MATEMATIČKI INSTITUT

Savremena računska tehnika i njena primena Knjiga 4

Ю. А. Степаненко

ДИНАМИКА ПРОСТРАНСТВЕННЫХ МЕХАНИЗМОВ

Savremena računska tehnika nalazi sve širu primenu u različitim oblastima čovečije delatnosti. Ovo za sobom povlači potrebu upoznavanja širokog kruga ljudi sa načinom rada i mogućnostima računara, kao i sa matematičkim metodama za rešavanje složenih zadataka pomoću računara. Za ovo je neophodno postojanje odgovarajuće literature. I serija Matematičkog instituta

SAVREMENA RAČUNSKA TEHNIKA I NJENA PRIMENA

ima osnovni zadatak da na potrebnom teorijskom i praktičnom nivou upozna čitaoca sa dostignućima u ovoj oblasti.

Ova publikacija nije periodična.

Rukopise opremljene za štampu slati na adresu: Matematički institut, 11000 Beograd, Knez Mihailova 35.

Redakcioni odbor - Comité de rédaction

Glavni urednik — Rédacteur en chef: Nedeljko Parezanović Sekretar — Secrétaire: Boško Jovanović

Članovi odbora — Membres du comité: Mirko Stojaković, Slaviša Prešić i Pavle Pejović

Tehnički urednik: Milan Čavčić

Izdaje: Matematički institut, 11000 Beograd, Knez Mihailova 35

MATEMATIČKI INSTITUT - INSTITUT »MIHAILO PUPIN«

Savremena računska tehnika i njena primena Knjiga 4

Ю. А. Степаненко

ДИНАМИКА ПРОСТРАНСТВЕННЫХ МЕХАНИЗМОВ

Republička zajednica za naučni rad SR Srbije i INSTITUT »MIHAILO PUPIN« učestvovali su u troškovima izdavanja ove publikacije.

Prema mišljenju Republičkog sekretarijata za kulturu SR Srbije ova publikacija je oslobođena poreza na promet.

ПРЕДИСЛОВИЕ

Многозвенные пространственные механизмы находят применение в различных областях техники. Они служат основой, например, универсальных манипуляторов, шагающих машин,механизмов рехабилитации и т. п. Часто такие механизмы работают в напряженном динамическом режиме, в связи с чем важное значение приобретает анализ их динамики. Однако в настоящее время можно назвать лишь несколько научных статей посвященных этому вопросу.

Такое положение объясняется серьезными трудностями , проистекающими из большого числа звеньев и степеней свободы. Уже сейчас известны примери механизмов с десятью и более степенями свободы, и можно предположить, что в ближайшем будущем появятся еще более сложные системы. Для их анализа необходимы специальные методы учитывающие особе тности объекта.

Сама структура пространственных механизмов подсказывает путь их исследования. Поскольку механизм представляет собой цепочки последовательно соединненых звеньев, естественно в основу метода анализа положить некоторую "рекуррентность", позволяющую последовательно двигаться вдоль цепей от одного звена к другому.

Такой метод обладает важными преимуществами. Во-первых, реккурентсоотношения легко поддаются алгоритмизации, и здесь эффективно могут быть применены вычислительные машины. Во-вторых, используя некоторые специальные приемы, можно осуществить машинное составление дифференциальных уравнений движения на цифровых машинах. Последнее очень

важно, поскольку эти уравнения имеют исколючительно громоздкий вид. Вывод уравнениы, занимающих десятки (если не сотни) страниц, перестает быть чисто технической задачей и эдесь колличество явно переходит в отрицательное качество.

В-третьих, такой подход поэволяет построить универ - сальный алгоритм для моделирования пространственных мехачизмов на цифровых вычислительных машинах. Такой алгоритм получает информацию в некотором цифровом коде о структуре и параметрах данного конкретного механизма и характере решаемой задачи. По этой информации он автоматически:

- 1) "собирает модель" данного конкретного механизма;
- 2) осуществляет анализ кинематики, статики и динамики, в частности "составляет" и интегрирует дифференциальные уравнения движения.

Нетрудно видеть какие возможности таит в себе такое моделирование. Достаточно сказать что любые "конструкти - вные" изменения механизма (в том числе изменения в кине - матической схеме) здесь достигаются с максимальной про - стотой--нужно только заменить несколько чисел во входной информации.

Настоящая книга посвящена разработке указанного выше метода. При этом рассматривается кинематика, статика и динамика механизмов и разрабатывается универсальный модетирующий алгоритм. Таким образом, содержание книги не сколько шире ее названия, однако в заголовок вынесена толь ко динамика, как наиболее важная часть.

При оценке тех или иных методов анализа решающим, по мнению автора, является критерий практики.Поэтому в книгу приводится значительное число вычислительных алгоритмов, которые можно практически использовать для решения различ ных задач статики, кинематики и динамики пространственных механизмов. Отладка и проверка алгоритмов осуществлялась

в Государственном научно-исседовательском институте машиноведения /г. Москва/. Коялеги по Лаборатории теории управления машинами и Лаборатории програмирования оказивали помощ в этой работе, и автор выражает им благодарность, равно как и руководству Института, поддерживащему эти исследования.

Книга построена таким образом, чтобы представлять интерес для специалистов различного профиля. Поэтому, изложение ведется как-бы на двух уровнях: идейном и алгоритмическом. Основные идеи и формулы даются без привлечения алгоритмических языков, вследствии чего для их понимания не требуется знание программирования вообще. Алгоритмический уровень включает в себя блок-схемы с ком-ментариями и тексты алгоритмов на языке АЛГОЛ.

Автор благодарит проф.А.Е.Кобринского,проф.В"И.Сергеева, и проф.М.Вукобратовича,оказавших ему существенную помощ,а также проф.Р.Томовича и проф.Н.Парезановича,взявших на себя труд рецензирования рукописи.

Книга была написана в апреле-сентябре 1971 г. во время работы в Лаборатории автоматики Института автоматики и телесвязы им. "Михаило Пупина"-одном из известных центров по изучению биокибернетичесних систем, г. Белград, СФР Югославия. Автор считает своим приятным долгом поблагодарить руководство и коллектив Лаборатории за внимание и помощь в этой работе.

Автор признателен Математическому институту в Белграде и Институту автоматики и телесвязы им."М.Пупина" совместно осуществившим издание книги.

Основные обозначеня

В работе используются трехмерные и многомерные векторы. Первые будем обозначать жирным шрифтом, например, е, г. Многомерные векторы обозначены буквами обычного шрифта и смысл обозначений объясняется в тексте. Все векторы предполагаются столбцами.

Матрицы обозначаются большими буквами, например, A, B . Номпоненты матриц указываются верхними индексами, например, A^{ij} , B^{jk} . Для обозначения матриц будет использоваться также следующая запись

$$\|A^{ij}\|_{n,m}$$
; $\|B^{ij}\|_{p,k}$

Иногда вместо одного из верхних индексов будет указываться значок \star . Это означает, что соответствующий индекс "пробегает" все свои значения. Таким способом будет обозначаться строки и столбцы матриц, например, $\mathbf{A}^{i\star}$ и $\mathbf{A}^{\star i}$. Будут использоваться также следующие обозначения

$$A = \|A^{*1}, A^{*2}, \dots, A^{*n}\|$$
; $A = \|A^{1*}\|_{A^{2*}}$

$$\vdots$$

$$A^{n*}$$

Множества также обозначаются большими буквами, а злементы множеств - соответствующими малыми (например X, х^і или Y, y^i). Если Y есть подмножество X, и некоторый элемент x^i входит в Y, это будет записыватся как

xi EY

Знак := обозначает "присваивание"; он используется в блок-схемах. Запись

a:=b

означает, что переменной а присваивается значение пере-

Следующие обозначения сохранятся на протяжении всей книги:

е; - единичный вектор оси кинематической пары;

гіј — вектор с началом в "центре" ј-го шарнира и концом в центре масс і-го звена;

 p^{i} — реакция /сила или момент/, развиваемые приводом i-го шарнира;

G_i,M_i — внешние сила и момент, действующие на i-е звено /приведены к центру масс/;

m_; - масса і-го звена;

I_{i1},I_{i2},I_{i3} - моменты инерции i-го звена относительно главных ценральных осей;

q_{il},q_{i2},q_{i3} — единичные векторы, направленые вдоль главных центральных осей;

 Λ , Λ_1 , Λ_2 — матрицы связи для замкнутой кинематической цепи;

N1,N2 — номера начальной и конечной базовой цепи;

nl,n2 - базовые звенья в цепях Nl и N2;

 $ξ1_{i}, ξ2_{i}ξ3, ξ4_{ij}, ξ5_{i}, ξ6_{i}, ξ7_{ij}$ - характерические числа, см. гл. I;

Nc — общее число кинематических цепей механизма.

ВВЕДЕНИЕ

В настоящем Введении приводятся необходимые сведения из теории механизмов, даются основные определения и обсуждаются особенности принятого метода анализа.

Основным в структурном анализе механизмов является понятие кинематической пары. Под кинематической парой понимается два звена механизма, соединенные друг с другом с помощью некоторого устройства, допускающего их относительное перемещение. Способы соединения звеньев в кинематиче ские пары могут быть разнообразвыми. Простейшими примерами могут служить цилиндрический шарнир или прямолинейные направляющие.

В зависимости от способа соединения звеньев на их относительное движение накладываются определенные ограни - чения или связи. Эти связи могут отбирать различное число степеней свободы в относительном движении тел, и по этому признаку кинематические пары делятся на пять классов.Кинематические пары различных классов представлены в таблице I, взятой из [1] *. Кинематические пары пятого класса оставляют одну степень свободы в относительном движении звеньев пары первого класса - пять.

Кроме деления на классы кинематические пары подразделяются также на виды. Пары первого вида допускают макси мальное число вращательных относительных движений (в парах I-III класса допускается три вращения относительно перпендикулярных осей, в парах IV класса - два, в парах V класса

^{*} Библиография приводится в конце Введения.

- одно). Пары второго вида по сравнению с парами первого вида накладывают ограничение на одно из вращательных движений, пары третьего вида накладывают ограничения на два вращательных движения (см. таблицу 1).

Класс пар	Число усло- вий связи	Нисло степе- ней свободы	Виды пар		
			I BUB	П вид	Ш вид
I	1	5	Nucre houseous from Inet Annythiasus 3 2 Asparavemus 0 1 I 1		
	.2	4	Departurement 0 8	Angekassis 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Ш	3	3		Nean Basternik Brown Pari Annychranus 2 1 Depanvenus 1 2 A 1 A 1	Huses basiltenial from the Rengchassis 1 2 Papamuvermas 2 1
<u>IV</u>	4	2		Approximation 1 / Depression 2 / 2	
V	5	1	Vasse Brantered Brancher Januckaums I B Bepanwennus 2 J	Naces deuxerned Speed Naces Allegen assus: 0 / Daysanu vennus: 3 2	Muse deskende (ferentier) Jangskoemses 0 0 Japanuvennus 2 2 8

Ta6. 1

Помимо пар, у которых допускаемые относительные движения звеньев являются независимыми, существуют пары со

связанными движенями. Простейшим примером может служить винтовая пара, у которой поступательное и вращательное относительное перемещение звеньев связаны линейной зависимостью.

На кинематических схемах механизмов приняты следую - щие обозначения для различных кинематических пар(рис.В.1.)

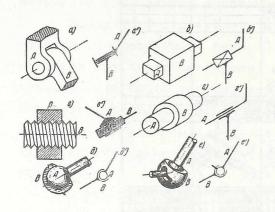


Рис. В. 1

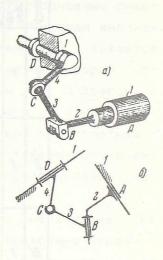


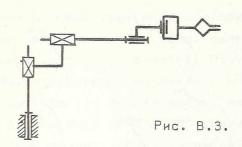
Рис. В.2

Из кинематических пар составляются кинематические це пи. На рис. В.2 показан пример пространственного механизма рулевого управления [1]. Он состоит из четырех кинема тических пар, причем каждое звено входит только в две кинематическия пары. Одно звено механизма (на рис. звено 1) принято за неподвижное. Неподвижное звено обычно называется стойкой.

Цепь показанная на рис. В. 2 является простой замкнутой цепью. В замкнутой цепи каждое звено входит в две кинематические пары.

Другой пример пространственного механизма показан на рис. В.З. Это механизм универсального манипулятора с пятью

степенями свободы [2]. Цепь состоит из шести кинематиче - ских пар, однако, последнее звено в отличии от предыдущего примера, входит только в одну кинематическую пару.Это пример простой разомкнутей кинематической цепи.



В рассмотренных примерах каждое звено входило не более чем в две кинематические пары. Однако, известны механизмы, в которых одно звено может входить в большее число пар. На рис. В.4 показана кинематическая схема робота [3] (экзоскелетон). Модель содержит двенадцать подвижных звеньев, соединенных в кинематические пары III кл. I вида . Здесь звенья 4 и 8, представляющие корпус робота, входят каждое в три кинематические пары.

Цепи, в которых какие-либо звенья входят в более чем две кинематические пары называют сложными цепями.

В данной книге условимся о следующей терминологии . Под кинематической цепью всегда будем понимать простую цепь, разомкнутую или замкнутую. Сложные цепи будем рассма тривать, как состоящие из простых, соединенных друг с дру гом, и в этом случае будем говорить о взаимосвязанных цепях или о механизме со взаимосвязанными цепями.

При такой точке эрения замкнутые цепи не обязательно должны соединятся крайними звеньями с неподвижной стойкой, а могут присоединяться к звеньям других цепей. Аналогично для разомкнутых цепей. Например, для механизма представле-

нного на рис. В.4 одну кинематическую цепь образуют звенья (1-7), а две другие разомкнутые цепи образуются "руками" и "туловищем" (8-10 и 11-12);вэтом случае присоединены к подвижным звеньям других цепей.

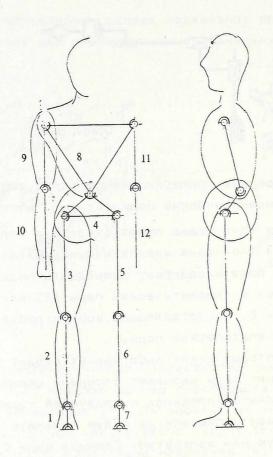


Рис. В.4

Вопрос о разбивнии кинематической схемы механизма на цепи рассматривается в шестой главе книги, посвященной анализу взаимосвязанных цепей. В четырех предыдущих главах рассматриваются механизмы, состоящие из одной цепи, разом-кнутой или замкнутой.

При анализе динамики столь дложных систем важное значение имеет выбор соответствующего метода. Поскольку исследование предполагается проводить с помощью цифровых вычи слительных машин (ЭЦВМ), то метод должен выбираться с учетом их специфики.

Нак указывалось выше, чрезвычанно сложной и грамоздкой операцией для простриственных цепей является составление дифференциальных уравнений движетия. Поэтому важно разработать алгоритмы, позволяющие составлять эти уравнения на ЭЦВМ. Для того, чтобы это было возможно, метод составления уравнений не должен быть необходимо связян с дифференцированием каких-либо аналитических выражений, поскольку проводить численное дифференцирование на ЭЦВМ крайне нежелательно.

По этой причине в книге не используются уравнения Лагранжа второго рода, а составление уравнений осуществляется с помощью кинетостатического подхода [4].

При использовании этого подхода задача естественно распадается на три этапа. Первый - анализ кинематики, который включает в себя расчет положений, линейных скоростей и ускорений центров масс, а также угловых скоростий и ус - корений всех звеньев механизма. Второй - решает задачу перехода от угловых и линейных ускорений звеньев к главному вектору и главному моменту сил инерции. Наконец, третий этап - анализ статики, при котором решается задача равно - весия механизма под действием внешних сил и сил инерции.

В изложенном плане есть одно неясное место. Для то - го, чтобы определить линейные и угловые ускорения звеньев, нужно знать относительные ускорения в каждом шарнире. Однако, приступая к расчету мы их не знаем, и как раз их определение является нашей задачей. В этом случае можно воспользоваться тем, что уравнения механики линейны относи - тельно вторых производных. Тогда можно задать векторы от-

носительных ускорений с точностью до постоянного множителя и для этих ускорений рассчитать состветствующие силы и моменты сил инерции.

Согласно принципа Даламбера миханиэм должен нахо - ходиться в равновесии под дейстием заданных сил и сил инер ции. Отсюда можно получить достаточно условий для опреде - ления неизвестных постоянных множителей и затем истинных значений относительных ускорений.

Таким образом, составление уравнений осуществляется в следующей последовательности (более подробно каждый из этапов разбирается в следующих главах):

- 1. Для данного момента времени t₁ по известным относительным перемещениям звеньев в каждом шарнире(эти перемещения принимаются в качестве обобщенных координат х) рассчитываются положения осей всех звеньев и осей шарниров в абсолютной системе координат. В программу заложено соответствующее уравнение, позволяющее определять проекции вектора при его вращении вокруг некоторой оси.
- 2. Выполняется кинематический анализ с целью определить угловые ускорения и ускорения центров масс всех звеньев механизма. Для этого в програму заложены соответствующие векторные формулы, позволяющие определить искомые ускорения, не прибегая к дифференцированию уравнений положения.
- 3. Рассчитываются силы и моменты сил инерции всех эвеньев механиэма. Необходимые для этого линейные и угло вые ускорения были определены на втором этапе. В программу заложены соответствующие уравнения динамики.
- 4. Составляются уравнения кинестатического равнове сия механизма согласно принципа Даламбера. Для этого на каждом звене определяется сумма внешних сил и моментов развиваемых приводами, и сил и моментов сил инерции. Суммар-

ные векторы приводятся к центрамм масс. Последовательно"ра эрывается" базовая кинематическая цепь в каждом из шарниров и составляются условия равновесия относительно оси разорванного шарнира. В программу заложен алгоритм для при ведения всех сил и моментов к соответствующей оси.

5. В результате выполнения пункта 4 получается систе ма линейных алгебраических уравнений относительно неизвестных значения ускорений.

На этом заканчивается составление уравнений. Теперь, считая найденные ускорения постоянными в течение малого отрезка времени Δt , можно осуществить интегрирование уравнений и определить значение координат x и скоростей \dot{x} в можент времени $t_2=t_1+\Delta t$. Эти координаты и скорости можно рассматривать как начальные условия для можента времени t_2 . Используя эти начальные условия, можно вновь провести составление дифференциальных уравнений для момента времени t_2 , т.е. вернуться к п. І вышеприведенного описания и т.д.

В заключение несколько слов о приводимых в книге алгоритмах. Наждая из рассматриваемых задач завершается разработной соответствующего алгоритма, который записывается на языке АЛГОЛ-60. Этими текстами можно пользоваться при решении соответствующих задач, однако, их все же не следует рассматривать, как программы для вычислений. Автор не ставил своей задачей создать справочник или библиотеку программ, и АЛГОЛ используется только потому, что позволяет строго и компактно записать алгоритмы решения задач.

Такая точка зрения приводила, например, к тому, что если стоял выбор между ясностью и простотой записи с одной стороны и требуемым объмом рабочей памяти с другой, то пред почтение всегда отдавалось первому. При этом автор руко - водствовался известным высказыванием Немминга [5], что це-

лью расчетов является понимание, а не число, и считал, что это высказывание еще в большей степени относится к форме представления алгоритмов. В результате требуется сравни тельно большой объем памяти (хотя вообщем вполне доступный для исследования даже сложных механизмов на современных универсальных ЭЦВМ). Изменив последовательность расчетов и объединив некоторые процедуры и соответствующие циклы можно кое-что выгодать в объеме памяти, даже не изменяя сколь ко-нибудь значительно вычисления. И , разумеется, можно существенно снизить объем памяти, если пожертвовать скоростью счета и промежуточные результаты те хранить в памяти, а каждый раз вычислять. Все эти вопросы должны решаться зависимости от конкретной задачи и конкретной машины.Соответствующие изменения (если они требуются) носят техниче ский характер, не затрагивают существа работы алгоритма и поэтому здесь обсуждаться подробно не будут.

При записи алгоритмов предполагается, что в библиотеке стандартных программ имеются программы для осуществления основных векторных и матричных операций. Обращения к этим программам условно записываются в виде обращения к следующим процедурам.

1. СКАЛ (a,b,c)
Процедура вычисляет скалярное произвёдение векторов,

 $a = b \cdot c$;

2. ВЕҢТ (a,b,c)
Процедура вычисляет векторное произведение векторов,

 $a = b \times c$;

3. МТВ (a,B,c,n,m) - вычисляет произведение матрицы $n \times m$ на вектор C ,

4. МТМТ (A,B,C,n,m,p) - вычисляет произведение матрицы В размерности $\mathbf{n} \times \mathbf{m}$ на матрицу С размерности $\mathbf{m} \times \mathbf{p}$,

$$A = BC$$
;

- 5. ОБР (A,B) вычисляет $A=B^{-1}$, где A и B матрицы
- 6. НОРМ (a,b) осуществляет нормализацию вектора b ,

$$a = b/[b]$$
;

7. МОД (b) - действительная процедура, которая переменной присваивает значение модуля вектора b,

$$a: = MOД (b)$$
;

8. АЛГ УР (c,A,b,n) - процедура решает матричное уравтение Ac=b и определяет компоненти n - мерного век - тора c - матрица A - квадратная $n \times n$.

Предполагая, что в этих процедурах целые числа n,m, p, указывающие размерности матриц и векторов, вызываются эначениями. То-есть, для умножения матрицы 3×3 на вектор можно записать эбращение к процедуре как МТВ (a,B,c,3).

Библиография и комментарий

- 1. Артоболевский И.И. Теория механизмов и машин. Москва 1953.
- Spizig J.S. Vandelbare Automaten zum Ein-Aus-und Weiter geben Werkstatts'technik, (57), 1967, Heft 10.
 - Содержит описание промишленных роботов типа "Versatran".
- 3. Vukobratovich M. et all. Development of active exoske leton, Proceeding IV IFAC Symposium on Automatic Cont-

rol, Dubrovnik, 1971.

Работа содержит описание метода синтеза двуногой походки и экспериментального двуногого макета, созданного в Институте автоматики и телекоммуникации им. "Михайлы Пупина" (г. Белград).

4. Первая работа по применению метода кинетостатики к анализу пространственных механизмов принадлежит Н.Г. Бруевичу (Труды ВВА им. Жуковского вып. 36,37, 1937 г.).Однако, в дальнейшем этот подход, равно как и вообще динамика пространственных механизмов, надолго выпали из поля эрения исследователей.

В последнее время интерес к этой теме возник вновь. Здесь следует отметить пять работ. На VI съезде по теории машин и механизмов (Ленинград, 1970) были доложены работы Ю.Ф.Мороштина и А.Г.Овакимова. Они содержали аналитические методы исследования динамики пространствен ных механизмов. Позднее работы А.Г.Овакимова опублико ваны в журнале "Машиноведение" (№ 2,6, 1970 г.); в них предлагается использовать уравнения Лагранжа II рода и введенные автором аналоги скоростей и ускорений.

В работе Уикера (Труды американского общества инже - неров-механиков, серия Конструирование и технология ма-шиностроения №1, 1969 г. "Динамика пространственных механизмов") также используются уравнения Лагранжа и рассматривается пример пространственного механизма с одной степенью свободы. Весьма показательна в этом смысле работа Нана (Kahn M. The near-minimum time control open -loop articulated kinematic chains. AIF-106, School of Humanities and Science, Stanford, December, 10, 1969.). Автор также пользуется уравнениями Лангранжа и приводит пример исследования динамики манипулятора с тремя степенями свободы. Кан приводит полные формулы дифферен -

циальных уравнений движения этого манипулатора, которые заняли около 16 журнальных страниц, набранных петитом. Легко представить как возрастает сложность этих уравнений при переходе к моделям с большим числом степеней свободы.

Особый интерес представляет работа выполненная Вукобратовичем и его коллегами (Vukobratovich et all. Mathematical modelling of anthropomorphic systems., Progress
Report, No 1, SRS (Social Rehab. Serv., 1969). Во-первых потому что здесь рассматривается сложная модель двуногой локомоционной системы с большим числом степенеы
свободы. Во вторых в работе используется кинетостатический подход. В третьих, в работе сделан значительный
шаг по пути формализации исследования с помощью цифровых машин (описание структуры в цифровом коде, рекуррен
тные соотношения и пр.).

5. Hamming R.W. Numerical methods, Mc Graw-Hill, 1962.

Гл. І ДИНАМИЧЕСКИЕ МОДЕЛИ

§1 Модели кинематических пар

Как уже указывалось, кинематические пары могу быть различного класса и вида. Эта классификация отражает тип и количество различных перемещений, допускаемых парой. Простейшими и наиболее часто встречающимися на пракзике, являются пары V класса - поступательные и вращательные. Как будет показано ниже, комбинируя кинематические цепочки с этими простейшими парами можно "набирать" или моделировать различные кинематические пары других классов.

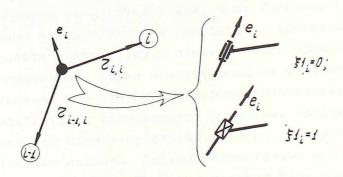


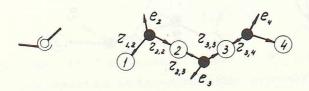
Рис 1.1

Условимся изображать на рисунках пару V класса черным кружком (рис. 1.1) и будем ставить ей в соответствив

некоторое характеристическое число. В дальнейшем понадоб - ятся различные характеристические числа, поэтому условимся, что будем их изображать двумя знаками: буквой ξ и справа от нее какой-либо цифрой. По этой цифре будем различать ха рактеристические числа. Тип пары будем указывать с помощью числа $\xi 1_{i}$

Единичный вектор оси пары обозначим е_і. Кинематические пары такого вида для сокращения в дальнейщем будем называть просто шарнирами. Некоторую точку внутри шарнира будем называть центром шарнира и из этого центра проведем радиусы-векторы к центрам масс двух соседних звеньев, связанных этим шарниром. На рисунке такие вектори обозначены

где перный индекс указывает номер звена, а второй - номер шарнира. Центры масс будем изображать условно кружкам, внутри которых будем записывать номер звена. Кинематические пары, показанные на рис. 1.1 является теми простейшими "монтажными единицами", из которых можно набирать различные кинематические пары других классов. Пакажем как это деляется.



Допустим нужно промоделировать кинематическуя пару, показанную на рис 1.2. Тогда строим цепочку, состоящую из трех последовательно расположенных вращательвых шарнинов, связанных стержнями 2 и 3. Если теперь положить, что все оси шарниров перпендикулярны, а длины звеньев 2 д 3 ратвны нулю,

$$\xi l_2 = \xi l_3 = \xi_4 = 0$$
;
 $e_3 \perp e_2$; $e_4 \perp e_3$; $e_4 \perp e_2$;
 $r_{22} = r_{23} = r_{33} = r_{34} = 0$.

то мы получим модель соответствующую сферической кинематической паре.

Пусть, например, имеется цилиндрическая пара IV класса, которая допускает одновременно вращение и поступательное перемещение. Ее можно промоделировать цепочкой, содержащей два шарнира, один из которых поступательный, а другой - вращательный. Оси шарниров параллельны, а длина стержиня 2 равна нулю (рис. 1.3).

$$\xi 1_2 = 1$$
; $\xi 1_3 = 0$;
 $e_2 \perp e_3$; $r_{22} = r_{23} = 0$.

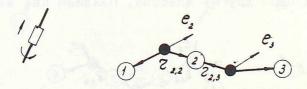


Рис 1.3

Аналогичним образом можно представить все кинематические пары I - V классов, I - II вида. Особый случай представляют кинематические пары, у которых связаны некоторые движения, например, винтовые кинематические пары. Они также могут быть промоделированы с помощью цепочки звеньев с простыми шарнирами, однако, прежде чем показать это, необ ходимо рассмотреть особенности днижения в шарнирах.

Для осуществления относительного перемещения, шарниры часто снабжаются специальными приводами. Мы будем разли чать два типа привода: кинематический и силовой. В первом типе привода обеспечивается заранее заданный закон относительного движения. Силовой привод реализует не закон движения, а некоторые силы или моменты, которые воздействуют на звенья кинематической пары.

Для того, чтобы не оговаривать каждый раз какой шарнир поступательный имеется в виду, мы будем везде говорить о реакции привода, подразумевая, что в случае поступательной пары - это сила, а вращательной - момент. Реакцию сило-

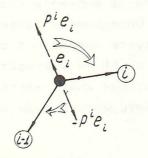


Рис 1.4

вого привода і-го шарнира обозначим Р^і. Будем считать , что реакция привода коллинеарны оси кинематической пары . Привод воздействует на соседние эвенья і-ой кинематиче - ской пары равными и противоположно направленными реакциями, так что /см. рис. 1.4/.

воздействие на i-и звено -
$$p^ie_i$$
 (1.2) воздействие на (i-l)-е звено - $-p^ie_i$

Введем ещё одно характеристическое число $\xi 2_{j}$, которому будем приписывать следующие значения, в зависимости от типа привона;

Теперь вернемся к вопросу о моделировании кинематических пар со связаными движениями, например, винтовой кинематической пары. Винтовую пару можно моделировать также, как цилиндрическую, показанную на рис 1.3, однако, следует считать, что привод одного из шарниров /например, 3/ осуществляется кинематически. Определение закона относительного движения $\mathbf{x}^3(\mathbf{t})$ в этом случае необходимо осуществлять исходя из того, как изменяется $\mathbf{x}^2(\mathbf{t})$ и учитывать при этом кинематическую зависимость, которая имеет место в винтовой паре. Аналогичный прием может быть использован и в случаях других кинематических пар.

§2 Модели кинематических цепей

Нак указывалось выше, кинематические цепи могут быть разомкнутые и замкнутые, и соответствующие примеры показа-

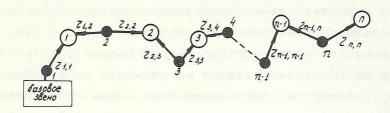
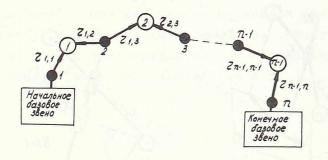


Рис 1.5



Puc 1.6

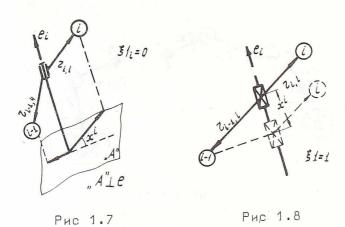
ны на рис. 1.5 и 1.6. Тип цепи будем характеризовать целым числом $\xi 3$

Примером разомкнутой цепи может служить механизм манипулятора, а замкнутоь - механизм на рис. В.2 (см. Введение).То

звено механизма, к которому присоединена кинематическая цепь, будем называть базовым звеном для данной цепи, а ту цепь, которой принадлежит базовое звено - базовой цепью.

Всв шарниры и звенья цепи пронумеруем последовательно, начиная от базового звена; базовому звену, от которого начинается номерация, присвоим нулевой номер. Для замкну тых цепей в соответствии с порядком номерации будем различать начальное и конечное базовые звенья (рис. 1.6.).

В качестве координат цепи будем рассматривать отно сительные перемещения в каждом из шарниров и обозначим их $\mathbf{x^i}$, где \mathbf{i} -номер кинематической пары. Ниже $\mathbf{x^i}$ будем называть относительными координатами. На рис. 1.7 и 1.8 по-



казано, как определяется $\mathbf{x^i}$ в случае вращательной и поступательной пар.

Для вращательной парй $\mathbf{x}^{\mathbf{i}}$ определяется, как угол между проекциями векторов

двух соседних эвеньев на плоскость перпендикулярную оси

шарнира e_i . Положительным считается направление отсчета, которое для наблюдателя смотрящего из конца e_i кажется происходящим против часовой стрелки. В случае поступательной пары перемещение \mathbf{x}^i будем отсчитывать от перпендикуляра из центра масс звена, несущего направляющие на ось пары (на рис. 1.8 звено i-1). За положительное направление для \mathbf{x}^i принимается направление вектора \mathbf{e}_i .

При вращательных шарнирах может встретится особный случай, когда какой-либо из векторов r_{ij} , входящий в определение угла, оказывается параллельным оси шарнира e_j . Тогда его проекцияна плоскость перпендикулярную оси шарнира совпадает со следом оси на этой плоскости и определить угол указанным выше способом невозможно. Немозможно определить угол также при $r_{ij} = 0$. Указанные случай будем на зывать "особыми" и при них свяжем со звеном некоторый единичный вектор ρ_{ij} перпендикулярный e_j . Этот вектор ρ_{ij} будем использовать при определении угла поворота, вместо вектора $(-r_{ij})$, так как показано на рис. 1.9.

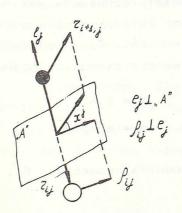


Рис 1.9

Для того, чтобы в дальнейшем не проверять в алгорит-мах каждый раз параллельности векторов \mathbf{r}_{ij} и \mathbf{e}_{j} или \mathbf{r}_{ij} =0

целесообразно предусмотреть это при подготовке исходных данных. Для этого введем число $\xi \mathbf{4}_{\mathbf{i}\mathbf{j}}$, характеризующее ука занные особые случаи:

$$\xi \mathbf{4}_{ij} = \left\{ \begin{array}{ll} 1 & \text{если } \mathbf{r}_{ij} & \text{е}_{j} & \text{или } \mathbf{r}_{ij} = 0 \\ \\ 0 & \text{в другех случаях} \end{array} \right.$$
 (1.5)

Остановимся вкратце на связи относительных координат с числом степеней свободы механизма и введем ряд обозначений. Обозначим через X множество относительных координат данной цепи. Элементами этого множества $\mathbf{x}^{\mathbf{i}}$, а число элементов обозначим \mathbf{n} . Очевидно, \mathbf{n} равно числу кинематических пар цепи. В случае разомкнутой цепи число степе ней свободы механизма равно \mathbf{n} . В случае замкнутой цепи, число степеней свободы будет меньше числа кинематическех пар. Обозначим число степеней свободы $\mathbf{n}_{\mathbf{u}}$. Теперь выделим из множества относительных координат $\mathbf{n}_{\mathbf{u}}$ независимых координат, которые однозначно определяют положение цепи. Эти координаты будем называть основными или \mathbf{u} — координатами и обозначать $\mathbf{u}^{\mathbf{i}}$. Все остальные координаты назовем дополнительными и обозначим $\mathbf{s}^{\mathbf{i}}$. Таким образом множество \mathbf{X} включает в себя два непересекающихся подмножества \mathbf{U} и \mathbf{S}

$$X = U + S$$

$$n = n_{U} + n_{S}$$
(1.6)

где

 ${
m n}_{
m S}$ - число элементов множества ${
m S}$.

^{*)} С точки эрения удобства отсчета координат иногда полезно "искусственно" задавать особый случай (т.е. $\xi 4=0$) безотносительно к условиям (1.5). Это дает возможност опре делять относительные координаты не между \mathbf{r} , а между век торами ρ .

Подмножество S , очевидно, будет пустым для открытой цепи. Подмнодество основных координат U в свою очередь разобьем на два подмножества Z и Y с числом элементов n_Z и n_Y соответственно. Подмножество Z включает в себя координаты тех шарниров, которые оснащены кинематическим при
водом. Основные координаты будем назначать с таким усло вием, втобы они включали в себя все координаты, соответствующие кинеметическим приводам. Подмножество Y включает
в себя остальные u- координаты. В них либо нет привода ,
либо привод силовой. Подмножества Z и Y также не пересекаются

$$U = Z + Y$$

$$n_{u} = n_{z} + n_{y}$$
(1.7)

В дальнейшем для краткости будем говорить u - координаты, y - координаты, а также u - шарниры, y - шарниры, подразумевая, что координаты, соответствующие этим шарнирам,отнесены к подмножествам U, Y и т.д.

Номерацию элементов в каждом из подмножеств будем осуществлять независимо.

По характеристическому вислу $\xi 2_j$ можно судить входит ли \mathbf{x}^j в подмножество \mathbf{Z} . Для того, чтобы указать принадлежит ли \mathbf{x}^j подмножеству основных координат \mathbf{U} введем еще одно характеристическое число $\xi \mathbf{5}_i$

$$\xi 5_{j} = \begin{cases} 1 & \text{если } x^{j} \in U \\ 0 & \text{если } x^{j} \notin U \end{cases}$$
 (1.8)

Относительное перемещение в ј - ой кинематической паре можно выразить вектором х^је^ј , а скорости и ускорения в относительном движении (для простоты будем говорить просто относительные скорость и ускорение) - векторами $\dot{x}^{\dot{j}}$ е \dot

Для сокращения обозначений ниже будут использоваться следующие многомерные векторы (все векторы - столбцы,штрихом обозначается операция транспонирования)

$$x = (x^{1}, x^{2}, ..., x^{n})'; \quad \dot{x} = (\dot{x}^{1}, \dot{x}^{2}, ..., \dot{x}^{n})';$$

$$y = (y^{1}, y^{2}, ..., y^{ny})'; \quad \ddot{x} = (\ddot{x}^{1}, \ddot{x}^{2}, ..., \ddot{x}^{n})';$$

$$z = (z^{1}, z^{2}, ..., z^{nz})'; \quad y = (y^{1}, y^{2}, ..., y^{ny})';$$

$$s = (s^{1}, s^{2}, ..., s^{ns})'; \quad \ddot{s} = (\ddot{s}^{1}, \ddot{s}^{2}, ..., \ddot{s}^{ns})';$$

$$(1.9)$$

Кинематическая схема механизма может содержать не сколько разомкнутых и замкнутых цепеы, соединенных друг с другом, например, как показано на рис. 1.10. Здесь имеется

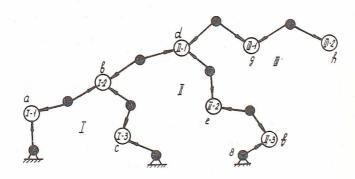


Рис 1.10

три цепи. Первая замкнутая образована звеньями а, b, с и ба зовыми звеньями для нее является стойка. Вторая цепь, так-же замкнутая, образована звеньями d, e, f, начальным базовым звеном служит звено 2 в первой цепи, а конечным стойка. Последная третья цепь - разомкнутая, она содержит звет

нья g и h , и базой для неё слежит эвено d .

Номера кинематических цепей будем указывать римскими цифрами, а звенья и кинематические пары в каждой цепи будем номеровать независимо. На рис. I.11 показаны отдельные цепи, составляющие механизм, полная схема которого приво -

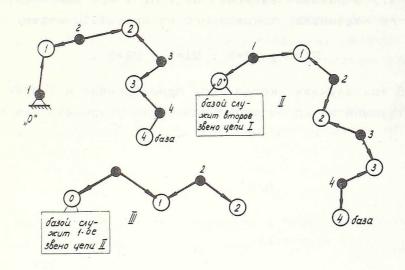


Рис 1.11

дилась выше (рис. I.10) и указана номерация пар и эвеньев. В тех случаях, когда это необходимо, будет указыватся номер цепи последним нижним индексом при соответствующих переменных, например,

ит. п.

Здесь k - томер цепи, i,j - номера шарниров и эвеньев в данной цепи. Однако, в большинстве случаев мы будем опу - скать этот индекс, чтобы не загромождать обозначения. Так в главах II-V книги будут рассматричаться только отдель-

ные цепи и в указании номера цепи нет необходимости.

Для каждой цепи будем указывать номера ее базовых це пвй (то-есть тех цепей, в которйх расположены базовые звенья) и номера базовых звеньев в этих цепях, Номера базовых цепей обозначим N1 , N2 (состветстченно начальная и конеч ная цепь), а базовых звеньев n1 , n2 . Так например, для цепи 2-го механизма, показанного на рис. 1.10 имеем:

N1=1; N2=0; N1=2; N2=0.

В тех случаях когда цепь присоединена к стойке, соотчетствующий номер базовой цепи будем указывать как нулевой.

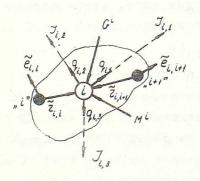
§3 Параметры звеньев

Звено механизма будем рассматривать как твердое тело с массой m_1 и тремя моментами инерции J_{i1} , J_{i2} , J_{i3} . Со звеном свяжем систему координат, начало которой распо - ложено в центре масс, а оси направлены вдоль главных центральных осей инерции. Эту систему координат назовем внутренней и соответствующий нормированный ортогональный базис обозначим q_{i1} , q_{i2} , q_{i3} .

Во всех случаях будем считать, что базисные векторы прономерованы таким оразом, что образуют правую тройку век торов, то-есть

$$q_{i3} = q_{i1} \times q_{i2}$$

Моменты инерции прономерованы в порядке убывания их величин; моменту инерции $J_{ ext{il}}$ состветствует орт. $q_{ ext{il}}$ и т.д (рис. 1.12).



Puc. 1.12

Каждому шарниру ставим в соответствие единичный векторы \tilde{e}_{ij} , направленный вдоль оси шарнира. Векторы \tilde{e}_{ij}

 $\tilde{\mathbf{r}}_{\mathbf{ij}}$ будем задавать в проекциях на оси внутренней системы координат, то-есть:

$$\tilde{e}_{ij} = \tilde{e}_{ij}^{1} q_{i1} + \tilde{e}_{ij}^{2} q_{i2} + \tilde{e}_{ij}^{3} q_{i3};$$

$$\tilde{r}_{ij} = \tilde{r}_{ij}^{1} q_{i1} + \tilde{r}_{ij}^{2} q_{i2} + \tilde{r}_{ij}^{3} q_{i3},$$
(1.10)

где в $\stackrel{\sim}{e_{ij}}^k$ и r_{ij}^k первый индекс указывает номер звена , второй – номер шатнира, а верхний индекс – номер проекции на соответствующую ось внутренней системы координат.

Остановимся на связи введенных ранее векторов е , , rij с векторами е ij , rij . Векторы е и rij определя- ют направление осей шарниров и расстояние центров масс от центров шарниров в неподвижной системе координат после того, как механизм уже "собран".

Векторы $\tilde{\mathbf{e}}_{\mathbf{i}\mathbf{j}}$ и $\tilde{\mathbf{r}}_{\mathbf{i}\mathbf{j}}$ напротив никак не связаны с расположением звена относительно неподвижной системы координат и являются внутренними параметрами звена. Разумеется , после "сборки" механизма соотчетствующие векторы $\mathbf{e}_{\mathbf{j}}$ и $\tilde{\mathbf{e}}_{\mathbf{i}\mathbf{j}}$ совместятся; поэтому для того, чтобы подчеркнуть связь между $\mathbf{e}_{\mathbf{j}}$ и $\tilde{\mathbf{e}}_{\mathbf{i}\mathbf{j}}$, они обозначены одной и той же буквой (ана глогично для $\tilde{\mathbf{r}}_{\mathbf{i}\mathbf{j}}$).

Остановимся на поступательных шарнирах. В таких шарнирах принимаются специальные меры для того, чтобы исклю тить относительное вращение направляющих и ползуна. Это мож но сделать, выполнив, например, направляющие плоскими или введя штифты, как ,оказано на рис. 1.13. Следовательно, для характеристики поступательного шарнира недостаточно одного вектора e_{ij} , а необходимо ввести еще один единичный вектор (обозначир его ρ_{ij}), который определяет либо плоскость базирования, либо штифт и тем самым указывает относительное положение элементов шарнира при "сборке".

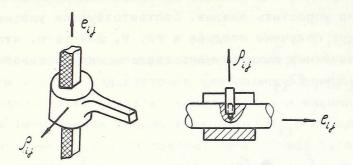


Рис 1.13

Кроме того, при наличии поступательного шарнира следует различать случаи ползуна и направляющих. Если в звене расположен ползун, то в нем можно некоторую точку принять условно за центр шарнира и относительно этой точки определить вектор $\tilde{\mathbf{r}}_{\mathbf{ij}}$. Если же в звене расположены направляющие, то для них понятие центра шарнира теряет смысл до тех пор, пока пара не собрана и на направляющих не установлен ползун. Но и в последнем случае центр шарнира окажется подвижным и будет перемещаться вместе с ползуном.

Поэтому условимся в случае направляющих располагать \hat{r}_{ij} вдоль перпендикуляра из центра масс на оси направляющих. (рис. 1.14). От этого же положения будем вести отсчет координаты $\mathbf{x}^{\mathbf{j}}$.

Следует выделить случаи, когда звено механизма представляет собой тонкий однородный стержень (рис. 1.15). Примени тельно к стержню целесообразно

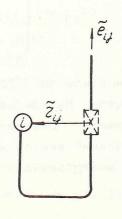
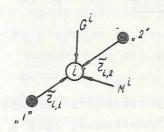


Рис 1.14

вести анализ не по общим формулам, а вывести специальные уравнения, учитывающие специфику объекта. При этом можно существенно упростить анализ. Соответствующие уравнения динамики будут получены позднее в гл. V. Для того, чтобы выделить отмеченные выше случаи, введем характеристические числа $\xi 6_1$ и $\xi 7_{11}$.



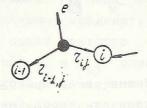


Рис. 1.15

Рис. 1.16

Нижние индексы при $\xi 6_{f i}$, $\xi 7_{f ij}$ указывают номер звена или пары в цепи. На каждое звено могут действовать внешние силы и моменты. Приведем их к центру масс звена и получив тийся главный вектор и главный момент обозначим $G_{f i}$ и $M_{f i}$ где ${f i}$ номер звена в цепи (рис. 1.12).

§4 Символическое описание механизмов

Введенных выше параметров (характеристические числа, векторы и т.д.) недостаточно для полного описания модели. Необходимо также указать, какие звенья и какими своими шар нирами (или какими своими векторами \tilde{e}_{ij}) объединены в кинематические пары и составляют цепь. Кроме того, с точки зрения составления алгоритмов, удобно всю введенную ранее информацию объединить в несколько массивов. Ниже будут подробно описаны такие массивы. Разумеется, именно такое распределение параметров по массивам не является обязательным и выполнено с единственной целью упростить алгоритмы.

Первый массив описывает структуру механизма и для не го введем обозначение STR. Он представляет собой матрицу с целочисленными компонентами. Каждая строка соответствует одной цепи (в порядке их номерации). Внутри строки информация по столбцам распределяется следующим образом:

- 1. характеристическое число ξ3, указывающее тип цепи (см. 1.4);
- 2. номер начальной базовой цепи (N1);
- 3. номер началного базового звена (nl);
- 4. номер конечной базовой цепи (N2);
- 5. номер конечного базового звена (n2);
- 6. число кинематических пар цепи (n);
- 7. число основных координат (n_{ij}) ;
- 8. число у-координат (n_y) ;
- 9. λ_1 специальные числа, характеризующие тип связи
- 14. λ₆ для замкнутой цепи (см. ниже гл. II §2)

Четвертый пятый, девятый - четырнадцатый столбцы запол -

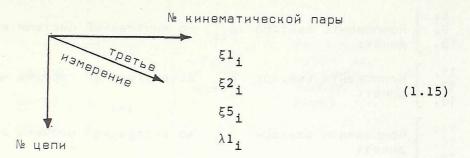
няются только для замкнутых цепей. Относительно информации в 6-8 столбцах см. (1.6, 1.7).

Условно матрицу можно представить в виде

Второй массив КР (кинематические пары) описывает параметры и соединения звеньев в каждой цепи. Этот массив трехмерный. Первые два измерения условимся называть как для матриц строками и столбцами. Тогда каждая строка КР соответствует одной цепи в порядке их номерации. Номера столбцов соответствуют номерации кинематических пар в цепи. На "пересвчении" строки и столбца помещаются 4 целых числа в следующей последовательности (по третьму измерению массива):

- 1. характеристическое число $\xi \mathbf{1}_{\mathtt{i}}$, указывающее тип кинематическей пары (см. 1.1);
- 2. характеристическое число $\xi 2_{i}$, указывающее тип в данном шарнире (см. 1.3);
- 3. характеристическое число $\xi 5_{i}$, указывающее относится ли данный шарнир к основным шарнирам (т.е. x^{i} ε U, см. /1.8/);
- 4. характеристическое число $\lambda 1_{f i}$, которое используется при анализе статики. Его назначение будет объяснено поэже в гл. IV.

Условно массив КР можно представить в виде



Следующий третий массив устанавливает в какой последова - тельности звенья соединены в кинематические пары. Массив трехмерный, назван ZV (звенья). Номера строк соответствуют номеру цепи, а столбцов - номеру звена в цепи (вклю - чая базовые звенья). По третьему измерению расположены 27 действительных чисел в следующей последовательности:

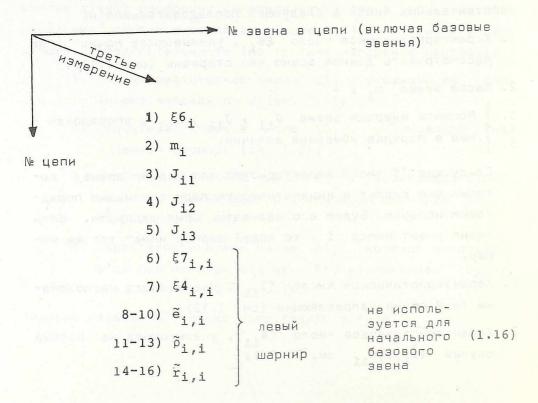
- 1. Характеристическое число $\xi 6_{ extbf{i}}$, указывающее можно ли рассматривать данное эвено как стержень (см. 1.11);
- 2. Масса звена т.;
- 3. $\begin{cases} & \text{Моменты инерции звена} & \text{J}_{i1} \text{ , J}_{i2} \text{ , J}_{i3} \text{ , упорядочен -} \\ & \text{4.} \\ & \text{5.} \end{cases}$ ные в порядке убывания величин;

Следующие 11 чисел характеризуют тот шарнир звена, которым оно входит в кинематическую пару с меньшим порядковым номером. Будем его называть левым шарниром. Если звено имеет номер і, то левый шарнир имеет тот же номер.

- 6. Характеристическое число $\xi 7_{ ext{ii}}$, указывающее расположены ли в звене направляющие (см. 1.12);
- 7. Характеристическое число $\xi 4_{ii}$, указывающее на особый случай $(\tilde{r}_{ii} \mid \mid \tilde{e}_{ii} \mid cm. /1.5/);$

```
8. 9. \frac{1}{9}. \frac{1}{9}.
```

Векторы \tilde{e}_{ii} , $\tilde{\rho}_{ii}$, \tilde{r}_{ii} относятся к левому шарниру. Следующие одиннадцать чисел указывают аналогичную информацию для правого шарнира (номер которого равен i+1). Под правым понимается второй шарнир звена, которым оно входит в данную цепь. Условно массив представим в виде:



17)
$$\xi 7_{i,i+1}$$
18) $\xi 4_{i,i+1}$
19-21) $\tilde{e}_{i,i+1}$
22-24) $\tilde{\rho}_{i,i+1}$
25-27) $\tilde{r}_{i,i+1}$

Первый нижний индекс і везде указывает номер звена в цепи. В случае стержня (ξ 6=1) используются только два первые момента инерции (J_{i1} , J_{i2}). Векторы $\tilde{\rho}$ используются только в тех случаях, когда соответствующий шарнир поступательный, или имеет место особый случаы (ξ 4=1). Начальному базовому звену всегда присваиваем нулевой номер. Относительно ZV важно отметить, что векторы \tilde{e} , $\tilde{\rho}$ и \tilde{r} определены в проекциях на внутреннюю систему координат. Это, дает возможность задавать параметры каждого звена независимо от его расположения в цепи, а затем уже по определенному алгоритму (см. ниже) осуществлять "сборку" механизма, "соединяя" звенья в той последовательности, как указано в ZV

В каждой строчке массива ZV описываются только те шарниры, которые входят в данную кинематическую цепь. Если некоторое звено содержит также другие шарниры, с помощью которых оно входит в другие цепи, то параметры этих шарниров будут фигурировать в других строчках ZV . Таким образом информация об одном звене может находиться в различных строках и столбцах ZV , если, конечно, звено входит в различные цепи.

