

РАЗРАБОТКА МЕТОДИКИ РАЗБИЕНИЯ СВЯЗАННЫХ МЕЖДУ СОБОЙ ПОДЗАДАЧ ДЛЯ ПАРАЛЛЕЛЬНОГО ВЫПОЛНЕНИЯ КИНЕМАТИЧЕСКИХ ПРЕОБРАЗОВАНИЙ МАНИПУЛЯТОРА

А.А. Внуков

Кафедра кибернетики и мехатроники
Российский университет дружбы народов
ул. Миклухо-Маклая, 6, Москва, Россия, 117198

А.С. Зазулина

Кафедра математического обеспечения
систем обработки информации и управления
Московский государственный институт электроники и математики
Б. Трехсвятительский пер., 3/12, стр. 8, Москва, Россия, 109028

В статье рассматриваются виды параллелизма в модульных алгоритмах прямой и обратной задач кинематики робота-манипулятора, возможные методики их разбиения на отдельные подзадачи для последующего распределения этих подзадач на заранее известную многопроцессорную структуру общей памяти с интегральной характеристикой потенциала параллельности архитектуры. Предложен метод оценки эффективности разбиения.

Ключевые слова: модульный алгоритм кинематическая схема робот-манипулятор.

Постановка задачи. Необходимо разработать методику выделения отдельных связанных между собой подзадач в последовательном алгоритме, а также определения степени их зависимости друг от друга. Разбиение осуществляется для последующего параллельного распределения данного алгоритма на заданную многопроцессорную структуру с интегральной характеристикой потенциала параллельности архитектуры. Поставленную задачу можно представить схематически следующим образом, разделив ее на два последовательных этапа (рис. 1).

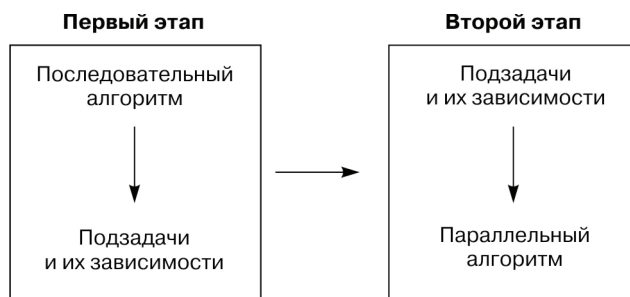


Рис. 1. Постановка задачи

На первом этапе выделяются подзадачи модульного алгоритма и их зависимости друг от друга. На втором этапе определяется параллельное распределение этих подзадач по процессорам многопроцессорной вычислительной системы.

Целью исследования является выполнение следующих работ:

- классификация последовательных алгоритмов по степени возможного параллелизма и выделение группы алгоритмов, для которых распределение на многопроцессорную систему имеет смысл;
- определение приоритетов требований к подзадачам (равный объем вычислений, размер передаваемых данных, частота передачи данных и др.);
- разработка метода выделения подзадач алгоритма в соответствии заданным требованиям;
- параллельное распределение полученных подзадач и оценка эффективности разбиения.

Объектом исследования выступают несколько типов модульных алгоритмов, решающих различные прикладные задачи. В таких задачах должна присутствовать возможность ускорения времени их работы путем параллельного выполнения, т.е. они должны обладать свойствами определенной степени параллелизма, например:

- прямая и обратная задача кинематики манипулятора (функциональный параллелизм).
- перемножение матриц размера $M \times N$ (параллелизм по данным).

При параллелизме данных вычисления сводятся к выполнению однотипной обработки большого набора данных, а при функциональном параллелизме вычисления состоят в выполнении разных операций над одним и тем же набором данных.

Рассмотрим более подробно постановку этих прикладных задач.

1. Прямая и обратная задача кинематики промышленного робота-манипулятора

1.1. Прямая задача кинематики

Первая и основная задача кинематики — определение функции положения. Для пространственных механизмов наиболее эффективными методами решения этой задачи являются векторный метод и метод преобразования координат.

Рассмотрим схемы манипуляторов (рис. 2), которые имеют шесть степеней подвижности, позволяющие расположить захват в любом месте рабочей области и с любой ориентацией в пределах ограничений движения в сочленениях. Справа на рис. 2 показан промышленный робот Kawasaki FS003N, входящий в состав учебного робототехнического комплекса EDU-Pack System лаборатории кафедры кибернетики и мехатроники РУДН.

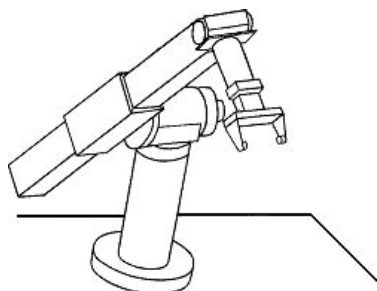


Рис. 2. Кинематические схемы роботов-манипуляторов с шестью степенями свободы

Исполнительный орган состоит из шести звеньев, каждое из которых связано со следующим звеном с помощью подвижного соединения. Подвижные соединения могут быть двух типов: призматические, обеспечивающие относительное скольжение, и вращательные.

