

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное агентство по образованию

УНИВЕРСИТЕТ ИТМО

Факультет компьютерных технологий и управления

Кафедра систем управления и информатики

Студент: Артемов Кирилл

группа Р4135

РЕФЕРАТ

Вычислительно эффективные алгоритмы
решения прямой и обратной задач динамики

Преподаватель

_____ С. А. Колюбин

«___» _____ 2016 г.

Санкт-Петербург, 2016 г.

Содержание

1	Введение	2
2	Мотивация	3
3	Общие методы решения задач	5
3.1	Динамические модели	5
3.2	Уравнения движения	6
3.3	Алгоритмы основанные на уравнениях Лагранжа	8
3.4	Алгоритмы основанные на уравнениях Ньютона-Эйлера	19
3.5	Алгоритмы основанные на уравнениях Аппеля	23
3.6	Алгоритмы основанные на уравнениях Кейна	25
3.7	Эффективность алгоритмов	26
4	Работы по алгоритмам	28
5	Открытые проблемы	30
	Литература	31

1 Введение

Первые работы в области робототехнических систем были написаны более двух десятков лет тому назад. Тем не менее, как в прошлом веке, так и сейчас сообщество робототехники сосредоточено на проблеме вычислительной эффективности. Многие из наиболее эффективных алгоритмов использующиеся сейчас в динамике, которые применимы к широкому классу механизмов, были разработаны учеными очень тесно связанными с роботами.

Наряду с тем, что эффективность алгоритмов имеет несомненно важное значение для моделирования и управления все более сложными механизмами, работающими на более высоких скоростях, другие аспекты динамики также играют важную роль. Алгоритмы формулируются в соответствии с уравнениями движения, что способствует более простой разработке и реализации. Кроме того, важно, чтобы алгоритмы могли решать задачи для как можно более общего случая устройств с разной геометрией и структурой.

В этой работе рассмотрены общие методы решения прямой и обратной задач динамики: кратко описаны основные положения математических моделей динамики роботов, уравнения динамики, наиболее эффективные на сегодняшний день алгоритмы расчета ее, сделан обзор литературе по теме, а также существующие нерешенные задачи в области алгоритмов расчета динамики.

2 Мотивация

В принципе, решение задач динамики для роботов не представляет особых трудностей - робототехническое устройство это, чаще всего, система твердых тел и как получить уравнения движения таких систем известно достаточно давно. Но на практике не все так просто и встает вопрос о поиске способа получения динамики робота в реальном времени, что приводит к необходимости использовать вычислительно эффективные алгоритмы.

В простейшей форме, уравнение движение для робототехнического механизма может быть представлено вектором дифференциальных уравнений[1]:

$$\ddot{q} = \phi(q, \dot{q}, \tau)$$

где q - вектор обобщенных координат, \dot{q} и \ddot{q} производные по времени, и τ - вектор сил. Решая такое уравнение относительно q необходимо рассчитать значение функции ϕ и выполнить операции интегрирование. Для систем твердых тел сложных робототехнических механизмов основной проблемой является поиск значения функции ϕ .

Для не сложной системы твердых тел, наиболее простым будет выразить уравнения функции для каждого из параметров, результатом чего станет набор простых функций для каждой из степеней свободы. Далее, вычисляя их, получим числовые значения.

К сожалению, такой подход не работает для более сложных систем, в которых число параметров стремительно растет с увеличением количество тел в системе. Что требует применения специальных методов и алгоритмов. Все динамические алгоритмы для сложных многосвязных тел оценивают уравнения движения поэтапно. На каждом из этапов получаются значения, которые используются на следующем этапе вычислений и так далее.

Динамика робототехнических систем представляет собой отношения между действующими на механизм силами и ускорениями, которые они производят. Как правило, робототехнические механизмы моделируются как система твердых тел, в таком случае динамика такой системы также рассматривается как динамика твердого тела. Существуют две основные задачи динамики:

1. прямая задача – даны силы/моменты, найти ускорения;
2. обратная задача – даны ускорения, найти силы/моменты.

Прямая задача в основном используется для моделирования. Обратная задача более распространена, например для: управления движением и силой робота в режиме реального времени; планирования траекторий и оптимизации; при разработке роботов; а также, как один из элементов алгоритмов прямой задачи.

Также задачами динамики являются:

1. вычисление коэффициентов управлений движением;
2. определение параметров инерции – оценка параметров инерции механизмов робота посредством измерения их динамического поведения;
3. гибридная динамика: даны силы в некоторых сочленениях и ускорения в других, найти неизвестные силы и ускорения.

Для определение всех необходимых значений параметров в режиме реального времени стоит задача вычисления их, тут-то и пригождаются вычислительно эффективные алгоритмы.

3 Общие методы решения задач

3.1 Динамические модели

Формирование эффективных уравнений динамики роботов, которые могут быть рассчитаны на ЭВМ за минимальное время, является одной из важнейших задач в робототехнике. Ее решение необходимо для моделирования динамики манипуляторов в масштабе реального времени, для разработки эффективных алгоритмов управления роботами с учетом динамики [2], для повышения эффективности исследования и разработки манипуляторов.

Одни из первых результатов в этой области принадлежат Кане [3] и Виттенбургу [4]. Полученные ими уравнения справедливы не только для роботов, но и для более широкого класса систем, состоящих из шарнирно связанных твердых тел. В дальнейшем было разработано большое количество алгоритмов формирования динамических уравнений манипуляторов, в которых использовались различные способы описания кинематики, расчета кинематических и динамических величин, а также различные формы уравнений динамики системы тел.

Описание кинематики – это способ задания систем координат, связанных со звеньями манипулятора, и выбора параметров, которые однозначно определяют взаимное положение звеньев и конфигурацию всего манипулятора. В представлении Денавита-Хартенберга [5] начала систем координат расположены в шарнирах, а их оси формируются по правилам, которые определяются кинематикой манипулятора. В другом методе описания кинематики локальные системы координат привязаны к центрам масс звеньев, а их оси направлены вдоль главных осей инерции. Параметры, определяемые относительно таких систем координат удобны для динамического анализа.

Еще одной характеристикой методов математического моделирования манипуляторов является способ расчета кинематических и динамических величин, определяющих математическую модель манипулятора. Для этого используются однородные координаты и матрицы преобразования координат размерности 4×4 , определяющие относительное положение и ориента-

цию звеньев манипулятора [6]; матрицы поворотов размерности 3×3 и вектора относительных перемещений [6]; формулы Родриго [7]; ортогональные тензоры [8]; кватернионы; метод векторных параметров с использованием групп Ли .

Хотя вычислительная эффективность того или иного метода формирования динамических уравнений зависит в первую очередь от особенностей его реализации (использования рекурсивных преобразований, динамических аналогий и др.), можно отметить и существенную роль выбора подходящего способа расчета модели манипулятора. Например, матрицы преобразования однородных координат размерности 4×4 , обладающие универсальностью в кинематическом описании, практически не используются в задачах реального времени из-за больших вычислительных затрат, необходимых для выполнения операций над ними. В то же время, использование матриц поворотов размера 3×3 позволяет получить эффективные алгоритмы расчета кинематики и динамики.

Эффективно использование кватернионов, ортогональных тензоров, однако в ряде задач (например, при управлении в декартовых координатах) предпочтительнее использовать матричные представления.

