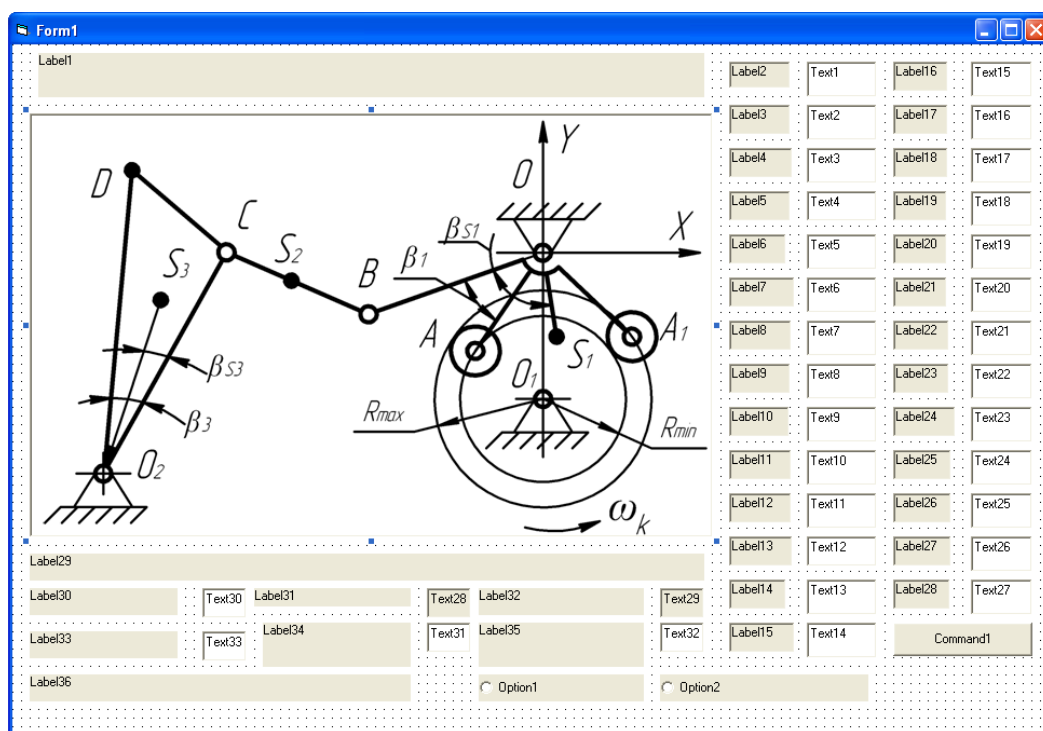


Рис. 4.43. Интерфейс формы *Form1*

На форме *Form1* размещены следующие элементы интерфейса:

- *PictureBox1* – графическая область для размещения кинематической схемы механизма;
- *Label1, 2, ..., 23* – поля для указания имен переменных, являющихся исходными данными;
- *Text1, 2, ..., 22* – ячейки для ввода значений исходных данных;
- *Option1* и *Option2* – выбор направления вращения кулачка;

- *CommandButton1* – командная кнопка для организации события присвоения данных.

В соответствии назначением на всех указанных элементах формы *Form1* необходимо сделать поясняющие надписи (рис. 4.44).

КИНЕМАТИЧЕСКАЯ СХЕМА МЕХАНИЗМА

The diagram shows a mechanism with a crank of length R_{max} rotating around point O_1 with angular velocity ω_k . A connecting rod of length S_2 is attached to the crank at point A and to a slider at point B . The slider moves along a horizontal guide. A second connecting rod of length S_3 is attached to the slider at point C and to a crank of length R_{min} rotating around point O_2 . The slider is also connected to a vertical guide at point D . The mechanism is shown in a position where the crank O_1A is at an angle β_{S1} to the horizontal, the connecting rod AB is at an angle β_1 to the horizontal, and the connecting rod BC is at an angle β_3 to the horizontal. The slider is at a distance S_2 from point B and S_3 from point C . The vertical guide is at a distance R_{min} from point O_2 .

XO1	0	m1	15.3
YO1	0.175	m2	1.08
XO2	-0.435	m3	31.5
YO2	-0.44	J1	0.146
Rmin	0.12	J2	0.005
Mmax	30	J3	8.44
O1A	0.090	LS1	0.11
O1B	0.220	LS2	0.11
BC	0.200	LS3	0.90
O2C	0.630	BS1	20
O2D	0.770	BS2	0.5
B1	22.5	BS3	10
B3	8.1	Rnc	500
WK	100	OK	

Линейные размеры в м, угловые в град., массы в кг, моменты инерции масс в кгм², сила в Н, скорость кулачка в об/мин.

ШАГ РАСЧЕТА: 1 УГОЛ УДАЛЕНИЯ: 60 УГОЛ СБЛИЖЕНИЯ: 60

ЧИСЛО ГРУПП: 1 УГОЛ СТОЯНИЯ ДАЛЬНОГО: 240 УГОЛ СТОЯНИЯ БЛИЖНЕГО: 0

НАПРАВЛЕНИЕ ВРАЩЕНИЯ КУЛАЧКА: ☒ ПО ЧАСАМ ☐ ПРОТИВ ЧАСОВ

Рис. 4.44. Ввод пояснений на форме *Form1*

Для удобства выполнения этапа отладки проекта рекомендуется сразу сохранить числовые значения переменных, чтобы не заносить их в соответствующие ячейки при каждом запуске проекта.

Для формы *Form1* предусмотрены следующие события:

- загрузка формы;
- нажатие командной кнопки.

Код события загрузки формы *Form1* должен обеспечить свойства формы и всех элементов интерфейса. Код следующий:

```
Private Sub Form_Load()
Form1.Caption = "ВВОД ИСХОДНЫХ ДАННЫХ"
Form1.Top = 800 : Form1.Left = 500 : BackColor = QBColor(11)
Form1.Height = 9300 : Form1.Width = 14500
Label1.FontSize = 16 : Label1.Alignment = 2
Label1.Caption = "КИНЕМАТИЧЕСКАЯ СХЕМА МЕХАНИЗМА"
End Sub
```


Событие нажатия командной кнопки после заполнения ячеек для исходных данных механизма осуществляет присвоение этих величин соответствующим переменным. После этого происходит расчет постоянных величин (подпрограмма *CONSTVEL*), выгрузка текущей формы *Form1* и загрузка следующей формы *Form2*.

(Примечание: в исходных данных не хватает величины межосевого расстояния, поэтому ее необходимо вычислить самостоятельно на основании схемы механизма.)

Код события нажатия командной кнопки:

```
Private Sub Command1_Click()  
Rem ПРИСВОЕНИЕ ИСХОДНЫХ ДАННЫХ  
X01 = Val(Text1): YO1 = Val(Text2): XO2 = Val(Text3): YO2 = Val(Text4)  
R1 = Val(Text5): M = Val(Text6): O1A = Val(Text7): O1B = Val(Text8)  
BC = Val(Text9): O2C = Val(Text10): O2D = Val(Text11)  
BETTA1 = Val(Text12) : BETTA3 = Val(Text13): OM = Val(Text14)  
MS1 = Val(Text15): MS2 = Val(Text16): MS3 = Val(Text17): JS1 = Val(Text18)  
JS2 = Val(Text19): JS3 = Val(Text20): LS1 = Val(Text21): LS2 = Val(Text22)  
LS3 = Val(Text23): BS1 = Val(Text24): BS2 = Val(Text25): BS3 = Val(Text26)  
PPR = Val(Text27): F1 = Val(Text28): F3 = Val(Text29): F0 = Val(Text30)  
F2 = Val(Text31): f4 = Val(Text32): J = Val(Text33)  
Rem НАПРАВЛЕНИЕ ВРАЩЕНИЯ КУЛАЧКА  
If Option1 = False Then  
ZZ = -1  
End If  
If Option2 = False Then  
ZZ = 1  
End If  
Rem РАСЧЕТ ПОСТОЯННЫХ ВЕЛИЧИН  
PI = 4 * Atn(1): RAD = PI / 180: DEG = 1 / RAD  
Rem ДЛИНА ТОЛКАТЕЛЯ  
L1 = O1A  
Rem РАСЧЕТ МЕЖОСЕВОГО РАССТОЯНИЯ  
Call PMY(XO1, YO1, L, FL)  
Rem РАСЧЕТ ПОСТОЯННЫХ ВЕЛИЧИН КУЛАЧКОВОГО ПРИВОДА  
Call constvel(R2, B0, G0, G1, T, T1, A1, A2, N1, N3)  
Rem Перевод значения максимального хода толкателя в радианы  
M = M * RAD  
Me.Hide Rem Скрыть текущую форму  
Form2.Show Rem Загрузить Form2  
End Sub
```

Теперь имеется перечень переменных, которые необходимо объявить на уровне *Module 3*:

```
Public Nmax As Integer, ZZ As Integer, AB As Double, AS1 As Double
Public LBD As Double, BC As Double, CD As Double, DK As Double
Public XF As Double, KF As Double, W1 As Double, BS2 As Double
Public DS4 As Double, FS5 As Double, BBS2 As Double, MS1 As Double
Public MS2 As Double, MS3 As Double, JS1 As Double, JS2 As Double
Public JS3 As Double, LS1 As Double, LS2 As Double, LS3 As Double
Public BS1 As Double, BS3 As Double, O1B As Double, O2C As Double
Public XO1 As Double, YO1 As Double, XO2 As Double, YO2 As Double
Public O2D As Double, PI As Double, OM As Double, DEG As Double
Public RAD As Double
```

Блоки 4–9 (см. рис. 4.41) выполняются отдельно для фазы удаления (форма *Form2*) и фазы сближения (форма *Form3*). При этом назначение и конструкция форм *Form2* и *Form3* идентичны.

Форма *Form2* (рис. 4.45) содержит:

- элементы для выбора закона изменения ускорения толкателя на фазе удаления;
- поле для демонстрации таблиц результатов расчетов параметров кулачка и контркулачка;
- шесть командных кнопок для вывода различных таблиц и перехода к следующей форме.

На форме *Form2* размещены следующие элементы интерфейса:

- *Lable1*, 2 – поля для пояснения;
- *Option 1*, ..., *10* – флажок для выбора закона движения толкателя;
- *List1* – область для вывода таблиц результатов расчетов;
- *CommandButton1*, ..., *5* – командные кнопки для выбора результатов;
- *CommandButton6* – командная кнопка для завершения расчетов.

В соответствии назначением на всех указанных элементах формы *Form2* необходимо сделать поясняющие надписи (рис. 4.46).

Положение и размеры элементов интерфейса заданы с помощью мыши, другие свойства, включая текст, – с помощью панели свойств.

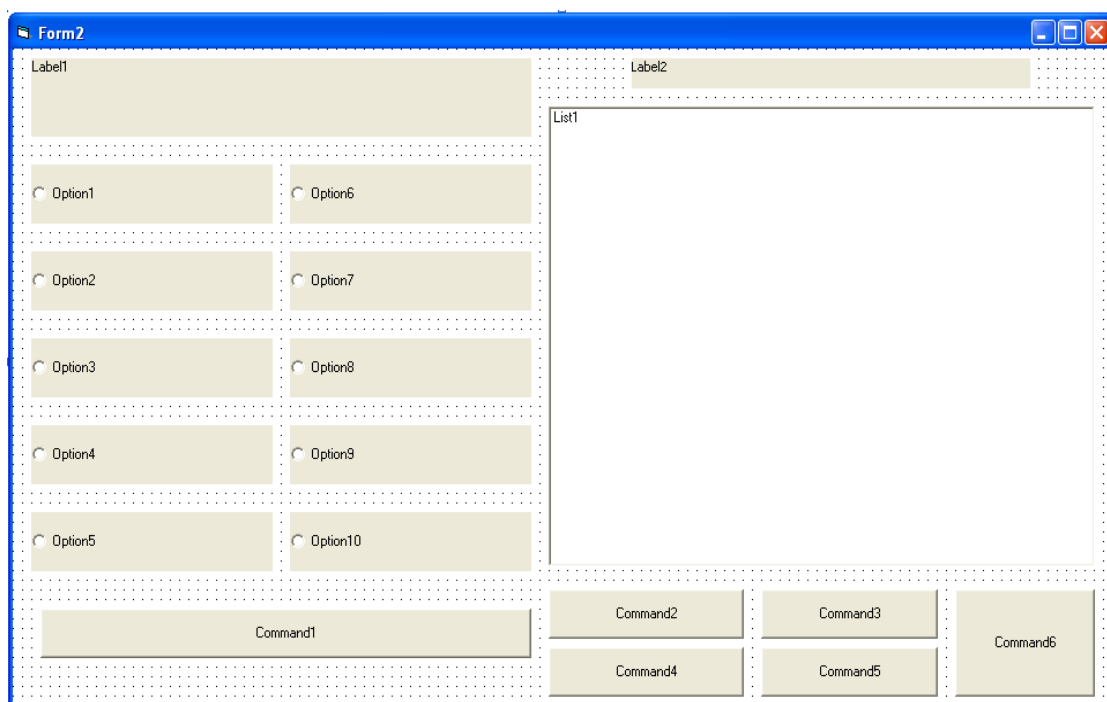


Рис. 4.45. Интерфейс формы Form2

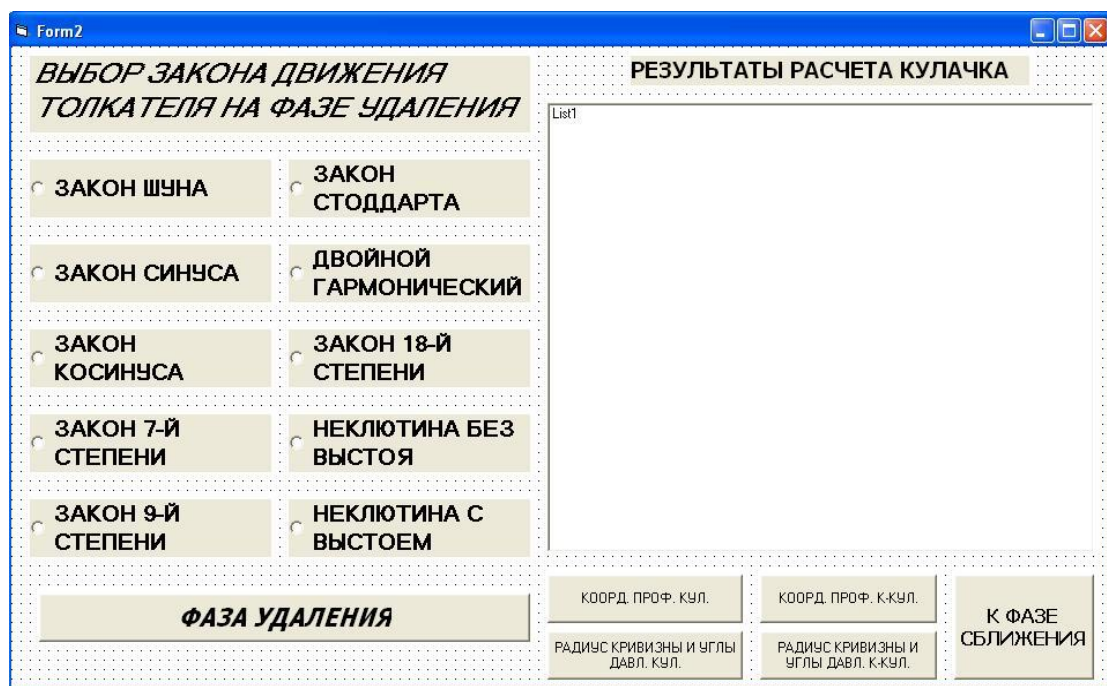


Рис. 4.46. Пояснения на форме Form2

Для формы Form2 предусмотрены следующие события:

- загрузка формы;
- активация командных кнопок.

Коды событий загрузки формы Form2 и активации командных кнопок следующие:

```

Private Sub Form_Load()
Form2.Caption = "ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ"
Form2.Top = 800 : Form2.Left = 500 : BackColor = QBColor(11)
Form2.Height = 9300 : Form2.Width = 14500
End Sub

Private Sub Command1_Click()
ReDim P(Nmax), P1(Nmax), P2(Nmax), S3(Nmax - 1), S4(Nmax), S5(Nmax)
ReDim S6(Nmax), S7(Nmax), S8(Nmax), FUN1(N1), FUN2(N1), FUN3(N1)
ReDim FUN4(N1), FUN41(N1), FUN5(N1), FUN6(N1), FUN7(N1), FUN8(N1)
ReDim FUN16(N1), FUN17(N1), FUN18(N1), FUN181(N1), FUN19(N1)
ReDim FUN20(N1), FUN21(N1), FUN22(N1)

Rem ПОДГОТОВКА ТАБЛИЦЫ
List1.Clear : List1.FontSize = 14
znak1 = "      FK": znak2 = "      FT": znak3 = "      FT1": znak4 = "      FT2"
znak5 = "===== "
znak6 = "КИНЕМАТИКА ТОЛКАТЕЛЯ НА ФАЗЕ УДАЛЕНИЯ"
znak7 = "      ГРАД"
List1.AddItem Format$(znak6, "")
List1.AddItem Format$(znak5, "")
List1.AddItem Format$(znak1, "") + "      " + Format(znak2, "") + _
"      " + Format(znak3, "") + "      " + Format(znak4, "")
List1.AddItem Format$(znak7, "") + "      " + Format(znak7, "") + "      " _
+ Format(znak7, "") + "      " + Format(znak7, "")
List1.AddItem Format$(znak5, "")
Rem ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ
If Option1 = True Then
GoTo 1
End If
If Option2 = True Then
GoTo 2
End If
If Option3 = True Then
GoTo 3
End If
If Option4 = True Then
GoTo 4
End If
If Option5 = True Then
GoTo 5
End If
If Option6 = True Then
GoTo 6
End If