Описанные три массива STR , KP и ZV содержат те параметры механизма, которые должны быть подготовлены заранее до начала расчетов.

§5 Форма записи алгоритмов

В следующих главах книги все параграфы будут строиться по единому плану: сначала формулируется задача, затем описывается метод ве решения, затем приводится блок схема алгоритма и, наконец, записывается текст алгоритма блок-схема и текст алгоритма снабжаются кратким коммента рием.

Блок-схемы приводятся с целью нагладно показать ло-гическую структуру алгоритма. Поэтому в блок-схемах используются, как правило, те же обозначения, что и в соответствующих формулах. Все изменения обозначений каждый раз оговариваются. В частности, если требуется определить тип цели, то в блок-схемах будет указано число ξ_{N} , а не STR N,1, поскольку логическая структура алгоритма не требует, чтобы ξ_{N} хранилось неприменно в матрице STR. Однако, в этом случае в жертву логической ясности принесены некоторые вопросы практического построения вычислительных программ. В частности блок-схемы только в малой степени отражают распределение рабочей памяти машины и хранение информации.

Для того, чтобы восполнить этот пробел, в книге помимо блок-схем приводятся также тексты алгоритмов.Эти тексты представляют собой примеры практической реализации алгоритмов. После небольших технических изменений, связанных с их подгонкой под возможности используемого транслятора, эти тексты могут использоваться для решения конкретных задач.

При записи алгоритмов за основу принят язык АЛГОЛ-60. По сравнению с классическим вариантом этого языка сделано

два небольших добавления. В качестве идентификаторов допусказтся использование всех строчных букв латинского, греческого и русского алфавитов. Во вторых выеден новый эператор

(a,b)

где а и э - некоторые идентификаторы, причем идентифи - катор " b^* описан ранее оператора ИМЯ.

Этот оператор означает, что ниже в таксте алгоритма оба идентификатора в и в обозначают одну и ту же переменную или массив и оба они могут с равным правом использоваться во всех других операторах.

Эти добавления сделаны с вдинственной целью прибли - зить обозначения в алгоритмах к соответствующим обозначе - ниям а блоксхемах и формулах. В частности, в алгоритмах часто приходится использовать один и тот же массив для хранения различной информации. В этом случае оператор ИМЯ поваволяет различную по смыслу информацию обозначать соответствующими буквами.

Если указанные два дополнения вызовут какие-либо тру дности при практическом использовании алгоритмов. То эти дополнения легко исключить. В частности можно сузить алфавит, первобозначие соответствующие идентификаторы. Операторы ИМЯ не совершают никаких вычислении, поэтому их можно просто исключить, также первобозначие соответствующие первобозначие соответствующие первобозначие.

В книге рассматривается значительное число различных задач и приводится довольно много различных алгоритмов При этом составленные ранее алгоритмы часто используются в посследующих параграфах. Поэтому наиболее удобной формой представления алгоритмов оказались процедуры. В Приложении Трается список всех процедур (всего более двадцати).

В процедурах используется ряд глобальных идентификаторов. Во первых, как глобальные используются массивы STR, KP, ZV, которые описаны в предыдущем параграфе. Эти массивы объединяют ту информацию, которая предварительно подготавливается исследователем и затем вводится в ЭЦВМ. Кроме того, для хранения информации, получаемой в результате работы алгоритма, необходимо иметь еще ряд мяссивов. Описание этих массивов дается в Приложении.

Гл. II ГЕОМЕТРИЯ ПРОСТРАНСТВЕННЫХ ЦЕПЕЙ

§1 "Сборка" разомкнутых цепей

"Сборка" будет состоять в том, втобы "соединить" все звенья в той последовательности, как указано в ZV, то-есть совместить в пространстве оси соответствующих шарнинов и их центры. После "сборки" необходимо определить положение механизма в абсолютной системе координат, то-есть найти:

- направления осей шарниров и главных осей инерции всех звеньев ($\mathbf{e_i}$, $\mathbf{q_{ij}}$) ;
- все векторы r_{ij} .

В процессе "сборки" будем предполагать, что все относительные координаты $\mathbf{x}^{\mathbf{i}}$ равны нулю (перемещение "собранного" механизма на величины $\mathbf{x}^{\mathbf{i}} \neq 0$ обсуждается в следующем параграфе).

Пусть некоторая часть механизма уже "собрана" и последнее "собранное" звено имеет номер i-1. К нему нужно присоединить звено "i" таким образом, чтобы шарнир A зве на "i" "собрался" с шарниром B цепи (рис. 2.1).

Поскольку (i-1)-е звено уже "собрано"(в дальнейшем условимся опускать кавычки), то известны проекции векторов $r_{i-1,i}$ и e_i в неподвижной системе координат. В результате сборки вектор $\tilde{e}_{i,i}$ должен совпасть с e_i .

Рассмотрим сначала случай, когда i-ый шарнир враща-

тельный. После совмещения осей шарниров звено "1" должно быть установлено таким образом, чтобы угол $\mathbf{x}^2\!=\!0$. Для определения \mathbf{x}^* необходимо найти проекции $\mathbf{x}_{2-1,1}$ и \mathbf{x}_{2-1} на плоскость перпендикулярную \mathbf{a}_2 . Для упрощения обозначений условимся ниже опускать нижниз индексы пои \mathbf{a}_2 . \mathbf{x}_2 и \mathbf{x}_3 и \mathbf{x}_4 .

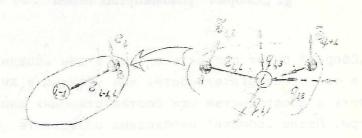


Рис. 2.1

Разложим У и Т на две составляющие: параллельные ч пво пендикулярные е Перпендикулярные составляющие будут

$$\vec{r}_{N} = e \times (r \times e)$$

$$\vec{r}_{N} = \vec{e} \times (\vec{r} \times \vec{e}) \qquad (2.1)$$

Нормализуем составляющие $\mathfrak{T}_{\mathfrak{M}}$ и $\mathfrak{T}_{\mathfrak{M}}$, обозначив соответствующие вдиничные векторы \mathfrak{A} и \mathfrak{A} , и поменяем знак у $\mathfrak{T}_{\mathfrak{M}}$

$$a = \frac{x}{|x_N|} ; \quad \tilde{a} = \frac{\tilde{x}_N}{|\tilde{x}_N|}$$
 (2.2)

Условие $\mathbf{x} = \mathbf{0}$ приводит к тому, что векторы а н $\tilde{\mathbf{a}}$ долж-

Условия сборки можно записать в виде вы выдражива др.

где A - матрица перехода от внутренней системы координат

1 - го звена к неподвижным координатам»

Напомним, что векторы ё и ã заданы в проекциях на внутренние оси, а е и а суть те же векторы, но в проекциях на неподвижные оси.

Задача состоит в том, чтобы найти матрицу Å, так как соотношения аналогичные (2.3) справедливы для любого внуттреннего вектора 1-го звена. Для определения А недостаточно двух уравнений (2.3), поскольку А имеет 9 неизвествых компонент. Поэтому дополним (2.3) условием

$$b = e \times a$$
; $\tilde{b} = \tilde{e} \times \tilde{\epsilon}$ (2.4)

Условив (2.4) кроме того обеспечивает линейную независимость трех векторов, участвующих в определении А, что важно в дальнейшем.

Записывая (2.3 , 2.4) в проекциях получим

где верхние индексы при $\tilde{\mathbf{a}}$, $\tilde{\mathbf{e}}$, $\tilde{\mathbf{b}}$ указывают номер проекции на подвижные оси.

Решим (2.5) относительно $A^{\mbox{ij}}$. Расширенная матрица В будет имет вид

$$B = \begin{bmatrix} \tilde{a}^{1} & \tilde{a}^{2} & \tilde{a}^{3} & 0 & | & a^{X} \\ 0 & \tilde{a}^{1} & \tilde{a}^{2} & \tilde{a}^{3} & 0 & | & a^{Y} \\ | & 0 & | & \tilde{a}^{1} & \tilde{a}^{2} & \tilde{a}^{3} & a^{Z} \\ | & 0 & | & \tilde{a}^{1} & \tilde{a}^{2} & \tilde{a}^{3} & a^{Z} \\ | & \tilde{e}^{1} & \tilde{e}^{2} & \tilde{e}^{3} & 0 & | & e^{X} \\ | & 0 & | & 0 & | & \tilde{e}^{1} & \tilde{e}^{2} & \tilde{e}^{3} & e^{Z} \\ | & \tilde{b}^{1} & \tilde{b}^{2} & \tilde{b}^{3} & 0 & | & b^{X} \\ | & 0 & | & \tilde{b}^{1} & \tilde{b}^{2} & \tilde{b}^{3} & b^{Z} \end{bmatrix}$$

$$(2.6)$$

Определитель системы не равен нулю в силу (2.4) .

В случае поступательной пары или "особого случая"вращательной пары определение А оказывается более простым. Здесь нет необходимости искать проекции векторов и доста точно просто положить во втором уравнении (2.3)

$$a = \rho$$
 ; $\tilde{a} = \tilde{\rho}$

После определения матрицы перехода бекторы \mathbf{r} и е находятся по (2.3) . Проекции ортов внутренней системы координат $(\mathbf{q_{ij}})$ будут определяться просто столбцами \mathbf{A} . Например, проекции $\mathbf{q_{i2}}$ будут $\mathbf{A_i^{12}}$, $\mathbf{A_i^{22}}$, $\mathbf{A_i^{23}}$.

Блок-схема алгоритма для сборки i-ой кинематической пары приведена на рис. 2.2. Предполагается, что (i-1) - ϵ эвено уже собрано и известна матрица перехода для этого эвена A_{i-1} . Поэтому номер кинематической пары i и матрица A задаются, как формальные параметры.

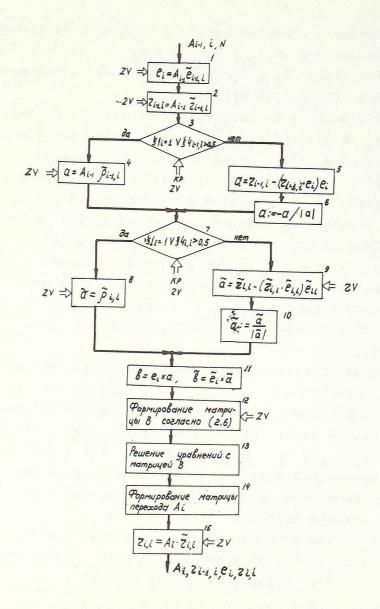


Рис. 2.2. Блок-схема алгоритма СБОРКА ПАРы

Алгоритм "присовдиняет" i-е звено к (i-1)-ому и определяет векторы $r_{i-1,i}$, $r_{i,i}$, e_i и матрицу перехода для i-го звена A_i . Алгоритм построен в соответсвии с формулами (2.1) - (2.6) .

В блоках 1.2 рассчитывается орт оси кинематической пары ${\bf e_i}$ и вектор ${\bf r_{i-1,i}}$. Блок 3 проверяет тип кинема - тической пары $(\xi {\bf l_i})$ и наличие особого случая, когда ${\bf r_{i-1,i}}||{\bf e_i}|$. Если пара поступательная или имеет место особый случай, то используется вспомогательный вектор ${\bf \rho_{i-1,i}}^{\circ}$ Блок 7 осуществляет аналогичную проверку для ${\bf i-ro}$ звена.

Широкая стрелка рядом с блоком указывает на вызов из массива или запись в массив некоторой информации. Рядом со стрелкоы записано наименование массива. Например, в блоке I компоненты вектора $\tilde{e}_{i-1,i}$ берутся из массива ZV. Подобные обозначения будут использоватся и в других блок - схемах.

В процедуре используются глобальные массивы KP, ZV. В тексте алгоритма обозначения e0 , r0 соответствуют век торам \tilde{e} , \tilde{r} , в формулах (2.1) - (2.6) .

procedure CБОРКА ПАРы (N, i, rl, e, r2, A);
value N, i; integer N, i; array A, rl, e, r2;
begin

array a, b, e0, r0[1:3], B[1:9,1:9], c[1:9];
integer j, k; real s;

```
for j: = 1,2,3 do
     begin e0[j] : = ZV[N,i-1,j+18];
           r0[j] := ZV[N,i-1,j+24] end;
     MTB(e,A,e0,3,3); MTB(r1,A,r0,3,3);
     if KP[N,i,1] = 1VZV[N,i-1,18] > 0.5 then
           begin for j:=1,2,3 do
           a[j] = ZV[N,i-1,j+21];
           MTB(a,A,a,3,3);
           go to ml end;
     СКЛ (s,e,rl);
     for j := 1,2,3 do a[j] := s \times e[j] - rl[j];
     HOPM (a,a);
comment закончен расчет требуемых параметров для (i-l)-го
        звена /блоки 1-6 блок-схемы/;
     ИМЯ (a0,r2);
ml: \underline{if} KP[N,i,1] = 1V ZV[N,i,7] > 0.5 \underline{then}
     begin for j: = 1,2,3 do
           a0[j] := ZV[N,i,j+10];
           go to m2 end;
     for j: = 1,2,3 do begin
         e0[j]: = ZV[N,i,j+7];
      r0[j] : = ZV[N,i,j+13] end;
     СКЛ (s,e0,r0);
     for j: = 1,2,3 do a0[j]: = r0[j]-s \times e0[j];
     HOPM (a0, a0);
m2:
    (00,r0);
     BEHT (b,e,a); BEHT (b0,e0,a0);
comment переходим к формированию матрицы В;
     for j: = 1 step 1 until 9 do
     for k: = 1 step 1 until 9 do
           B[j,k] := 0;
```

```
for j: = 1,2,3 do begin

B[1,j]: = B[2,j+3]: = B[3,j+6]: = a0[j];

B[4,j]: = B[5,j+3]: = B[6,j+6]: = e0[j];

B[7,j]: = B[8,j+3]: = B[8,j+6]: = b0[j];

c[j]: = a[j];

c[j+3]: = e[j];

c[j+6]: = b[j] end;

AMT YP (c,B,c,9);

for j: = 1,2,3, do begin

for k: = 1,2,3 do

A[j,k]: = c[3×(j-1)+k];

a[j]: = ZV[N,i,j+24] end;

MTB (r2,A,a,3,3)
```

end процедуры;

Для сборки всей разомкнутой цепи необходимо, очевидно, применить алгоритм СБОРКА ПАРы последовательно ко всем кинематическим парам цепи. При этом должна быть известна матрица перехода \mathbf{A}_{O} для базового звена. Вопрос о том, как определить \mathbf{A}_{O} в случае, когда механизм состоит из нескольких цепей, разбирается в гл. VI. В часном случае, когда базой служит неподвижная стойка, матрица \mathbf{A}_{O} превращается просто в единичную матрицу $\mathbf{3} \times \mathbf{3}$.

Глобальные идентификаторы процедуры СБОРКА ЦЕПИ : STR, KP, ZV, EO, RO, QO

```
procedure CEOPHA LETM (N,A);
     value N; integer N; array A;
begin
     array e, r1, r2[1:3];
     integer i,j,k,n;
     n: = STR[N,6];
     for j: = 1,2,3 do for k: = 1,2,3 do
```

```
QO[N,0,j,k]: = A[j,k];
for i: = 1 step 1 until n do
begin

CБОРНА ПАРы (N,i,rl,e,r2,A);
for j: = 1,2,3 do begin

EO[N,i,j]: = e[j];

RO[N,i-1,2,j]: = rl[j];

RO[N,i,1,j]: = r2[j];

for k: = 1,2,3 do

QO[N,i,j,k]: = A[j,k] end

end end процедуры;
```

Алгоритм вычисляет векторы e_i , r_{ij} и матрицы A_i после "сборки" и заносит их в массивы E0, R0, Q0 соответственно. В результате выполнения процедуры заполняется N-я строка этих массивов.

§2 Определение положения разомкнутых цепей

Сборка механизма осуществлялась при условии, что относительные координаты $\mathbf{x^i}$ во всех шарнирах механизма равны нулю. Теперь будем перемещать цепь в положение, отвечатющее некоторым значениям координат $\mathbf{x^i}$. Для нового полонжения цепи необходимо определить направление осей всех шарниров е , векторы из центров шарниров к центрам масс соседних звеньев \mathbf{r} и направления осей внутренней системы коорденат всех звеньев $\mathbf{q_i}$.

Прежде всего рассмотрим поворот некоторого вектора ${\bf r}$ на угол ${\bf x}$ вокруг оси, заданной единичным вектором ${\bf e}$. Для расчета нового положения ${\bf r}$ (обозначим его ${\bf r_i}$) можно воспользоваться формулой

$$r_i = r \cos \phi + (1-\cos \phi) (e \cdot r) e + e \times r \sin \phi$$
 (2.7)

Поскольку при расчете перемещений механизма много - кратно придется поворачивать векторы, целесообразно составить стандартный алгоритм. В приведенном ниже алгольном тексте вектор ${\bf r}$ поворачивается вокруг ${\bf e}$ на угол ${\bf x}$. После поворота, проекции вектора вновь помещаются в массив ${\bf r}$.

$$r[i]$$
: = g1 × $r[i]$ + (1-g1) × a×e $[i]$ + g × b $[i]$ end процедуры;

Теперь можно переходить к расчету положения всей цепи. Для этого будем последовательно обходить все шарниры цепи в порядке номерации. В каждом ј-ом шарнире будем осуществлать поворот или перемещение, в зависимости от типа шарнира, на величину х^ј. При этом в перемещении участвуют все звенья, расложенные в цепи дальше того шарнира, в котором осуществляется перемещение. "Дальше" тут следует понимать в смысле принятого направления обхода цепи. Для разомкнутой цепи это означает, что мы перемещаем как целое весь участок цепи от данного шарнира до свободного конца.

Если рассматриваемый шарнир вращательный, то в повороте участвуют все последующие звенья и, следовательно, нуж но рассчитать новые положения векторов е, r, q каждого из звеньев.

Допустим поворот осуществляется в j-ом шарнире. Тогда в формулу (2.7) или алгоритм ПОВОРОТ нужно вместо х подставить $\mathbf{x}^{\mathbf{j}}$, а вместо е подставить $\mathbf{e}_{\mathbf{j}}$ и таким образом будут определены ось и угол поворота. Далее вместо \mathbf{r} будем подставлять обозначение поворачиваемого вектора; сначала единичные векторы осей шарниров $\mathbf{e}_{\mathbf{i}}$ для всех шарниров, номер которых $\mathbf{i} > \mathbf{j}$, затем векторы $\mathbf{r}_{\mathbf{i}\mathbf{i}}$, $\mathbf{r}_{\mathbf{i},\mathbf{i}+\mathbf{l}}$ из центров шарниров к центрам масс звеньев для всех \mathbf{i} -ых звеньев, номер которых $\mathbf{i} \geqslant \mathbf{j}$ и, наконец, базис внутренней системы координат $\mathbf{q}_{\mathbf{i}\ell}(\ell=1,2,3)$ для звеньев, номер которых $\mathbf{i} \gg \mathbf{j}$.

После осуществления всех этих поворотов часть разом-кнутой цепи, расположенная за j-ым шарниром окажется повернутой на угол $\mathbf{x}^{\mathbf{j}}$.

Если рассматривается перемещение в поступательном

шарнире. то изменятся проекции только одного вектора, а именно того вектора \mathbf{r}_{ij} , который проведен из центра этого шарнера в центр масс звена, несущего направляющие. Это изменение, очевидно, равно $\mathbf{x}^{\mathbf{j}}\mathbf{e}_{\mathbf{j}}$. Все остальные звенья переместятся поступательно и, следовательно, проекции векторов \mathbf{e} , \mathbf{r} , \mathbf{q} этих звеньев не изменятся.

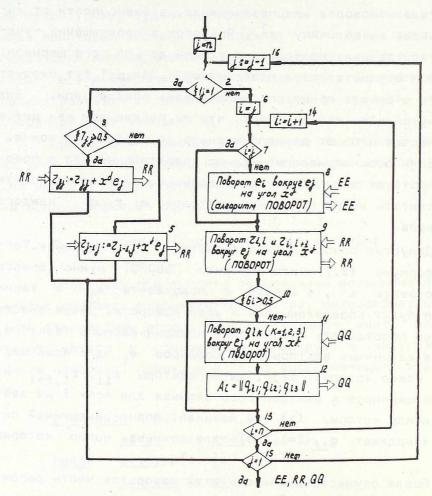


Рис. 2.4. Блок-схема алгоритма для расчета положения разомкнутой цепи

В соответствии с описанным процессом составлех алгоритм ПОЛОЖЕНИЕ, блок-схема которого приведена на рис. 2.4. Алгоритм осуществляет перемещение в шарнирах на величины $\mathbf{x}^{\mathbf{j}}$. После перемещения определяются векторы \mathbf{e} , \mathbf{r} для всех шарниров и звеньев цепи и матрицы перехода $\mathbf{A}.$

Цикл с параметром ј (блоки 1-16) осуществляет последовательный перебор всех шарниров цепи от n до 1 . Блок 2 проверяет тип ј-го шарнира (т.е. поступательный или вращательный). В случае поступательного шарнира проверяется в каком из звеньев, образующих ј-ю кинематическую пару, расположены направляющие (блок 3). Для этого звена рассчитывается новое значение вектора г . Если ј-я пара вращательная, то строится цикл с параметром і , в котором перебираются все звенья цепи с номерами большими или равными ј (блоки 6-14). Для каждого і-го звена рассчитывается поворот векторов е_і, r_{і,і}, r_{і,і+l} вокруг помощью алгоритма ПОВОРОТ (блоки 8,9). Если рассматрива ется звено с номером ј (i=j) , то поворот собственной оси, очевидно, можно не рассчитывать, что учитывается в блоке 7. Блок 11 рассчитывает поворот базиса внутренней системы координат $\mathtt{q}_{\mathtt{i}\ell}$. В случае стержня (блок 10) положение базиса не рассчитывается и блоки 11, 12 работают.

Ниже приводится текст алгоритма (процедура ПОЛОЖЕ - НИЕ). Формальный параметр N указывает номер цепи. В процедуре используются глобальные массивы STR , KP , ZV (см. гл. I $\S4$) и массивы E0 , R0, Q0, в которых хранятся значения e_i , r_{ii} , r_{ii}

procedure ПОЛОЖЕНИЕ (N);
value N; integer N;
begin

array r1, r2, e[1:3];

```
integer i,j,k,n,q; real a;
n: = STR[N,6];
for j: = n step -1 until 1 do
begin
         for k: = 1,2,3 do begin
         e[k] : = EE[N,j,k] : = E0[N,j,k];
         RR[N,j,1,k] := RO[N,j,1,k];
         RR[N,j,2,k] := R0[N,j,2,k];
          for i: = 1,2,3 \pm 0
          QQ[N,j,k,i] := QO[N,j,k,i] end;
          a: = X[N,j,1];
          if KP[N,j,1] = 1 then go to ml;
          for i: = j step 1 until n do
          begin
               if i = j then go to m2;
               for k: = 1,2,3 \text{ do}
                  rl[k] := EE[N,i,k];
               ПОВОРОТ (rl,e,a);
               for k: = 1,2,3 \text{ do}
                        EE[N,i,k] := rl[k];
          m2: for k: = 1,2,3 do
               begin rl[k] := RR[N,i,l,k];
                r2[k] := RR[N,i,2,k] end;
                ΠΟΒΟΡΟΤ (rl,e,a);
                ПОВОРОТ (r2,e,a);
                for k: = 1, 2, 3 do
          begin
                RR[N,i,1,k] := rl[k];
                RR[N,i,2,k] := r2[k] \underline{end};
           if ZV[N,i,1] > 0.5 then
                go to m3;
           for q : = 1, 2, 3 do
```

```
begin
              for k: = 1,2,3 do
               rl[k] := QQ[N,i,k,q];
             ΠΟΒΟΡΟΤ (rl,e,a);
             for k: = 1,2,3 do
               QQ[N,i,k,q] := rl[k]
         end;
m3: end i;
 go to m4;
 ml: if ZV[N,j,6] > 0.5 then
          for k: = 1, 2, 3 do
               RR[N,j,l,k] := RR[N,j,l,k] + a \times e[k]
          else
          for k: = 1,2,3 do
               RR[N,j-1,2,k] := RR[N,j-1,2,k] + a \times e[k];
 m4: end j;
 end процедуры;
```

Переходим к определению координат отдельных точек механизма после перемещения. Предварительно несколько расширим понятие радиус-вектора \mathbf{r}_{ij} . До сих пор \mathbf{r}_{ij} проводился из центра шарнира в центр масс того же самого звена, которому принадлежит шарнир. В дильнейшем это не будем считать обязательным и примем, что \mathbf{r}_{ij} определяет вектор с началом в центре любого \mathbf{j} -го шарнира цепи и концом в центре масс любого \mathbf{i} -го звена этой же цепи.

Тогда радиус-вектор из центра 1-ого шарнира цепи к центру масс i-го звена разомкнутой цепи можно записать как /см. рис. 1.5/

$$r_{i1} = \sum_{j=1}^{i-1} (r_{j,j} - r_{j,j+1}) + r_{i,i}$$
 (2.9)

 $B \times o$ дящие в эту правую часть векторы рассчитываются с помощью алгоритма ПОЛОЖЕНИЕ после чего расчет r_{il} не представляет труда.

Несколько сложнее случай, когда требуется определить положение некоторой точки "а", связанной с i-ым звеном. Пусть радиус-вектор этой точки в подвижной внутренней системе координат будет \tilde{h} .

Тогда прежде всего нужно получить проекции h в неподвижной системе координат

$$h = A_i \tilde{h}$$

гди ${\tt A}_{f i}$ - матрица перехода, определенная для данного поло-жения эвена.

Тогда, обозначив радиус-вектор из центра первого шарнира к точке "а" через \mathbf{r}_{al} имвем

$$r_{al} = r_{il} + h$$

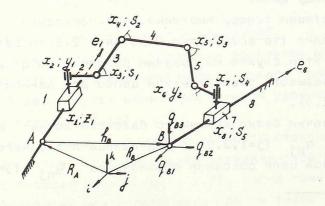
соответствующий алгоритм не сложен, поэтому его текст приводить не будем.

§3 Условия связи для замкнутых цепей

Для замкнутой цепи будем считать известным положение двух крайних базовых звеньев. В сравнении с разомкнутой цепью это значительно осложняет задачу определения положе ния. Если раньше все относительные координаты были основными, и значения всех $\mathbf{x}^{\mathbf{i}}$ можно было задавать независимо, то для замкнутой цепи можно задавать независимо значения только части относительных координат.

В главе I отнодительные координаты х делились на основные и-координаты и дополнительные s-координаты. Число основных координат равно числу степеней свободы цепи и при заданных значениях основных координат положение всего механизма определяется однозначно. Значения основных координат мы будем считать известными.

Дополнительные координаты должны в этом случае определяться из условия, что базовые звенья занимают заданное положение в пространстве.



Pug. 2.5

На рис. 2.5 показан пример замкнутой цепи содержащей восемь кинематических пар и обозначены все координаты * . В данном частном случае множество X содержит 8 елементов , а подмножества U , Z , Y , S соответствено 3,1,2 и 5.

За основные координаты могут быть приняты три элемента X; в данном случае это координаты в первом, втором и шестом шарнирах. Для цепи выбирается некоторое направление обхода и соответственно этому направлению номеруются независимо элементы каждого из подмножеств U, Z, Y, S (см. рис. 2.5).

Выберем на базовых звеньях характерные точки, кото - рыв назовем полюсами. Положение полюсов в неподвижной системе координат будем определять радиус-векторами $R_{
m A}$ и $R_{
m B}$. Для подвижных баз в качестве полюсов возмем их центры масс.

В тех случаях, когда базой служит неподвижная стойка за полюсы могут быть приняты любые две удобные точки базы. На рис. 2.5 представлен именно такой случай и точки А и В выбраны на направляющих таким образом, что в этих точках соответствующие координаты \mathbf{x}^1 и \mathbf{x}^8 равны нулю. Через \mathbf{h}_B обозначен вектор между точками А и В направленный от начала к концу цепи.

Представим теперь мысленно, что последнее базовое звено свободно (то-есть звено 8 на рис. 2.5 не связано со стойкой). В этом случае мы получим разомкнутую цепь, кото - рую будем называть эквивалентной цепью для данной замкну - той (рис. 2.6).

Обозначим базис конечного базового звена замкнутой цепи через \mathbf{q}_{Bj} (j=1,2,3) . Для базиса последнего звена эквивалентной цепи сохраним обозначения \mathbf{q}_{nj} (j=1,2,3) .

[★] В примере предполагается, что е₁ и е₃ лежат в одной плоскости.

Вектор от точки А к точке В для эквивалентной цепи обозначим h .

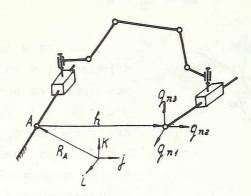


Рис. 2.6

В том случае если

$$h = h_B$$
 $q_{nj} \cdot q_{Bj} = 1$ $(j=1,2,3)$ }

эквивалентная цепь совпадает с замкнутой.

Задача определения положения замкнутой цепи таким об разом сводится к задаче определения такого положения эквивалентной разомкнутой цепи, при котором удовлетворяются условия замыкания (2.10).

Условия замыкания удобно записать несколько в иной форме. Допустим последнее звено эквивалентной цепи не совпадает с базовым звеном (рис. 2.7). Тогда можно определить такое поступательное перемещение Δh и такое угловое перемещение Δr , после выполнения которых система координат

"ћ" совместится с системой координат "В" (см. рис. 2.7). Требуемов поступательное перемещение Δh равно

$$\Delta h = h_B - h \tag{2.11}$$

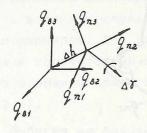


Рис. 2.7

Угловое перемещение $\Delta \gamma$ определяется из условия, чтобы при повороте вокруг $\Delta \gamma$ на угол ϕ $|\Delta \gamma|$ векторы q_{nj} совместятся с q_{Bj} (j=1,2,3). Вопрос определения $\Delta \gamma$ для заданных положений q_{nj} и q_{Bj} рассматривается в следующем параграфе.

Тогда условия замыкания можно записать в следующей простой форме

$$\Delta h = 0$$

$$\Delta \gamma = 0$$

$$(2.12)$$

Выберем некоторую систему координат с ортогональным базисом $\mathbf{q_i^0}$ (\mathbf{i} =1,2,3) и спроектируем (2.12) на оси этой системы координат. Получим шесть скалярных уравнений. Однако, не все эти уравнения могут и должны использоваться при определении зависимых координат $\mathbf{s^i}$. В общем случае связь, накладываемая на последнее звено эквивалентной цепи, отнимает у этого звена шесть степеней свободы. Однако возможно, что это звено и до наложения связей обладало меньшим, чем шесть числом степеней свободы и тогда соответствующими условиями связей воспользоваться невозможно.

^{*} Рассматриваемое здесь свойство цепей тесно связано с понятием семейства механизмов, см. И.И.Артоболевский "Теория механизмов и машин", Москва, 1953.

Простым примером такого случая может служить плоский четырехзвенный механизм, показанный на рис. 2.8. Пунктиром показана эквивалентная разомкнутая цепь; механизм располо-

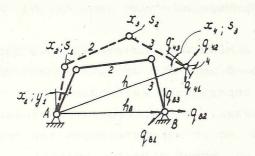


Рис. 2.8

жен в плоскости $(q_2^0$, $q_3^0)$. Очевидно, в этом случае проекции h_B и h на q_1^0 тождественно равны нулю, а векторы q_{B1} и q_{41} всегда коллинеарны независимо от значений \mathbf{x}^1 . Кроме того, если эквивалентная цепь установлена таким образом, что q_{B3} совпадает с q_{43} , то условие $q_{42} \cdot q_{B2} = 1$ удовлетворится автоматически. Следовательно, в данном случае имеются только три независимых условия связи

$$(h_B - h) \cdot q_3^0 = 0$$

 $(h_B - h) \cdot q_2^0 = 0$ (2.13)
 $q_{B3} \cdot q_{43} = 1$

Для замкнутого четырехзвенника на рис. 2.7 множество координат X состоит из четырех элементов. Поскольку положение всего механизма однозначно определяется, если задан только один любой относительный угол, то множество U содержит только один элемент, а множество S , следова. Тельно, три. На рисунке показан один из примеров соответт

ствия между элементами U , S и X .

Таким образом имеется три неизвестные дополнительные координаты $\mathbf{s^1}$, $\mathbf{s^2}$, $\mathbf{s^3}$ и три условия связи (2.13),позволяющие определить $\mathbf{s^i}$ через основную координату $\mathbf{y^1}$. В рассмотренном примере мы имеем минимальное число зависимых координат $\mathbf{s^i}$.

Вернемся к механизму показанному на рис. 2.5 , 2.6 . Здесь множество S состоит из пяти элементов и, следова тельно, для их определения необходимо иметь пять независимых уравнений связи. При перемещениях эквивалентного механизма рис. 2.6 могут изменяться все три проекции вектора h , следовательно, можно записать условия

$$(h_B - h) \cdot q_1^0 = 0$$

 $(h_B - h) \cdot q_2^0 = 0$
 $(h_B - h) \cdot q_3^0 = 0$

Кроме того, должны удовлетворяться условия

$$q_{B3} \cdot q_{n3} = 1$$
 $q_{B2} \cdot q_{n2} = 1$
(2.14)

определяющие угловое положение последнего звена. Что ка сается последнего уславия ${\bf q}_{\rm B1}$ · ${\bf q}_{\rm n1}$ = 1 , то оно удовле творяется автоматически, если удовлетворено (2.14).

Таким образом в различных механизмах может быть различное число независимых уравнений связей и соответству ющее число зависимых координат $\mathbf{s}^{\mathbf{i}}$. Поэтому, необходимо некоторым образом отметить, какие из уравнений связей нужно использовать для данного конкретного механизма.

Введем характеристические числа λ_{r} , $(r=1,2,\ldots,6)$.

Первые три числа (r=1,2,3) соответствуют трем проекциям первого уравнения (2.12) на оси $q_{\mathbf{r}}^{\mathfrak{d}}$, следующие три числа (r=4,5,6) соответствуют аналогичным проекциям второго уравнения (2.12).

Условимся приписывать $\lambda_{ extbf{r}}$ следующие значения

$$\lambda_{\mathbf{r}} = egin{dcases} 1 \text{, всли соответствующая проекция} & (2.12) \\ & \text{должна использоваться как уравнение} \\ & \text{связи} & (2.15) \\ \hline 0 \text{ - в противном случае} & \end{array}$$

Имея произвол в выборе ортов \mathbf{q}_{1}^{0} можно направить их таким образом, чтобы удобно было определить нужные проекции, то-есть числа $\lambda_{\mathbf{r}}$.

Например, если для механизма на рис. 2.8 система координат выбрана совпадаюней с абсолютными ноординатаме (i, j, k), то числа $\lambda_{\mathbf{r}}$ будут равны: 0 ; I ; I ; 0 ; 0 ; I. Для механизма на рес. 2.5 имеем: I ; I ; I ; 0 ; I ; I .

Совокупность чисел $\lambda_{\mathbf{r}}$ можно определить анализируя кинематическую схему замакнутой цепи; такой анализ может бить предворительно выполнен человеком, либо по соответствующему алгоритму, вычислительной машиной. Мы не будем останавливаться на таком алгоритме принципиально он возможен поскольку это отвлечет нас от основных вопросов. В дальнейшем будем считать совокупность чисел $\lambda_{\mathbf{r}}$ известной. Числа $\lambda_{\mathbf{r}}$ помещаются в последние столбцы матрицы STR (см. (1.14)).

Рассмотрим теперь какая существует зависимость между дополнительными координатами **s** и основными координатаме **u** в случае малых перемещений эквивалентной цепи.

Зададим малые перемещения $\Delta x^{f i}$ во всех шарнирах эк-

вивалентной цепи. В резултате последнее звено получит некоторое поступательное и угловое перемещения.

Поступательное и угловое перемещение выше были обо- значены векторами Δh и $\Delta \gamma$ соответственно. Для малых поступательных перемещений можно записать зависимость

$$\sum_{j=1}^{n} a_{j} \Delta x^{j} = \Delta h \qquad (2.17)$$

Векторные коэффициенты а легко найти из следующих рассуждений. Допустим перемещение Δx^j осуществляется только в одном j-ом шарнире. Тогда Δh , вызванное этим перемещением будет:

в случае вращательной ј-ой кинематической пары

$$\Delta h = \Delta x^{j} e_{j} \times r_{nj}$$

в случае поступательной ј-ой кинематической пары

$$\Delta h = \Delta x^{j} e_{j}$$

Подставляя эти выражения в (2.17) и учитывая, что пере - мещения рассматриваются только в одном шарнире получим

Для малых угловых перемещений можно записать

$$\sum_{j=1}^{n} b_{j} \Delta x^{j} = \Delta \gamma$$
 (2.19)

Векторные коэффициенты b; можно определить анало - гичным образом. Пусть перемещение осуществляется только в j-ом шарнире.

Тогда в случае вращательной ј-ой пары

$$\Delta \gamma = \Delta x^{j} e_{j}$$

в случае поступательной ј-ой пары

$$\Delta \gamma = 0$$
.

Подставляя в (2.17) получим

Теперь отвлечемся от уравнений (2.17) - (2.20), опре делим некоторую вспомогательную операцию, которой в даль - нейшем будем часто пользоваться.

Пусть имется два множества С и Е , между элемен - тами которых c_i и ξ_i установлено взаимно однозначное соответствие. Элементами множества С могут быть действительные числа, векторы, матрицы и т.п. Элементами множества Е могут быть только целые числа 0 и I .

Тогда выделим из $\bar{\mathbb{C}}$ подмножество тех элементов, ко - торым в $\bar{\mathbb{E}}$ соответствуют элементы равные единице. Получен - ное подмножество обозначим $\bar{\mathbb{D}}$ и его элементы \mathbf{d}_i проно -

меруем вновь, придерживаясь той последовательности, в кото рой аналогичные элементы были прономерованы в С . То-есть

$$c_{\mathbf{i}} \in D$$
 всли $\xi_{\mathbf{i}} = 1$ (2.21) $c_{\mathbf{i}} \notin D$ всли $\xi_{\mathbf{i}} = 0$

и между элементами с_і и d_і существует соответствие

$$d_j = c_i$$
 где $j = \sum_{k=1}^i \xi_k$

Эту операцию условимся изображать так

$$d = \mathsf{U}(\mathsf{c},\xi) \tag{2.22}$$

где под Ц понимается некоторый оператор, совершающий описанные выше действия.

Соответствующий алгоритм для оператора Ц будет приведен в конце параграфа.

Теперь применим оператор Ц к левым частам уравне - ний (2.17), (2.19) с целью разделить основные и дополнительные координаты.

$$u = \mathsf{L}(x,\xi5) \; ; \; b^u = \mathsf{L}(b,\xi5) \; ;$$

$$s = \mathsf{L}(x,\,\eta) \; ; \; a^S = \mathsf{L}(a,\,\eta) \; ;$$

$$a^u = \mathsf{L}(a,\xi5) \; ; \; b^S = \mathsf{L}(b,\,\eta) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; b^S = \mathsf{L}(b,\,\eta) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(b,\,\eta) \; ;$$

$$\mathsf{L}(b,\,\eta) \; ; \; \mathsf{L}(b,\,\eta) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(b,\,\eta) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(b,\,\eta) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(b,\,\eta) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

$$\mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ; \; \mathsf{L}(a,\xi5) \; ;$$

Параметры $\xi 5_{ extbf{i}}$ описаны в (1.8) . Тогда (2.17) , (2.19) ,

можно записать в виде

$$\sum_{j=1}^{n} a_{j}^{s} \Delta s^{j} = -\sum_{j=1}^{n} a_{1}^{u} \Delta u^{j} + \Delta h$$

$$(2.23)$$

$$\sum_{j=1}^{n} b_{j}^{s} \Delta s^{j} = -\sum_{j=1}^{n} b_{j}^{u} \Delta u^{j} + \Delta Y$$

Спроектируем эти уравнения на оси координат с бази сом q_{j}^{O} (j=1,2,3) . При проектировании возмем только те проекции, для которых соответствующие характеристические числа $\lambda_{r}=1$ (cм(2.15)) .

Подобной операцией мй будем пользоватся неоднократно поэтому остановимся на ней подробнее. Пусть имеются неко торые два вектора а и b и базис $\mathbf{q}_{\mathbf{j}}$ (\mathbf{j} =1,2,3) . Имеется также шесть чисел $\lambda_{\mathbf{k}}$, каждое из которых может принимать только значение 0 или I . Тогда возмем шесть скалярных произведений а и b на $\mathbf{q}_{\mathbf{j}}$ и поставим им в соответствие числа $\lambda_{\mathbf{k}}$

a·q1 , a·q2 , a·q3 , b·q1 , b·q2 , b·q3
$$\lambda_1 \ , \quad \lambda_2 \ , \quad \lambda_3 \ , \quad \lambda_4 \ , \quad \lambda_5 \ , \quad \lambda_6$$

Далее выберем только те скалярные произведения, для которых соответствующее число $\lambda_k=1$. Эти скаларные произведения обозначим $\mathbf{c^i}$ и вновь прономеруем,придерживаясь последованельности их расположения в (2.24) . Легко видеть, что номер \mathbf{i} для $\mathbf{c^i}$ можно определить как

$$i = \sum_{\ell=1}^{k} \lambda_{\ell}$$

Числа ${f c}^{f i}$ будем рассматривать как компоненты некоторого многомерного вектора ${f c}$.

Описанную выше операцию условимся изображать в виде

$$c = \Pi (a,b,q,\lambda)$$
 (2.25)

и под П будем формально понимать некоторый оператор,осуществляющий все описанные выше действия. То-есть оператор П рассчитывает $\mathbf{c}^{\mathbf{i}}$ согласно:

$$\mathbf{c^i} = \left\{ \begin{array}{ll} \mathbf{a} \cdot \mathbf{q_j} & \text{всли} & \mathbf{k} \leqslant 3 \ \Lambda \ \lambda_{\mathbf{k}} = 1 \ ; \\ \\ \mathbf{b} \cdot \mathbf{q_j} & \text{всли} & \mathbf{k} > 3 \ \Lambda \ \lambda_{\mathbf{k}} = 1 \ ; \end{array} \right. \tag{2.26}$$

где

$$i = \sum_{\ell=1}^{k} \lambda_k ;$$

$$j = \begin{cases} k & \text{при } k \leqslant 3 \\ k-3 & \text{при } k > 3 \end{cases};$$

$$k = 1, 2, ..., 6$$

Теперь применим оператор П к ортам і , j , k неподвижной системы координат

$$c_1 = \Pi (i,o,q^0,\lambda) ;$$

$$c_2 = \Pi (j,0,q^0,\lambda) ;$$
 $c_3 = \Pi (k,0,q^0,\lambda) ;$
 $d_1 = \Pi (0,i,q^0,\lambda) ;$
 $d_2 = \Pi (0,j,q^0,\lambda) ;$
 $d_3 = \Pi (0,k,q^0,\lambda) .$

Векторы c_1 , c_2 , c_3 будем рассматривать как столбцы не-которой матрицы $n_{\rm g} \times 3$, которую обозначим S_1

$$B_1 = || c_1 c_2 c_3 || n_{S,3}$$
 (2.27)

Аналогично из векторов d_1 , d_2 , d_3 составим матрицу

$$B_2 = || d_1 d_2 d_3 || n_{S,3}$$
 (2.28)

Легко проверить, что применение оператора Π к двум векторам a , b эквивалентно умножению этих векторов на матрицы B_1 и B_2 .

$$\Pi(a,b,q^{\circ},\lambda) \equiv B_1 a + B_2 b \qquad (2.29)$$

Теперь вернемся к уравнению (2.23) и применим к его коеффициентам оператор П . Согласно (2.29) получим

$$\beta_3 \Delta s = \beta_4 \Delta u + \beta_1 \Delta h + \beta_2 \Delta r \qquad (2.30)$$

где ј-ые столбцы матриц Б₃ и Б₄ определяются выражени ями

$$B_3^{*j} = B_1 a_j^3 + B_2 b_j^S$$
; $(j=1,2,...,n_S)$;

$$5_{4}^{*j} = -5_{1}a_{j}^{u} - 5_{2}b_{j}^{u}$$
; $(j=1,2,...,n_{u})$.

Полученные уравнения представляют собой систему $n_{_{\mathbf{S}}}$ линейных алгебраических уравнений. Решения (2.30) относительно $\Delta \mathbf{s}$ запищем в виде

$$\Delta s = \Lambda \Delta u + \Lambda_1 \Delta h + \Lambda_2 \Delta r \qquad (2.31)$$

где Λ , Λ_1 , Λ_2 матрицы размерности $n_s^{\times}n_u$, $n_s^{\times}3$, $n_s^{\times}3$ соответственно.

Уравнение (2.31) будем називать уравнением связи , а матрицы Λ , Λ_1 , Λ_2 - матрицами связи.

Матрицы Λ , Λ_1 , Λ_2 представляют собой важные характеристики замкнутой цепи. Поэтому для расчета этих матриц составим специальный алгоритм "МАТРИЦЫІ СВЯЗИ".

Прежде всего запишем алгоритм для осуществления операций, которые выше были обозначены как оператор U . Алгоритм оформлен как процедура КОМП. Формальный параметр с обозначает матрицу $\mathsf{n} \times \mathsf{m}$, к строкам или столбцам которой применяется оператор U . В зависимости от значения булевской переменной B возможны два режима работы.

Допустим В \equiv true , тогда алгоритм обрабатывает множество строк матрицы С . В соответствии с описанием оператора Ц (см.(2.21, 2.22)) выделяются строки, которым соответствуют значения $\xi=1$. С целью экономии оперативной памяти эти строки вновь помещаются в матрицу С подряд начиная с первой строки. те строки матрицы С , которым до обработки соответствовало $\xi=1$ также остаются в матрице С , однако, сдвигаются и располагаются вслед за строками , для которых $\xi=1$. Так например, если множество ξ было $\{0,1,0,1,0\}$, а множество С было $\{a,b,c,d,e\}$, где под элементами a,b . . . следует понимать строчки матрицы, то после выполнения алгоритма множество С будет иметь вид

 $\{b,d,a,c,e\}$. Таким образом этот алгоритм осуществляет перекомпоновку матрицы, и соответствующая процедура названа КОМП. Если В \equiv false, то алгоритм аналогичным образом обработывает множество столбцов матрицы С.

```
procedure ΚΟΜΠ (c,ξ,n,m,B);
         value n,m,B; integer n,m;
boolean Β; array C; integer array ξ;
begin integer i,j,k,nl,ml;
         if B then begin nl: = n; ml: = m end
              else begin nl: = m; ml: = n end
         j: = 0; <u>for</u> i: = 1 <u>step</u> 1 <u>until</u> n <u>do</u>
              j:j+\xi[i]; k: = nl-j;
         begin array d[1:j,1:ml],
              e[1:k,1:m1], c1[1:n1,1:m1];
         for i: = 1 step 1 until nl do
         for j: = 1 step 1 until ml do
             \underline{if} B \underline{then} cl[i,j] := c[i,j]
             \underline{\text{else}} \ \text{cl[i,j]: = c[j,i];}
         p: = j: = 0;
         for i: = 1 step 1 until nl do
              if \xi[i] = 1 then
              for k: = 1 step 1 until ml do
              begin j: = j+1; d[j,k]: = cl[i,k] end
              else for k: = 1 step 1 until ml do
              begin p: = p+1; e[p,k]: = cl[i,k]
              end:
         for k: = 1 step 1 until ml do begin
         for i: = 1 step 1 until j do cl[i,k]: = d[i,k];
         for i: = 1 step until p do cl[j+i,k]: = e[i,k] end;
         for i: = 1 step 1 until nl do
         for j: = 1 step 1 until ml do
```

```
<u>if</u> B <u>then</u> c[i,j]: = cl[i,j]

<u>else</u> c[j,i]: = cl[i,j]

<u>end</u> end процедуры;
```

Теперь запишем алгоритм выполняющий действия, которые выше были условно обозначены как оператор П. Этот алторитм /процедура МАСКА/ построен в соответствии с (2.24) столбцы массива ${\bf q}$ определяют векторы ${\bf q}_1^{\rm O}$, ${\bf q}_2^{\rm O}$. Вектор ${\bf q}_1^{\rm O}$ введен в алгоритм только для того, чтобы обратиться к процедуре СКЛ /скаларное произвенение/.

```
procedure MACHA (a,b,q,\lambda,c);
array a,b,q,c; integer array λ;
begin
         real p; array q1[1:3];
         integer i,j,k,l;
         i: = 0:
         for k: = 1 step 1 until 6 do
         begin if \lambda[k] = 0 then go to ml to ml else
                                              i := 1 + \lambda \lceil k \rceil
             if k < 3 then j: = k else j: = k-3;
              for 1: = 1,2,3 do
              q1[1]: = q[1,j];
             if k < 3 then
             СКЛ (p,ql,a)
            else СКЛ (p,ql,b);
              c[i]: = p;
         ml: end
```

Теперь переходим к алгоритму расчета матриц связи Λ , Λ_1 , Λ_2 и вектора

$$s = s_0 + \Delta s$$

end процедуры;

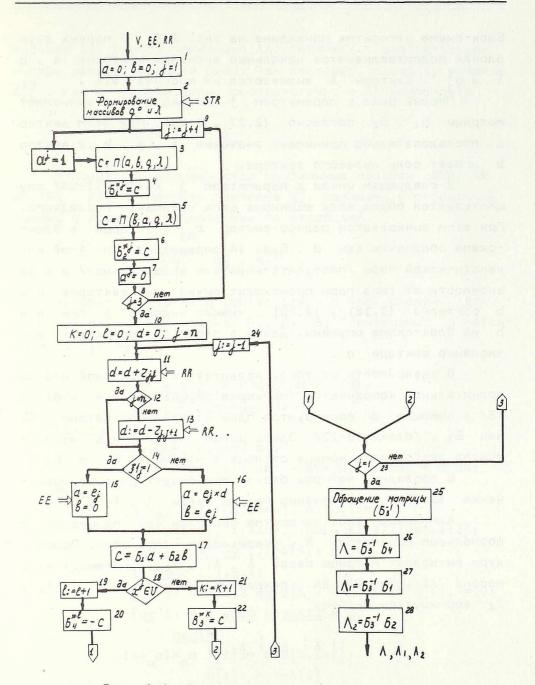


Рис. 2.9. Блок-схема алгоритма матрицы связи

Блок-схема алгоритма приведена на рис. 2.9. В первых двух блоках подготавливаются начальные значения векторов a , b λ и q_1^0 . Векторы λ вызываются из массива STR .