Среди самых современных методов моделирования динамики манипуляторов можно отметить подходы, основанные на использовании нейронных сетей [9], пространственных операторов [10], групп Ли, методов нечеткой логики [11]. Для описания динамики сложных структур (параллельных роботов, манипуляторов с большой избыточностью степеней подвижности, роботов-гуманоидов) используются методы расчета динамики в операционном пространстве роботов и другие.

3.2 Уравнения движения

Есть две основные формы записи уравнений движения [13]:

1. уравнение в конфигурационном пространстве:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

где q, \dot{q} и \ddot{q} – положение, скорость и ускорение звена робота, τ – силы/моменты приложенные к звену, $M(q)$ – тензор инерции, $C(q, \dot{q})$ – матрица Колиолисовых и центробежных сил, $G(q)$ – вектор гравитации;

2. уравнение в операционном пространстве:

$$\Lambda(x)\ddot{x} + \mu(x, \dot{x}) + \rho(x) = f$$

где x – положение схвата, \dot{x} – скорость схвата, f – действующие силы на схват, $\Lambda(x)$ – матрица инерции в операционном пространстве, $\mu(x, \dot{x})$ – содержит Кориолисовы и центробежные составляющие, $\rho(x)$ – вектор гравитации.

Эти уравнения показывают функциональную зависимость: H – функция от q , Λ функция от x . Однако, эти зависимости часто отбрасывают, для более простого понимания.

Строго говоря, коэффициенты уравнений зависят не только от q, \dot{q} и f_{ext} , но также от динамической модели робототехнической системы. Т.е. описание механизма отдельно для каждой из его частей: звенья, сочленения и их характеризующие их параметры. Динамическая модель состоит из:

- кинематической модели;
- множества параметров инерции.

Для описание инерции твердого тела нужно десять параметров инерции: масса, расположение центра масс, шесть вращательных параметров инерции. Однако, когда тела соединяют друг с другом их степени свободы ограничиваются и некоторые из этих параметров могут не оказывать влияния на поведение системы. Поэтому при составлении динамической модели робототехнического механизма от кинематической модели и измерения ее динамических характеристик, процедура сводится к идентификации значений множества основных параметров инерции, которые поддаются наблюдению в полученной системе.

3.3 Алгоритмы основанные на уравнениях Лагранжа

Существует четыре главных задачи динамики, которые решаются путем реализации описанных ниже алгоритмов.

1. обратная задача динамики, когда требуется вычислить необходимые моменты/силы приводов звеньев для достижения заданной траектории (положение, скорость и ускорение);
2. прямая задача динамики, когда необходимо определить траекторию движения по заданным моментам/силам приводов звеньев;
3. определение матрицы инерции в конфигурационном пространстве;
4. определение матрицы инерции в операционном пространстве.

Обратная задача решается для непосредственного управления роботом. Прямая – для моделирования. Матрицы инерции находятся для анализа и управления.

Далее рассмотрены алгоритмы для решения этих задач. Сначала представлены алгоритмы основанные на уравнениях Эйлера-Лагранжа, после на методе Ньютона-Эйлера, Кейна и Аппеля [14].

Алгоритм Uicker и Kahn

Первая версия алгоритма была разработана в 1965 году J.J. Uicker и в основном использовалась для замкнутых кинематических цепей. В 1969 году M. Kahn доработал его таким образом, что стало возможным использовать его и для разомкнутых кинематических цепей. Также в алгоритм были внесены доработки в 1971 и 1981 годах, тогда N. Orleandea и T. Berea написали программную реализацию этого метода в форме программного пакета для анализа динамики роботов. В дальнейшем различные модификации численных и символьных вычислений производили S.Mahil, M. Renaud, M. Tomas и Tesar, R. Waters, J. Hollerbach и многие другие.

Используя этот алгоритм можно решать как прямую, так и обратную задачи динамики. Также он позволяет вычислять матрицы динамической модели робота: матрицу инерции, матрицу Кориолисовых и центробежных

сил и вектор гравитации. Уравнения динамики для робота с n степенями свободы, соответствующие рассматриваемому алгоритму, представлены ниже.

$$P_i = \sum_{j=i}^n \left(\sum_{k=1}^j \left[\text{tr} \left(\frac{\partial W_j}{\partial q_i} J_j \frac{\partial W_j^T}{\partial q_k} \right) \right] \ddot{q}_k \right. \quad (1)$$

$$\left. + \sum_{k=1}^j \sum_{l=1}^j \left[\text{tr} \left(\frac{\partial W_j}{\partial q_i} J_j \frac{\partial^2 W_j^T}{\partial q_k \partial q_l} \right) \dot{q}_k \dot{q}_l \right] - m_j \vec{q}^T \frac{\partial W_j}{\partial q_i} r_{jo} \right) \quad (2)$$

где P_i – сила/момент привода, W_i – матрица трансформации от базовой к локальной системе координат i -ого звена, J_i – матрица инерции i -ого звена относительно локальной системы координат, m_i – масса звена i , r_{j0} – вектор от центра масс звена i к началу базовой системы координат, выраженный в локальной системе координат звена i , \vec{g} – вектор гравитации.

Матрица W_i выражается, как:

$$W_i = A_1^0 A_2^1 \dots A_i^{i-1} \quad (3)$$

где A_k^{k-1} – матрица трансформации (4×4) между системами координат i и $i-1$.

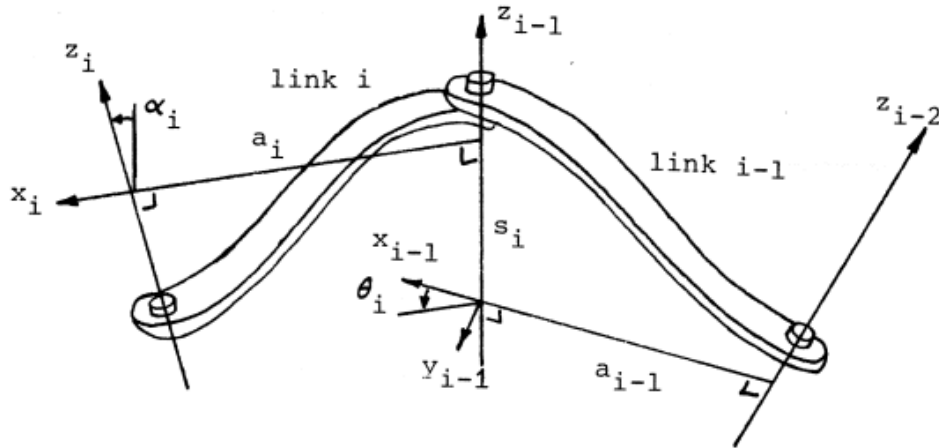


Рис. 1: Оси координат двух соседних звеньев

Применение матриц однородных преобразований позволяет использовать одно уравнение и для призматических сочленений, и для вращательных. Системы координат в роботе выбираются в соответствии с конвенцией

Денавита-Хартенберга. Таким образом, матрица трансформации из $i - 1$ системы координат в i , представляет собой:

$$A_i^{i-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_i \cos(q_i) & \cos(q_i) & -\sin(q_i) \cos(a_i) & \sin(q_i) \sin(a_i) \\ a_i \sin(q_i) & \sin(q_i) & \cos(q_i) \cos(a_i) & -\cos(q_i) \sin(a_i) \\ s_i & 0 & \sin(a_i) & \cos(a_i) \end{bmatrix} \quad (4)$$