```

```

If Option7 = True Then
GoTo 7
End If
If Option8 = True Then
GoTo 8
End If
If Option9 = True Then
GoTo 9
End If
If Option10 = True Then
GoTo 10
End If
1 Call LOWSUNA(P(), P1(), P2()): GoTo 11
2 Call LOWSIN(P(), P1(), P2()): GoTo 11
3 Call LOWCOS(P(), P1(), P2()): GoTo 11
4 Call LOWSEVEN(P(), P1(), P2()): GoTo 11
5 Call LOWNINE(P(), P1(), P2()): GoTo 11
6 Call LOWSTODART(P(), P1(), P2()): GoTo 11
7 Call DOUBLEGARM(P(), P1(), P2()): GoTo 11
8 Call EIGHTEEN(P(), P1(), P2()): GoTo 11
9 Call LOWNECL(P(), P1(), P2()): GoTo 11
10 Call LOWNECLWITH(P(), P1(), P2()): GoTo 11
11 Rem КИНЕМАТИЧЕСКИЕ ПАРАМЕТРЫ ФАЗЫ УДАЛЕНИЯ
For i = 1 To N1
F = (i - 1) * F0 * ZZ: FUN1(i) = F: S3(i) = M * P(i): S4(i) = A1 * P1(i)
S5(i) = A2 * P2(i): S = S3(i): S1 = S4(i): S2 = S5(i): F9 = F
List1.AddItem Format$(F, " 000.0") + " " + Format(S3(i) * DEG, "000.0")_
+ " " + Format(S4(i), "0.0000") + " " + Format(S5(i), "0.0000")
Rem РАСЧЕТ КООРДИНАТ ПРОФИЛЯ КУЛАЧКА
Call KOORDPROF1(B0, F9, 0, 0, S, X0, Y0, R0, AlfaR0)
FUN2(i) = X0: FUN3(i) = Y0: FUN4(i) = R0: FUN41(i) = AlfaR0
Rem РАСЧЕТ КООРДИНАТ ПРОФИЛЯ КОНТРКУЛАЧКА
Call KOORDPROF1(B0, F9, T * RAD, -G1, S, X0K, Y0K, R0K, AlfaR0K)
FUN16(i) = X0K: FUN17(i) = Y0K: FUN18(i) = R0K: FUN181(i) = AlfaR0K
Rem РАСЧЕТ РАДИУСА КРИВИЗНЫ И УГЛОВ ДАВЛЕНИЯ
Rem КУЛАЧКА
Call UGLIDAVL1(B0, F9, 0, S, S1, S2, RKR, X3, Y3, D0)
FUN5(i) = RKR: FUN6(i) = D0 * DEG: FUN7(i) = X3: FUN8(i) = Y3
Rem РАСЧЕТ РАДИУСА КРИВИЗНЫ И УГЛОВ ДАВЛЕНИЯ
Rem КОНТРКУЛАЧКА
Call UGLIDAVL1(B0, F9, T * RAD, S, S1, S2, RKRK, x3K, y3K, d0K)
FUN19(i) = RKRK: FUN20(i) = d0K * DEG: FUN21(i) = x3K: FUN22(i) = y3K
Next
500 End Sub

```

```

Private Sub Command2_Click()
Rem ПОДГОТОВКА ТАБЛИЦЫ
List1.Clear
znak1 = "    КООРДИНАТЫ ПОФИЛЯ КУЛАЧКА"
znak2 = "    НА ФАЗЕ УДАЛЕНИЯ"
znak3 = "=====
znak4 = "  FK": znak5 = "   XO": znak6 = "   YO": znak7 = "  RO"
znak8 = "  град": znak9 = "   м": znak10 = " FRO":
List1.AddItem Format$(znak1, ""): List1.AddItem Format$(znak2, "")
List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak4, "") + "   " + Format(znak5, "") + "   " + _
Format(znak6, "") + "   " + Format(znak7, "") + "   " + Format(znak10, "")
List1.AddItem Format$(znak8, "") + "   " + Format(znak9, "") + "   " + _
Format(znak9, "") + "   " + Format(znak9, "") + "   " + Format(znak8, "")
List1.AddItem Format$(znak3, "")
For i = 1 To N1
List1.AddItem Format$(FUN1(i), "000.0") + "   " + Format(FUN2(i), "0.00000") + _
"   " + Format(FUN3(i), "0.00000") + "   " + Format(FUN4(i), "0.00000") + _
"   " + Format(FUN41(i), "000.0")
Next
End Sub

```

```

Private Sub Command3_Click()
List1.Clear
znak1 = "    КООРДИНАТЫ ПОФИЛЯ КОНТРКУЛАЧКА"
znak2 = "    НА ФАЗЕ УДАЛЕНИЯ "
znak3 = "=====
znak4 = "    FK": znak5 = "    XO": znak6 = "    YO": znak7 = "    RO"
znak8 = "    град": znak9 = "    м": znak10 = " FROK"
List1.AddItem Format$(znak1, ""): List1.AddItem Format$(znak2, "")
List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak4, "") + "   " + Format(znak5, "") + "   " + _
Format(znak6, "") + "   " + Format(znak7, "") + "   " + Format(znak10, "")
List1.AddItem Format$(znak8, "") + "   " + Format(znak9, "") + "   " + _
Format(znak9, "") + "   " + Format(znak9, "") + "   " + Format(znak8, "")
List1.AddItem Format$(znak3, "")
For i = 1 To N1
List1.AddItem Format$(FUN1(i), "  000.0") + "   " + _
Format(FUN16(i), "0.00000") + "   " + Format(FUN17(i), "0.00000") + _
"   " + Format(FUN18(i), "0.00000") + "   " + Format(FUN181(i), "###.0")
Next
End Sub

```

```

Private Sub Command4_Click()
List1.Clear: znak1 = "ПАРАМЕТРЫ РАДИУСА КРИВИЗНЫ "

```

```

znak2 = "            И УГЛОВ ДАВЛЕНИЯ КУЛАЧКА"
znak3 = "            НА ФАЗЕ УДАЛЕНИЯ"
znak4 = "=====
znak5 = "   FK": znak6 = "   RAD": znak7 = "   FRAD": znak8 = "   ХСК"
znak9 = "   YСК": znak10 = "   ГРАД": znak11 = "   М": znak12 = "   ГРАД "
znak13 = "   М": znak14 = "   М": List1.AddItem Format$(znak1, "")
List1.AddItem Format$(znak2, ""): List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak4, "")
List1.AddItem Format$(znak5, "") + "   " + Format(znak6, "") + "   " _
+ Format(znak7, "") + "   " + Format(znak8, "") + "   " + Format(znak9, "")
List1.AddItem Format$(znak10, "") + "   " + Format(znak11, "") + "   " _
+ Format(znak12, "") + "   " + Format(znak13, "") + "   " + Format(znak14, "")
List1.AddItem Format$(znak4, "")
For i = 1 To N1
List1.AddItem Format$(FUN1(i), "   000.0") + "   " + _
Format(FUN5(i), "0.00000") + "   " + Format(FUN6(i), "00.0") + "   " + _
Format(FUN7(i), "0.0000") + "   " + Format(FUN8(i), "0.0000")
Next
End Sub

```

```

Private Sub Command5_Click()
List1.Clear
znak1 = "   ПАРАМЕТРЫ РАДИУСА КРИВИЗНЫ"
znak2 = "   И УГЛОВ ДАВЛЕНИЯ КОНТРКУЛАЧКА"
znak3 = "   НА ФАЗЕ УДАЛЕНИЯ"
znak4 = "=====
znak5 = "   FK": znak6 = "   RAD": znak7 = "   FRAD": znak8 = "   ХСК"
znak9 = "   YСК": znak10 = "   ГРАД": znak11 = "   М": znak12 = "   ГРАД "
znak13 = "   М": znak14 = "   М": List1.AddItem Format$(znak1, "")
List1.AddItem Format$(znak2, ""): List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak4, ""):
List1.AddItem Format$(znak5, "") + "   " + Format(znak6, "") + "   " _
+ Format(znak7, "") + "   " + Format(znak8, "") + "   " + Format(znak9, "")
List1.AddItem Format$(znak10, "") + "   " + Format(znak11, "") + "   " _
+ Format(znak12, "") + "   " + Format(znak13, "") + "   " + Format(znak14, "")
List1.AddItem Format$(znak4, "")
For i = 1 To N1
List1.AddItem Format$(FUN1(i), "   000.0") + "   " + _
Format(FUN19(i), "0.00000") + "   " + Format(FUN20(i), "00.0") + "   " + _
Format(FUN21(i), "0.00000") + "   " + Format(FUN22(i), "0.00000")
Next
End Sub

```

```

Private Sub Command6_Click()
Me.Hide: Form3.Show
End Sub

```

Теперь имеется перечень переменных, которые необходимо объявить:

- на уровне *Module3*:

```
Public P() As Double, P1() As Double, P2() As Double, S As Double
Public F9 As Double, S3() As Double, S4() As Double, S5() As Double
Public S1 As Double, S2 As Double
```

- на уровне формы *Form2*:

```
Dim FUN1() As Double, FUN2() As Double, FUN3() As Double
Dim FUN4() As Double, FUN41() As Double, FUN5() As Double
Dim FUN6() As Double, FUN7() As Double, FUN8() As Double
Dim FUN16() As Double, FUN17() As Double, FUN18() As Double
Dim FUN181() As Double, FUN19() As Double, FUN20() As Double
Dim FUN21() As Double, FUN22() As Double, X0 As Double, Y0 As Double
Dim R0 As Double, X0K As Double, Y0K As Double, R0K As Double
Dim RKRK As Double, x3K As Double, y3K As Double, d0K As Double
Dim AlfaR0 As Double, AlfaR0K As Double
```

Форма *Form3* повторяет назначения предыдущей формы *Form2* с той лишь разницей, что рассматривается фаза сближения (рис. 4.47).

На форме *Form3* размещены те же элементы управления, что и на форме *Form2*. Они обладают аналогичным расположением и другими свойствами.

На форме *Form3* также требуется ввести пояснения для ее элементов (рис. 4.48).

Коды событий, предусмотренных в форме *Form3*:

```
Private Sub Form_Load()
Form3.Caption = "ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ"
Form3.Top = 800: Form3.Left = 500: BackColor = QBColor(11)
Form3.Height = 9300: Form3.Width = 14500
End Sub

Private Sub Command1_Click()
ReDim FUN1(N3), FUN2(N3), FUN3(N3), FUN4(N3), FUN41(N3), FUN5(N3)
ReDim FUN6(N3), FUN7(N3), FUN8(N3), FUN9(N3), FUN10(N3), FUN11(N3)
ReDim FUN111(N3), FUN12(N3), FUN13(N3), FUN14(N3), FUN15(N3)
List1.Clear: List1.FontSize = 14 : znak1 = "    FK": znak2 = "    FT"
znak3 = "    FT1": znak4 = "    FT2"
znak5 = "===== "
znak6 = "КИНЕМАТИКА ТОЛКАТЕЛЯ НА ФАЗЕ СБЛИЖЕНИЯ"
znak7 = "ГРАД": List1.AddItem Format$(znak6, "")
```



```

List1.AddItem Format$(znak5, "")
List1.AddItem Format$(znak1, "") + "    " + Format(znak2, "") + _
    "    " + Format(znak3, "") + "    " + Format(znak4, "")
List1.AddItem Format$(znak7, "") + "    " + Format(znak7, "") + "    " + _
    Format(znak7, "") + "" + Format(znak7, "")
List1.AddItem Format$(znak5, "")
Rem ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ
If Option1 = True Then
GoTo 1
End If
If Option2 = True Then
GoTo 2
End If
If Option3 = True Then
GoTo 3
End If
If Option4 = True Then
GoTo 4
End If
If Option5 = True Then
GoTo 5
End If
If Option6 = True Then
GoTo 6
End If
If Option7 = True Then
GoTo 7
End If
If Option8 = True Then
GoTo 8
End If
If Option9 = True Then
GoTo 9
End If
If Option10 = True Then
GoTo 10
End If
1 Call LOWSUNA(P(), P1(), P2()): GoTo 11
2 Call LOWSIN(P(), P1(), P2()): GoTo 11
3 Call LOWCOS(P(), P1(), P2()): GoTo 11
4 Call LOWSEVEN(P(), P1(), P2()): GoTo 11
5 Call LOWNINE(P(), P1(), P2()): GoTo 11
6 Call LOWSTODART(P(), P1(), P2()): GoTo 11
7 Call DOUBLEGARM(P(), P1(), P2()): GoTo 11
8 Call EIGHTEEN(P(), P1(), P2()): GoTo 11
9 Call LOWNECL(P(), P1(), P2()): GoTo 11

```

```

10 Call LOWNECLWITH(P(), P1(), P2()): GoTo 11
11 Rem КИНЕМАТИЧЕСКИЕ ПАРАМЕТРЫ ФАЗЫ СБЛИЖЕНИЯ
    For i = N3 To 1 Step -1
S6(i) = M * P(i): S7(i) = -A1 * P1(i): S8(i) = A2 * P2(i)
S = S6(i): S1 = S7(i): S2 = S8(i)
FUN1(i) = ZZ * (F1 + F2 + (N3 - i) * F0): F9 = FUN1(i)
List1.AddItem Format$(FUN1(i), "000.0") + " " + _
Format(S6(i) * DEG, "000.0") + " " + Format(S7(i), "0.0000") + " " + _
Format(S8(i), "0.0000")
Rem РАСЧЕТ КООРДИНАТ ПРОФИЛЯ КУЛАЧКА
Call KOORDPROF1(B0, F9, 0, 0, S, X0, Y0, R0, AlfaR0)
FUN2(i) = X0: FUN3(i) = Y0: FUN4(i) = R0: FUN41(i) = AlfaR0
Rem РАСЧЕТ КООРДИНАТ ПРОФИЛЯ КОНТРКУЛАЧКА
Call KOORDPROF1(B0, F9, T * RAD, -G1, S, X0K, Y0K, R0K, AlfaR0K)
FUN9(i) = X0K: FUN10(i) = Y0K: FUN11(i) = R0K: FUN111(i) = AlfaR0K
Rem РАСЧЕТ РАДИУСА КРИВИЗНЫ И УГЛОВ ДАВЛЕНИЯ
Rem КУЛАЧКА
Call UGLIDAVL1(B0, F9, 0, S, S1, S2, RKR, X3, Y3, D0)
FUN5(i) = RKR: FUN6(i) = D0 * DEG: FUN7(i) = X3: FUN8(i) = Y3
Rem РАСЧЕТ РАДИУСА КРИВИЗНЫ И УГЛОВ ДАВЛЕНИЯ
Rem КОНТРКУЛАЧКА
Call UGLIDAVL1(B0, F9, T * RAD, S, S1, S2, RKRK, x3K, y3K, d0K)
FUN12(i) = RKRK: FUN13(i) = d0K * DEG: FUN14(i) = x3K: FUN15(i) = y3K
Next
    End Sub

```

```

Private Sub Command2_Click()
Rem ПОДГОТОВКА ТАБЛИЦЫ
List1.Clear^ znak1 = "КОРДИНАТЫ ПРОФИЛЯ КУЛАЧКА"
znak2 = "    НА ФАЗЕ СБЛИЖЕНИЯ"
znak3 = "=====
znak4 = "    FK": znak5 = "    XO": znak6 = "    YO": znak7 = "    RO"
znak8 = "    град": znak9 = "    м": znak10 = "    FRO"
List1.AddItem Format$(znak1, ""): List1.AddItem Format$(znak2, "")
List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak4, "") + " " + Format(znak5, "") + " " + _
Format(znak6, "") + " " + Format(znak7, "") + " " + Format(znak10, "")
List1.AddItem Format$(znak8, "") + " " + Format(znak9, "") + " " + _
Format(znak9, "") + " " + Format(znak9, "") + " " + Format(znak8, "")
List1.AddItem Format$(znak3, "")
For i = N3 To 1 Step -1
List1.AddItem Format$(FUN1(i), "000.0") + " " + _
Format(FUN2(i), "0.00000") + " " + Format(FUN3(i), "0.00000") + " " + _
Format(FUN4(i), "0.00000") + " " + Format(FUN41(i), "0.0")
Next
    End Sub

```

```

Private Sub Command3_Click()
Rem ПОДГОТОВКА ТАБЛИЦЫ
List1.Clear
znak1 = "КОРДИНАТЫ ПРОФИЛЯ КОНТРКУЛАЧКА"
znak2 = "                НА ФАЗЕ СБЛИЖЕНИЯ"
znak3 = "=====
znak4 = "    FK": znak5 = "    XO": znak6 = "    YO": znak7 = "    RO"
znak8 = "    град": znak9 = "    м":  znak10 = "    FROK"
List1.AddItem Format$(znak1, ""): List1.AddItem Format$(znak2, "")
List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak4, "") + "    " + Format(znak5, "") + "    " + _
Format(znak6, "") + "    " + Format(znak7, "") + "    " + Format(znak10, "")
List1.AddItem Format$(znak8, "") + "    " + Format(znak9, "") + "    " + _
Format(znak9, "") + "    " + Format(znak9, "") + "    " + Format(znak8, "")
List1.AddItem Format$(znak3, "")
For i = N3 To 1 Step -1
List1.AddItem Format$(FUN1(i), "    000.0") + "    " + _
Format(FUN9(i), "0.00000") + "    " + Format(FUN10(i), "0.00000") + "    " + _
Format(FUN11(i), "0.00000") + "    " + Format(FUN11(i), "0.0")
Next
End Sub

```

```

Private Sub Command4_Click()
List1.Clear
znak1 = "ПАРАМЕТРЫ РАДИУСА КРИВИЗНЫ"
znak2 = "                И УГЛОВ ДАВЛЕНИЯ КУЛАЧКА":
znak3 = "                НА ФАЗЕ СБЛИЖЕНИЯ"
znak4 = "=====
znak5 = "    FK": znak6 = "    RAD": znak7 = "    FRAD": znak8 = "    XCK"
znak9 = "    YCK": znak10 = "    ГРАД": znak11 = "    М"
znak12 = "    ГРАД ": znak13 = "    М ": znak14 = "    М "
List1.AddItem Format$(znak1, ""): List1.AddItem Format$(znak2, "")
List1.AddItem Format$(znak3, ""): List1.AddItem Format$(znak4, "")
List1.AddItem Format$(znak5, "") + "    " + Format(znak6, "") + "    " + _
+ Format(znak7, "") + "    " + Format(znak8, "") + "    " + Format(znak9, "")
List1.AddItem Format$(znak10, "") + "    " + Format(znak11, "") + "    " + _
+ Format(znak12, "") + "    " + Format(znak13, "") + "    " + Format(znak14, "")
List1.AddItem Format$(znak4, "")
For i = N3 To 1 Step -1
List1.AddItem Format$(FUN1(i), "    000.0") + "    " + _
Format(FUN5(i), "0.00000") + "    " + Format(FUN6(i), "0.00") + "    " + _
Format(FUN7(i), "0.00000") + "    " + Format(FUN8(i), "0.00000")
Next
End Sub

```

```

Private Sub Command5_Click()
List1.Clear
znak1 = "ПАРАМЕТРЫ РАДИУСА КРИВИЗНЫ"
znak2 = "                И УГЛОВ ДАВЛЕНИЯ КОНТРКУЛАЧКА"
znak3 = "                НА ФАЗЕ СБЛИЖЕНИЯ"
znak4 = "=====
znak5 = "  FK": znak6 = "    RAD": znak7 = "    FRAD": znak8 = " XCK"
znak9 = "  YCK": znak10 = " ГРАД": znak11 = " М": znak12 = "    ГРАД"
znak13 = "  М": znak14 = "    М" : List1.AddItem Format$(znak1, "")
List1.AddItem Format$(znak2, ""): List1.AddItem Format$(znak3, "")
List1.AddItem Format$(znak4, "")
List1.AddItem Format$(znak5, "") + "  " + Format(znak6, "") + "  " _
+ Format(znak7, "") + "  " + Format(znak8, "") + "  " + Format(znak9, "")
List1.AddItem Format$(znak10, "") + "  " + Format(znak11, "") + "  " _
+ Format(znak12, "") + "  " + Format(znak13, "") + "  " + Format(znak14, "")
List1.AddItem Format$(znak4, "")
For i = N3 To 1 Step -1
List1.AddItem Format$(FUN1(i), "    000.0") + "    " + _
Format(FUN12(i), "0.00000") + "    " + Format(FUN13(i), "0.00") + "    " + _
Format(FUN14(i), "0.00000") + "    " + Format(FUN15(i), "0.00000")
Next
End Sub

