

РАЗРАБОТКА АЛГОРИТМА ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ В МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ НА ПРИМЕРЕ РЕШЕНИЯ ЗАДАЧ ДИНАМИКИ ПРОМЫШЛЕННЫХ МАНИПУЛЯЦИОННЫХ РОБОТОВ

А.А. Внуков¹, М.А. Лисенков²

¹Кафедра кибернетики и мехатроники
Российский университет дружбы народов
ул. Орджоникидзе, 3, Москва, Россия, 117923

²Кафедра математического обеспечения
систем обработки информации и управления
Московский государственный институт электроники и математики
Трехсвятительский пер., 3, Москва, Россия, 109028

В статье рассматривается модель многопроцессорной системы, которая позволяет существенно ускорить выполнение модульных алгоритмов решения прямой и обратной задач динамики промышленного манипуляционного робота. Предложена методика выполнения данных алгоритмов на разработанной процессорной структуре с общей памятью. Разработана модель специализированного мультипроцессорного вычислителя, используемого для ускорения вычислений. Произведено моделирование аппаратно-программного комплекса для задач динамики и получены оценки эффективности работы параллельного алгоритма.

Ключевые слова: параллельный алгоритм, динамика промышленного робота.

Актуальность данной тематики обусловлена тенденцией к снижению темпов роста тактовой частоты микропроцессоров и возрастанием внимания к использованию всех возможностей многоядерных и многопроцессорных систем. Проблемы распараллеливания и распределения вычислений сегодня являются одним из основных и крайне важных аспектов при построении программных и аппаратных комплексов.

Большинство задач робототехники ряда дисциплин, среди которых механика и приводы роботов, информационные, электронные и вычислительные устройства роботов, методы искусственного интеллекта, управление сложными мехатронными системами, моделирование и исследование робототехнических комплексов, обладают высокой вычислительной сложностью и интенсивностью, связанной с частым повторением, и требуют соответствующих высокопроизводительных вычислительных ресурсов. Одним из решений этой проблемы является распараллеливание вычислений, как на аппаратном, так и на программном уровне. Таким образом, функциональные и распределенные вычисления обладают огромной важностью в сфере робототехники.

В настоящее время робототехника является быстроразвивающимся направлением науки и техники, которое связано с созданием и применением роботов и робототехнических систем. Робот представляет собой универсальный автомат

для воспроизведения двигательных и интеллектуальных функций человека. Среди различных роботов важным классом является манипуляционный робот (рис. 1). Практической целью создания роботов явилась передача им тех видов деятельности, которые для человека являются трудоемкими, монотонными, вредными для здоровья и жизни. Для управления работой промышленного робота-манипулятора необходима вычислительная система, которая в реальном времени решает прямую и обратную задачи преобразования координат. Вычислительная система реализует систему управления, систему технического зрения, информационную систему и другие системы. Для реализации такого программного обеспечения необходимы системы прежде всего с высоким быстродействием, а возможности современных вычислительных систем часто ниже требуемых.



Рис. 1. Промышленный робот-манипулятор KUKA KR 360 450-2 PA

Вычислительные системы для роботов-манипуляторов могут быть многопроцессорными. Распараллеливание процессов выполнения и обработки данных может быть организовано по функциональному признаку, т.е. параллельно будут выполняться, например, системы технического зрения, система управления, система распознавания, а также информационная система. Также распараллеливание может производиться с целью сокращения времени выполнения алгоритмов этих и других систем.

Основные задачи исследования:

- разработка концепции многопроцессорной системы;
- разработка эффективного алгоритма параллельного вычисления для решения прямой и обратной задачи динамики промышленного манипулятора;
- разработка программного обеспечения, позволяющего реализовать и смоделировать поставленные задачи.

Цель исследования. Программная система последовательно должна решать две динамические задачи для шестиосевого манипулятора. Прямая задача по заданному закону движения определяет обобщенные силы звеньев, обратная задача — по известным обобщенным силам — закон движения в заданных промежутках вре-

мени. При этом программа должна моделировать выполнение параллельного алгоритма в многопроцессорной системе и выполнение последовательного алгоритма на одном сложном процессоре.

Модель шестизвенного промышленного манипуляционного робота. Манипулятором называют разомкнутую механическую систему (цепь), состоящую из твердых тел, которые последовательно соединены между собой при помощи шарниров, или призматических (телескопических) сочленений, т.е. это исполнительный механизм промышленного робота, оснащенный приводами и рабочим органом, с помощью которого осуществляется выполнение рабочих функций. Способность воспроизводить движения, подобные движениям рук человека, достигается приданием манипулятору нескольких степеней свободы (рис. 2), по которым осуществляется управляемое движение с целью получения заданного движения рабочего органа — схвата. Число степеней свободы — это число возможных перемещений системы. Оно равно числу обобщенных координат, под которыми понимают независимые переменные, однозначно определяющие положение манипулятора в пространстве.

