

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Саратовский национальный исследовательский
государственный университет имени Н.Г. Чернышевского»

Кафедра Математического обеспечения
вычислительных комплексов
и информационных систем

**ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ МОДЕЛИРОВАНИЯ
ПЕРЕХОДНЫХ ПРОЦЕССОВ В НЕЛИНЕЙНЫХ
КОМБИНИРОВАННЫХ ДИНАМИЧЕСКИХ СИСТЕМАХ**

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 273 группы
направления 01.04.02 – Прикладная математика и информатика (ПМИ)
факультет Компьютерных наук и информационных технологий (КНиИТ)
Волгуцкова Алексея Александровича

Научный руководитель
профессор, д. ф.-м. н. _____

Андрейченко Д.К.

Зав. кафедрой _____

Андрейченко Д.К.

Саратов 2017

Введение.

Комбинированные динамические системы (КДС) представляют собой математические модели в форме обыкновенных дифференциальных уравнений и связанных с ними посредством граничных условий и условий связи уравнений в частных производных при соответствующих начальных условиях. В данных математических моделях, обыкновенные дифференциальные уравнения моделируют динамическое поведение объектов с сосредоточенными по пространству параметрами, а уравнения в частных производных моделируют динамику объектов с распределёнными по пространству параметрами. При численном моделировании переходных процессов в нелинейных КДС обычно выполняется дискретизация по независимым пространственным координатам модельных уравнений в частных производных на основе проекционного метода Галёркина, в результате чего численное моделирование сводится к численному решению жестких систем обыкновенных дифференциальных уравнений достаточно большой размерности. Независимость вычисления скалярных произведений в проецированном методе Галёркина гарантирует возможность распараллеливания алгоритма вычисления правых частей обыкновенных дифференциальных уравнений. Вместе с тем, система обыкновенных дифференциальных уравнений обычно оказывается «жёсткой», а для её численного интегрирования применяют «жёстко устойчивые» методы, например, ФДН-метод, однако их применение вынуждает вычислять матрицу Якоби правых частей уравнений. Хорошо известно, что наибольшие затраты компьютерного времени при реализации «жёстко устойчивых» методов связаны именно с вычислением матрицы Якоби. Дополнительный ресурс параллелизма в данном случае состоит в том, что как следует из независимости вычисления скалярных произведений методом Галёркина, столбца матрицы вычисляются независимо, что также может быть распараллелено. Представляет интерес дополнить данную схему распараллеливания

распараллеливанием вычисления правых частей обыкновенных дифференциальных уравнений.

Ранее были предложены эффективные реализации данной схемы распараллеливания на основе многопоточности с использованием OpenMP 3.x либо Microsoft Concurrency Runtime, поскольку стандартные средства данных технологий обеспечивают динамическую балансировку вычислительной нагрузки.

Целью работы является распараллеливание данного класса задач на графических процессорах. А также сравнение эффективности данного алгоритма с методом конечных разностей. В качестве модельной задачи предполагается рассмотреть задачу программного управления подвижным объектом – манипулятором с учетом упругих деформаций элементов его конструкции.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

В **первой главе** рассмотрена структура математических моделей ряда современных технических систем, содержащих объекты управления с сосредоточенными по пространству параметрами и динамически связанные с ними через граничные поверхности объекты управления с распределенными по пространству параметрами, в форме КДС, т.е. в форме систем связанных посредством граничных условий и условий связи обыкновенных дифференциальных уравнений и уравнений в частных производных при соответствующих начальных условиях. Структурной схеме КДС на рис. 1 с кусочно-непрерывной входной вектор-функцией $\mathbf{x}(t) = (x_1(t), \dots, x_{N_x}(t))^T$ и непрерывной выходной вектор-функцией $\mathbf{y}(t) = (y_1(t), \dots, y_{N_y}(t))^T$ непрерывного времени t могут, например, соответствовать следующие модельные уравнения

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{x}, \mathbf{y}, \mathbf{h}) \quad (1.1)$$

$$\dot{\mathbf{u}} = \mathbb{F}(t, \mathbf{u}, \mathbf{x}, \mathbf{y}) + L(t, \mathbf{y}, \mathbf{u})\dot{\mathbf{y}}, \mathbf{r} \in \Omega; \quad (1.2)$$

$$\mathbb{G}(\mathbf{u}, \mathbf{y})|_S = 0; S = \partial\Omega; \quad (1.3)$$

$$\mathbf{h} = \int_S \mathbb{H}(\mathbf{u}) dS; \quad (1.4)$$

$$\mathbf{y}(0) = \mathbf{y}_0, \mathbf{u}(\mathbf{r}, 0) = \mathbf{u}_0(\mathbf{r}) \quad (1.5)$$

равновесному состоянию КДС. В равновесном состоянии $\dot{() = 0, \mathbf{x}_0 = Const$.