```

```

Private Sub Command6_Click()
Me.Hide : Form4.Show
End Sub

```

Теперь имеется перечень переменных, которые необходимо объявить:

- на уровне *Module3*:

```
Public S6() As Double, S7() As Double, S8() As Double
```

- на уровне формы *Form3*:

```

Dim FUN1() As Double, FUN2() As Double, FUN3() As Double
Dim FUN4() As Double, FUN41() As Double, FUN5() As Double
Dim FUN6() As Double, FUN7() As Double , FUN8() As Double
Dim FUN9() As Double, FUN10() As Double, FUN11() As Double
Dim FUN111() As Double, FUN12() As Double, FUN13() As Double
Dim FUN14() As Double, FUN15() As Double, X0 As Double, Y0 As Double
Dim R0 As Double, X0K As Double , Y0K As Double, R0K As Double
Dim RK RK As Double, x3K As Double, y3K As Double, d0K As Double
Dim AlfaR0 As Double, AlfaR0K As Double

```

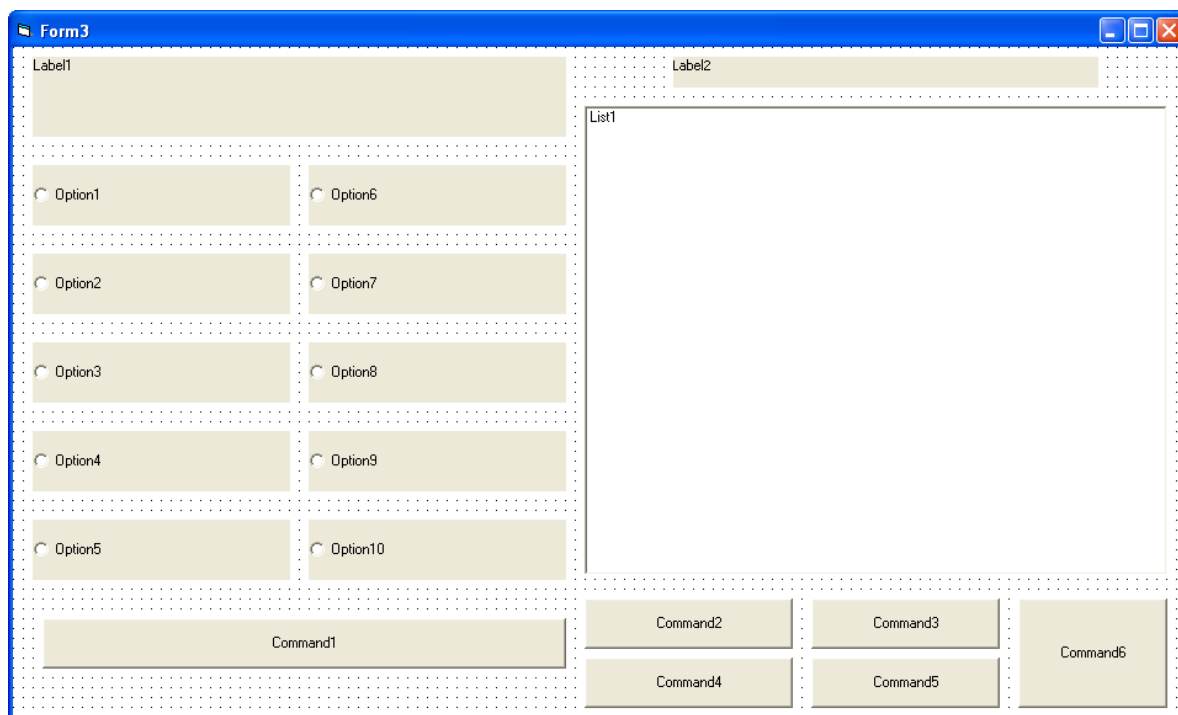


Рис. 4.47. Интерфейс формы Form3

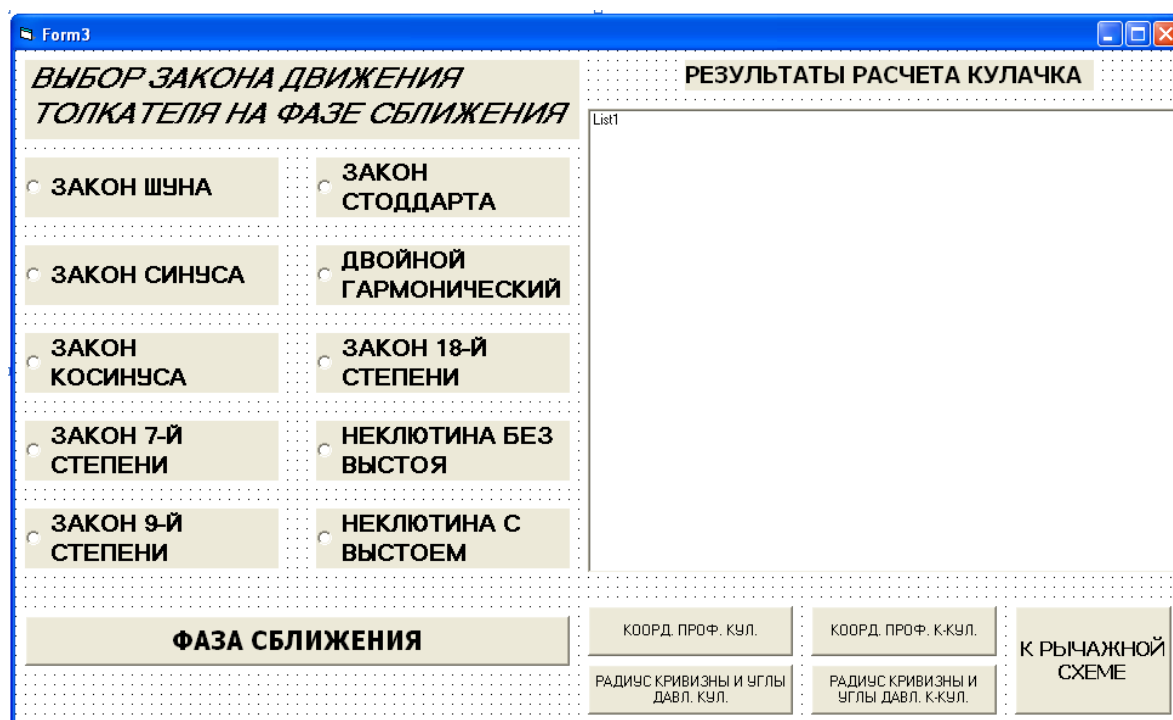


Рис. 4.48. Пояснения на форме Form3

Теперь определены все необходимые параметры кулачкового привода (см. рис. 4.41). Приступаем к исследованию рычажной схемы: толкателя $0-1$ и группы Ассура 2–3 (см. рис. 4.39 и 4.40).

Блоки 10–20 (см. рис. 4.41) показывают последовательность кинематического и силового расчета рычажного механизма (подробнее см. в подразд. 4.1 и 4.2). Для этих расчетов предназначена форма *Form4*. На ней предусмотрены:

- возможность выбора фазы движения толкателя;
- область для демонстрации таблицы результатов расчетов;
- область для вывода графиков;
- командные кнопки для управления событиями.

Интерфейс формы *Form4* (рис. 4.49) содержит следующие элементы:

- *Lable1* – поле для пояснения;
- *Check1, 2* – выбор фазы движения механизма;
- *List1* – область для вывода таблиц результатов расчетов;
- *Text1* – область для ввода пояснений для каждого графика;
- *PictureBox1* – графическая область для размещения графиков результатов расчетов;
- *CommandButton1, 2* – командные кнопки для вывода таблиц кинематического и силового расчетов соответственно;
- *CommandButton3, ..., 6* – командные кнопки для вывода графиков кинематического и силового расчетов.

На всех указанных элементах *Form4* в соответствии с их назначением необходимо сделать поясняющие надписи (рис. 4.50), кроме области *Text1*, содержание которой меняется в процессе выполнения программы в зависимости от демонстрируемого графика.

После разработки интерфейса формы *Form4* переходим к созданию программных кодов для следующих ее процедур и событий:

- загрузка формы;
- нажатие командных кнопок.

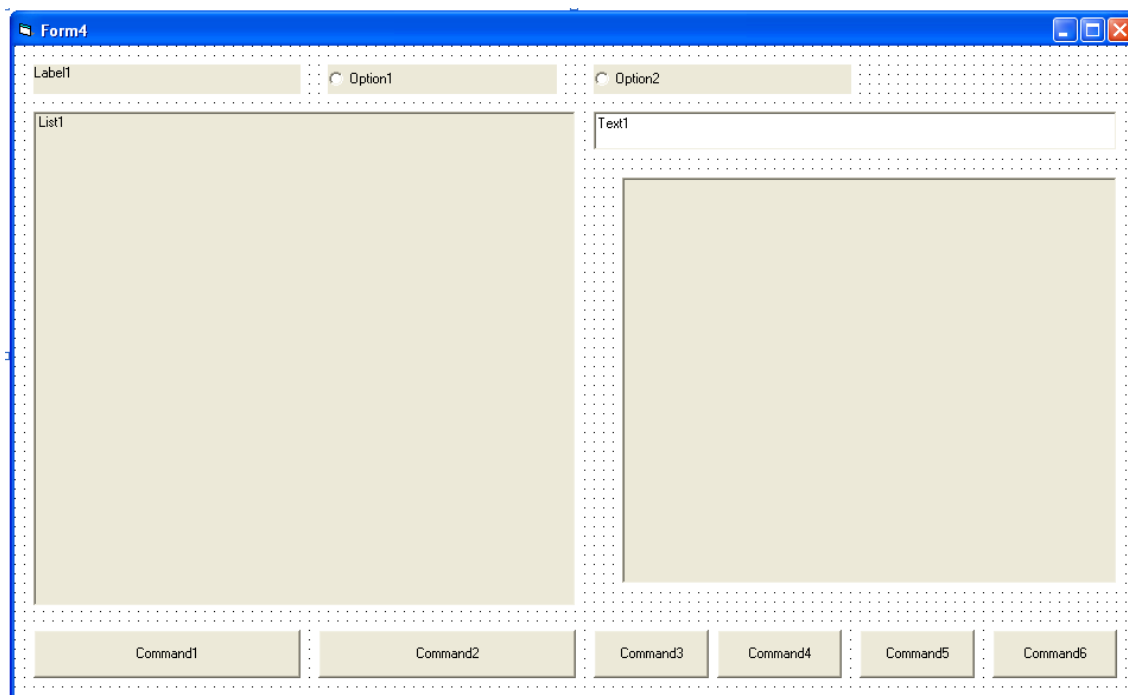


Рис. 4.49. Интерфейс формы Form4

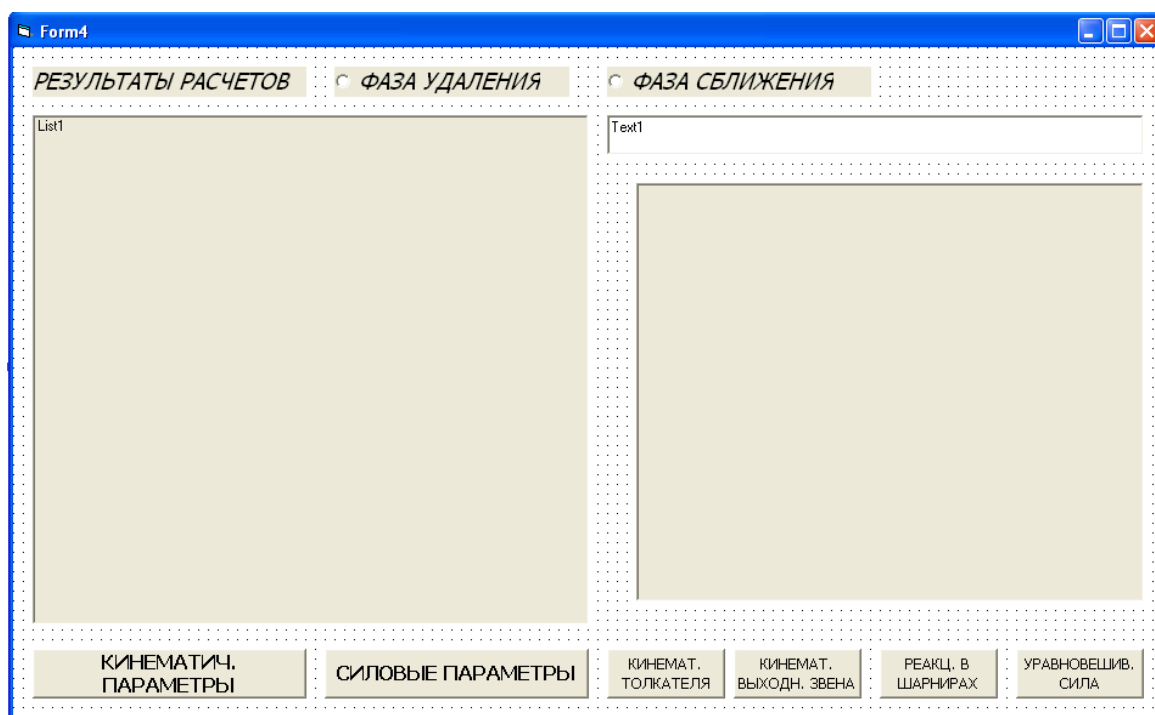


Рис. 4.50. Пояснения на форме Form4

Коды данных событий:

```
Private Sub Form_Load()
Form4.Caption = "РЕЗУЛЬТАТЫ РАСЧЕТОВ"
Form4.Top = 800 : Form4.Left = 500 : BackColor = QBColor(11)
Form4.Height = 9300 : Form4.Width = 14500
End Sub
```

```

Private Sub Command1_Click()
Rem СОЗДАНИЕ МАССИВОВ ДЛЯ ВЫВОДА ГРАФИКОВ
ReDim FUNGR1(1 To N1) As Double, FUNGR2(N1) As Double
ReDim FUNGR3(N1) As Double, FUNGR4(1 To N1) As Double
ReDim FUNGR5(1 To N1) As Double
Rem СОЗДАНИЕ МАССИВОВ ДЛЯ ВЫВОДА ТАБЛИЦ
ReDim FUN1(N1) As Double, FUN2(N1) As Double, FUN3(N1) As Double
ReDim FUN4(N1) As Double, FUN5(N1) As Double, FUN6(N1) As Double
ReDim FUN7(N1) As Double, FUN8(N1) As Double
Rem СОЗДАНИЕ МАССИВОВ ПЕРЕМЕННЫХ
ReDim L2(J) As Double, L2(J) As Double, L3(J) As Double, XA(J) As Double
ReDim YA(J) As Double, xc(J) As Double, yc(J) As Double, vc(J) As Double
ReDim fvc(J) As Double, wc(J) As Double, fwc(J) As Double, va(J) As Double
ReDim fva(J) As Double, wa(J) As Double, fwa(J) As Double, rd(J) As Double
ReDim BD(J) As Double, Z(J) As Double, fx1(J) As Double, fx31(J) As Double
ReDim f14(J) As Double, f24(J) As Double, f31(J) As Double, re(J) As Double
ReDim be(J) As Double, r4(J) As Double, b4(J) As Double, r(J) As Double
ReDim fp(J) As Double, B3(J) As Double, qp(J) As Double, B(J) As Double
ReDim rct(J) As Double, rat(J) As Double
Rem ПОДГОТОВКА ТАБЛИЦЫ 1
List1.Clear : List1.FontSize = 14 : znak1 = "    FK" : znak2 = "    F1"
znak3 = "    F2" : znak4 = "    F3"
znak5 = "===== "
znak6 = "КИНЕМАТИКА МЕХАНИЗМА " : znak7 = "    ГРАД"
List1.AddItem Format$(znak6, "") : List1.AddItem Format$(znak5, "")
List1.AddItem Format$(znak1, "") + "    " + Format(znak2, "") + "    " + _
Format(znak3, "") + "    " + Format(znak4, "")
List1.AddItem Format$(znak7, "") + "" + Format(znak7, "") + "" + _
Format(znak7, "") + "" + Format(znak7, "")
List1.AddItem Format$(znak5, "")
Rem РАСЧЕТ РЫЧАЖНОГО МЕХАНИЗМА
Rem ВВОД ИСХОДНЫХ ДАННЫХ
L2(1) = BC: L3(1) = O2C: xc(1) = XO2: yc(1) = YO2: re(1) = O2D
be(1) = БЕТТА3 : vc(1) = 0: wc(1) = 0: fvc(1) = 0: fwc(1) = 0: Z(1) = -1
Rem ПЕРЕВОД УГЛОВ В РАДИАНЫ
БЕТТА1 = БЕТТА1 * RAD : БЕТТА3 = БЕТТА3 * RAD
BS1 = BS1 * RAD : BS2 = BS2 * RAD : BS3 = BS3 * RAD
Rem РАСЧЕТ ПОСТОЯННЫХ ВЕЛИЧИН
Bmin = FNARC((L ^ 2 + L1 ^ 2 - R2 ^ 2) / (2 * L1 * L))
Bmax = FNARC((L ^ 2 + L1 ^ 2 - R1 ^ 2) / (2 * L1 * L))
Rem НАЧАЛЬНОЕ ПОЛОЖЕНИЕ ТОЛКАТЕЛЯ
FO1BN = 2 * PI - FL - Bmax - БЕТТА1
If FO1BN < 0 Then FO1BN = FO1BN + 2 * PI
If FO1BN > 2 * PI Then FO1BN = FO1BN - 2 * PI
ОМ = ОМ * PI / 30 Rem УГЛОВАЯ СКОРОСТЬ КУЛАЧКОВОГО ВАЛА