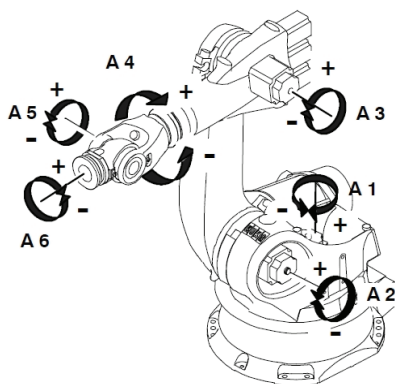


Рис. 2. Робот-манипулятор с шестью степенями свободы

Фактически промышленный манипуляционный робот представляет манипулятор и блок управления с контроллером и ручным пультом. Манипулятор промышленного робота по своему функциональному назначению должен обеспечивать движение выходного звена и, закрепленного в нем, объекта манипулирования в рабочем пространстве по произвольной заданной траектории. При этом его захват должен находиться с заданными ориентацией и направлением подхода к промежуточным и конечной точкам перемещения. Для полного выполнения этого требования основной рычажный механизм манипулятора должен иметь не менее шести подвижностей (степеней свободы), причем движение по каждой из них должно быть управляемым.

Промышленный робот с шестью степенями свободы является сложной автоматической системой. Эта система сложна как в изготовлении, так и в эксплуатации.

Динамическая модель манипуляционного робота. При рассмотрении динамической модели будем использовать уравнения динамики, необходимые для динамического управления роботами. Целью динамического управления является получение требуемого динамического отклика управляемого от ЭВМ манипулятора, чтобы этот отклик соответствовал некоторому заранее определенному множеству критериев. Эти критерии могут быть выражены через импульс и силы реакции и инерции, воздействующие на схват либо на объект. В общем случае проблема управления заключается в получении основных уравнений динамики робота-манипулятора в форме динамической модели физического манипулятора и в последующем определении законов управления, позволяющих достичь желаемого динамического отклика.

В связи с тем, что для построения модели динамики переходных процессов и дальнейшего анализа полученных уравнений необходима аналитическая форма, решено использовать для получения уравнений динамики метод Лагранжа—Эйлера. Основные уравнения:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) = \tau_i,$$

где $L = L(q_i, \dot{q}_i, t) = T - V$ — функция Лагранжа; $T = T(q_i, \dot{q}_i, t)$ — полная кинетическая энергия системы; $V = V(q_i, t)$ — полная потенциальная энергия системы; q_i — обобщенные координаты (θ_i); τ_i — обобщенные силы и моменты.

Прямая и обратная задачи динамики манипулятора. Прямая задача динамики формулируется так: задано движение механической системы, требуется определить обобщенные силы, обеспечивающие это движение. В манипуляторе обобщенные силы Q_i — это проекции сил (например, силы привода) на оси кинематических пар моментов (если пары вращательные) и сил (если эти пары поступательные). Движение определяется изменением во времени обобщенных координат, q_i , которыми являются углы поворота во вращательных и смещения — в поступательных кинематических парах. Для ее решения нам необходимо получить уравнение, связывающее обобщенные силы и движение механической системы.

Основное уравнение Лагранжа II рода в матричной записи выглядит следующим образом:

$$\sum_{i=j}^n \sum_{k=1}^i \text{tr} (U_{ij} H_i U_{ik}^T) \ddot{q} + \sum_{i=j}^n \sum_{k=1}^i \sum_{l=1}^i \text{tr} (U_{ij} H_i U_{ikl}^T) \dot{q}_k \dot{q}_l - \sum_{i=j}^n m_i G^T U_{ij} R_i^* = Q_j. \quad (1)$$

При решении прямой задачи обобщенные координаты $q_l(t)$ — известные функции времени. Дифференцируя их, можно получить \dot{q}_l и \ddot{q}_l . Зная $q_l, \dot{q}_l, \ddot{q}_l$, можно вычислить все величины входящие в левую часть уравнения (1). Следовательно, можно вычислить обобщенные силы Q_j . Этим и определяется алгоритм решения первой задачи динамики.

Обратная задача динамики состоит в определении закона движения по заданным обобщенным силам. Решение обратных задач динамики позволяет определить основные требования, которые должны быть наложены на систему, чтобы движение с заданными свойствами было возможным, а также законы изменения и структуру управляющих сил.