Переменной величиной сочленения вращательного типа является угол φ , а переменной величиной призматического соединения — величина s .

Для рассматриваемой конструкции манипулятора начало системы координат можно поместить в основание плеча. Таким образом, мы имеем два вращательных шарнира, после которых следует призматическое соединение, за которым идут три вращательных шарнира с пересекающимися осями. Данная конструкция описывается четырьмя наборами переменных: s_i , φ_i , a_i и θ_i .

Переменные s_i и φ_i соответствуют относительным перемещениям звеньев в кинематических парах и являются обобщенными координатами манипулятора, определяющими конфигурацию механизма в рассматриваемом положении. Переменные a_i и θ_i определяются конструктивным исполнением звеньев манипулятора, в процессе движения они остаются неизменными.

При определении функций положения механизма в прямой задаче находят закон изменения абсолютных координат выходного звена по заданным законам изменения относительных или абсолютных координат звеньев. При решении прямой задачи о положении схвата манипулятора обычно используют метод преобразования координат. Из множества методов преобразования координат, отличающихся друг от друга правилами выбора осей локальных систем координат, обычно используется метод Денавита и Хартенберга.

Опишем два вида матриц:

— матрицы M , определяющие отношение между подвижными системами координат соседних звеньев;

— матрицы T , определяющие положение и ориентацию каждого звена механизма в неподвижной или базовой системе координат.

Воспользуемся однородными координатами трехмерного проективного пространства PR^3 , в которых движение евклидова пространства R^3 можно представить линейным преобразованием

$$\bar{r}_i = M_{ij} \cdot \bar{r}_j,$$

где M_{ij} — матрица 4×4 вида $\begin{vmatrix} U_{ij} & b \\ 000 & 1 \end{vmatrix}$.

Это преобразование эквивалентно преобразованию в евклидовом пространстве $\bar{r}_i = U_{ij} \cdot \bar{r}_j + b$, где $[\bar{r}_i, \bar{r}_j] \in R^3$, т.е. преобразованию, которое включает поворот, определяемый матрицей U_{ij} размерностью 3×3 , и параллельный перенос, задаваемый вектором \bar{b} размерностью 3. В однородном пространстве положение точки будут определять не три x , y и z , а четыре величины x' , y' , z' и t' , которые удовлетворяют следующим соотношениям:

$$x = x'/t', y = y'/t', z = z'/t'.$$

Обычно принимают $t' = 1$. У матрицы поворота U_{ij} элементами u_{ij} являются направляющие косинусы углов между новой осью i и старой осью j . Вектор $\vec{b} = (x, y, z)$ — трехмерный вектор, определяющий положение начала новой системы координат i в старой системе j . Выбор расположения осей должен соответствовать решаемой задаче. При решении задачи о положениях в прямой задаче необходимо определить положение выходного звена как функцию перемещений в приводах. Выбор расположения и ориентации локальных систем координат должен обеспечивать выполнение этой задачи. При использовании метода Денавита и Хартенберга оси координат располагаются по трем правилам.

В прямой задаче необходимо определить положение схвата манипулятора и связанной с ним системы координат $Mx_n y_n z_n$ по отношению к неподвижной или базовой системе координат $Kx_0 y_0 z_0$. Это осуществляется последовательными переходами из системы координат звена i в систему координат звена $i - 1$. Согласно принятому методу, каждый переход включает в себя последовательность четырех движений: двух поворотов и двух параллельных переносов, осуществляемых в указанной последовательности (рис. 3):

— поворот i -той системы вокруг оси x_i на угол $-\theta_i$ до параллельности осей z_i и z_{i-1} (положительное направление поворота при наблюдении с конца вектора x_i против часовой стрелки);

— перенос вдоль оси x_i на величину $-a_i$ до совмещения начала системы координат O_i с точкой пересечения осей x_i и z_{i-1} (отсчет по оси x_i от точки пересечения оси x_i и оси z_{i-1});

— перенос вдоль оси z_{i-1} на величину $-s_i$, после которого начало системы координат O_i оказывается в начале координат O_{i-1} системы $(i - 1)$ (отсчитывается по оси z_{i-1} от ее начала координат O_{i-1} до точки ее пересечения с осью x_i);

— поворот вокруг оси z_{i-1} на угол $-\varphi_i$, до тех пор пока ось x_i не станет параллельной оси x_{i-1} (положительное направление поворота при наблюдении с конца вектора z_{i-1} против часовой стрелки).