Были рассмотрена «жесткие» системы обыкновенных дифференциальных уравнений и ФДН-метод для решения этих уравнений. Пусть символ $\lambda_j, j = \overline{1, N}$ обозначает некоторое собственное значение матрицы J . Говорят, что система обыкновенных дифференциальных уравнений является «нежесткой», если $|\lambda_j| = O(1), j = \overline{1, N}$. Если же среди собственных значений есть как $|\lambda_j| = O(1)$, так и $|\lambda_j| \gg 1$, то задача является «жесткой», и для ее численного интегрирования используются специальные методы, например, метод, основанный на формулах «дифференцирования назад» (ФДН-метод, или метод Гира).

Рассмотрим для упрощения ФДН-метод при постоянном шаге $h_1 = h_2 = \dots = h_n = h_{n+1} = h$. Пусть $\lambda \in \mathbb{C}$ – некоторое собственное значение матрицы Якоби. Говорят, что метод численного интегрирования системы обыкновенных дифференциальных уравнений является A -устойчивым, если область S его численной устойчивости

$$S \supset \mathbb{C}^- = \{z : \operatorname{Re} z \leq 0\}, z = h\lambda$$

содержит левую комплексную полуплоскость. Это означает, что метод будет устойчив для любых значений шага h , если только все собственные значения матрицы Якоби $J = \partial \mathbf{f}(t, \mathbf{x}) / \partial \mathbf{x}$ лежат в левой комплексной полуплоскости.

Метод называется $A(\alpha)$ -устойчивым, если его область устойчивости содержит сектор комплексной плоскости

$$S \supset S_\alpha = \{z : |\arg(-z)| \leq \alpha\}, z = h\lambda$$

Это означает, что численное интегрирование можно выполнять с «относительно немалым» шагом h , если большие собственные значения λ , $|\lambda| \gg 1$ матрицы Якоби попадают в сектор S_α .

Во **второй** главе рассмотрено программное управление движением плоского двухзвенного манипулятора. Расчетная схема манипулятора приведена на рис. 2.