Особенностью обратных задач динамики является то, что решение для управляющих сил находится в форме зависимости от состояния системы, т.е. от обобщенных координат и их производных, в отличие от прямой задачи динамики, когда эти значения определяются в функциях времени.

Итак, здесь известными считаются обобщенные силы $Q_j(t)$ и значения обобщенных координат и их скоростей в начальный момент t_0 , т.е. $q_l(t_0)$ и $\dot{q}_l(t_0)$. Подставляя в (1) величины $q_l(t_0)$, $\dot{q}_l(t_0)$ и $Q_j(t_0)$, можно привести это уравнение к следующему виду:

$$\sum_{k=1}^n \alpha_{kj}(t_0) \ddot{q}_k(t_0) = \beta_j(t_0), \quad (j = 1, 2, \dots, n),$$

где $\alpha_{kj}(t_0)$ и $\beta_j(t_0)$ — известные числа.

Аргумент t_0 означает, что эти числа определены для момента t_0 и, следовательно, зависят от выбора этого момента. Фактически они зависят от $q_l(t_0)$, $\dot{q}_l(t_0)$, $Q_j(t_0)$. Из n уравнений (1) можно найти n неизвестных величин: $\ddot{q}_k(t_0)$, $k = 1, 2, \dots, n$. Решение уравнений удобно записать в матричной форме. Для этого введем матрицу α размером $n \times n$ с элементами α_{kj} , матрицу-столбец β с элементами β_j и матрицу столбец \ddot{q} с элементами \ddot{q}_k . Система уравнений (1) примет вид $\alpha \ddot{q} = \beta$, а ее решением будет $\ddot{q} = \alpha^{-1} \beta = \gamma$.

Найти закон движения манипулятора — значит найти функции $q_l(t)$ или в матричном виде — $q(t)$, где q — матрица-столбец с элементами q_l . Чтобы найти $q(t_0)$, необходимо проинтегрировать дифференциальное уравнение. В этом уравнении матрица γ может быть вычислена для любого момента t , если для этого момента известны матрицы-столбцы q , \dot{q} , Q (элементы матрицы \dot{q} — это \dot{q}_l , а элементы матрицы Q — это Q_j). Способ вычисления $\gamma(t)$ тот же, что и способ вычисления $\gamma(t_0)$, описанный выше.

Фактически через матрицы α и β осуществляется связь между прямой и обратной задачами динамики. При решении обратной задачи находится \ddot{q} , которое должно практически совпадать с \ddot{q} , заданным при решении прямой задачи.

Используем для интегрирования дифференциального уравнения алгоритм Эйлера. Будем двигаться по оси времени малыми шагами Δt , начиная с t_0 . На каждом шаге будем считать постоянными матрицы \dot{q} , \ddot{q} . Тогда на шаге номер m получим:

$$\begin{aligned} \dot{q}^{(m+1)} &= \dot{q}^{(m)} + \ddot{q}^{(m)} \Delta t, \\ q^{(m+1)} &= q^{(m)} + \dot{q}^{(m)} \Delta t. \end{aligned}$$

Для шага $m = 0$ правые части известны, что позволяет вычислить $\dot{q}^{(1)}$ и $q^{(1)}$. После этого делается следующий шаг ($m = 1$), на котором правые части также известны, и т.д. В результате q, \dot{q}, \ddot{q} становятся известными в дискретных точках оси времени, разделенных интервалами Δt . Чем меньше Δt , тем меньше погрешность расчета, но выше его трудоемкость. В нашем случае из-за равноускоренного движения \ddot{q} вычисляется только в начальный момент.

Разработка модели многопроцессорной вычислительной системы. Для наиболее эффективной реализации алгоритмов решения задач динамики была разработана вычислительная архитектура (ВА) на СБИС (сверхбольшая интегральная схема). Данная ВА представляет собой совокупность процессорных элементов (ПЭ), которые имеют постоянные неизменяемые соединения между собой и, таким образом, обеспечивается вычислительная сеть. В качестве ПЭ могут использоваться систолические чипы, транспьютеры и подобные им цифровые сигнальные процессоры (ЦСП).

Процессорное поле состоит из одинаковых типов ПЭ, которые имеют одинаковые внешние интерфейсы для передачи и приема данных. Такое процессорное поле будет однородным. В нем не требуется использование между ПЭ дополнительных устройств и интерфейсных схем для согласования форматов данных и характеристик электротехнических сигналов, это упрощает (удешевляет) процессорную сеть, делает ее более надежной и быстрой, так как отсутствуют дополнительные промежуточные устройства. Расстояние между ПЭ должны быть одинаковыми или желательно, чтобы были практически одинаковыми. В этом случае время на передачу данных и электрических сигналов между всеми ПЭ сети на процессорной плате или кристалле заказной СБИС будет одинаковым. Процессорный элемент является определенным с точки зрения его топологических характеристик, входных и выходных потоков данных (входов и выходов ПЭ) и процессорных функций (операции, которые выполняет ПЭ), которые в данном случае являются одинаковыми для всех ПЭ.