Необходимо отметить, что знак угла поворота не имеет значения, так как в матрицах перехода используются направляющие косинусы (четные функции). Целесообразно рассматривать угол, обеспечивающий кратчайший поворот оси старой системы i до совмещения (параллельности) с соответствующей осью новой $(i - 1)$. Перемещения начала координат определяются как координаты начала старой системы O_i в новой O_{i-1} .

Матрицы перехода их системы O_i в систему O_{i-1} можно записать так:

$$M_i = M_i^\theta \cdot M_i^a \cdot M_i^s \cdot M_i^\varphi,$$

где матрицы в порядке следования в формуле слева направо: матрица поворота вокруг оси x_i на угол $-\theta_i$, матрица переноса вдоль оси x_i на $-a_i$, матрица переноса вдоль оси z_{i-1} на $-s_i$, матрица поворота вокруг оси z_{i-1} на угол $-\varphi_i$.

Положение некоторой произвольной точки M в системе координат звена i определяется вектором \mathbf{r}_{M_i} , а в системе координат звена $(i - 1)$ — вектором $\mathbf{r}_{M_{i-1}}$. Эти радиусы связаны между собой через матрицу преобразования координат M_i следующим уравнением:

$$\bar{\mathbf{r}}_{M_{i-1}} = M_i \cdot \bar{\mathbf{r}}_{M_i},$$

где M_i — матрица перехода из i -той системы координат в $(i - 1)$ -ую.

$$M_i = \begin{vmatrix} \cos \varphi_i & -\cos \theta_i \cdot \sin \varphi_i & \sin \varphi_i \cdot \sin \theta_i & a_i \cdot \cos \varphi_i \\ \sin \varphi_i & \cos \theta_i \cdot \cos \varphi_i & -\cos \varphi_i \cdot \sin \theta_i & a_i \cdot \sin \varphi_i \\ 0 & \sin \theta_i & \cos \theta_i & s_i \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

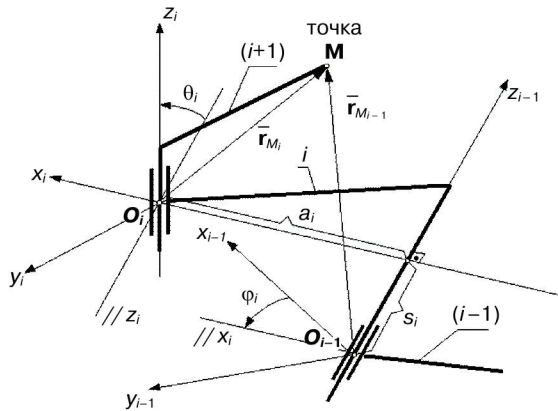


Рис. 3. Переход между системами координат

За начальное положение манипулятора принимается такое, в котором все относительные обобщенные координаты равны нулю. Переход из системы координат i -того звена к неподвижной или базовой системе записывается в виде

$$\bar{\mathbf{r}}_{M_0} = M_1 \cdot M_2 \cdot \dots \cdot M_i \cdot \bar{\mathbf{r}}_{M_i} \text{ или } \bar{\mathbf{r}}_{M_0} = T_i \cdot \bar{\mathbf{r}}_{M_i},$$

где $T_i = M_1 \cdot M_2 \cdot \dots \cdot M_i$ — матрица преобразования координат i -той системы в координаты базовой системы координат.

Радиус $r_{M_6} = 0$, а радиус r_{M_0} определяется по формуле

$$\bar{\mathbf{r}}_{M_0} = T_n = T_6,$$

т.е. положение выходного звена манипулятора определяется матрицей T_n . Элементы этой матрицы определяют положение центра схвата точки M и ориентацию его в пространстве. Четвертый столбец определяет декартовы координаты точки M (проекции вектора r_{M_0} на оси координат). Третий столбец содержит направляющие косинусы оси z_n системы координат, связанной со схватом, или вектора подхода \bar{A} , который характеризует направление губок схвата (рис. 4). Второй столбец

определяет направление оси y_n или вектора ориентации \bar{O} , который проходит через центр схвата по оси перпендикулярной рабочим поверхностям его губок. В первом столбце содержатся направляющие косинусы оси x_n или вектора $(\bar{O} \times \bar{A})$. Углом подхода схвата α называется угол между вектором подхода \bar{A} и базовым вектором $R_\alpha = (\bar{O} \times \bar{k}_0)$, где \bar{k}_0 — орт вектора \bar{z}_0 неподвижной или базовой системы координат. С учетом сказанного матрица T_n может быть представлена в следующем виде (рис. 4) и является решением прямой задачи кинематики робота-манипулятора

$$T_n = \begin{vmatrix} (\bar{O} \times \bar{A})_x & O_x & A_x & r_n M_x \\ (\bar{O} \times \bar{A})_y & O_y & A_y & r_n M_y \\ (\bar{O} \times \bar{A})_z & O_z & A_z & r_n M_z \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