Первый цикл с параметром j /блоки 3-9/ вычисляет матрицы $\mathbf{5}_1$, $\mathbf{5}_2$ согласно (2.27 , 2.28). При этом вектор а последовательно принимает значения \mathbf{i} , \mathbf{j} , \mathbf{k} ,а вектор \mathbf{b} играет роль нулевого вектора.

В следующем цикле с параметром ј /блоки II-24/ осуществляется обход всех шарниров цепи начиная с последнего. При этом вычисляется радиус-вектор \mathbf{r}_{nj} , который в блок-схеме обозначен как \mathbf{d} . Блок 14 определяет тип ј-ой кинематической пары /поступательная или вращательная/ и в за висимости от типа пары происходит вычисление векторов \mathbf{a} и \mathbf{b} согласно (2.18), (2.20); нижние индексы ј при \mathbf{a} и \mathbf{b} на блок-схеме опущены. Далее в блоке 17 происходит вычисление вектора \mathbf{c} .

В зависимости от того, является $\mathbf{x}^{\mathbf{j}}$ основной или дополнительной координатой /проверка осуществляется в блоке 18/ с помощью \mathbf{c} формируется один из столбцов матриц $\mathbf{b}_{\mathbf{i}}$ или $\mathbf{b}_{\mathbf{3}}$. /блоки 19-22/ Здесь целые числа ℓ и k используются для расчета номера столбца в матрицах $\mathbf{b}_{\mathbf{i}}$ и $\mathbf{b}_{\mathbf{3}}$.

В последних четырех блоках происходит решение урав - нения (2.30) и вычисление матриц связи Λ , Λ_1 , Λ_2 .

Алгольный текст алгоритма оформлен как процедура с формальным параметром N , указывающем номер цепи. Проце - дура вычисляет матрицы свази Λ , Λ_1 , Λ_2 и помещает их в массив $\Lambda\Lambda$. Массив $\Lambda\Lambda$ трехмерный. Из матриц Λ , Λ_1 , Λ_2 формируется матрица

$$| | \Lambda | \Lambda_1 | \Lambda_2 | | n_s \times (n_u + 6)$$

и эта матрица помещается по второму и третьему измерениям

массива $\Lambda\Lambda$. Первое измерение $\Lambda\Lambda$ /"строки"/ соответствует номеру рассматриваемой цепи. Так например, элементу матрицы $\Lambda^{ ext{ij}}_{1k}$, где k - номер цепи, соответствует в $\Lambda\Lambda$ компонента

$$\Lambda\Lambda[k,i,nu+j]$$

В процедуре используются глобальные массивы STR, KP , EE, RR , $\Lambda\Lambda$. Компоненты векторов $\mathbf{q_i^O}$ помещены в глобальный массив \mathbf{q} /векторы расположены по столбцам/

$$q = ||q_1^0, q_2^0, q_3^0||$$
.

Алгоритм построен в соответствии с блок схемой. Для экономии памяти ЭВМ массивы ${\sf E_1},\,{\sf E_2},\,{\sf E_4}$ "совмещены" с массивами ${\sf A_1},\,{\sf A_1},\,{\sf A_2}$

```
procedure MATPИЦы СВЯЗИ (N);
value N; integer N;
```

begin

<u>integer</u> n, ns, nu; n: = STR[N,6]; nu: = STR[N,7]; ns: = n-nu;

begin

array 53[1:ns,1:ns], c1,c[1:ns], a,b,d[1:3], λ [1:ns,1:nu], Λ 1, Λ 2[1:ns,1:3]; integer array ξ , λ [1:6]; integer i,j,k,1; for i: = 1,2,3 do b[i]: = a[i]: = d[i]: = 0; for i: = 1 step 1 until 6 do begin λ [i]:= STR[N,i+8]; ξ [i]: = 1- λ [i]

end;

```
ИМЯ (B1,\Lambda1); ИМЯ (Б2,\Lambda2); ИМЯ (Б4,\Lambda);
for j: = 1,2,3 \text{ do}
begin a[j]: = 1;
MACHA (a,b,q,\lambda,c);
for i: = 1 step 1 until ns do
     51[i,j] : = c[i];
MACHA (b,a,q,\xi,c);
for i: = 1 step 1 until ns do
52[i,j] : = c[i];
a[j]: = 0
end j;
k := 1 := 0;
for j: = n step -1 until 1 do
begin
for i: = 1,2,3 \text{ do}
begin a[i]: = EE[N,j,i]; b[i]: = 0;
       d[i] := d[i] + RR[N,j,l,i];
\underline{if} j\neq n \underline{then} d[i]:=d[i]-RR[N,j,2,i] \underline{end}
if KP[N,j,1] = 0 then begin
       for i: = 1,2,3 \text{ do } b[i]: = a[i];
       BEHT (a,a,d) end
MTB (c,B1,a,3,3);
MTB (c1,B2,b,3,3);
if KP[N,j,3]: = 1 then
begin 1: = 1+1;
for i: = 1 step 1 until ns do
       54[i,1] : = -(c[i]+cl[i]) \underline{end}
else begin k: = k+1;
for i: = 1 step 1 until ns do
       53[i,k] : = c[i] + cl[i] end
 end j;
 OBP (63,63);
```

<u>end</u> end процедуры;

§4 "Сборка" и определение положений для замкнутых цепей

При решении задачи сборки и определения положения замкнутой цепи будем рассматринать эквивалентную разомкнутую цепь и путем последовательных итерраций приводить эквивалентную цепь в положение, удовлетворяющее условиям замыкания, полученным в предыдущем параграфе. В этом случае можно воспользоваться уже имеющимся алгоритмом ПОЛОЖЕНИЕ.

Будем предполагать, что известны значения основных координат u и векторы h_B , q_{Bj} ($j{=}1,2,3$), которые характеризуют положение замкнутой цепи. Значения дополнительных координат s обычно можно предсказать c некоторой точностью. Будем считать, что для s известны приближенные значения. Совокупность начальных значений u_0 , s_0 обозначим x_0 .

Требуется найти "точные" значения s , удовлетворяющие условиям замыкания (2.12) .

Для этого "собираем" эквивалентную цепь с помощью алгоритма СБОРКА , а затем перемещаем согласно значениям \mathbf{x}_{0} с помощью алгоритма ПОЛОЖЕНИЕ . После выполнения этих процедур известны единичные векторы осей е , векторы из центров шарниров к центрам масс соседних звеньев \mathbf{r} и базисные векторы \mathbf{q} для всех звеньев в положении цепи , сотответствующем значениям относительных координат \mathbf{x}_{0} .

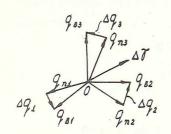
Теперь можно рассчитать вектор h между полюсами крайних эвеньев эквивалентной цепи

$$h = r_{n,n} - r_{0,1} + \sum_{j=1}^{n} (r_{j,j} - r_{j,j+1})$$

и затем отклонения Δh и Δq , которые характеризуют разницу в расположении базового эвена и последнего звена эквивалентной цепи.

$$\Delta h = h_B - h$$
;
 $\Delta q_j = q_{Bj} - q_{nj}$; (j=1,2,3)

Теперь определяем вектор углового перемещения $\Delta \Upsilon$ (см.рис. 2.10). Этот вектор /см. предыдущий параграф/ определяет



то угловое перемещение, которое нужно сообщить последнему звену эквивалентной цепи, для того чтобы отклонения Δq_j стали равными нулю. Воспользовавшись малостью Δq_j можно записать

$$\Delta q_{\ell} = \Delta \gamma \times q_{n\ell}$$

$$\Delta q_{r} = \Delta \gamma \times q_{nr}$$
(2.33)

Рис. 2.10

где Δq_ℓ и Δq_r два любые не равные нулю приращения из совокупности Δq_j (j=1,2,3). Очевидно, что если q_{nj} не совпадают с q_{Bj} (j=1,2,3), то по крайней мере два приращения из Δq_j отличны от нуля.

Согласно (2.33) вектор $\Delta \gamma$ перпендикулярен $\Delta q \mathcal{L}$ и $\Delta q_{\mathbf{r}}$ и, следовательно, параллелен их векторному произведению.

Обозначим единичный вектор, направленный вдоль $\Delta\gamma$ через с .

$$\Delta \Upsilon = \theta c ; \theta = |\Delta \Upsilon|$$
 (2.34)

Тогда можно записать

$$c = \alpha \frac{\Delta q_{\ell} \times \Delta q_{r}}{|\Delta q_{\ell} \times \Delta q_{r}|}$$
 (2.35)

где α - постоянный множитель, который может принимать зна чения ± 1 .

Множитель α определим из следующих соображений. Вектор $\Delta \gamma$ должен иметь такое направление, чтобы поворот всех векторов \mathbf{q}_{nj} происходил против часовой стрелки, если смотреть из конца $\Delta \gamma$. Для этого необходимо чтобы выполнялось условие

$$(q_{n\ell} \times \Delta q_{\ell}) \cdot c > 0$$

Подставляя сюда с из (2.35) получим

$$\alpha(q_{n\ell} \times \Delta q_{\ell}) \cdot (\Delta q_{\ell} \times \Delta q_{r}) > 0$$

откуда

$$\alpha = \text{sign} \left[(q_{n\ell} \times \Delta q_{\ell}) \cdot (\Delta q_{\ell} \times \Delta q_{r}) \right] \qquad (2.36)$$

формула (2.35) позволяет определить с во всех случаях кроме $\Delta q_{\ell} \mid \Delta q_r$. Эти векторы могут быть параллельны только, если вектор $\Delta \gamma$ лежит в плоскости векторов $q_n \ell$ и $q_n r$; тогда оба приращения Δq_{ℓ} и Δq_r перпендикулярны указанной плоскости. Но в этом случае $\Delta \gamma$ перпендикулярен третьему орту, который обозначим $q_n s$, и следовательно*,

^{*} Вектор ΔY не может быть перпендикулярен ${
m q}_{
m ns}$ одновременно одному из векторов ${
m q}_{
m n\ell}$ или ${
m q}_{
m nr}$, поскольку

приращение Δq_s должно по модулю быть больше и Δq_ℓ и Δq_r . Очевидно справедливо и обратное утверждение, что если среди Δq_ℓ и Δq_r находится максимальное по модулю приращение, то векторы Δq_ℓ и Δq_r не могут быть параллельны . Следовательно, (2.35) можно всегда пользоватйся, если предположить, что одно из приращений входящих в эту формулу/на пример Δq_ℓ / максимально по модулю

$$|\Delta q_{\ell}| = \max(|\Delta q_{j}|)$$
; $j=\ell,r,s$
(j)

Подставляя (2.34) в первое уравнение (2.33) опреде - ляем множитель

$$\theta = \frac{|\Delta q_{\ell}|}{|c \times q_{\ell}|} \tag{2.37}$$

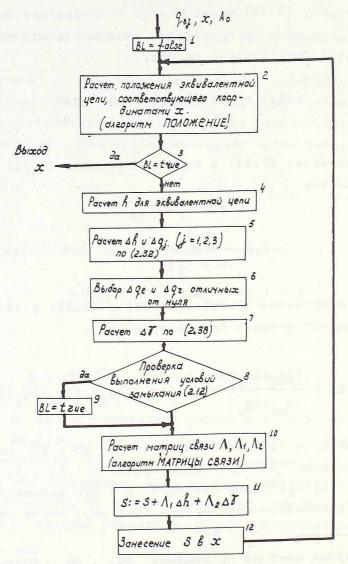
Наконец, подставляя (2.35) , (2.36) и (2.37) в (2.34) определяем искомое угловое перемещение $\Delta \Upsilon$.

$$\Delta \gamma = \frac{\Delta q_{\ell} \times \Delta q_{r}}{|q_{n\ell} \cdot \Delta q_{r}|} \text{ sign } [(q_{n\ell} \times \Delta q_{\ell}) \cdot (\Delta q_{\ell} \times \Delta q_{r})] \quad (2.38)$$

Теперь, когда известны значения Δh и $\Delta \Upsilon$ можно с помощью процедуры МАТРИЦы СВЯЗИ найти Λ , Λ_1 , Λ_2 , и затем по (2.31) требуемые значения ${\bf x}$. Эти значения ${\bf x}$ определены в линейном приближении. Поэтому, после того как

в этом случае одно из приращений Δq_ℓ , Δq_r должно быть равно нулю, что противоречит принятым условиям относительно Δq_ℓ и Δq_r .

эквивалентную цепь переместим в положение, соответствующее ${\bf x}$, условия (2.12) могут оказаться неудовлетворенными с требуемой точностью. Тогда нужно присвоить ${\bf x}_0$ значения ${\bf x}$ и повторить весь процесс.



Функциональная блок-схема алгоритма ЗАМКНУТАЯ ЦЕПЬ Рис. 2.11

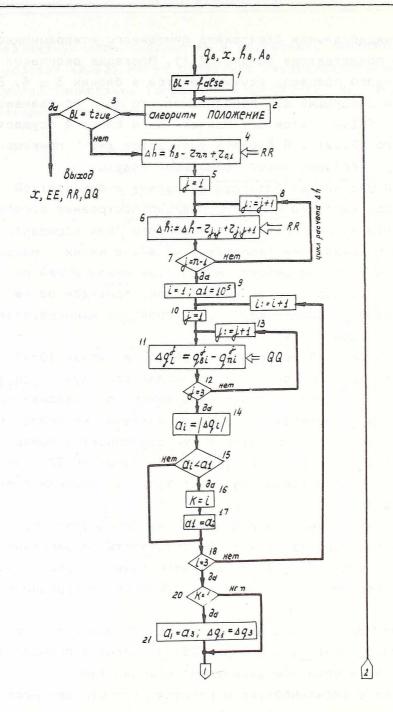
функциональная блок-схема описанного итеррационного процесса представлена на рис. 2.11. Проверка окончания итеррационного процесса осуществляется в блоках 3 и 8. Если требуестя следующий шаг приближения, то BL присваивается значение false /блок 9/. Расчет h в блоке 4 осущест вляется по (2.32) . В блоке 6 выбирается два приращения Δq_{ρ} и Δq_{r} , которые имеют найбольший модуль.

функциональная блок-схема сделана очень краткой для того, чтобы наглядно показать принцип построения алгоритма. Однако, при этом многие частные вопросы /как проверять условия замыкания, как распределять s, в каких массивах хранится нужная информация и т.п./ не нашли в ней отражения. Поэтому, наряду с функциональной, приведем более детальную рабочую блок-схему, по которой уже непосредственно строится алгоритм /рис. 2.12/.

В блоках 6-8 вычисляется Δh , а в блоках 10-13 вычисляются Δq_i . Одновременно вычисляются модули $|\Delta q_i|$ и помещаются в массив а . Значение которого присваиваится /блок 14/ и выбирается наименьший, значение которого присваивается а1 . При этом в k запоминается номер i наименьшего из приращений $|\Delta q_i|$. В блоках 20-22 происходит выбор двух векторов Δq_i (i=1,2,3) с большими эначе - ниями модуля.

Расчет углового перемещения $\Delta \Upsilon$ осуществляется в блоке 23 согласно (2.34 — 2.38) . Точность приближения оцениваестя в блоке 24 и в случае, если точность достаточна в блоке 25 формируется команда на окончание итеррационного процесса.

Последний цикл с параметром і /блоки 29-33/ служит для распределения s . В блоке 29 проверяется принадлеж - ность і-ой координаты "основным" координатам. Если і-я координата к "основной" не относится то-есть является "до



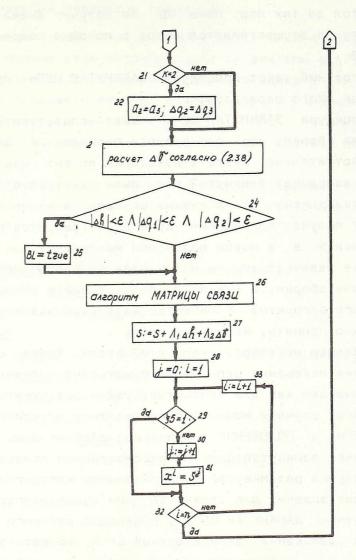


Рис. 2.12. Рабочая блок-сжема алгоритма ЗАМКНУТАЯ ЦЕПЬ

полнительной"/, то вычисляется значение индекса ј и координате $\mathbf{x^i}$ присваивается значение $\mathbf{s^j}$.

После этого процесс вновь возвращается к блоку 2 и

повторяется до тех пор, пока BL не получит значения true. В этом случае осуществляется выход с помощью условного оператора 3.

Алгольный текст процедуры ЗАМКНУТАЯ ЦЕПЬ приводится в конце этого параграфа.

Процедура ЗАМКНУТАЯ ЦЕПЬ может использоваться не только для сборки, но и для расчета перемещений замкнутой цепи. Действительно, пусть в некоторый момент времени t_1 значения координат замкнутой цепи были \mathbf{x}_0 . Спустя малый промежуток времени Δt значения основних и кинематических координат получат приращения Δy и Δr . Требуется определить значения \mathbf{s} в новом положении механизма. Нак видно, постановка задачи здесь не отличается от сформулированной выше задачи сборки, и она может быть решена с помощью разработанного алгоритма. В качестве начальных значений \mathbf{s} здесь можно принять, например, $\mathbf{s} = \mathbf{s}_0$.

Падведем некоторые итоги этой главы. Задачи сборки и определения положения цепи могут решаться в значительной части одинаково как для замкнутых, так и разомкнутых це глей. В обоих случаях можно воспользоваться алгоритмами СБОРКА ЦЕПИ и ПОЛОЖЕНИЕ , и разница состоит лишь в том, что в случае замкнутой цепи мы рассматриваем эквивалентную цепь. В случае разомкнутой цепи указанные алгоритмы полностью решают задачи. Для замкнутой цепи нужно воспользовать ся алгоритмом ЗАМКНУТАЯ ЦЕПЬ , с помощью которого осуще ствляется "замыкание" эквивалентной цепи, то-есть опреде ляются значения дополнительных координат s , удовлетворяющих условиям связи. Одновременно последний алгоритм определяет матрицы связи Λ , Λ_1 , Λ_2 , которые играют важную роль при анализе кинематики в следующей главе.

В заключение приведем текст процедуры ЗАМКНУТАЯ ЦЕПЬ которая осуществляет расчет положения замкнутого механизма согласно опиранному алгоритму.

Формальные параметры процедуры:

N - номер цепи; A2 - матрица перехода для конечного базового звена /с помощю этой матрицы задаются векторы $q_{Bj}/;\ hb$ - вектор из центра масс начального базового звена к центру масс конечного /соответствует h_B в формулах/; ϵ - малая величина, задающая точность вычислений.

Глобальные идентификаторы STR, KP, EE, RR, QQ, E0, R0, Q0, X, $\Lambda\Lambda$.

В результате выполнения процедуры определяются значения всех относительных координат ${\bf x}$, и соответствующие значения ${\bf e_i}$, ${\bf r_{ij}}$, ${\bf A_i}$, ${\bf A_i}$, ${\bf A_1}$, ${\bf A_2}$.

```
procedure 3ΑΜΚΗΥΤΑЯ ЦΕΠЬ (N,A2,hb,ε);
value N; integer N;
array A2, hb; real ε;
begin
      array \Delta h, \Delta \gamma, a, b[1:3], \Delta q[1:3,1:3], c,d[1:3];
      integer i,j,n,nu,ns;
      real al, a2, a3;
      boolean BL;
      BL: = false;
      n: = STR[N,6]; nu: = STR[N,7];
      ns: = n-nu;
begin array h, \gamma[1:ns], \Lambda[1:ns,1:nu], \gamma[1:ns]
         \Lambda 1, \ \Lambda 2[1:ns,1:3];
ml: ПОЛОЖЕНИЕ N
if BL then go to m2;
for i: = 1, 2, 3 do
\Delta h[i] := hb[i] - RR[N,n,l,i] + RR[N,0,2,i];
for j: = 1 step 1 until n-1 do
for i: = 1,2,3 do
\Delta h[i] := \Delta h[i] + RR[N,j,l,i] - RR[N,j,2i];
```

```
al: = 10^5;
 for i: = 1,2,3 do begin a[i]: = 0;
 for j: = 1,2,3, do begin
        \Delta d[j,i] := A2[j,i] - QQ[N,n,j,i];
        a[i] := a[i] + \Delta q[j,i] + 2 end;
 if a[i] < al then
 begin al: = a[i]; k: = i end
 end i;
 if k = 1 then begin a[1] := a[3];
 for i: = 1,2,3 do
     \Delta q[1,i] := \Delta q[3,i] \text{ end};
 if k = 2 then begin a[2]: = a[3];
        for i: = 1, 2, 3 do
        \Delta q[2,i] := \Delta q[3,i] \text{ end};
for i: = 1,2,3 do
 begin d[i] := \Delta q[1,i];
     b[i]: = \Delta q[2,i];
   c[i] := QQ[N,n,l,i]
 end;
 BEHT (\Delta \gamma, d, b);
 BEHT (d,c,d);
 СКЛ (al,\Delta\gamma,d);
 al: = sign (al);
СКЛ (a2,b,c);
 a2: = a2 \times sign (a2);
 for i: = 1, 2, 3 do
 \Delta \gamma [i] = al \times \Delta \gamma [i] / a2
 al: = sqrt(a[1]); a2: = sqrt(a[2]);
 a3: = MOJ(\Delta h);
 if a1 < \epsilon \Lambda a2 < \epsilon \Lambda a3 < \epsilon then
 BL: = true;
 for i: = 1 step 1 until ns do
```

```
begin
```

m2:

Гл. ІІІ КИНЕМАТИКА ПРОСТРАНСТВЕННЫХ ЦЕПЕЙ

§1 Определение скоростей

При анализе кинематики могут нозникнуть задачи двух типов. Пярвая задача ставится следующим образом. Имеется некоторая цепь /разомкнутая или замкнутая/, для которой известни значения относительных скоростей $\dot{\mathbf{u}}$ в шарнирах, принятых за основные. Требуется определить угловые скорости всех звеньев цепи $\mathbf{w_i}$ и линейные скорости их центров масс $\mathbf{v_i}$. Для замкнутых цепей подлежат определению также значения относительных скоростей $\mathbf{s_i}$ в дополнительных шарнирах.

Предполагается, что для рассматриваемого механизма уже решена задача определения положений /см. предшествующую главу/ и известны все векторы e и r , а для замкнутых цепей также и матрецы связи Λ , Λ_1 , Λ_2 .

Вторая задача в некотором смысле обратна первой. Здесь будем считать известными для некоторого звена цепи угловую (ω) и поступаетельную (v) скорости. Определению подлежат относительные скорости \dot{x} в шарнирах, которые обеспечивают заданые значения ω и v.

Рассмотрим сначала первую задачу. В процессе её реше-

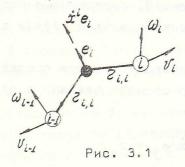
ния мы будем пользоваться двумя известными соотншениями механики, что угловая скорость тела равна сумме его переносной и относительной скорости

и что для двух точев А и В, принадлежащих телу справедливо

$$v_B - v_A = \omega \times r_{AB}$$

где ω - угловая скорость тела; \mathbf{r}_{AB} - вектор с началом в точке A и концом в точке B .

Рассмотрим i-ю кинематическую пару, образованнуя i-ым и /i-1/-ым звеньями /рис. 3.1/. Движение /i-1/-го звена бу-



дем рассматривать как переносное, а движение в i-ом шарнире
- как относительное. Тогда справедливы соотношения

в случае поступательной кинематической пары

$$w_{i} = w_{i-1}$$
, /3.7/
 $v_{i} = v_{i-1} - w_{i-1} \times r_{i-1,i} + \dot{x}^{i} e_{i}$;

в случае вращательной кинематической пары

$$w_{i} = w_{i-1} + \dot{x}^{i}e_{i}$$

$$v_{i} = v_{i-1} - w_{i-1} \times r_{i-1,i} + w_{i} \times r_{i,i}$$
/3.2/

Выражения /3.1, 3.2/ можно рассматривать как рекуррентные: если последовательно обходить все звенья цепи от начального базового, то по этим соотношениям можно рассчитать скорости всех звеньев.

Поскольку для разомкнутой цепи известны все $\dot{\mathbf{x}}^{\mathbf{i}}$, то та-ким образом первая задача может быть решена полностью. Для замкнутой цепи известна только часть относительных скоростей, а именно $\dot{\mathbf{y}}$ и $\dot{\mathbf{z}}$. Недостающие значения $\dot{\mathbf{s}}$ необходимо определить из условий связи.

Уравнения связи /2.31/ записаны для молых угловых и линейных перемещений. Однако, аналогичные соотношения справедливы для скоростей, причем, в этом случае снимается требование об их "малости".

Действительно, скорость полюса последнего звена эквивалентной цепи следующим образом связана с относительной скоростью в j-ом шарнире

$$(v_n)_j = \dot{x}^j e_j \times r_{nj}$$

Сравнивая с /2.17/ видим, что для линейных скоростей справедливо соотношение

$$\sum_{j=1}^{n} a_{j} \dot{x}^{j} = v_{n},$$

где векторные коэффициенты а, выражаются /2.18/.

Аналогично для угловой скорости последнего звена эквивалентной цепи справедливо

$$\sum_{j=1}^{n} b_{j} \dot{x}^{j} = \omega_{n},$$

где b_і выражается /2.20/.

Отсюдя следует, что для скоростей справедливы уравнения, аналогичные /2.31/, а именно

$$\dot{s} = \Lambda \dot{u} + \Lambda_1 (v_n - v_o) + \Lambda_2 (\omega_n - \omega_o)$$
 /3.3/

где Λ , Λ_1 , Λ_2 уже известные матрицы свзи; $\mathbf{v_0}$, ω_0 и $\mathbf{v_n}$, ω_n - скорости начального и конечного базовых звеньев. В тех случаях, когда цепь замкнута на стойку, соответствующие скорости $\mathbf{v_0}$, ω_0 , $\mathbf{v_n}$, ω_n будут равны нулю.

После того, как для замкнутой цепи по /3.3/ определены все s, задача расчета скоростей решается для эквивалентной разомкнутой цепи. Блок-схема алгоритма определения скоростей приведена на рис. 3.2.

Комментарий к блок-схеме алгоритма СКОРОСТИ. В блоке 1 проверяется тип цепи /разомкнутая или замкнутая/ Последующие блоки 2-15 работают в случае замкнутой цепи, определяют значения в и присваивают их соответствующи компонентам вектора х . В блоках 2-7 проверяется принадлежность относительных координат к той или иной группе и формируется вектор и. Здесь і-номер кинематической пары, ј- номер компоненты и в блоке в согласно с /3.3/ вычисляются s.

В блоках 9-14 /цикл по $\mathbf{i}/$ осуществляется распределение $\dot{\mathbf{s}}^{\mathbf{j}}$ среди соответствующих компонент $\dot{\mathbf{x}}^{\mathbf{i}}$.

Собственно расчет скоростей начинается с блока 16. В случае разомкнутой цепи мы, минуя всю предшествующую часть алгоритма, попадаем непосредственно к этому блоку. Расчет линейных и угловых скоростей осуществляется в блоках 17, 19, 20 согласно формулам /3.1/, /3,2/.

Для работы процедуры необходимо, чтобы предварительно были известны значения \mathbf{v}_{o} и ω_{o} для начального базового звена, а в случае замкнутой цепи, также и значения \mathbf{v}_{n} и ω_{n} для конечного базового звена.

В процедуре СКОРОСТИ осуществляется расчет $v_{\mathtt{i}}$ и $\omega_{\mathtt{i}}$ согласно описанному алгоритму. Формальные параметры: N - номер

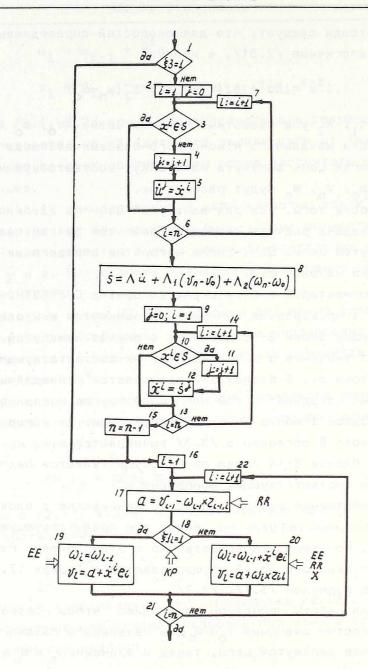


Рис. 3.2. Блок-схема алгоритма определения скоростей

цепи; v_0 , $v_{ t p}$ - поступательные скорости начального и конечного базовых звеньев; ω_0 , $\omega_{ t p}$ - угловые скорости базовых звеньев.

Глобальные идентификаторы: STR, KP, EE, RR, $\Lambda\Lambda$, X.

Обозначениям \dot{s} в формулах соответствует идентификатор ds в прогоамме. Векторы a, b, c, d используются как вспомогательные при вычислениях.

В результате выполнения процедуры СКОРОСТЬ полностью заполняются матрицы VV и $\Omega\Omega$, то-есть рассчитываются угловые скорости и линейные скорости центров масс всех звеньев механизма.

```
procedure CHOPOCTH (N, vo, wo, vn, wn);
value N; integer N;
array vo, wo, vn, wn;
begin
        array a,b, c, d [1:3];
        integer n, ns, nu, i,j;
        n: = STR[N,6]; nu: = STR[N,7];
if STR[N,1]=1 then go to m,;
        ns: = n-nu;
begin
        array ds[1:n<sub>c</sub>], du[1:nu],c,c<sub>1</sub>[1:ns],
        \Lambda[1:ns,1:nu], \Lambda_1, \Lambda_2[1:ns,1:3];
        j := 0;
for i: = 1 step 1 until n do
        if KP[N,i,3]=1 then
        begin j:=j+1;
        du[j]:=X[N,i,2] end;
for j: = 1 step 1 until ns do
begin
       for j: = 1 step 1 until nu do
            \Lambda[i,j] := \Lambda\Lambda[N,i,j];
```

```
for j:=1,2,3 do begin
                 \Lambda_1[i,j]: = \Lambda\Lambda[N,i,j+nu];
               \Lambda_2[i,j] := \Lambda \Lambda[N,i,j+nu+3]
    end end i;
  for i: = 1, 2, 3 do
             begin vn[i] := vn[i] - vo[i];
             \omega n[i] := \omega n[i] - \omega o[i] end;
    MTB(d, A, du, ns, nu);
   MTB(c, \Lambda_1, vn, ns, 3);
    MTB(c_1, \Lambda_2, \omega n, ns, 3);
    for j: = 1 step 1 until ns do
             ds[j]:=ds[j]+c[j]+cl[j];
    comment конец расчета s;
             j := 0;
    for i: = 1 step 1 until n do
             if KP[N,i,3] \neq 1 then
             begin j: = j+l;
             X[N,i,2] := ds[j] end
     end блока;
             n:=n-1;
     comment переходим к расчету скоростей;
m_1: for i: = 1,2,3 do
             begin VV[N,0,i] := v_0[i];
             \Omega\Omega[N,0,i] := \omega [i] end;
     for i: = 1 step 1 until n do
     begin
              for j: = 1,2,3 do begin
     if KP[N,i,1]=1 then \Omega\Omega[N,i,j]:=\Omega\Omega[N,i-1,j]
              else \Omega\Omega[N,i,j] := \Omega\Omega[N,i-1,j] + dx[i]xEE[N,i,j];
              a[j] := \Omega\Omega[N,i-1,j]; b[j] := \Omega\Omega[N,i,j];
              c[j] := RR[N, i-1, 2, j]; d[j] := RR[N, i, 1, j] end;
     BEKT (a,a,c); BEKT(b,b,d);
     for j := 1,2,3 do c[j] := VV[N,i-1,j]-a[j];
```

Теперь переходим к решению второй задачи, сформулированной в начале этого параграфа. Пусть имеется разомкнутая

цепь, и последнее звено этой цепи должно иметь скорости $\mathbf{v_n}$ и $\mathbf{\omega_n}$. Требуется определить $\dot{\mathbf{x}}$ обеспечивающие такие скорости. При этом будем предполагать, что заданные скорости $\mathbf{v_n}$ и $\mathbf{\omega_n}$

возможны для рассматриваемой цепи.

Эта задача может быть сведена к уже рассмотренной выше первой задаче для замкнутой цепи. Заданные скорости можно условно россматривать как скорости конечного базового звена. Тогда для такой цепи справедливо уравнение /3.3/. Однако, прежде чем им воспользоваться необходимо решить вопрос какие из относительных координат считать "основными" /ů/ и какие "дополнительными" /s/.

Для этого вернемся к исходному матричному уравнению /2.31/. Очевидно, матрицы связи Λ , Λ_1 , Λ_2 зависят от того, какие из отнтсительных координат выбраны за основные, но число строк этих матриц не зависит от выбора основных координат. Это число строк /оно было обозначено n_s / зависит от числа единиц среди характеристических чисел λ_r /см./2.15//, а последние определяются структурой цепи.

Таким образом матричное уравнение /3.3/ определяет $\mathbf{n_s}$ независимых условий, связывающих относительные скорости в шарнирах цепи.

В данной задаче рассмотрим три случая, когда число шар ниров цепи n равно $n_{\rm s}$, или $n < n_{\rm s}$, или $n > n_{\rm s}$.

- I. Допустим $\mathbf{n}=\mathbf{n_s}$. Тогда имеется столько относительных скоростей сколько условий связи. В этом случае можно считать, что все шарниры в замкнутой цепи "дополнительные", и отнтсительные скорости можно непосредственно определить из уравнения /3.3/, положив в нем $\hat{\mathbf{u}}=\mathbf{0}$.
- 2. Пусть \mathbf{n} < $\mathbf{n}_{\mathbf{s}}$. Тогда имеется больше уравнений связи, чем относительных координат. Такой случай невозможен, если только $\lambda_{\mathbf{r}}$ определены правильно. Действительно, характеристические числа $\lambda_{\mathbf{r}}$ указывают вдоль каких направлений возможны независимые угловые и линейные перемещения последнего звена /см. гл. II, §3/. фактически $\mathbf{n}_{\mathbf{s}}$ указывают число степеней свободы последнего звена разомкнутой цепи, и это число заведомо не может быть больше числа шарниров .
- 3. Последний случай $n > n_s$ воэможен. Действительно, последнее звено может обладать не более, чем шестью степенями свободы, а число шарниров цепи может быть больше. Тогла число уравнений $/n_s/$ оказывается меньше числа неизвестных и, для того чтобы воспользоваться /3.3/, нужно задаться значениями $/n n_s/$ относительных скоростей. Эти $/n n_s/$ скоростей будем условно рассматривать как "основные", а скорости, которые нужно определить как "дополнительные". После этого "дополнительные" скорости определяются через "основные" скорости и скорости и определяются через "основные" скорости и скорости "базовых" звеньев v_n , w_n с помощью /3.3/. В "основные", конечно, должны быть включены все скорости в шарнирах с кинематическим приводом.

Таким образом, вторая задача для разомкнутой цепи может быть решена в следующей последовательности:

- рассматриваем цепь как замкнутую и соответственно хапактеристическому числу $\xi 3$ присваиваем значение ноль; заданные скорости \mathbf{v}_n и ω_n рассматриваем как скрости конечного базового звена;

[.] Напомним , что под шарнирами здесь понимаются пары V класса /см. гл. I/.

- определяем характеристические числа $\lambda_{\mathbf{r}}$ /r=1,2,...,6/ согласно /2.15/ и подсчитываем

$$n_{s} = \sum_{i=1}^{6} \lambda_{i}$$

- всли $n>n_s$, то задаемся относительными скоростями в $/n-n_s/$ шарнирах, эти шарниры рассматриваем как "основные" и соответственно задаем значения характеристическим числам $\xi 5$ /см./1,8//;
- с помощью алгоритма МАТРИЦы СВЯЗИ определяем матрицы

- по уравнению

$$\dot{s} = \Lambda \dot{u} + \Lambda_1 (v_n - v_o) + \Lambda_2 (\omega_n - \omega_o),$$

где å - скорости в "основных" шарнирах, определяюм относительные скорости s в остальных шарнирах.

Теперь переходим к решению второй задачи для замкнутой цепи. Она несколько сложнее, поскольку теперь может быть задана скорость любого промежуточного звена цепи.

Пусть требуется определить $\dot{\mathbf{x}}$ такие, чтобы к-е эвено цепи $/0 < \mathbf{k} < \mathbf{n} /$ имело угловую и поступательную скорость $\boldsymbol{\omega}_{\mathbf{k}}$, $\boldsymbol{v}_{\mathbf{k}}$. Тогда поступим следующим образом, Разобъем заданную цепь на две замкнутые цепи:

первая цепь включает звенья с номерами О≤і≤k для неё к-е звено является конечным базовым; вторая цпь включает звены от к-го до n-го и для неё к-е звено явлается начальным базовым.

При таком разбиении для каждой из составляющих цепей известны скорости базовых звеньев и, следовательно, задача сводится к уже рассмотренному выше случаю. Остается только задать соответствующие значения характеристическим числам §3 и §5, назначить "основные" шарниры в каждой из цепей и далее можно решать задачу по описанной выше методике.

§2 Определение ускорений

При расчете ускорений также могут возникнуть две задачи. Первая ставится следующим образом. Имеется некоторая цепь /разомкнутая или замкнутая/, для которой известны значения координат \mathbf{x} , относительные скорости $\dot{\mathbf{x}}$, и относительные ускорения $\ddot{\mathbf{u}}$ в основных шарнирах. Необходимо определить векторы угловых ускорений $\varepsilon_{\mathbf{i}}$ и линейных ускорений $\mathbf{w}_{\mathbf{i}}$ центров масс всех звеньев цепи. Для замкнутых цепей требуется определить также значения $\ddot{\mathbf{s}}$ дополнительных координат.

Предполагается, что для рассматриваемой цепи уже решены задачи определения положений и скоростей, в связи с чем известны е, ${\bf r}$, ${\bf v}$ и ${\bf \omega}$ для всех звеньев и кинематических пар, а для замкнутых цепей известны также матрицы ${\bf \Lambda}$, ${\bf \Lambda}_1$ и ${\bf \Lambda}_2$.

Вторая задача обратна первой в смысле заданной и отыскиваемой информации. Здесь требуется найти значения относительных ускорений $\ddot{\mathbf{x}}$, которые обеспечивают движение одного из звеньев цепи с заданными ускорениями ϵ и \mathbf{w} .

Рассмотрим сначала первую задачу. При её решении будем пользоваться следующими двумя соотношениями механики. Пусть имеется твердое тело, для котопого определены переносные угловые скорость и ускорение $/\omega_{\rm пер}$, $\varepsilon_{\rm пер}/$ и относительные угловые скорость и ускорение $/\omega_{\rm отн}$, $\varepsilon_{\rm отн}/$. Тогда угловое ускорение $/\varepsilon/$ равно

$$\varepsilon = \varepsilon_{\text{nep}} + \varepsilon_{\text{oth}} + \omega_{\text{nep}} \times \omega_{\text{oth}}$$

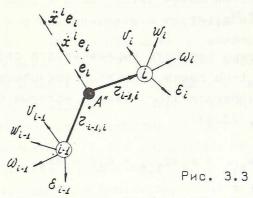
Для двух точек А и В, принадлежащих твердому телу, которое движется с угловыми скоростью и ускорением ω и ϵ , спра-

ведливо

$$w_B - w_A = \varepsilon \times r_{AB} + \omega \times (\omega \times r_{AB})$$

где r_{AB} - вектор с началом в т.А и концом в т.В.

Сначала рассмотрим задачу для разомкнутой цепи. На рис. 3.3. показана i-я кинематическая пара, образованная



i-ым и /i-1/-ым звеньями. Скорости и ускорения /i-1/-го звена будем рассматривать как переносные, а скорость $\dot{\mathbf{x}}^i$ и ускорение $\ddot{\mathbf{x}}^i$ в i-ом шарнире - как относительные. Тогда справедливы следующие зависимости:

в случае поступательной кинематической пары

$$\varepsilon_{i} = \varepsilon_{i-1}$$
 (3.5)

в случае вращательной пары

$$\varepsilon_{i} = \varepsilon_{i-1} + \ddot{x}^{i} e_{i} + \dot{x}^{i} \omega_{i-1} \times e_{i}$$
 (3.6)

Переходим к определению линейных ускорений. Обозначим через "А" ту точку неподвижного пространства, в которой в данный момент времени находится центр i-го шарнира. Ускорение точки /i-1/-го звена, которая совпадает с точкой "А" обозначим $w_{\rm A,i-1}$. Тогда можем записать /cm.puc 3.3/

$$w_{A,i-1} = w_{i-1}^{-\epsilon} x_{i-1}^{xr} x_{i-1,i}^{-\omega} x_{i-1}^{x} x_{i-1,i}^{(\omega)}$$
 (3.7)

Для точки i-го звена мгновенно совпадающей с "А" можно записать: в случае поступательной кинематической пары

$$w_{A,i} = w_{A,i-1} + \ddot{x}^{i}e_{i} + q\dot{x}^{i}\omega_{i-1}xe_{i}$$
 (3.8)

в случае вращательной пары

$$w_{A,i} = w_{A,i-1} \tag{3.9}$$

Теперь можно записать выражения для скорости и ускорения центра масс i-го звена. При этом воспользуемся угловыми скоростью и ускорением этого звена, которые были определены выше /см./3.5/, /3.6//

$$w_{i} = w_{A,i} + \varepsilon_{i} xr_{i,i} + \omega_{i} x(\omega_{i} xr_{i,i})$$
 (3.10)

Выражения /3.5/ - /3.10/ можно рассматривать как рекуррентные соотношения: если последовательно обходить звенья цепи начиная с базового, то с их помощью можно определить все интересующие нас ускорения.

Однако, прежде чем строить алгоритм эти выражения целесообразно привести к несколько иной форме. Дело в том, что алгоритм расчета ускорений в дальнейшем понадобится при анализе динамики. Для этого необходимо иметь возможность представить угловые и линейные ускорения /є,w/ в виде суммы векторов, следующего вида

$$\varepsilon_{i} = \sum_{j=1}^{i} \alpha_{ij} \ddot{x}^{j} + \theta_{i} + \varepsilon_{o}$$

$$w_{i} = \sum_{j=1}^{i} \beta_{ij} \ddot{x}^{j} + \eta_{i} + w_{o}$$
(3.11)

где $\epsilon_{
m O}$, $\rm w_{
m O}$ угловое и линейное ускорение начального базового эвена.

Такое представление, очевидно, всегда возможно, поскольку уравнения механики линейны относительно вторых производных. Векторные коэффициенты $\alpha_{\bf ij}$ и $\beta_{\bf ij}$ будут в этом случае функциями относительных координат, а векторы $\theta_{\bf i}$ и $n_{\bf i}$ относительных координат и скоростей.

Представление в виде /3.11/ нужно по следующей причине. Если при анализе кинематики предполагались известными все $\ddot{x}^{\dot{j}}$ / $\dot{j}=1,2,\ldots,n$ / то при анализе динамики как раз определение ускорений $\ddot{x}^{\dot{j}}$ является задачей анализа. Поэтому при анализе динамики будут использоваться не значения угловых и линейных ускорений, а коэффиценты в уравнениях /3.11/; при этом подразумевается, что для этих ускорений справедливы разложения /3.11/ /подробнее см. гл.V/.

Теперь вернемся к формулам /3.5/ - /3.10/ и првобразуем их к виду, удобному для определений в форме /3.11/. Для этого запишем в такой же форме ускорения предшествующего /i-l/-го звена /рис. 3.3/.

$$\varepsilon_{i-1} = \sum_{j=1}^{i} \alpha_{i-1,j} \ddot{x}^{j} + \theta_{i-1} + \varepsilon_{0}
w_{i-1} = \sum_{j=1}^{i} \beta_{i-1} \ddot{x}^{j} + \eta_{i-1} + w_{0}$$
(3.12)

Подставляя /3.11/ и /3.12/ в выражения для углового ускорения /3.5/ и /3.6/, получим:

для случая вращательной кинематической пары

$$\sum_{j=1}^{i} \alpha_{ij} \ddot{x}^{j} + \theta_{i} = \sum_{j=1}^{i-1} \alpha_{i-1,j} \ddot{x}^{j} + \ddot{x}^{i} e_{i} + \dot{x}^{i} \omega_{i-1} x e_{i} + \theta_{i-1}$$

Приравнивая здесь коэффициенты с одинаковым номером ј, получим

$$\alpha_{ij} = \alpha_{i-1,j}$$
 при $k \le j \le i-1$;

$$\alpha_{ii} = e_{i};$$

$$\theta_{i} = \theta_{i-1} + \dot{x}^{i}\omega_{i-1}xe_{i}$$

$$(3.13)$$

Для случая поступательной кинематической пары аналогично, подставляя /3.11/ и /3.12/ в /3.5/, получим

$$\alpha_{ij} = \alpha_{i-1,j} \text{ при } k \leq j \leq i-1;$$

$$\alpha_{ii} = 0;$$

$$\theta_{i} = \theta_{i-1}$$

$$(3.14)$$

Выражения /3.13/, /3.14/ поэволяют вычислить векорные коэффициенты в выражении углового ускорения /3.11/ по известным коэффициентам в выражении углового ускорения предшествующегозвена.

Переходим к определению ускорения центра масс i-го эвена. Вновь подставляя /3.11/, /3.12/ в /3.7/ - /3.10/, аналогично предыдущему получим, опуская все промежуточные выкладки.

Для вращательной кинематической пары

$$\beta_{ij} = \beta_{i-1,j} + \alpha_{i-1,j} x(r_{ii} - r_{i-1,i}) \text{ при } k \leq j \leq i-1;$$

$$\beta_{ii} = \theta_{i} xr_{ii};$$

$$\eta_{i} = \eta_{i-1} + \theta_{i-1} x(r_{ii} - r_{i-1,i}) + \hat{x}^{i} (\omega_{i-1} x)$$

$$xe_{i}) xr_{ii} + \omega_{i} x(\omega_{i} xr_{ii}) - \omega_{i-1} x(\omega_{i-1} xr_{i-1,i})$$
(3.15)

Для поступательной кинематической пары

$$\beta_{ij} = \beta_{i-1,j} + \alpha_{i-1,j} \times (r_{ii} - r_{i-1,i}) \text{ при } k \leq j \leq i-1;$$

$$\beta_{ii} = e_{i};$$

$$\eta_{i} = \eta_{i-1} + \theta_{i-1} \times (r_{ii} - r_{i-1,i}) + 2\dot{x}^{i}\omega_{i-1} \times e_{i} + \cdots + \omega_{i} \times (\omega_{i} \times r_{ii}) - \omega_{i-1} \times (\omega_{i-1} \times r_{i-1,i})$$

$$(3.16)$$

Таким образом получены рекуррентные соотношения для определения всех коэффициентов; с помощью этих коэффициентов искомые ускорения можно найти по /3.11/.

Поскольку для разомкнутой цепи известны все $\mathbf{x_i}$, то полученные выражения полностью решают первую задачу. В случае замкнутой цепи известны только основные ускорения $\ddot{\mathbf{u}}$. Недостающие значения $\ddot{\mathbf{s}}$ нужно определить из условия замыкония цепи.

Дла этого запишем согласно /3.11/ ускорения $\varepsilon_{\mathbf{n}}$ и $\mathbf{w}_{\mathbf{n}}$ для последнего звена эквивалентной цепи.

$$\varepsilon_{n} = \sum_{j=1}^{n} \alpha_{nj} \ddot{x}^{j} + \theta_{n} + \varepsilon_{o};$$

$$w_{n} = \sum_{j=1}^{n} \beta_{nj} \ddot{x}^{j} + \eta_{n} + w_{o}$$
(3.17)

Дла замкнутой цепи ускорения базовых звеньев считаем известными. Кроме того можно определить векторы $\theta_{\bf j}$ и $\eta_{\bf j}$, поскольку они зависят от относительных координат ${\bf x}$ и скоростей ${\bf \dot x}$, которые также известны. Далее согласно /3.13/ и /3.16/ можно записать

$$\alpha_{\text{nj}} = \begin{cases} \dot{e}_{\text{j}} & \text{если j-я пара вращательная} \\ 0 & \text{если j-я пара поступательная} \end{cases}$$
 (3.18)
$$\beta_{\text{nj}} = \begin{cases} e_{\text{j}} x r_{\text{nj}} & \text{если j-я пара вращательная} \\ e_{\text{j}} & \text{если j-я пара поступательная} \end{cases}$$

Воэвращаясь к гл. II § 3 видим, что коэффицивнты $\alpha_{\rm nj}^{\rm N}$ в /3.18/ выражаются так же как коэффицивнты $a_{\rm j}^{\rm ,b}$ в уравнениях /2.18/, /2.20/.

Тогда /3.17/ можно переписать в виде

$$\sum_{j=1}^{n} a_{j} \ddot{x}^{j} = \varepsilon_{n} - \varepsilon_{o} - \theta_{n};$$

$$\sum_{j=1}^{n} b_{j} \ddot{x}^{j} = w_{n} - w_{o} - \eta_{n}$$
(3.19)

Сравнивая /3.19/ с /2.17/ и /2.19/ видим, что тни отличаются только правыми частями, в которых

$$\Delta h$$
 заменено на $/\epsilon_n$ - ϵ_o - $\theta_n/$ (3.20)
 $\Delta \gamma$ заменено на $/w_n$ - w_o - $\eta_n/$

Уравнения /2.17/ и /2.19/ были исходными при выводе уравнения связи /2.31/, в которое Δh и $\Delta \gamma$ вошли линейно. Следовательно, если в уравнениях связи заменить Δs и Δu на \ddot{s} и \ddot{u} , а вместо Δh и $\Delta \gamma$ подставить соответствующие выражения из /3.20/, то это уравнение останется справедлиным.

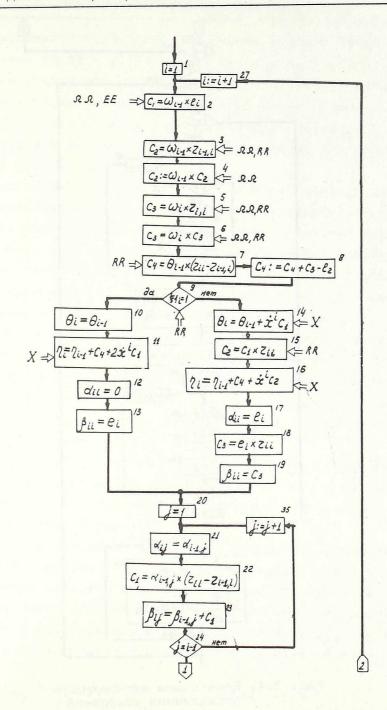
$$\ddot{s} = \Lambda \ddot{u} + \Lambda_1 (w_n - w_0 - \eta_n) + \Lambda_2 (\varepsilon_n - \varepsilon_0 - \theta_n)$$
 (3.21)

С помощью этого уравнения можно определить неизвестные ускорения $\ddot{\mathbf{s}}$ в дополнительных шарнирах. Матрижы Λ , Λ_1 , Λ_2 ^{на-}ходятся с помощью алгоритма МАТРИЦы СВЯЗИ.