Процессорное поле определяется глобальной вычисляемой моделью, а каждый ПЭ, определяется локальной моделью. В общем случае обобщенная вычисляемая модель процессорного поля описывается тройкой (G, F, T) , где специфицирующей G -геометрию процессорной сети, т.е. определяет глобальную модель, F — процессорную функцию для G (глобальной модели), т.е. определяет глобальную модель и T — время выполнения для F (локальная модель), т.е. определяет модель управления процессорным полем. Для модели управления задаются длительности фаз коммуникации, вычислений и иногда длительность собственно фазы управления, во время которой на процессорное поле подается один поток команд. Один поток команд — это команды, которые поступают разом на все ПЭ. Геометрия процессорной сети составляет глобальную модель процессорного поля и есть не что иное, как соединение ПЭ между собой. Локальная модель F процессорного поля определяется процессорной функцией для каждого ПЭ.

Таким образом, топологические особенности ПЭ, входные и выходные потоки данных для каждого ПЭ, процессорная функция для каждого ПЭ — эта совокупность является заданием глобальной и локальной моделей процессорного поля.

Для подключения платы процессорного поля к вычислительной или технической системе необходима интерфейсная система. Интерфейсная система рассматривается главной вычислительной системой, как внешнее устройство. Главная вычислительная система и интерфейсная система могут функционировать одновременно. Необходима специализированная интерфейсная система. Такой интерфейс поддерживает работу одного или нескольких процессорных полей и обеспечивает минимальные потери времени, не связанные с вычислительным процессом. Для организации правильного пространственно-временного потока данных на процессорном поле нужно иметь:

— специальным образом организованную память (на основе модели магазинной памяти);

— устройство управления такой памятью, выполняющее как операции ввода/вывода данных в (из) систолического поля, так и синхронизацию работы с процессорами сети по фазам коммуникации и вычисления (в качестве устройства управления можно использовать универсальный программируемый процессор).

Разработка специализированного мультипроцессорного вычислителя.

Рассматривается задача разработки специализированного вычислителя для умножения двух ленточных матриц, используемого при преобразовании координат.

Требуется:

— разработать специализированный вычислитель;

— использовать его в универсальной вычислительной структуре как сопроцессор или вычислительное устройство;

— использовать в технической системе в качестве встраиваемой платы.

Некоторые этапы разработки макета вычислителя:

1) выбираем правильный шестиугольник в качестве топологии ПЭ (вместо реальной);

2) задаем процессорную функцию следующим образом (рис. 3):

$$a' = a$$

$$b' = b$$

$$c' = a * b + c$$

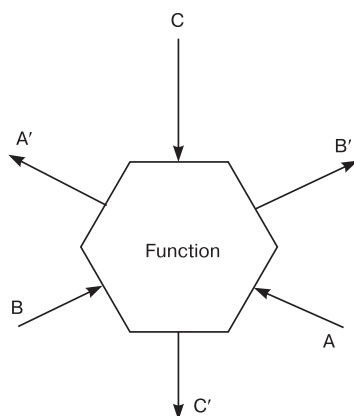


Рис. 3. ПЭ и выполняемая функция

3) задаем входные и выходные потоки и их направления (рис. 4):

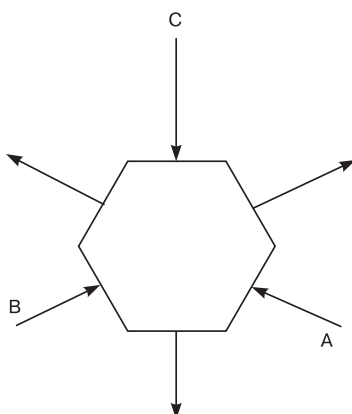


Рис. 4. Направление потоков данных

4) нужно задать примерную длительность фазы коммуникации и вычисления на основе времени выполнения соответствующих команд ПЭ, выбрав их несколько большими, чем это требуется для коммуникации и вычислений для их корректного завершения;

5) математическая постановка задачи умножения матриц. Зададим операцию $A \cdot B = C$, где матрицы (4×4) имеют вид:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ 0 & b_{32} & b_{33} & b_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ 0 & c_{32} & c_{33} & c_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Матрицы A, B — это пятидиагональные ленточные матрицы; C — шестидиагональная ленточная матрица. На рисунке 5 показано использование общей памяти (ОЗУ) и буферной памяти для хранения промежуточного результата.