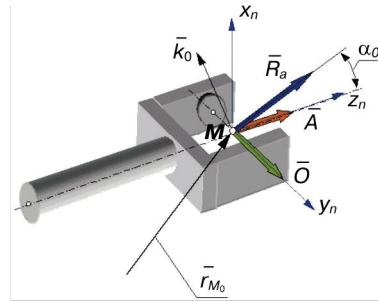


Рис. 4. Матрица схвата манипулятора

1.2. Обратная задача кинематики

В обратной — по заданному закону движения схвата находят законы изменения координат звеньев, обычно, линейных или угловых перемещений в приводах. Эту задачу можно сформулировать так: задана кинематическая схема манипулятора и известны положение и ориентация схвата в системе координат стойки. Требуется определить значения обобщенных координат, которые обеспечат заданное положение схвата.

Задать положение схвата, как и любого твердого тела, можно с помощью шести величин. Обычно первые три из них — это координаты центра схвата, затем две — это направляющие косинусы одной из координатных осей схвата и последняя — это один из направляющих косинусов другой координатной оси схвата. Этими шестью величинами могут быть шесть наддиагональных элементов матрицы T_n :

$$T_n = \begin{vmatrix} \cdot & (t_n)_{12} & (t_n)_{13} & (t_n)_{14} \\ \cdot & \cdot & (t_n)_{23} & (t_n)_{24} \\ \cdot & \cdot & \cdot & (t_n)_{34} \\ \cdot & \cdot & \cdot & \cdot \end{vmatrix}$$

Приравняв шесть заданных величин соответствующим элементам матрицы T_n , получим систему шести уравнений (в общем случае трансцендентных), неизвестными в которых являются обобщенные координаты.

Если $n = 6$, т. е. число неизвестных равно числу уравнений, то обычно можно отыскать вполне определенные значения обобщенных координат.

Если манипулятор имеет больше шести степеней свободы, т.е. число неизвестных превышает число уравнений, то одному и тому же положению схвата могут соответствовать различные наборы значений обобщенных координат.

И, наконец, если $n < 6$, то решения не существует, т.е. за счет меньшего, чем шесть, числа обобщенных координат невозможно получить наперед заданные произвольные положение и ориентацию схвата.

Однако если требуется лишь попадание центра схвата в определенную точку пространства, а ориентация схвата может быть любой, то для этой цели годится манипулятор с тремя степенями свободы. В этом случае при решении задачи потребуется составить лишь три уравнения для отыскания трех неизвестных.

Не всегда обратную задачу кинематики удастся решить аналитически. Приведем простейший алгоритм численного решения системы трансцендентных уравнений.

Задача состоит в отыскании значений m обобщенных координат по заданным m элементам матрицы T_m причем $m \leq 6$. Для простоты изложения будем считать, что манипулятор обладает шестью степенями свободы и $m = n = 6$. Тогда искомыми являются все обобщенные координаты манипулятора, которые обозначим q_1, q_2, \dots, q_6 . Заданными считаем шесть наддиагональных элементов матрицы T_n . Значения q_i определим методом последовательных приближений.

В нулевом приближении дадим обобщенным координатам произвольные начальные значения $q_1^{(0)}, q_2^{(0)}, \dots, q_6^{(0)}$.

Постановка задачи разбиения. Целью работы является определение метода оптимальной подготовки данных для последующего параллельного распределения подзадач модульного алгоритма по процессорам вычислительной системы типа коллективного решения с общей физической памятью (пять процессорных элементов (ПЭ): один управляющий и четыре рабочих). Максимально возможное количество процессорных элементов в такой архитектуре ограничено: оно может быть не более шести.

Управляющий процессорный элемент определяет порядок выполнения подзадач модульного алгоритма, а рабочие выполняют непосредственно вычисления. При подготовке проводится также анализ зависимости выделяемых подзадач друг от друга. Используется архитектура с общей физической памятью, потому что в ней снижены затраты на обмен данными между рабочими процессорами, которые обязательно будут происходить, из-за наличия зависимости между подзадачами.

Верхний уровень вычислительной модели системы коллективного решения с общей физической памятью представлен на рис. 5 в виде структурной схемы (процессоров, памяти, каналов передачи данных), определяющей типы управления, размера модуля, коммуникационной структуры. Нижний уровень содержит ПЭ с вычислительной моделью современного транспьютера — транспьютероподобных ЦСП (ТП ЦСП).