После определения $\ddot{\mathbf{s}}$ становится известной /см./I.6// вся совокупность относительных ускорений $\ddot{\mathbf{x}}$ и можно переходить к анализуу эквивалентной открытой цепи, так как описывалось выше.

Блок-схема алгоритма определения ускорений /для первой задачи/ приведена на рис. 3.4.

В первом цикле с параметром і /блоки І-27/ вычисля- ются эначения векторов $\alpha_{\bf ij}$, $\beta_{\bf ij}$, $\theta_{\bf i}$, $\eta_{\bf i}$, входящие в выра-



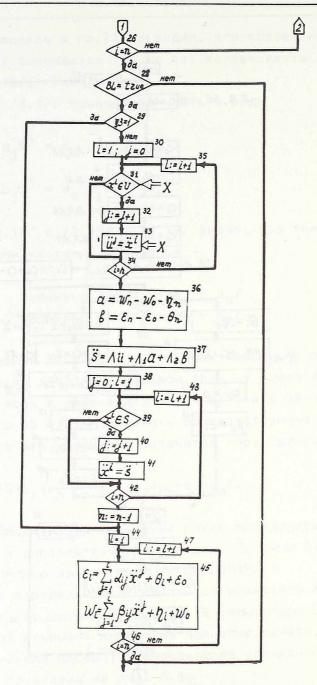


Рис. 3.4. Блок-схема алгоритма определения ускорений

жения ускорений /см./3.11//. Вычисление осуществляется согласно формулам /3.13 - 3.16/. Через c_1 , c_2 , c_3 , c_4 эдесь обозначены некоторые вспомогательные векторы, используемые при вычислениях промежуточных результатов. Важно отметить, что в этом цикле используются в частности $\alpha_{\rm oj}$, $\theta_{\rm o}$, $\beta_{\rm oj}$, $\eta_{\rm o}$ /при i=1/. Перед началом вычислений этим величинам необходимо присвоить нулевые значения.

В алгоритме предусмотрено два режима работы и выбор режима осуществляется с памащью булевской переменной BL /блок 28/. Если BL \equiv false, то алгоритм определяет только значения α_{ij} , β_{ij} , θ_{i} , η_{i} и на этом заканчивает работу. Такой режим используется например, при "составлении" дифференциальных уравнений /см. гл. У/. Если BL \equiv true, то опредеяются эначения ускорений \mathbf{w}_{i} и ϵ_{i} для всех звеньев /i = 1,2, ..., \mathbf{n} /. При этом для замкнутых цепей /проверяется в блоке 29/ предварительно определяют значения ускорений дополнительных координат $\ddot{\mathbf{s}}$ /см. блоки 30-43/. Для этого сначала формируется вектор $\ddot{\mathbf{u}}$ в блоках 30-35. Вычисление $\ddot{\mathbf{s}}$ осуществляется в блоке 37 с помощью матриц связи Λ , Λ_{1} и Λ_{2} . Значения $\ddot{\mathbf{s}}$ заносятся $\ddot{\mathbf{s}}$ $\ddot{\mathbf{s}}$ в блоках 39-43.

Теперь, после того как известны все относительные ускорения $\ddot{\mathbf{x}}$, рассчитываются векторы ускорений $\mathbf{w_i}$ и $\mathbf{\epsilon_i}$ /цикл по \mathbf{i} , блоки 45-47/. При этом используются ускорения $\mathbf{w_o}$ и $\mathbf{\epsilon_o}$ начального базового звена.

В случае замкнутой цепи можно определять ускорения ${arepsilon_i},$ ${f w}_i$ только для первых /n-1/ звеньев, поскольку для n-го звена эти ускорения известны /как для конечного базового звена/.

Ниже текст алгоритма записан на АЛГОЛе как процедура УСКОРЕНИЯ. В этой процедуре коэффициенты α_{ij} , β_{ij} , θ_{i} и η_{i} помещены в трехмерные массивы w и ϵ следующим образом

где стрелки указывают на соответствие, а значек * как и раньше показывает, что соответствующий индекс пробегает все значения. Другими словами, по третьему измерению w, ε по-мещаются компоненты векторов α , β , θ , η . Например,

 $\alpha_{ij}^{k} \rightarrow w[i,j,k]$

Массивы w, є имеют размерность [0:n, 1:n+l, 1:3].

Массивы VV и $\Omega\Omega$ описывались ранее в алгоритме СКОРО-СТЬ /см. предыдущий параграф/ В массивах ϵ 0, w0, ϵ_n , w $_n$ хранятся ускорения базовых звеньев, соответственно ϵ_0 , w $_0$, ϵ_n , w $_n$. Как и раньше Λ , Λ_1 , Λ_2 матрица связи. С помощью булевской переменной BL осуществляется выбор режима работы /см. выше/. Если BL \equiv true то вычисляются значения ω_i , ϵ_i , которые помещаются в i-ый "столбец" массивов W и ϵ соответственно.

procedure ychopehus (N,w0,ε0,wn,επ,W,ε,BL);
value N, BL; integer N; boolean BL;
array w0,ε0,wn,επ,W,ε;
begin

array c1,c2,c3,c4,ω1,ω2,e,r1,r2,r3,θ;η[1:3];
integer i,j,k,l,n,ns,nu;
n: = STR[N,6]; nu: = STR[N,7]; ns: = n-nu;

begin

array u[1:nu], s, s1, s2[1:ns];
if BL then 1: = 1 else 1: = n+1;
for i: = 0 step 1 until n do
 for j: = 1 step 1 until 1 do
 for k: = 1,2,3 do
 W[i,j,k]: = ε[i,j,k]: = 0;
for i: = 1 step 1 until n do

```
begin
    begin \omega 1[k] := \Omega\Omega[N,i-1,k];
    \omega^{2}[k] := \Omega\Omega[N,i,k]; e[k] := EE[N,i,k];
    r1[k] := RR[N,i-1,2,k]; r2[k] := RR[N,i,1,k];
    r3[k] := r2[k]-r1[k]; \theta[k] := \varepsilon[i-1,1,k];
    \eta[k] := w[i-1,1,k] \text{ end};
BEKT (c1,\omega1,e); BEKT (c2,\omega1,r1);
BEKT (c2,\omega1,c2); BEKT (c3,\omega2,r2);
BEKT (c3,\omega2,c3); BEKT (c4,\theta,r3);
for k := 1,2,3 do
    c4[k] : = c4[k]+c3[k]-c2[k];
if KP[N,i,1] = 1 then for k := 1,2,3 do
    begin \varepsilon[i,l,k]: = \theta[k];
    w[i,1,k]: = \eta[k] + 2xX[N,i,2]xcl[k];
    if 7 BL then begin
   W[i,i,k]: = e[k]; \varepsilon[i,i,k]: = 0 end end
else begin
BEKT (c2,c1,r2); BEKT (c3,e,r2);
for k: = 1,2,3 do begin
    \varepsilon[i,1,k] := \theta[k] + X[N,i,2] \times c1[k];
    W[i,1,k] := \eta[k] + c4[k] + X[N,i,2] \times c2[k];
    if \exists BL then begin \varepsilon[i,i,k] := e[k];
    W[i,i,k] := c3[k] end end end;
if 7 BL then
    for j: = 1 step 1 until i-1 do begin
    for k := 1,2,3 do cl[k] := \varepsilon[i-1,j,k];
   BEKT (cl,cl,r3);
    for k := 1,2,3 do \epsilon[i,j,k] := \epsilon[i-1,j,k];
    W[i,j,k] := W[i-1,j,k]+cl[k]
end j ; end i ;
if 7 BL then go to mm;
```

```
if STR[N,1] = 1 then go to ml;
   begin array \Lambda[1:ns,1:nu], \Lambda 1,\Lambda 2[1:ns,1:3];
   for i: = 1 step 1 until ns do
   begin for j: = 1 step 1 until nu do
     \Lambda[i,j] := \Lambda\Lambda[N,i,j];
   <u>for</u> j: = 1,2,3 <u>do</u> begin
    \Lambda 1[i,j] := \Lambda \Lambda[N,i,j+nu];
     \Lambda 2[i,j] := \Lambda \Lambda[N,i,j+nu+3]
   end end;
   j: = 0; for i: = 1 step 1 until n do
     if KP[N,i,3] = 1 then
     begin j: = j+l;
      u[j]:=X[N,i,3] end;
   for k: = 1,2,3 do begin
      c2[k]: = wn[k]-w0[k]-W[n,1,k];
      c3[k]: = \varepsilon n[k] - \varepsilon 0[k] - \varepsilon[n,1,k] end;
   MTB (s2, \Lambda 2, c3, ns, 3); MTB (s1, \Lambda 1, c2, ns, 3);
   MTB(s, \lambda, u, ns, nu)
   end блока;
   for k: = 1 step 1 until ns do
      s[k]: = s[k]+s1[k]+s2[k];
   j: = 0; for i: = 1 step 1 until n do
   if KP[N,i,3] \neq 1 then
     begin j: = j+1; X[N,i,3]: = s[j] end;
   n:=n-1;
ml: for k: = 1, 2, 3 do
      begin \omega 1[k] := w0[k];
      \omega 2[k] := \varepsilon 0[k] end;
   for i: = 1 step 1 until n do
   if KP[N,i,1] = 1 then
      for k: = 1,2,3 <u>do begin</u>
      \omega 1[k] := \omega 1[k] + X[N,i,3] \times EE[N,i,k];
     W[i,1,k] := W[i,1,k] + \omega 1[k]  end
```

```
else begin for k: = 1,2,3 do
  begin cl[k]: = ε[i-1,1,k];
  e[k]: = X[N,i,3]xEE[N,i,k];
  r2[k]: = RR[N,i,1,k];
  r1[k]: = r2[k]-RR[N,i-1,2,k] end k;

BEKT(cl,cl,rl); BEKT(c2,e,r2);
for k: = 1,2,3 do
  begin ω2[k]: = ω2[k]+e[k];
  ω1[k]: = ω1[k]+ci[k]+c2[k];
  W[i,1,k]: = W[i,1,k]+ω1[k];
  ε[i,1,k]: = ε[i,1,k]+ω2[k]end k;
```

<u>end</u> услочвого оператора и цикла по i; mm: <u>end end</u> процедуры;

Теперь рассмотрим вторую из задач, сформулилированных в начале этого параграфа. Пусть для открытой цепи заданы угловое и линейное ускорение последнего звена $/\epsilon_n$, w_n /. Необходимо определить относительние ускорения \ddot{x} , обеспечивающие ϵ_n и w_n на последнем звене.

Аналогично тому, как это делалось для скоростей, будем рассматривать замкнутую цепь, конечное базовое звено имеет ускорения ε_n и w_n . В такой интерпретации задача сводится к рассмотренному выше случаю.

Для определения относительных ускорений мы имеем матричное уравнение /3.21/. Оно определяет $\mathbf{n_s}$ независимых сототношений, и такое число относительных ускорений можно определить. Тут возможны два случая, когда число шарниров цепи $\mathbf{n_s}$ или больше $\mathbf{n_s}$ /случай $\mathbf{n< n_s}$ невозможен, см. предыдущий параграф/.

Если $n=n_{_{\mathbf{S}}}$, то уравнение /3.21/ можно записать в виде

$$\ddot{\mathbf{x}} = \Lambda_1 (\mathbf{w}_{\mathbf{n}} - \mathbf{w}_{\mathbf{0}} - \boldsymbol{\eta}_{\mathbf{n}}) + \Lambda_2 (\boldsymbol{\varepsilon}_{\mathbf{n}} - \boldsymbol{\varepsilon}_{\mathbf{0}} - \boldsymbol{\theta}_{\mathbf{n}}),$$

откуда можно определить относительные ускорения во всех ша-

рнирах.

Если $n > n_s$, то число независимых условий связи меньше числа неизвестных, и для того чтбы воспользоваться /3.21/, нужно задаться значениями относительных ускорений в $/n-n_s/$ шарнирах. В это число должны быть включены все шарниры с кинематическим приводом, поскольку для них заданы законы движения и значит известны относительные ускорения. Шарниры, в которых задаются ускорения, будем условно рассматривать как "основные", а остальные шарниры как "дополнительные".

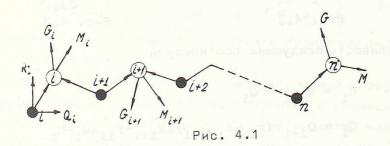
В соответствии с таким представлением нужно задать значения характеристических чисел $\xi 3$ и $\xi 5$ и затем с помощью алгоритма МАТРИЦы СВЯЗИ определить Λ , Λ_1 и Λ_2 . После этого /3.21/ определяются неизвестные $\ddot{\mathbf{s}}$.

Во второй задаче для замкнутой цепи могут быть заданы ускорения для некоторого внутреннего звена. Тогда задачу можно решать также, как в аналогичном случае для скоростей /см. предшествующий параграф/, то-есть разбить данную цепь на две, таким образом, чтобы звено, для которого заданы усткорения, было базовым для этих двух цепей. После этого задача сводится к уже рассмотренному случаю определения ускорений замкнутых цепей.

Гл. IV **СТАТИКА ПРОСТРАНСТВЕННЫХ ЦЕПЕЙ**

§1 Реакции в кинематических парах разомкнутых цепей

Разорвем мысленно кинематическую цепь в некотором i-ом шарнире и рассмотрим равновесие оставшегся свободного конца цепи /рис. 4.1/. Воздействие отброшенной части цепи включая i-ый привод заменим силой R_i и моментом Q_i . Эту силу и момент будем называть полной реакцией в i-ом шарнире.



Для определения полных реакций приведем все внешние силы G_j и моменты $M_j/j=i,i+1,\ldots,n/$ к центру i-го шарнира. Реакции приводов во всех последующих j-ых шарнирах /j>i/ можно не учитывать, поскольку каждый привод действует на два соседних звена с равными и противоположно направленными силами или моментами, которые при приведении дадут в сумме ноль.

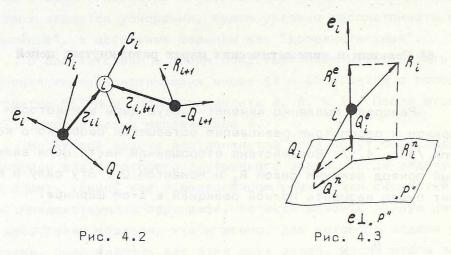
Тогда для полных реакций можно записать

$$R_{i} = -\sum_{j=i}^{n} G_{j}$$

$$Q_{i} = -\sum_{j=i}^{n} (M_{j} + r_{ji} \times G_{j})$$
 (4.1)

где n - число звеньев цепи.

Для построения алгоритма удобнее иметь рекуррентную формулу, связывающую полные реакции в двух соседних шарнирах. На рис. 4.2. показано i-е звено цепи; из рисунка очевидна



справедливость следующих соотношений

$$R_{i} = R_{i+1} - G_{i}$$

$$Q_{i} = Q_{i+1} + (r_{ii} - r_{i,i+1}) xR_{i+1} - r_{ii} xG_{i} - M_{i}$$
(4.2)

Двигаясь вдоль цепи от последнего звена к первому можно по /4.2/ определить все полные реакции.

Разложим полные реакции в i-ом шарнире на две составляющие: одну параллельную вектору e_i и другую перпендикулярную /рис. 4.3/. Параллельные составляющие для силы и момента обозначим соответственно R_i^e и Q_i^e , перпендикулярные R_i^n и Q_i^n . Составляющие можно определить из соотношений

$$R_i^e = (R_i \cdot e_i) e_i$$
;

$$R_{i}^{n} = R_{i} - R_{i}^{e};$$

$$Q_{i}^{e} = (Q_{i} \cdot e_{i})e_{i};$$

$$Q_{i}^{n} = Q_{i} - Q_{i}^{e}$$

$$(4.3)$$

Перпендикулярные составляющие не могут вызвать перемещений механизма и только нагружают шарнирнов соединение. Эти составляющие нужны для учета сил трения. Одна из параллельных составляющих /момент для поступательных пар и сила для вращательных/ также воспринимается конструкцией шарнира, вызывая соответствующую силу трения. Что касается другой параллельной составляющей, то для того чтобы механизм оставался неподвижным, она должна уравновешиваться реакцией і-го привода. Реакции привода выше были обозначены рі. Тогда условия статического равновесия будут

для поступательной пары
$$p^{i} = R_{i} \cdot e_{i}$$
 для вращательной пары $p^{i} = Q_{i} \cdot e_{i}$ (4.4)

Условия /4.4/ позволяют определить требуемые силу или момент в каждом приводе.

Разомкнутая цепь образует с базовым звеном первую кинематическую пару. Воздействие цепи и первого привода на базовое звено равно силе $/-R_1/$, отнесенной к центру шарнира, и моменту $/-Q_1/$. Приведем эти силы и момент к полюсу базового звена и получивщуюся силу и момент обозначим $R_{\rm A}$, $Q_{\rm A}$

$$R_{A} = -R_{1}$$
; $Q_{A} = r_{01} \times R_{1} - Q_{1}$

Векторы R_A и Q_A будем называть воздействием цепи на базовое звено. Они понадобятся при анализе взаимосвязанных цепей /см.гл VI/. Блок-схема алгоритма для определения реакций приведена на рис. 4.4. Алгоритм рассчитывает все реакции в кинематических парах /силы и моменты/ в виде двух составляющих параллельных и перпендикулярных оси кинематической пары. Модули перпендикулярных составляющих для силы и момента по-

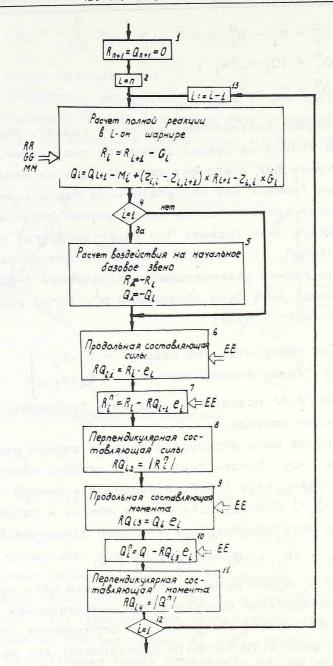


Рис. 4.4. Блок-схема алгоритма расчета реакций

мещаются на место 2-ой и 4-ой компонент вектора RQ. Значения параллельных составляющих /знак определяется соответственно направлению вектора оси кинематической пары $\mathbf{e_i}$ / помещаются на место 1-ой и 3-ей компонент RQ/сила-в первую и момент-в третью/.

Алгоритм построим согласно формулам /4.2/ - /4.4/. Значение параметра цикла і соответствует номеру кинематической пары; обход всех шарниров осуществляется от конца цепи к началу.

Ниже приводится текст алгоритм определения скоростей. Массив RQ имеет размерность $|1:n,\ 1:4|$. В результате выполнения процедуры заполняется RQ

$$RQ = \begin{pmatrix} R_{i} \cdot e_{i} \\ |R_{i}^{n}| \\ Q_{i} \cdot e_{i} \\ |Q_{i}^{n}| \end{pmatrix}$$

и вычисляется воздействие цепи на базовое звено: сила RA и момент RQ. Глобальные массивы: STR, EE, RR, MM, QQ.

```
procedure CTATUHA PA3. LETUM (N,RQ,RA,QA);

value n; integer N;

array RQ,RA,QA;

begin

array a,b,g,R,Q [1:3];

integer i,j,k,n;

real c1, c2;

n: = STR[N,6];

for i: = 1,2,3 do R[i]: = Q[i]: = 0;

for i: = n step - 1 until 1 do

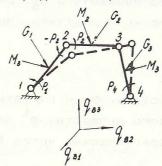
begin
```

```
for j: = 1,2,3 do
 begin a[j]: = RR[N,i,1,j] - RR[N,i,2,j];
      g[j]: = GG[n,i,j]; b[j]: = RR[N,i,1,j]
 end;
     BEKT(a,R,a): BEKT(b,g,b);
for j := 1,2,3 do
\underline{\text{begin}} \ R[j] := R[j] - g[j];
Q[j] := Q[j] - MM[N,i,j] + a[j] - b[j];
     a[j]: = EE[N,i,j] end;
     СКЛ(c1,R,a); СКЛ(c2,Q,a)
for j: = 1,2,3 do begin a management restricted and suppose
     a[j]: = R[j] - c1 \times a[j];
     b[j]: = Q[j] - c2 \times a[j] end;
     RQ[i,1]: = c1; RQ[i,3]: = c2;
     RQ[i,2]: = MOД(a);
     RQ[i,4]: = MOД(b);
\underline{if} \ i = 1 \ \underline{then} \ \underline{for} \ j : = 1,2,3 \ do
     begin RA[N,j] = -R[j];
     QA[N,j] = -Q[j] end
end i;
end процедуры;
```

§2 Определение реакций в случае замкнутых цепей

Если в случае разомкнутой цепи можно было по известной внешней нагрузке определить реакции всех приводов, то в зам-кнутых цепях картина сложнее. Здесь число приводов может быть больше числа степеней свободы* и тогда для того, чтобы механизм находился в равновесии, между реакциями этих приводов должно поддерживаться определенное соответствие.

Рассмотрим в качестве примера плоский четырехэвенный механизм /рис. 4.5/. Предположим, что он имеет силовые при-



Puc. 4.5

воды в 1, 2 и 4-ом шарнирах; привод в третьем шарнире отсутствует $p^3=0$. Механизм обладает одной степенью свободы и можно задать только одно независимое виртуальное перемещение $/\delta x^1/$, например, в 1-ом шарнире. Записав уравнение работ на этом вертуальном перемещений получим одно линейное уравнение, связывающее моменты приводов

^{*} В этом случае, конечно, все приводы не могут быть кинематическими.

$$\sum_{j=1}^{3} \left[G_{j} \cdot \delta_{j} + M_{j} \cdot e_{j} \quad \sum_{k=1}^{j} \delta_{x}^{k} + p^{j} \delta_{x}^{j} \right] + p^{4} \delta_{x}^{4} = 0$$
 (4.5)

где $\delta_{\mathbf{j}}$ - вектор перемещения центра масс \mathbf{j} -го звена.

Леремещения $\delta \mathbf{x}^{\mathbf{k}}$ /k = 2,3,4/ и векторы δ ј можно определить с помощью алгоритма ЗАМКНУТАЯ ЦЕПЬ /гл. II/.

Сбедовательно, в этом примере можно задать независимо реакции, например, в трех первых преводах /считаем, что в третьем приводе действует нулевая реакция/. Тогда реакция четвертого привода определится из /4.5/.

Для более сложных пространственных цепей можно в общем случае записать столько уравнений связи между воздействиями, сколько можно задать независимых виртуальных перемещений, или сколькими степенями свободы обладает цепь. Число степеней свободы ранее было обозначено $\mathbf{n}_{\mathbf{u}}$. Следовательно, столько реакций приводов можно определить. Число заданных реакций приводов равно

$$n_s = n - n_u$$

Те приводы, реанций которых будем считать известными необходимо отметить. Для этого множеству p^i поставим в соответствие множество характеристических чисел λl_i , определяемых с помощью соотношения

$$\lambda 1_{i} = \begin{cases} 1 - если реакция p^{i} задана; \\ 0 - если реакцию p^{i} необходимо \\ определить \end{cases}$$
 (4.6)

При этом совершенно необязательно, чтобы множество известных реакций каким-то образом соответствовало множеству основных координат.

Множество известных реакций в этом параграфе будем обозначать $\mathbf{p_1^i}$ / \mathbf{i} = $1,2,\ldots,n_{_S}$ /, множество реакций, которые подлежат определению $\mathbf{p_0^i}$ / \mathbf{i} = $1,2,\ldots,n_{_U}$ /. Соответственно, координаты в этих шарнирах будем обозначать $\mathbf{x_1^j}$ и $\mathbf{x_0^j}$.

Рассмотрим некоторую замкнутую цепь. Будем задавать виртуальные перемещения на множестве координат U. В дальнейшем удобнее пассматривать не виртуальные перемещения, а скорости виртуальных перемещений, то-есть $\delta u^i/\delta t$, где δt - некоторый отрезок времени. Условимся эти скорости называть "виртуальными" скоростями.

Скорость центра масс і-го звена и его угловую скорость, обусловленные $\dot{\mathbf{u}}^{\mathbf{j}}$ будем обозначать $\mathbf{v}_{\mathbf{i}\mathbf{j}}$ и $\omega_{\mathbf{i}\mathbf{j}}$ соответственно, а скорость относительного перемещения в і-ом шарнире - как $\dot{\mathbf{x}}_{\mathbf{j}}^{\mathbf{i}}$. Тогда условия равновесия цепи можно записать так

$$\sum_{i=1}^{n-1} (G_{i} \cdot v_{ij} + M_{i} \cdot \omega_{ij} + p^{i} \dot{x}_{j}^{i}) + p^{n} \dot{x}_{j}^{n} = 0$$

$$(j = 1, 2, ..., n_{ij})$$
(4.7)

Пременим к множествам P и \dot{X} орератор U /см.гл.II,§3/ и выделим подмножества соответствующие известным и неизвестным реакциям приводов.

Теперь перепишем /4.7/ в виде

$$\sum_{i=1}^{n_{u}} \dot{x}_{0j}^{i} p_{0}^{i} = -\sum_{i=1}^{n-1} (G_{i} \cdot v_{ij} + M_{i} \cdot \omega_{ij}) - \sum_{i=1}^{n} p_{1}^{i} \dot{x}_{1j}^{i}$$
(4.8)

Скорости v_{ij} , ω_{ij} , \dot{x}_{oj}^i и x_{1j}^i можно определить по алгоритму СКОРОСТИ /см.гл.III,\$I/. Обозначив прæую часть /4.8/ через a^i и введя матрицу

$$C = \| x_{oj}^{i} \|_{n_{u}, n_{u}}$$

запишем уравнение /4.8/ в виде

$$Cp_{O} = a (4.9)$$

откуда можно определить воздействия $extbf{p}_{\mathbf{0}}^{\mathbf{i}}.$

$$p_{O} = C^{-1}a$$
 (4.10)

Соотвестсвующий алгоритм будет рассмотрен поэже, а сейчас переходим к определению реакций в шарнирах.

Воздействие рассмотриваемой цепи на конечное базовое звено обозначим как силу $R_{\rm B}$ и момент $Q_{\rm B}$. Эти силу и момент будем считать приведенными к полюсу конечного базового звена. Сила $R_{\rm B}$ и момент $Q_{\rm B}$ должны удовлетворять условию: если отбросить данную цепь и приложить к базовому звену $R_{\rm B}$ и $Q_{\rm B}$, то конечная базовая цепь должна находиться в равновесии.

Рассмотрим эквивалентную кинематическую цепь для данной замкнутой. Дла того чтобы эта цепь была подобна замкнутой с точки зрения статики к её последнему звену необходимо приложить силу $\mathbf{G}_{\mathbf{n}}$ и момент $\mathbf{M}_{\mathbf{n}}$ удовлетворяющие условиям

$$G_n = -R_B;$$

$$M_n = -Q_B.$$

Задача будет состоять в том, чтобы определить G_n и M_n , после чего можно рассматривать статику эквивалентной цепи по алгоритму, описанному в предыдущем параграфе.

Предположим, что уже решены задачи "сборки" и перемещения цепи, так что эквивалентная цепь в рассматриваемый момент совпадает с замкнутой цепью и известны все векторы е, $\mathbf{r} \quad \text{и мотрицы связи } \Lambda, \ \Lambda_1, \ \Lambda_2. \ \text{Известны также векторы* } \mathbf{q}_1^{\text{O}}$ /i = 1,2,3/ и все реакции приводов \mathbf{p}^1 .

Для определения G_n и M_n будем задавать последнему звену эквивалентной цепи поступательные и угловые скорости последовательно вдоль каждой из осей q_i^o /i = 1,2,3/. Обозначим их v_{ni} и ω_{ni} , где i-номер оси q_i^o . Как уже указывалось выше при "сборке" цепи /см.гл.II, 2/, для последнего звена могут оказаться невозможными линейные или угловые перемещения вдоль некоторых из осей q_i^o . Эти случаи отмечаятся с помощью вектора λ .

Допустим $\lambda_1=1$, то-всть возможно поступательное перемещение последнего звена вдоль q_1^{O} . Зададим некоторую скорость

^{*} См. гл. II, §3.

 $^{
m v}_{
m nl}$, в этом направлении. Мощность ${
m g}_{
m l}$ для силы ${
m G}_{
m n}$ будет

$$g_1 = (G_n \cdot q_1^0) | v_{n1} |$$
 (4.11)

Теперь подсчитаем мощность для всех внешних сил и моментов, а также реакции приводов в случае , когда последнее звено имеет скорость V_{n1} . Для этого вернемся к главеIII, где рассматривается кинематика, и определим относительные скорости \dot{x} в шарнирах цепи, при которых обеспечивается V_{n1} . Скорости основных координат примим нулевыми $/\dot{u}=0/$, а будем искать значения скоростей \dot{s} в дополнительных шарнирах. Поскольку скорость V_{n1} задана по одному из направлений, "разрешенным" вектором λ , то, очевидно, эту скорость можно обеспечить задавая определенные скорости в дополнительных шарнирах. Согласно /2.31/ можем записать

$$\dot{s} = \Lambda_1 v_{n1} \tag{4.12}$$

Теперь присваиваем значения компонент s соответствующим компонентам x, после чего можно по алгоритме СКОРОСТИ рассчитать скорости центров масс и угловые скорости всех эвеньев. Мощность внешней нагрузки и реакций приводов можно записать как

$$g_{s} = \sum_{i=1}^{n-1} (G_{i} \cdot v_{i} + M_{i} \cdot \omega_{i} + p^{i} \dot{x}^{i}) + p^{n} \dot{x}^{n}$$
 (4.13)

где скорости ${f v}_{f i}$, ${f \omega}_{f i}$ должны, конечно, соответствовать за-данной скорости последнего звена ${f v}_{
m nl}$.

Суммарная мощность всех сил и моментов должна быть равна нулю, тогда используя /4.11/ можно записать

$$g_1 = -g_s \tag{4.14}$$

Проекция реакции $\mathbf{G}_{\mathbf{n}}$ на $\mathbf{q}_{\mathbf{1}}^{\mathbf{O}}$ равна

$$(G_n \cdot q_1^0) = -\frac{g_s}{|v_{nj}|}$$
 (4.15)

Подобным же образом можно определить другие проекции

силы $\mathbf{G_n}$ на оси $\mathbf{q_i^o}$. При определении $\mathbf{M_n}$ будем действовать аналогично, задавая угловые скорости последнего звена $\mathbf{\omega_{ni}}$ /i = 1,2,3/.

После того, как определены реакции G_n и M_n будем рассматривать их как внешние силу и момент действующие на последнее звено эквивалентной цепи. Тогда определение реакции во всех шарнирах можно осуществить с помощью процедуры СТА-ТИКА РАЗ ЦЕПИ /см. предыдущий параграф/.

Описанный выше способ анализа статики замкнутых цепей может быть осуществлен с помощью следующего алгоритма /рис. 4.6/.

Комментарий к функциональной блок-схеме.

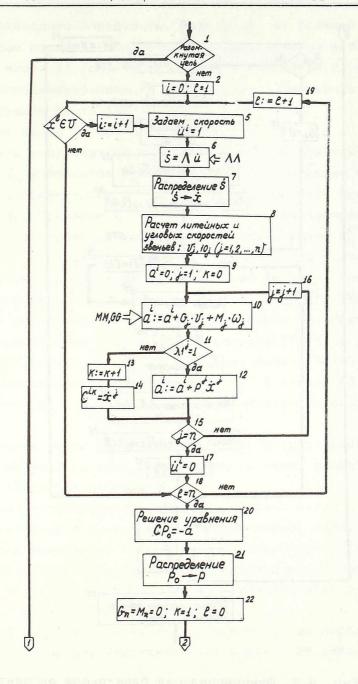
Блок I проверяет является ли цепь замкнутой. Остальная часть алгоритма делится на три части. В первой /блоки 2-21/ определяются все неизвестные реакции приводов $\mathbf{p^j}$, во-второй /блоки 22-44/ определяются реакции $\mathbf{G_n}$ и $\mathbf{M_n}$ на последнем звене эквивалентной цепи и, наконец, в третьей части /она состоит из одного блока 45/ проводится анализ статики эквивалентной цепи.

Первая часть алгоритма:

Содержит цикл с параметром 1 /блоки 2-19/, значения которого соответствуют номеру кинематической пары в цепи. Для каждой пары в блоке 3 проверяется является-ли 1-я координанта основной и, если да, то вычисляется значение индекса і, который определяет номер основной координаты.

После этого задается скорость $\dot{\mathbf{u}}^{\mathbf{i}}=\mathbf{1}$ и в блоках 6,7 определяются $\dot{\mathbf{s}}$ и заносится в $\dot{\mathbf{x}}$. В следующем блоке 8 осуществляется расчет скоростей /линейных и угловых/ всех звеньев цепи.

В цикле по ј /блоки 9-16/ осуществляется расчет мощностей для внешних сил G и моментов M и определяются матрица C и вектор a в формуле /4.9/. Значение j здесь соответствувт номеру звена, а значение k - номеру реакции привода p_0



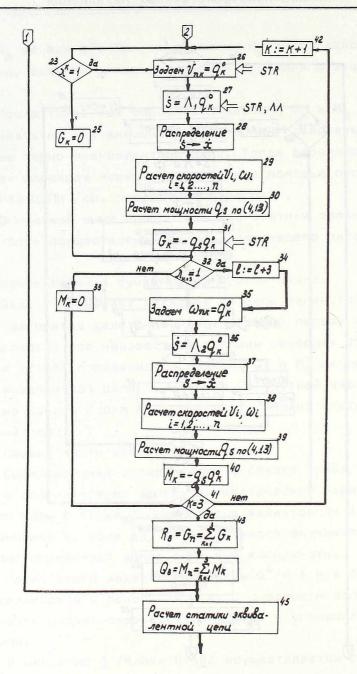


Рис. 4.6. Функциональная блок-схема алгоритма для расчета статики замкнутой Цепи

которую необходимо определить. Выделение тех реакций приводов, которые подлежат определению осуществляется в блоке 11 с помощью чисел $\lambda 1$ /см./4.6//. Далее в блоках 20,21 осуществляется определение реакций приводов $\mathbf{p}_{\mathbf{0}}$ и занесение их в соответствующие места массива $\mathbf{p}_{\mathbf{0}}$

Вторая чюсть алгоритма:

В цикле с параметром k /блоки 22-42/ вычисляются провенции сил G_n и моментов M_n на оси \mathbf{q}_k^o . Составляющие вдоль этих осей обозначены \mathbf{G}_k и \mathbf{M}_k . Значение k соответствует номеру орта \mathbf{q}_k^o . При этом в блоке 23 проверяется возможно ли поступательное перемещение последнего звена вдоль \mathbf{q}_k^o и если да $/\lambda_k=1$ /, то задается скорость $\mathbf{v}_{nk}=\mathbf{q}_k^o$. После этого в блоке 27 осуществляется расчет скоростей \mathbf{s} . Затем следует анлиз кинематики /определение скоростей, алгоритм СНОРОСТЬ, блок 29 /и расчет мощности по /4.13/ в блоке 30. Составляющая \mathbf{G}_k определяется в блоке 31 по формуле /4.15/ с учетом значения скорости $\mathbf{v}_{nk}=\mathbf{q}_k^o$. В следующих блоках 32-40 аналогичным образом находится составляющая \mathbf{M}_k . В блоках 43,44 усилие и момент \mathbf{G}_n , \mathbf{M}_n определяются суммированием составляющих \mathbf{G}_k и \mathbf{M}_k .

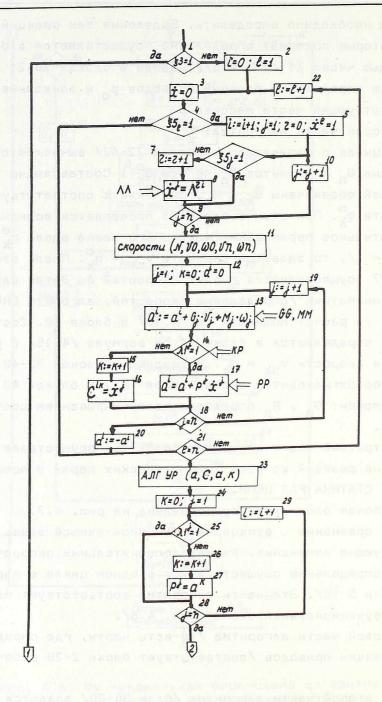
В третьей части алгоритма /блок 45/ осуществляется определение реакций во всех кинематических парах с помощью алгоритма СТАТИКА РАЗ ЦЕПИ.

Рабочая блок-схема представлена на рис. 4.7.

По сравнению с функциональной блок-схемой эдесь сделаны следующие изменения. Расчет дополнительных скоростей **s** и их распределение осуществляются в одном цикле с парометром ј /блоки 5-10/. Эта часть алгоритма соответствует блокам 5-7 на функциональной схеме /рис. 4.6/.

Первой части алгоритма /то-есть части, где определяются реакции приводов /соответствуют блоки 2-29 рабочей блок--схемы.

Во второй части алгоритма /блок 30-60/ задаются ${
m v}_{
m nk}$ и



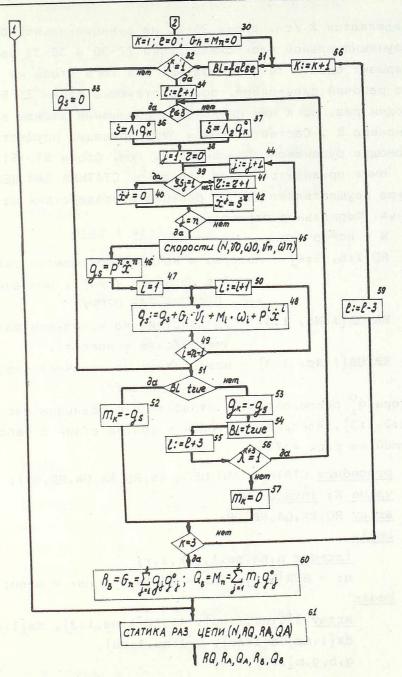


Рис. 4.7. Рабочая блок-схема алгоритма СТАТИКА ЗАМ ЦЕПИ

определяются \dot{x} /см. блоки 26-28 на функциональной блок-схеме/. На функциональной блок-схеме блоки 27-30 и 36-39 фактически совершают одни и те же операции. Для того чтобы не повторяться в рабочей блок-схеме, соответствующие блоки 35-50 выписаны один раз, но к ним происходит обращение дважды для каждого номера k. Соответствующая переадресация осуществляется с помощью булевской переменной BL /см. блоки 31, 51, 56/.

Ниже приводится текст процедуры СТАТИНА ЗАМ ЦЕПИ. Процедура осуществляет расчет реакций и воздействий на базовые звенья. Формальные параметры:

N - номер цепи;

RQ[1:n, 1:4] — массив, в который помещаются значения реакций в шарнирах /см.§І, описание процедуры СТАТИКА РАЗ ЦЕПИ/;

RA,QA[1:Nc, 1:3] — воздействие на начальное базовое звено /сила и момент/;

RB,QB[1:Nc, 1:3] — воздействие на конечное базовое звено /сила и момент/.

Векторы $\mathbf{q_i^o}$ расположены по столбцам в глобальном массиве $\mathbf{qq[1:3,1:3]}$. Алгоритм построен в соответствии с рабочей блок--схемой на рис. 4.7.

procedure CTATUHA ЗАМ ЦЕПИ (N,RQ,RA,QA,RB,QB);
value N; integer N;
array RQ,RA,QA,RB,QB;
begin

integer n,ns,nu,i,j,k,1,r;
n: = STR[N,6]; nu: = STR[N,7]; ns: = n-nu;

<u>begin</u>

array Λ[1:ns,1:nu], Λ1,Λ2[1:ns,1:3], dx[1:n],
ds[1:ns],a[1:nu], C[1:nu,1:nu],
q,b,g,m[1:3];
real gs;
boolean BL;

```
if STR[N,1] = 1 then go to ml;
     for j: = 1,2,3 \text{ do}[b]j : = 0;
     for i: = 1 step 1 until ns do
     begin for j: = 1 step 1 until nu do
                 \Lambda[i,j] := \Lambda\Lambda[N,i,j];
                 for j: = 1,2,3 do begin
                 \Lambda l[i,j] := \Lambda \Lambda[N,i,j+nu]
                 \Lambda 2[i,j] := \Lambda \Lambda[N,i,j+nu+3]
     end end;
for i: = 1 step 1 until n do
    dx[i] := X[N,i,2];
i: = 0;
for 1: = 1 step 1 until n do
begin
     for j: = 1 step 1 until n do
                 X[N,j,2] := 0;
     if KP[N,i,3] = 0 then go to m2;
     i: = i+1; r: = 0; X[N,1,2]: = 1;
     for j: = 1 step 1 until n do
                if KP[N,j,3] = 0 then
     begin r: = r+1;
                 X[N,j,2]:=\Lambda[r,i] end;
     CKOPOCTH (N,b,b,b,b);
     k: = 0; a[i]: = 0;
     for j: = 1 step 1 until n do
     begin for rl: = 1,2,3 do
     a[i]: = a[i]+GG[N,j,rl]\times VV[N,j,rl]+
                 MM[N,j,r1] \times \Omega\Omega[N,j,r1];
     if KP[N,j,4] = 1 then
     a[i]: = a[i]+PP[N,j]xX[N,j,2]
     else begin k: = k+l;
     C[i,k]: = X[N,j,2] end
     end j ;
```

```
a[i]: = -a[i];
     m2: end 1;
     АЛГ.УР(å,C,a,k);
     k := 0;
     for j: = 1 step 1 until n do
     if KP[N,j,4] \neq 1 then
     begin k: = k+1;
           PP[N,j]: = a[k] end
     for k: = 1,2,3 do
          GG[N,n,k] := MM[N,n,k] := 0;
     k: = 1; 1: = 0;
comment начало цикла по k;
m6:
     BL: = false;
     if STR[N,k+8] \neq 1 then
     begin gs: = 0; go to m3 end;
     1: = 1+1;
m5: for j: = 1,2,3 do
     q[j] := qq[j,k];
     if 1 < 3 then
     MTB(ds, \Lambda 1, q, ns, 3)
     else MTB (ds, \lambda2, q, ns, 3);
     r: = 0; for j: = 1 step 1 until n do
     if STR[N,j,3] = 0 then begin
     r: = r+1; X[N,j,2]: = ds[r] end
     else X[N,j,2]: = 0;
     СКОРОСТИ (N,b,b,b,b);
     gs: = PP[N,n]xX[N,n,2];
     for i: = 1 step 1 until n-1 do
     begin for j: = 1,2,3 do
     gs: = gs+GG[N,i,j]xVV[N,i,j]+
     MM[N,i,j] \times \Omega\Omega [N,i,j] + PP[N,i] \times X[N,i,2] end;
m3: if BL then begin
     m[k] : = -gs; go to m4 end;
```

```
g[k]: = -gs; BL: = true; 1: = 1+3;
     if STR[N,k+11] = 1 then go to m5;
     m|k| : = 0;
     if k < 2 then
m4:
          begin k: = k+l; go to m6 end;
     for j: = 1,2,3 do
     for k: = 1,2,3 do
     begin
     RB[N,k]: = GG[N,n,k]: = g[j]xqq[k,j];
     QB[N,k]: = MM[N,n,k]: = m[j]xqq[k,j];
     end;
ml:
    CTATИКА РАЗ ЦЕПИ (N,RQ,RA,QA);
     for i: = 1 step 1 until n do
          X[N,i,2] := dx[i]
     end end процедуры;
```

§3 Статика механизмов с упругими связями

На практике часто встречаются случаи, когда силовые приводы механизмов представляют собой следящие системи. Та-кие системы обладают определенной статической характеристи-кой

$$p^{i} = f(y^{i} - y^{i}_{0})$$

где $p^{\mathbf{i}}$ - реакция \mathbf{i} -го привода; $y^{\mathbf{i}}$ - относительная координата в \mathbf{i} -ом шарнире;

 $\mathbf{y_o^i}$ - относительная координата, соответствующая нулевой реакции.

$$p^{i} = f(0) = 0.$$

В статике поведение такого привода во многом аналогично упругой связи /возможно нелинейной/, у которой можно изменять /регулировать положение равновесия. Другими словами, можно рассматривать в статике силовой привод как такой кинематеческий привод, который соединен с приводимыми звеньями посредством упругой связи.

Под нагрузкой такая упругая связь будет деформимоваться и поэтому каждому силовому приводу можно приписать некоторою жесткость, которую будем обозначать $\mathbf{c_i}$, где \mathbf{i} -номер привода. Обычно в силовых приводах $\mathbf{c_i}$ довольно большая величина, так что деформации не превышают нескольких градусов. Будем предполагать что это так, после чего можно линеаризовать статическую характеристику привода в пределах деформации. Тогда $\mathbf{c_i}$ = \mathbf{const} и можно исследовать механизм с линейными упругими связями. Перемещения в силовых приводах, вызванные нагрузкой условимся называть деформацией.

Пусть в отсутствии нагрузки относительные координаты равны \mathbf{x}_{o} . После этого к механизму приложены известные внешние силы и моменты \mathbf{G}_{i} , \mathbf{M}_{i} . Необходимо определить координаты \mathbf{x} в новом положении статического равновесия. При этом примем что кинематические приводы не деформируются, то-есть

$$z = z_0$$

Для замкнутых цепей будем считать, что в дополнительных шарнирах /координаты s/ привод отсутствует.

Обозначим деформацию в силовых приводах

$$\Delta y^{i} = y^{i} - y^{i}_{O} \tag{4.19}$$

Она связана с реакциями приводов соотношением

$$\Delta y^{i} = y^{i} - y^{i}_{0} = -c_{i}p^{i}$$
; (i = 1,2,..., n_{y}) (4.20)

Реакции приводов необходимые для равновесия цепи определяются в помощью алгоритмов СТАТИКА РАЗ ЦЕПИ или СТАТИКА ЗАМ ЦЕПИ. Обращаясь к этим алгоритмам, видим, что в них используются векторы \mathbf{r}_{ij} , которые сами являются функциями неизвестных координат \mathbf{y}^i . Таким образом, эти алгоритмы совместно с /4.20/ определяют \mathbf{y}^i как неявную функцию, разрешить которую аналитически относительно \mathbf{y}^i не представляется возможным. Поэтому будем находить \mathbf{y}^i последовательными приближениями.

Последовательное приближение можно осуществлять следующим образом:

- 1. Для данных значений \mathbf{x}_{o} с помощью алгоритмов ПОЛОЖЕНИЕ и ЗАМКНУТАЯ ЦЕПЬ /последний для замкнутых цепей, см.гл.II, $\S2,3$ / определяем векторы \mathbf{r}_{ij} , \mathbf{e}_{i} . Под \mathbf{x}_{o} теперь будем понимать не обязательно значения относительных координат, при которых упругие связи не деформированы, но те значения, с которых начинается приближение. Воздействия приводов, соответствующие \mathbf{x}_{o} обозначим \mathbf{p}_{o} .
- 2. Определяем воздействия приводов ${
 m p}_{
 m O}$. Для разомкнутой цепи

используем алгоритм СТАТИКА РАЗ ЦЕПИ, для замкнутой кроме того СТАТИКА ЗАМ ЦЕПИ. В случае замкнутой цепи нужно задать вектор $\lambda 1$, который указывает в каких шарнирах нужно определить реакции /см. предыдущий параграф/. Поскольку в данной задаче определяются реакции в основных приводах вектор $\lambda 1$ можно задать как

$$\lambda \mathbf{1}^{\dot{\mathbf{1}}} = \begin{cases} 1 & \text{если } \mathbf{x}^{\dot{\mathbf{1}}} \in \mathbf{S} \\ 0 & \text{в противном случае} \end{cases}$$
 (4.21)

3. Определяим деформации Ду. Согласно

$$\Delta y^{i} = -c_{i}(p^{i} - p_{o}^{i})$$
 (4.22)

4. Определяем максимальное значение деформации

$$b = \max_{(i)} ([\Delta y^{i}]), \quad i = 1, 2, ..., n_{y}$$
 (4.23)

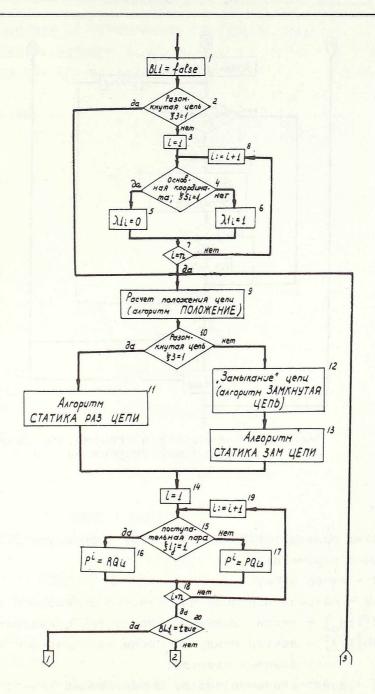
5. Если $b \le \epsilon$,где ϵ - заданное малое число, то процесс итеррации считаем законченным. В противном случае присваиваем y_0 новые значения

$$y_0 := y_0 + \Delta y \tag{4.24}$$

6. Переходим к п.І

Блок-схема описанного процесса представлена на рис. 4.8 /алгоритм ДЕФОРМАЦИЯ/.

В блоках 2,10 проверяется тип цепи и в случае разом-кнутой цепи работают блоки 9,11, а в случае замкнутой цепи - блоки 3-8, 12,13. В блоках 3-8 /цикл с параметрам і /осуществляется формирование вектора $\lambda 1$. Окончание итеррационного процесса определяется с помощью булевской переменной BL; если все деформации $\Delta y^{i} \leq \varepsilon$ /i = 1,2,..., n_{y} /, то BL получает значения true /см. блоки 21,26/. После того как BL=true и итеррацию можно считать законченной другой булевской переменной BL1 присваивается значение true и выполняется еще один расчет реакций в блоках 9,11 или 12,13 соответственно для разомкнутых и замкнутых цепей.