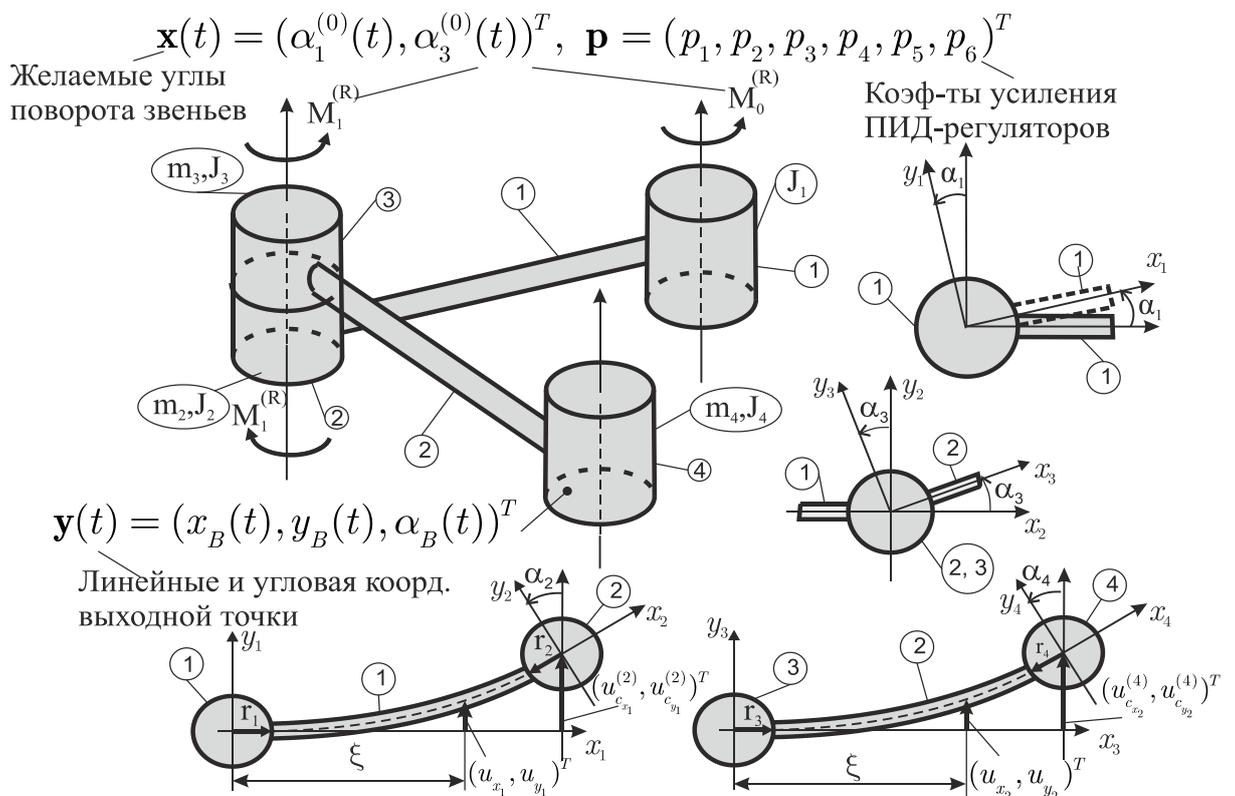


Рис. 2. Плоский двухзвенный манипулятор с упругими звеньями и ПИД-регуляторами

В данной комбинированной динамической модели компоненты входной вектор-функции $\mathbf{x}(t) = (\alpha_1^{(0)}(t), \alpha_3^{(0)}(t))^T$ суть безразмерные «желаемые» значения углов поворота упругих звеньев, а компоненты выходной вектор-функции $\mathbf{y}(t) = (x_B(t), y_B(t), \alpha_B(t))^T$ суть линейные и угловая координаты выходной точки манипулятора.

Так же были рассмотрены различные типы регуляторов. Регулятор — в теории управления устройство, которое следит за состоянием объекта управления как системы и вырабатывает для неё управляющие сигналы. Регуляторы следят за изменением некоторых параметров объекта управления и реагируют на их изменение с помощью некоторых алгоритмов управления в соответствии с заданным качеством управления. Регуляторы в подавляющем большинстве работают по принципу отрицательной обратной связи с целью компенсировать внешние возмущения, действующие на объект управления и отработать заданный извне или заложенный в системе закон управления (программу).

Далее были приведены модельное уравнение движения плоского

двухзвенного манипулятора с упругими звеньями и ПИД-регуляторами и ПД-регуляторами.

В третьей главе рассмотрены вычислительные устройства на которых был протестирован алгоритм используемый в работе. В частности, был рассмотрен графический процессор и его отличия от центрального. Технологии параллельных вычислений, такие как OpenMP и Microsoft Concurrency Runtime с помощью которых было произведено распараллеливание алгоритма, используемого в работе.

Ниже приводятся результаты численного моделирования координат выходной точки обоих плоских двухзвенных манипуляторов при их переводе из одного состояния в другое и последующем возврате в исходное состояние. В начальном состоянии направления осевых линий упругих звеньев составляют угол β (в радианах).

Далее показаны результаты моделирования координат выходной точки манипулятора с ПИД-регуляторами при программном управлении его движением рисунок 3. А так же и для манипулятора с ПД-регуляторами рисунок 4.

Из рисунков видно, что ПИД-регулятор обладает т.н. свойством астатизма, т.е. позволяет подавить нежелательную ошибку даже при постоянно действующем внешнем возмущении.

Параметры обратных связей для ПИД-регулятора и для ПД-регулятора были ранее выбраны на основе параметрического синтеза по линеаризованной математической модели согласно. Выбор параметров обратных связей при помощи параметрического синтеза по линеаризованной модели в данном случае позволяет значительно уменьшить отклонение координат выходной точки манипулятора от заданной траектории.