Оси локальной системы координат звена i представляются единичными векторами $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$. Эти системы координат зависят только от геометрических параметров системы, следовательно не совпадают с главными осями инерции. Таким образом, вместо трех моментов инерции, получается тензор инерции. Uicker использовал тензор инерции J_i размерности 4×4 , который включает в себя и моменты инерции, и массы звеньев:

$$J_i = \begin{bmatrix} m_i & m_i x_{i0} & m_i y_{i0} & m_i z_{i0} \\ m_i x_{i0} & \frac{1}{2}(-I_{xx} + I_{yy} + I_{zz}) & I_{xy} & I_{xz} \\ m_i y_{i0} & I_{xy} & \frac{1}{2}(I_{xx} - I_{yy} + I_{zz}) & I_{yx} \\ m_i z_{i0} & I_{xz} & I_{yz} & \frac{1}{2}(I_{xx} + I_{yy} - I_{zz}) \end{bmatrix} \quad (5)$$

где x_{i0}, y_{i0}, z_{i0} – координаты центра масс звена i выраженные в базовой системе координат, I_{xx}, I_{xy}, \dots – моменты инерции вокруг соответствующих осей.

Матрица инерции J_i описывает распределение массы в звене i , а также зависит от положения центра масс относительно закрепленной системы координат.

Из матрицы трансформации A_i^{i-1} мы видим, что ее частная производная относительно q_i может быть определена аналитически. Тоже самое справедливо и для матрицы W_i . В этом случае нет рекурсивных отношений, поэтому метод Uicker принадлежит к классу нерекурсивных методов.

Теперь, когда мы выяснили общую структуру метода, детальнее рассмотрим как решаются прямая и обратная задачи динамики.

Обозначим \dot{q}_i и \ddot{q}_i , как скорость и ускорение и запишем уравнение в

следующей форме:

$$P_i = \sum_{j=1}^n H_{ij}(q) \ddot{q}_j + \sum_{k=1}^n \sum_{l=1}^n C_{kl}^i(q) \dot{q}_k \dot{q}_l + g_i(q) \quad (6)$$

где H_{ij} , C_{kl}^i и g_i – скалярные функции зависящие от вектора обобщенных координат $q = [q_1, \dots, q_n]^T$. Что бы доказать это, заметим, что в соответствии с уравнением 1, эти функции зависят от частных производных матриц W_i . В свою очередь матрицы W_i это функции обобщенных координат, из чего следует, что частные производные матриц W_i также зависят только от обобщенных координат. Теперь, представим модель 6 в матричной форме:

$$P = H(q) \ddot{q} + \dot{q}^T C(q) \dot{q} + g(q) \quad (7)$$

где $P = [P_1, \dots, P_n]^T$ – вектор управляющих сил/моментов, $H(q)$ – матрица $n \times n$ инерции системы, $C(q)$ – матрица Кориолисовых и центробежных сил, $g(q)$ – n -вектор соответствующий действию гравитации на систему.

В модели 7 $\dot{q}^T C(q) \dot{q}$ это n -вектор:

$$\dot{q}^T C(q) \dot{q} = \begin{bmatrix} \dot{q}^T C^1(q) \dot{q} \\ \vdots \\ \dot{q}^T C^n(q) \dot{q} \end{bmatrix} \quad (8)$$

где $C^i(q)$ матрица $n \times n$, элементы которой соответствуют $C_{kl}^i(q)$ в уравнении 6.

Итак, имея уравнение 7 мы можем решить обратную задачу динамики, т.е. по заданным (q, \dot{q}, \ddot{q}) определить управляющие силы/моменты.

Чтобы решить прямую задачу, нужно определить $q(t)$ по известным моментам $P(t)$ для $t > t_0$ и начальными условиями $q(t_0)$ и $\dot{q}(t_0)$. Для это представим уравнение 7 в следующей форме:

$$P = H(q) \ddot{q} + h(q, \dot{q}) \quad (9)$$

где $h(q, \dot{q}) = \dot{q}^T C(q) \dot{q} + g(q)$.

Отсюда, выражая \ddot{q} получим:

$$\ddot{q} = H(q)^{-1} (P - h(q, \dot{q})) \quad (10)$$

$q(t)$ находится интегрированием полученного выражения.

Вычислительная сложность рассмотренного метода составляет $O(n^4)$. Такой алгоритм не подходит для расчета динамики современных роботов в реальном времени. Уравнение 1 на 16-битном микропроцессоре вычисляется за 500 мс, для манипулятора с 3-мя степенями свободы и более 5 секунд для 6-степенного. Его можно несколько ускорить пренебрегая Кориолисовыми и центробежными скоростями на медленных режимах работы [14].

Алгоритмы S. Mahil, S. Megahed и M. Renaud

Эти алгоритмы представляют собой модификации Uicker-Kahn алгоритма, в которых авторы избавились от частной производной в уравнении и сократили количество арифметических операций.

В 1979 S.Mahil предложил заменить матрицы трансформации Денавита-Хертенберга формулой поворота Родрига. Чтобы описать суть этой формулы, представим вращение i звена относительно $i - 1$ на угол q_i . Введем вектор \vec{r}_i прикрепленный к i звену перед началом вращения. После вращения, в соответствии с формулой Родрига, этот вектор можно выразить, как:

$$\vec{r}_i = \vec{r}_i \cos(q_i) + (1 - \cos(q_i))(\vec{e}_i \times \vec{r}_i)\vec{e}_i + \vec{e}_i \times \vec{r}_i \sin(q_i) \quad (11)$$

где \vec{e}_i – единичный вектор на оси прикрепленной к сочленению i .

Применим эту формулу для выражения матрицы трансформации A_i^{i-1} , которая описывает вращение i системы координат относительно $i - 1$. Заметим, что векторы $\vec{q}_{ij}, j = 1, 2, 3$ – единичные векторы осей i системы. Тогда, матрица A_i^{i-1} :

$$A_i^{i-1} = \begin{bmatrix} \vec{q}_{i1} & \vec{q}_{i2} & \vec{q}_{i3} \end{bmatrix} \quad (12)$$

Теперь, выразим вектор \vec{q}_{ij} , через формулу Родрига:

$$\vec{q}_{ij} = \vec{q}_{ij}^0 \cos(q_i) + (1 - \cos(q_i))(\vec{e}_i \times \vec{q}_{ij}^0) + \vec{e}_i \times \vec{q}_{ij}^0 \sin(q_i) \quad (13)$$

где \vec{q}_{ij}^0 – вектор оси j i -ой системы координат перед вращением.