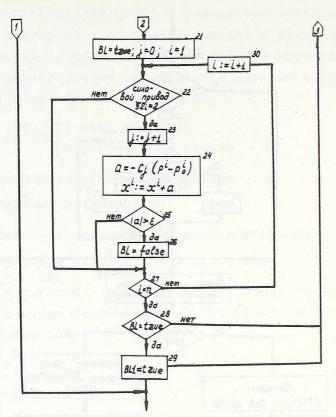


Рис. 4.8. Блок-схема алгорытма определения деформаций упругих связей

Ниже приводится текст алгоритма /процедура ДЕФОРМАЦИЯ/. Формальные параметры:

N - номер цепи;

А2 - матрица перехода для конечного базового звена;

 $C[1:n_{_{\mathbf{v},\mathbf{v}}}]$ - массив значений жесткостей приводов;

hb[1:3] - вектор между полосами начального и конечного базовых эвеньев;

 ϵ - действительное число, определяющее точность вычислений. Глобальные массивы: STR, KP, EE, RR, MM, GG, $\Lambda\Lambda$, PP. Перед началом вычислений в массив PP должны быть занесены значения P_O .

```
procedure ДЕФОРМАЦИЯ (N, A2, hb, c, є);
      value N; integer N;
      array A2, hb, c; real ε;
begin
      integer i, j, k, n, nu, ns;
      n: = STR[N,6]; nu: = STR[N,7]; ns: = n-nu;
begin
      array pl [1:n], RQ [1:n,1:4], RA, QA,
            RB, QB[N:N,1:3];
      boolean BL, BL1;
      real a;
      BLl: = false;
      if STR[N,1] = 1 the go to ml;
      for i: = 1 step 1 until n do
          if KP[N,i,3] = 1 then KP[N,i,4] := 0
                      else KP[N,i,4]: = 1;
     if STR[N,1] = 1 then
ml:
      begin ПОЛОЖЕНИЕ (N);
      СТАТИКА РАЗ ЦЕПИ (N, RQ, RA, QA) end
     else begin
      ЗАМКНУТАЯ ЦЕПЬ (N,A2,hb,\epsilon);
      СТАТИКА ЗАМ ЦЕПИ (N,RQ,RA,QA,RB,QB)
      end;
      for i: = 1 step 1 until n do
      if KP[N,i,1] = 1 then pl[i] := RQ[i,1]
                     else pl[i] := RQ[i,3];
      if BL1 then go to m2;
          BL: = true; j: = 0;
      for i: = 1 step 1 until n do
      if KP[N,i,2] = 2 then
      begin j:=j+1;
          a: =-c[j]x(pl[i]-PP[N,i]);
          PP[N,i]: = pl[i];
```

X[N,i,1]: = X[N,i,1]+a;if abs(a) > ϵ then BL: = \underline{false}

end;

if TBL then go to ml else

begin BL1: = true; go to ml end;

m2: <u>end</u> <u>end</u> процедуры;

Гл. V ДИНАМИКА ПРОСТРАНСТВЕННЫХ ЦЕПЕЙ

§1 Силы и моменты сил инерции

В главе III ускорения центров масс звеньев цепи и угловые ускорения были получены в виде

$$w_{i} = \sum_{j=1}^{i} \beta_{ij} \ddot{x}^{j} + \eta_{i} + w_{o}$$

$$\varepsilon_{i} = \sum_{j=1}^{i} \alpha_{ij} \ddot{x}^{j} + \theta_{i} + \varepsilon_{o}$$
(5.1)

где коэффициенты $\alpha_{\bf ij}$, $\beta_{\bf ij}$, $\theta_{\bf i}$, $\eta_{\bf i}$ определяются с помощью алгоритма УСКОРЕНИЯ /см.гл.III,§2/.

Силы инерции каждого звена будем приводить к его центру масс и полученные главный вектор и главный момент обозначать соответственно $\mathbf{f_i}$ и $\mathbf{M_i}$, где \mathbf{i} -номер звена. Для простоты в дальнейшем будем называть $\mathbf{f_i}$ и $\mathbf{M_i}$ просто силой и моментом сил инерции.

Силу инерции можно определить по известному уравнению

$$f_{i} = -m_{i}w_{i}$$

Или используя /5.1/ получим

$$f_{i} = \int_{j=1}^{i} a_{ij} \ddot{x}^{j} - m_{i} w_{o} + a_{i}^{o}$$
 (5.2)

где
$$a_{ij} = -m_i \beta_{ij}$$
; $a_i^0 = -m_i \eta_i$.

Для определения момента инерции воспользуемся динамическими уравнениями Эйлера,

$$\begin{split} \widetilde{M}_{i}^{1} &= -J_{i1}\widetilde{\epsilon}_{i}^{1} + (J_{i2} - J_{i3})\widetilde{\omega}_{i}^{2}\widetilde{\omega}_{i}^{3} \\ \widetilde{M}_{i}^{2} &= -J_{i2}\widetilde{\epsilon}_{i}^{2} + (J_{i3} - J_{i1})\widetilde{\omega}_{i}^{3}\widetilde{\omega}_{i}^{1} \\ \widetilde{M}_{i}^{3} &= -J_{i3}\widetilde{\epsilon}_{i}^{3} + (J_{i1} - J_{i2})\widetilde{\omega}_{i}^{1}\widetilde{\omega}_{i}^{2} \end{split}$$
(5.3)

где верхняя волнистая черта указывает, что векторы M, ϵ , ω берутся в проекциях на внутреннюю /подвижную/ систему координат; верхний индекс указывает номер проекции, то-есть

$$\tilde{M}_{i}^{j} = M_{i} \cdot q_{ij}; \qquad \tilde{\epsilon}_{i}^{j} = \epsilon_{i} \cdot q_{ij}$$
 (5.4)

Для того чтобы перейти к неподвижной системе координат нужно воспользоваться матрицей перехода /см.гл.II,§I/.

$$A_{i} = \|A_{i}^{jk}\|_{3\times3}$$

Тогда

$$M_{i} = A_{i} \begin{pmatrix} \tilde{M}_{i}^{1} \\ \tilde{M}_{i}^{2} \\ \tilde{M}_{i}^{3} \end{pmatrix}$$

$$(5.5)$$

Умножим левые части /5.3/ на матрицу $\mathbf{A_i}$ и заменим проекции ускорения $\mathbf{\tilde{\epsilon}_i^j}$ их выражениями /5.4/. Получим уравнение для проекций момента сил инерции на оси неподвижной системы координат

$$\begin{split} M_{i}^{j} &= -\left[(A_{i}^{j1}J_{i1}q_{i1}^{1} + A_{i}^{j2}J_{i2}q_{i2}^{1} + A_{i}^{j3}J_{i3}q_{i3}^{1}) \, \epsilon_{i}^{1} + \right. \\ &+ (A_{i}^{j1}J_{i1}q_{i1}^{2} + A_{i}^{j2}J_{i2}q_{i2}^{2} + A_{i}^{j3}J_{i3}q_{i3}^{2}) \, \epsilon_{i}^{3}) + \\ &+ (A_{i}^{j1}J_{i1}q_{i1}^{3} + A_{i}^{j2}J_{i2}q_{i2}^{3} + A_{i}^{j3}J_{i3}q_{i3}^{2}) \, \epsilon_{i}^{2} + \\ &+ \psi_{i}^{j} \; ; \qquad (j = 1, 2, 3) \; ; \qquad (5.6) \end{split}$$

$$\psi_{i} = \begin{pmatrix} \psi_{i}^{1} \\ \psi_{i}^{2} \\ \psi_{i}^{3} \end{pmatrix} = A_{i} \begin{pmatrix} (J_{i2} - J_{i3}) & (\omega_{i} \cdot q_{i2}) & (\omega_{i} \cdot q_{i3}) \\ (J_{i3} - J_{i1}) & (\omega_{i} \cdot q_{i3}) & (\omega_{i} \cdot q_{i1}) \\ (J_{i1} - J_{i2}) & (\omega_{i} \cdot q_{i1}) & (\omega_{i} \cdot q_{i2}) \end{pmatrix} (5.7)$$

верхний индекс ј указывает проекцию на соответствующую ось ниподвижной системы координат, то-есть

$$M_{i}^{1} = M_{i} \cdot i; \quad M_{i}^{2} = M_{i} \cdot j; \quad M_{i}^{3} = M_{i} \cdot k.$$

Матрица перехода следующим образом выражается через базис внутренней системы координат

$$A_{i} = ||q_{i1}q_{i2}q_{i3}||$$

где q_{ij}-векторы≏столбцы. Отсюда

$$A_{i}^{jk} = q_{ik}^{j}$$

Подставляя выражения компонент $A_{f i}^{f jk}$ в /5.6/ запишем выражние для момента инерции в виде

$$M_{i} = -T_{i}\varepsilon_{i} + \psi_{i} \tag{5.8}$$

где $\mathtt{T_i}$ -матрица $\mathtt{3x3}$, компоненты которой $\mathtt{T_i^{jk}}$ равны

$$T_{i}^{jk} = \sum_{l=1}^{3} A_{i}^{jl} J_{il} q_{il}^{k} = \sum_{l=1}^{3} q_{il}^{j} q_{il}^{k} J_{il}$$
 (5.9)

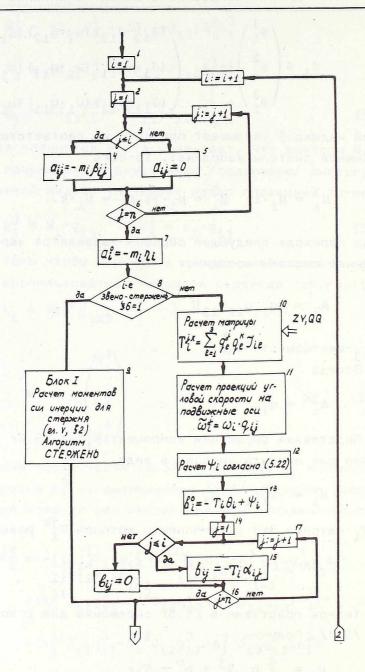
Теперь подставим в /5.8/ выражение для угловых ускорений /5.1/. Получим

$$M_{i} = \int_{j=1}^{i} b_{ij} \ddot{x}^{j} + b_{i}^{O} - T_{i} \varepsilon_{O}$$

$$b_{ij} = -T_{i} \alpha_{ij} ; b_{i}^{O} = -T_{i} \theta_{i} + \psi_{i}$$

$$(5.10)$$

где



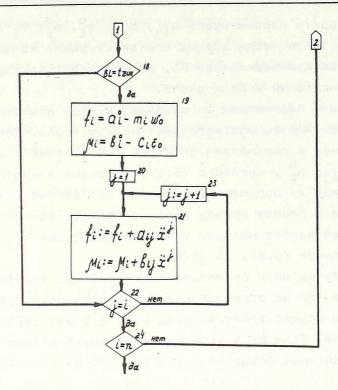


Рис. 5.1. Блок-схема алгоритма определения сил и моментов сил инерций

Определение сил и моментов инерции для замкнутых и разомкнутых цепей не отличается, существенно, только для зам-кнутых цепей следует рассматривать эквивалентную цепь.

Ниже приводится блок-схема алгоритма для расчета сил и моментов сил инерции /рис. 5.1/. В тех случаях, когда эвено можно рассматривать как стержень, определение моментов
сил инерции можно осуществить более простым способом. Поэтому в алгоритме предусмотрен блок I для расчета момента
сил инерции стержня. Подробно этот вопрос рассматривается
в следующем параграфе.

Алгоритм рассчитывает a_{ij} , b_{ij} , a_i^o , b_i^o , T_i в формулах /5.2/, /5.10/ и. если задана соответствующая команда с помощью булевской переменной BL, рассчитывает векторы f_i , M_i . Комментарый к блок-схеме.

Цикл с параметрам і включает в себя все блоки алгоритма. Значение і соответствует номеру звена. Первый внутренний цикл с параметром ј /блоки 2-6/ и блок 7 вычисляет векторы $\mathbf{a_{ij}}$, $\mathbf{a_{i}^{o}}$, согласно /5.2/. Условный оператор 8 проверяет можно ли рассматривать звено как стержень и, если нет, включается в работу правая цепочка блоков. В этой цепи осуществляется расчет матрицы \mathbf{T} и затем векторов $\mathbf{b_{ij}}$ и $\mathbf{b_{i}^{o}}$ согласно формулам /5.9/, /5.10/.

В случае всли булевская переменная BL = false /проверка в блоке 18/ на этом заканчивается выполнение расчетов при данном і и осуществляется выход на условный оператор 22 в конце цикла. Если BL = true, то включаются в работу блоки 19-23, в которых осуществляется расчет сил и моментов сил инерции.

В приводимом ниже тексте алгоритма с целью экономии оперативной памяти компоненты векторов a_{ij} , b_{ij} , a_i^O , b_i^O помещаются в массивы w и ϵ , где ранее находилась информация об ускорениях. Информация об ускорениях /после того как она использована для расчета a_{ij} , b_{ij} , a_i^O , b_i^O / стирается.

Индексы по первым двум измерениям мяссивов w и ϵ соответствуют индексам i, j у векторов a_{ij} и b_{ij} . Векторы a_i и b_i^0 помещаются в /n+1/-ые столбцы массивов w и ϵ соответственно.

Матрицы $\mathbf{T_i}$ после расчета помещаются в массив QQ , где ранее находились матрицы перехода $\mathbf{A_i}$. Матрицы перехода в дальнейшем не понадобятся и таким образом удается сократить требуемый объем памяти.

Предполагается, что предварительно были решены задачи кинематики и массивы $\Omega\Omega$, W , ϵ являются полностью заполнен-

ными /см. процедуры СКОРОСТЬ И УСКОРЕНИЕ/.

procedure ИНЕРЦИЯ (N, W, ϵ , BL);

begin

С помощью булевской переменной BL указывает один из двух возможных режимов работы. При BL \equiv false рассчитываются векторыа $_{ij}$, b_{ij} , a_i^o , b_i^o , и матрицы T, этот режим используется при составлении дифференциальных уравнений. Если BL \equiv true, то рассчитываются силы и моменты сил инерции f_i , $M_i/$; они помещаются в первые столбцы массивов W и ϵ .

```
value N; integer N;
array W, &; boolean BL;
integer n,i,j,k,p,s,r,l;
array a[1:3], T[1:3,1:3];
integer array M[1:6];
real c, cl, m;
    n: = STR[N,6];
if BL then 1: = 1 else 1: = n+1;
for i: = 1,2,3 do M[i]: = M[i+3]: = i;
for i: = 1 step 1 until n do
begin m := ZV[N,i,2]
    for j: = 1 step 1 until n do
   if \exists BL then for k: = 1,2,3 do
    if j < i then W[i,j,k] : = -mxW[i,j,k]
               else W[i,j,k]: = 0;
    for k: = 1,2,3 do
               W[i,l,k] := -mxW[i,l,k];
    if \BL then
    begin
               for j: = 1,2,3 do for k: = 1,2,3 do
               T[j,k] := QQ[N,i,j,k];
               for j: = 1,2,3 do
               begin
```

```
r: = M[j+1]; s: = M[j+2];
                c: = c1: = 0;
                for k := 1,2,3 do
                begin
                c: = c+\Omega\Omega[N,i,k]xQQ[N,i,k,r];
                cl: = cl+\Omega\Omega[N,i,k] \times QQ[N,i,k,s]
                end;
                a[j] : = cxclx(ZV[N,i,r+2]-ZV[N,i,s+2])
                end j;
               MTB(a,T,a,3,3);
                for j: = 1,2,3 do
                \varepsilon [i,n+l,j]: = \varepsilon [i,n+l,j]+a[j]
   end условного оператора;
   for j: = 1,2,3 do for k: = 1,2,3 do
   begin c: = 0;
   for p: = 1,2,3 do
   c: = c+QQ[N,i,j,p]xQQ[N,i,k,p]xZV[N,i,p+2];
   T[j,k] := c
   end;
  for j: = 1,2,3 do for k: = 1,2,3 do
                QQ[N,i,j,k]: = T[j,k];
   for j: = 1 step 1 until 1 do
   begin
                for k: = 1,2,3 do
               a[k] := \varepsilon[i,j,k];
                MTB(a,T,a,3,3);
                for k: = 1,2,3 do
                \varepsilon[i,j,k] = a[k]
        end j:
end i;
```

<u>end</u> процедуры;

§2 Особенности анализа динамики стержневых механизмов

Под стержнем будем понимать однородный круговой цилиндр, диаметром которого можно пренебречь в сравнении с длиной. Такой объект будем характеризовать длиной 21, массой m и двумя моментами инерции J_N и J_S относительно оси перпендикулярной стержню и проходящий через его центр масс, и относительно оси стержня.

Отметим, что хотя у стержня не учитывается диаметр, момент инерции J_s относительно продольной оси считается конечной величиной. Такое дапущение представляет собой известную идеализацию, однако, напомним, что подобные идеализации широко используются в механике. Напримет, в модели "материальная точка" учитывается конечная масса и в то же время размеры считаются пренебрежимо малыми.

Определение главного вектора сил инерции для стежня не отличается от общего случая, рассмотренного в предыдущем параграфе. Что касается главного момента, то эдесь можно воспользоваться уравнениями Эйлера, однако, для стержня это не выгодно. В настоящем параграфе будут получены некоторые другие уравнения, позволяющие определять момент сил инерции значительно проще.

Уравнение для момента сил инерции твердого тела можно записать как

$$M = -\frac{d}{dt}(e\omega) = -e\frac{d\omega}{dt} - \frac{de}{dt}\omega \qquad (5.11)$$

где 6-тензор инерции тела.

В выражении /5.11/ неудобен последний член в правой

части, поскольку произвадная от тензора инерции в неподвижной системе координат имеет сложный вид. Обычный путь преодоления этих трудностей состоит в том, что уравнение /5.11/проектируют на подвижные оси координат, связанные с телом, и дифференцируют в этой подвижной састеме. В результате получают известные динамические уравнения Эйлера /5.3/.

Для стержня возможен другой путь. Последний в /5.11/ можно для этой модели представить в виде

ет
$$\tau = (\omega \cdot \mathbf{s}) (\mathbf{s} \mathbf{x} \omega)$$
 (5.12)

s-единичный вектор, направленный вдоль оси стержня.Вектор τ будем называть эквивалентным угловым ускорением.

Тогда выражение для момента сил инерции можно представить в виде

$$M = -e(\varepsilon + \tau) \tag{5.13}$$

где ε-угловое ускорение тела.

Обозначим через s единичный вектор, направленный вдоль оси стержня. Далее разложим M и є на две составляющие, одну перпендикулярную оси стержня /обозначим её нижним индексом N/, другую параллельную оси стержня /нижний индекс s/.

$$M_{\rm N} = (s \times M) \times s; \quad M_{\rm S} = (M \cdot s)s$$
 $\varepsilon_{\rm N} = (s \times \varepsilon) \times s; \quad \varepsilon_{\rm S} = (\varepsilon \cdot s)s$
(5.14)

Из /5.12/ непосредственновидно, что т перпендикулярно оси стержня.

Воспользуемся тем, что стержень является симметричной фигурой и его момент инерции относительно любой оси перпендикулярной s и проходящей через центр масс равен J_N . Тогда /5.13/ можно записать в виде

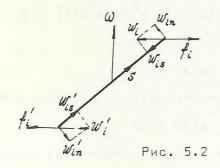
$$M = -J_{N}(\varepsilon_{N} + \tau) - J_{S}\varepsilon_{S} \qquad (5.15)$$

Отметим, что полученное уравнение определяет момент сил инерции в неподвижной системе координат и в них не вхо-дят проекции угловой скорости на подвижные оси.

Переходим к выводу формулы /5.12/ для эквивалентного углового ускорения.

Пусть имеется стержень, движущийся с угловой скоростью ω /рис. 5.2/. Рассмотрим некоторую точку i стержня, отстоящую от его центра масс на растояние ρ . Эта точка будет иметь осестремительное ускорение w_i равное

$$w_{i} = \rho \omega \times (\omega \times s) \tag{5.16}$$



Разложим $\mathbf{w_i}$ на две составляющие: перпендикулярную стер- who $\mathbf{w_{in}}$ и параллельную $\mathbf{w_{is}}$.

$$w_{in} = (s \times w_i) \times s; w_{is} = (s \cdot w_i)s (5.17)$$

Так как стержень симметричен, то по другую сторону от центра масс найдется точка $\mathbf{i'}$ на расстоянии ρ от центра масс, ускорения которой будут $\mathbf{w'_{in}}$, $\mathbf{w'_{is}}$.

$$w'_{in} = -w_{in}$$
; $w'_{is} = -w_{is}$

Элементарные силы инерции f_i , f_i' для точек i и i' будут, очевидно, пропорциональны ускорениям. При этом составляющие сил инерции зависящие от w_{is} и w_{is}' будут направлены вдоль оси стержня в разные стороны и их равнодействующая будет равна нулю. Следовательно, момент сил инерции обусловлен

только составляющими ускорений w_{in} и w_{in}' .

Теперь подберем такое угловое ускорение τ , которое вызывает в точках i и i' ускорения w_{in} и w_{in}' , то-есть для τ должно выполняться

$$p\tau \times s = w_{in}$$
 (5.18)

Для стержня осестремительные ускорения всех его точек лежат в одной плоскости /плоскости векторов s и ω / и τ должно быть перпендикулярно этой плоскости.

Подставим в /5.18/ выражения для $w_{
m in}$ и $w_{
m i}$ из равенств /5.17/, /5.16/. Получим

$$\tau \times s = [s \times (\omega \times (\omega \times s))] \times s$$

откуда, учитывая что тls получим

$$\tau = s \times (\omega \times (\omega \times s))$$

Преобразуя правую часть получим окончательно

$$\tau = (\omega \cdot s) (s \times \omega)$$

Алгоритм для расчета момента сил инерции стержня весьма прост и поэтому здесь не приводится. Он непосредственно следует формулам /5.12/ - /5.15/.

Теперь рассмотрим вопрос, в каких случаях можно пользоваться моделью типа стержень и какие при этом получюятся ошибки.

Пусть имеется однородный круговой цилиндр диаметром ${\tt d}$ и длиной ${\tt 21}$. Свяжем с ним систему координат с базисом ${\tt q}_1$, ${\tt q}_3$ причем ось ${\tt q}_1$ направим вдоль оси цилиндра. Запишем динамические уравнения Эйлера для момента сил инерции.

$$M^{1} = -J_{s} \varepsilon^{1}$$

$$M^{2} = -J_{N} \varepsilon^{2} + (J_{N} - J_{s}) \omega^{1} \omega^{3}$$

$$M^{3} = -J_{N} \varepsilon^{3} + (J_{s} - J_{N}) \omega^{1} \omega^{2}$$
(5.19)

где верхние индексы указывают номер проекции на оси подвижной системы координат, то-есть

$$\varepsilon^{j} = \varepsilon \cdot q_{j}; \quad M^{j} = M \cdot q_{j}; \quad \omega^{j} = \omega \cdot q_{j}$$

Для упрощения обозначений здесь опущена верхняя волнистая черта, которой раньше отмечались векторы заданные в проекциях на подвижные оси координат.

Дорустим теперь имеется стержень, у которого моменты инерции J_N , J_s , масса m и длина такие же как у цилиндра. Запишем для стержня выражение момента сил инерции. Для того чтобы можно было сравнить результат с /5.19/ перепишем основное уравнение /5.15/ в проекциях на подвижные оси. Тогда

$$M^{1} = -J_{s}\varepsilon_{s} = -J_{s}\varepsilon^{1}$$

$$M^{2} = -J_{N}(\varepsilon_{N} \cdot q_{2}) - J_{N}(\tau \cdot q_{2})$$

$$M^{3} = -J_{N}(\varepsilon_{N} \cdot q_{3}) - J_{N}(\tau \cdot q_{3})$$
(5.20)

Воспользуемся выражением /5.12/ для т и запишем

$$\begin{aligned} \tau \cdot \mathbf{q}_{\mathbf{j}} &= \mathbf{q}_{\mathbf{j}} \cdot (\omega \cdot \mathbf{q}_{1}) \, (\mathbf{q}_{1} \mathbf{x} \omega) &= \omega^{1} \mathbf{q}_{\mathbf{j}} \cdot (\mathbf{q}_{1} \mathbf{x} \omega) &= \omega^{1} \omega \cdot (\mathbf{q}_{\mathbf{j}} \mathbf{x} \mathbf{q}_{1}) \\ \end{aligned}$$
 Тогда
$$\begin{aligned} \tau \cdot \mathbf{q}_{2} &= -\omega^{1} \omega \cdot \mathbf{q}_{3} &= -\omega^{1} \omega^{3} \\ \\ \tau \cdot \mathbf{q}_{3} &= \omega^{1} \omega \cdot \mathbf{q}_{2} &= \omega^{1} \omega^{2} \end{aligned}$$

Подставляя эти выражения в /5.20/ и воспользовавшись тем, что

$$\varepsilon_{\rm N}$$
 · ${\rm q}_2$ = ε^2 ; $\varepsilon_{\rm N}$ · ${\rm q}_3$ = ε^3

получим

$$M^{1} = -J_{s} \varepsilon^{1}$$

$$M^{2} = -J_{N} \varepsilon^{2} + J_{N} \omega^{1} \omega^{3}$$

$$M^{3} = -J_{N} \varepsilon^{3} - J_{N} \omega^{1} \omega^{2}$$

$$(5.21)$$

Сравнивая /5.21/ с /5.19/ видим, что при переходе к стержню был "потерян" момент

$$\Delta = J_s \omega^1 (\omega^2 q_3 - \omega^3 q_2)$$

Этот момент направлен перпендикулярно стержню и его модуль равен

$$\left[\Delta \mid = J_{s} \mid \omega^{1} \omega_{N}\right] \tag{5.22}$$

где ω -составляющая угловой скорости, перпендикулярная оси стержня.

$$\omega_{N} = (s \times \omega) \times s$$

Из /5.22/ следует, что $[\Delta M]$ -величина переменная: она равна нулю, когда вектор угловой скорости совпадает с осью цилиндра или перпендикулярен ей. Максимального значения $|\Delta M|$ достигает, когда угол между ω и осью цилиндра равен $\pi/4$.

Интересно отметить, что ΔМ завесит только от угловой скорости. Выпишем из уравнений Эйлера /5.19/ составляющую момента сил инерции, зависящую от угловой скорости; обозначим её

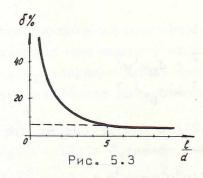
$$\dot{M}_{\omega} = (J_{N} - J_{s}) \omega^{1} (\omega^{3} q_{2} - \omega^{2} q_{3})$$
 (5.23)

Тогда относительную ошибку определим как отношение

$$\delta = \frac{\Delta M}{M_{(1)}}$$

Подставляя сюда значения $\Delta \dot{M}$ и $\dot{M}\omega$ из /5.22/ и /5.23/ и учитывая, что для цилиндра

$$J_N \simeq \frac{m}{12}L^2$$
; $J_S = \frac{m}{8}d^2$; $L = 21$,



$$\delta = \frac{2}{\frac{4}{3} \frac{L^2}{d^2} - 1}$$

На рис. 5.3 представлен график зависимости ошибки δ от отношения длины цилиндра к диаметру. Из графика, например, следует, что при отношении длины к диаметру больше пяти, эта ошибка меньше шести процентов.

§3 Дифференциальные уравнения движения цепей

Сначала рассмотрим разомкнутые цепи. Для составления дифференцияльных уравнений движения воспользуемся принципом Даламбера. Силы и моменты сил инерции звеньев сложим с внешними силами и моментами и полученные векторы обозначим $\Gamma_{\bf i}$ и ${\bf L_i}$.

$$\Gamma_{i} = f_{i} + G_{i}$$

$$L_{i} = M_{i} + M_{i}$$
(5.24)

Подставляя сюда выражения для сил и моментов инерции /5.2/, /5.10/ можем записать

$$\Gamma_{i} = \sum_{j=1}^{i} a_{ij} \ddot{x}^{j} + a_{i}^{0} - m_{i} w_{o} + G_{i}$$

$$L_{i} = \sum_{j=1}^{i} b_{ij} \ddot{x}^{j} + b_{i}^{0} - T_{i} \varepsilon_{o} + M_{i}$$
(5.25)

Изменим в /5.25/ верхний предел суммирования на n и положим

$$a_{ij} = b_{ij} = 0$$
 при $j > i$

Далее применим к уравнению /2.25/ оператор Ц с тем, чтобы разделить переменные относящиеся к кинематическим парам с силовым и кинематическим приводом* /см.гл.II,§3/

$$\ddot{y} = L(\ddot{x}, \eta);$$
 $\ddot{z} = L(\ddot{x}, \xi 2);$
 $a^{Y} = L(a, \eta);$ $a^{Z} = L(a, \xi 2);$
 $b^{Y} = L(b, \eta);$ $b^{Z} = L(b, \xi 2);$ (5.26)

^{*} Неприводные шарниры будем рассматривать как частный случай силового привода при $p^{\mathbf{i}} = 0$.

где:
$$\eta = (1-\xi 2_1, 1-\xi 2_2, \dots, 1-\xi 2_n)'$$

Тогда выражения для сил $\Gamma_{\mathbf i}$ и моментов $\mathbf L_{\mathbf i}$ можно записать в виде

$$\Gamma_{i} = \sum_{j=1}^{r} a_{ij}^{y} \ddot{y}^{j} + \bar{a}_{i} - m_{i} w_{o}$$

$$L_{i} = \sum_{j=1}^{r} b_{ij}^{y} \ddot{y}^{j} + \bar{b}_{i} - T_{i} \varepsilon_{o}$$

$$\bar{a}_{i} = a_{i}^{o} + \sum_{j=1}^{r} a_{ij}^{z} \ddot{z}^{j} + G_{i}$$

$$\bar{b}_{i} = b_{i}^{o} + \sum_{j=1}^{r} b_{ij}^{z} \ddot{z}^{j} + M_{i}$$
(5.27)

Законы относительного движения в шарнирах с кинематическим приводом известны, поэтому члены зависящие от \ddot{z} включины в выражения \ddot{a}_i и \ddot{b}_i .

Под действием Г, L и реакций приводов р цепь должна находиться в равновесии. Будем задавать виртуальные скорости в у-шарнирах. При этор примем, что выртуальных перемещений в шарнирах с кинематическим приводом не происходит.

Пусть задана скорость $\dot{\mathbf{y}}^{\mathbf{J}}$ в j-ом шарнире. Алгоритм СКОРОСТЬ позволяет рассчитать скорости центров масс $\mathbf{v}_{\mathbf{i}\mathbf{j}}$ и угловые скорости $\omega_{\mathbf{i}\mathbf{j}}$ звеньев цепи, а также для замкнутых цепей относительные скорости $\mathbf{x}_{\mathbf{j}}^{\mathbf{i}}$ в s-шарнирах. Тогда должно выполняться

$$\sum_{i=1}^{n} (\Gamma_{i} \cdot v_{ij} + L_{i} \cdot \omega_{ij} + p^{i}\dot{x}_{j}^{i}) = 0$$

$$(j = 1, 2, ..., n_{y})$$
(5.28)

где второй индекс ј при переменных ${f v}$, ${f \omega}$ и нижний индекс при ${f x}$ указывает, что эти скорости обусловлены только виртуальной скоростью ${f \dot y}^{f j}$.

Уравнение /5.28/ справедливо как для разомкнутых так и для замкнутых цепей. Однако, для разомкнутых цепей его можно записать проще. В разомкнутой цепи относительное пере-

мещение будет осуществляться только в одном j-ом шарнире, а звенья предшествующие этому шарниру в цепи, будут оставаться неподвижными. Поэтому, принимая для простоты $\dot{\mathbf{y}}^{\dot{\mathsf{J}}} = 1$ можно для разомкнутой цепи записать

$$\sum_{i=j}^{n} (\Gamma_{i} \cdot v_{ij} + L_{i} \cdot \omega_{ij}) = -p^{j}; (j = 1, 2, ..., n_{y}) (5.29)$$

Подставим в уравнение /5.28/ выражения для Γ и L /5.27/ и сгруппыруем члены при $\ddot{\mathbf{v}}$. В результате получим дифференци-альные уравнения движения в виде*

$$H\ddot{y} = H_1 p + H_2 w_0 + H_3 o + \zeta$$
 (5.30)

где H, H_1 , H_2 , H_3 - матрицы имеющие размерности соответсвено $n_y x n_y$; $n_y x n_3$

$$\begin{split} \mathbf{H}^{\dot{1}\dot{j}} &= \sum_{k=1}^{n} (\mathbf{v}_{k\dot{1}} \cdot \mathbf{a}_{k\dot{j}}^{Y} + \omega_{k\dot{1}} \cdot \mathbf{b}_{k\dot{j}}^{Y}); \\ \mathbf{H}_{\dot{1}}^{\dot{1}\dot{j}} &= -\dot{\mathbf{x}}_{\dot{\mathbf{i}}}^{\dot{j}}; \\ \mathbf{H}_{\dot{2}}^{\dot{2}\dot{j}} &= -\sum_{k=1}^{n} \mathbf{m}_{k} \mathbf{v}_{k\dot{\mathbf{i}}}^{\dot{j}}; \\ \mathbf{H}_{\dot{3}}^{\dot{1}\dot{j}} &= -\sum_{k=1}^{n} (\mathbf{T}_{k} \omega_{k\dot{1}})^{\dot{j}} \end{split}$$
 (5.31)

где верхние индексы ј при v_{ki}^j и $(T_k\omega_{ki})^j$ указывают, что берется ј-я компонента этих векторов.

Вектор ζ в /5.33/ определяется выражением

$$\zeta^{i} = -\sum_{k=1}^{n} (v_{ki} \cdot \bar{a}_k + \omega_{ki} \cdot \bar{b}_k)$$
 (5.32)

Как уже указывалось, при расчете виртуальных скоростей

^{*} Ниже трехмерные векторы везде обозначаются буквами обычного /не жирного/ шрифта.

считается, что в шарнирах с кинематическим приводом движение отсутсвует, поэтому соответствующие столбцы матрицы \mathbf{H}_1 будут состоять из нулей. Для того чтобы уменьшить колличество вычислений целесообразно сразу же исключить из рассмотрения информацию о реакциях кинематических приводов. Для этого воспользуемся оператором Ц /см. гл. II, § 3/ и выделим из \mathbf{P} подмножество реакций не кинематических приводов. Обозначим его \mathbf{P} , а его элементы \mathbf{P}^1 .

Тип привода указывает характеристическое число $\xi 2;$ для кинематических приводов $\xi 2_{\hat{1}} = 1$ /см.гл.І, $\hat{\$}$ 1/. Тогда

$$\bar{p} = \coprod (p \cdot \eta)$$

где
$$\eta = (1 - \xi 2_1, 1 - \xi 2_2, \dots, 1 - \xi 2_n)'$$
 (5.34)

Очевидно размерность вектора \bar{p} будет $\bar{n}=n-n_z$. В случае разомкнутой цепи $\bar{n}=n_y$.

Аналогично применим $\mathring{\mathbb{L}}$ к столбцам матрицы \mathbb{H}_1 . Получившуюся матрицу обозначим

$$H_4^{*i} = \coprod (H_1^{*i}, \eta);$$
 (5.35)

Тогда дифференциальное уравнение /5.30/ примет вид

$$H\ddot{y} = H_4 \bar{p} + H_2 w_0 + H_3 \epsilon_0 + \zeta$$
 (5.36)

где H_4 -матрица размерности n_{y} х n

$$\bar{p} = (\bar{p}^1, \bar{p}^2, \ldots, \bar{p}^{\bar{n}})'$$

Если задавать виртуальные скорости равными единице матрица ${
m H}_1$ превратится в единичную матрицу и дифференцияльное уравнение примет вид

$$H\ddot{y} = \bar{p} + H_2 w_0 + H_3 \varepsilon_0 + \zeta$$

Уравнения /5.36/ можно рассматривать как систему линейных алгебраических уравнений относительно $\ddot{y}^{\dot{j}}$. Решая эту систему получим

$$\ddot{y} = Ap + Bw_O + C\varepsilon_O + g \qquad (5.37)$$

где
$$A = H^{-1}H_4$$
; $B = H^{-1}H_2$; $C = H^{-1}H_3$; $g = H^{-1}\zeta$

Матрицы A, B, C имеют размерность соответственно $n_{y}x\bar{n}$, $n_{y}x^{3}$, $n_{y}x^{3}$. Если разомкнутая цепь присоединена к неподвижной стой-ке, то получим частный случай

$$\ddot{y} = A\bar{p} + g \tag{5.38}$$

Теперь переходим к составлению дифференциальных уравнений для замкнутых цепей. Для этого вернемся к уравнениям /5.25/. Если для разомкнутых цепей множества X можно было разделить только на два непустых подмножества Y и Z, то теперь можно выделить еще одно, третье, подмножество дополнительных координат S. Основные координаты отмечались характеристическим числом $\xi 5$ / $\xi 5_1$ =1 для основных координат, см.гл. II/. Аналогично тому, как это делалось в /5.26/ выделим из уравнений /5.25/ члены зависящие от S, применив оператор

$$\ddot{s} = L(\ddot{x}, \psi);$$

$$a^{S} = L(a, \psi);$$

$$b^{S} = L(b, \psi);$$

$$\psi = (1-\xi 5_{1}, 1-\xi 5_{2}, \dots, 1-\xi 5_{n})'$$
(5.39)

В результате выражения /5.27/ для сил Γ и моментов L теперь запишутся в виде

$$\Gamma_{i} = \sum_{j=1}^{n} a_{ij}^{y} \ddot{y}^{j} + \sum_{j=1}^{n} a_{ij}^{s} \ddot{s}^{j} + \bar{a}_{i} - m_{i} w_{o}$$

$$L_{i} = \sum_{j=1}^{n} b_{ij}^{y} \ddot{y}^{j} + \sum_{j=1}^{n} b_{ij}^{s} \ddot{s}^{j} + \bar{b}_{i} - T_{i} \varepsilon_{o}$$
(5.40)

где \bar{b}_i и \bar{a}_i определяются также как раньше /см./5.27//. Составим матрицы A_i^Y , A_i^S , B_i^Y , B_i^S из векторов-столбцов a_{ij}^Y , a_{ij}^S , b_{ij}^S следующим образом

$$A_{i}^{Y} = \|a_{i1}^{Y}, a_{i2}^{Y}, \dots, a_{in_{Y}}^{Y}\|; A_{i}^{S} = \|a_{i1}^{S}, a_{i2}^{S}, \dots, a_{in_{S}}^{S}\|;$$

$$B_{i}^{Y} = \|b_{i1}^{Y}, b_{i2}^{Y}, \dots, b_{in_{Y}}^{Y}\|; B_{i}^{S} = \|b_{i1}^{S}, b_{i2}^{S}, \dots, b_{in_{S}}^{S}\|;$$
(5.41)

Тогда /5.40/ можно записать в виде

$$\Gamma_{i} = A_{i}^{y} + A_{i}^{s} + \bar{a}_{i} - m_{i} w_{o}$$

$$L_{i} = B_{i}^{y} + B_{i}^{s} + \bar{b}_{i} - T_{i} \varepsilon_{o}$$

$$(5.42)$$

Добавочные ускорения можно найти с помощью связи Λ , Λ_1 , Λ_2 . Согласно /3.21/ имеем

$$\ddot{s} = \Lambda \ddot{u} + \Lambda_1 (w_n - w_o - \eta_n) + \Lambda_2 (\varepsilon_n - \varepsilon_o - \theta_n) \quad (5.43)$$

Применим к Λ и $\ddot{\mathfrak{u}}$ оператор \mathfrak{U} с тем, чтобы разделить переменные \mathfrak{y} и \mathfrak{z} . Координаты \mathfrak{z} отмечаются числом $\mathfrak{z}_{\mathfrak{z}}$ основные координаты – числом $\mathfrak{z}_{\mathfrak{z}}$. По условию все кинематические координаты включены в число основных. Тогда

$$\ddot{z} = \coprod (\ddot{x}, \xi 2);$$

 $\ddot{y} = \coprod (\ddot{x}, \xi 5 - \xi 2).$

В матрице Λ оператор Ц применим к каждой строке. Предварительно сформируем вспомогательные векторы $\eta 1$, $\eta 2$ размерности n_{η} .

$$\eta 1 = L(\xi 2, \xi 5);$$

 $\eta 2 = L(\eta, \xi 5).$

где

Теперь с помощью полученных векторов выделим из Λ строки, соответсвующие ${\bf z}$ и ${\bf y}$ координатам, и из них сфориру- ем две матрицы $\Lambda_{\bf z}$ и $\Lambda_{\bf y}$.

$$\Lambda_{y}^{i*} = \sqcup (\Lambda^{i*}, \eta_{1});$$

$$\Lambda_{z}^{i*} = \sqcup (\Lambda^{i*}, \eta_{2}).$$
(5.44)

Уравнение /5.43/ теперь можно записать в виде

$$\ddot{\mathbf{s}} = \Lambda_{\mathbf{y}} \ddot{\mathbf{y}} - \Lambda_{1} \mathbf{w}_{0} - \Lambda_{2} \varepsilon_{0} + \Lambda_{1} \mathbf{w}_{n} + \Lambda_{2} \varepsilon_{n} + \sigma, \tag{5.45}$$

$$\sigma = \Lambda_{\mathbf{z}} \ddot{\mathbf{z}} - \Lambda_{1} \eta_{n} - \Lambda_{2} \theta_{n}$$

Подставим это уравнение в /5.42/. Тогда

$$\Gamma_{i} = C_{i}\ddot{y} + C_{i}^{1}w_{o} + C_{i}^{2}\varepsilon_{o} + C_{i}^{3}w_{n} + C_{i}^{4}\varepsilon_{n} + \bar{c}_{i};$$

$$L_{i} = D_{i}\ddot{y} + D_{i}^{1}w_{o} + D_{i}^{2}\varepsilon_{o} + D_{i}^{3}w_{n} + D_{i}^{4}\varepsilon_{n} + \bar{d}_{i},$$
(5.46)

$$\Gamma_{AB} = C_{i} = A_{i}^{Y} + A_{i}^{S}\Lambda_{y};$$

$$C_{i} = A_{i}^{Y} + A_{i}^{S}\Lambda_{y};$$

$$C_{i}^{1} = -A_{i}^{S}\Lambda_{1} - m_{i}I;$$

$$D_{i}^{1} = -B_{i}^{S}\Lambda_{1} - T_{i};$$

$$C_{i}^{2} = -A_{i}^{S}\Lambda_{2};$$

$$D_{i}^{2} = -B_{i}^{S}\Lambda_{2};$$

$$D_{i}^{3} = B_{i}^{S}\Lambda_{1}$$

$$C_{i}^{4} = A_{i}^{S}\Lambda_{1};$$

$$D_{i}^{4} = B_{i}^{S}\Lambda_{2};$$

$$D_{i}^{5} = B_{i}^{S}\Lambda_{2}$$

где I - единичная матрица 3х3.

Далев подставим выражения /5.47/ для сил Г и моментов в уравнение /5.28/. После преобразований получим

$$H_{5}\ddot{y} = H_{4}\ddot{p} + H_{6}w_{o} + H_{7}\varepsilon_{o} + H_{8}w_{n} + H_{9}\varepsilon_{n} + \zeta_{1}$$

$$(5.48)$$

$$H_{5}^{j*} = \sum_{i=1}^{n} (v_{ij}C_{i} + \omega_{ij}D_{i});$$

$$H_{6}^{j*} = \sum_{i=1}^{n} (v_{ij}C_{i}^{1} + \omega_{ij}D_{i}^{1});$$

$$H_{7}^{j*} = \sum_{i=1}^{n} (v_{ij}C_{i}^{2} + \omega_{ij}D_{i}^{2});$$

$$H_{8}^{j*} = \sum_{i=1}^{n} (v_{ij}C_{i}^{3} + \omega_{ij}D_{i}^{3});$$

$$H_{9}^{j*} = \sum_{i=1}^{n} (v_{ij}C_{i}^{4} + \omega_{ij}D_{i}^{4});$$

$$\zeta_{1}^{j} = \sum_{i=1}^{n} (v_{ij} \cdot \bar{c}_{i} + \omega_{ij} \cdot \bar{d}_{i})$$

$$(j = 1, 2, \dots, n_{Y})$$

$$(5.49)$$

Здесь при умножении на матрицы векторы ${
m v}_{
m ij}$, ${
m \omega}_{
m ij}$ следует рассматривать как матрицы $1{
m x}3$.

Матрица ${
m H_4}$, стоящая перед вектором реакций приводов выражается также как и в уравнениях для открытых цепей /см. /5.35//, /5.36/.

Теперь решая /5.48/ относительно ў получим окончательно дифференциальные уравнения движения в виде

$$\ddot{y} = A\overline{p} + Bw_0 + C\varepsilon_0 + Dw_n + E\varepsilon_n + g$$
 (5.50)

где
$$A = H_5^{-1}H_4$$
; $B = H_5^{-1}H_6$; $C = H_5^{-1}H_7$; $D = H_5^{-1}H_8$; $E = H_5^{-1}H_9$; $g = H_5^{-1}\zeta_1$ (5.51)

Здесь матрицы при \overline{p} , w_{o} , ε_{o} и вектор g обозначены теми же буквами, что и в уравнении для разомкнутой цепи /5.37/. Однако, теперь они выражаются другими формулами.

Будем считать что /5.50/ являются общими дифференциальными уравнениями как для разомкнутой, так и для замкнутой цепи. В случае разомкнутой цепи матрицы A, B, C и вектор g определяются /5.37/, а матрицы D и E равны нулю. В случае замкнутой цепи матрицы A, B, C, D, E и вектор g определяются /5.51/.

В полученном основном дифференциальном уравнении /5.50/ матрицы A, B, C, D, E являются сложными нелинейными функциями относительных координат x, а вектор g зависит кроме того от относительных скоростей \dot{x} , ускорений в z-шарнирах и внешней нагрузки /G, M/. Эти зависимости можно проследить, если просмотреть все полученные ранее алгоритмы. Однако, окончательные формулы явно выражающие указанные матрицы и викторы через x, \dot{x} , G и M чрезвычайно громоздки. B данном слу-

чае они и не нужны, поскольку для определения значений компонент этих матриц в каждый момент времени можно построить
алгоритм. Фактически это уже было в основном выполнено раньше, и теперь осталось только добавить сравнительно небольшую
часть.

Важно отметить, что алгоритм для расчета A, B, C, D, E и g не ведет расчет по заранее полученным формулам, а фактически повторяет процесс вывода этих формул, осуществляя просчет всех промежуточных результатов, Поэтому описанный способ можно назвать машинным составлением уравнений.

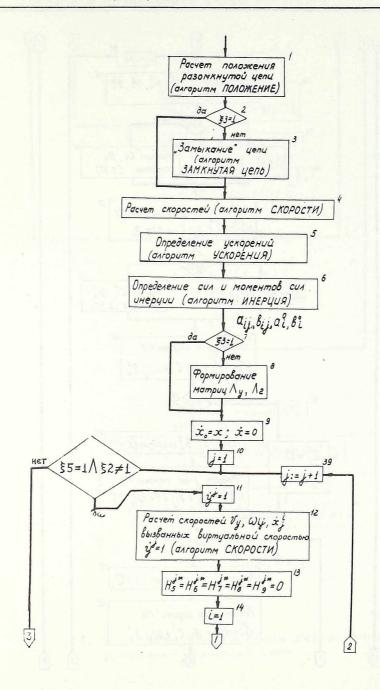
Блок-схема алгоритма для машинного составления урав-

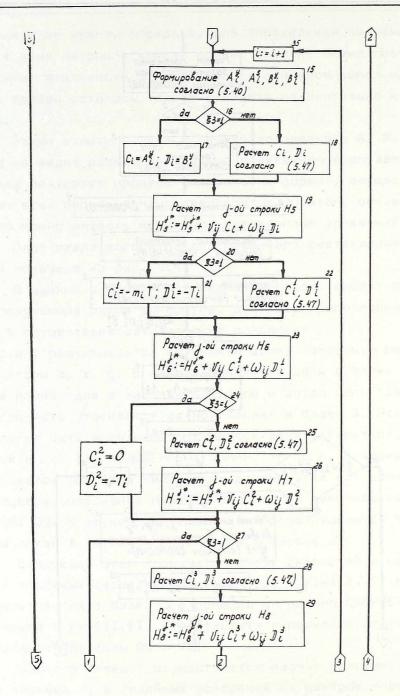
В данный алгоритм как составные части входят почти все полученные ранее алгоритмы. Для некоторого момента времени t осуществляются следующие операции:

В блоке I расчитывается положение цепи, то-есть определяются векторы е, r, q. В случае замкнутой цепи в блоке I проводится расчет для эквивалентной цепи и затем цепь "замыкается" /то-есть уточняется ее положение/ в блоке 3. Эти операции могут быть выполнены при помощи двух алгоритмов, полученных в гл. II: ПОЛОЖЕНИЕ и ЗАМКНУТАЯ ЦЕПЬ. В блоке 2 осуществляется проверка типа цепи: разомкнутая или замкнутая. В процессе выполнения первых трех блоков заполняются массивы EE, RR, QQ. В случае замкнутой цепи рассчитываются также матрицы связи Λ , Λ_1 , Λ_2 и помещаются в массив $\Lambda\Lambda$.

Следующий этап - расчет угловых скоростей всех звеньев $\omega_{\bf i}$ и линейных скоростей центров масс $v_{\bf i}/i=1,2,\ldots,n/$. Он осуществляется в блоке 3 с помощью алгоритма СКОРОСТИ, полученного в гл.III,§I. В результате выполнения этого блока заполняются массивы $\Omega\Omega$ и VV.

Далее в блоке 5 осуществяется расчет угловых ускорений всех звеньев $\varepsilon_{f i}$ и линейных ускорений их центров масс $w_{f i}$. При этом рассчитываются не сами ускорения, а векторные коэф-





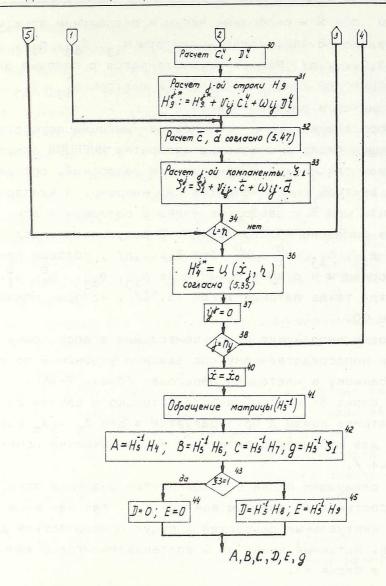


Рис. 5.4. Блок-схема составления дифференциальных уравнений движения цепи

фициенты при $\ddot{\mathbf{x}}$ и свободные члены в разложении этих ускорений согласно /3.11/, то-есть венторы $\alpha_{\mathbf{i}\mathbf{j}}$, $\beta_{\mathbf{i}\mathbf{j}}$, $\alpha_{\mathbf{i}}^{\mathsf{O}}$, $\beta_{\mathbf{i}}^{\mathsf{O}}$ /i, $\mathbf{j}=1,2,\ldots,n$ /. Расчет осуществляется с помощью алгоритма УСКОРЕНИЯ /см.гл. III, §2/ и венторы $\alpha_{\mathbf{i}\mathbf{j}}$, $\beta_{\mathbf{i}\mathbf{j}}^{\mathsf{O}}$, $\alpha_{\mathbf{i}}^{\mathsf{O}}$, $\beta_{\mathbf{i}}^{\mathsf{O}}$ помещаются в массивы \mathbf{W} и ϵ .