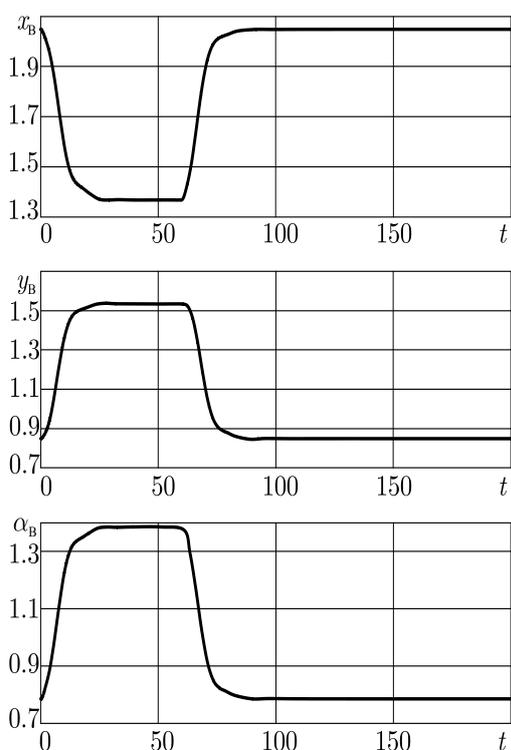


Рис. 3. Результаты моделирования координат манипулятора с ПИД-регуляторами

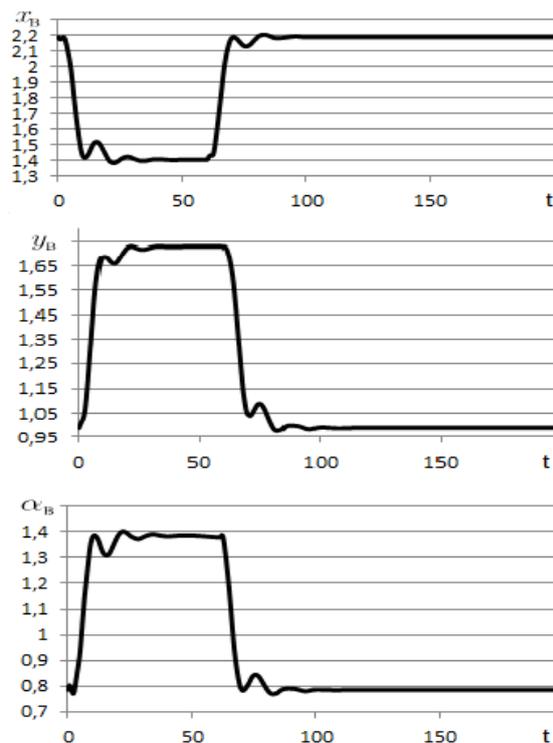


Рисунок 4. Результаты моделирования координат манипулятора с ПД-регуляторами.

В таблицах 1 и 2 указано время выполнения последовательной версии программы, и выполняется сравнение эффективности использованного в работе алгоритма вычисления матрицы Якоби по сравнению с традиционно используемым методом конечных разностей при решении задачи численного моделирования управляемого движения манипулятора с ПИД и ПД регуляторами на компьютере с процессором Intel i7 3610QM, в зависимости от числа $N_1 = N_2 = N$ слагаемых в формулах.

Таблица 1. Сравнение эффективности алгоритмов с ПИД регулятором, с

Процессор Intel Core i7 3610QM

| $N_1=N_2$ | Алгоритм представленный в работе | Метод конечных разностей |
|-----------|----------------------------------|--------------------------|
| 10 | 3,297 | 3,062 |
| 10 | 3,188 | 3,517 |
| 10 | 3,36 | 3,429 |

| | | |
|----|-------|--------|
| 10 | 3,219 | 3,534 |
| 10 | 3,157 | 3,452 |
| 15 | 6,031 | 67,969 |
| 15 | 7,265 | 67,908 |
| 15 | 6,093 | 68,153 |
| 15 | 7,5 | 65,784 |
| 15 | 7,86 | 66,257 |

Таблица 2. Сравнение эффективности алгоритмов с ПД регулятором, с
Процессор Intel Core i7 3610QM

| $N_1=N_2$ | Алгоритм представленный в работе | Метод конечных разностей |
|-----------|----------------------------------|--------------------------|
| 10 | 3,103 | 4,531 |
| 10 | 3,021 | 4,289 |
| 10 | 3,347 | 4,608 |
| 10 | 3,459 | 4,845 |
| 10 | 3,2 | 4,169 |
| 15 | 6,765 | 71,29 |
| 15 | 7,018 | 75,09 |
| 15 | 6,79 | 73,944 |
| 15 | 6,35 | 69,294 |
| 15 | 6,742 | 72,09 |

Как видно из таблицы 1 и 2 применение используемого в работе алгоритма является более эффективным по сравнению с методом конечных разностей.