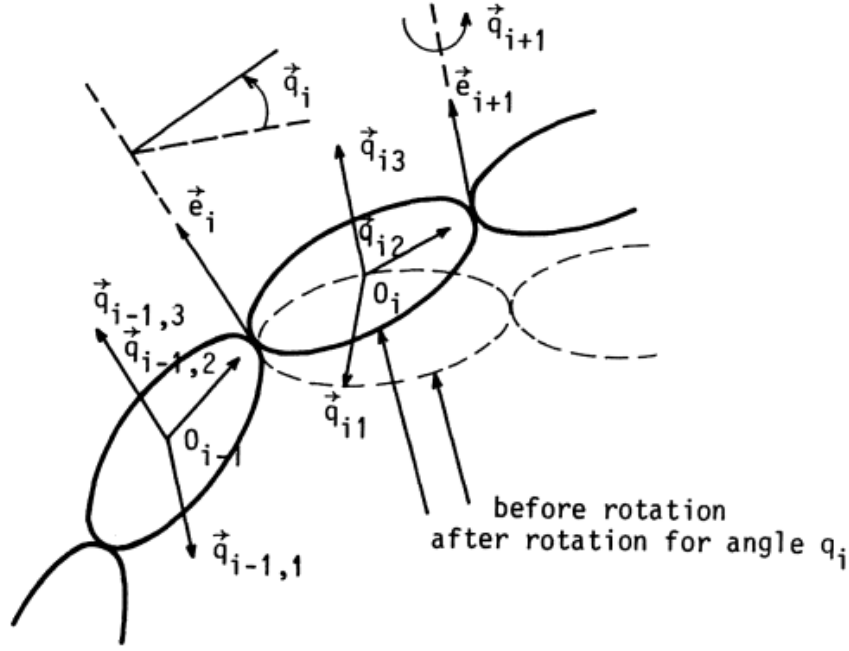


Рис. 2: Вращение звена i

В отличие от кинематики Денавита-Хартенберга, формула Родрига применяется к локальным системам координат, которые прикреплены к звеньям в произвольных местах. Наиболее удачным расположением обычно является центр масс звена, тогда системы координат вращения и инерции совпадают, что упрощает уравнение динамики, а также тензор инерции состоит только из главных моментов инерции звена.

В этом алгоритме фигурирует только два момента инерции – продольный и поперечный. Такое приближение возможно только для цилиндрических звеньев, длина которых много больше их диаметра. Это накладывает ограничения на применение этого алгоритма.

Алгоритм выводится из уравнения Лагранжа в матричной форме:

$$P = \sum_{i=1}^n (H^i(q)\ddot{q} + D^i(q, \dot{q})\dot{q} + g^i(q)) \quad (14)$$

где P – вектор прикладываемых моментов/сил, $H^i(q)$, $D^i(q, \dot{q})$ – $n \times n$ матрицы и $g^i(q)$ – n -вектор.

Элементы матриц в уравнении 14 могут быть выражены с использованием осей сочленений, локальных систем координат и векторов переноса, которые получаются применением формулы Родрига. Для вращательных

сочленений:

$$H_{jk}^i = (Jn_i + m_i(\vec{r}_{ji} \times \vec{r}_{ki}))(\vec{e}_j \times \vec{e}_k) + Js_i(\vec{e}_j \times \vec{q}_{i1})(\vec{e}_k \times \vec{q}_{i1}) - (\vec{e}_j \times \vec{q}_{ki})(\vec{e}_k \times \vec{q}_{ki})$$

где $(j \leq k \leq i)$, Jn_i – поперечный момент инерции звена i , Js_i – продольный момент инерции звена i , m_i – масса звена i , \vec{r}_{ij} – вектор от j -ого сочленения к центру масс i -ого звена, в инерциальной системе.

Элементы матриц $D^i(q, \dot{q})$ выражаются в следующей форме:

$$D_{kl}^i = \sum_{j=p+1}^i \frac{\partial H_{kl}^i}{\partial q_j} \dot{q}_j - \frac{1}{2} \sum_{j=1}^q \frac{\partial H_{jl}^i}{\partial q_k} \dot{q}_j \quad (15)$$

где p и q зависят от k и l . Это выражение включает частную производную от матриц и должны быть преобразованы к более простой для вычисления форме. Одно из уравнения, для вращательных сочленений:

$$\begin{aligned} \frac{\partial H_{kl}^i}{\partial q_j} = & (Jn_i + m_i \vec{r}_{ki} \times \vec{r}_{li})(\vec{r}_k \times \vec{r}_j \times \vec{r}_l) + \\ & + m_i[\vec{r}_k \times \vec{r}_j(\vec{R}_{kj} \times \vec{e}_j \times \vec{r}_{li}) - \vec{e}_l \times \vec{r}_{ki}(\vec{R}_{kj} \times \vec{e}_j \times \vec{e}_l) - \\ & - \vec{e}_k \times \vec{r}_{li}(\vec{R}_{kj} \times \vec{e}_j \times \vec{r}_{li})] + Js_i \vec{e}_l \times \vec{q}_{ik}(\vec{e}_k \times \vec{e}_j \vec{q}_{i1}) \end{aligned}$$

где \vec{R}_{kj} – вектор от центра масс звена j к центру масс звена i .

Выражение в матричной форме связывает элементы матриц Кориолисовых и центробежных сил с элементами матрицы инерции. Тем не менее, это выражение получается достаточно сложным.

Этот алгоритм принадлежит к классу нерекурсивных, основанных на уравнениях Лагранжа. Сравнивая его с алгоритмом Uicker-Kahn, можно увидеть, что количество арифметических операций в этих алгоритмах имеют одинаковый порядок. Из этого следует, что его также практически невозможно реализовать для вычисления динамики в реальном времени. Тем не менее, основное достоинство этого алгоритма заключается в том, что он позволяет понять смысл динамических параметров в элементах матриц модели.

В 1981 году М. Renaud предложил эффективную процедуру вычисления параметром матриц частных производных, которые упоминались в

алгоритме Uicker-Kahn. Алгоритм основывается на следующем свойстве матриц Денавита-Хартенберга [16]:

$$\frac{\partial A_k^{k-1}}{\partial q_k} = Q A_k^{k-1} \quad (16)$$

где, если звено вращательное:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (17)$$

если звено призматическое:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

Основываясь на отношении 16:

$$\frac{\partial W_j}{\partial q_i} = \begin{bmatrix} 0 \\ \omega_i \end{bmatrix} W_j, (i \leq j) \quad (19)$$

$$\frac{\partial W_j}{\partial q_i \partial q_k} = \begin{bmatrix} 0 \\ \omega_i \end{bmatrix} \begin{bmatrix} 0 \\ \omega_k \end{bmatrix} W_j, (i \leq k \leq j) \quad (20)$$

где ω_i – матрица 3×3 , элементы которой получены непосредственно из матрицы W_j .

Этот алгоритм также, принадлежит к классу нерекурсивных, основанных на уравнениях Лагранжа. Тем не менее, количество численных операций по прежнему очень велико, что делает невозможным его применение для расчета в режиме реального времени.

Также, стоит отметить, что в 1983 году М. Renaud предложил еще одну модификацию предыдущего алгоритма, вводя тензорное исчисление, а также ряд рекурсивных соотношений для вычисления динамической модели матриц. Но и эта модификация не привела к значительному сокращению числа операций. Однако, автор показал [17], что полученная аналитическая

модель путем применения его процедур эффективна при вычислении. Но, к сожалению, порядок формирования модели не автоматизирован и должен осуществляться вручную. Это процедура достаточно сложна и не позволяет избежать ошибок. Таким образом, этот алгоритм также не может рассматриваться для вычислений в реальном времени.

Алгоритм Vukobratovic-Potconjak

В этом алгоритме, для описания расположения звеньев в пространстве, также как и предыдущем, используется формула Родрига. Такой подход облегчает динамический анализ механизмов, так как используются только главные моменты инерции.

Кинематическая часть этого алгоритма представляется выражениями:

$$\omega_i = N(i)\dot{q} \quad (21)$$

$$\dot{v}_i = M(i)\dot{q} \quad (22)$$

где ω_i – вектор угловых скоростей звена i выраженный в i -ой локальной системе координат, v_i – вектор линейных скоростей центра масс i -ого звена относительно i -ой локальной системы координат, $N(i), M(i)$ – $3 \times n$ матрицы, зависящие от обобщенных координат, \dot{q} – $n \times 1$ вектор скоростей.