Определение сил и моментов сил инерции осуществляется в следующем блоке 6 с помощью алгоритма ИНЕРЦИЯ /см.гл. IV, $\S I/$. Здесь так же как и при расчете ускорений, определяются не сами векторы сил и моментов сил инерции, а векторные ко-эффициенты при $\ddot{\mathbf{x}}$ и свободные члены в разложении этих сил и моментов согласно /5.2/, /5.10/. В результате определяются векторы $\mathbf{a}_{\mathbf{i}\mathbf{j}}$, $\mathbf{b}_{\mathbf{i}\mathbf{j}}$, $\mathbf{a}_{\mathbf{i}}^{\mathsf{O}}$, $\mathbf{b}_{\mathbf{i}}^{\mathsf{O}}$ / \mathbf{i} =1,2,..., $\mathbf{n}/$, которые помещаются в массивы \mathbf{w} и ϵ вместо векторов $\alpha_{\mathbf{i}\mathbf{j}}$, $\beta_{\mathbf{i}\mathbf{j}}$, $\alpha_{\mathbf{i}}^{\mathsf{O}}$, $\beta_{\mathbf{i}}^{\mathsf{O}}$. Рассчитываются также матрицы \mathbf{T} /см./5.10//, которые помещаются в массив QQ.

После выполнения подготовительных этапов можно приступать к непосредственному составлению уравнений по способу, описанному в настоящем параграфе /блоки 7-45/.

В блоке 8, который работает только в случае замкнутой цепи, матрица связи Λ преобразуется в две $\Lambda_{\mathbf{y}}$ и $\Lambda_{\mathbf{z}}$ соответственно для шарниров с силовым и кинематическим приводом /см./5.44//.

В следующем блоке 9 запоминаются значения осносительных скоростей $\dot{\mathbf{x}}$ /с помощью вектора $\dot{\mathbf{x}}_{0}$ /, так как ниже при расчете виртуальных скоростей $\dot{\mathbf{x}}$ будут присваиваться другие значения. Истинные значения $\dot{\mathbf{x}}$ востанавливаются в конце алгоритма в блоке 40.

Цикл с параметром ј /блок II-39/ осуществляет следующие операции: задает виртуальные скорости последовательно во всех у-шарнирах и рассчитывает мощность внешних силинерции /и моментов/. В результате рассчитываются матрицы H_4 , H_5 , H_6 , H_7 , H_8 , H_9 для дифференциального уравнения в форме /5.48/. Здесь значения ј указывает номер у-шарнира,

в котором задается виртуальная скорость.

Внутренний цикл с параметром і /блоки 15-35/ обходит все звенья цепи /і указывает номер звена/ и рассчитывает мощность внешней нагрузки G, M и сил и моментов инерции f, μ , для случая когда задана виртуальная скорость шарнира типа Y с номером J. Расчет матриц $H_5 \dots H_9$ осуществляется согласна формулам /5.47/. В блоках 16,20,24,27 проверяится тип цепи и в расчет вносятся необходимые коррективы для замкнутых и разомкнутых цепей. Матрица H_4 рассчитывается вне цикла по I0 с помощью процедуры I1 с помощью процедуры I2 комот I3 гомощью процедуры I3 гомощью процедуры I4 гомощью процедуры I4 гомощью процедуры I6 гомощью процедуры I4 гомощью процедуры I6 гомощью процедуры I1 гомощью процедуры I3 гомощью процедуры I4 гомощью процедуры I6 гомощью процедуры I4 гомощью процедуры I6 гомощью процедура I8 гомощью процедура

Последние блоки /41-45/ осуществляют переход от дифференциального уравнения в форме /5.48/ к уравнению в форме /5.50/, то-есть рассчитывают матрицы A, B, C, D, E и многомерный вектор g.

Ниже приводится текст алгоритма машинного составления уравнений /процедура ДИФ УР/. Формальные параметры:

N - номер цепи;

A1, A2 $\begin{bmatrix} 1:3, \ 1:3 \end{bmatrix}$ — матрицы перехода для базовых звеньев; v0, $\omega0$, $\epsilon0$, — векторы угловых и линейных скоростей и усноw0 $\begin{bmatrix} 1:3 \end{bmatrix}$ — рений начального базового звена; vn, ωn , ϵn , — векторы угловых и линейных скоростей и ускорений для конечного базового звена; vn — рений для конечного базового звена; vn — рений для конечного базовых звеньев;

A, B, C, D, матрицы в дифференциальном уравнении /5.50/; E, g рюссчитываются в результате выполнения алгоритма;

є - дийствительное число, определяющее точность вычислений положения замкнутой цепи;

Параметры A2, vn, ωn , wn, εn , hb, ε задаются только для зам-кнутых цепей.

Глобальные массивы STR, KP, ZV /см.гл.I/.

```
procedure ΔΜΦ ΥΡ (N,A1,A2, v0,ω0, vn,ωn,
            v0, \varepsilon0, wn, \varepsilon n, hb, \varepsilon, A, B, C, D, E, g);
value N; integer N;
array A1,A2,v0,\omega0,vn,\omegan,w0,\epsilon0,wn,\epsilonn,hb,
       A, B, C, D, E, g;
real \epsilon;
begin
     integer n, nx, ny, nz, nu, nn, ns;
     n: = STR[N,6]; nu: = STR[N,7], ny: = STR[N,8];
     nz: = nu-ny; ns: = n-nu; nn: = n-nz;
begin
    array a[1:n], Ay[1:ns,1:ny], Az[1:ns,1:nz],
   \Lambda 1, \Lambda 2 [1:ns, 1:3], W, \varepsilon [0:n, 1:n+1, 1:3],
     H5[1:ny,1:ny], b[1:3];
    integer i,j,k,p,r,q;
    for i: = 1,2,3 do b[i]: = 0;
     ПОЛОЖЕНИЕ (N);
     if STR[N,1] = 0 then
     ЗАМКНУТАЯ ЦЕПЬ (N,A2,hb,\epsilon);
     CHOPOCTH (N, v0, \omega0, vn, \omega n);
     ΥCHOPEHUЯ (N,w0,ε0,wn,εn,w,ε,false);
     ИНЕРЦИЯ (N,W,\varepsilon, false);
comment приступаем к формированию матриц
     Λ1, Λ2, Λy, Λz;
     for i: = 1 step 1 until ns do
     for j: = 1,2,3 do
    begin \Lambda l[i,j] := \Lambda \Lambda[N,i,j+nu];
            \Lambda 2[i,j] := \Lambda \Lambda[N,i,j+nu+3]
     end;
     k := p := r := 0;
for i: = 1 step 1 until n do
    \underline{if} KP[N,i,3] = 1 then
     begin k: = k+1;
```

```
if KP[N,i,2] = 1 then
                                            begin p: = p+1;
                                            for j: = 1 step 1 until ns do
                                                                      \Lambda z[j,p] := \Lambda \Lambda[N,j,k]
                                            end else
                                           begin r: = r+1;
                                           for j: = 1 step 1 until ns do
                                                                      \Lambda y[j,r] := \Lambda \Lambda[N,j,k]
                                           end
                          end цикла по і и расчета матриц \Lambda 1, \Lambda 2, \Lambda y, \Lambda z;
                          for i: = 1 step 1 until n do begin
                                           dx[i] := x[N,i,2]; x[N,i,2] := 0 end;
                       j := 1;
comment начало цикла по ј, В котором расчитываются матрицы
                                   H_4, H_5 ,..., H_0 и ζ, см блон-схему;
                          mj: if KP[N,j,3] = 1 \land KP[N,j,2] \neq 1 then
                         begin X[N,j,2]: = 1;
                          CKOPOCTH (N,b,b,b,b);
                          ИМЯ (Н4, А); ИМЯ (Н6, В); ИМЯ (Н7, С);
                          ИМЯ (H8,D); ИМЯ (H9,E); ИМЯ (\zeta,a); да ва от тоск
                         for i: = 1 step 1 until ny do
                          H5[j,i]:=0;
                          for i: = 1, 2, 3 do
                         H6[j,i]: = H7[j,i]: = H8[j,i]: = H9[j,i]: = 0;
                         for i: = 1 step 1 until n do
                          begin
                          array Ay, By[1:3, 1:ny], As, Bs[1:3, 1:ns],
                                                    a,b[1:1, 1:3];
                          p: = r: = 0;
                         if KP[N,k,2] \neq 1 then begin (\sqrt{a}, \sqrt{a}, \sqrt{a
                         if KP[N,k,3] = 1 then begin
```

```
p: = p+1;
   for g: = 1,2,3 do begin
   Ay[q,p] := W[i,k,q];
   By[q,p] := \varepsilon[i,k,q]
   end end
   else begin r: = r+l;
   for q: = 1,2,3 do begin
   As[q,r]:=W[i,k,q];
   Bs[q,r] := \varepsilon[i,k,q]
   end end end цикла по k;
   for k: = 1,2,3 do
   \underline{\text{begin}} \ a[1,k] := VV[N,i,k];
       b[1,k] := \Omega\Omega[N,i,k] \underline{end};
comment приступаем к расчету матрицы H5;
   begin
      array C,D[1:3, 1:ny];
    if STR[N,1] = 0 then
    begin
      MTMT(C, As, Ay, 3, ns, ny);
      MTMT(D,Bs,Ay,3,ns,ny);
      for k: = 1 step 1 until ny do
      for p: = 1,2,3 <u>do begin</u>
      C[p,k] := C[p,k] + Ay[p,k];
     D[p,k] := D[p,k] + By[p,k]
    end end
    else for k: = 1 step 1 until ny do
       for p: = 1,2,3 do
       begin C[p,k]: = Ay[p,k];
       D[p,k] := By[p,k] end;
    MTMT(C,a,C,1,3,ny); about 1 202
    MTMT(D,b,D,1,3,ny);
    for k: = 1 step 1 until nu do 1 = 2 4 4 9 3 1
       H5[j,k] := H5[j,k] + C[1,k] + D[1,k]
    end блока;
```

 $\frac{\text{comment}}{\text{H}_6 \cdots \text{H}_9}$ и вектор ζ . Составление этих матриц выполняется согласно блок-схеме и уназанным формулам. Это чисто техническая часть алгоритма, и мы позволим себе не выписывать её подробно. Запишем только последние операторы цикла по j и будем считать что в этом цикле /см. блок-схему/ составляются матрицы $\text{H}_4 \cdots \text{H}_9$ и ζ ;

§4 Моделирование пространственных цепей

При конструировании сложных механизмов весьма важно иметь возможность оценить различные варианты конструкции с точки эрения их динамических свойств и нагрузок. Осуществлять физическое моделирование /то-есть строить уменьшенную копию механизма/ обычно долго и дорого, и к нему приходится прибегать в крайних случаях.

Другой путь состоит в моделировании механизма на цифровых ЭВМ. При этом под моделью понимается такая программа,
которая, образно говоря, с точки зрения динамики "ведет себя
как настоящий механизм". Задавая в такую модель законы изменения управляющих воздействий в приводах и внешних сил можно
на выходе получить законы движения всех звеньев механизма,
динамические нагрузки, динамическое рассогласование/то-есть
отклонение от "идеальных" законов/ и тд. Модель должна также давать возможность решать обратную задачу, то-есть определить реакции приводов, при которых обеспечивается заданное
движение механизма. Попутно, конечно, могут быть определены
все динамические нагрузки.

Моделирование на ЭЦВМ имеет очевидные преимущества в смысле дешевизны и, главное, универсальности. На такой модели не представляет труда изменить параметры механизма, в том числе и конструктивные, и "просмотреть" различные варианты. Однюко, моделирование на ЭЦВМ представляется делом очень не простым, если идти по "традиционному пути", то-есть сначала выводить дифференциальные уравнения, а затем их программировать. Вывод этих уравнений чрезвычайно трудоемок и не внушает доверия для сложных механизмов, ибо полученные необозримые формулы очень трудно проверить. Кроме того, при

изменении конструктивных параметров /например типа шарниров/ большую часть выводов нужно повторять. Строго говоря такой путь нельзя назвать моделированием, а скорее расчетом по готовым формулам.

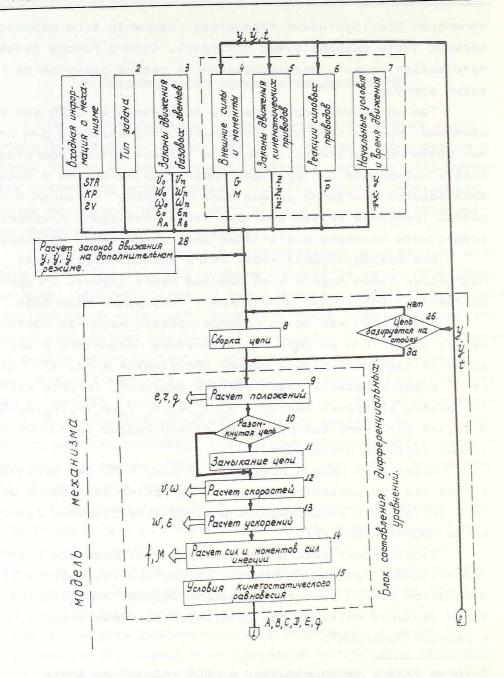
Тем не менее, моделирование механизмов на ЭЦВМ все же возможно, причем весьма универсальное, справедливое для самых различных механизмов, При этом человек должен подготовливать информацию о структуре, параметрах механизма и решаемой задаче. Программа должна сама "собирать" механизм и выдавать требуемые результаты. Для этого необходимо главное: осуществить машинное составление дифференциальных уравнений.

Вся предшествующая часть книги давала материал для подготовки такой модели и её основой может служить алгоритм составления уравнений, полученный в предыдущем параграфе.

Блок-схема модели приведена на рис. 5.5. Она включает в себя два уже рассмотренных ранее алгоритма для сборки цепи и составления дифференциальных уравнений движения /алгоритмы СБОРКА и ДИФ УР/.

Модель может работать в двух режимах, основном и дополнительном. Разделение режимов осуществляется в блоке 22. В основном режиме решается следующая задача: определяется закон движения механизм под действием известных внешних сил и реакций приводов*.

^{*} Такую задачу часто называют второй задачей динамики



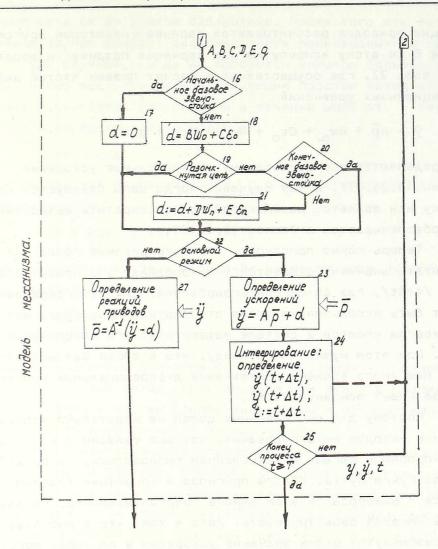


Рис. 5.5. Блок-схема моделирования пространственного механизма, садержащего одну цепь

Реакции приводов рассчитываются заранее в некотором другом блоке 6 /к этому вопросу мы еще вернемся позднее/ и вводятся в блок 22, где осуществляется расчет правых частей дифференциальных уравнений

$$\ddot{y} = A\bar{p} + Bw_o + C\varepsilon_o + Dw_n + E\varepsilon_n + g$$

и определяются все неизвестные относительные ускорения /блоки 18,21,27/. В тех случаях, ногда цепь базируется на стойку или является разомкнутой можно сократить вычисление, что обеспечивается в блоках 16,19,20.

Теперь можно приступить к интегрированию /блок 24/ и расчитать значения скоростей \dot{y} и координат y в момент времени /t+ Δ t/, где Δ t-шаг интегрирования. Для интегрирования может быть использована любая стандартная процедура, которые всегда имеются в системе математического обеспечения ЭВМ*. При этом нужно иметь ввиду, что в общем балансе затрат машинного времени составление диференциальных уравнений занимает основную часть.

Поэтому для многозвенных цепей не жедательно использование методов интегрирования, которые связаны с многократными пробными шагами и вычислением производных, например,
метода Рунге-Кутта. Методы прогноза и коррекции /например
Адамса - Башфорда/ в этом смисле предпочтительнее, однако,
вдесь имеется своя трудность. Дело в том, что в многозвенных разомкнутых цепях значения ускорений в шарнирах могут
в процессе движения изменяться на несколько порядков и, следовательно, интегрирование желательно вести с переменным
шагом. Для реализации методов прогноза и коррекции с переменной величиной шага нужны уже довольно сложные программы,

^{*} Относительно различных методов интегрирования см. Hamming R.W. Numerical methods, Mc.Graw-Hill, 1962.

которые есть не во всякой библиотеке. Кроме того эте методы в лучшим случае требуют два вычисления производных на один шаг. В тех случаях, когда не требуется слишком большая точнасть, можно воспользоваться следующим простым методом: ускорения \ddot{y} считаются постоянными в течение шага Δt , а величина шага определяется из условия

$$\Delta t = \epsilon/\beta$$

где ε-задаваемая малая величина, а β определялось по следующему выражению

$$\beta = \max_{(i)} ([\ddot{y}^{i}(t)], [\dot{y}^{i}(t)])$$

Тогда приращения фазовых координат на одном шаге интегрирования удовлетворяют условиям

$$\Delta \dot{y}^{1} \leq \varepsilon$$
 $\Delta y^{i} \leq \varepsilon$, $i = 1, 2, ..., n_{y}$

и задавая различные значения є можно регулировать точность вычислений. Такой метод использовался при анализе разомкнутых механизмов с шестью степенями свободы и показал вполне удовлетворительную точность и быстодийствие.

После интегрирования определяются новые значения фазовых координат $y(t+\Delta t)$, $\dot{y}(t+\Delta t)$. Эти значения могут рассматриваться как начальные условия для следующего шага интегрирования. Если процесс не закончен, то эта информация подается по правой цепи обратной связи /рис. 5.5/ к начаяу алгоритма. Таким образом на каждом шаге происходит составление уравнений и затем их интегрированые. Процесс продолжается до тех пор Пока время t не станет равно /или больше/ заданного значения T /блок 25/.

Если базовые звенья неподвижны, то сборку можно осуществить только один раз, в начале работы алгоритма, а новые начальные условия посовать сразу на блок составления уравнений. В случае подвежных базовых звеньев сборку приходится выполнять на каждом шаге интергирования, так как изменяется положение крайних шарниров цепи.

Пунктирная стрелка от блока интегрирования к цепи обратной связи указывает, что при некоторых метадах интегрирования обращение к блоку составления уравнений осуществляется несколько раз в течение одного шага.

Помимо описанного основного режима возможен "дополнительный" режим работы, на который осуществляется выход по стрелке "нет" из условного блока 22. На этом режиме решается следующая задача: считаются известными законы движения звеньев механизма и определяются реакции приводов, обеспечивающие такой динамический режим*.

В этом случав все приводы, реакции которых нас интересуют должны быть описаны как силовые /то-есть соответстующие характеристические числа $\xi 2_i$ дожны быть равны двум/. Тогда реакции этих приводов войдут в число компонент вектора \bar{p} /см.§ 3/. На дополнительном режиме известны законы изменения всех относительных координат, и, следовательно, известны \bar{y} . Обозначив

$$d = Bw_0 + C\varepsilon_0 + Dw_0 + E\varepsilon_0 + g$$

где все члены в правой части известны, из линейного уравнения

$$Ap = \ddot{y} - d$$

можно определить реакции приводов, что осуществляется в блоках 18,21,27.

Расчет р может быть выполнен для любого момента времени, для которого известны эначения относительных координат, скоростей и ускорений. Эти значения далжны либо вводиться в виде таблиц, либо расчитываться в некотором блоке вне

^{*} Задачи такого типа часто называются первыми задачами динамики.

модели механизма /на рис. 5.5, блок 28/.

Помимо описанных двух задач динамики модель позволяет получить также разнообразную другую информацию о двежении механизма. Действительно, в процесс составления уравнений приводится расчет положений скоростей, ускорений, сил и мюментов сил инирции и вся эта информация может быть выведена на печать /см. пунктирные двойные стрелки на блок-схеме рис. 5.5/.

Для того чтобы модель "работала" необходимы некоторые вспомогательные блоки, обслуживающие модель. С одням таким блоком мы уже встретились /блок 28/ при описании даполнительного режима. Остальные блоки показаны в верхней части схемы на рис. 5.5. С помощью этих блоков подготавливается и вводится информация о 1. структуре и параметрах механизма /блок 1/, 2. рабочем режиме /блоки 4-7/, 3. движении бозовых звеньев /блок 3/, 4. характере решаемой задаче /блок 2/.

Подготовка информации о структуре и параметрах падробно обсуждалась в гл. I. Там эта информация объединялясь в три массива STR , KP и ZV, которые вводятся в начале работы программы.

Все остальные вспомогательные блоки /2-7, 27/ далжны программироваться в зависимости от конкретной задачи. В блоке 4 рассчитывается внешняя нагрузка, в блоке 5 - закан движения в шарнирах с кинематическими приводами, в блоке 6 - реакции силовых приводов. Для работы этих блоков может понадабиться информация о фазовых координатах механизма и времени, поэтому к этим блокам подходит цепь обратной связи /см. рис. 5.5/.

Описанная модель позволяет решать различные динарические задачи. Например, если в блоке 6 реакции приводов определять как

$$p^{i} = k^{i} (a^{i} - \bar{x}^{i})$$
 (5.52)

где \bar{x}^i -относительные координаты в шарнирах, где дейстуют $\bar{p}^i;$ a^i, k^i - положительные константы,

то получим модель механизма с линейными связями. В этом случае можно изучать, напремер, собственные колебания механизма. На сложно ввести в упругие связи и некоторые нелинейные черты, например

$$\bar{p}^{i} = k_{1}^{i}(a^{i} - \bar{x}^{i}) + k_{2}^{i}(a^{i} - \bar{x}^{i})^{3}$$
 (5.53)

или учесть демпфирование

$$\bar{p}^{i} = k_{1}^{i}(a^{i}-\bar{x}^{i}) + k_{2}^{i}(a^{i}-\bar{x}^{i}) - c_{1}^{i}\bar{x}^{i}-c_{2}^{i}(\bar{x})^{2} \operatorname{sign}(\bar{x})$$
 (5.54)

Для изучения вынужденных колебаний механизма можно ввести гармоническое возбуждение в реакции приводов

$$\bar{p}^{i} = k^{i}(a^{i} - \bar{x}^{i}) + A \sin\omega t + B \cos\omega t \qquad (5.55)$$

или задать гармоническое возбуждение каких-либо кинематических координат

$$z^{j} = A_{1} \sin \omega t + B_{1} \cos \omega t$$

Если гармонический закон движения задать для базовых звеньев /то-есть ${
m R}_{
m A}$ или ${
m R}_{
m B}/$, то получим механизм на вибрирующем основании.

Если в формулах /5.52/-/5.55/ задавать параметры ${\bf a}^{\bf i}$ как функции времени, то получим модель следящей системы, исполнительным органом которой является данный механизм. Для таких следящих систем модель позволяет расчитать переходные процессы и подобрать оптимальные значения коэффициентов усиления ${\bf k}^{\bf i}$.

Перечисленные примеры представляют собой лишь малую часть задач, которые можно решать с помощью такой модели. Возможности модели в этом смысле трудно переоценить. Мы остановимся еще только на одной: внесение изменений в "конструкцию" механизма".

В модели это дастигается максимально просто. Напри-

на вращательный достаточно заменить всего одно число в матрице КР. Если требуется замкнуть или разомкнуть цепь, присоединить ее к другим цепям и т.п. все это достигается также за счет изменения нескольких чисел во вводимых массивах STR. KP, ZV.

Важно отметить, что модель не рассчитана на какой-то один механизм. В алгоритм модели фактически заложены основные законы механики и последовательность их применения. Модель начинает описывать тот или иной конкретный механизм только после того как в нее введены данные структуре и параметрах этого механизма. Тогда в алгоритме "собирается" заданный механизм и затем применительно к нему решаются задачи динамики. Такое построение алгоритма делает его пригодным для моделирования самых различных замкнутых и разомкнутых механизмов.

Гл. VI ВЗАИМОСВЯЗАННЫЕ ЦЕПИ

§1 Геометрия взаимосвязанных цепей

Введем некоторые новые термины и обозначения. Пусть имеется некоторая цепь и к ее внутренним /не базовым/ звеньям присоединены другие цепи. Эти присоединенные цепи будем называть сателлитами, а основную цепь базой или базовой цепью.

Введем понятие автономности цепи. Цепь будем называть автономной в том случае, если ег "сборку", расчет положений и скоростей можно осуществлять независимо от ег сателлитов, и кроме того ег базовые звенья также принадлежат автономным цепям или являются стойкой. При таком определении необходимо, чтобы по крайней мере одна цепь механизма базировалась только на стойку. Если механизм целиком состоит из автонамных цепей, то сателлиты не накладывают кинематических связей на базовые цепи.

Во многих случаях механизм можно разбить на цепи таким образом, что все они будут автономны. Например. модель "человека", приведенная во введеии рис. В.4 состоит из автономных цепей. Действительно разомкнутые цепи "туловища" и "рук", очевидно, не накладывают кинематических связей на движение ног.

Разбиение механизма на цепи обычно можно выполнить различным образом. Рассмотрим это на примере плоского семизвенного механизма, приведенного на рис. 6.1. Механизм обладает двумя степенями свободы, что следует из общей структурной формулы*

^{*} Артоболевский И.И. "Теория механизмов и машин" Москва, 1953. г.

w = 3n - 2p

где n-число подвижных звеньев; p-число кинематических пор.

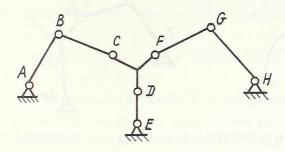


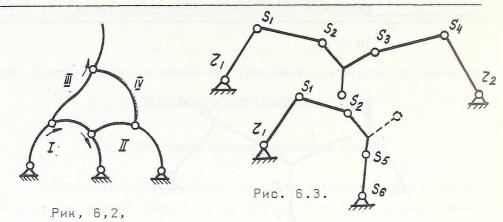
Рис. 6.1.

Допустим, привод в шарнирах A и B осущетвляется кинематически, тогда относительные координаты в этих шарнирах мы должны принять за основные.

Рассмотрим два варианта разбиения механизма на цепи. Сначала объединим в одну /первую/ цепь кинематические пары A, B, C, D, E, а в другую цепь - пары F, G, H. При таком варианте цепи получаются автономными. Дийствительно, задав относительное положение звеньев в шарнирах A, B, мы тем самым однозначно определим положение всех звеньев первой цепи; сателлит/цепь F, G, H/ является ведомой частью и не накладывает на базу кинематических связей.

Другой вариант разбиения: в первую цепь объединяем кинематические пары E, D, F, G, H, а во вторую - A, B, C. Теперь цепи не являются автономными. Сателлит/A,B,C/ теперь целиком определяет положение звеньев базовой цепи.

Как видно, автономность может явиться следствием удачного разбиения на цепи. Однако, не всегда механизм можно разбить на автономные цепи. Покажем это вновь на примере механизма на рис. 6.2. Предположим, что кениматическим приводом оснащены не шарниры A и B как раньше, а шарниры A и H. Тогда при любом варианте разбиения, цепи не получаются автономными. Действительно для определения положения звеньев, на-



пример, цепи ABCDE недостаточно знать угол поворота только в одном шарнире A, но нужно также знать и положение сателлита FGH.

Рассмотрим сначала случай, когда механизм можно разбить на автономные цепи и допустим такое разбиение выполнено. Тода можно значительную часть анализа /в частности анализа геометрии/ проводить отдельно для каждой цепи.

В предшествующих четырех главах рассматривались отдельные цепи разомкнутые и замкнутые, и при этом все время предполагалось, что базовые звенья могут принадлежать другим целям. По этому, все результаты полученные выше можно использовать при анализе взаимосвязанных автономных цепей и задача состоит в том, чтобы найти соответствующие характеристики для базовых звеньев.

Для того чтобы воспользоваться алгоритмами полученными в гл. II нужно для каждой разомкнутой цепи знать расположение в пространстве ортов \mathbf{q}_{oj} /j=1,2,3/ для внутренней систимы координат начального базового звена. Дла замкнутых цепей нужно также знать аналогичные орты для конечного базового звена \mathbf{q}_{nj} /j=1,2,3/ и вектор \mathbf{h}_B от полюса начального базового звена к полюсу конечного базового звена. Рассмотриму как можно получить эту информацию для механизмов с автономными цепями.

В механизме содержащем автономные цепи всегда имеется по крайней мере одна цепь, которая

- 1. либо является разомкнутойи соединена со стойкой;
 - 2. либо является замкнутой и двумя концами соединена со стойкой.

Случаи, когда механизм вообще не связан со стойкой будут рассмотрены позднее.

Одну из указанных в пп 1,2 цепей обозначим номером 1. Для нее известно положение одого или обоих базовых звеньев /поскольку базами служит стойка/ и, следовательно, можно эту цепь собрать и переместить с помощью алгоритмов СБОРКА РАЗ ЦЕПИ, ПОЛОЖЕНИЕ, ЗАМКНУТАЯ ЦЕПЬ /см. гл. II/.

На рис. 6.2 показан условный пример. где цепи обозначены просто линиями, а их соединения - точками. Стрелки указывают направления базовых цепей и служат для того, чтобы отличать направление базовой цепи от сателлита. Здесь цепь 1 замкнутая.

Далее, после того кок выделена цепь 1, среди оставшихся имеется по нрайней мере одна, которая является сателлитом цепи 1 и при этом

- 1. либо яляется открытой;
- 2. либо является закрытой и конечным базовым эвеном для нее служит_стойка;
- 3. является закрытой и конечным базовым звеном для нее служит одно из звеньев цепи I.

На рис. 6.2 имеются сразу две цепи, удовлетворяющие условиям п 1 и 2. Эти цепи прономерованы III и II.

Далее, среди оставшихся цепей имеется по крайней мере одна, для которой базовые звенья находятся в уже прономерованных цепях или являются стойкой /на рис. цепь IV/. Действуя таким образом можно прономеровать все цепи механизма. Поскольку кажная цепь базируется либо на предшествующие цепи, положение которых уже известно, либо на стойку, то из-

вестны q_{oj} , q_{nj} и h_B и можно воспользоваться алгоритмами гл. II.

Алгоритм для сборки механизма с автономными цепями имвет следующий простой вид

procedure CEOPHA MUXAHU3MA (Nc): value Nc; integer Nc; begin integer i, n, j, k, s, nl, Nl; array rl, r2, e[1:3], A[1:3,1:3]; for i: = 1,2,3 do begin for j: = 1,2,3 do A[i,j]: = 0; A[i,i]: = 1 end for i: = 1 step 1 until Nc do begin n: = STR[i,6]; if $i \neq 1$ then begin N1: = STR[i,2]; n1: = STR[i,3]; for k: = 1,2,3 do for s: = 1,2,3 do A[k,s] := Q0[i,j,k,s] end;for j: = 1 step 1 until n do begin C5OPKA ПАРы (i,j,rl,e,r2,A); for k: = 1,2,3 do begin E0[i,j,k] := e[k]; R0[i,j,l,k] := r1[k];R0[i,j,2,k] := r2[k] end;for s: = 1, 2, 3 do Q0[i,j,k,s] := A[k,s] end end j,i; end процедуры;

В результате выполнения этой процедуры полностью запалняются массивы E0, R0, Q0, где хранятся значения соответствено e_i , r_{ij} , q_{ij} после сборки. Формальный параметр Nc указывает число цепей в механизме. Предполагается, что цепи прономерованы таким способом, как описано выше, и первая цепь базируется на стойку. Поэтому в начале выполнения процедуры мотрица перехода A формируется как единичная матрица. Глобальные массивы: STR, KP, ZV, E0, R0, Q0.

После сборки необходимо рассчитать положение механизма, соответствующее заданным значениям относительных координат. Для этого можно воспользоваться алгоритмами ПОЛОЖЕ-НИЕ и ЗАМКНУТАЯ ЦЕПЬ.

При расчете положения замкнутой цепи необходимо знать матрицы перехода для базовых звеньев и вектор \mathbf{h}_{B} между полюсами базовых звеньев /см. гл. II §3/. Матрицы связи после сборки хранятся в Q0.

Если базой служит стойка, то удобно принять в качестве полюса начало неподвижной системы координат. Тогда, например, если замкнутая цепь обоими канцами базируется на стойку, то $\mathbf{h_B} = \mathbf{0}$. В других случаях необходимо специально рассчитывать $\mathbf{h_B}$ перед тем, как обратиться к алгоритму ЗАМ-КНУТАЯ ЦЕПЬ.

Обозначим радиус-векторы из начала неподвижной системы координат к полюсам базовых звеньев как R_{ij} , где первый индекс i - указывает номер цепи, а второй j - указывает к какому из базовых звеньев проведен радиус. / в случае j=1 к начальному базовому звену, а j=2 - к конечному/. Тогда

$$h_{Bi} = R_{i2} - R_{i1}$$
 (6.1)

Примем; что для стойки $R_{\mbox{ij}}=0$, а для подвижных базовых звеньев в качестве полюса выбирается их центр масс.

В случае когда механизм состоит из автономных цепей, каждая i-я цепь базируется либо на стойку, либо на цепи с номером меньшим i, для которых известны положения всех эвеньев.

Тогда для определения R_{ij} будем действовать следующим образом. Прежде всего выделяем все цепи, которые базируются на стойку. Для этого просматриваем 2-ой столбец матрицы STR и если в i-ой строке стоит ноль, эначить i-я цепь имет стойку начальным базовым звеном, то-есть $R_{il}=0$. Значение R_{il} занесем в массив RR, где хранятся все векторы r_{ij} . В массиве RR свободны ячейки /см. описание массива, гл I, § 4/.

RR[i,0,1,k]; k = 1,2,3

и в эти ячейки поместим $R_{il} = 0$.

После того как просмотрены все начальные базовые звенья, переходим к 4-ой строке матрицы STR, где помещены номера конечных базовых цепей. Если в i-ой строке стоит ноль, это означает, что конечное базовое звено является стойкой.

Радиусы $R_{ ext{i}\,2}$ будем помещать также в массив RR в следующие ячейки.

$$RR[i,n,2,k];$$
 $k = 1,2,3$

В случае базирования на стойку заносим в эти ячейки $R_{12} = 0$. Первая цепь /при принятом способе номерации/ непременно далжна базироваться на стойку. Следовательно, если она замкнутая, то для нее

$$h_{B1} = 0$$

Теперь можно определить положение первой цепи с помощью алгоритмов ПОЛОЖЕНИЕ и ЗАМКНУТАЯ ЦЕПЬ. Вторая цепь может базироваться либо на первую цепь, либо на стойку. Если, например, начальное базовое эвено принадлежит первой цепи и имеет в этой цепи номер n1, то радиус вектор R_{21} можно определить как

$$R_{21} = R_{11} + \sum_{j=1}^{n_1} r_{j,j,2} - \sum_{j=0}^{n_1-1} r_{j,j+1,2}$$

где последний индекс "2" указывает номер цепи.

После того как найдены R_{21} и R_{22} для второй цепи заносим эти векторы в соответствующие ячейки массива RR и определяем положение второй цепи. Теперь можно переходить к анализу третьей цепи и т.д.

В общем случае для i-ой цепи радиус-векторы R можно определить по формуле

$$R_{i1} = R_{N1,1} + \sum_{j=1}^{n1} r_{j,j,N1} + \sum_{j=0}^{n1-1} r_{j,j+1,N1}$$

$$R_{i2} = R_{N2,2} + \sum_{j=1}^{n2} r_{j,j,N2} + \sum_{j=0}^{n2-1} r_{j,j+1,N2}$$
(6.2)

где N1, N2 - номера базовых цепей, а n1, n2 - номера базовых звеньев в этих цепях. Номера N1, N2, n1,n2 хранятся в матрице STR /см. гл. I/.

Блок-схема алгоритма для определения положения механизма приведена на рис. 6.4. Алгоритм содержит один цикл с параметром і /блоки 2-16/, в котором осуществляется обход всех цепей механизма. Блок 2 определяет является ли начальное базовое звено стойкой и в зависимости от этого в следующих блоках 3,4 осуществляется расчет радиус-вектора R_{i1} . Блокй 5 и 13 слижат для разделения случаев разомкнутых и замкнутых цепей. В блоках 6-10 осуществляется расчет радиус-вектора R_{i2} и матрицы связи A_{i2} для конечного базового звена.

В результате выполнения алгоритма определяются векторы e_i, r_{ij} и матрицы перехода A_i и связи Λ , Λ_1 , Λ_2 . Эти выкторы и матрицы заносятся в массивы EE, RR, QQ, $\Lambda\Lambda$. Кроме того рассчитываются все радиус-векторы R_{ij} , которые также помещаются в RR.

В алгоритме используется матрица связи только конечного базового звена; матрица связи начального базового звена использовалась при "сборке". Предполагается, что механизм уже был предварительно собран.

В алгольном текствалгоритма /процедура ПОЛОЖЕНИЕ МЕХА-НИЗМА/ используются формальные параметры: N_{i} - число цепей механизмо; ε - точность итеррационного процесса для замкнутых цепей.

Глобальные идентификаторы E0, R0, Q0, EE, RR, QQ, $\Lambda\Lambda$, STR, KP. Придполагается, что массивы E0, R0, Q0 предваритель-

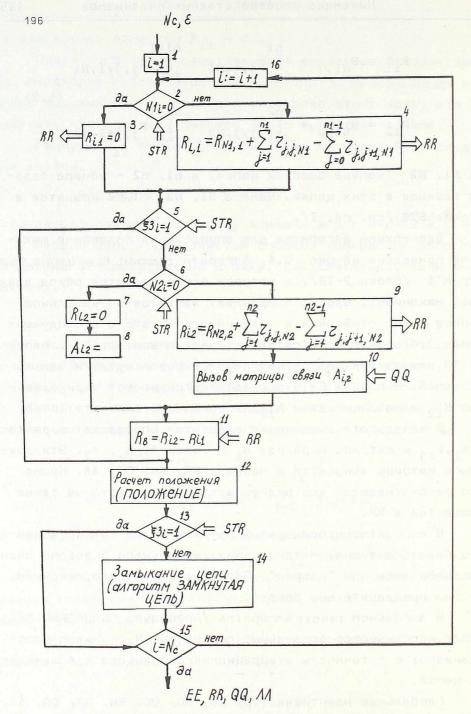


Рис. 6.4. Блок-схема алгоритма для определения положения механизма

```
но заполнены с помощью алгоритма СБОРКА МЕХАНИЗМА
     procedure ΠΟΛΟЖΕΗИΕ ΜΕΧΑΗИЗΜΑ (Nc, ε);
     value Nc; integer Nc; real ε;
begin array A2[1:3,1:3], hb[1:3];
     integer i,j,k,N1,N2,n1,n2;
     for i: = 1 step 1 until Nc do
     begin if STR[i,2] = 0 then begin
     for j: = 1,2,3 \text{ do } RR[i,0,j]: = 0;
     go to ml end;
     N1: = STR[i,2]; n1: = STR[i,3];
     for j := 1,2,3 do RR[i,0,1,j] := RR[N1,0,1,j] +
                       RR[N1,n1,1,j] - RR[N1,0,2,j];
      for k: = 1 step 1 until nl - 1 do
     for j := 1,2,3 \text{ do } RR[i,0,1,j] := RR[i,0,1,j] +
                       RR[N1,k,1,j] - RR[N1,k,2,j];
      if STR[i,1] = 1 then go to m3;
ml:
      N2: = STR[i,4]; n: = STR[i,6];
      if N2 = 0 then begin
      for j: = 1,2,3 do begin RR[i,n,2,j]: = 0;
      for k := 1,2,3 do A2[j,k] := 0;
      A2[j,j]:=1 end;
      go to m2 end условного оператара;
      n2: = STR[i,5];
      for j := 1,2,3 do
      RR[i,n,2,j] := RR[N2,0,1,j] + RR[N2,n2,1,j] -
                            RR[N2,0,2,j];
      for k: = 1 step 1 until n2 - 1 do
      for j: = 1,2,3 do
      RR[i,n,2,j] := RR[i,n,2,j] + RR[N2,k,1,j] -
                                   RR[N2,k,2,j];
      for j: = 1,2,3 do for k: = 1,2,3 do
      A2[j,k] := QQ[N2,n2,j,k];
      for j := 1,2,3 do hb[j]: = RR[i,n,2,j] - RR[i,0,1,j];
m2:
```

m3: <u>if</u> STR[i,1] = 1 <u>then</u> ПОЛОЖЕНИЕ (i) <u>else</u> ЗАМКНУТАЯ ЦЕПЬ (i, A2, hb, ε) <u>end</u> i; <u>end</u> процедуры;

Теперь переходим к случаю, когда маханизм нельзя разбить на автономные цепи. Рассмотрим его на примере механизма, показаного на рис. 6.1. Будем предполагать, что киниматический привод осуществляется в шарнирах А и Н после чего, как было показано выше в механизме нельзя выделить автономные цепи.

Выполним разбиение на цепи следующим образом. Первая жепь: ABCFGH; вторая цепь: ABCDE. На рис 6.3 эти цепи показаньи отдельно и обозначены относительные координаты.

Запишем уравнение связи для первой цепи в форме /2.29/

$$B_{3I} \begin{pmatrix} \Delta s_{1} \\ \Delta s_{2} \\ \Delta s_{3} \\ \Delta s_{4} \end{pmatrix} = B_{4I} \begin{pmatrix} \Delta z_{1} \\ \Delta z_{2} \end{pmatrix} + B_{1I} \Delta h_{I} + B_{2I} \Delta \gamma_{I} \quad (6.3)$$

Смысл всех обозначений в этой формуле описывается в гл. II, § 3. Римскими цифрами указан номер цепи. Поскольку для последнего звена эквевалентной цепи можно указать только три независимых перемещения /два поступательных и одно вращательное/, то матрица E_{31} в данном случае будет иметь размерность $\mathsf{3x4}$. Соответственно матрицы E_{41} , E_{11} и E_{21} будут $\mathsf{3x2;3x3;3x3}$.

Для второй цепи уравнение связи можно записать в виде

$$\mathsf{B}_{3II}\begin{pmatrix} \mathsf{\Delta s}_1 \\ \mathsf{\Delta s}_2 \\ \mathsf{\Delta s}_5 \\ \mathsf{\Delta s}_6 \end{pmatrix} = \mathsf{B}_{4II}\mathsf{\Delta z}_1 + \mathsf{B}_{1II}\mathsf{\Delta h}_{II} + \mathsf{B}_{2II}\mathsf{\Delta \delta}_{II} \quad (6.4)$$

где матрици $\mathsf{E}_{\mathsf{3II}}$ также имеет размерность $\mathsf{3x4}$, а матрицы

 $\mathsf{B}_{4\mathsf{II}}$, $\mathsf{B}_{1\mathsf{II}}$, $\mathsf{B}_{2\mathsf{II}}$ соответственно: 3x1; 3x3; 3x3.

Выражая /6.3/, /6.4/ определяют шесть ленейных уравнений относительно шести приращений дополнительных координат Δs_i /i = 1,2,...,6/.

Аналогично можно действовать и в случае других механизмов с неавтономными цепями. Мы не будем здесь более подробно останавливаться на этом вопросе, поскольку такие случаи, повидимому, достаточно редки. Для того чтобы механизм
нельзя было разбить на автономные цепи, он должен обладать
несколькими степенями свободы, иметь несколько замкнутых
цепей и кроме того иметь специальное распределение кинематических приводов по различным цепям.

В заключение этого параграфа остановимся на механизмах, у которых ни одна из цепей не связана со стойкой. Такие ме-ханизмы будем называть свободными. Выберем в свободном ме-ханизме некоторое звено и определим его положение относительно неподвижной системы координат. Это звено назовем основным. Для основного звена должны быть известны орты внутренней системы координат.

Далее, при сборки и расчете положения механизма, основное звено можно рассматривоть как "временную стойку" и отталкиваясь от него прономеровать все звенья цепи механизма. В дальнейшем сборка и расчет положений может быть выполнен таким же образом, как и для механизмов, связанных состойкой.

§2 Кинематика взаимосвязанных цепей

Рассмотрим случай, когда механизм можно разбить из автономные цепи. Тогда для расчета скоростей /скоростей центров мосс звеньев $\mathbf{v_{ij}}$ и угловых скоростей $\omega_{\mathbf{ij}}$ /каждой цепи можно воспользоваться алгоритмом СКОРОСТИ, полученным в гл. III, §1.

Для работы этого алгоритма наобходимо предворительно определить скорости базовых звеньев данной цепи ${\bf v}_{\rm oj}$, ${\bf \omega}_{\rm oj}$, ${\bf v}_{\rm nj}$, ${\bf \omega}_{\rm nj}$, ${\bf \omega}_$

Номера цепий, которые являются базовыми для данной j-ой цепи, можно прочитать во 2-ом и 4-ом столбцах матрицы STR /см. гл. I, §4/; выше эти номера были обозначены N1 и N2. Номера базовых звеньев в цепях N1 и N2 помещаются в 3 и 5 столбцах STR и они обозначены n1,n2 соответственно.

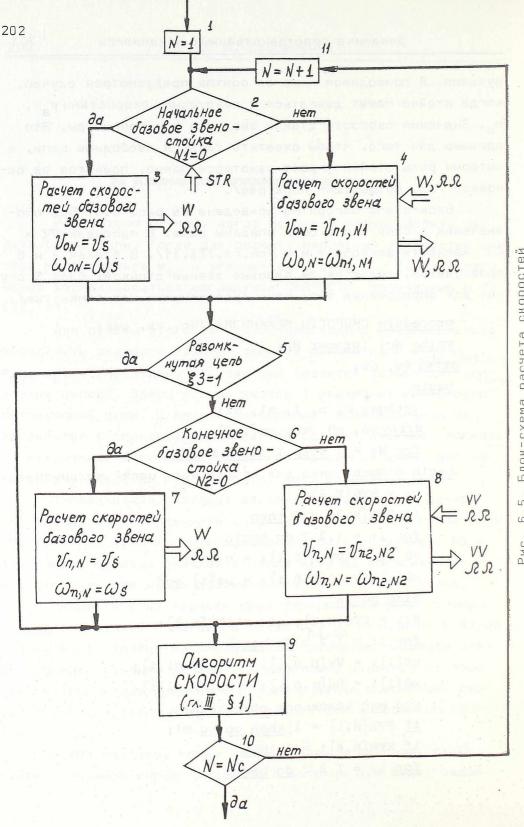
Скорости этих звеньев можно прочитать в матрице W. Например, для начального базового звена $v_{nl,Nl}$ находится в N1-ой строке и n1-столбце матрицы VV. Это значение скорости следует предварительно записать ј-ую строку нулевого столбца VV. Аналогично для конечного базового звена и угловных скоростей. После этого можно воспользоваться алгоритмом СКОРОСТИ.

В тех случаях, когда базовым звеном является стойка, соответствующий номер N1 или N2 в матрице STR будет задан

нулевым. В приводимом ниже алгоритме предусмотрен случай, когда стойка может двигаться с некоторыми скоростями $\mathbf{v_s}$, $\mathbf{\omega_s}$. Значения скорости стойки вводятся как параметры. Это сделано для того, чтобы охватить также и свободные цепи, в которых роль стойки играет некоторое звено, принятое за основное /см. предыдущий параграф/.

Блок-схема алгоритма приведена на рис. 6.5. Все обозначения в блок-схеме были описаны выше. В массивах VV и $\Omega\Omega$ хранястя векторы V и ω /см.гл.III,§1/. В блоках 2 и 6 проверяется, являются ли базовые звенья стойкой; блок 5 служит для определения типа цепи /разомкнутые или замкнутые/.

```
procedure CKOPOCTH MEXAHH3MA (Nc, vs, ws);
value Nc; integer Nc;
array vs, ws;
begin
     integer N, n, i, nl, Nl;
    <u>array</u> v0, ω0, vn, ωn[1:3];
     for N: = 1 step 1 until Nc do
begin comment цикл для обхода всех цепей механизма:
     n: = STR[N,6];
     if STR[N,2] = 0 then
     for i: = 1,2,3 do begin
     VO[i]: = VV[N,0,i]: = Vs[i];
     \omega 0[i] := \Omega\Omega[N,0,i] := \omega s[i] end
     else begin
     N1: = STR[N,2]; n1: = STR[n,3];
     for i: = 1,2,3 \text{ do begin}
     VO[i] := VV[N,0,i] := VV[N1,n1,i];
     \omega 0[i] := \Omega\Omega[N,0,i] := \Omega\Omega[N1,n1,i]
     end end условного оператора;
     if STR[N,1] = 1 then go to ml;
     \underline{if} STR[N,4]: = 0 then
     for i: = 1,2,3 do begin
```



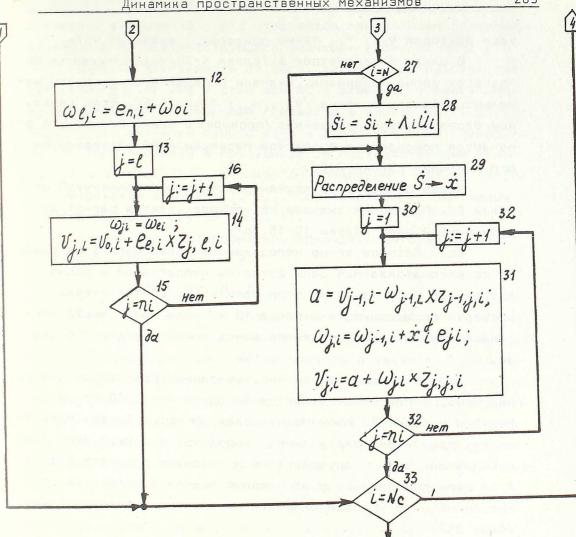
```
vn[i]: = VV[N,n,i]: = vs[i];
ωn[i]: = ΩΩ[N,n,i]: = ωs[i] end
else begin
N1: = STR[N,4]; n1: = STR[N,5];
for i: = 1,2,3 do begin
vn[i]: = VV[N,n,i]: = VV[N1,n1,i];
ωn[i]: = ΩΩ[N,n,i]: = ΩΩ[N1,n1,i]
end end;
ml: СНОРОСТИ (N, v0, ω0, vn, ωn)
end N;
end процедуры;
```

В дальнейшем, при анализе динамики часто придется рассчитывать виртуальные скорости. При этом относительная скорость будет задаваться только в одном из основных шарниров какой - либо одной цепи.

В этом случае можно воспользоваться алгоритмом СКО-РОСТИ МЕХАНИЗМА, однако это невыгодно, так как этот алгоритм рассчитан на общий случай, когда розможны относительные скорости во всех шарнирах.

Поэтому для расчета виртуальных скоростей целесообразно составить отдельный алгоритм, который назовем ВИРТУАЛ. На рис. 6.6 приведена блок-схема такого алгоритма. Особенность этого алгоритма по сравнению с СКОРОСТИ МЕХАНИЗМА состоит в том, что здесь выделяется целый ряд специальных случаев /неподвижные базовые звенья, виртуальная скорость задается в данной или другой цепи и.т.п./. Для каждого такого случая расчет ведется различным образом, благодаря чему удается сократить общее число вычислений.

В качестве формальных параметров выступают два целых числа: номер цепи N и номер шарнира 1, в которых задается виртуальная скорость. В блоке 2 соответствующей переменной присваивается значение единица. Все остальные х считаются равными нулю. Будем считать, что перед работой алгоритма



Блок-схема алгоритма ВИРТУАЛ

всем векторам v_{ij} , ω_{ij} также присвоены значения ноль.