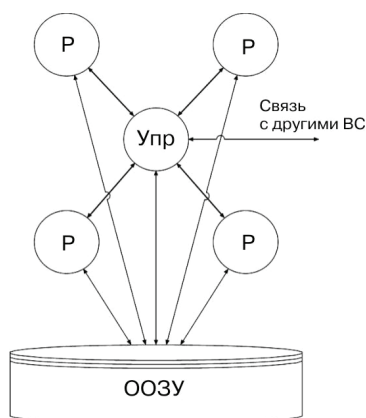


Рис. 5. Вычислительная система коллективного решения с общей физической памятью

Функционирование модели такой ВС можно представить как параллельную обработку n битовых слоев (число n бит в машинном слове, обрабатываемых параллельно при выполнении машинных инструкций), на каждом из которых независимо преобразуются m бит (число слов, обрабатываемых одновременно данной ВС). Произведение $P = n \cdot m$ согласно классификации Фенга определяет интегральную характеристику потенциала параллельности архитектуры или максимальную степень параллелизма данной ВС.

При разработке методики рассматриваются алгоритмы, реализующие различные практические задачи (с разными степенями параллелизма), для выявления различий в методиках разбиения

на подзадачи. Алгоритмы рассматриваются с точки зрения структуры как самого алгоритма, так и его программной реализации (последовательный алгоритм). Рассмотрим алгоритмы приведенных выше прикладных задач (прямая и обратная задачи кинематики, перемножение матриц).

1. Параллелизм по данным (перемножение матриц)

Как уже упоминалось, единичная операция, которая выполняется в этом алгоритме N раз — это перемножение отдельных элементов (a, b) двух матриц и прибавление полученного результата к значению элемента (c) третьей матрицы, в которой содержится решение. Выполнение каждой из этих операций не зависит от других, т.е. результатом разбиения является множество не зависящих друг от друга подзадач, каждая подзадача — единичная операция на некотором наборе данных. Разбиение такого алгоритма строится на анализе его собственной структуры и практически не зависит от программной реализации.

2. Функциональный параллелизм (прямая и обратная задача кинематики)

В случае функционального параллелизма результатом разбиения будет набор подзадач и их зависимостей друг от друга. При разбиении таких алгоритмов необходимо оценивать не только структуру алгоритма, но и его программную реализацию, потому что она может оказывать влияние на зависимость подзадач друг от друга.

Возьмем в качестве примера алгоритм решения обратной задачи кинематики (с точностью 0,001):

- 1) считать начальные данные (матрица T_6 , известные параметры манипулятора);
- 2) задать начальные приближенные значения $q_1^{(0)}, q_2^{(0)}, \dots, q_6^{(0)}$;
- 3) присвоить $q_j^{(k-1)}$ текущие приближенные значения;
- 4) вычислить матрицы $T_6^{(k-1)}$ и $U_6^{(k-1)}$;

5) вычислить $q_j^{(k)}$ по методу Крамера, используя систему уравнений:

$$(t_6)_{12} = (t_6)_{12}^{(k-1)} + \sum_{j=1}^6 (U_{6j})_{12}^{(k-1)} (q_j^{(k)} - q_j^{(k-1)}),$$

$$(t_6)_{13} = (t_6)_{13}^{(k-1)} + \sum_{j=1}^6 (U_{6j})_{13}^{(k-1)} (q_j^{(k)} - q_j^{(k-1)}),$$

$$(t_6)_{14} = (t_6)_{14}^{(k-1)} + \sum_{j=1}^6 (U_{6j})_{14}^{(k-1)} (q_j^{(k)} - q_j^{(k-1)}),$$

$$(t_6)_{23} = (t_6)_{23}^{(k-1)} + \sum_{j=1}^6 (U_{6j})_{23}^{(k-1)} (q_j^{(k)} - q_j^{(k-1)}),$$

$$(t_6)_{24} = (t_6)_{24}^{(k-1)} + \sum_{j=1}^6 (U_{6j})_{24}^{(k-1)} (q_j^{(k)} - q_j^{(k-1)}),$$

$$(t_6)_{34} = (t_6)_{34}^{(k-1)} + \sum_{j=1}^6 (U_{6j})_{34}^{(k-1)} (q_j^{(k)} - q_j^{(k-1)});$$

6) сравнить $q_j^{(k-1)}$ и $q_j^{(k)}$;

7) если разность по модулю, полученная в пункте 6 больше 0,001, то перейти к п. 3, в ином случае перейти к пункту 8;

8) принять полученные приближенные значения как решение обратной задачи кинематики и вывести их на экран.