```



```

If Option2 = True Then GoTo 12 Rem ВЫБОР ФАЗЫ ДВИЖЕНИЯ
NX = 1: NY = N1: NS = 1: GoTo 13
12 NX = N3: NY = 1: NS = -1
Rem СОЗДАНИЕ МАССИВА С ВОЗМОЖНОСТЬЮ ВЫБОРА
Rem ГРАНИЦ И ШАГА
13          For i = NX To NY Step NS
Rem УГОЛ ПОВОРОТА ВХОДНОГО ЗВЕНА В ЗАВИСИМОСТИ ОТ ФАЗЫ
If Option2 = True Then GoTo 14
F = (i - 1) * F0
Rem ФАЗА УДАЛЕНИЯ
11 FO1B = FO1BN + S3(i) * ZZ: FO1B1 = S4(i): FO1B2 = S5(i)
GoTo 15 ПЕРЕХОД ЧЕРЕЗ ФАЗУ СБЛИЖЕНИЯ
Rem ФАЗА СБЛИЖЕНИЯ
14 FO1B = FO1BN + S6(i) * ZZ: FO1B1 = S7(i): FO1B2 = S8(i)
F = (F1 + F2 + (N3 - i) * F0)
Rem КИНЕМАТИКА ТОЛКАТЕЛЯ
Rem КИНЕМАТИКА ТОЧКИ A(1)
15 Call pkt(XO1, YO1, O1B, FO1B, BETTA1, XA(1), YA(1))
Call PA(0, 0, 0, 0, O1B, 0, FO1B, FO1B1, FO1B2, va(1), fva(1), wa(1), fwa(1))
Rem КИНЕМАТИКА ТОЧКИ S1
Call pkt(XO1, YO1, AS1, FO1B, BS1, XS1, YS1)
Call PA(0, 0, 0, 0, AS1, BS1, FO1B, FO1B1, FO1B2, VS1, FVS1, WS1, FWS1)
Rem КИНЕМАТИКА ГРУППЫ 2-3
J = 1
Call PK1(FBC, FO2C, FBC1, FO2C1, FBC2, FO2C2, 0, 0, 0, 0, 0, 0, 0, 0)
Rem КИНЕМАТИКА ТОЧКИ S3
Call PA(0, 0, 0, 0, LS3, BS3, FO2C, FO2C1, FO2C2, vs3, fvs3, WS3, FWS3)
Call pkt(xc(1), yc(1), LS3, FO2C, BS3, XS3, YS3)
Rem КИНЕМАТИКА ТОЧКИ S2
Call PA(va(1), fva(1), wa(1), fwa(1), LS2, BS2, FBC, FBC1, FBC2, VS2, FVS2,
WS2, FWS2)
Rem КИНЕМАТИКА ТОЧКИ D
Call PA(0, 0, 0, 0, O2D, BETTA3, FO2C, FO2C1, FO2C2, vd, fvd, wd, fwd)
Call pkt(xc(1), yc(1), O2D, FO2C, BETTA3, XD, YD)
Rem СИЛОВОЙ РАСЧЕТ МЕХАНИЗМА
Rem ВНЕШНИЕ СИЛЫ ЗВЕНА O2C
qp(1) = PPR: r(1) = O2D: fp(1) = 0: B(1) = BETTA3
Rem ГЛАВНЫЙ ВЕКТОР И МОМЕНТ СИЛ ЗВЕНА O2C
Call PBC(1, FO2C, FO2C2, WS3, FWS3, 0, LS3, BS3, JS3, MS3, PO2C, fpO2C,
mO2C)
Rem ВНЕШНИЕ СИЛЫ ЗВЕНА BC
qp(1) = 0: r(1) = 0: fp(1) = 0: B(1) = 0
Rem ГЛАВНЫЙ ВЕКТОР И МОМЕНТ СИЛ ЗВЕНА BC
Call PBC(1, FBC, FBC2, WS2, FWS2, 0, LS2, BS2, JS2, MS2, PCB, fpBC, mBC)

```

```

Rem РАСЧЕТ ТАНГЕНЦИАЛЬНЫХ РЕАКЦИЙ В ШАРНИРАХ
Rem ГРУППЫ 2-3
rct(1) = FNPRT(mO2C, LS3, L3(1), PO2C, FO2C, fpO2C, 0)
rat(1) = FNPRT(mBC, LS2, L2(1), PCB, FBC, fpBC, 0)
Rem СИЛОВОЙ РАСЧЕТ ГРУППЫ 2-3
Call PC1(rat(1), rct(1), 0, PCB, PO2C, 0, FBC, FO2C, fpBC, fpO2C, rb, RO2, rc, _
frb, FRO2, frc)
Rem СИЛОВОЙ АНАЛИЗ ВХОДНОГО ЗВЕНА
Rem ВНЕШНИЕ СИЛЫ ВХОДНОГО ЗВЕНА
qp(1) = rb: r(1) = O1B: fp(1) = frb + PI: B(1) = 0
Rem ГЛАВНЫЙ ВЕКТОР И МОМЕНТ СИЛ ВХОДНОГО ЗВЕНА
Call PBC(1, FO1B, FO1B2, WS1, FWS1, 0, LS1, 0, JS1, MS1, PO1B,
fpO1B, _mO1B)
Rem РАСЧЕТ УРАВНОВЕШИВАЮЩИХ МОМЕНТА И СИЛЫ
MUR1 = -mO1B - PO1B * LS1 * Sin(fpO1B - FO1B): pur1 = MUR1 / O1B
Rem РАСЧЕТ РЕАКЦИЙ В ОПОРЕ ТОЛКАТЕЛЯ
Rem РАСЧЕТ ПРОЕКЦИЙ РЕАКЦИЙ В ОПОРЕ ТОЛКАТЕЛЯ
RO1X = -pur1 * Cos(FO1B + PI / 2) - PO1B * Cos(fpO1B)
RO1Y = -pur1 * Sin(FO1B + PI / 2) - PO1B * Sin(fpO1B)
Rem РАСЧЕТ ПОЛНОЙ РЕАКЦИИ В ОПОРЕ ТОЛКАТЕЛЯ
Call PMY(RO1X, RO1Y, RO1, FRO1)
Rem ФОРМИРОВАНИЕ МАССИВОВ РЕЗУЛЬТАТОВ РАСЧЕТОВ
FUN1(i) = F: FUN2(i) = FO1B * DEG: FUN3(i) = FBC * DEG
FUN4(i) = FO2C * DEG : FUN5(i) = rb: FUN6(i) = RO1: FUN7(i) = MUR1
FUNGR1(i) = FO2C * DEG : FUNGR2(i) = FO2C1: FUNGR3(i) = FO2C2
FUNGR4(i) = rb: FUNGR5(i) = MUR1
Rem ВЫВОД НА ЭКРАН ТАБЛИЦЫ 1
List1.AddItem Format$(FUN1(i), " 000.0") + " " + Format(FUN2(i), "000.00")_
+ " " + Format(FUN3(i), "0.00") + " " + Format(FUN4(i), "00.00")
502 Next
503 End Sub

Private Sub Command2_Click()
List1.Clear : List1.FontSize = 14 : znak1 = " FK" : znak2 = " RB"
znak3 = " RO1" : znak4 = " Мyp"
znak5 = "===== "
znak6 = " КИНЕМАТИКА МЕХАНИЗМА НА ФАЗЕ " : znak7 = " H "
List1.AddItem Format$(znak6, "") : List1.AddItem Format$(znak5, "")
List1.AddItem Format$(znak1, "") + " " + Format(znak2, "") + " " + _
Format(znak3, "") + " " + Format(znak4, "")
List1.AddItem Format$(znak7, "") + "" + Format(znak7, "") + "" + _
Format(znak7, "") + "" + Format(znak7, "")
List1.AddItem Format$(znak5, "")
If Option2 = True Then GoTo 117
For i = 1 To N1
List1.AddItem Format$(FUN1(i), " 000.0") + " " + _
Format(FUN5(i), "000.00 ") + " " + Format(FUN6(i), "000.00") + " " + _
Format(FUN7(i), " 0.000")
Next

```

```

117 For i = NX To NY Step NS
List1.AddItem Format$(FUN1(i), " 000.0") + " " + _
Format(FUN5(i), "000.00 ") + " " + Format(FUN6(i), "000.00") + " " + _
Format(FUN7(i), " 0.000")
Next
End Sub

```

```

Private Sub Command3_Click()
Picture1.Cls
NazvGR1 = "КИНЕМАТИКА ТОЛКАТЕЛЯ"
Form4.Text1.Text = NazvGR1
Call graf(S3(), Form4, 2, NazvGR1)
End Sub

```

```

Private Sub Command4_Click()
Picture1.Cls
Form4.Cls
NazvGR2 = "КИНЕМАТИКА ВЫХОДНОГО ЗВЕНА"
Form4.Text1.Text = NazvGR2
Call graf(FUNGR1(), Form4, 3, NazvGR2)
End Sub

```

```

Private Sub Command5_Click()
Picture1.Cls
Form4.Cls
NazvGR3 = "РЕАКЦИЯ В ШАРНИРЕ В"
Form4.Text1.Text = NazvGR3
Call graf(FUNGR4(), Form4, 4, NazvGR3)
End Sub

```

```

Private Sub Command6_Click()
Picture1.Cls
Form4.Cls
NazvGR4 = "УРАВНОВЕШИВАЮЩАЯ СИЛА"
Form4.Text1.Text = NazvGR4
Call graf(FUNGR5(), Form4, 5, NazvGR4)
End Sub

```

Теперь имеется перечень переменных, которые необходимо объявить:

- на уровне *Module3*:

```
Public F As Double
```

- на уровне *Form4*:

```

Dim NazvGR1 As String, NazvGR2 As String, NazvGR3 As String,
Dim NazvGR4 As String, FUNGR1() As Double, FUNGR2() As Double
Dim FUNGR3() As Double, FUNGR4() As Double, FUNGR5() As Double

```

Dim FUN1() As Double, FUN2() As Double, FUN3() As Double
 Dim FUN4() As Double, FUN5() As Double, FUN6() As Double
 Dim FUN7() As Double, FUN8() As Double, FO1B As Double
 Dim FO1B1 As Double, FO1B2 As Double, FBC As Double, FO2C As Double
 Dim FBC1 As Double, FO2C1 As Double, FBC2 As Double, FO2C2 As Double
 Dim vs3 As Double, fvs3 As Double, WS3 As Double, FWS3 As Double
 Dim XS3 As Double, YS3 As Double, XS1 As Double, YS1 As Double
 Dim VS2 As Double, FVS2 As Double, WS2 As Double, FWS2 As Double
 Dim VS1 As Double, FVS1 As Double, WS1 As Double, FWS1 As Double
 Dim vd As Double, fvd As Double, wd As Double, fwd As Double
 Dim XD As Double, YD As Double, PO2C As Double, fpO2C As Double
 Dim mO2C As Double, PCB As Double, fpBC As Double, mBC As Double
 Dim PO1B As Double, fpO1B As Double, mO1B As Double, rb As Double
 Dim RO2 As Double, rc As Double, frb As Double, FRO2 As Double
 Dim frc As Double, RO1X As Double, RO1Y As Double, RO1 As Double
 Dim FRO1 As Double, NX As Double, NY As Double, NS As Double

Разработка проекта завершена. После запуска проекта происходит последовательное появление форм отраженных в нем различных данных, как исходных, так и полученных (рис. 4.51–4.67). Задействованные командные кнопки отличаются выделением.

ВВОД ИСХОДНЫХ ДАННЫХ

КИНЕМАТИЧЕСКАЯ СХЕМА МЕХАНИЗМА

Линейные размеры в м, угловые в град., массы в кг, моменты инерции масс в кгм2, сила в Н, скорость кулачка в об/мин.

ШАГ РАСЧЕТА	1	УГОЛ УДАЛЕНИЯ	60	УГОЛ СБЛИЖЕНИЯ	60
ЧИСЛО ГРУПП	1	УГОЛ СТОЯНИЯ ДАЛЬНОГО	240	УГОЛ СТОЯНИЯ БЛИЖНЕГО	0
НАПРАВЛЕНИЕ ВРАЩЕНИЯ КУЛАЧКА		<input type="radio"/> ПО ЧАСАМ		<input checked="" type="radio"/> ПРОТИВ ЧАСОВ	

XO1	0	m1	15.3
YO1	0.175	m2	1.08
XO2	-0.435	m3	31.5
YO2	-0.44	J1	0.146
Rmin	0.12	J2	0.005
Mmax	30	J3	8.44
O1A	0.090	LS1	0.11
O1B	0.220	LS2	0.11
BC	0.200	LS3	0.90
O2C	0.630	BS1	20
O2D	0.770	BS2	0.5
B1	22.5	BS3	10
B3	8.1	Pnc	500
WK	100	OK	

Рис. 4.51. Ввод исходных данных

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ УДАЛЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☒ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☐ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

ФАЗА УДАЛЕНИЯ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

КИНЕМАТИКА ТОЛКАТЕЛЯ НА ФАЗЕ УДАЛЕНИЯ

FK ГРАД	FT ГРАД	FT1 ГРАД	FT2 ГРАД
000,0	000,0	0,0000	0,0000
- 001,0	000,0	0,0027	0,3136
- 002,0	000,0	0,0109	0,6237
- 003,0	000,0	0,0245	0,9271
- 004,0	000,1	0,0432	1,2202
- 005,0	000,1	0,0670	1,5000
- 006,0	000,2	0,0955	1,7634
- 007,0	000,3	0,1284	2,0074
- 008,0	000,5	0,1654	2,2294
- 009,0	000,6	0,2061	2,4271

КООРД. ПРОФ. КУЛ. КООРД. ПРОФ. К-КУЛ. К ФАЗЕ СБЛИЖЕНИЯ
 РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. КУЛ. РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. К-КУЛ.

Рис. 4.52. Результаты расчета кинематики толкателя на фазе удаления

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ УДАЛЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☒ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☐ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

ФАЗА УДАЛЕНИЯ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

КООРДИНАТЫ ПРОФИЛЯ КУЛАЧКА НА ФАЗЕ УДАЛЕНИЯ

FK град	XO м	YO м	RO м	FRO град
000,0	0,10550	0,05718	0,12000	28,5
-001,0	0,10648	0,05533	0,12000	27,5
-002,0	0,10744	0,05347	0,12001	26,5
-003,0	0,10837	0,05161	0,12004	25,5
-004,0	0,10929	0,04975	0,12008	24,5
-005,0	0,11021	0,04790	0,12016	23,5
-006,0	0,11112	0,04605	0,12028	22,5
-007,0	0,11203	0,04423	0,12044	21,5
-008,0	0,11296	0,04242	0,12066	20,6

КООРД. ПРОФ. КУЛ. КООРД. ПРОФ. К-КУЛ. К ФАЗЕ СБЛИЖЕНИЯ
 РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. КУЛ. РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. К-КУЛ.

Рис. 4.53. Координаты профиля кулачка на фазе удаления

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ УДАЛЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☒ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☐ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

ФАЗА УДАЛЕНИЯ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

ПАРАМЕТРЫ РАДИУСА КРИВИЗНЫ И УГЛОВ ДАВЛЕНИЯ КУЛАЧКА НА ФАЗЕ УДАЛЕНИЯ

FK ГРАД	RAD М	FRAD ГРАД	XCK М	YCK М
000,0	0,12000	22,1	0,0000	0,0000
- 001,0	0,15354	22,0	-0,0296	-0,0157
- 002,0	0,21333	21,7	-0,0828	-0,0430
- 003,0	0,35042	21,1	-0,2054	-0,1045
- 004,0	0,98755	20,3	-0,7767	-0,3865
- 005,0	1,20756	19,2	1,1937	0,5811
- 006,0	0,37708	18,0	0,4488	0,2138
- 007,0	0,22566	16,4	0,3134	0,1461

КООРД. ПРОФ. КУЛ. КООРД. ПРОФ. К-КУЛ. К ФАЗЕ СБЛИЖЕНИЯ
 РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. КУЛ. РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. К-КУЛ.

Рис. 4.54. Параметры радиуса кривизны и углов давления кулачка на фазе удаления

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ УДАЛЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☒ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☐ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

ФАЗА УДАЛЕНИЯ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

КООРДИНАТЫ ПРОФИЛЯ КОНТРКУЛАЧКА НА ФАЗЕ УДАЛЕНИЯ

FK град	XO М	YO М	RO М	FROK град
000,0	0,16633	0,00000	0,16633	360,0
- 001,0	0,16630	-0,00290	0,16633	359,0
- 002,0	0,16622	-0,00581	0,16632	358,0
- 003,0	0,16606	-0,00871	0,16629	357,0
- 004,0	0,16583	-0,01161	0,16624	356,0
- 005,0	0,16552	-0,01451	0,16616	355,0
- 006,0	0,16512	-0,01741	0,16603	354,0
- 007,0	0,16461	-0,02029	0,16586	353,0
- 008,0	0,16400	-0,02317	0,16563	352,0

КООРД. ПРОФ. КУЛ. КООРД. ПРОФ. К-КУЛ. К ФАЗЕ СБЛИЖЕНИЯ
 РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. КУЛ. РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. К-КУЛ.

Рис. 4.55. Координаты профиля контркулачка на фазе удаления

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ УДАЛЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☐ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☐ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

ПАРАМЕТРЫ РАДИУСА КРИВИЗНЫ И УГЛОВ ДАВЛЕНИЯ КОНТРКУЛАЧКА НА ФАЗЕ УДАЛЕНИЯ

FK ГРАД	RAD М	FRAD ГРАД	XCK М	YCK М
000,0	0,16633	09,9	0,00000	0,00000
- 001,0	0,14252	10,0	0,02042	-0,01224
- 002,0	0,12491	10,2	0,03534	-0,02160
- 003,0	0,11153	10,6	0,04650	-0,02897
- 004,0	0,10118	11,2	0,05499	-0,03490
- 005,0	0,09305	11,8	0,06152	-0,03974
- 006,0	0,08663	12,6	0,06655	-0,04373
- 007,0	0,08153	13,5	0,07044	-0,04703

КООРД. ПРОФ. КУЛ. КООРД. ПРОФ. К-КУЛ. К ФАЗЕ СБЛИЖЕНИЯ

РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. КУЛ. РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. К-КУЛ.

ФАЗА УДАЛЕНИЯ

Рис. 4.56. Параметры радиуса кривизны и углов давления контркулачка на фазе удаления

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ СБЛИЖЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☐ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☐ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

КИНЕМАТИКА ТОЛКАТЕЛЯ НА ФАЗЕ СБЛИЖЕНИЯ

FK ГРАД	FT ГРАД	FT1 ГРАД	FT2 ГРАД
300,0	030,0	0,0000	0,0000
- 301,0	030,0	-0,0027	-0,3136
- 302,0	030,0	-0,0109	-0,6237
- 303,0	030,0	-0,0245	-0,9271
- 304,0	029,9	-0,0432	-1,2202
- 305,0	029,9	-0,0670	-1,5000
- 306,0	029,8	-0,0955	-1,7634
- 307,0	029,7	-0,1284	-2,0074
- 308,0	029,5	-0,1654	-2,2294
- 309,0	029,4	-0,2061	-2,4271

КООРД. ПРОФ. КУЛ. КООРД. ПРОФ. К-КУЛ. К РЫЧАЖНОЙ СХЕМЕ

РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. КУЛ. РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. К-КУЛ.

ФАЗА СБЛИЖЕНИЯ

Рис. 4.57. Результаты расчета кинематики толкателя на фазе сближения

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ СБЛИЖЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☐ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☒ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

КООРДИНАТЫ ПРОФИЛЯ КУЛАЧКА НА ФАЗЕ СБЛИЖЕНИЯ

	FK град	XO м	YO м	RO м	FRO град
-	300,0	-0,00128	0,16633	0,16633	270,4
-	301,0	0,00162	0,16632	0,16633	89,4
-	302,0	0,00452	0,16626	0,16632	88,4
-	303,0	0,00742	0,16613	0,16629	87,4
-	304,0	0,01030	0,16592	0,16624	86,4
-	305,0	0,01318	0,16563	0,16616	85,5
-	306,0	0,01603	0,16525	0,16603	84,5
-	307,0	0,01887	0,16478	0,16586	83,5
-	308,0	0,02167	0,16421	0,16563	82,5

КООРД. ПРОФ. КУЛ. КООРД. ПРОФ. К.КУЛ. К РЫЧАЖНОЙ СХЕМЕ
 РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. КУЛ. РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. К.КУЛ.