В цикле с параметром і /блоки 4-36/осуществляется обход всех цепей механизма. Сначала определяются скорости начального базового звена v_{oi} , ω_{oi} ; блоки 4,5/. Если начальное базовое звено неподвижно/проверка в блоке 6/, то это отмечается присвоением булевской переменной v_{oi} 00 соответсвующего значения /блоки 7,8/.

Далее разделяются случаи розомкнутой и замкнутой цепи /блок 9/. Проследим сначала как осуществляется расчет для разомкнутой цепи /блоки 10-15,35/.

Если базовое звено неподвижно и номер цепи і не равен N, то рассматриваемая цепь остается неподвижной и расчет для данной цепи можно на этом закончить. Соответствующая проверка осущесвляестя в блоке 10 и, если имеет место описанный случай, осуществляется выход в конец цикла к блоку 34.

Допустим одно из условий, указанных предыдущем абзаце, нарушено, тогда осуществляется выход из блока 10 по ветви "нет" и в блоке 11 происходит новая проверка. Теперь проверяется, где задана виртуальная скоротсть, в данной цепи /i=N/или другой. Если в другой, то всем звеньям рассматриваемой і-ой цепи присваиваются одинаковые значения скоростей, равные линейной и угловой скорости начального базового звена /блок 35/.

Если виртуальная скорость задана в данной цепи, то в движении участвуют только звенья, номер которых /по номерации внутри цепи/ больше или равен 1. Они получают одинаковую угловую скорость и различные скорости центров масс, которые рассчитываются в цикле по ј/блоки 14-16/.

Теперь рассмотрим как осуществляется расчет для зам-кнутых цепей. В этом случае из блока 9 осуществляется выход по ветви "нет". После этого определяются скорости $v_{\rm ni}$, конечного базового звена и, если базовое звено неподвижно

/проверка в блоке 19/, это отмечается присвоением булевской переменной Bn соответствующего значения в блоках 20 и 21.

Далее в зависимости от того, являются ли базовые звенья неподвижными /B0 \equiv true или Bn \equiv true/ и, в какой цепи задана виртуальная скорость /проверка в блоках 22,23,25,27/, вычисляются значения скоростей \dot{s} в дополнительных шарнирах. После этого значения \dot{s} присваиваются соответствующим компонентам \dot{x} /блок 29/ и в цикле с параметром \dot{j} /блоки 31-33/ осуществляется расчет v_{ij} , ω_{ij} . Соответствующие векторные формулы выписаны внутри блока 31.

Алгольный текст процедуры приводить не будем и запишем только обращение к ней

ВИРТУАЛ /Nc, N, 1/

Теперь переходим к определению ускорений.

В главе III был получен алгоритм для расчета ускорений отдельной цепи. Им можно воспользоваться также в случае механизма с автономными цепями. Ситуация здесь аналогична той, которая имела место при определении скоростей. Для того чтобы воспользоваться алгоритмом УСКОРЕНИЯ нужно предварительно для каждой цепи определить линейные и угловые ускорения базовых звеньев. Поскольку предполагается, что механизм разбит на автономные цепи, то это всегда можно выполнить. Информацию о номерах базовых цепей и звеньев можно получить из матрицы STR ток же, как это делалось ранее.

Следовательно, блок-схема определения ускорений будет иметь совершенно такой же вид, как и блок-схема для скоростей, показанная на рис. 6.4. В ней нужно только заменить все обозначения v и w соответственно на w и є , а в блоке 9 вместо алгоритма СКОРОСТИ использовать алгоритм УСКОРЕНИЯ. Поэтому мы не будем приводить текст алгоритма /он аналогичен тексту процедуры СКОРОСТИ МЕХАНЕЗМА/ и ограничимся тем, что запишем только обрашение н соответствующей процедуре УСКОРЕНИЯ МЕХАНИЗМА.

УСКОРЕНИЯ МИХАНИЗМА (Nc, ws, εs, W, ε); эдесь Nc-число цепей механизма; ws и εs - ускорения стойки;

W, є - массивы, в которых помещаются рассчитанные значения линейных w и угловых є ускорений звеньев цепи. Массивы W и є имеют совершенно такое же строение как соответсвующие массивы VV и $\Omega\Omega$ для скоростей. Однако, в отличии от массивов VV и $\Omega\Omega$, которые были описаны как глобальные, массивы W и є ранее не встречались и потому введены в число формальных параметров.

При составлении дифференциальных уравнений движения механизма /см. ниже §4/ прийдится решать одну специфическую задачу кинематики. Она формулируется так: относительные скорости \dot{x}_{j}^{i} и ускорения \ddot{x}_{j}^{i} во всех шарнирах механизма равны нулю, кроме одного шарнира одной цепи, в котором $\ddot{x}_{N}^{1}=1$. Здесь N – номер цепи; 1 – номер шарнира. Необходимо определить линейные и угловые ускорения w_{ij} , ε_{ij} / для этаго случюя.

Как видно, такая постановка очень близка к той, которая имела место при разработке алгоритма ВИРТУАЛ. Там также задавалась единичная относительная скоротсть в одном из шарниров. Поэтому возмем за основу этот алгоритм /блок-схема на рис. 6.6/ и посмотрим какие необходимо внести в него изменения для данной задачи.

Прежде всего заменим все обозначения v, ω, ẋ, ἑ, ů, соответственно на w, ε, ẍ, š, ü. Далее при расчете дополнительных ускорений в в блоках 24 и 26 нужно воспользоваться формулами /см. гл. III, §2/:

$$\dot{s}_{i} = \Lambda_{2i} (\epsilon_{ni} - \epsilon_{oi} - \theta_{i});$$

$$\dot{s}_{i} = \Lambda_{li}(w_{ni} - w_{oi} - \eta_{i}).$$

Получившийся после этих изменений алгоритм назовем УСНО-РЕНИЯ 2 и обрашение к нему запишем в виде.

УСКОРЕНИЯ 2 (Nc, N, 1, WW, $\varepsilon\varepsilon$); где N и 1 номер цепи и звена, в которых задано $\ddot{x}_N^1=1$. Полный текст алгоритма приводить не будем, поскольку он легко может быт составлен по блок-схеме ВИРТУАЛ с помощью указанных изменений.

§3 Статика взаимосвязанных цепей

Статика отдельных цепей рассматривалась в гл. IV. При этом были разработаны два алгоритма СТАТИКА РАЗ ЦЕПИ и СТАТИКА ЗАМ ЦЕПИ, которые поэволяют определить реакции в кинематических парах и реакции приводов соответственно для разомкнутых и замкнутых цепей.

Переходя к анализу статики взаимосвязанных цепей нужно учитывать, что сателлиты воздействуют на свои базовые цепи определенными силами и моментами. Отсюда следует, что прежде чем будет решена задача статики для некоторой цепи, необходимо решить задачи статики для всех ее сателлитов.

Вернемся к § 1 этой главы, где описан способ номерации цепей. Согласно этому способу i-я кинематическая цепь всегда базируется либо на цепи, номер которой меньше i, либо на стойку. Отсюда следует, что если некоторая цепь имеет сателлиты, то их номера будут больше, чем номер этой цепи.

Тогда можно решать задачу статики, начиная с последней цепи /то-есть цепи с наибольшим номером/ и далее последовательно двигаться к первой цепи. При этом, очевидно, приступая к анализу любой цепи мы будем всегда знать необходимые данные о ее сателлитах.

В процедурах СТАТИНА РАЗ ЦЕПИ и СТАТИНА ЗАМ ЦЕПИ предусмотрено определение воздейсвий на базовые звенья. Для этого в процедуры введены формальные параметры $R_{\tilde{A}}$, Q_{A} , R_{B} , Q_{B} , где R_{A} , Q_{A} - сила и момент для начального базового звена; R_{B} , Q_{B} - для конечного. Силы привидены к полюсам /центрам масс/ базовых звеньев.

Следовательно, определив R_{A} , Q_{A} для некоторого сате-

ллита мажно в матрице STR прочитать номер состветствующей базовой цепи N1 и базового звена n1 и затем прибавить R_A , $\mathbb{Q}_{\underline{\lambda}}$ к внешним силам и моментам, действующим на это звено.

$$G_{n1,N1} := G_{n1,N1} + R_A ;$$

$$M_{n1,N1} := M_{n1,N1} + Q_A .$$
(6.5)

Аналогично для конечного базового звена

$$G_{n2,N2}$$
: = $G_{n2,N2}$ + R_B ;
 $M_{n2,N2}$: = $M_{n2,N2}$ + Q_B .

После того, как такая операция выполнена для всех сателлитов i-ой цепи можно переходить к анализу статики самой i-ой ципи. Эта цепь в свою очередь может являться сателлитом для цепей с меньшим номером и т.д.

Таким образом, алгоритм расчета статики механизма с автономными цепями может быть представлен следующей блок-схемой /рис. 6.7/. Текст процедуры /назовем ее СТАТИКА МЕ-ХАНИЗМА/ легко может быть составлен по этой блок-схеме и поэтому мы его здесь приводить не будим.

Теперь рассмотрим механизм с упругими связями /см.гл. IV, §3/. Под действием внешней нагрузки такой механизм из-меняет свое положение за счет деформации упругих связей.

Предположим, как и раньше, что кинематические приводы не деформируются /эквивалентны жесткой связи между соответствующими звеньями/, а в дополнительных шарнирах привод отсутствует. Тогда упругие связи наложены только в У-шарнирах и их деформация связана с развиваемой реакцией соотношением /см. /4.20//

$$\Delta y_{k}^{i} = y_{k}^{i} - y_{ok}^{i} = -c_{ik}p_{k}^{i}$$
 (6.7)

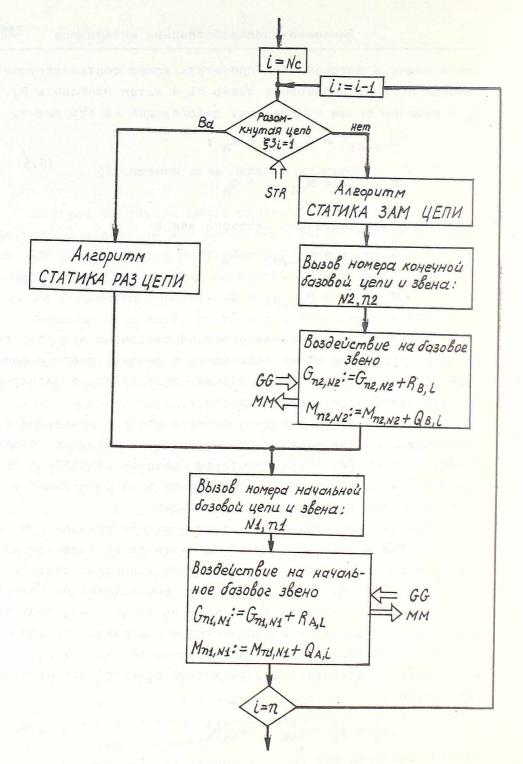


Рис. 6.7. Блок-схема алгоритма анализа статика механизма

тельно должен быть соответствующим образом сформирован вектор $\lambda 1$ /подробнее см.гл.IV,§ 3/. Поскольку в данной задаче подлежат определению реакции только в У-шарнирах, то для этих шарниров должно быть задано $\lambda 1 = 1$, а для остальных $\lambda 1 = 0$;

- 3. Определяются деформации согласно /6.7/;
- 4. Определяются максимальное значение деформаций $b = \max \ (\left[\Delta y_j^i \right]) \\ \text{(ij)} \\ \text{где j-номер цепи,i номер звена.}$
- 5. Если $b \le \epsilon$, сде ϵ заданное число, определяющее точность приближения, то процесс итерраций считаем законченным. В противном случае присваиваем y_j^i новые значения

$$y_{j}^{i} := y_{0j}^{i} + k\Delta y_{j}^{i} ; (0 \le k \le 1)$$

и переходим к nl.

Наличие постоянного коэффициента k отличает последнюю формулу от той, которая использовалась в алгоритме ДЕФОРМА-ЦИЯ. Этот коэффициент введен для того чтобы обеспечит монотонную сходимость процесса приближения к положению равновесия. Оптимальное значение подбирается опытным путем в процессе расчетов.

§4 Динамика и моделирование механизмов с взаимосвязанными цепяами

Расчет сил и моментов сил инерции механизма со взаимосвязанными цепями можно осуществить используя результаты полученные для осдельной цепи в гл.V. Поскольку такой расчет прийдется неоднократно проводить при составлении дифференциальных уравнений, составим соответствующий алгоритм.

Будем предполагать, что уже осуществлен анализ кине- матики и определены угловые ускорения ϵ_{ij} всех звеньев и ускорения их центров масс \mathbf{w}_{ij} /эдесь первый индекс i-номер звена, второй j-номер цепи/. Эти ускорения помещены соответстенно в массивы $\epsilon \epsilon$ и WW.

Теперь последовательно применяя алгоритм ИНЕРЦИЯ /см. гл. V/ ко всем цепям механизма определим силы инерции \mathbf{f}_{ij} и моменты сил инерции \mathbf{M}_{ij} , которые также поместим в массивы WW и $\epsilon\epsilon$ на место \mathbf{w}_{ij} и ϵ_{ij} соответственно.

Алгоритм оформлен как процедура с формальным параметром Nc , указывающим число цепей. Массивы WW и $\varepsilon\varepsilon$ - глобальные

```
procedure ИНЕРЦИЯ МЕХАНИЗМА (Nc);

value Nc; integer Nc;

begin

integer i, j, k, n;

j: = STR[1,6];

for i: = 1 step 1 until Nc do

if j < STR[i,6] then j: = STR[i,6];

begin
```

где k – номер цепи; с і тестокость связи і-ой кинематической пары в k-ой цепи.

С другой стороны реакции приводов сами являются функциями положения механизма и, следовательно, координат у. Для кажного значения у реакции р можно определить с помощью алгоритма СТАТИКА МЕХАНИЗМА, описанного выше. Таким образом этот алгоритм и уравнение /6.7/ определяют у как неявную функцию.

В гл. IV был построен итеррационный процесс для определения деформаций связей в отдельно взятой цепи /алгоритм ДЕФОРМАЦИЯ/. Однако, непосредсвенно воспользоваться им для анализа взаимосвязанных цепей затруднительно по следующей причине.

Указанный алгоритм с помощью последовательных итерраций каждую цепь приводит в положение статического равновесия. Но в механизме цепи взаимосвязаны и при перемещении одной из них изменяется положение других. При этом может нарушиться положение равновесия тех цепей, которые были рассмотрены и "уравновешены" ранее. Следовательно, при использовании алгоритма ДЕФОРМАЦИЯ пришлось бы для каждой цепи многократно повторять итеррационный процесс, что явно нецелесообразно.

Поэтому для взаимосвязанных цепей лучше построить процесс заново, с тем чтобы "деформировать " все цепи совместно и так же совместно осуществлять последовательное приближение к положению ровновесия. За основу здесь можно взять алгоритм СТАТИКА МЕХАНИЗМА.

Тогда, процесс определения деформаций упругих связей будет состоять из следующих этапов

- 1. Для данных значений относительных углов ${f x}_{\sf O}$ определяется положение механизма /то-есть векторы ${f e}$, ${f r}/$ с помощью алгоритма ПОЛОЖЕНИЕ МЕХАНИЗМА;
- 2. Определяются реакции приводов, соответствующие \mathbf{x}_{O} , с помощью алгоритма СТАТИКА МЕХАНИЗМА. Предвори-

```
array W,ε[1:j, 1:1, 1:3];
for i: = 1 step 1 until Nc do

begin n: = STR[i,6];
for j: = 1 step 1 until n do

for k: = 1,2,3 do

begin W[j,1,k]: = WW[i,j,k];
ε[j,1,k]: = εε[i,j,k]

end;

ИНЕРЦИЯ (i, W, ε, true);
for j: = 1 step 1 until n do

for k: = 1,2,3 do

begin WW[i,j,k]: = W[j,1,k]
εε[i,j,k]: = ε[j,1,k]
end i;
end end end процедуры;
```

Для составления дифференциальных уравнений движения механизма с автономными цепями можно воспользовать алгоритмом ДИФ УР, полученным в гл. V. Этот алгоритм составляет уравнения в форме /5.50/, куда входят ускорения /линейные и угловые/ базовых звеньев. Ускорения базовых звеньев в свою очередь зависят от вторых производных ў, что видно из выражений /3.11/. Следовательно, ускорения базовых звеньев в форме /3.11/ следует подставить в дифференциальные уравнения соответствующих цепей /5.50/ после чего сгруппировав члены при ў получим дифференциальные уравнения движения всего метханизма.

Такое способ рационален в случае, когда механизм состоит из одной или двух цепей. Однако, если механизм содержит много цепей, то при реализации такого способа потребуется слишком большой объем памяти для хранения всех промежуточных результатов при подстановках и группировках.

Поэтому в настоящем параграфе будет описан другой спо-

соб составления дифференциальных уравнений, который, хотя и связан с несколько более продолжительными вычислениями, однако требует существенно меньший обьем помяти.

Прежде всего введем некоторые обозначения. Прономеруем подряд все звенья и шарниры механизма, придерживаясь следующего порядка: сначала все звенья и шарниры 1-ой цепи в той це последовательности, как они были прономерованы ранее, затем номеруем все звенья и шарниры второй цепи, в том порядке, как это делалось ранее и т.д. Базовые звенья не будем включють в эту немерацию /в противном случае эти звенья были бы пронамерованы по несколько раз/. Общее число подвижных звеньев механизма обозначим s1, шарниров - s2.

Объединим все координаты y_j^i в один многомерный вектор - столбец, который обозначим σ

$$\sigma \equiv \begin{pmatrix} \sigma^{1} \\ \sigma^{2} \\ \vdots \\ \sigma^{s3} \end{pmatrix} \equiv \begin{pmatrix} y_{1} \\ y_{2} \\ \vdots \\ y_{NC} \end{pmatrix}$$

$$(6.8)$$

где все уј - векторы-столбцы; нижний индекс при у указывает номер цепи; №С- число цепей;

$$s3 = \int_{j=1}^{Nc} n_{yj}$$
 (6.9)

Аналогично поступим с координатами тех шарниров, которые имеют силовой привод $\bar{\mathbf{x}}$ и с реакциями этих приводов $\bar{\mathbf{p}}$. Составим из них два многомерных вектора, которые обозначим $\bar{\mathbf{q}}$ и $\bar{\mathbf{g}}$

$$q \equiv \begin{pmatrix} q^{1} \\ q^{2} \\ \vdots \\ q^{s} \end{pmatrix} \equiv \begin{pmatrix} \overline{x}_{1} \\ \overline{x}_{2} \\ \vdots \\ \overline{x}_{N_{C}} \end{pmatrix}; \quad q \equiv \begin{pmatrix} g^{1} \\ g^{2} \\ \vdots \\ g^{s} 4 \end{pmatrix} \equiv \begin{pmatrix} \overline{p}_{1} \\ \overline{p}_{2} \\ \vdots \\ \overline{p}_{N_{C}} \end{pmatrix}; \quad (6.10)$$

где $\mathbf{s4}$ - число шарниров с силовым приводом,и нижние индексы при \mathbf{x} , \mathbf{p} и указывают номер цепи.

Теперь можно переходить к составлению дифференциальных уравнений. Для этого воспользуемся кинетостатическим подходом.

Предположим, что механизму сообщена вертуальная скорость $\dot{\sigma}^j=1$ в одном j-ом шарнире. В результате звенья механизма приобретут некоторые улговые и линейные скорости, которые обозначим v_{ij} , ω_{ij} , а в шарнирах появятся относительные скорости \dot{x}^i_j . Здесь i-номер звена, j-номер основного шарнира, в котором задана виртуальная скорость $\dot{\sigma}^j$.

Согласно принципу Даламбера суммарная мощность внешних сил, сил инерции и реакций приводов, подсчитанная для заданной виртуальной скорости, должна быть равна нулю

$$\sum_{i=1}^{s1} [(F_i + G_i) \cdot v_{ij} + (M_i + Q_i) \cdot \omega_{ij}] + \sum_{i=1}^{s4} g^i \dot{q}_j^i = 0 \quad (6.11)$$

где **F**i, Q_i – главный вектор и главный момент сил инерции i-го звена;

 ${\tt G_i}, {\tt M_i}$ - внешняя сила и момент, действующие на ${\tt i-e}$ звено. Силы и моменты считаются приведенными к центру масс звена.

Отметим, что уравнения /6.11/ линейны относительно сил и моментов сил инерции.

Нак было показано в гл. V силы и моменты сил инерции можно представить в виде линейной формы от ускорений ö

$$F_{i} = \sum_{k=1}^{s1} a_{ik} \ddot{\sigma}^{k} + \alpha_{i}$$

$$Q_{i} = \sum_{k=1}^{s1} b_{ik} \ddot{\sigma}^{k} + \beta_{i}$$

$$(6.12)$$

где векторные коэффициенты a_{ik} и b_{ik} суть функции от координат x, а векторы α_i , β_i зависят от координат x, скоростей \dot{x} и ускорений в z-шарнирах.

Мы не будем пытаться здесь получит формулы выражающие a_{ik} , b_{ik} , α_{i} , β_{i} через свои аргументы. Вместо этого воспользуемся линейностью /6.12/ и будем задавать такие значения $\ddot{\sigma}$, \dot{x} и \ddot{z} , при которых все члены в правых частях /6.12/ кроме одного обращаются в ноль.

Например, положим ў равным нулю. Тогда получим

$$F_{i} = \alpha_{i} ; \qquad Q_{i} = \beta_{i}$$
 (6.13)

С помощью алгоритмов УСКОРЕНИЯ МЕХАНИЗМА и ИНЕРЦИЯ МЕХАНИЗМА /см. гл. V, §1/ можно расчитать силы и моменты инерции при любых значениях $\ddot{\mathbf{y}}$, $\ddot{\mathbf{z}}$, $\dot{\mathbf{x}}$ и значит при тех, которые сейчас задаем. Следовательно, с помощью этих алгоритмов рассчитать $\mathbf{F_i}$, $\mathbf{Q_i}$ и по /6.13/ найти $\mathbf{\alpha_i}$, $\mathbf{\beta_i}$. Далее положим: $\ddot{\mathbf{z}} = \mathbf{0}$, $\dot{\mathbf{x}} = \mathbf{0}$

$$\vec{\sigma}^{k} = \begin{cases} 1 & \text{при } k = r; \\ 0 & \text{при } k \neq r \end{cases}$$

$$k = 1, 2, \dots, NN$$

$$(6.14)$$

то-есть задаем единичное относительное ускорение в одном шарнире.

Легко видеть, /см. гл. V/ что при $\ddot{\mathbf{z}}$ = 0, $\dot{\mathbf{x}}$ = 0 коэффициенты $\alpha_{\dot{\mathbf{i}}}$, $\beta_{\dot{\mathbf{i}}}$ также обращаются в ноль. Тогда в правых частях /6.12/ остается только по одном члену и мы получим

$$F_{i} = a_{ir}$$

$$Q_{i} = b_{ir}$$
(6.15)

Силы $\mathbf{F_i}$ и моменты $\mathbf{Q_i}$ для этого случая можно расчитать с помощью алгоритмов УСКОРЕНИЕ 2 и ИНЕРЦИЯ МЕХАНИЗМА:

Таким способом можно определить все a_{ik} , b_{ik} , α_{i} , β_{i} .

Теперь подставим выражения /6.12/ в уравнения /6.11/. Группируя все чвены зависящие от \ddot{y} и изменив порядок суммирования приведем дифференциальные уравнения движения механизма к следующей матричной форме

$$D\ddot{\sigma} = Hg + h \tag{6.16}$$

где D - квадратная матрица размерности s3 x s3 с копмон<mark>ентами</mark>

$$D^{ij} = \sum_{k=1}^{Nz} (a_{kj} \cdot v_{ki} + b_{kj} \cdot \omega_{ki})$$
 (6.17)

.Н - матрица размерности s3 x s4 с компонентами

$$H^{ij} = -\dot{q}_i^j \tag{6.18}$$

h - вектор столбец с NN компонентами

$$h^{i} = -\sum_{k=1}^{s1} \left[(\alpha_{k} + G_{k}) \cdot v_{ki} + (\beta_{k} + M_{k}) \cdot \omega_{ki} \right]$$
 (6.19)

В вектор h вошли все члены зависящие от скоростей x, ускорений в шарнирах с кинематическим приводом \ddot{z} и внешней нагрузки G, M.

Векторы a_{kj} , b_{kj} , α_k , β_k в этих формулах определяются по спотобу описанному выше.

Таким образом расчет компоненты $D^{\mathbf{i}\,\mathbf{j}}$ матрицы D включает в себя следующие этапы:

- 1. задаем $\ddot{\sigma}^{\dot{j}} = 1$; $(\ddot{\sigma}^{\dot{l}} = 0 \text{ при } 1 \neq j; \quad l = 1, 2, ..., NN);$
- 2. определяем для такого случая силы и моменты сил инерции $\mathbf{F}_{\mathbf{k}}$, $\mathbf{Q}_{\mathbf{k}}$; где \mathbf{k} номер звена;
- 3. задаем виртуальную скорость $\mathring{\sigma}^{\hat{1}} = 1$ $(\mathring{\sigma}^{\hat{1}} = 0 \quad 1 \neq i; \quad 1 = 1, 2, \dots, NN)$
- 4. о̀пределяем скорости v_{ki} , ω_{ki} , вызванные заданной относительной скоростью $\mathring{\sigma}^{\dot{\text{l}}}$;

5. определяем суммарную мощность внешних сил и сил инерции $^{\rm Ha\ V}{
m ki}^{\rm M}$ $^{\rm W}{
m ki}^{\rm C}$

Результат всех этих операций будет определять значение компоненты $\mathtt{D}^{\mathtt{i}\mathtt{j}}.$

Для составления дифференциальных уравнений по этому способу понадобится еще одна процедура для расчета мощности. Процедура "берет" векторы сил и моментов из массивов GG и MM и скалярно умножает на векторы линейных и улговых скоростей, которые хранятся в массивах VV и ΩΩ соответственно.

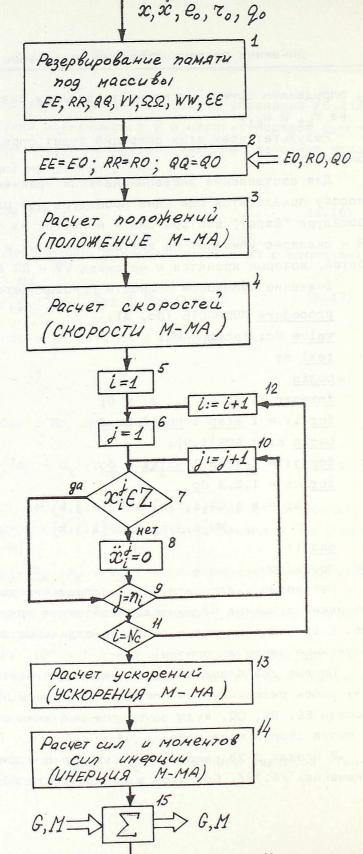
Значение суммарной мощности присваивается параметру а.

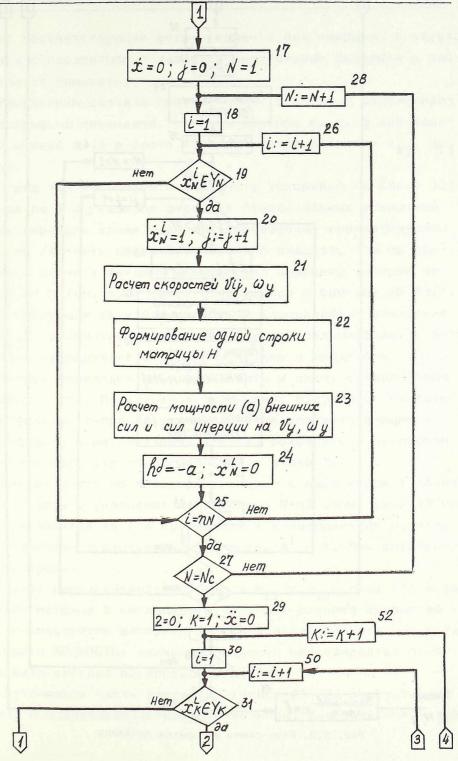
end процедуры;

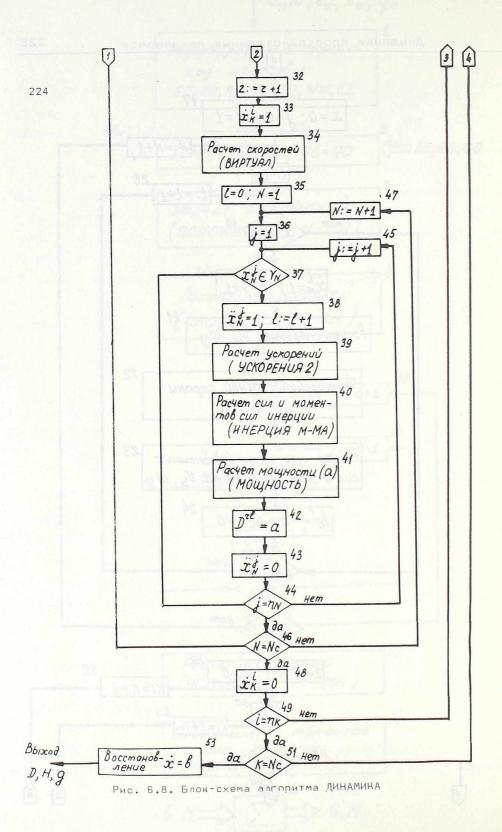
Переходим к алгоритму для составления дифференциальных уравнений движения механизма. Блок-схема предстявлена на рис. 6.8. Она включает в себя как отдельные блоки почти все полученные ранее алгоритмы.

Первые два блока осуществляют подготовительные операции: эдесь резервируется оперативная память и заполняютоя массивы EE, RR, QQ, куда заносятся значения векторов е, r, q после сборки механизма.

В блоках 3-28 рассчитывают вектор h в дифференциальных уравнениях /6.16/. При этом в первой части /блоки 3-15/ опре-







дляются соответствующие силы и моменты сил инерции, а во второй цикл с параметром N задаются виртуальные скорости и расситывается мощность.

Рассмотрим сначала первую часть. В блоке 3 рассчитивается положение механизма, то-есть векторы е, \mathbf{r} , \mathbf{q} для заданных значений \mathbf{x} , а в блоке 4 определяются скорости $\mathbf{v}_{\mathbf{ij}}$, $\mathbf{\omega}_{\mathbf{ij}}$ звеньев.

Перед тем как перейти к расчету ускорений /в блоке 13/ в циклах по і и јгасятся значения относительных ускорений во всех шарнирах кроме шарниров, оснашенных кинематичесими приводами /то-есть кроме $\ddot{\mathbf{z}}$ /. Из этого следует, что мы приступаем к расчету тех составляющих сил инерции, которые не зависят от $\ddot{\sigma}$ /см. выше текст относящийся к формуле /6.13//.

В следующих двух блоках 13, 14 определяются ускорения w_{ij} , ϵ_{ij} и рассчитываются силы и моменты сил инерции. В блоке 15 они суммируются с внешними силами и моментами.

Теперь переходим ко второй части и циклу с параметром N /блоки 17-28/. Поскольку в этом цикле будут рассчитываться виртуальные скорости, и определятся соответствующие х, то "действительные" значения х нужно сохранить в некотором другом моссиве: это осуществляется в блоке 16.

Внутри цикла по N содержится цикл с параметром і /блоки 19-26/. Здесь і указывает номер звена N-ой цепи. Блок 19 проверяет относится ли і-я координата к σ -координатам и, если да, то сообщается виртуальная скорость $\dot{\mathbf{x}}^{\dot{\mathbf{i}}} = \mathbf{1}$. Все остальные $\dot{\mathbf{x}}$ равны нулю.

Далее рассчитываются скорости v_{1r} и ω_{1r} /блок 21/ и формируется матрица H согласно /6.18/. Для расчета скоростей здесь используется алгоритм ВИРТУАЛ, а для расчета мощности – алгоритм МОЩНОСТЬ; значение мощности присваивается j-ой номпоненте вектора h, согласно /6.19/.

Оставшаяся часть алгоритма /блоки 29-52/ рассчитывает матрицу D в дифференциальных уравнениях. Она содержит четыре

цикла, вложенные друг в друга. Первые два с пораметрами ${\bf k}$ и ${\bf i}$ служат для того, чтобы задавать виртуальные скорости во всех ${\bf y}$ - шарнирах и рассчитывать соответствующие значения скоростей ${\bf v}$ и ${\bf \omega}$.

Для каждой виртуальной скорости последовательно задаются единичные ускорения во всех у - шарнирах и для этих ускорений рассчитываются силы и моменты сил инерции и их мощность. Указанные операции осуществляются в циклах с параметрами N и j /блоки 36-47; N - номер цепи, j - номер шарнира, в котором задается ускорение $\ddot{y} = 1/$. Таким образом матрица N рассчитывается построчно, согласно N

В последнем блоке алгоритма восстанавливаются значения скоростей х, которые были запомнены в блоке 16.

Текст алгоритма составления уравнений /процедура ДИНА-МИКА/ приводится в конце этого параграфа, а сейчас остановимся кратко на моделировании механизмов с взаимосвязанными цепями. Оно может быть выполнено аналогично тому, как моделировался механизм с одной цепью в гл. V, §4. Остается в силе также все сказанное в этом параграфе о возможностях такого моделирования.

Блок-схема модели также будет иметь вид аналогичный, показанному на рис. 5.5. Однако, теперь в блоке составления дифференциальных уравнений должен использоваться только что описанный алгоритм ДИНАМИКА. Уравнения составяются в форме /6.16/ и, следовательно блоки 16-21 становятся излишними. Формулы в блоках 23, 27 теперь следует записать как

$$\ddot{y} = D^{-1} (H\bar{p} - h);$$

 $\bar{p} = H^{-1} (D\ddot{y} - h).$

Ниже приводится текст процедуры ДИНАМИКА. Формальные параметры D, H, h имеют тот же смысл, что и соответствующие обозначения в /6.16/; Nc - число цепей; vs, ws, ws, єs - скорости и ускорению стойки.

```
procedure ΔΗΗΑΜΗΚΑ (Nc,D,H,h,ε,vs,ωs,ws,εs);
    value Nc; integer Nc;
    array D, H, h, vs, ωs, ws, εs; real ε;
    begin integer i, j, k, n, N, l, q, r;
          real a;
          n: = STR[1,6];
          for i: = 1 step 1 until Nc do
          if n < STR[j,6] then n := STR[j,6]
    begin array VV, \Omega\Omega, WW, \varepsilon\varepsilon, EE[1:Nc,1:n,1:3],
          b[1:Nc,1:n], RR[1:Nc,1:n,1:2,1:3],
           QQ[1:Nc,1:n,1:3,1:3];
    for i: = 1 step 1 until Nc do
    for j: = 1 step 1 until n do
    for k := 1,2,3 do
    begin EE[i,j,k]: = E0[i,j,k];
           RR[i,j,1,k] := R0[i,j,1,k];
           RR[i,j,2,k]: = R0[i,j,2,k]
    end;
    ПОЛОЖЕНИЕ МАХАНИЗМА (Nc, \epsilon);
    CHOPOCTЬ MEXAHИЗМА (Nc, vs, ws);
    for i: = 1 step 1 until Nc do
    begin n: = STR[i,6];
           for j: = 1 step 1 until n do
           if KP[i,j,2] \neq 1 then X[i,j,3] := 0
     end;
     ΥCKOPEHUA MEXAHU3MA (NC, WS, εS, WW, εε);
     ИНЕРЦИЯ МЕХАНИЗМА (Nc);
<u>comment</u> прибавляем силы инерции к внешним силам и запомннаем
     значения скоростей;
     for i: = 1 step 1 until Nc do
     begin n: = STR[i,6];
           for j: = 1 step 1 until n do
           begin b[i,j] := X[i,j,2];
                  X[i,j,2] := 0;
```

```
for k: = 1,2,3 \text{ do}
           begin
           GG[i,j,k]: = GG[i,j,k] + WW[i,j,k];
           MM[i,j,k] := MM[i,j,k] + \varepsilon\varepsilon[i,j,k]
     end end end i;
comment приступаем к расчету матрицы Н и вектора h;
     j := 0;
     for N: = 1 step 1 until Nc do
     begin n: = STR[N,6];
            for i: = 1 step 1 until n do
            begin
            \underline{if} KP[N,i,3] \neq 1 V KP[N,i,2] = 0 \underline{then}
                 go to ml;
            X[N,i,2] := 1; j:= j+1;
            ВИРТУАЛ (Nc, N, i);
           формируем і-ю строку матрицы Н;
comment
     1:=0; for q: = 1 step 1 until Nc do
            begin r: = STR[q,6];
            for k: = 1 step 1 until r do
            if KP[q,k,2] = 2 then
            begin 1: = 1+1;
            H[i,1] := X[q,k,2]
     end end;
comment расчитываем j-ю компоненту вектора h;
            MOЩНОСТЬ (Nc, a);
            h[j]: = -a; X[N,i,2]: = 0; X[N,i,3]: = 0;
ml
      end i;
      end N;
comment приступаем к расчету матрицы D;
      r: = 0;
      for k: = 1 step 1 until Nc do
      begin n: = STR[k,6];
            for i: = 1 step 1 until n do
```

```
begin
     if KP[k,i,3] \neq 1 \land KP[k,i,2] then go to m2;
r = r+1; X[k,i,2] : = 1;
ВИРТУАЛ (Nc, k, i);
1: = 0; for N: = 1 step 1 until Nc do
      begin q: = STR[N,6];
      for j: = 1 step 1 until q do
      begin X[N,j,3]: = 1; 1: = 1+1;
      УСКОРЕНИЯ 2 (Nc, N, j, WW, εε);
      инерция МЕХАНИЗМА (Nc);
      MOWHOCT (Nc, a);
      D[r,1]: = a; X[N,j,3]: = 0;
end N;
end j; X[k,i,2] := 0;
end i;
end k;
     for i: = 1 step 1 until Nc do
      begin n: = STR[i,6];
      for j: = 1 step 1 until n do
      X[i,j,2] := b[i,j]
      end
end end процедуры;
```

приложение і

Библиотека алгоритмов

Ниже для каждого из разработанных в книге алгоритмов записывается его обозначение с указанием формальных парметров, приводится краткая характеристика выполняемых операций и дается ссылка на главу и параграф, в которых содержится описание алгоритма.

- 1. СБОРКА ПАРы /N, i, rl, e , r2, $\Lambda/$ осуществяет "сборку" i-ой кинематической пары и определяет компоненты векторов $r_{i-1,i}$, r_{ii} , и e , а также матрицу перехода A_i после сборки; см. гл. II, §1.
- 2. ПОВОРОТ /r, e, $\mathbf{x}/$ осуществляет поворот вектора \mathbf{r} вокруг е на угол \mathbf{x} ; см. гл. II, $\S 2$.
- 3. ПОЛОЖЕНИЕ /N/ осуществляет расчет положения всех звеньев /векторы r/ и осей шарниров /e/ для N-ой цепи; см. гл. II, \$2.
- 4. МАТРИЦы СВЯЗИ $/{
 m N}/$ осуществляет расчет матриц связи Λ , Λ_1, Λ_2 для замкнутой цепи с номером ${
 m N};$ см. гл. ${
 m II},$ §3.
- ΚΟΜΠ /C, ξ, n, m, В/ осуществляет перестановку строк и столбцов матрицы С в соответствии со значениями компонент вектора ξ; см. гл. II, §3.
- 6. МАСКА /a, b, q, λ , c/ осуществляет опицеальную обработку массивов; см. гл. II, §3.
- 7. ЗАМКНУТАЯ ЦЕПЬ /N, A2, hb, ϵ / осуществляет расчет поло-

жения звеньев и осей шарниров для замкнутых цепей; см. гл. II, §4.

- 8. СКОРОСТИ /N, v0, $\omega 0$, vn, $\omega n/$ осуществляет расчет угловых скоростей и скоростей ценртов масс разомкнутых и замкнутых Чепей; см. гл. III, §1.
- 9. УСНОРЕНИЯ /N, w0, ϵ 0, wn, ϵ n, W, $\bar{\epsilon}$, BL// осуществляет расчет угловых ускорений и ускорений разомкнутых и зам-кнутых цепей; см. гл. III, $\S 2$.
- 10. СТАТИКА РАЗ ЦЕПИ /N, RQ, RA, QA/ осуществляет расчет реакций в кинематических парах и усилий и моментов приводов для разомкнутых жепей; см. гл. IV , §1.
- 11. СТАСИКА ЗАМ ЦЕПИ /N, RQ, RA, QA, RB, QB/ осуществляет расчет реакций в кенематических парах и усилий и моментов приводов для замкнутых Цепей; см. гл. IV, §2.
- 12. ДЕФОРМАЦИЯ /N, A2, hb, c, є/ для механизма с упругими связями расчитывается изменение положения под действием нагрузки; см. гл. IV, §3.
- 13. ИНЕРЦИЯ /N, W, ε, BL/ определяются силы и моменты силинерции звеньев N-ой цепи, приведенные к центрам масс; см. гл. ∇, §1.
- 14. ДИФ УР /N , A1, A2, v0, ω 0, vn, ω n, w0, ϵ 0, wn, ϵ n, hb, ϵ , A, B, C, D, E, g/ осуществляет составление дифференциальных уравнений движения N-ой цепи; см. гл. V, §3.

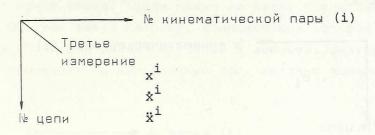
Все последующие алгоритмы применимы к механизмам с автономными кинематическими цепяпи. Цепи могут быть как разом-кнутые, так и замкнутые. Во всех алгоритмах Nc-число цепей.

15. СБОРКА МЕХАНИЗМА /Nc/ - осуществляет "сборку" механизма; см. гл. VI, \S 1.

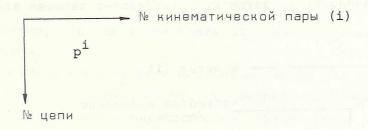
- 16. ПОЛОЖЕНИЕ МАХАНИЗМА /Nc, є/ осуществляет расчет положения звеньев и осей шарниров механизма; см. гл. VI, §1.
- 17. СКОРОСТИ МЕХАНИЗМА /Nc, vs, ω s/ определяет угловые скорости и скорости центров масс всех звеньев; см. гл. VI, §2.
- 18. ВИРТУАЛ /Nc, N, 1/ осуществляет расчет угловых и линейных скоростей в специальном случае, когда задана относительная скорость только в одном шарнире; см. гл. VI, § 2.
- 19. УСКОРЕНИЯ МЕХАНИЗМА /Nc,ws, ε s, WW, $\varepsilon\varepsilon$ / осуществляет расчет угловых ускорений и ускорений центров масс всех звеньев; см. гл. VI, $\S 2$.
- 20. УСКОРЕНИЯ 2 /Nc, N, 1, WW, εє/ осуществляет расчет линейных и угловых ускорений в специальном случае, когда задано относительное ускорение только в одном шарнире; см. гл. VI, § 2.
- 21. СТАТИНА МЕХАНИЗМА /Nc, RQ/ осуществляет расчет реакций в шарнирах и усилий и моментов приводов; см. гл. VI, §3.
- 22. ИНЕРЦИЯ МЕХАНИЗМА /Nc/ осуществляет расчет сил и мо- ментов сил инерции приведенных к ценрам масс звеньев; см. гл. VI, $\S 4$.
- 23. МОЩНОСТЬ /Nc, a/ осуществляет расчет мощности; см. гл. VI, §4.
- 24. ДИНАМИКА /Nc, D, H, h, ϵ , vs, ω s, ws, ϵ s/ осуществляет составление дифференциальных уравнений движения; см. гл. VI, §4.

В указанных процедурах используется ряд глобальных идентификаторов. Глобальные массивы STR, KP , ZV описаны в гл. I, §4. Ниже приводится описание остальных глобальных массивов.

Номера строк /первое измерение/ во всех случаях соответствует номеру цепи. Массив относительных координат x; номера столбцов соответствуют номеру кинематической пары в цепи. По третьему измерению помещаются последовательно $\mathbf{x}^{\mathbf{i}}$, $\mathbf{\ddot{x}^{i}}$, $\mathbf{\ddot{x}^{i}}$.



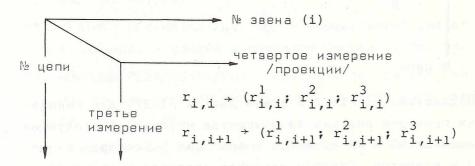
Информация о реакциях приводов $/p^{\mathbf{i}}$, см. /1.2// помещается в массив РР. В \mathbf{i} -ый столбец массива помещена реакция \mathbf{i} -го привода на \mathbf{i} -е эвено, то-есть



Воздействие на /i-l/-е звено дается /1.2/. Для кинематических приводов реакция записывается нулевой. В следующих двух массивах GG и MM хранится информация о векторах внешних сил и моментов. Столбцы массивов соответствуют номеру звена в цепи, компоненты /проекции на неподвижные оси координат/ помещены по третьему измерению в последовательности. соответствующей базису неподвижной системы i, j, k.

В следующих двух массивах ЕЕ и RR будут храниться осстветственно единичные векторы осей кинематических пар и векторы из центров шарниров к центрам масс эвеньев. Массив ЕЕ трехмерный массив

В массиве RR на пересечении скроки и столбца помещаются два вектора \mathbf{r}_{ii} и $\mathbf{r}_{i,i+1}$. Их компоненты расположены соответственно по третьему и четвертому измерения в такой последовательности: сначала $\mathbf{r}_{i,i}$, затем $\mathbf{r}_{i,i+1}$. Условно запишем это в виде



Кроме массивов EE и RR введено еще два массива E0 и R0 с аналогичным строением. В E0 и R0 будут храниться эначения $\mathbf{e_i}$ и $\mathbf{r_{ij}}$ соответствующие нулевым эначениям относительных координат $\mathbf{x^i}$.

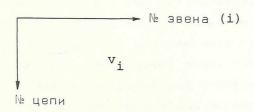
Следующий массив QQ – четырехмерный. В него помещаются базисы $/\mathbf{q_{i1}},\ \mathbf{q_{i2}},\ \mathbf{q_{i3}}/$ внутренних систем координат всех эвеньев. Будем рассматривать векторы $\mathbf{q_{ij}}$ как векторы-столбцы и составим из них матрицу $\mathbf{A_i}$.

$$A_{i} = ||q_{i1}; q_{i2}; q_{i3}||$$
 (П.1)

где і - номер звена. Тогда можно наглядно представить себе массив QQ, как такую матрицу, компонентами которой являются матрицы /П.1/ для различных цепей и звеньев. Векторы задаются в проекциях на неподвижные оси. Условно запишем это в виде

Звенья номеруются все в том числе оба базовых звена для зам-кнутой цепи.

Следующие массивы VV и $\Omega\Omega$ предназначены для хранения значений соответственно скоростей центров масс звеньев, угловых скоростей звеньев. Оба массива трехмерные и имеют одинаковое строение, например, для скоростей ${\bf v_i}$;



Последний глобальный массив $\Lambda\Lambda$ предназначен для хранения матриц связи для замкнутых цепей. Смысл матриц связи и строение этого массива объяснены в гл. II, §3.

Спецификация глобальных идентификаторов /массивы/:

1. STR[1:Nc, 1:14] - integer;

```
2. KP[1:Nc, 1:nn, 1:4] — integer где nn — максимальное число кинематических пар в одной из цепей
```

```
- real ;
3. ZV[1:Nc, 1:nn, 1:27]
4. X[1:Nc, 1:nn, 1:3]
                          - real ;
5. PP[1:Nc, 1:nn]
                           - real ;
6. GG[1:Nc, 1:nn, 1:3]
                          - real ;
7. MM[1:Nc, 1:nn, 1:3]
                        - real ;
8. E0, EE[1:Nc, 1:nn, 1:3] - real;
9. R0, RR[1:Nc,0:nn,1:2,1:3] - real;
10. Q0,QQ[1':Nc,0:nn,1:3,1:3] - real;
                         - real ;
11. VV[1:Nc, 0:nn, 1:3]
12. \Omega\Omega[1:Nc, 0:nn, 1:3]
                          - real ;
13. AA[1:Nc, 1:nns, 1:nnu+6] - real,
```

где nns и nnu - максимальное число дополнительных и основных шарниров соответственно.

приложение п

Краткое описание алгоритмического языка АЛГОЛ-60

АЛГОЛ представяет собой универсальный язык для записи аглоритмов, по форме нопоминающий обычное математическое описание. В нем используются буквенные обозначения для различных переменных и обычные знаки математических операций. Благодаря этому алгоритмы, записнные на АЛГОЛе легко читаемы и изучение этого языка не может составить каких-либо трудностей для инженера. Если и нельзя сказать, что во всех следует программировать на АЛГОЛе, то в смысле ясности и простоты записи алгоритмов он не имеет себе кокурентов.

В настоящее время имеется ряд подробных руководств для изучения АЛГОЛа, часть из которых приводится в конце этого Приложения. В них АЛГОЛ излагается с большой сторостью и подробностью, в связи с чем для человека не знакомого с программированием изучение АЛГОЛа может представить трудности. Достоточно сказать, что для описания грамматики АЛГОЛа используется еще один специальный язык, так называемый юзык Бэкуса. На этом языке можно формально очень кратко записать правила АЛГОЛа, однако, для того чтобы понимать язык Бэкуса нужен определенный "алгоритмический навык". Вероятно, для того чтобы "знать АЛГОЛ" нужен юзык Бэкуса, но для практического использования АЛГОЛа такая формализация излишне.

В настоящем Приложении ставится ограниченная задача дать необходимый минимум свединий для чтения и понимания АЛГОЛьных тексов. При этом мы будем стремиться к максимальной простоте

изложения, рассчитывая на человека совершенно не знакомого с программированием. Конечно, некоторые тонкости АЛГОЛа при таком изложении останутся за "бортом", но зато автор надеется показать, что АЛГОЛ действительно очень прост. Для более глубокого изучения можно обратиться к любому из указанных в конце источников.

I Основные принципы построения программ на АЛГОЛЕ

Весь алгольный текст можно разделить на две части:

- 1. описание обозначений;
- 2. операторы, указывающие каким образом производить вычис-

Допустим для расчетов нужны три пеменные ${f x}$, ${f y}$, ${f z}$, которые могут принимать действительные значения, матрица ${f A}$ размерности ${f 3x2}$ и целое число ${f i}$. На АЛГОЛе описание этих переменных будет иметь вид

begin

real x, y, z;
array A [1:3, 1:2];
integer i;

end;

Здесь служебное слово <u>begin</u> указывает на начало алгоритма /или некоторой части алгоритма/, а слово <u>end</u> - на его конец. Часть алгольного текса расположенная между этими словами называется блоком.

Служебные слова в АЛГОЛе либо набираются жирным шрифтом, либо подчеркиваются. Мы будем пользоваться вторым обозначением, как более наглядным.

Описания всегда помещаются в начале блока. В описании сначала указывается служебное слово определяющее тип переменных /real - действительные, array - матрицы или, как принято говорить в АЛГОЛе, массивы, integer - целые/ и затем через запятую указываются переменные этого типа. Описания переменных различного типа отделяются друг от друга точкой с запятой. Размерность массива указывается в квадратных скоб-ках после обозначения массива, так, как показано в примере.