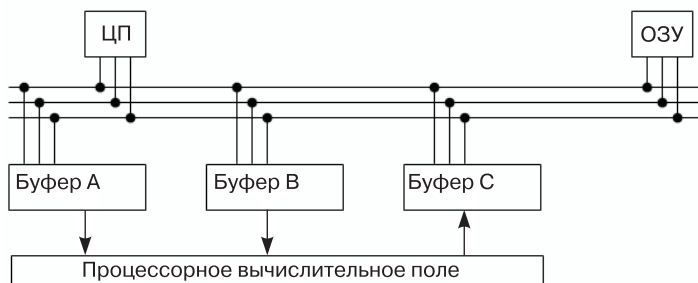


Рис. 5. Организация вычислительной среды

6) создаем диаграмму, которая показывает направление потоков данных и определяет длительность фаз коммуникаций и вычислений (рис. 6). Таким образом, диаграмма представляет собой уже готовую структуру будущего процессорного поля. Достаточно произвести замену узловых точек на «шестиугольные» ПЭ (рис. 7).

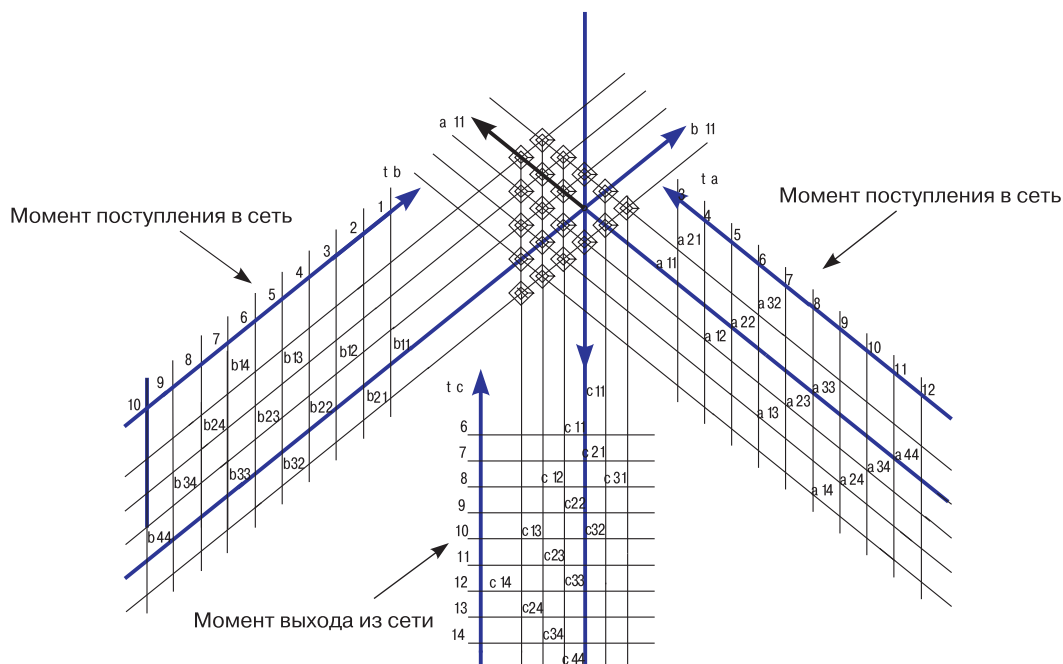


Рис. 6. Диаграмма следования элементов матриц в потоках данных с отчетами времени

На рисунке 6 отсчеты времени определяют моменты времени поступления или выхода элементов потоков данных для процессорного поля.

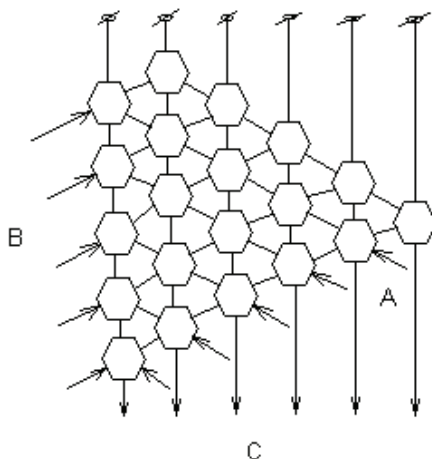


Рис. 7. Вычислительное поле, состоящее из 20 ПЭ

Разработка программного обеспечения для решения прямой и обратной задачи динамики

Программа обеспечивает повышение производительности и точности в распределенных вычислениях. Она состоит из нескольких вычислительных блоков, включающих блок моделирования многопроцессорной обработки данных (рис. 8). Данный блок моделирует параллельное выполнение арифметических операций на различных процессорных элементах.