Схематически этот алгоритм можно изобразить следующим образом (рис. 6):

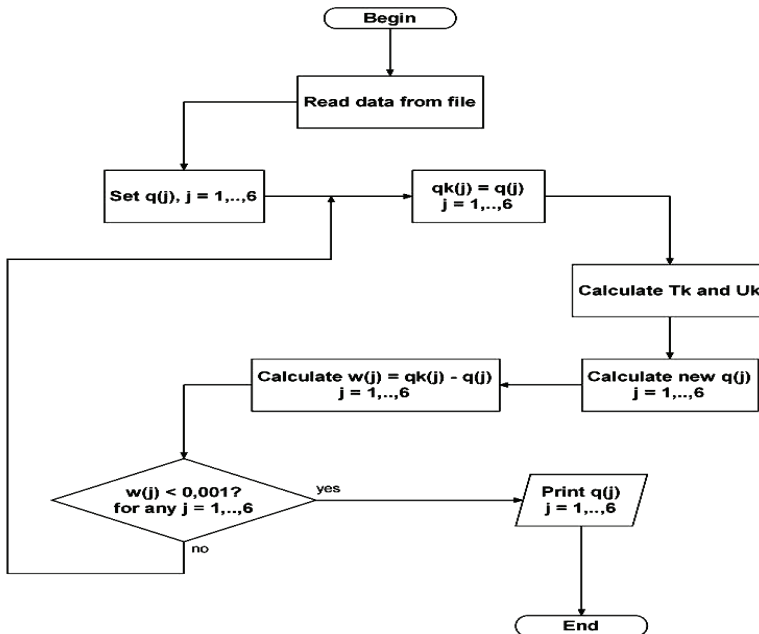


Рис. 6. Схематическое изображение алгоритма обратной задачи кинематики

Анализ схемы позволяет выделить отдельные подзадачи, выполняющиеся в цикле N раз (расчет матриц T и U , вычисление набора q_j , оценка точности) последовательно. Таким образом, разбиение вида — единичная операция: один шаг цикла — невозможно, потому что два шага цикла не могут рассчитываться одновременно. Необходимо более подробное разбиение также и этих подзадач для каждого шага цикла. Оно может выглядеть следующим образом (рис. 7, на схеме показаны также подзадачи прямой задачи кинематики, выполнение которой подготавливает данные для решения обратной задачи). Подзадачи с 3-ей по 15-ю являются составными частями одной итерации и выполняются для данных начальных условий N раз, где N зависит от заданной точности и начального набора координат (q_0). Итерации выполняются последовательно.