ФАЗА СБЛИЖЕНИЯ

Рис. 4.58. Координаты профиля кулачка на фазе сближения

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ СБЛИЖЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☐ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☒ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

ПАРАМЕТРЫ РАДИУСА КРИВИЗНЫ И УГЛОВ ДАВЛЕНИЯ КУЛАЧКА НА ФАЗЕ СБЛИЖЕНИЯ

	FK ГРАД	RAD М	FRAD ГРАД	XСК М	YСК М
-	300,0	0,16633	-9,89	0,00000	0,00000
-	301,0	0,14249	-9,80	0,00002	0,02384
-	302,0	0,12472	-9,55	0,00040	0,04161
-	303,0	0,11103	-9,12	0,00102	0,05528
-	304,0	0,10023	-8,51	0,00178	0,06606
-	305,0	0,09154	-7,72	0,00264	0,07470
-	306,0	0,08448	-6,76	0,00356	0,08170
-	307,0	0,07869	-5,63	0,00451	0,08741

КООРД. ПРОФ. КУЛ. КООРД. ПРОФ. К.КУЛ. К РЫЧАЖНОЙ СХЕМЕ
 РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. КУЛ. РАДИУС КРИВИЗНЫ И УГЛЫ ДАВЛ. К.КУЛ.

ФАЗА СБЛИЖЕНИЯ

Рис. 4.59. Параметры радиуса кривизны и углов давления кулачка на фазе сближения

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ СБЛИЖЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☐ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☒ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

КООРДИНАТЫ ПРОФИЛЯ КОНТРКУЛАЧКА НА ФАЗЕ СБЛИЖЕНИЯ

	FK град	XO м	YO м	RO м	FROK град
-	300,0	0,05637	0,10594	0,12000	62,0
-	301,0	0,05821	0,10494	0,12000	61,0
-	302,0	0,06004	0,10391	0,12001	60,0
-	303,0	0,06187	0,10286	0,12004	59,0
-	304,0	0,06369	0,10180	0,12008	58,0
-	305,0	0,06553	0,10072	0,12016	57,0
-	306,0	0,06739	0,09963	0,12028	55,9
-	307,0	0,06926	0,09854	0,12044	54,9
-	308,0	0,07116	0,09744	0,12066	53,9

ФАЗА СБЛИЖЕНИЯ

Рис. 4.60. Координаты профиля контркулачка на фазе сближения

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ

ВЫБОР ЗАКОНА ДВИЖЕНИЯ ТОЛКАТЕЛЯ НА ФАЗЕ СБЛИЖЕНИЯ

☐ ЗАКОН ШУНА
 ☐ ЗАКОН СТОДДАРТА

☐ ЗАКОН СИНУСА
 ☐ ДВОЙНОЙ ГАРМОНИЧЕСКИЙ

☒ ЗАКОН КОСИНУСА
 ☐ ЗАКОН 18-Й СТЕПЕНИ

☐ ЗАКОН 7-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА БЕЗ ВЫСТОЯ

☐ ЗАКОН 9-Й СТЕПЕНИ
 ☐ НЕКЛЮТИНА С ВЫСТОЕМ

РЕЗУЛЬТАТЫ РАСЧЕТА КУЛАЧКА

ПАРАМЕТРЫ РАДИУСА КРИВИЗНЫ И УГЛОВ ДАВЛЕНИЯ КОНТРКУЛАЧКА НА ФАЗЕ СБЛИЖЕНИЯ

	FK ГРАД	RAD М	FRAD ГРАД	XCK М	YCK М
-	300,0	0,12000	-22,10	0,00000	0,00000
-	301,0	0,15334	-22,20	-0,02856	-0,01719
-	302,0	0,21033	-22,52	-0,07779	-0,04591
-	303,0	0,32517	-23,03	-0,17755	-0,10279
-	304,0	0,65559	-23,72	-0,46564	-0,26459
-	305,0	7,13522	-24,57	-6,12547	-3,41926
-	306,0	0,95102	-25,54	0,93887	0,51549
-	307,0	0,49264	-26,63	0,53892	0,29154

ФАЗА СБЛИЖЕНИЯ

Рис. 4.61. Параметры радиуса кривизны и углов давления контркулачка на фазе сближения