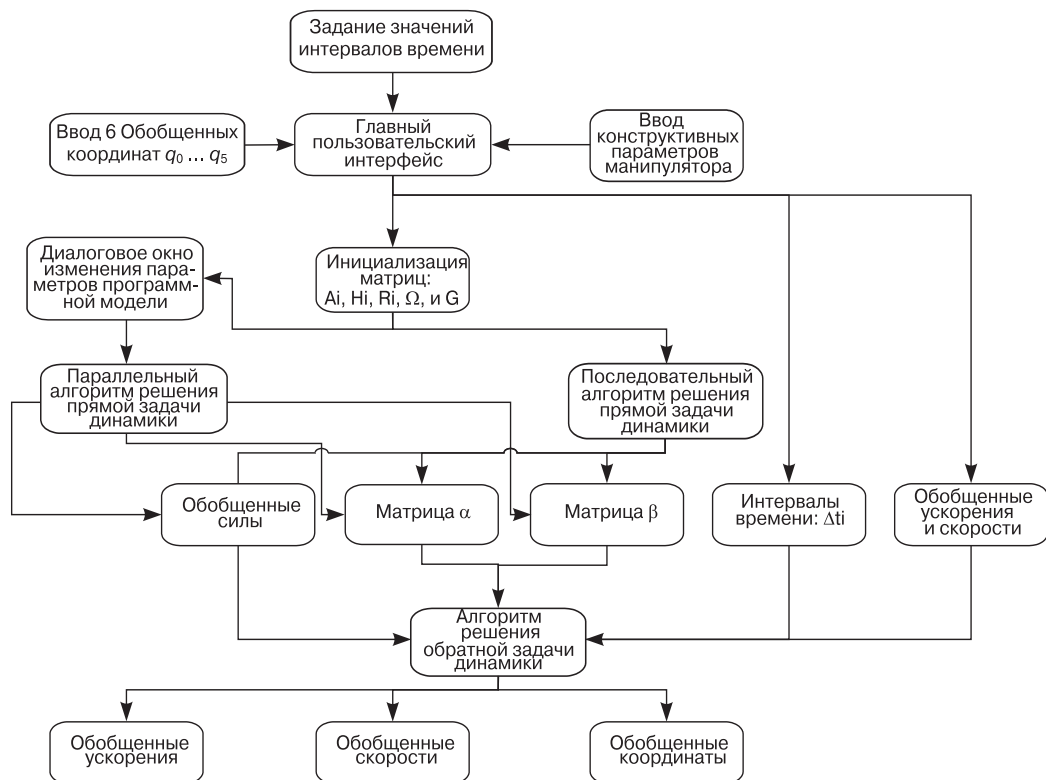


Рис. 8. Структурная схема программы

Последовательность выполнения программного приложения:

- 1) задание матриц $A_0 \dots A_5$, матрицы-строки G^T (ускорения свободного падения), матриц Ω для вращательного и поступательного видов звеньев;
- 2) задание координат центров тяжести звеньев манипулятора и других параметров для нахождения матриц $H_0 \dots H_5$;
- 3) задание матриц-столбцов $R_0 \dots R_5$;
- 4) нахождение матриц U_{ij} ($U_{00} \dots U_{55}$);
- 5) нахождение матриц U_{ijk} ($U_{000} \dots U_{555}$);
- 6) нахождение $Q_0 \dots Q_5$: для каждого Q решаются уравнения Лагранжа II рода, т.е., вычисляются левые части каждого уравнения. Таким образом, находятся Q_i . При этом в программе формируются элементы матриц α и β , которые записываются в файл и являются входными данными для программы, решающей обратную задачу динамики;
- 7) задается матрица α (считывается из памяти);
- 8) определитель данной матрицы проверяется на равенство нулю;
- 9) если определитель не равен 0, вычисляется обратная матрица α^{-1} ;
- 10) задается матрица-столбец β (считывается из памяти);

- 11) вычисляется матрица $Y = \alpha^{-1} \cdot \beta$ (т.е. вычисляется столбец \ddot{q} в начальный момент времени);
- 12) задаются начальные условия для системы уравнений;
- 13) задаются промежутки времени Δt (предусмотрена проверка на неотрицательность Δt);
- 14) находятся q, \dot{q} . Нахождение q, \dot{q} производится в шести точках (результаты выводятся на экран).