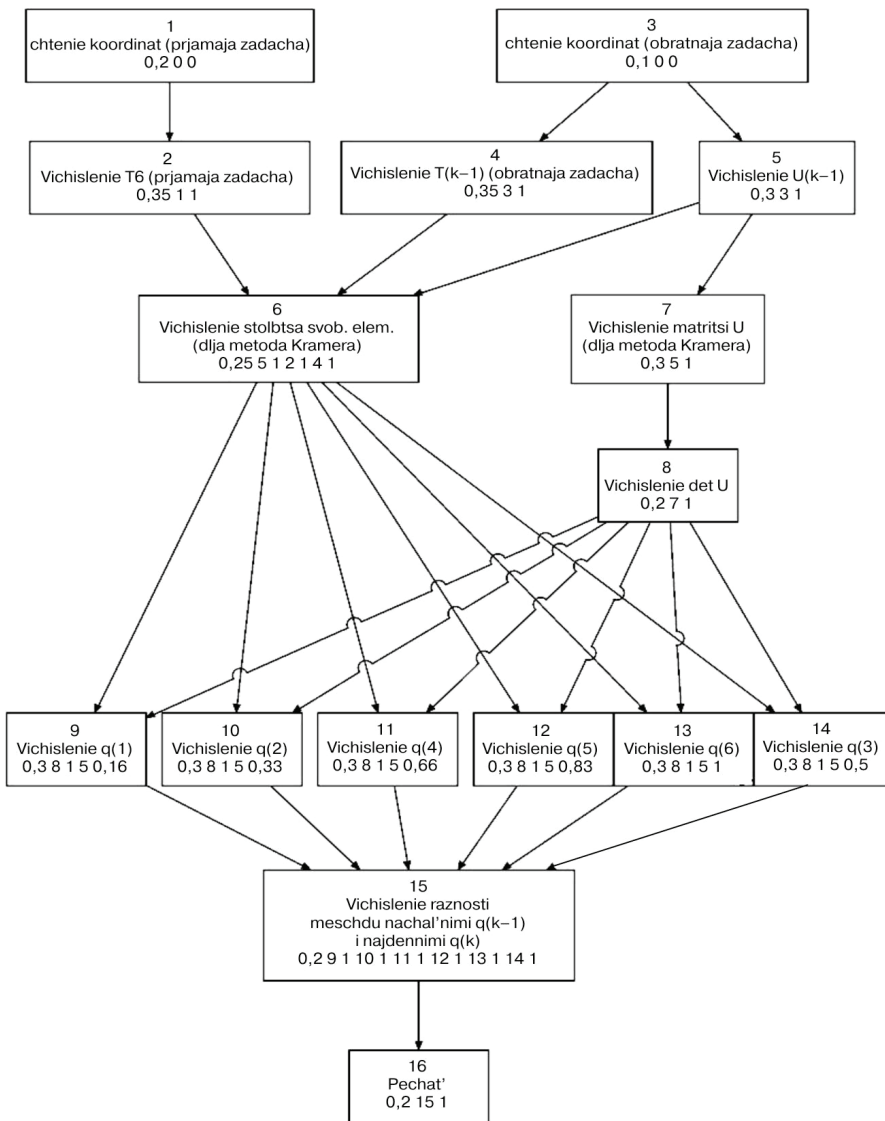


Рис. 7. Зависимость подзадач обратной задачи кинематики

На рис. 7 приведены также время выполнения подзадачи, ее зависимость от других. Распределение данных подзадач, полученное с помощью реализованного на языке Java алгоритма, дало по минимальным значениям замеров времени уменьшение времени работы по сравнению с последовательным алгоритмом с 14,95 мсек до 7,2 мсек, т.е. до 48%. Язык, на котором был реализован алгоритм, был выбран из-за своей кросс-платформенности.

Полученный результат согласуется с ранее опубликованными исследованиями [1; 4], в которых распределение подзадач по процессорам вычислительной системы с локальной памятью осуществлялось эвристически вручную путем разбиения последовательного алгоритма на модули по их функциональной принадлежности и загрузки в разные процессоры для параллельного выполнения. Балансировка загруженности процессоров производилась также вручную.

Некоторые из полученных результатов были также представлены в материалах международных симпозиумов и конференций, состоявшихся в 2008 г. [5; 6].

Оценка эффективности. При оценке эффективности использованы замеры времени. В качестве характеристик эффективности реализации параллельных алгоритмов можно использовать следующие:

- 1) абсолютное время выполнения;
- 2) ускорение на N процессорах сети по сравнению с одним;
- 3) загруженность вычислительной работой транспьютеров;
- 4) качественная оценка достигнутого ускорения;
- 5) стоимость реализации.

Для оценки эффективности полученного разбиения используется качественная оценка достигнутого ускорения при выполнении распределенного алгоритма на многопроцессорной системе, как наиболее наглядная характеристика (рис. 8).

Функции $\hat{y} = N$, $\hat{y} = N/\ln N$, $\hat{y} = \log_2 N$ позволяют разделить координатную плоскость, образованную осями N (количество процессорных элементов) и \hat{y} (оценка ускорения вычисления), и классифицировать достигнутое ускорение и сами области как области: недостижимого, сомнительного, оптимистичного и пессимистичного ускорения.

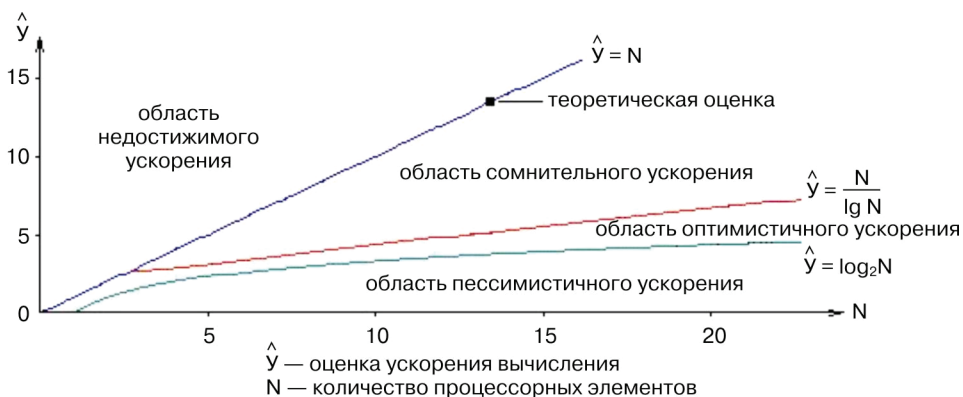


Рис. 8. Графическое представление качественной оценки ускорения

Точка с координатами (N, \hat{y}) , принадлежащая одной из областей, однозначно определяет принадлежность реализации ускорения \hat{y} на N процессорных элементах к одной из этих областей и, соответственно, к одному из классов оценки ускорения. Если достигнутое ускорение попадает в область оптимистичных ускорений, то полученное разбиение последовательного алгоритма может считаться приемлемым.

Вычислительный эксперимент с использованием разработанной методики показал достижение ускорения в 2,076 раза при выполнении на четырех процессорах, что соответствует области оптимистического ускорения ($2 \leq \hat{y} \leq 2,885$ при $N = 4$) (см. рис. 8), и позволяет сделать вывод о достаточно хорошем качестве алгоритма распределения подзадач по процессорам вычислительной системы типа коллективного решения с общей памятью.