Подставляя уравнения 21 и 22 в кинетическую энергию системы, получим:

$$E_k = \sum_{i=1}^n \left(\frac{1}{2} m_i v_i^T v_i + \frac{1}{2} \omega_i^T J_i \omega_i \right) \quad (23)$$

и, используя матричную форму представления уравнения Лагранжа:

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{q}} \right) - \frac{\partial E_k}{\partial q} + \frac{E_p}{\partial q} = P \quad (24)$$

получим динамическую модель робота:

$$P = H(q)\ddot{q} + h(q, \dot{q}) \quad (25)$$

В выражении выше m_i обозначает массу i звена, J_i – диагональную матрицу 3×3 с главными моментами инерции, E_k и E_p – кинетическую и потенциальную энергии всей системы, P – вектор управляющих моментов/сил.

Матрицы модели задаются следующими выражениями:

$$H(q) = \sum_{i=1}^n (m_i M(i)^T M(i) + N(i)^T J_i N(i)) \quad (26)$$

$$h(q, \dot{q}) = \frac{\partial(E_p - E_k)}{\partial q} + \dot{H}(q)\dot{q} \quad (27)$$

Частные производные в последнем выражении это матрица инерции, $H(q)$, которая вычисляется из рекурсивных отношений, которые не представлены здесь по причине их внушительной сложности.

Количество арифметических операций в это алгоритме в несколько раз меньше, чем в алгоритме Uicker-Kahn.

Алгоритмы R. Waters и J. Hollerbach

Эти алгоритмы разрабатывались для решения обратной задачи динамики, как частного случая алгоритма Uicker-Kahn. Алгоритм не позволяет вычислять матрицу инерции системы, так как вторая производная обобщенных координат фигурирует в уравнении не явно. Именно поэтому эти алгоритмы не эквивалентны.

В соответствии с алгоритмом R. Waters, управляющие силы/моменты выражаются следующим уравнением:

$$P_i = \sum_{j=i}^n \left[tr\left(\frac{\partial W_j}{\partial q_i} J_j \ddot{W}_j^T\right) - m_j \ddot{g}^T \frac{\partial W_j}{\partial q_i} r_{i0} \right] \quad (28)$$

со следующими рекурсивными отношениями:

$$W_j = W_{j-1} A_j^{j-1} \quad (29)$$

$$\dot{W}_j = \dot{W}_{j-1} A_j^{j-1} + W_{j-1} \frac{\partial A_j^{j-1}}{\partial q_j} \dot{q}_j \quad (30)$$

$$\ddot{W}_j = \ddot{W}_{j-1}A_j^{j-1} + 2\dot{W}_{j-1}\frac{\partial A_j^{j-1}}{\partial q_j}\dot{q}_j + W_{j-1}\frac{\partial^2 A_j^{j-1}}{\partial q_j^2}\dot{q}_j^2 + W_{j-1}\frac{\partial A_j^{j-1}}{\partial q_j}\ddot{q}_j \quad (31)$$

Эти рекурсивные соотношения уменьшают количество арифметических операций, требующихся на вычисление управляющих сил/моментов до n^2 . Уменьшение количества операций по сравнению с алгоритмом Uicker-Kahn, связано с тем, что в рассматриваемом алгоритме матрица инерции системы не вычисляется явно, что не требует вычислять частные производные $\frac{\partial W_j}{\partial q_k \partial q_l}$. Однако, несмотря на внушительное уменьшение количество вычислений, этот алгоритм также не позволяет применять его для вычисления динамики в реальном времени.

Еще более значительное сокращение количества сложений и умножений в уравнении было предложено J. Hollerbach, который заметил, что частная производная $\frac{\partial W_j}{\partial q_i}$ может быть выражена как $(\frac{\partial W_j}{\partial q_i})W_j^i$, где

$$W_j^i = A_{i+1}^i A_{i+2}^{i+1} \dots A_{j-1}^j \quad (32)$$

матрица трансформации из i -ой в j локальную систему координат. Подставляя полученные выражения в уравнение 28, получим:

$$P_i = \text{tr}(\frac{\partial W_i}{\partial q_i})D_i - \vec{g}^T \frac{\partial W_i}{\partial q_i} c_i \quad (33)$$

где

$$D_i = J_i \ddot{W}_i^T + A_{i+1}^i D_{i+1}, \quad (34)$$

$$c_i = m_i r_{i0} + A_{i+1}^i c_{i+1}. \quad (35)$$

Угловое ускорение \ddot{W}_i^T вычисляется в прямой рекурсии от базы к схвату как и в алгоритме Waters. D_i и c_i вычисляются в обратной рекурсии от схвата к базе.

Таим образом, полученные соотношения позволяют достичь $30n - 592$ операций умножения и $675n - 464$ операций сложения.

Несколько позже, Hollerbach еще более уменьшил количество операций применив вместо матриц трансформации 4×4 матрицы поворота 3×3

и векторы переноса. С этими изменениями количество операций, необходимых для вычисления моментов/сил стало: умножений $412n - 277$, сложений $320n - 201$.

3.4 Алгоритмы основанные на уравнениях Ньютона-Эйлера

В этой части представлены алгоритмы основанные на уравнениях Ньютона-Эйлера. Их чаще используют для управления роботами.

Из уравнений Ньютона-Эйлера определяются силы/моменты действующие на звенья. Если принять за F_i силу действующую на центр масс звена i , выраженную в локальной системе координат этого звена, а за m_i массу i звена, то второй закон Ньютона запишется, как:

$$F_i = m_i w_i \quad (36)$$

где w_i ускорение центра масс звена i . Уравнение Эйлера устанавливает взаимосвязь между моментами/силами и моментами инерции:

$$M_i = I_i \epsilon_i + \omega_i \times (I_i \omega_i) \quad (37)$$

где I_i – матрица 3×3 с главными моментами инерции на диагонали, ϵ_i – угловое ускорение звена i , ω_i – угловая скорость звена i .

Все векторы выражены в локальной системе координат i -ого звена, прикрепленной к центру масс. Оси локальной системы координат совпадают с осями инерции звена.

Выражая уравнения в неподвижной инерциальной системе координат уравнения имеют ту же форму, а матрица инерции находится по формуле: $J_i = A_i I_i A_i^T$. Матрица трансформации A_i из локальной системы координат i -ого звена в зафиксированную систему. J_i матрица инерции i звена относительно базовой системы координат.

Далее рассмотрены алгоритмы Ньютона-Эйлера в неподвижной инерциальной и локальной системах координат.

Алгоритм Vukobratovic-Stepanenko

Уравнения динамики Ньютона-Эйлера впервые были применены к моделированию в 1973 году М. Vukobratovic и J. Stepanenko. Тогда была введена концепция рекурсивного решения прямой и обратной задач динамики. Этот алгоритм также упоминается как кинестатический, основанный на принципе Д'Аламбера.