Разработка последовательного и параллельного алгоритма для многопроцессорной системы, решающей задачи динамики манипулятора. Одним из основных элементов решения прямой и обратной задач динамики, поддающихся распараллеливанию, является перемножение матриц. К тому же, в прямой задаче обращение к перемножению матриц происходит более 30 раз. Поэтому имеет смысл разрабатывать параллельный и последовательный алгоритмы именно для вычисления этого элемента.

Ускорение работы последовательного алгоритма осуществляется с помощью создания специального высокопроизводительного вычислительного блока. На основе разработанной структуры можно собрать заказную СБИС, в которой будет реализовано процессорное поле с 20 ПЭ с постоянными соединениями, как показано на рис. 7.

При каждом перемножении матриц будем использовать разработанное систолическое поле, которое будет являться спецвычислителем, обеспечивающим параллельную обработку данных вместо последовательного алгоритма умножения матриц, реализованного последовательной программой.

Последовательный алгоритм аналогичен параллельному алгоритму (рис. 9) и имеет лишь то отличие, что не использует вычислительных блоков для аппаратного умножения матриц преобразования, а все операции выполняются последовательно на одном сложном процессоре.

Сравнительная оценка эффективности последовательного и параллельного алгоритмов. Последовательный алгоритм выполнит умножение шести матриц преобразования A_i размерности (4×4) за 320 тактов, во время которых производится 320 операций умножения и 320 операций сложения. При этом может потребоваться определенное количество раз $256 \sim 384$ ($4 \times 64 - 4 \times 96$ раз для $T_{\text{посл}}$) пересылки данных между ЦП и ОЗУ в зависимости от системы команд (с командой операции умножения с накоплением или без нее).

Время выполнения параллельного и последовательного алгоритма с учетом операций умножения (\times), сложения ($+$), пересылки (\rightarrow) можно оценить: $T_{\text{паралл}} = 75 (\times, +) + (16 \sim 32)$ (внешняя \rightarrow); $T_{\text{посл}} = 320 (\times, +) + (256 \sim 384)$ (внутренняя \rightarrow). Эти оценки являются оценками сверху для времени выполнения.

Таким образом, оценка ускорения выполнения параллельного алгоритма, по сравнению с последовательным алгоритмом, выполненным на одном сложном процессоре, составит, примерно, величину $(320)/(75) = 4,266$ без учета затрат времени на выполнение операций внешней и внутренней передачи данных.