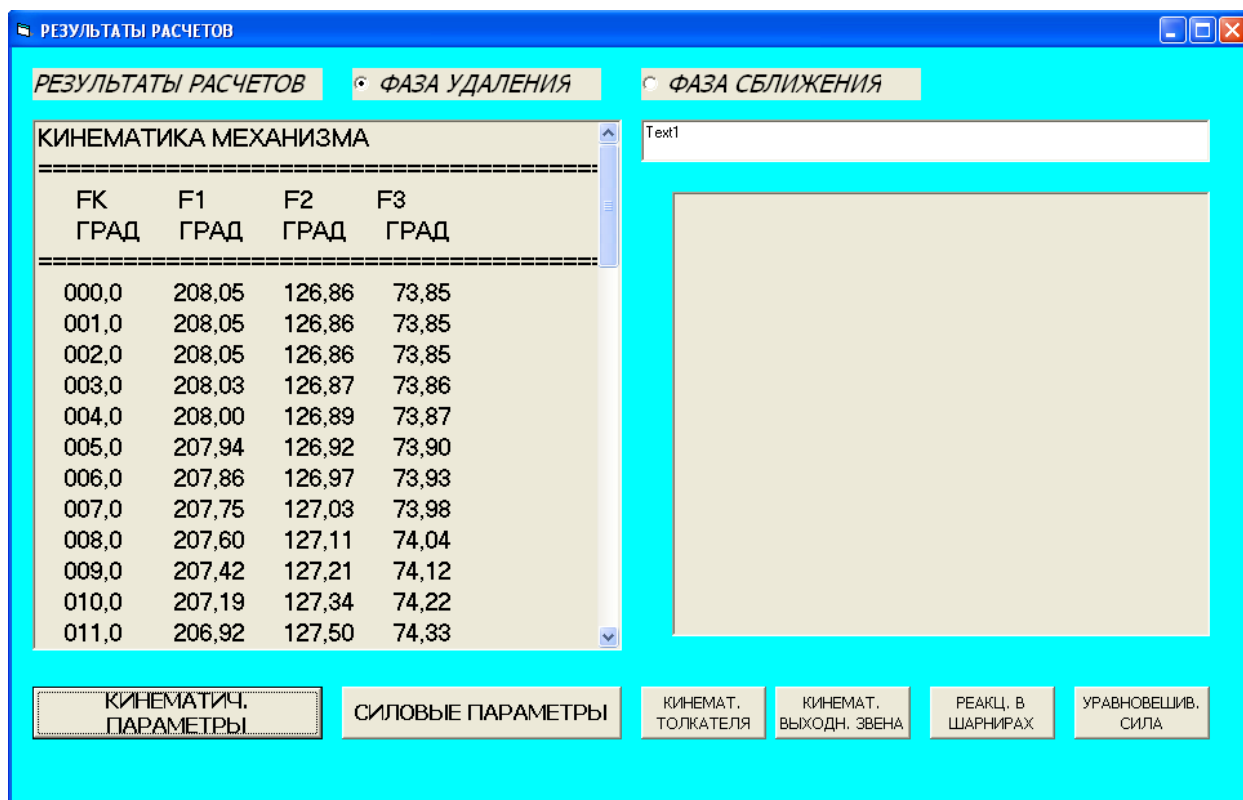


Рис. 4.62. Кинематические параметры механизма

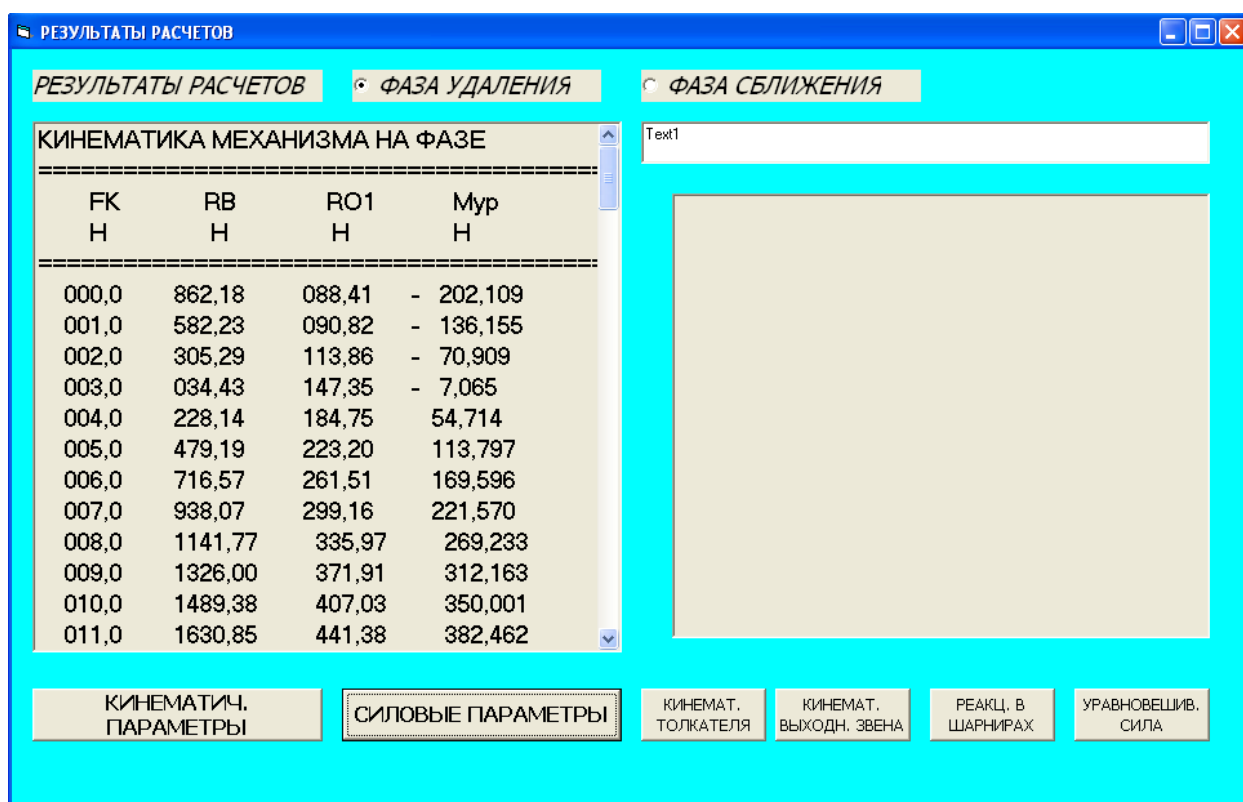


Рис. 4.63. Силовые параметры механизма

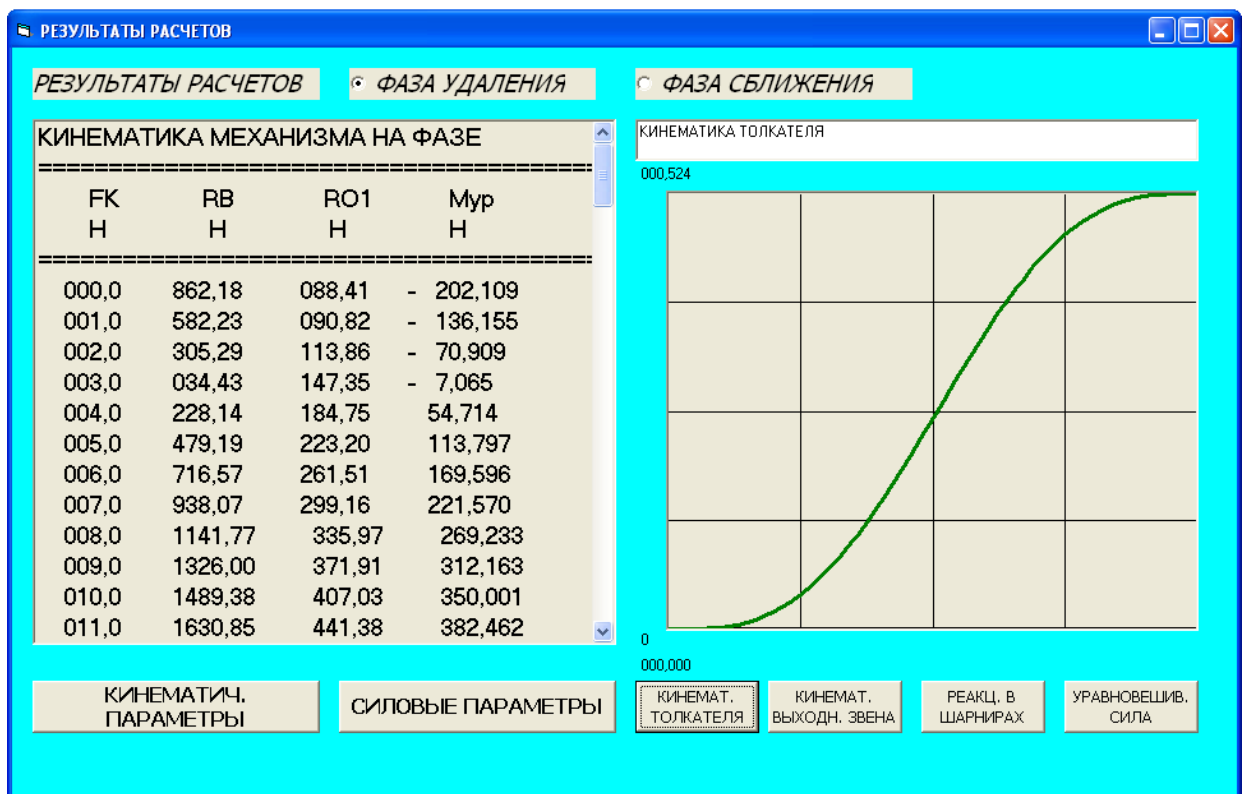


Рис. 4.64. График перемещения толкателя

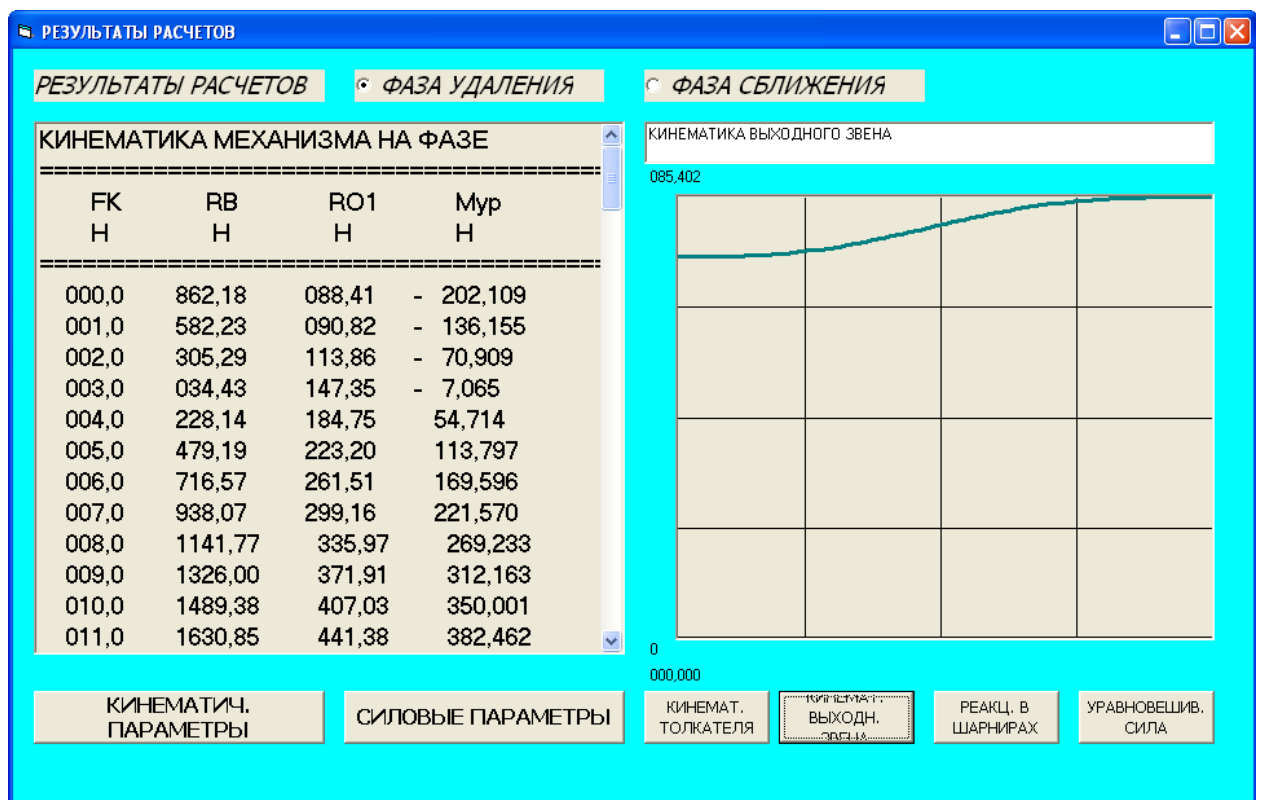


Рис. 4.65. График перемещения выходного звена

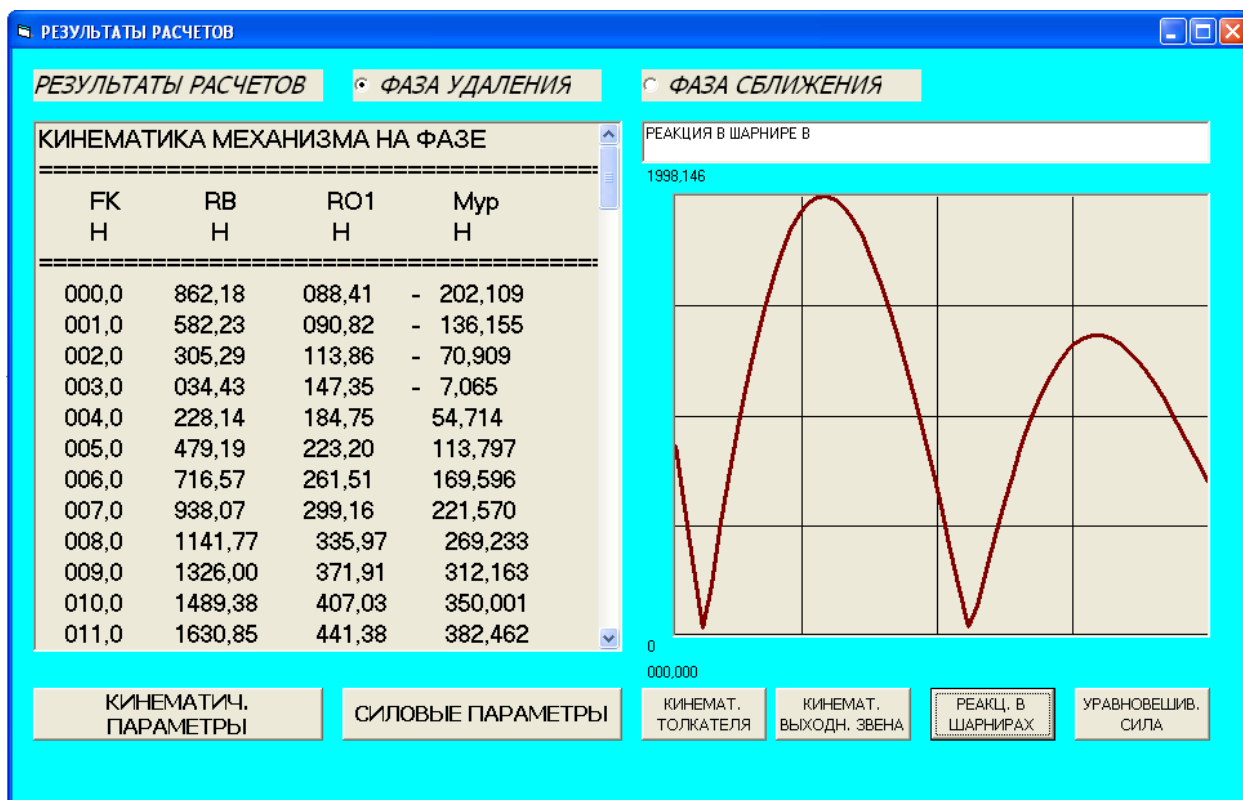


Рис. 4.66. График реакции в шарнире В

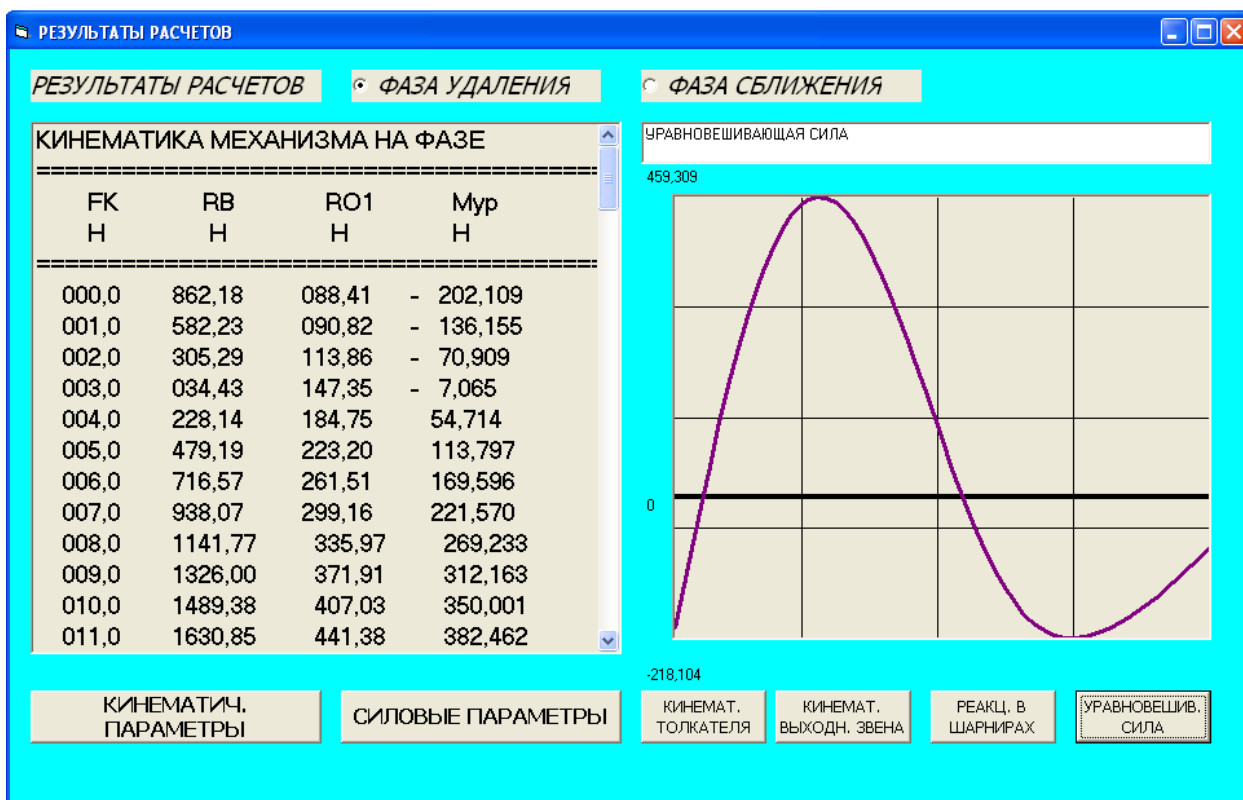


Рис. 4.67. График уравновешивающей силы

По завершении работы с проектом для останова программы необходимо выполнить команды *RUN-END*.

РАЗДЕЛ 5

ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

После изучения теоретических основ кинематического и кинетостатического исследования рычажных и кулачково-рычажных механизмов, ознакомления с примерами выполнения этих расчетов на ЭВМ с использованием языка программирования *Visual Basic* студенты могут самостоятельно работать над проектом.

Перед началом программирования следует продумать структуру проекта, назначение форм и модулей, перечень параметров и последовательность ввода-вывода информации. Особое внимание нужно уделить интерфейсу использующихся форм и видам событий, с помощью которых пользователь будет управлять процессом.

5.1. Содержание задания

Для выполнения самостоятельной работы необходимо получить у преподавателя вариант задания (см. подразд. 5.2).

Целью расчетов могут быть кинематические или кинетостатические параметры работы механизма (по указанию преподавателя).

Отчет по работе должен отражать следующие пункты:

1. Название работы.
2. Цель работы.
3. Схема механизма и исходные данные.
4. Расчетная схема.
5. Блок-схема программы расчета.
6. Распечатка программы.
7. Распечатка результатов расчетов.
8. Выводы.

При определении массово-инерционных характеристик звеньев механизмов используются следующие формулы:

1) для звена в виде стержня:

$$m_i = 15 \cdot L_i, \quad (5.1)$$

где m_i – масса звена, кг;

L_i – длина звена, м;

$$J_{Si} = \frac{m_i \cdot L_i^2}{12}, \quad (5.2)$$

где J_{Si} – момент инерции массы звена относительно оси, проходящей через его центр масс, кг·м²;

$$\left. \begin{aligned} X_{Si} &= 0,5 \cdot L_i, \\ Y_{Si} &= 0, \end{aligned} \right\} \quad (5.3)$$

где X_{Si} , Y_{Si} – соответствующие координаты точки центра масс звена относительно центра шарнира (рис. 5.1, а). (**Примечание:** для треугольного звена центр масс расположен на пересечении медиан (рис. 5.1, б).)

Положение точки центра масс нужно найти самостоятельно из треугольника, для которого известны длины сторон;

2) для звена в виде ползуна (центр масс – в центре ползуна):

$$m_i = 5 \cdot m_1. \quad (5.4)$$

Данные о силе полезного сопротивления нужно получить у преподавателя.

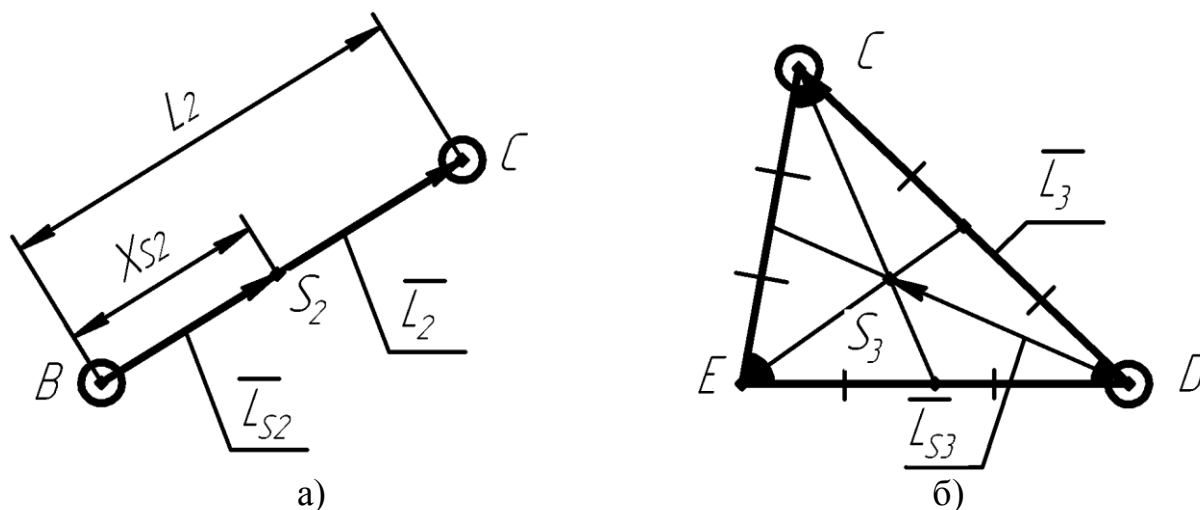
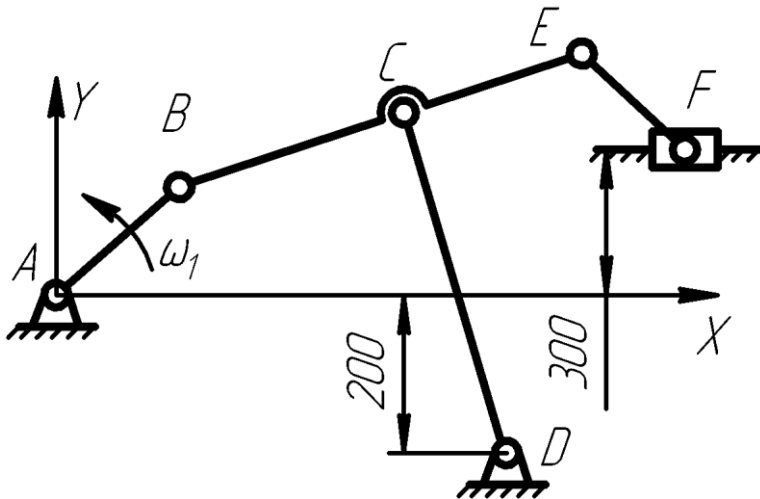


Рис. 5.1. Положение точки центра масс звена

5.2. Варианты задания

Варианты задания для самостоятельной работы содержат кинематическую схему механизма и таблицу числовых значений геометрических параметров механизма. Закон движения входного звена задан в виде угловой скорости его вращения, мин^{-1} .

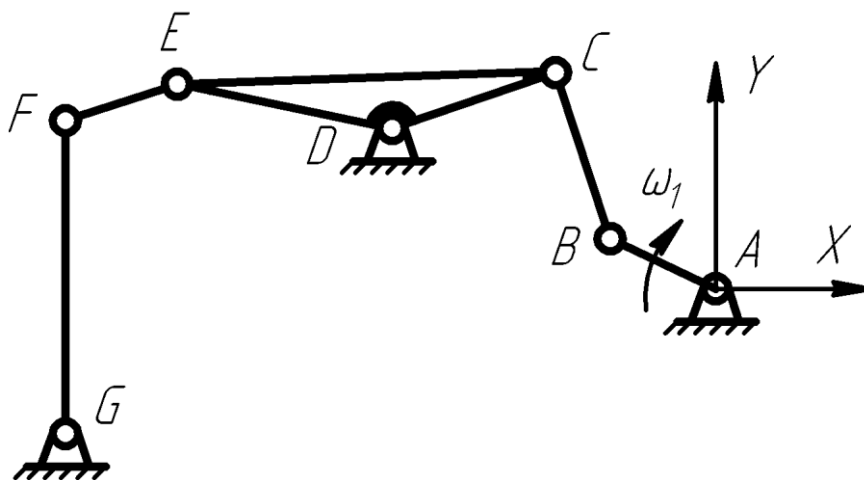
ВАРИАНТ 1



$AB, \text{ м}$	0,2
$BC, \text{ м}$	0,8
$DC, \text{ м}$	0,7
$CE, \text{ м}$	0,3
$EF, \text{ м}$	0,8
$AD, \text{ м}$	0,7
$\omega_1, \text{ мин}^{-1}$	300

Рис. 5.2. Кинематическая схема механизма

ВАРИАНТ 2



$AB, \text{ м}$	0,125
$BC, \text{ м}$	0,2
$DC, \text{ м}$	0,2
$DE, \text{ м}$	0,15
$EC, \text{ м}$	0,3
$EF, \text{ м}$	0,15
$GF, \text{ м}$	0,35
$X_D, \text{ м}$	- 0,3
$Y_D, \text{ м}$	0,2
$X_G, \text{ м}$	- 0,55
$Y_G, \text{ м}$	- 0,1
$\omega_1, \text{ мин}^{-1}$	240

Рис. 5.3. Кинематическая схема механизма

ВАРИАНТ 3

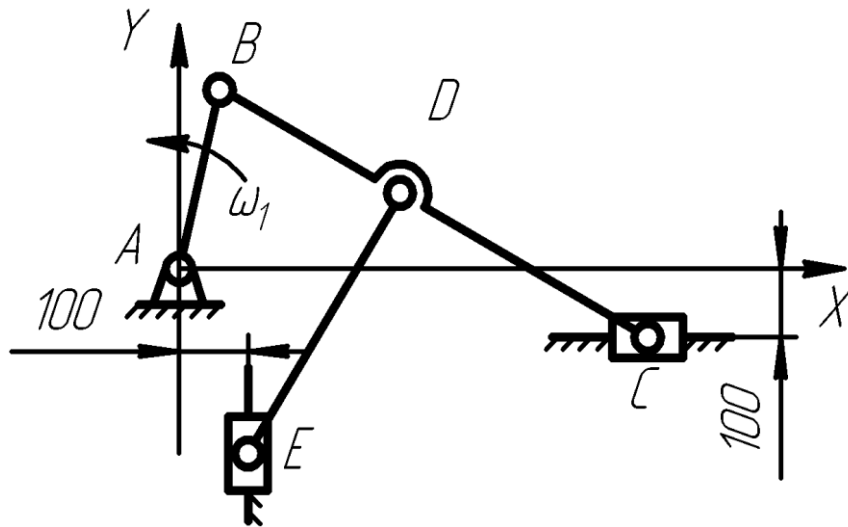


Рис. 5.4. Кинематическая схема механизма

$AB, \text{ м}$	0,3
$BC, \text{ м}$	0,7
$BD, \text{ м}$	0,2
$DE, \text{ м}$	0,7
$\omega_1, \text{ мин}^{-1}$	270

ВАРИАНТ 4

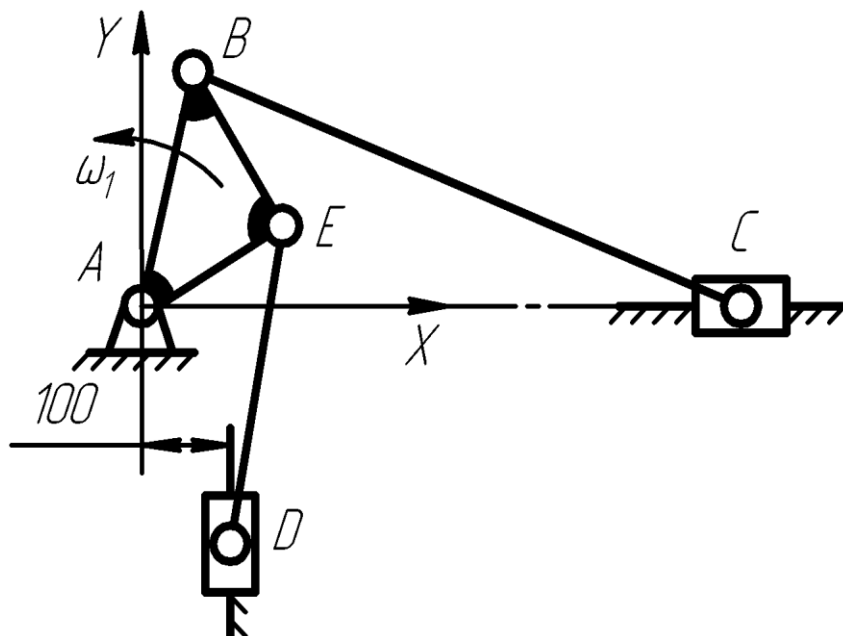


Рис. 5.5. Кинематическая схема механизма

$AB, \text{ м}$	0,4
$AE, \text{ м}$	0,25
$BE, \text{ м}$	0,25
$BC, \text{ м}$	0,8
$ED, \text{ м}$	0,6
$\omega_1, \text{ мин}^{-1}$	180

ВАРИАНТ 5

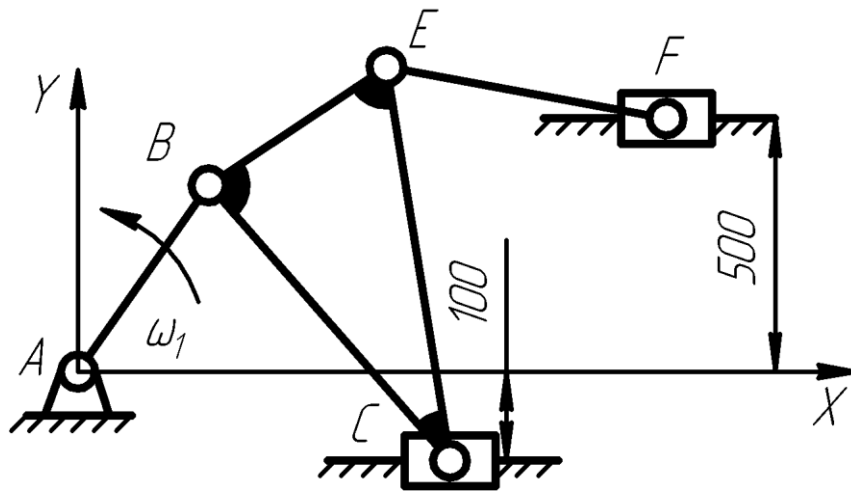


Рис. 5.6. Кинематическая схема механизма

$AB, \text{ м}$	0,3
$BC, \text{ м}$	0,8
$BE, \text{ м}$	0,5
$CE, \text{ м}$	0,8
$EF, \text{ м}$	0,5
$\omega_1, \text{ мин}^{-1}$	180

