

УДК 517.977:519.68

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ УЛУЧШЕНИЯ УПРАВЛЕНИЯ

М.Р. Коваленко

Институт программных систем РАН

Россия, 152020, Переславль-Залесский, м. Ботик

E-mail: madmax@botik.ru

Г.А. Матвеев

Институт программных систем РАН

Россия, 152020, Переславль-Залесский, м. Ботик

E-mail: gera@prime.botik.ru

В.И. Осипов

Институт программных систем РАН

Россия, 152020, Переславль-Залесский, м. Ботик

E-mail: val@pereslavl.ru

Е.А. Трушкова

Институт программных систем РАН

Россия, 152020, Переславль-Залесский, м. Ботик

E-mail: katerina@tea.pereslavl.ru

Ключевые слова: задача оптимального управления, метод улучшения, параллельный алгоритм, T-система, OpenTS, язык программирования T++

Key words: optimal control problem, improvement method, parallel algorithm, T-system, OpenTS, programming language T++

Представлены результаты разработок алгоритмической и программной части программного комплекса улучшения и оптимизации законов управления для приложений в различных областях (ПК ISCON), реализующей параллельный алгоритм итерационного улучшения начального приближения управления динамической системы. Применение алгоритма и анализ эффективности распараллеливания демонстрируется на примере исследования маневров нештатной посадки вертолета с определением границы безопасной зоны [1].

A PARALLEL ALGORITHM OF CONTROL IMPROVEMENT / M.P. Kovalenko (Program Systems Institute, Pereslavl-Zalessky, 152020, Russia, E-mail: madmax@botik.ru), G.A. Matveev (Program Systems Institute, Pereslavl-Zalessky, 152020, Russia, E-mail: gera@prime.botik.ru), V.I. Osipov (Program Systems Institute, Pereslavl-Zalessky, 152020, Russia, E-mail:), E.A. Trushkova (Program Systems Institute, Pereslavl-Zalessky, 152020, Russia, E-mail: katerina@tea.pereslavl.ru). There are presented results of algorithmic and program code part of a program code complex "Improvement and Synthesis of Control" (PC ISCON). There is sufficiently universal iterative improvement algorithm of initial approximate control oriented to parallel computations. Algorithm application and performance analysis of parallel program are shown by investigation of safe non-standard landing maneuvers performed by a helicopter with determination of a safe zone as an example [1].

1. Введение

Разработка и реализация методов синтеза оптимальных целевых законов управления является кардинальной проблемой теории и практики управления. Ее решение связано с бесконечномерными обратными задачами, аппроксимируемыми при численной реализации многомерными задачами, требующими неограниченно растущих вычислительных ресурсов для приближения к точным решениям. Это приводит к неизбежному выводу о необходимости реализации методов синтеза управления на современных высокопроизводительных вычислительных системах параллельной архитектуры. Парадигма параллельных вычислений в высшей степени соответствует самой природе рассматриваемых задач, связанных с множественностью однотипных операций на верхнем уровне, таких как:

- решение обратной задачи через множество решений прямой (прямо или косвенно),
- формирование и численная реализация полей управлений (ситуационных управлений).

Представленный алгоритм и его программная реализация являются результатами разработок части программного комплекса улучшения и оптимизации законов управления для приложений в различных областях (ПК ISCON), реализующей параллельный алгоритм улучшения начального управления динамической системы. Предложен достаточно универсальный алгоритм итерационного улучшения начального приближения управления, ориентированный на параллельные вычисления, который, в отличие от известных процедур такого рода, может оперировать не только с аналитическими аппроксимациями любой сложности, но и непосредственно с исходной имитационной моделью. Последнее необходимо для практической реализации полученного решения.

Применение алгоритма и анализ эффективности его распараллеливания в рамках Т-системы демонстрируется в вычислительных экспериментах на примере исследования маневров безопасной нештатной посадки реального вертолета с определением границы безопасной зоны [1].