Оценка ускорения вычислений \hat{y} по замерам времени выполнения программной реализации последовательного и имитации параллельного алгоритма на ПК под управлением многозадачной ОС Windows XP является не совсем корректной, так как программа выполняется на фоне процессов, запущенных ОС. Чем меньше системных и прикладных процессов будет запущено, тем более точными окажутся замеры времени выполнения последовательной программы и ее модулей. При замерах времени на фоне длительных прикладных процессов (создание архивов) замеры времени увеличиваются, а вычисляемая оценка \hat{y} занижается. Минимальные значения замеров времени выполнения последовательной программы и ее распределяемых модулей в условиях выполнения замеров каждый раз на фоне разного неизвестного числа процессов покажут результат ускорения, наиболее приближенный к действительному. Поэтому правомерно предположить, что реальная «чистая» оценка ускорения \hat{y} будет несколько лучше (выше) приведенной оценки ускорения по методике использования минимальных значений замеров времени с учетом вклинивающихся процессов в 2,076 раза.

$$\hat{y} = \frac{\min(T_{\text{послед}} + \sum T_{\text{другие процессы 1}})}{\min(T_{\text{паралл}} + \sum T_{\text{другие процессы 2}})}$$

Ускорение вычислений получено по разработанной методике, основанной на выделении подзадач модульного алгоритма и их зависимости друг от друга, и автоматизации балансировки загрузки процессоров вычислительной нагрузкой до начала вычислений.

Заключение. Рассмотрена постановка задачи разработки методики выделения отдельных связанных между собой подзадач в последовательном алгоритме, а также определения степени их зависимости друг от друга. Задача решается в рамках разработки многопроцессорной системы коллективного решения с общей физической памятью и программного комплекса по автоматической подготовке данных для ее работы в составе робототехнического комплекса.

Решение поставленной задачи предусматривает научное использование нового оборудования и программ учебного робототехнического комплекса Kawasaki — EDU-Pack System в лаборатории Мехатронных систем и робототехники кафедры кибернетики и мехатроники РУДН.

ЛИТЕРАТУРА

- [1] *Внуков А.А., Родкина Л.В.* Результаты моделирования и исследования алгоритмов параллельных вычислений прямой и обратной задач кинематики манипулятора «Системные проблемы качества, математического моделирования и информационных технологий»: Материалы Международной конференции и Российской научной школы. Сочи, 1 окт. — 10 окт. 2000. Часть 4. — М.: НИИ «Автоэлектроника», 2000. — С. 35—38.
- [2] Патент US4493048 (А). Дата публикации: 1985-01-08 Изобретатель(и): Kung Hsiang-Tsung [US]; Leiserson Charles E [US] Заявитель(и): UNIV CARNEGIE MELLON [US].
- [3] *Внуков А.А., Гайсина Л.Ф., Селескерова Е., Сударкина Е.В.* Разработка спецвычислителя для прямого преобразования координат «Интеллектуальные системы» // Труды Седьмого Международного симпозиума (Краснодар, 26 июня — 30 июня 2006 г.) / Под ред. К.А. Пупкова. — М.: РУСАКИ, 2006. — С. 507—510.
- [4] *Зайцева Е.В., Внуков А.А.* Разработка универсальной вычислительной среды для реализации параллельных алгоритмов робота-манипулятора на современной элементной базе: Труды III Международной конференции «Параллельные вычисления и задачи управления» РАСО'2006 памяти И.В. Прангишвили. Москва 2—4 октября 2006 г. Институт Проблем управления им. В.А. Трапезникова РАН. — М.: Институт проблем управления им. В.А. Трапезникова РАН, 2006. — С. 999—1024.
- [5] *Зазулина А.С., Внуков А.А.* Распределение модульного последовательного алгоритма для интеллектуальной системы коллективного решения «Интеллектуальные системы» // Труды Восьмого Международного симпозиума (Н.Новгород, 30 июня — 4 июля 2008 г.) / Под ред. К.А. Пупкова. — М.: РУСАКИ, 2008. — С. 187—191.
- [6] *Зазулина А.С., Внуков А.А.* Разработка алгоритмов и программного обеспечения для параллельного распределения для вычислительных систем типа коллективного решения: Труды Четвертой Международной конференции «Параллельные вычисления и задачи управления» РАСО'2008. Москва, 27—29 октября 2008 г. Институт Проблем управления им. В.А. Трапезникова РАН. — М.: Институт проблем управления им. В.А. Трапезникова РАН, 2008. — С. 1049—1070.

DEVELOPMENT OF THE TECHNIQUE OF SPLITTING OF THE SUBTASKS LINKED AMONG THEMSELVES FOR PARALLEL EXECUTION OF KINEMATIC CONVERSIONS OF THE MANIPULATOR

A.A. Vnukov, A.S. Zazulina

In article parallelism sorts in modular algorithms of direct and return tasks of kinematics of the robot-manipulator, possible techniques of their splitting into separate subtasks for the subsequent allocation of these subtasks on in advance known multiprocessor structure of the total memory with the integral characteristic of potential of parallelism of the architecture are considered. The method of an splitting efficiency estimation is offered.

Key words: Kinematic structure, modular algorithm, robot-manipulator.