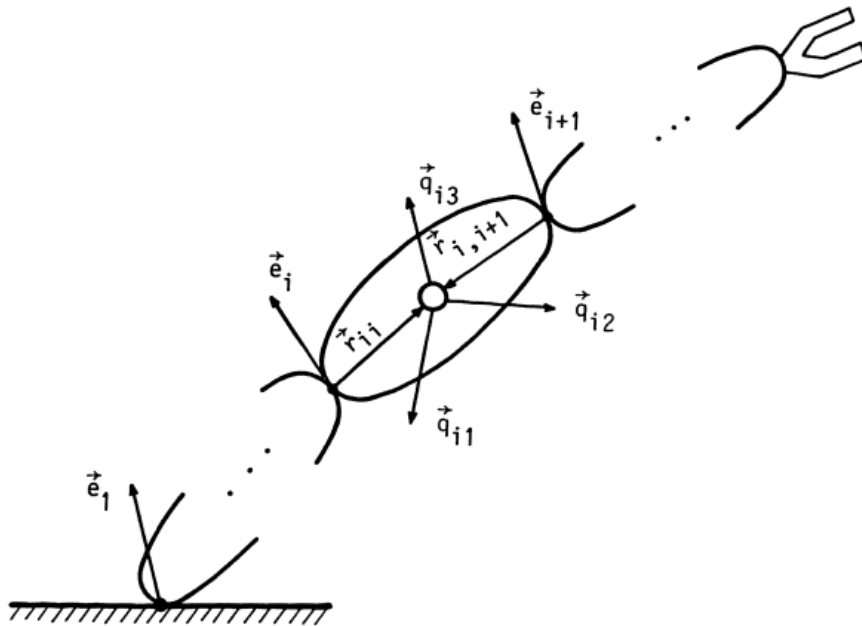


Рис. 3: Звено манипулятора

На рисунке 3 представлено звено с необходимыми обозначениями для дальнейших выкладок. \vec{q}_{ij} , ($j = 1, 2, 3$) – единичный вектор оси j локальной системы координат звена i , \vec{r}_{ii} – вектор от i сочленения до центра масс i звена, $\vec{r}_{i,i+1}$ – вектор от $i + 1$ сочленения до центра масс i -ого звена, \vec{e}_i – единичный вектор i сочленения, $i = 1, \dots, n$

Угловая скорость $\vec{\omega}_i$, угловое и линейное ускорения $\vec{\epsilon}_i$, \vec{w}_i находятся из следующих уравнений:

$$\vec{\omega}_i = \vec{\omega}_{i-1} + \dot{q}_i \vec{e}_i \quad (38)$$

$$\vec{\epsilon}_i = \sum_{j=1}^i \vec{a}_{ij} \ddot{q}_j + \vec{a}_i^0 \quad (39)$$

$$\vec{w}_i = \sum_{j=1}^i \vec{\beta}_i \ddot{q}_j + \vec{\beta}_i^0 \quad (40)$$

где коэффициенты $\vec{a}_{ij}, \vec{a}_i^0, \vec{\beta}_i^0$ получаются из соответствующих рекурсивных соотношений. Например \vec{a}_i^0 :

$$\vec{a}_i^0 = \vec{a}_{i-1}^0 + \dot{q}_i \vec{\omega}_{i-1} \times \vec{e}_i \quad (41)$$

Заметим два существенных свойства объединяющие эти отношения:

1. свойства рекурсивности;
2. ускорения \ddot{q}_i в обоих уравнениях, что позволяет находить матрицы инерции и решать прямую задачу динамики.

Инерционная сила и момент могут быть выражены, как:

$$\vec{F}_i = \sum_{j=1}^i \vec{a}_{ij} \ddot{q}_j + \vec{a}_i^0 \quad (42)$$

$$\vec{M}_i = \sum_{j=1}^i \vec{b}_{ij} \ddot{q}_j + \vec{b}_i^0 \quad (43)$$

где b_{ij} и b_i^0 получаются из уравнений динамики Эйлера и выражают только моменты инерции, но не тензор инерции. Уравнение движения в матричной форме:

$$P = H(q)\ddot{q} + h(q, \dot{q}) \quad (44)$$

где $H(q)$ – матрица инерции $n \times n$:

$$H_{ik} = -\vec{e}_i \sum_{j=\max(i,k)}^n (\vec{b}_{jk} + \vec{r}_{ji} \times \vec{a}_{jk}), (i, k = 1, \dots, n) \quad (45)$$

и $h(q, \dot{q})$ вектор с n элементами:

$$h_i = -\vec{e}_i \sum_{j=1}^n (\vec{r}_{ji} \times (\vec{a}_k^0 + \vec{G}_j) + \vec{b}_j^0) \quad (46)$$

где \vec{G}_j вектор гравитации, действующий на звено j . Все уравнения записаны для вращательных звеньев, но могут быть легко приведены к форме

для призматических путем обнуления тех членов уравнений, которые представляют угловые скорости и ускорения.

Для рассмотренного алгоритма, количество умножений равно $\frac{3}{4}n^3 + 28n^2 + \frac{525}{2}n$, а для операций сложения $\frac{4}{3}n^3 + 20n^2 + \frac{530}{2}n$.

Для решения только обратной задачи кинематики этот алгоритм можно представить в несколько иной форме. Тогда, кинематика представляется следующими тремя рекурсивными выражениями:

$$\omega_i = \omega_{i-1} + \dot{q}_i e_i, \quad (47)$$

$$\epsilon = \epsilon_{i-1} + \ddot{q}_i e_i + \dot{q}_i (\omega_{i-1} \times e_i), \quad (48)$$

$$\begin{aligned} w_i = w_{i-1} - \epsilon_{i-1} \times r_{i-1,i} + \epsilon_i \times r_{ii} - \omega_{i-1} \times (\omega_{i-1} \times r_{i-1,i}) \\ + \omega_i \times (\omega_i \times r_{ii}) \end{aligned}$$

а динамика уравнениями Ньютона-Эйлера:

$$F_i = m_i w_i \quad (49)$$

$$M_i = J_i \epsilon_i + \omega_i \times (J_i \omega_i) \quad (50)$$

где $J_i = A_i I_i A_i^T$ матрица 3×3 инерции звена i относительно базовой системы координат.

Обозначив R_i и M_i силу и момент, с которыми звено i действует на звено $i - 1$, получим следующие рекурсивные отношения.

$$R_i = R_{i+1} - F_i - G_i \quad (51)$$

$$M_i = M_{i+1} + (r_{ii} - r_{i,i+1}) \times R_{i+1} + r_{ii} \times (F_i + G_i) - M_i \quad (52)$$

где R_{n+1}, M_{n+1} сила и момент действующие на n -ое звено (схват). Эти соотношения вычисляются в обратной рекурсии от схвата к базе $i = n, \dots, 1$.

Управляющие моменты вычисляются, как:

$$P_i = M_i e_i \quad (53)$$

а силы, для призматических звеньев:

$$F_i = R_i e_i \quad (54)$$

Количество операций умножения и сложения равны $225n$ и $152n$, соответственно.

Также этот алгоритм был усовершенствован J. Luh и M. Walker, для чего все кинематические и динамические переменные были выражены в локальных системах координат звеньев. Это привело к еще большему сокращению численных операций для вычисления динамики: $150n - 48$ умножений и $131n - 48$ сложений.

Необходимо отметить, что этот алгоритм решает только обратную задачу динамики.

Одним из основных достоинств этого алгоритма является простота описания сложных сочленений, имеющих 6-ть степеней свободы. Однако это достоинство одновременно является и недостатком, так как усложняет реализацию алгоритма по отношению к простым механизмам.