ВАРИАНТ 6

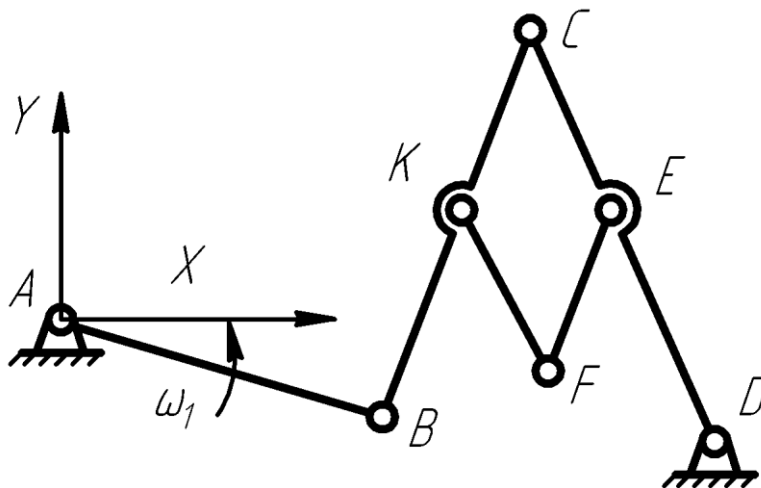
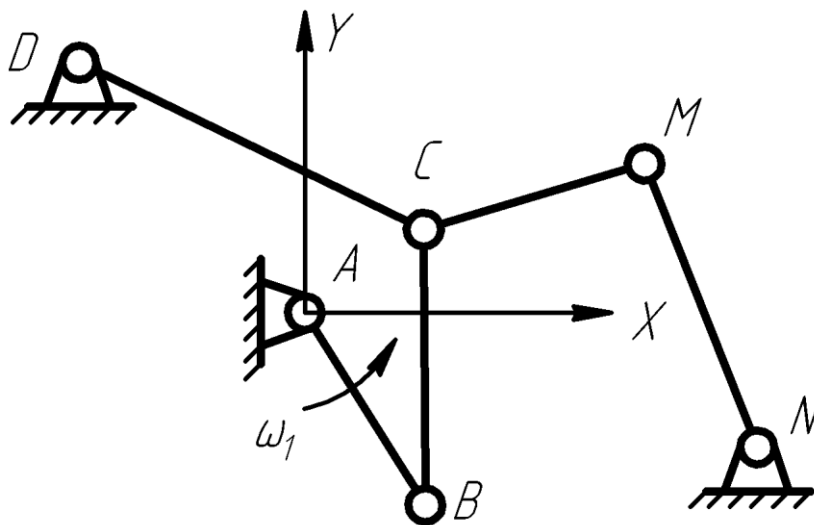


Рис. 5.7. Кинематическая схема механизма

$AB, \text{ м}$	0,25
$BC, \text{ м}$	0,7
$BK, \text{ м}$	0,35
$DC, \text{ м}$	1,0
$CE, \text{ м}$	0,3
$EF, \text{ м}$	0,5
$KF, \text{ м}$	0,5
$X_D, \text{ м}$	0,8
$Y_D, \text{ м}$	- 0,4
$\omega_1, \text{ мин}^{-1}$	240

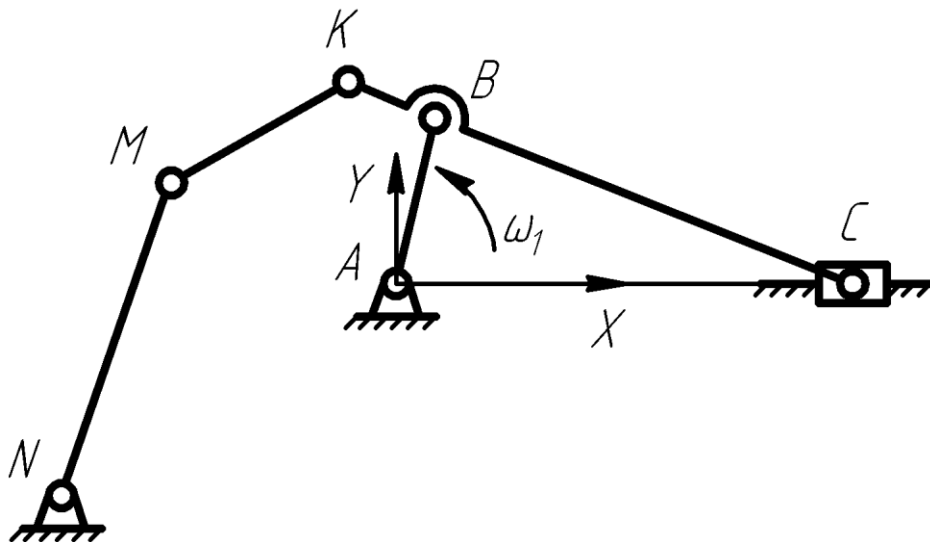
ВАРИАНТ 7



$AB, \text{ м}$	0,3
$BC, \text{ м}$	0,6
$DC, \text{ м}$	0,6
$CM, \text{ м}$	0,4
$NM, \text{ м}$	0,5
$X_D, \text{ м}$	-0,5
$Y_D, \text{ м}$	0,7
$X_N, \text{ м}$	0,58
$Y_N, \text{ м}$	0,14
$\omega_1, \text{ мин}^{-1}$	200

Рис. 5.8. Кинематическая схема механизма

ВАРИАНТ 8



$AB, \text{ м}$	0,35
$BC, \text{ м}$	0,7
$BK, \text{ м}$	0,2
$KM, \text{ м}$	0,4
$NM, \text{ м}$	0,8
$X_N, \text{ м}$	-0,8
$Y_N, \text{ м}$	-0,7
$\omega_1, \text{ мин}^{-1}$	200

Рис. 5.9. Кинематическая схема механизма

ВАРИАНТ 9

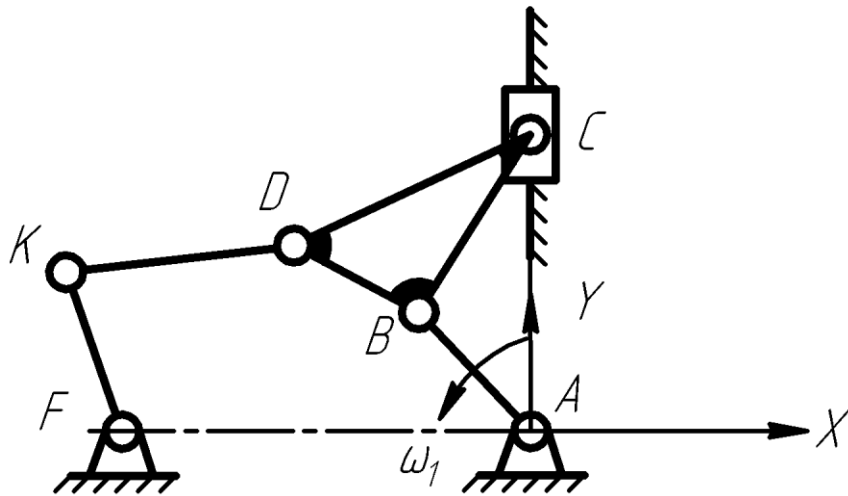


Рис. 5.10. Кинематическая схема механизма

$AB, \text{ м}$	0,3
$BC, \text{ м}$	0,6
$BK, \text{ м}$	0,35
$DC, \text{ м}$	0,6
$BD, \text{ м}$	0,25
$DK, \text{ м}$	0,3
$KF, \text{ м}$	0,4
$X_F, \text{ м}$	-0,6
$Y_F, \text{ м}$	0
$\omega_1, \text{ мин}^{-1}$	210

ВАРИАНТ 10

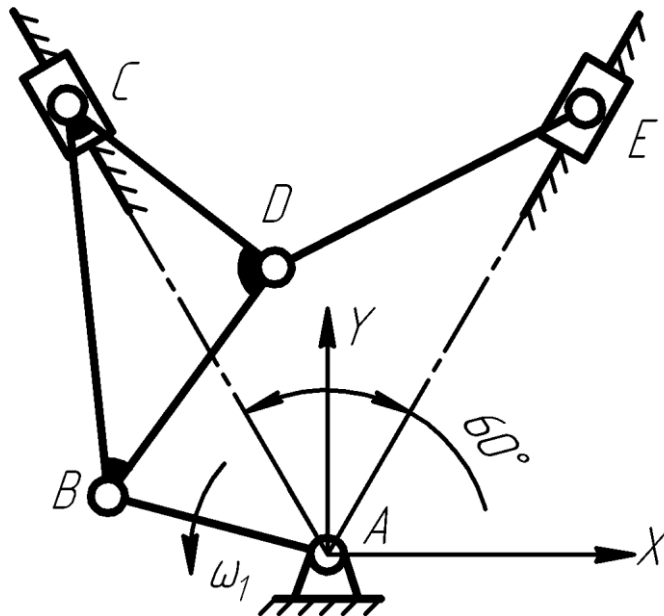


Рис. 5.11. Кинематическая схема механизма

$AB, \text{ м}$	0,75
$BC, \text{ м}$	1,75
$BD, \text{ м}$	0,6
$DC, \text{ м}$	1,5
$DE, \text{ м}$	1,5
$\omega_1, \text{ мин}^{-1}$	800

ВАРИАНТ 11

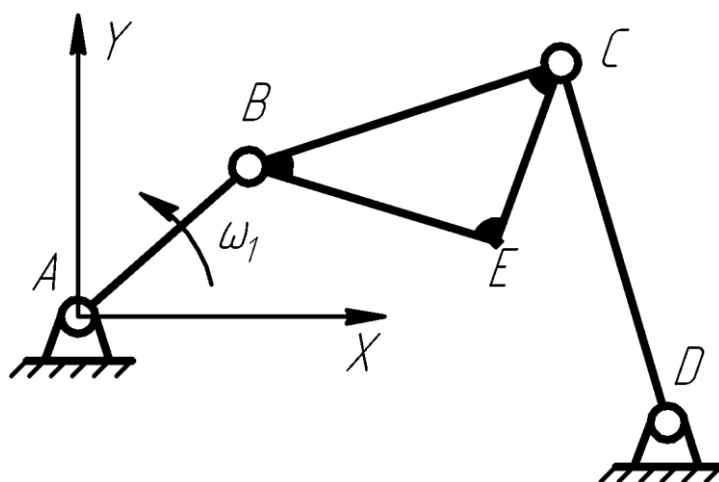


Рис. 5.12. Кинематическая схема механизма

$AB, \text{ м}$	0,2
$BC, \text{ м}$	0,6
$BE, \text{ м}$	0,5
$CE, \text{ м}$	0,5
$DC, \text{ м}$	0,8
$X_D, \text{ м}$	0,8
$Y_D, \text{ м}$	– 0,4
$\omega_1, \text{ мин}^{-1}$	210

ВАРИАНТ 12

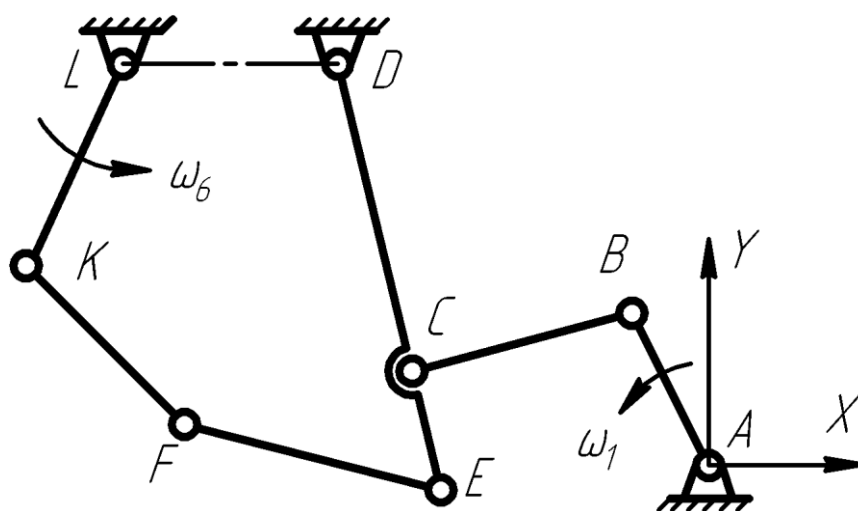


Рис. 5.13. Кинематическая схема механизма

$AB, \text{ м}$	0,5
$BC, \text{ м}$	1,1
$DC, \text{ м}$	1,0
$CE, \text{ м}$	0,5
$FK, \text{ м}$	1,0
$LK, \text{ м}$	0,4
$X_D, \text{ м}$	– 1,25
$Y_D, \text{ м}$	1,0
$X_L, \text{ м}$	– 2,75
$Y_L, \text{ м}$	1,0
$\omega_1, \text{ мин}^{-1}$	120
$\omega_6, \text{ мин}^{-1}$	180

ВАРИАНТ 13

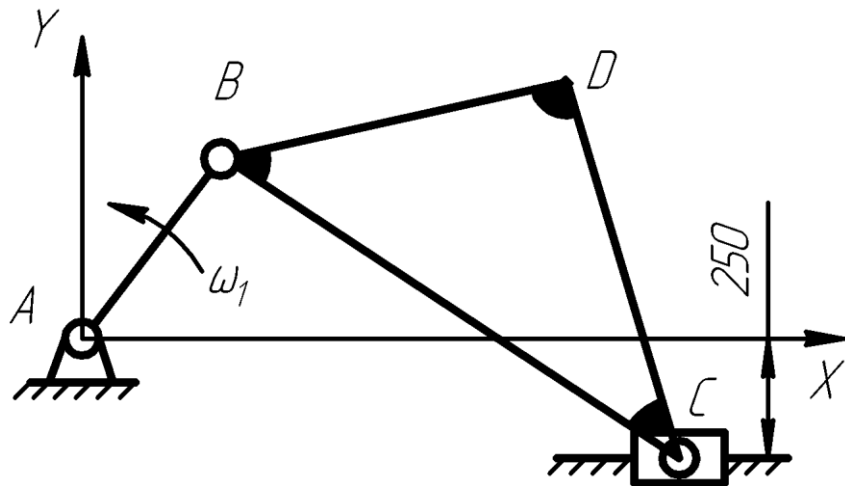


Рис. 5.14. Кинематическая схема механизма

$AB, \text{ м}$	0,5
$BC, \text{ м}$	2,5
$BD, \text{ м}$	1,75
$CD, \text{ м}$	1,25
$\omega_1, \text{ мин}^{-1}$	60

ВАРИАНТ 14

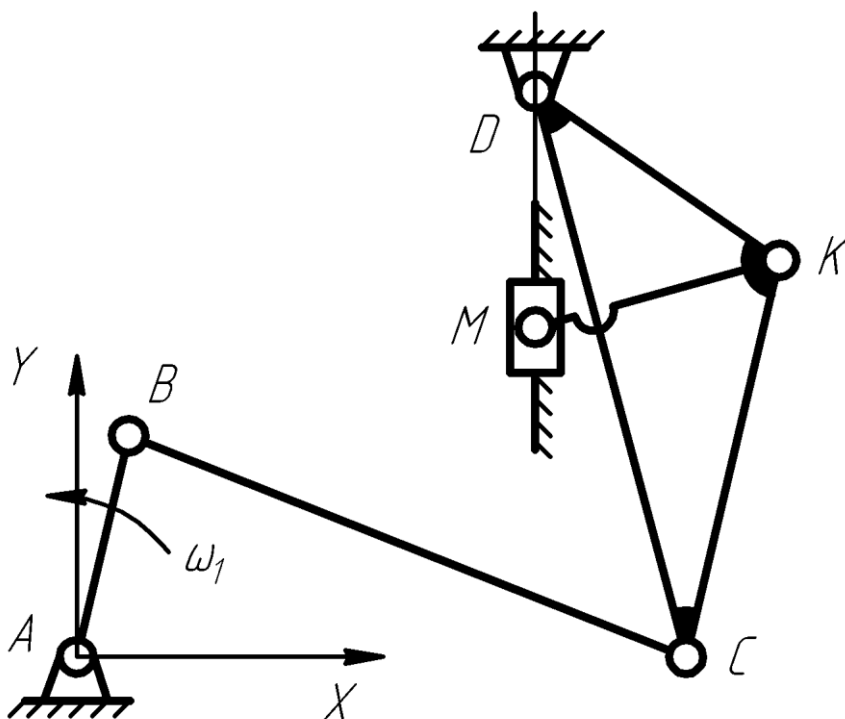


Рис. 5.15. Кинематическая схема механизма

$AB, \text{ м}$	0,4
$BC, \text{ м}$	1,2
$DC, \text{ м}$	1,0
$CK, \text{ м}$	0,8
$DK, \text{ м}$	0,3
$KM, \text{ м}$	0,6
$X_D, \text{ м}$	0,8
$Y_D, \text{ м}$	0,8
$\omega_1, \text{ мин}^{-1}$	260

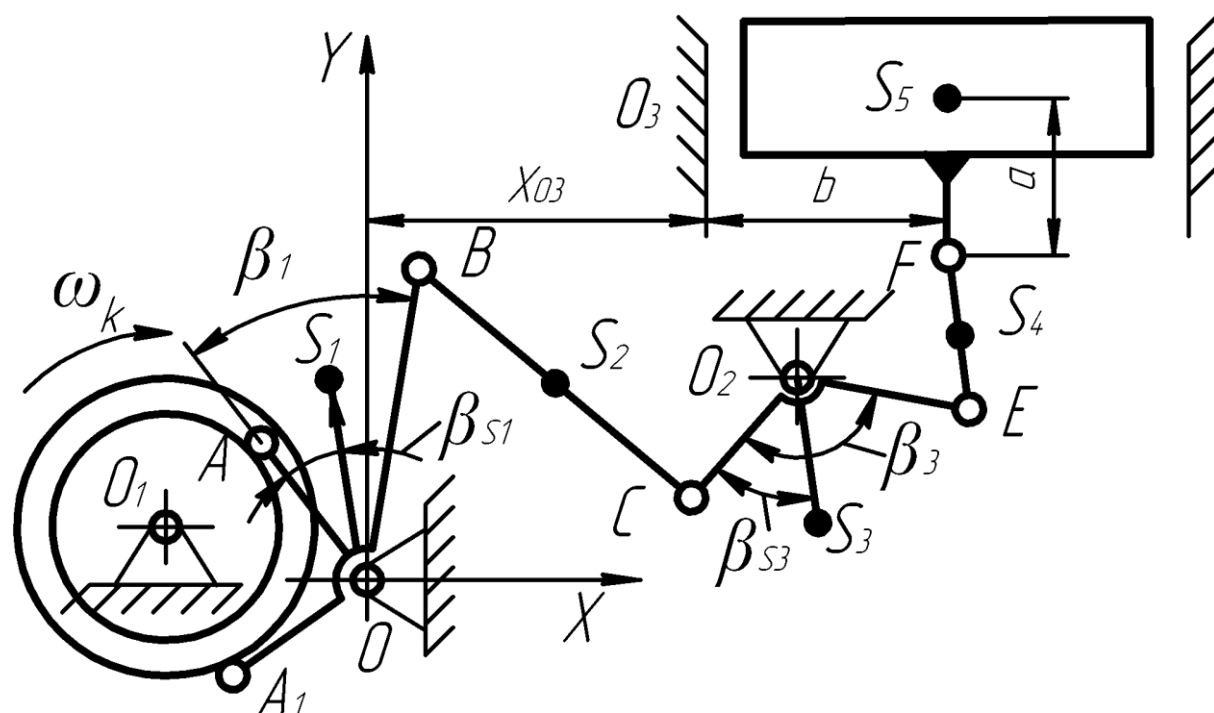


Рис. 5.16. Кинематическая схема механизма

X_{O_1} , мм	Y_{O_1} , мм	X_{O_2} , мм	Y_{O_2} , мм	X_{O_3} , мм	b , мм
-136,88	64,7	818	188	900	133
a , мм	OA , мм	R_{\min} , мм	R_{\max} , мм	OB , мм	OS_1 , мм
62	75	140	170	214	80,7
BC , мм	BS_2 , мм	O_2C , мм	O_2E , мм	O_2S_3 , мм	EF , мм
775	297	150	220	75,7	166
ES_4 , мм	β_1 , град	β_{S_1} , град	β_3 , град	β_{S_3} , град	$\varphi_{y\partial}$, град
80	35,1	25,4	96,2	67,7	125
$\varphi_{cm.d}$, град	φ_{cb} , град	m_1 , кг	m_2 , кг	m_3 , кг	m_4 , кг
55	125	3,15	1,25	1,92	0,25
m_5 , кг	J_{S_1} , кг·м ²	J_{S_2} , кг·м ²	J_{S_3} , кг·м ²	J_{S_4} , кг·м ²	ω_k , мин ⁻¹
6,7	0,0595	0,0796	0,0382	0,0008	200