Описания переменных одновременно выполняют функции /частичные/ распределения памяти ЭВМ. "Прочитав" это описание транслятор* отведет 10 ячеен памяти для хранения значений переменных х, у, z, i и компонент матрицы А.

После того как описаны переменные можно выполнять с ними различные вычислительные опереции. Например, блок может быть записан как

```
begin integer i;
    real x, y, z; array A[1:3, 1:2];
    x: = 1;
    y: = 2×x;
    i: = 0;

ml: i: = i+1;
    A[i,1]: = x;
    A[i,2]: = y;
    z: = y;
    y: = x+y;
    x: = z;
    if i < 2 then go to ml
end</pre>
```

^{*} Транслятором называется специальная программа, которая "переводит" программу с АЛГОЛа на язык вычислительной машины.

Ниже уже знакомого описания переменных здесь записано десять операторов. Эти операторы указывают, какие и в какой последовательности выполнять вычисления. Каждый оператор отделен от другого точкой с запятой. Для наглядности все операторы выписаны в столбик, что, вообще говоря, совсем не обязательно.

В первом же операторе встречается новое обозначение := . Это обозначение оператора присваивания. Запись a: = b следует понимать как: "переменной a присвоить значение переменной b". В рассматриваемом примере в первом операторе переменной x присваивается значение равное единице.

Оператор присваивания является важным элементом АЛГОЛа, поэтому остановимся на нем подробнее. В приведенном примере он встречается девять раз. Слева от знака := всегда должно стоять обозначение накоторой переменной, а справа можеть стоять обозначение некоторой переменной или некоторое выражение, как, например, во втором и восьмом операторах. Во втором операторе х умнажается на два и полученный результат присваивается у. Восьмой оператор более интересен. Здесь обозначение паременной у встречаетсякак слева, так и справа от оператора присваивания. Этот оператор следует понимать так: "взять то эначение у, которое было перед выполнением этого оператора, прибавить к нему эначение х и полученный результат вновь присвоить переменной у". Восьмой оператор хорошо илюстрирует различный смысл, который имеют знаки = и := . Запись со знаком равенства y = y + x лишена смысла для всех $x \neq 0$, в то время как оператор присваивания будет выполняться для любых /разумных/ значений х.

В пятом и шестом операторах значения х и у присваиваются различным компонентам матрицы А. Эти операторы показывают как в АЛГОЛе записываются отдельные элементы массивов. После обозначения массива в квадратных скобках указываются номера компонент по каждому измерению. Номера компонент могут задаваться как числами, так и какой-либо переменной, в

рассматриваемом примере переменной і.

Любой оператор в АЛГОЛе может быть "помечен". Метка должна представлять собой некоторую комбинацию букв и цифр*. Метка отделяется от оператора знаком двоёточие. В премере метка стоит перед четвертым оператором. Метки служат для того, чтобы к этому оператору можно было перейти из какого-либо другого места программы. В данном случае переход на метку транительного опертора.

Последний в /П.1/ оператор if i < 2 then go to ml следует читать как: "если значение переменной i меньше или равно двум, то перейти к метке ml и выполнять оператор, записанный вслед за меткой". Операторы такого типа называются условными операторами. Часть оператора стоящая между служебными словами if и then называется условием /в данном случае i < 2/. После then стоит некоторый другой оператор, который выполняется в том случае, если выполняется условие. В данном случае после then записан оператор перехода: go to ml, который означает: "перейти к метке ml".

Операторы в АЛГОЛ - программе выполняются в той последовательности, как они записаны. С помощью операторов перехода и условных операторов можно прервать "обычный" порядок выполнения программы и продолжить расчет начиная с некоторого другого оператора, намеченного меткой, В частности, с помощью условных опереторов и операторов перехода можно организовывать цеклы, то-есть повторение некоторого участка программы заданное число раз.

В рассмотренном примере /П.1/ имеется один цикл, который образуется меткой ml и последним условным оператором. Часть программы, расположенная между этой меткой и условным оператором будет повторена три раза до тех пор пока целая

^{*} Во многих трансляторах требуется, чтобы метка всегда начиналось с буквы.

переменная і /параметр цикла/ не станет равной 3.

Приведенный премер представляет собой полную алгольную программу, хотя и в простейшем варианте. Усли этот текст нанести на ленту или перфокарту, а затем ввести в ЗЦВМ, то транслятор по нему составит рабочую программу на языке машины. После выполнения рабочей программы будет рассчитана матрица А со слудующими компонентами

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{bmatrix}$$

Конечно, в этом примере использовалась только небольшая часть возможностей АЛГОЛа. В следующих пунктах более подробно обсуждаются эти возможности, в частности как описываются различные переменные, как представляются числа, как строятся циклы, процедуры, переключатели и т.д. Но предварительно остановимся на одном из основных понятий в АЛГОЛе - понятии блока.

Как уже указывалось, преведенный премер программы представляет собой один блок. В более сложных случаях программа может состоять из нескольких блоков, расположенных один в другом, напремер так, как показано ниже

```
l.
begin real x;

2.
begin real y;

3.
begin real y,z;

3.
end;

2.
end;
```

1. end;

Здесь указаны только граничные для блоков слова $\underline{\text{begin}}$ и $\underline{\text{end}}$ и описания переменных, а текст блоков заменен многоточием. Цифры над словами не имеют отношения к АЛГОЛу и проставлены только для того, чтобы указать какие $\underline{\text{begin}}$ и end соответствуют каким блокам.

В примере имеется четыре блока. В первый блок входят все остальные, третий блок кроме того входит во второй. Заметим сразу, что блоки могуть быть целиком расположены один в другом, но не могут частично пересекаться. Например канструкция

1. begin
2. begin
1. end;
2. end;

совершенно недопустима в АЛГОЛе и будет воспринята транслятором как два блока целиком расположенных один в другом со всеми вытекающими отсюда последствиями.

Каждый блок независимо от того является ли он внешним или внутренним может содержать описание некоторых переменных. В примере /П.2/ в каждом из блоков описаны действительные переменые x, y, z, a. И тут действует один чрезвычайно важ-

ный принцип. Описание выполненное в некотором блоке справед-ливо только для этого блока. Например в /П.2/ переменную можно использовать только в блоке 3, а переменную а только в блоке 4. В то же время пеменную ${\bf x}$ можно использовать во всех блоках, так как она описана в самом внешнем из блоков.

Особенно интересен случай с переменной у. Она описана во втором и третьем блоках. Второй блок является внешним по отношению к третьему и вообще говоря описанные в нем переменные /или массивы/ являются глобальными по отношению к третьему блоку. Однако, если глобальная и локальная переменные обозначены одинаковой буквой/ или сочетанием букв и цифр/, то предпочтение всегда отдается локальной переменной, и в блоке 3 можно использовать только тот у, который локально описан в этом блоке. Важно пенять, что дважды описав у во втором и третьем блоках, мы тем самым описали две различные переменные, только обозначили их одной буквой*. При этом "жизненное пространство" для каждой из переменных будет четко разграничено. Внутри третьего блока у будет обозначать величину описанную в этом блоке локально, а вне этого блока у будет обозначать величину, описанную в блоке 2.

Таким образом, можно сформулировать два правила, касающихся использования переменных

- 1. Переменные, описанные во внутреннем блоке нельзя вызвать из внешнего блока.
 - 2. Переменные, описанные во внешнем блоке можно вызвать из внутреннего блока за исключением случая, когда во внутреннем блоке описано точно такое же обозначение для локальной переменной.

Из этих правил следует, например, что внутренние блоки могут вообще не содержать описаний переменных, и тогда в них будут использоваться только глобальные величины. Такие блоки

^{*} Для этих переменных будет выделены две отдельные ячейки памяти.

встречаются очень часто.

С другой стороны, допустим в некотором внутреннем блоке используются глобальные величины x, y, z, и допустим для выполнения вычислений нужно еще три действительные переменные. Тогда эти три действительные переменные можно обозначить любыми буквами, отличными от x, y, z, например, a, b, c. При этом не имеет значения, если во внешнем блоке уже имелись переменные обозначенные этими буквами: во внутреннем блоке под a, b, c будут пониматься новые переменные и для них будут отведены отдельные ячейки памяти.

Благодаря принципу локализации блоки преобретают некоторую автономность и их можно програмировать в значительной степени независимо от остальной программы.

Может возникнуть вопрос, почему бы не описать все переменные сразу в начале первого блока и затем уже нигде не описывать локальных переменных. Во многих случаях так и делают, но это не всегда удобно.

Дело в том, что при выходе из блока все ячейки памяти занятые под локальные переменные освобождаются и в дальней-шем могут использоваться для хранения другой информации, например, локальных переменных в другом блоке. Допустим имеем два примера записи программы.

begin with more many water							
	begin www.asaasaasaa						
	<pre>array A[1:8,1:8];</pre>						
	end;						
	begin						
	<pre>array B[1:4,1:16]</pre>						
	Sale Halls (Fig. 4) P 40 923						

В обоих случаях программа состоит из трех блоков, причем блоки 2 и 3 включены в общий для них внешний блок 1. Для проведения вычислений /для хранения промежуточных результатов/в блоке 2 нужна матрица А, а в блоке 3 - матрица В.

В левом премере эти матрицы описаны локально, каждая в своем блоке. Это означает, что память под матрицу А будет отведена после того, как в процессе вычеслений мы "войдем" в блок 2. После выхода из блока 2 все 64 ячейки, которые занимала матрица А, считаются свободными.

Следующим записан блок 3. После входа в него должна быть отведена память под матрицу В. В принципе матрица В может быть организована в тех же ячейках памяти, где раньше располагалась матрица А.

В правом примире обе матрицы описаны во внешнем блоке и, следовательно, для блоков 2 и 3 являются глобальными. В этом случае не происходит перераспределения памяти и в программе должна быть отведена память под оба эти массива*. Следовательно, при локальном описании переменных можно экономить оперативную память ЗЦВМ.

^{*} Конечно, не всегда дело будет обстоять так хорошо, как в этом премере, и здесь многое зависит от свойств транслятора. Однако, общий принцип остается верным: после выхода из блока все ячейки занятые под хранение локальных величин освобождаются и могут использоваться для хранения другой информации.

Экономия памяти не единственная причина, по которой следует пользоваться локальным описанием переменных. В сложеных программах отдельные блоки часто программируются в различное время и даже разными людьми. При глобальном описании всех переменных следить за соответствием обозначений было бы крайне трудно. Если же те переменные, которые используются только в данном блоке, описывать локально, то их можно обозначать любыми буквами не заботясь об обозначениях в других блоках. Это существенно облегчает "стыковку" отдельных блоков, тоскольку нужно следить за соответствием обозначений только общих переменных.

Блоки автономны не только в смысле описаний переменных, но также расположенных в них меток. Относительно меток существует также два твердых правила:

- 1. из внешнего блока нельзя перейти на метку внутренниго блока;
- 2. из внутреннего блока можно переходить на метки внешнего блока за исключением случая, когда во внутреннем блоке имеется точно такая же метка.

Рассматрим две программы

begir	<u>1</u>				begin
	go to	ml;		COM:	ррі: п н. н.
	begin			15 9 19	begin
	ml: .			•	go to pl;
	end;.				<u>end</u> ;
end;					end

В правой программе переход на мекту pl возможен, а в левой перейти на ml нельзя, и эта программа будет воспринята транслятором как ошибочная

Еще один пример

end

Здесь имеются две одинаковые метки во внутреннем и внешнем блоках. Оператор перехода /go to ml/ расположен во внутреннем блоке. В этом случае предпочтение отдается внутренней метке и переход будет осущестлен так как показано стрелкой.

Приведенные выше правила относительно меток можно сформулировать несколько в иной форме. Все метки расположенные в данном блоке считаются локализированными в нем, подобно локальным переменным этого блока. Нельзя войти в блок минуя его начало /то-есть на внутреннюю метку/; вход в блок возможен только "через" слово begin и все следующие за ним описания локальных переменных.

Локализация меток играет ту же роль, что и локализация переменных. Программируя отдельные блоки можно обозначения для меток выбирать произвольно /независимо от других блоков/ и следить только за теми метками на которые осуществляется выход во внешний блок. Например, программа

end

записана правильно в смысле меток и тут не будет никакой путаницы при осуществлении переходов.

Теперь мы переходим еще к одному основному понятию АЛ-ГОЛа - к процедуре. Процедура это фактически тоже блок, но обладающий следующими особенностями:

- этому блоку дается некоторый заголовок;
- в блоке могут использоваться параметры, которые указываются в заголовке;
- этот блок можно выполнить в любом месте программы, для чего достаточно записать в этом месте заголовок и указать фактически параметры*.

Вопрос для чего дужны и как записываются процедуры рассмотрим на простом примере. Пусть в процессе вычеслений требуется многократно осуществлять векторное произведение различных векторов. Очевидно, не рационально каждый раз, когда нужно умножить векторы писать соответсвующий алгоритм. Тогда можно составить процедуру

```
procedure VEKT(a,b,c);
array a, b, c;
begin

a[1]: = b[2] x c[3] - b[3] x c[2];
a[2]: = b[3] x c[1] - b[1] x c[3]; (Π.3)
a[3]: = b[1] x c[2] - b[2] x c[1]
```

^{*} Там, где справедливо описание процедуры /см. ниже/

На то что эта процедура указывает служебное слово <u>procedure</u>. Буквы VEKT образуют название процедуры. Для названия могут использоваться любые буквы латинского алфавита и цифры, но первой обязательно должна стоять буква.

Вслед за названием в скобках указываются параметры процедуры. В данном случае это три вектора a, b, c. Процедура делает следующее: умножает b вектор. на c и результат присваивает вектору a.

После списка параметров /в скобках/ следует спецификация этих параметров, то-есть указание к какому типу они относятся . В данном случае все три параметра массивы, что и указывается служебным словом <u>array</u>.

Вслед за спецификацией следует тело процедуры. Это фактически блок, в котором могут использоваться указанные в заголовке пораметры.

Теперь, если в каком-либо месте программы нужно умножить например, векор ${\bf z}$ на вектор ${\bf y}$ и результат присвоить вектору ${\bf x}$, то для этого достаточно записать

VEKT
$$(x, z, y)$$
 $(\Box.4)$

После этого транслятор по заголовку VEKT найдет нужную процедуру, вместо параметров a, b, c "подставит" векторы x, z, y и осуществит требуемые вычисления. Приведенная выше /П.4/ запись называется обращением к процедуре. В обращении можно указывать любые требуемые векторы, а транслятор сам определит какому параметру соответствует какой вектор и осуществит требуемые "подстановки".

Таким образом с помощью процедур можно существенно упростить и уменьшить программы.

Рассмотрим теперь вопрос, где в программах нужно помещать описания процедур и как процедуры связаны с блочной структурой программы.

Описание процедуры /например, такое как приведеный выше текст процедуры VEKT/ может быть помещено в начале любого блока, тамже где помещаются описания переменных, например,

```
begin

array p,q,s[1:3];

procedure VEKT(a,b,c);

array a,b,c;

begin . . . end;

/вычисление векторов р и q/

VEKT(s,p,q);

end
```

описание процедуры

Относительно процедур действует такой же принцип локализации, как и относительно переменных. Если процедура описана в некотором блоке, то к ней можно обратиться из данного блока и из всех внутренних блоков, но нельзя обратиться из внешнего блока. Например, из двух программ

```
begin
    array p,q,s[1:3];
    procedure VEKT a,b,c;
    array a,b,c;
    begin . . .
    end;
    begin
        array x,y,z[1:3];
    . . VEKT(x,y,z); . . .
    end; . . .
    VEKT(s,p,q);
    end
```

begin
 array p,q,s[1:3];
 begin
 array x,y,z[1:3];
 procedure VEKT(a,b,c);
 array a,b,c;
 begin
 end;
 VEKT(x,y,z); . . .
end;
 VEKT(s,p,q); . . .
end

левая программа правильная, а правая нет. Разница состоит в том что в левой программе процедура VEKT описана во внешнем блоке, а в правой - во внутреннем. Ошибка здесь в том, что второе обращение к процедуре выполнено из внешнего блока, в то время как сама процедура описана локально во внутреннем блоке и, следовательно вне этого блока "не существует".

В этой вводной части мы познакомились с основными и пожалуй наиболее сложными понятиями АЛГОЛа: блоки, локальные и глобальные переменные, локальные метки, процедуры. Все это кординальные для АЛГОЛа вещи. Дальше у нас остоются в значительной степени технические вопросы: описание различных переменных и чисел, организация циклов, булевские переменные, условные выражения, процедуры - функции и т.д. Все эти вопросы будут расспотрены в следующих пунктах.

II Описание переменных и чисел

Различные переменные в АЛГОЛе могут быть обозначены некоторыми сочетаниями букв и цифр. Такие сочетания называются идентификаторами. Первым знаком идентификатора непременно должна быть буква, например:

x, yl, zv26, alfa.

В АЛГОЛе строго различаются переменные, которые могут принимать только целочисленные значения от переменных могущих принимать действительные значения. Первые переменные обозначаются служебным словом <u>integer</u> /целий/ и их описание иметет вид

integer i, j, kl, summa;

Дийствительные переменные обозначаются служебным словом <u>real</u>, напримир

real s0, tg26, skorost;

Кроме целых в действительных переменных в АЛГОЛе употребляются также булевские переменные, которые описываются с помощью служебного слова boolean, например

boolean bl, b2, povtor:

Считается, что булевские переменные могут принимать только два значения: истина и ложь, которые обозначаются служебными словами true и false*. Булевские переменные широко используются при осущестлении различных логических операций. Ниже мы еще остановимся на использовании булевских переменных.

Помимо отдельных переменных в АЛГОЛе можно описать массивы различной размерности. В предыдущем пункте уже встречалось описание массивов. Сейчас остановимся на этом подробнее.

При описании прежде всего указывается тип массива с помощью служебных слов, то-есть помощью служебных слов,

integer array
boolean array
real array

В последней строчке слово <u>real</u> можно опускать. Вслед

^{*} Реально в ЭЦВМ, в ячейки соответствующей булевской переменной будет находится одно из двух чисел, например 1 /в случае <u>true</u>/ и 0 /в случае <u>false</u>/.

за указанием типа массива пишется его идентификатор и в квадратных скобках указывается размерность, напримир,

В скобках для каждого измерения указываются границы для изменения индекса: слева нижняя граница и затем, через двоеточие - верхняя. Информация по каждому измерению отделяется друг от друга запятыми.

Например массив А1 трехмерный и каждая компонента этого массива может быть обозначена как

где і,ј,к могут принимать целые значения в границах

equation
$$1 \le i \le 4$$
; $1 \le j \le 3$; $0 \le k \le 2$

Аналогично для массива В2. Допустим одной из компонент этого массива нужно присвоить значение 0,1. На АЛГОЛе это можно записать так

$$B2[-5,2,18,1] := 0.1;$$

Число измерений массивов и значения границ индексов в АЛГОЛе не ограничивается. Однако, важно чтобы левая граница никогда не была больше правой. Напримир, запись

допустима, в то время как запись

ошибочна, так как по первому измерению неверно указаны граничные значения.

Граничные значения могут задаваться переменными или даже некоторыми выражениями, напримир

Переменные n и q должны быть глобальными для блока, в котором содержится описание массивов A, B, CD. Если n и q переменные типа real, то при определении верхней границы в АЛГОЛе соответсвующие числа будут автоматически округляться до ближайшего целого числа*.

Обратим внимание, что в вышеприведенном описании два массива В и CD имеют одинаковую размертость.

Целие числа записываются как обычно, например

2 - 10 + 1836

Знак + можно опускать. Запись вида

9,0 или 9.0

для целых чисел не допускается.

Дийствительные числа могут записываться либо как целые /если по значению они совпадают с целым числом/, либо в виде десятичной дроби. В последнем случае дробная часть отделяет-ся от целой точкой, например,

- 8 7.126 0.02

В АЛГОЛе существует особое обозначение, дла того чтобы записать степени числа 10. Оне записываются маленькой цифрой 10, рядом с которой указывается показатель степени, например

 10^5 , 10^{-6} , 10^{+5}

Ниже приводится два примера очень похожих программ

^{*} То-есть, если n - число типа \underline{real} , то при вычислении инденса вместо n будет подставлена целая часть числа /n+0,5/

В первой программе описание выполнено правильно, во второй имеется три ошбки.

Во первых во внутреннем блоке описана целая переменная n и, следовательно, глобальная переменная обозначенная такой же буквой во нутреннем блоке недоступна. Но при входе во внутренний блок локальное n не имеет определенного значения, поэтому граничные пары в массивах y, k, l не могут быть определены правильно. Аналогичная ошибка допущена и при описании массива p, где используется локальная переменная p.

Вторая ошибка состоит в том, что переменная ј, описанна ая во внутреннем блоке, вызывается из внешнего блока.

В третьих, в последнем операторе программы неверно записано целое число 100.

III Арифметические операторы и стандартные функции

Для обозначения арифметических операций в АЛГОЛе используются знаки похожие на обычные математические символы. Так сложение, умножение и деление обозначаются следующими знаками -, +, /, например,

$$x: = a + 0.2$$
; $y: = x \times s$; $a: = b/x$:

Возведение в степень изображается стрелкой, рядом с которой записывается показатель степени, например,

$$x: = a \uparrow 2$$
; $y: = y \uparrow 0.5$; $b: = s \uparrow 1$;

С помощью описанных операторов могут записываться различные математические выражения, например,

$$a\uparrow 2 + bxc + c\uparrow 2$$

((axb\frac{2}{c})x2+x)/2 (\Pi.5)

В выражениях могут употребляться только круглые скобки. Последовательность совершения операций такая же, которая принята в математике. Если нет скобок, то сначала выполняются все возведения в степень в парядке записи /то-есть слева направо/, затем все умножения и деления и потом сложения. Если проставлены скобки, то последовательно выполяются по описанному правилу выражения внутри скобок. Например, последнее выражение в /П.5/ будет вычисляться в следующей последовательности

$$b+2$$
; $axb+2$; $axb+2+c$; $(axb+2+c)x2$; $(axb+2+c)x2+x$; $((axb+2+c)x2+x)/2$.

При выполнении операций сложения, вычитания, умножения, деления и возведения в степень участвующие в них переменные могут быть как действительного, так и целого типа.

Результат выполнения операций +, -, х будет целым числом, только если оба аргумента целые числа. При делении результат всегда будет действительным числом. При возведении в степень результат будет целым числом, только в случае целого основания и целого неотрицательного показателя степени. Возведение отрицательного числа в действительную степень не определено.

В АЛГОЛе имеется еще одна арифметическая операция предназначенная исключительно для целых чисел. Эта операция деления целых, обозначаемая как ÷. В качестве резултата

берется целая часть от [a/b] и присваевается знак a/b .

В АЛГОЛе предусмотрены стандартные обозначения для некоторых общеупотребительных функций:

abs(x) - абсолютная величина X;

 $\operatorname{sqrt}(\mathbf{x})$ - қорень квадратный из \mathbf{x} ;

sin(x) - тригонометрические функции; cos(x)

 $\operatorname{arctg}(\check{\mathbf{x}})$ - главное значение обратной тригонометричесной функции;

ln(x) - натуральный лагорифм;

 $\exp(x)$ - показатвльная функция $/e^{x}/;$

entier(x) - целая часть числа x;

$$sign(x)$$
 - энак x. $sign(x) = \begin{cases} 1 \text{ при } x > 0 \\ 0 \text{ при } x = 0 \\ -1 \text{ при } x < 0 \end{cases}$

При записи указанных функций аргумент может быть арифметическим выражением, например,

$$sqrt(x\uparrow 2 + y\uparrow 2);$$

Аргумент всегда записывается в скобках, запись вида

sin x; sqrt x

ошибочна*.

Стандартные функции могут употребляться в арифметических выражениях наряду с другими переменными, например,

$$y: = a+b+c - \sin(a/b) + a/abs(x);$$

Кроме описанных стандартных фукций в АЛГОЛе можно составить и другие функции, требуемые в данной задаче. Это можно сделать с помощью процедур, которые уже описывались в П.І. Ниже будет описана еще одна форма процедур, так называемые процедуры-функции.

IV Логические выражения

Как уже указывалось, в АЛГОЛе помимо переменных типа действительной и целой, применяются также булевские переменные. Последние могут принимать только одно из двух значений true /истина/ и false /ложь/. Описание булевских переменных выполняется с помощью служебного слова boolean, например,

boolean b, c, D;

а присваивание значений осуществляется как обычно с помощью оператора присваивания, то-есть

b: =
$$\underline{\text{true}}$$
; c: = a > b; D: = a \neq b;

В последних двух примерах слева от знака оператора присваивания записано логическое выражение. Если соотношения записанные справа справедливы /например, при a=5, b=2/ то обе переменные C и D получат значение true, если несправедливы /например a=b=2/ — то false.

В логических выражениях могут использоваться следующие знаки соотношений.

^{*}Такая запись будет воспринята транслятором как новые идентификаторы.

Логические выражения могут строится и непосредственно из булевских переменных с помощью специальных булевских операций, обозначаемых как

$$\equiv \mid \neg \mid V \mid \Lambda \mid \neg$$
 (\square .6)

Эти операции называются соответственно: "совпадает, влечет, или, и, не".

Допустим эти операции применяются к двум логическим переменным х и у. Тогда результаты выполнения этих операций будут иметь значения, указанные в следующей таблице /значениям true соответствует плюс, а false - минус/.

х	+	+	-	=
У	+	-	+	-
$x \equiv y$	+	-		+
хэу	+	-	+	+
хүу	+	+	+	-
х Л У	+	U-1	-	-
٦x	-	-	+	+

С помощью знаков отношений, логических операций и скобок можно составлять сложеные логические выражения. Операции записаны в /П.6/ в порядке возрастания старшинства, относительно скобок /скобки только круглые/ действуют те же правила, как и в арифметических выражениях.

Для того чтобы показать, как используются булевские переменные рассмотрим простой пример.

Допустим нужно возвести действительную переменную z в квадрат, но только в том случае, если выполняется одно из следующих условий:

1. булевские переменные В и С имеют значения соответ-

ственно true и false;

2. В имеет значение \underline{false} , но при этом либо $a > b^2$ /a,b — тип \underline{real} / либо $a \le 0$ На АЛГОЛе это можно записать так

Булевские переменные большей частью используются в условных операторах, причем чаще всего применяюстя операции V, \wedge , \neg .

V Условные операторы и условные выражения

Мы уже дважды сталкивались с условными операторами в ППІ и IV. Рассмотрим теперь их подробнее.

Условные операторы начинаются служебным словом <u>if</u> /если/, за которым следует логическое выражение или булевская переменная. Затем пишется служебное слово <u>then</u> /тогда/ и далее оператор, который нужно выполнить. Например,

if A > 3 then x: = 0;if b then go to m21;

Если в этих примерах выполняются условия /то-есть переменная A действительно больше - равна трем, и b имеет значение \underline{true} /, то выполняются следующие за \underline{then} операторы /то-есть x: = 0 и go to m21/. Если условия не выполняются, то соответствующие операторы также не выполняются. В этом случае выполняется оператор, записанный вслед за условным.

В условном опереторе можно использовать не только операторы присваивания и перехода, как в этом премере, но так-же любые блоки, опереторы цикла, операторы процедур, например,

if B then begin end;

Блок будет выполнен, если В = true.

Наложено ограничение только на использование в условных опереторах других условных опереторов, то-есть нельзя писать

if B then if A then . . .

но вместо етого можно

<u>if B then begin if A then ... end;</u>или <u>if</u> B A <u>then</u> ... Возможна другая форма условных операторов, например

if B then x: = 1 else x: = 0;

Здесь добавлено еще одно служебное слово else /его можно перевести "в противном случае"/. Оператор записанный после этого слова будет выполняться в том случае, когда условие несправедливо /В \equiv false/. После else могут быть записаны любые опереторы, в том числе и условые, например

 \underline{if} B \underline{then} x: = 1 \underline{else} \underline{if} a > b \underline{then} x: = 0;

Условные выражения отличаются то условных операторов тем, что после слов then и else записаны не операторы, а арифметические или логические выражения /в частном случае могут быть записаны числа или идентификаторы переменных/

Например,

a: = if x > 1 then (c+d) else 0;

B: = if $c \le 1$ then d v else g < s;

В первом примере а будет присвоено значение /c+d/, если x>1 и ноль в противном случае. Во втором примере булевской переменной В присвоено значение логического выражения d>v/то-есть \underline{true} или $\underline{false}/$, если $c\le 1$ и выражения g< s в противном случае.

И, наконец, последним рассмотрим в этом пункте так называемое условное именующие выражение. Допустим, при выполнении некоторого условия а > b нужно перейти на метку М, а в противном случае на метку N. Это можно записать с помощью следующей конструкции

go to if a > b then M else N;

Правила АЛГОЛа допускают запись различных сложеных условых выражений, например,

$\frac{\text{if if a then b else if c then d else e then}}{\text{true else if f then false else true;}}$ ($\Pi.7$)

Разбор подобны примеров и описание способов их расшифровки можно найти в большинстве руководств по АЛГОЛу. Мы не будем здесь этим заниматься, поскольку автор не встречал случая, где такого рода сооружения были бы действительно необходимы.

VI Операторы перехода и переключатели

С оператором перехада мы также уже встречались. С его помощью можно перейти к любой доступной метке и продолжить выполнение программы с этой метки. "Доступный" эдесь следует понимать с точки эрения локализации меток в блоках и этот вопрос подробно обсуждался выше в П.1. Оператор перехода имеет вид

go to start;

где <u>go to</u> служебные слова, start - идентификатор метки.

В предыдущем пункте встречалась более сложная конструкция условных именующих выражений, с помощью которых можно переходить на одну из двух меток, например,

go to if B then metka 1 else metka 2;

где metka 1 и metka 2 идентификоторы двух меток.

В практике может встретиться случай, когда меток более двух и в зависимости от выполнения некоторых условий нужно

осуществлять переход на одну из них. Для этого в АЛГОЛе предусмотрен переключатель. Он строится следующим образом.

В начале блока, там где помещаются описание переменных, записывается служебное слово switch /переключатель/, затем идентификатор с указанием индекса в квадратных скобках, затем следует знак присваивания, после которого через запятые перечисляются все нужные метки. Например,

switch P: = M1, M2, M3, start, finish;

Для того чтобы осуществить переход к одной из указанных меток, нужно в тексте программы записать оператор перехода к переключателю, вида

go to P[k]

где k – индекс, "указывающий порядковый номер метки в правой части списка переключателя.

Например, если k=2 будет осуществлен переход на метку M2 , если k=4, то на метку start и $\tau.д.$

Вместо k может быть записано арифметическое выражение, если оно типа <u>real</u> , то результат будет автоматически округлен до ближайшего целого чесла. То-есть оперетор

обеспечит переход к метке М2.

Наконец, укажем еще на одну сложную конструкцию, когда в качестве метки в списке одного переключателя, указывается идентификатор другого переключателя. Например, в блоке

```
begin integer i,j;
switch P: = M,G[i];
switch G: = M1, M2;
i: = 1; j: = 2
...
go to P[j];
M: ... M1: ... M2: ...
```

end;

при записанном обращении ураствуют в работе оба переключателя и осуществляется переход на метку M1.

VII Цикли

Оперетор цикла одна из самых важных конструций АЛГОЛа. В П.1 мы уже встечались с примером /П.1/, где строился цикл с помощью условного оператора и оператора перехода. АЛГОЛ позволяет записать этот цикл гораздо проще и нагляднее.

<u>for</u> i: = 1,2 <u>do begin</u> . . . end;

Эта запись будет выполняться следующим образом. Сначала /эта переменная называется параметром цикла/ присваивается значение I и выполнятся блок*, записанный после слова do. Везде, где в этом блоке встерается i, оно будет восприниматься равным идинице. После окончания блока i присваивается значение 2 и вновь выполняется блок.

Список значений, которые должен принимать параметр цикла может содержать более днвэх элементов, и элементами этого списка могут быть арифметические выражения, например

for x: = 1.1,1.5+2 \times x,7,9+a do y: = x 2 y;

Здесь оперетор $y:=x\ 2+y$ будет выполнен четыре раза при x=1,1; 7,7; 7; 9+a.

Последний пример интересен также тем, что эдесь параметр цекла х - действительное число

Часто параметр цисла должен изменятэся от некоторого значения а до значения b с шагом с. Такой цикл можно записать как

^{*} Вместо блока может в частном случае стоять один оператор

for i: = a step c until b do begin ... end;

Здесь і будет последователлно принимать значения а, a+c, a+2c,... и для каждого значения і будет выполнен блок записанный после do. Выполнение цикла прекратится как толь-ко значение і выйдет за границу b, то-есть окажется справедливым одно из двухусловий

i <u>></u> b если b > a, или i < b если b < a.

Таким образом a, b, c могуть быть положительными или отрицательными числами, например

<u>for</u> i: = 10 <u>step - 1 until 0 do</u> <u>begin end;</u>

если і описано как целое, цикл будет выполнен одиннадцать раз, последний раз при і = 0.

Обратим внимание, что цикл

for i: = 1 step 1 until a do begin ... end;

при ${\bf a}=0$ будет все же выполняться один раз для ${\bf i}=1$. Третья форма записи цикла выглядит так

for x: = A while B do begin ... end;

где А - арифметическое выражение, а В - булевское.

Цикл выполняется так: проверяется В и, если В \equiv true, то x присваивается значение A и выполняется записанный вслед за do блок. Затим опять проверяется B и, если B true, вновь x присваивается значение A и выполняется блок и т.д. до тех пор, пока B \equiv true.

Если с самого начала $B \equiv \underline{false}$, то цикл не выполнятся ни одного раза. Заметим, что хотя параметру x всегда присваивается одно и то же значение A, само это значение может меняься в результате выполнения блока, записанного в цикле.

В заключение отметим несколько важных случаев. Во-первых в качестве оперетора, который должен выполняться в циклеможет вновь быть записан оператор цикла, и такие конструкции часто встречаются

for i: = 1 step 1 until n do
 for j: = 1 step 1 until m do
 for k: = j step - 1 until 1 do
 begin . . . end;

Во-вторых из цикла можно выйти на внешнюю метку и тем самым прервать выполнение цикла. Но нельзя войти на метку внутрь цикла, минуя его заголовок. Здесь действует олщий принцип локализации /см.пІ/

нельзя

begin ...
 M1: ...
for i: = 1 step 1 until 10 do
begin ...
 go to M1; ...
 go to M1; ...
 go to M1; ...
 go to M1; ...
 end
end;
begin ...
for i: = 1 step 1 until 10 do
begin ...
 M1: ... end
end;

Если параметр цикла действительное число, то нужно иметь в виду, что значения таких чисел вычисляется в машине приближенно.

Например, если записать

for x:=0.1 step 0.1 until 1.0 do begin ... end;

нельзя гарантировать выполнение цикла 10 раз, поскольку на десятом шаге \mathbf{x} может оказаться либо чуть больше, либо чуть меньше единицы. Для того чтобы гарантировать выполнение цикла ровно 10 раз, нужно записать цикл так

for x:=0.1 step 0.1 until 1.05 do begin ... end; После окончания цикла значение параметра цикла не

определено.

VIII Процедуры

Процедуры уже описывались в п.І, где преводился пример процедуры VEKT . Поэтому целесообразно возвратиться к п.І и перечитать все, что там говорилось по этому поводу.

Процедура описывается с помощью служебного слова procedure за которым следует идентификатор процедуры, затем в круглых скобках через запятые указываются все формальные параметры, приводится спецификация параметров* и, наконец, записывается сам блок, образующий тело процедуры. Например,

```
procedure NORMA (n,a,b);
array b,a; integer n;
begin ... end;
(□.8)
```

В частном случае тело процедуры может состоять из одного оперетора или процедура может совсем не иметь параметров.

В качестве формальных параметров могут использоваться идентификаторы переменных, массивов, меток и других процедур

```
procedure example 1 (a,b,c,d,e,f,g,h);
   real a,d; integer b;
   array c,e; boolean h;
   procedure f; label g;
   begin .... end;
```

Обратим внимание, что текст <u>procedure</u> f является описанием формального параметра, а не новой процедурой; с по-

^{*} Спецификацию приводить не обязательно, однако, наличие спецификации облегчает работу транслятора

мощью служебного слова <u>label</u> описываются метки.

В часном случае процедура может совсем не иметь параметров и состоять из одного оперетора, например

procedure example 2; $y: = x \uparrow 3;$

В тексте программы, в том месте где требуется выполнение процедуры должно быть записано обращение к ней, которое состоит из идентификатора процедуры и списка параметров в круглых скобках. Например, к процедуре /П.8/ можно обратиться как

NORMA (m,c,d) $(\Pi.9)$

Параметры записанные в обращении /то-есть m, c и d/ называются фактическими в отличии от параметров n, a , b, которые фигурируют в описании и называются формальными. Фактические и формальные параметры должны быть одного типа и соответствующей размерности.

Обращение к прожедуре /П.9/ выполняется, грубо говоря следующим образом:

- формальные параметры n, a, b "заменяются" в теле процедуры /П.8/ на параметры m, c, d;
- тело процедуры "переносится" в то место программы, где записано обращение;
- выполняются описанные в процедуре вычисления.

При этом программиста не должно беспокоить, если в теле процедуры описаны переменные с такими же идентификаторами, как фактические параметры*. Транслятор обеспечит "разделение" локальных идентификаторов и пораметров и никакой путаницы не произойдет.

Процедуры представляют собой, повидимому, наиболее сложную часть АЛГОЛа. Мы остановимся на двух важных вопросах:

^{*} Разумеется нельзя, чтобы локальные идентификаторы совпадали с формальными параметрами

каким образом формальные параметры "заменяются" на фактические и что происходит с глобальными идентификаторами, которые могут встретиться в теле процедуры.

При обращении к процедуре фактические параметры могут вызываться двумя способами: по имени и по значению. Каждому из этих способов соответствует свое описание процедуры. Если описание выполнено так, как в /П.8/, то параметры вызываются по имени. Можно считать, что каждый раз, когда в теле процедуры /П.8/ требуется вызвать, например, переменную п, осуществляется автоматическая переадрессация к переменной с идентификатором m.

Вызов по значению можно себе представить следующей приближенной схемой: формальные параметры вызываемые значением, рассматриваются как новые переменные или массивы, и для них отводится машинная память. Перед выполнением процедуры этим переменным или массивам присваиваются значения, которые имеют в данный момент соответствующие фактические парамитры; далее все вычисления проводятся с этими "новыми" переменными или массивами.

Для того чтобы параметр ${\bf n}$ вызвать значением, описание процедуры NORMA следует записать как

procedure NORMA(n,a,b); value n;
integer n; array a,b;
begin end;

где на вызов эначением указывает служебное слово value. Параметры, которые вызываются эначением, обязательно должны иметь спецификацию.

Обращение к такой процедуре можно писать в виде /П.9/, но теперь возможно также и обращение вида

NORMA (3,a,b);

где на месте параметра n указано просто его значение.

Естественно по значению нельзя вызывать метки, переключатели или процедуры /не функции/.

Параметры вызываемые значением "не существуют" вне тела процедуры и их можно использовать только в данной процедуре. Отсюда следует, что те параметры, которые должны меняться в процессе выполнения процедуры /то-есть параметрырезультаты/ следует вызывать только по имени. В противном случае мы не сможем воспользоваться полученными результатами.

С другой стороны вызов параметра значением позволяет ускорить работу процедуры в тех случаях, когда фактический параметр является сложным выражением. кроме того, вызов значением "защищает" соответствующий фактический параметр от изменения, когда такое изменение нежелательно. Наконец, возможны случаи, когда вызов значением необходим для правильной работы процедуры.

Например, если имеется процедура

```
procedure ABC(a,b,c);
real a, b, c;
begin ... b: = a+2; ...
end;
```

где параметр выстречается в левой части оператора присваивания, то к такой процедуре нельзя обратиться,

ABC
$$(x,y+z,s)$$
 ($\Pi.10$)

так как при этом в теле процедуры возникнет недопустимая нонструкция

$$y + z = a + 2$$

Однако, если параметр b вызывать значением, то-есть

```
procedure ABC(a,b,c);
value b; real(a,b,c);
begin ... b: = a ↑ 2; ...
end;
```

то обращение /П.10/ возможно.

Теперь переходим к вопросу о глобальных идентификаторах. На этот счет существует правило: глобальные объекты понимаются в том смысле, как они описаны в блоке, где расположено описание процедуры. Например, в программе

begin real a;

procedure F(y); y: = a; a: = 0;
begin real a,x;
a: = 1; F(x); ...

end end

тело процедуры \mathbf{F} ,состоит только из одного оператора \mathbf{y} : = \mathbf{a} , причем \mathbf{a} - глобальная переменная. Процедура вызывается во внутреннем блоке, где также описано \mathbf{a} . Однако, при выполнении $\mathbf{F}(\mathbf{x})$ будет использовано \mathbf{a} , описанное во внешнем блоке /то-есть \mathbf{a} =0/, поскольку описание самой процедуры также помещено во внешнем блоке. Если бы во внешнем блоке описание \mathbf{a} отсутствовало, то программа была бы просто ошибочной. Этот пример весьма важен и его следует хорошо понять.

Нам осталось разобрать два вопроса касающихся процедур:

- процедуры функции; так и катавиаттов о этемарал вод
 - рекурсивные процедуры и обращения.

Процедуры функции лучше всего рассмотреть на премере.
Пусть в программе неоднократно требуется суммировать компоненты различных одномерных массивов. Тогда можно записать
процедуру

Отличие этого описания от тех, с которыми мы встречались ранее, состоит во-первых в том, что перед процедурой указывается ее тип /служебное слово <u>real</u> / и во-вторых, что внутри тела процедуры ее идентификатор SM встречается как самостоятельная переменная в левых частях оператора присваивания.

Процедуры функции можно использовать в арифметических и булевских выражениях как обычные переменные, и в этом состоит определенное удобство. Например,

$$y: = x + 2 \times SM(b,m);$$

 $b: = SM(b,m) > a;$

где b - булевская переменная.

При обработке подобных выражений транслятор, встретив обозначение процедуры функции, автоматически обеспечит ее выполнение.

Простейшими примерами использования процедур функий являются стандартные функции $\sin(x)$, $\exp(x)$ и т.п.

Применительно к процедурам функций специально оговаривается случай, когда при их выполнении изменяются значения каких-либо глобальных величин. Это называется побочным эффектом и многие трансляторы его не разрешают.

Процедуры называются рекурсивными, если в их теле содержится обращение к ним самим. Например [5]

```
integer procedure factorial (n);
if n > 1 then
factorial: = n x factorial (n - 1) else 1;
```

Примером рекурсивного обращения к процедурам может служить

$$x: = \sin (a + \sin(b));$$

Большнство трансляторов запрещают рекурсивные процедуры и обращения. Это не вызывает серьезных затруднений, поскольку случаи, когда действительно нельзя обойтись без рекурсии весьма редки. Мы не будем эдесь касаться собственных объектов процедур /own/, поскольку этот вопрос еще недостаточно разобран в АЛГОЛе; не будем касаться также возможности записать тело процедуры в кодах машины, так как этот вопрос слишком специальный.

IX Пояснения в программах

В АЛГОЛе предусмотрены возможности включать в программу различный поясняющий текст. Это можно делять после служебного слова end , например

... end конец блока;

Текст можно помещать также после знака; и слово \underline{begin} . В этом случае в начале записывается служебное слово $\underline{comment}$ и затем следует текст, например

begin comment приступаем к расчету скоростей;; comment законечен анализ динамики;

Во всех указанных случаях пояснительный текст заканчивается знаком; и не должен содержать внутри себя этого знака.

Пояснительный текст может быть включен также в описание формальных параметров процедуры. Например, описание

procedure S(a,v); begin ... end;

можно выполнить так

<u>procedure</u> S(a) скорость: (v); begin ... end; Нак видно здесь вместо запятой, разделяющей формальные параметры использована конструкция

) пояснительный текст: (

Подобные поястения можно сделать к каждому формальному параметру.

Пояснительный тексй не влияет на выполнение программы и пропускается транслятором.

Х Заключительные замечания

Сначала несколько технических вопросов, Как уже указывалось, отдельные опереторы и описания одного типа отделяются друг от друга точкой с запятой*. Точка с запятой мажет не ставиться в трех случаях: перед словом <u>end</u>, перед словом <u>else</u> и в конце программы.

Все служебные слова в отлечнии от идентификаторов следует подчеркивать; неподчеркнутое слово, напремер, end будет воспринято транслятором как неописанный идентификатор.

Текст по строкам программы можно располагать произвольно; пробелы в АЛГОЛе не принимаются во внимание. Обычно принято граничные слово блока <u>begin</u> и <u>end</u> писать друг под другом и несколько слева от остального текста блока.

Весьма важно следить за изменением глобальных переменных в процедурах и за побочным эффектом в процедурах функциях. Напремер, вследствие побочного эффекта в процедуре функции

 $\frac{\text{real}}{\text{begin } x: = y + 2; c: = \sin(x) \frac{\text{end}}{x};$

^{*} В АЛГОЛе допускаются также "пустые" опереторы, то-есть конструкции в которых между знаками ";", разделяющими опереторы, фактически отсутствует запись оператора.

выражения

- 1) a: = H + c;
- 2) a: = c + H;

будут иметь различные значения /в первом случае $a = 2 \sin y^2$, а во втором $a = c + \sin y^2$ /.

При публикации АЛГОЛьных текстов иногда используются сокращения. Оне касаются главным образом операторов цикла:

полная форма

сокращенная форма

- 1) for i:=1 step b until c do | for i:=step b until c do
- 2) for i:=a step 1 until c do for i:=a until c do
- 3) for i:=1 step 1 until c do for i:=until c do

Иногда используются циклы без параметра

while A do B;

Однако все это не более чем сокращенния форма записи и пользоваться такими сокращениями не обязательно.

Большинство трансляторов вносит те или иные ограничения. Выше уже указывались некоторые из них /например, запрещение рекурсии/. При практическом использовании АЛГОЛьных программ эти ограничения необходимо учитывать. По этой же причине следует избегать излишне сложных конструкций типа /П.7/ или /примеры заимствованы из [5]/

В первом случае имеется связь между формальными параметрами, что может привести к нежелательным эффектам.

Например, простая процедура /она должна поменять значениями ${\bf x}$ и ${\bf y}/$

procedure trans(x,y); integer x,y;
begin integer t; t:=x; x:=y; y:=t end;
перестает работать.

Во втором примере /П.11/ возможна ошибка, если первый фактический параметр вызывается не значением /см.П.10/. В конце концов программа удлинится незначительно, если второй пример в /П.11/ записать как

s: = u+v;E(s,b,c);

однако это гарантирует от неприятностей.

С одной стороны "АЛГОЛ допускает", но с другой неясно как справится транслятор с подобной головоломкой, и даже если справится, то насколько хорошей окажется программа.

В заключение приводим список литературы для более углубленного изучения АЛГОЛа.

- 1. Backus et ell. Report on the Algorithmic Language ALGOL 60, "Numeriche Mathematic", 1960, 2, 2, 106-136.
- 2. Лавров С.С. Универсальный язык программирования, "Наука", 1967.
- 3. Мак-Кракен Д.Д. Программирование на АЛГОЛе, "Мир", 1964.
- 4. Агеев М.И. Основы алгоритмического языка АЛГОЛ 60, ВЦ АН СССР, 1964.
- 5. Брудно А.А. АЛГОЛ, "Наука", 1971.

Во этором примера УП.ПУ врамомиа одибив, воличувезви

Т Пор пуд помедан АПГАЛьных заможен мябрай цейтан дунгот Под с. 25 шикимания. Оне даспотея гладиня порежен оператого поделя

1. Backus et all Report on the Algorithmis.

Language ALGOL 60, "Numeriche Mathematic";

The second to the second second to the secon

ная при токо да выбранти пометорые и пометоры и пометорые и пометоры и пометорые и помето

A AFRES M.M. DONOGE ANTONNECKOTO SEMER ANTILITES TO SEMER ANTILITES TO SEMER ANTILITES TO SEMERAL ANTICIPATION OF THE SEMERAL AND SEMERAL ASSESSMENT OF THE SEMERAL ASSESSMENT

S. EGYAND & A. AMON, WESVAS TO STATE OF THE STATE OF THE

recolor, the rest and rest is assessed to the action of

Manual Andrews in the Company of the Andrews in the Company of the

procedure trensferred integer even

other to have proper as were

СОДЕРЖАНИЕ

Вве	здение	6
		eja, E
	Гл. І Динамические модели	
51	Модели кинематических пар	18
\$2	Модели кинематических цепей	23
\$3	Параметры звеньев	
\$4	Символическое описание механизмов	31
§ 5		35
3 7	Форма записи алгоритмов	40
	Гл. II Геометрия пространственных цепей	
٠.		
\$ 1	"Сборка" разомкнутых цепей	43
\$2	Определение положения разомкнутых цепей	52
₹3	Условия связи для замкнутых цепей	59
\$4	"Сборка" и определение положений для замкнутых	
	цепей	80
	Гл. III Кинематика пространственных цепей	
§ 1	Определение скоростей	92
§2	Определение ускорений	102
		.02
	Гл. IV Статика пространственных цепей	
	ценей	
§ 1	Реакции в кинематических парах разомкнутых цепей	117
\$2	Определение реакций в случае замкнутых цепей	
§ 3	Статика механизмов с мета	123
	Статика механизмов с упругими связями	138

Гл. V Динамика пространственных цепей

§ 1	Силы и моменты сил инерции	5
§ 2	Особенности анализа динамики стержневых	
	механизмов	3
§ 3	Дифференциальные уравнения движения цепей 160)
§4	Моделирование пространственных цепей 17	8
	Гл. VI Взаимосвязанные цепи	
	CONTRACTOR OF THE PROPERTY OF	
§ 1	Геометрия взаимосвязанных цепей	
	/"сборка" и определение положений/ 188	8
§2	Кинематика взаимосвязанных цепей 200	כ
€ 2	Статика взаимосвязанных цепей 210)
§ 4	Динамика и моделирование механизмов с	
	взаимосвязанными цепями	4
При	ложения)
Co	держание	9

SAVREMENA RAČUNSKA TEHNIKA I NJENA PRIMENA

U ovoj seriji Matematičkog instituta dosada su publikovane sledeće knjige:

- Nedeljko Parezanović
 Algoritmi i programski jezik FORTAN IV, Beograd, 1972., str. 272.
- Pavle Pejović i Nedeljko Parezanović
 Analogni elektronski računari i njihova primena Beograd, 1972., str.321.
- 3. Dragiša Stojanović
 Ekonomsko-matematički modeli linearnog programiranja,
 Beograd, 1973., str. 84.
- Jurij Stepanenko
 Dinamika prostornih mehanizama, Beograd, 1974., str.282

U pripremi za štampu:

5. Mirko Stojaković
Algoritmi i automati, Beograd,

Ю. А. Степаненко

динамика пространственных механизмов

Rukopis za snimanje kucala			31			Spomenka Jevill
Korekture izvršio	gwrsiili gwrsiili			ric	1.	Dr M. Vukobratović
Nacrt za korice					•	Milan Čavčić
Tiraž	12	110				500 primeraka
Štampanje završeno						septemb r a 1974.