Что касается количества арифметических операций, то их значительно больше, по сравнению с описанными ранее рекурсивными алгоритмами основанными на уравнениях Ньютона-Эйлера, так как алгоритм нерекурсивный.

3.5 Алгоритмы основанные на уравнениях Аппеля

Уравнения Аппеля основаны на функции Гиббса "энергии ускорения". Кинематика описана таким же образом как и в алгоритмах на базе Ньютона-Эйлера.

Относительное расположение соседних двух звеньев описывается формулой Родрига. Матрицы трансформации A_i^{i-1} между системами координат звена $i-1$ и локальной системы координат звена i получается таким же способом, как и в методе Ньютона-Эйлера. Отношения для кинетических параметров, также аналогичны:

$$\omega_i = A_i^{i-1} \omega_{i-1} + \dot{q}_i e_i \quad (55)$$

$$\epsilon_i = A_i^{i-1} \epsilon_{i-1} + \ddot{q}_i e_i + \dot{q}_i (\omega_i \times e_i) \quad (56)$$

$$w_i = A_i^{i-1} (w_{i-1} - \epsilon_{i-1} \times r_{i-1,i} - \omega_{i-1} \times (\omega_{i-1} \times r_{i-1,i})) + \epsilon_i \times r_{ii} + \omega_i \times (\omega_i \times r_{ii})$$

где e_i – единичный вектор на оси i -ого сочленения выраженный в локальной системе координат звена i , r_{ij} – вектор от сочленения j до центра масс звена i , выраженный в локальной системе координат звена i .

Динамика механизма описывается уравнениями Гиббса-Аппеля:

$$\frac{\partial G}{\partial \ddot{q}_i} = Q_i, (i = 1, \dots, n) \quad (57)$$

где G – функция "энергии ускорения" Q_i – обобщенная сила.

Функция "энергии ускорения" может быть выражена как сумма:

$$G = \sum_{i=1}^n G_i \quad (58)$$

где G_i – функция Гиббса для звена i , которая, в свою очередь, определится из выражения:

$$G_i = \frac{1}{2} m_i w_i^2 + \frac{1}{2} \epsilon_i^T J_i \epsilon_i + 2(\omega_i \times J_i \omega_i) \epsilon_i \quad (59)$$

Подставляя в последнее выражения рекурсивные отношения для кинематики робота, получаем динамическую модель робота в форме:

$$H(q) \ddot{q} + h_c(q, \dot{q}) = Q \quad (60)$$

где $Q = [Q_1, \dots, Q_n]$ – вектор обобщенных сил. Вектор обобщенных сил это разница, между управляющими силами/моментами и вектором описывающим действие гравитации на систему $g(q)$:

$$Q = P - g(q) \quad (61)$$

Подставляя полученное уравнение в уравнение 60, получим:

$$H(q) \ddot{q} + h(q, \dot{q}) = P \quad (62)$$

где $h(q, \dot{q}) = h_c(q, \dot{q}) + g(p)$.

Этот алгоритм позволяет решать как прямую, так и обратную задачи динамики. Вычислительная сложность: $\frac{7}{3}n^3 + 27n^2 + \frac{722}{3}n + 9$ – умножений, $\frac{10}{3}n^3 + \frac{43}{2}n^2 + \frac{931}{6}n + 6$ – сложений.

3.6 Алгоритмы основанные на уравнениях Кейна

Алгоритм основывается на динамических уравнениях Капе. Эти уравнения выражаются из основных теорем механики. Алгоритм Капе напрямую зависит от динамических уравнения Ньютона-Эйлера. Динамические уравнения Капе не ограничиваются лишь робототехническими системами, но также применяются и к более широкому классу механизмов, в том числе, например, они используются в космических аппаратах с несколькими соединенными твердыми телами.

Считается, что робот состоит из n соединенных твердых тел, каждое из которых имеет по 6 степеней свободы. Относительная ориентация соседних звеньев описывается следующими четырьмя параметрами Эйлера:

$$\begin{aligned}\epsilon_{il} &= e_{il} \sin\left(\frac{q_i}{2}\right), (l = 1, 2, 3) \\ \epsilon_{i4} &= e_{i4} \cos\left(\frac{q_i}{2}\right).\end{aligned}$$

где $e_i = (e_{i1}, e_{i2}, e_{i3})$ – единичный вектор на оси, вокруг которой i – 1 локальная система координат переходит в i в результате вращения, q_i – угол, $e_{i4} = 1$. Можно показать, что параметры Эйлера эквивалентны формуле Родрига, когда сочленения с одной степенью свободы связаны.

$$r' = r + 2\sin\left(\frac{q_i}{2}\right)e_i \times (e_i \times r) + 2\sin\left(\frac{q_i}{2}\right)\cos\left(\frac{q_i}{2}\right)(e_i \times r) \quad (63)$$

где r – вектор до вращения, r' – вектор после вращения на угол q_i вокруг оси e_i . Сделав замену $\epsilon_i = e_i \sin\left(\frac{q_i}{2}\right)$ и $\epsilon_{i4} = \cos\left(\frac{q_i}{2}\right)$, получим:

$$r' = r + 2\epsilon_i \times (e_i \times r) + 2\sin\left(\frac{q_i}{2}\right)\epsilon_{i4}(e_i \times r) \quad (64)$$

Кинематические соотношения в этом алгоритме состоят из выражения

для скоростей и ускорений звеньев в терминах параметров Эйлера:

$$\omega_i = \sum_{k=1}^i \omega'_k \quad (65)$$

где ω_i – угловая скорость звена i относительно базовой системы координат, ω'_k – относительная угловая скорость звена k относительно звена $k-1$. Это соотношение можно переписать в рекурсивной форме:

$$\omega_i = \omega_{i-1} + \omega'_i \quad (66)$$

Несложно заметить, что это уравнение совпадает с ранее рассмотренным. Таким образом, единственное различие заключается в том, что для вычисления кинематических и динамических параметров уравнений используется нерекурсивные соотношения.

3.7 Эффективность алгоритмов

В таблице 1 представлены результаты анализа методов описания динамики манипуляторов по форме уравнений, вычислительной эффективности (для манипуляторов с шестью степенями свободы); также отражено, обеспечивает ли данный алгоритм замкнутость уравнений и возможность решения прямой задачи динамики.

Как показывает анализ таблицы, многие эффективные в вычислительном плане методы не обеспечивают замкнутости системы уравнений динамики, что ограничивает их применение в задачах управления, а также при анализе влияния различных динамических коэффициентов на движение манипулятора.

В последние годы для повышения эффективности уравнений динамики широко применяются символьные преобразования и алгоритмы распараллеливания вычислений. Применение символьных преобразований, способствует уравниванию различных алгоритмов по вычислительной эффективности. Поэтому важными критериями оценки алгоритма становятся хорошая алгоритмируемость (удобство программирования), замкнутость

уравнений, возможность применения символьных преобразований и алгоритмов распараллеливания.

Кроме того, можно отметить, что в работах ряда авторов указывается, что во многих случаях (например в задачах управления роботами) наиболее подходящим способом описания динамики являются уравнения Лагранжа. Это отмечено и в работе [18], где утверждается, что при реализации динамических алгоритмов на параллельных процессорах зависимость данных в методах, основанных на уравнениях Ньютона-Эйлера, намного сильнее, чем при использовании уравнений Лагранжа.