2. Постановка задачи

Предполагается, что модель динамической управляемой системы представлена как дискретная во времени (что, как правило, выполняется при практической численной реализации). Рассматривается задача оптимального управления в стандартной форме:

$$(1) \quad \begin{aligned} x(t+1) &= f(t, x(t), u(t)), t \in T = \{0, 1, \dots, t_F\}, \quad x \in R^n, u \in R^p, \\ x(0) &= x_0, \quad x_{-F} \leq x(t_F) \leq x_{+F}, \quad u(t) \in D_u = \{u(t) \mid u_- \leq u(t) \leq u_+\}, \\ x_- \leq x(t) &\leq x_+, t \in T \setminus \{t_F\}, \quad F_0(x(t_F)) \rightarrow \min. \end{aligned}$$

Ограничения фазовых переменных снимаются с помощью замены задачи (1) оштрафованной задачей следующего вида:

$$\begin{aligned}
 (2) \quad & x(t+1) = f(t, x(t), u(t)), \quad t \in T = \{0, 1, \dots, t_F\}, \\
 & z(t+1) = z(t) + t_F^{-1} \delta(x(t)), \\
 & x(0) = x_0, \quad z(0) = 0, \quad u(t) \in D_u, \\
 & F(x(t_F), z(t_F)) = \beta_0 F_0(x(t_F)) + \beta_1^T z(t_F) + \beta_2^T \delta_F(x(t_F)) \rightarrow \min,
 \end{aligned}$$

где

$$\begin{aligned}
 \delta^i(x) &= -\min\{0, x^i - x_-^i\} + \max\{0, x^i - x_+^i\}, \\
 \delta_F^i(x) &= -\min\{0, x^i - x_{-F}^i\} + \max\{0, x^i - x_{+F}^i\}, \\
 i &= 1, \dots, n, \quad \beta_0 \in R, \quad \beta_1, \beta_2 \in R^n.
 \end{aligned}$$

Связанная с полученной системой задача улучшения ставится следующим образом: имеется начальное приближенное решение задачи (2) – элемент

$$\begin{aligned}
 m^I &= (x^I(t), u^I(t)), \quad u^I \in D_u, \quad x^I \in D_x(u^I), \\
 D_x(u^I(t)) &= \{x(t) \mid x(0) = x_0, x(t+1) = f(t, x(t), u^I(t))\}
 \end{aligned}$$

требуется найти элемент

$$m^{II} = (x^{II}(t), u^{II}(t)), \quad u^{II} \in D_u, \quad x^{II} \in D_x(u^{II}),$$

такой, что

$$F(x^{II}(t_F), z^{II}(t_F)) < F(x^I(t_F), z^I(t_F)).$$

3. Аналитическое описание алгоритма

Общие конструкции метода улучшения управления приведены в [2], где на основе принципа оптимальности Кротова элемент m^{II} ищется путем аппроксимации решения следующей задачи:

$$\begin{aligned}
 (3) \quad & y(t+1) = g(t, y(t), v(t)), \quad t \in T = \{0, 1, \dots, t_F\}, \\
 & s(t+1) = g_s(t, y, s) = s(t) + t_F^{-1} \delta(y(t) + x^I(t)) - t_F^{-1} \delta(x^I(t)), \\
 & y^0(t+1) = g_0(y^0, v) = y^0(t) + 0.5v^T(t)v(t), \\
 & y(0) = 0, \quad s(0) = 0, \quad y^0(0) = 0, \quad v(t) + u^I(t) \in D_u, \\
 & G_\alpha(y(t_F), s(t_F), y^0(t_F)) = \alpha y^0(t_F) + \\
 & + (1 - \alpha) F(y(t_F) + x^I(t_F), s(t_F) + z^I(t_F)) \rightarrow \min,
 \end{aligned}$$

где $y = x - x^I$, $s = z - z^I$, $v = u - u^I$, $g(t, y, v) = f(t, y + x^I, v + u^I) - f(t, x^I, u^I)$, α – некоторое действительное число из полуинтервала $(0, 1]$ (регулятор метода).