Данные в таблицах 3 и 4 характеризуют эффективность распараллеливания вычислений на основе предложенного алгоритма при решении задачи численного моделирования управляемого движения манипулятора с ПИД и ПД регуляторами на компьютере с процессором Intel i7 3610QM, а так же с графическим ускорителем GeForce GTX 750Ti, в

зависимости от числа $N_1 = N_2 = N$ слагаемых в формулах

$$u_y(\xi, t) = \sum_{n=0}^{N_j+4} u_{y_n}(t) T_n(2\xi / \ell_j - 1)$$

$$Q_1(\xi, t) = \sum_{n=0}^{N_j+2} Q_{1_n}(t) T_n(2\xi / \ell_j - 1) .$$

$$T_n(x) = \cos(n \arccos x), j = 1, 2, \ell_1 = 1$$

В таблице указано среднее время выполнения последовательного алгоритма и его параллельных трёх реализаций на CPU и одной реализации на GPU.

Таблица 3. Манипулятор с ПИД-регуляторами, с
Процессор Intel Core i7 3610QM и GeForce GTX 750Ti

| $N_1=N_2$ | Последовательная версия | Распараллеливание в стиле портфель задач | Динамическая балансировка | MS Concurrency | CUDA |
|-----------|-------------------------|--|---------------------------|----------------|----------|
| 10 | 3,2442 | 1,847 | 1,7724 | 1,0436 | 240.528 |
| 15 | 6,9498 | 3,3032 | 3,2404 | 1,747 | 462.522 |
| 20 | 11,8436 | 5,975 | 5,4812 | 2,8278 | 576.4726 |
| 25 | 23,86 | 9,2092 | 9,1938 | 4,778 | 921.12 |

Таблица 4. Манипулятор с ПД-регуляторами, с
Процессор Intel Core i7 3610QM и GeForce GTX 750Ti

| $N_1=N_2$ | Последовательная версия | Распараллеливание в стиле портфель задач | Динамическая балансировка | MS Concurrency | CUDA |
|-----------|-------------------------|--|---------------------------|----------------|----------|
| 10 | 3,226 | 1,4246 | 1,3062 | 1,1498 | 290,3197 |
| 15 | 6,733 | 2,61 | 2,3934 | 2,0436 | 600,734 |
| 20 | 12,6804 | 4,5374 | 4,6874 | 3,7284 | 764,31 |
| 25 | 20,2126 | 7,0216 | 7,726 | 6,1478 | 1261.238 |

Как видно из таблицы 3 и 4 вычисление данного алгоритма при помощи NVidia CUDA занимает значительно больше времени по сравнению с его

последовательным выполнением на центральном процессоре. Следовательно, его выполнение на графическом процессоре является не эффективным.

Заключение.

Численное моделирование переходных процессов в нелинейных комбинированных динамических системах характеризуется дополнительным ресурсом параллелизма при применении проекционного метода Галеркина и «жестко устойчивых» методов численного интегрирования обыкновенных дифференциальных уравнений, связанным с независимостью вычисления отдельных столбцов матрицы Якоби.

Предложенный параллельный алгоритм позволяет успешно распараллеливать численное моделирование выходных вектор-функций в нелинейных КДС на центральных процессорах.

Распараллеливание на основе предложенного метода позволяет существенно ускорять процесс численного моделирования в тех случаях, когда другие методы, например, использующие распараллеливание правых частей обыкновенных дифференциальных уравнений, оказываются неэффективными.

Реализация данного алгоритма на графическом процессоре не эффективна из-за того, что идёт постоянная пересылка данных между центральным и графическим процессорами.

Распараллеливание на основе предложенного метода заметно эффективней по сравнению с методом конечных разностей.