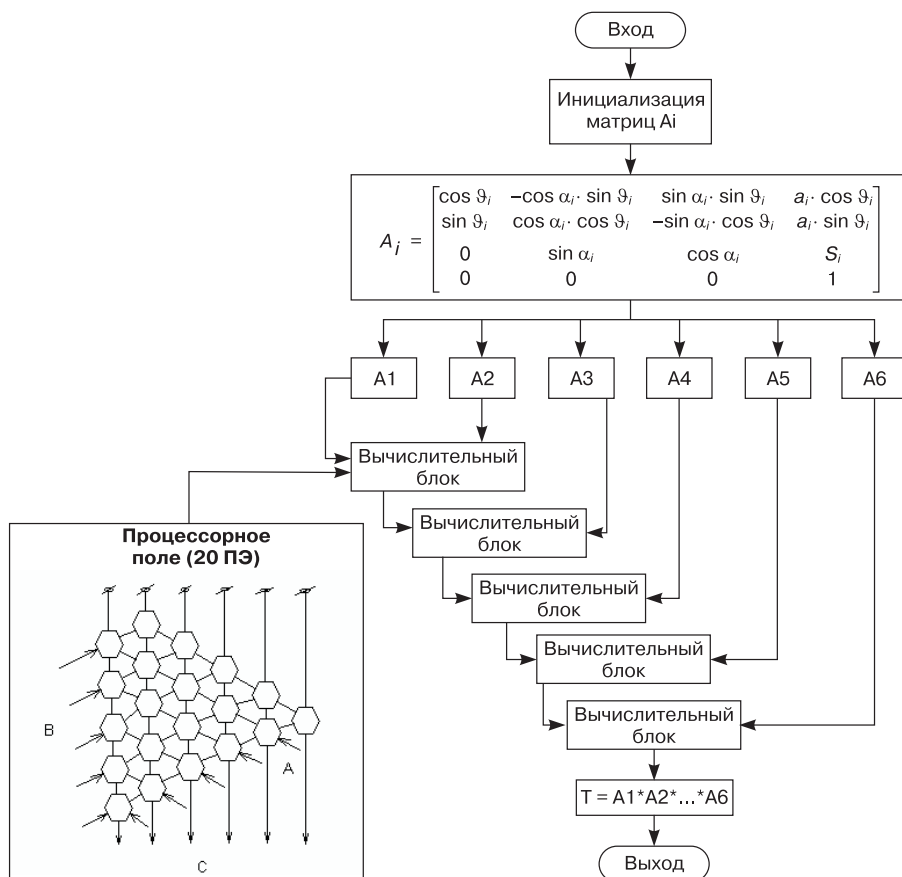


Рис. 9. Параллельный алгоритм для решения прямой задачи динамики

Заключение. Для реализации параллельного алгоритма решения прямой и обратной задач динамики решено использовать процессорную структуру, разработанную в дипломном проекте (см. рис. 7). На основе разработанной структуры можно собрать заказную СБИС, в которой будет реализовано процессорное поле с 20 ПЭ с постоянными соединениями. Стоит отметить, что данная СБИС является универсальной и может использоваться не только в качестве блока роботоманипулятора, она может быть встроена в различные технические устройства для ускорения вычислений.

Результаты, полученные в данном проекте при экспериментальных измерениях времени функционирования последовательного и параллельного алгоритмов, позволяют разработать различные варианты аппаратного и программного обеспечения многопроцессорной вычислительной системы для блока управления промышленного манипуляционного робота. Также достигнуто ускорение вычислений равное 4,266 раза по сравнению с вычислениями на одном сложном процессоре.

Авторами статьи зарегистрировано в Реестре программ для ЭВМ 10 августа 2009 г. и получено Свидетельство о государственной регистрации программы для ЭВМ № 2009614169 «Параллельная обработка данных в многопроцессорных системах для решения задач динамики промышленных манипуляционных роботов».

Таким образом, разработанное программное и аппаратное обеспечение является крайне полезным для трудоемких вычислений, а дальнейшее его совершенствование и модернизация позволит решать комплексные общие задачи, в том числе динамики роботов.

ЛИТЕРАТУРА

- [1] *Лисенков М.А.* Разработка и моделирование алгоритма параллельной обработки данных в многопроцессорных системах для решения задач динамики промышленных манипуляционных роботов: Дипл. раб. — М.: МИЭМ, 2009.
- [2] *Внуков А.А., Гайсина Л.Ф., Селескерова Е., Сударкина Е.В.* Разработка спецвычислителя для прямого преобразования координат «Интеллектуальные системы» // Труды VII Международного симпозиума (Краснодар, 26 июня — 30 июня 2006 г.) / Под ред. К.А. Пупкова. — М.: РУСАКИ, 2006. — С. 507—510.
- [3] *Механика промышленных роботов. Т. 1: Кинематика и динамика* / Под ред. К.В. Фролова, Е.И. Воробьева. — М.: Высшая школа, 1989.
- [4] *Шахинпур М.* Курс робототехники / Под ред. С.Л. Зинкевича. — М.: Мир, 1990.
- [5] *Попов Е.П.* Управление роботами-манипуляторами // Техника, кибернетика. — 1974. — № 6. — С. 51—56.
- [6] Схемы параллельных алгоритмов задач: www.parallel.ru
- [7] *Шпаковский Г.И.* Организация параллельных ЭВМ и суперскалярных процессоров. — Минск, 1996.
- [8] *Внуков А.А., Афонин М.С.* Разработка и реализация режимов платы цифрового управления промышленным роботом с электроприводом // Вестник РУДН. Серия «Инженерные исследования». — 2007. — № 4. — С. 71—86.
- [9] *Внуков А.А., Зозулина А.С.* Разработка методики разбиения связанных между собой подзадач для параллельного выполнения кинематических преобразований манипулятора // Вестник РУДН. Серия «Инженерные исследования». — 2009. — № 4. — С. 73—85.

DEVELOPMENT OF PARALLEL ALGORITHM OF DATA PROCESSING IN MULTIPROCESSOR SYSTEMS ON EXAMPLES OF TASKS OF DYNAMICS OF INDUSTRIAL MANIPULATION ROBOT

A.A. Vnukov¹, M.A. Lisenkov²

¹Cybernetic and mechatronics department
Peoples' Friendship University of Russia
Ordjonikidze str., 3, Moscow, Russia, 117923

²Department of Mathematical data processing and control
Moscow state institute of electronics and mathematics
B. Trehsvjetitsky, 3, Moscow, Russia, 109028

In this article the model of the multiprocessing system which allows to significantly speed up the implementation of modular algorithms for solving of direct and reverse tasks of dynamics of the industrial manipulation robot is considered. A method for the implementation of these algorithms on the established structure of the processor with shared memory is offered. Development of specialized multiprocessor field used to accelerate calculations. Produced simulation of the hardware-software complex for the problems of dynamics and estimations of an overall performance of parallel algorithm are received.

Key words: parallel algorithm, robots, dynamics.