Таблица 1: Алгоритмы

Форма уравнений	Авторы алгоритма	Число операций		Замкнутость уравнений	Прямая задача
		*	+		
Лагранж	Uicker/Kahn	66271	51548	+	+
	Vukobratovic/Potconjak	37189	5652	-	+
	Hollerbach 3x3	2195	1719	-	-
	Renaud	992	776	-	+
	Li	951	842	-	+
Ньютон-Эйлер	Vukobratovic/Stepanenko	2907	2068	+	+
	Walker/Orin	1771	1345	-	+
	Wang/Ravani	1659	1252	-	+
	Luh/Walker/Paul	792	662	-	-
	Balafoutis/Patel/Misra	489	420	-	-
Д'Аламбер	Lee/Nigam	2963	2209	+	+
Аппель	Попов	2929	2500	+	+
Кейн	Ma/Xu	1020	851	-	-

4 Работы по алгоритмам

При выводе уравнений динамики манипуляторов используются различные законы и формулировки общих уравнений динамики систем. Среди них можно выделить методы, основанные на уравнениях Лагранжа, Ньютона-Эйлера, Д'Аламбера, Гаусса, Аппеля, Кейна.

Уравнения динамики в форме Лагранжа впервые были получены в работе Uicker [19] и получили дальнейшее развитие в плане повышения эффективности в работах Kahn, Vukobratovic, Mahil, Renaud, Thomas и Tesar. Все перечисленные методы позволяли решать прямую и обратную задачу динамики, были удобны в алгоритмической реализации (кроме Renaud), но обладали низкой вычислительной эффективностью. Waters и Hollerbach [6] применили рекурсивные преобразования при выводе динамических уравнений, причем при использовании матриц поворотов 3×3 было получено значительное сокращение числа операций, но эти методы позволяли решать лишь обратную задачу динамики, поэтому не были пригодны для моделирования. Рекурсивные преобразования и формулы Родриго использовали Vukobratovic и Potkonjak, причем их метод позволял решать и прямую задачу динамики, хотя его вычислительная эффективность и не столь высока. Значительный прогресс в сокращении числа операций достигнут в работах Renaud и Li, также применивших рекурсивные соотношения.

Среди методов, использующих уравнения Ньютона-Эйлера и позволяющих решать прямую задачу динамики, можно отметить работы Vukobratovic и Stepanenko, Orin и Walker, Armstrong, Wang и Ravani. Во всех работах применяется рекурсия. Отметим также алгоритм Balafoutis [8], в котором за минимальное число операций решена обратная задача динамики.

Использование принципа Д'Аламбера для уравнений Лагранжа позволяет получить достаточно эффективные динамические соотношения, в которых в явном виде отражены эффекты влияния вращательного и поступательного движения звеньев на динамику манипулятора. В работе Попова получены уравнения динамики в явном виде с использованием уравнений Аппеля, позволяющие решать прямую и обратную задачи динамики. Урав-

нения Кейна особенно эффективны для расчета обобщенных моментов манипуляторов с замкнутыми кинематическими цепями. В работах Коноплева для описания динамики манипуляционных систем применяются агрегативные модели; метод удобен для применения символьных преобразований. Погореловым разработан пакет программ моделирования динамики широкого класса механических систем, включая роботы-манипуляторы. Интересный подход и язык программирования уравнений движения сложных механических систем, состоящих из твердых тел, предложен Сазоновым.

Полный анализ основных достижений в области моделирования динамики роботов, начиная с первых работ 60-70 годов прошлого века по 2000 гг., дан в работе Featherstone и Orin [12].

5 Открытые проблемы

Отметим, что, несмотря на постоянный рост мощности компьютеров, требование высокой вычислительной эффективности уравнений динамики остается критичным. Это объясняется тем, что, во-первых, в системах управления роботов используются как правило относительно медленные процессоры, и для решения уравнений динамики в реальном времени необходимы эффективные алгоритмы расчета. А, во-вторых, сложность механических структур современных роботов (параллельных, с избыточными степенями подвижности, так называемых роботов-гуманоидов) требует эффективных методов расчета динамики для задач их моделирования и управления.

Алгоритмы для расчета динамика роботов с гибкими звеньями и сочленениями, проблемы идентификации параметров динамических моделей, динамика физического контакта робота с окружающей средой, динамика роботов с подвижной базой – все эти задачи требуют усовершенствования.

Литература

- [1] Featherstone, R. (1987). Robot Dynamics Algorithms
- [2] Фу К., Гонсалес Р., Ли К. Робототехника.- М.: Мир, 1989.
- [3] Kane T., Dynamics, New York, Holt, Rinehart and Wiston, 1968.
- [4] Виттенбург Й. Динамика систем твердых тел.- М.: Мир, 1980.
- [5] Denavit J, Hartenberg R.S. A kinematic notation for lower-pair mechanisms based on matrices., J. Appl. Mech., 77, 1955, с.215-221.
- [6] Hollerbach J. A recursive Lagrangian formulation of manipulator dynamics and comparative study of dynamic complication complexity. IEEE Trans. on SMC, SMC-10, No 11, 1980, с.730-736.
- [7] Vukobratovic M., Stepanenko Y. Mathematical model of general anthropomorphic systems. Math Biosciences, Vol.17, 1973, с.191-242.
- [8] Balafoutis C, Patel R., Misra P. Efficient modeling and computation of manipulator dynamics using orthogonal cartesian tensors. IEEE J. of Rob. and Autom., 4, N 6, с.665-676.
- [9] Dapper, R. Maaf, V. Zahn, R. Eckmiller, Neural Force Control (NFC) Applied to Industrial Manipulators in Interaction with Moving Rigid Objects, Proceedings of the 1998 IEEE International Conference on Robotics Automation, Leuven, Belgium, May 1998.
- [10] G. Rodriguez, A. Jain and K. Kreutz-Delgado, "A Spatial Operator Algebra for Manipulator Modelling and Control," Int. J. Robotics Research, vol. 10, no. 4, pp. 371-381, 1991.

- [11] M. Emami, A. Goldenberg, I. Turksen, Fuzzy-Logic Dynamics Modeling of Robot Manipulators, Proceedings of the 1998 IEEE International Conference on Robotics Automation, Leuven, Belgium, May 1998.
 - [12] R. Featherstone, D. Orin, Robot Dynamics: Equations and Algorithms, Proceedings of the 2000 IEEE International Conference on Robotics Automation, San Francisco, CA, April 2000.
 - [13] Siciliano, B., and Khatib, O. (eds.) (2008). Springer Handbook of Robotics
 - [14] Вукобратович М., Стокич Д., Кирчански Н. Неадаптивное и адаптивное управление манипуляционными роботами. - М.: Мир, 1989.
 - [15] Шахинпур М. Курс робототехники пер. с англ. — М, Мир, 1990)
 - [16] Megahed S., Renaud M., "Minimization of the Computation Time Necessary for the Dynamic Control of Robot Manipulators 12th ISIR, Paris, 1982
 - [17] Renaud N., "An Efficient Iterative Analytical Procedure for Obtaining a Robot Manipulator Dynamic Model Proc. of First International Symp. of Robotics Research, Bretton Woods, New Hampshire, USA, 1983.
 - [18] Han J.-Y. Fault-tolerant computing for robot kinematics using linear arithmetic code. IEEE Int. Conf. Robotics and Automation, Cincinnati, May May 13-18, 1990, vol. 1, c.285-290.
- Uicker J.J. Dynamic force analysis of spatial linkages, ASME J. of appl. mech., June, 1967, c.418-424.