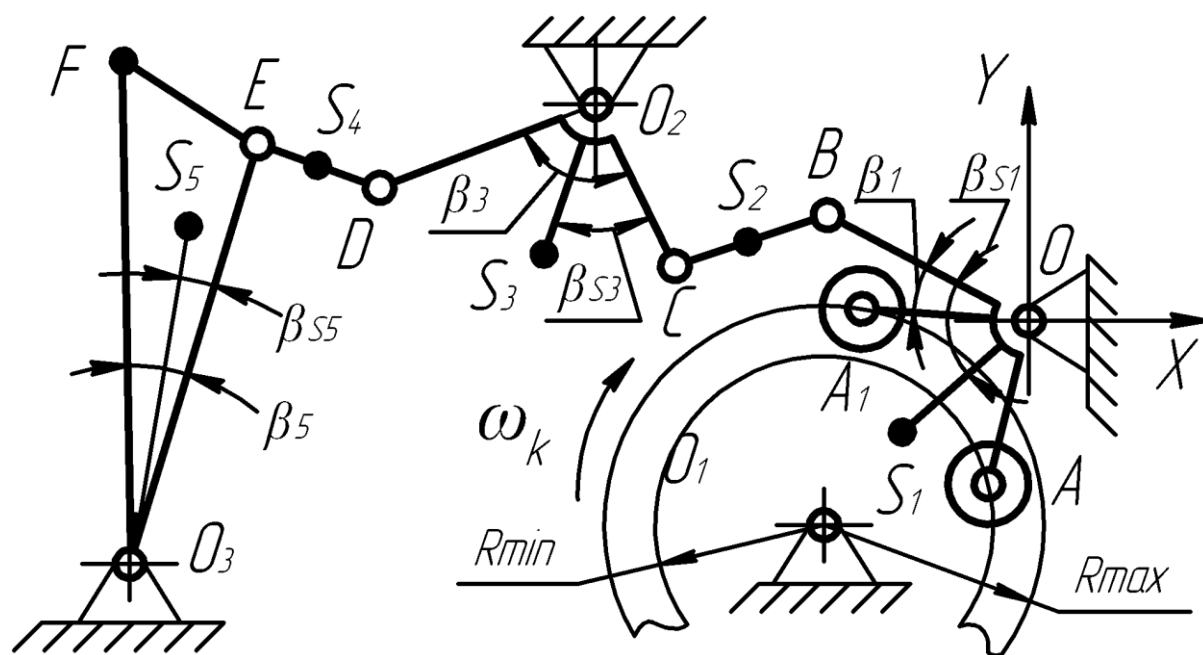


Рис. 5.17. Кинематическая схема механизма

$X_{O_1}, \text{ мм}$	$Y_{O_1}, \text{ мм}$	$X_{O_2}, \text{ мм}$	$Y_{O_2}, \text{ мм}$	$X_{O_3}, \text{ мм}$	$Y_{O_3}, \text{ мм}$
– 111	– 64	– 281	156	– 596	– 474
$R_{\min}, \text{ мм}$	$R_{\max}, \text{ мм}$	$OA, \text{ мм}$	$OB, \text{ мм}$	$OS_1, \text{ мм}$	$BC, \text{ мм}$
90,56	115	60	190	42	160
$BS_2, \text{ мм}$	$O_2C, \text{ мм}$	$O_2D, \text{ мм}$	$O_2S_3, \text{ мм}$	$DE, \text{ мм}$	$DS_4, \text{ мм}$
80	114	145	64,8	145	72,5
$O_3E, \text{ мм}$	$O_3F, \text{ мм}$	$O_3S_5, \text{ мм}$	$\beta_1, \text{ град}$	$\beta_{S_1}, \text{ град}$	$\beta_3, \text{ град}$
630,4	751	582	24	48	81,5
$\beta_{S_3}, \text{ град}$	$\beta_5, \text{ град}$	$\beta_{S_5}, \text{ град}$	$\varphi_{y\partial}, \text{ град}$	$\varphi_{см.\partial}, \text{ град}$	$\varphi_{сб}, \text{ град}$
38	25	12,3	70	0	70
$m_1, \text{ кг}$	$m_2, \text{ кг}$	$m_3, \text{ кг}$	$m_4, \text{ кг}$	$m_5, \text{ кг}$	$J_{S_1}, \text{ кг}\cdot\text{м}^2$
13,2	0,93	3,8	1,45	31,5	0,146
$J_{S_2}, \text{ кг}\cdot\text{м}^2$	$J_{S_3}, \text{ кг}\cdot\text{м}^2$	$J_{S_4}, \text{ кг}\cdot\text{м}^2$	$J_{S_5}, \text{ кг}\cdot\text{м}^2$	$P, \text{ Н}$	$\omega_k, \text{ мин}^{-1}$
0,003	0,02	0,004	8,44	850	115

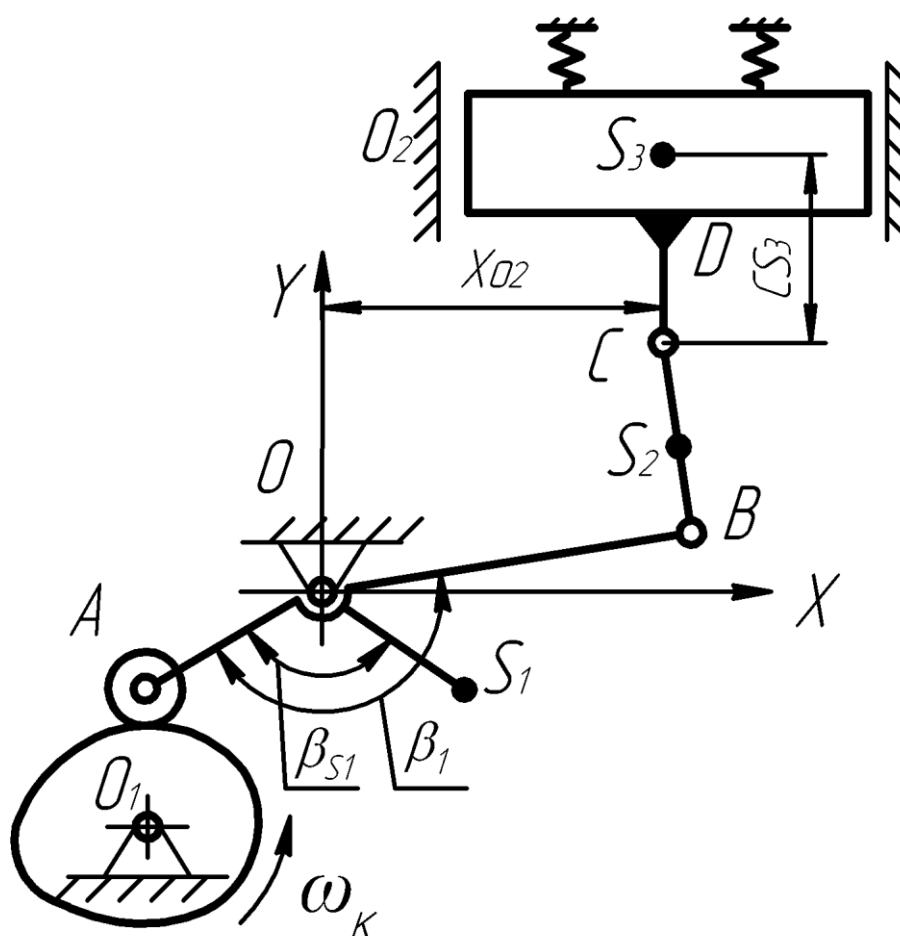


Рис. 5.19. Кинематическая схема механизма

X_{O_1} , мм	Y_{O_1} , мм	X_{O_2} , мм	R_{\min} , мм	R_{\max} , мм	OA , мм
- 1,4	- 83,5	200	65	81	41,27
OB , мм	OS_1 , мм	BC , мм	BS_2 , мм	CD , мм	CS_3 , мм
204	62,17	160	80	10	20
β_1 , град	β_{S_1} , град	$\varphi_{y\partial}$, град	$\varphi_{см.\partial}$, град	$\varphi_{сб}$, град	m_1 , кг
150	67	90	0	90	0,36
m_2 , кг	m_3 , кг	J_{S_1} , кг·м ²	J_{S_2} , кг·м ²	P , Н	ω_k , мин ⁻¹
0,07	0,3	0,285	0,0673	80	250

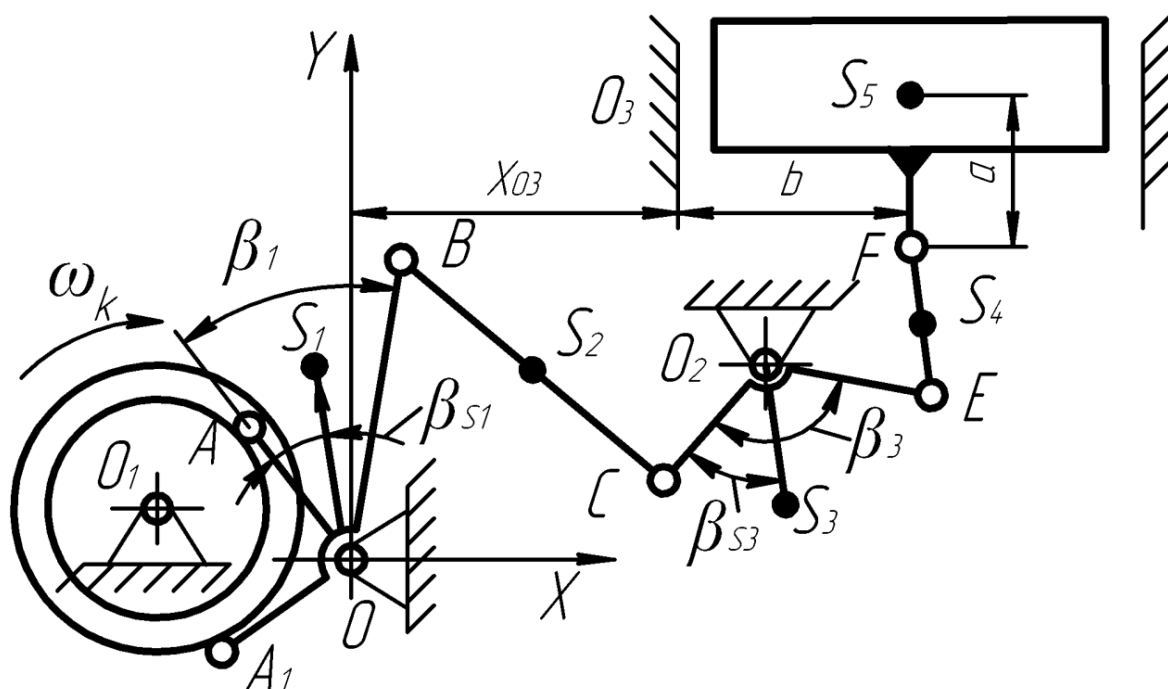


Рис. 5.20. Кинематическая схема механизма

X_{O_1} , мм	Y_{O_1} , мм	X_{O_2} , мм	Y_{O_2} , мм	X_{O_3} , мм	b , мм
-136,88	64,7	303,8	126	1093	500
a , мм	OA , мм	R_{\min} , мм	R_{\max} , мм	OB , мм	OS_1 , мм
45	70	124,5	152,5	228	75,16
BC , мм	BS_2 , мм	O_2C , мм	O_2E , мм	O_2S_3 , мм	EF , мм
260	199	150	225	56,6	346,5
ES_4 , мм	β_1 , град	β_{S_1} , град	β_3 , град	β_{S_3} , град	$\varphi_{y\partial}$, град
173	33,7	40,5	120	21,7	80
$\varphi_{ст.д}$, град	$\varphi_{сб}$, град	m_1 , кг	m_2 , кг	m_3 , кг	m_4 , кг
200	80	3,18	1,36	3,04	1,24
m_5 , кг	J_{S_1} , кг·м ²	J_{S_2} , кг·м ²	J_{S_3} , кг·м ²	J_{S_4} , кг·м ²	ω_k , мин ⁻¹
8,0	0,0394	0,039	0,0322	0,008	50

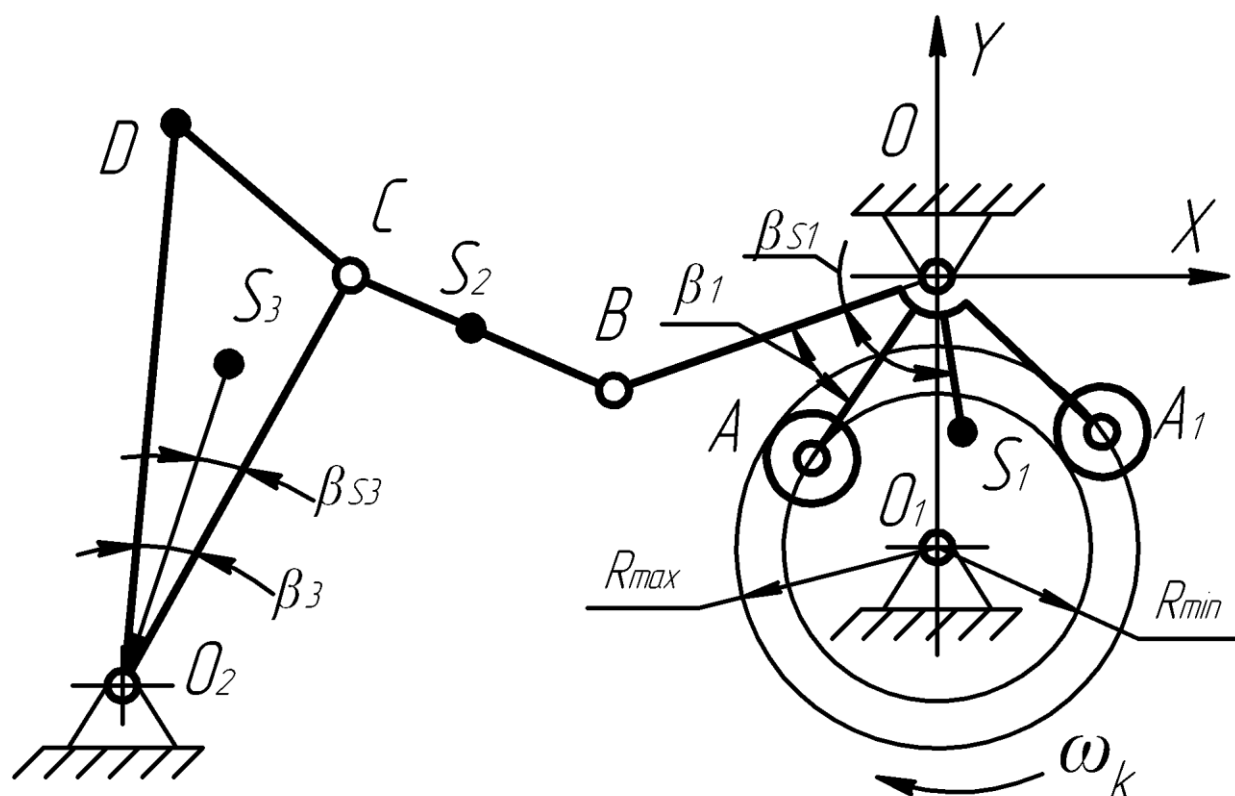


Рис. 5.21. Кинематическая схема механизма

X_{O_1} , мм	Y_{O_1} , мм	X_{O_2} , мм	Y_{O_2} , мм	R_{\min} , мм	R_{\max} , мм
0	-215	-660	-590	154	210
O_1A , мм	O_1B , мм	BC , мм	O_2C , мм	O_2D , мм	O_1S_1 , мм
120	316	200	680	750	64
BS_2 , мм	O_2S_3 , мм	β_1 , град	β_{S_1} , град	β_3 , град	β_{S_3} , град
100	738	26,86	34,2	8,6	8,2
$\varphi_{y\partial}$, град	$\varphi_{см.\partial}$, град	$\varphi_{сб}$, град	$\varphi_{см.б}$, град	m_1 , кг	m_2 , кг
75	0	75	210	17,6	1,9
m_3 , кг	J_{S_1} , кг·м ²	J_{S_2} , кг·м ²	J_{S_3} , кг·м ²	P , Н	ω_k , мин ⁻¹
40,8	0,181	0,012	9,56	850	90

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1. Зиновьев, В. А.** Курс теории механизмов и машин : учеб. / В. А. Зиновьев. – М.: Наука, 1972.
- 2. Морошкин, Ю. Ф.** / Ю. Ф. Морошкин // Доклады Академии наук СССР. – 1952. – Т. 82. – № 4.
- 3. Лебедев, П. А.** Новые методы анализа и синтеза плоских и пространственных механизмов и машин текстильной и легкой промышленности: дис. ... д-ра техн. наук / П. А. Лебедев. – Л.: ЛИТЛП, 1967.
- 4. Ильин, В. А.** Аналитическая геометрия: учеб. для вузов / В. А. Ильин, Э. Г. Позняк. – 7-е изд. стереот. – М.: Физматлит, 2004. – 204 с.
- 5. Теория** механизмов и машин: учеб. для вузов / К. В. Фролов [и др.]; под ред. К. В. Фролова. – 2-е изд., перераб. и доп. – М.: Высш. шк., 2008. – 496 с.: ил.
- 6. Левитский, Н. И.** Кулачковые механизмы : учеб. для вузов / Н. И. Левитский. – М.: Машиностроение, 1964. – 288 с.
- 6. Теория** механизмов и машин. Терминология: учеб. пособие / Н. И. Левитский [и др.]; под ред. К. Ф. Фролова. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2007. – 80 с.
- 7. Пискунов, Н. С.** Дифференциальное и интегральное исчисления: в 2 т.: учеб. / Н. С. Пискунов. – Т. 1. – СПб.: Мифрил. Гл. ред. физ.-мат. лит. 1996. – 416 с.
- 8. Браун, С.** VISUAL BASIC 6: учеб. курс / С. Браун; пер. Е. Матвеев. – СПб.: Питер, 2001.
- 9. Васильев, П. П.** Встроенные функции языка программирования VISUAL BASIC 6.0 / П. П. Васильев. – М.: Диалог-МИФИ, 2000. – 160 с.
- 10. Лукин, С. Н.** VISUAL BASIC 6.0. Самоучитель для начинающих / С. Н. Лукин. – М.: Диалог-МИФИ, 2001. – 544 с.