Будем искать функцию Кротова в виде

$$\varphi(t, y^0, y, s) = w(t) + \psi_0(t)y^0 + \psi^T(t)y + \xi^T(t)s,$$

где значения $w(t)$, $\psi_0(t)$, $\psi(t)$, $\xi(t)$ находятся из следующих приближенных соотношений (Кротова-Беллмана) для задачи (3):

$$\begin{aligned}
 (4) \quad & \varphi(t_F, y^0, y, s) \approx -G_\alpha(y, s, y^0), \\
 & \varphi(t, y^0, y, s) \approx \max_{v+u^I \in D_u} \{ \varphi(t+1, g_0(y^0, v), g(t, y, v), l(t, y, s)) \}, \\
 & t = t_F - 1, \dots, 0.
 \end{aligned}$$

При этом управление $v(t, y)$ (в форме синтеза) для задачи (3) есть управление, на котором достигается максимум в соотношениях (4).

Таким образом, управление (в форме синтеза) для задачи (2) записывается следующим образом: $u(t, x) = v(t, x - x^I(t)) + u^I(t)$, $t \in T \setminus \{t_F\}$, а искомый элемент $m'' = (x''(t), u''(t))$ получается с помощью уравнений исходной системы $x''(0) = x_0$, $u''(t) = u(t, x'')$, $x''(t) = f(t, x'', u'')$.

Алгоритм нахождения коэффициентов функции Кротова – $w(t)$, $\psi_0(t)$, $\psi(t)$, $\xi(t)$ – для задач без фазовых ограничений подробно описан в [3]. Он основан на линеаризации правых частей соотношений (4) в ряд до членов первого (второго) порядка в окрестности нуля и последующей замене производных их разностными аналогами. Шаги разностных схем выступают дополнительными регуляторами алгоритма.

Одна итерация алгоритма улучшения начального приближения $m^I = (x^I(t), u^I(t))$ состоит из следующих шагов.

- 1) Задаются значения регуляторов (параметров) метода.
- 2) Выбираются весовые коэффициенты оштрафованного функционала в виде

$$\beta_0 = 1, \quad \beta = (\beta_1^1, \dots, \beta_1^n, \beta_2^1, \dots, \beta_2^n)^T = 0,$$

- 3) если $x^I(t)$ удовлетворяет фазовым ограничениям, иначе в виде

$$\beta_0 = 1, \quad \beta^i = 0, i \in I_0, \quad \beta^i = P/(Sh^i), i \in I, \quad h^i = \begin{cases} (z^i(m))^I, & i = 0, \dots, n, \\ h^i = \delta_F^i(x^I(m)), & i = n+1, \dots, 2n, \end{cases}$$

$$I_0 = \{i \in \{1, \dots, 2n\} | h^i \leq \varepsilon\}, \quad I = \{1, \dots, 2n\} \setminus I_0, \quad P = \prod_{i \in I} h^i, \quad S = \sum_{i \in I} P/h^i, \quad \text{где } \varepsilon -$$

требуемая точность «попадания» траектории в фазовые ограничения.

- 4) Находится управление в форме синтеза в виде: $u(t, x) = A(t)x + B(t)$, $t \in T \setminus \{t_F\}$, где $A(t)$, $B(t)$ – матрицы размера $p \times n$ и $p \times 1$ соответственно при каждом фиксированном значении t .
- 5) С помощью уравнений исходной системы получается элемент $m'' = (x''(t), u''(t))$.
- 6) Если улучшение произошло, то повторяются шаги 2-5, иначе повторяются шаги 1-4 (с другими значениями параметров метода), либо итерации прекращаются (по достижении необходимой точности).

Большим преимуществом описанного алгоритма метода улучшения управления является его естественный параллелизм. А именно, алгоритм содержит крупные гранулы параллелизма (шаги алгоритма 2-5), вычисляемые независимо для каждого набора параметров метода. При этом для каждого набора параметров число итераций, сделанных до достижения указанной точности, заранее неизвестно. Следовательно, при дальнейшей программной реализации алгоритма заранее неизвестно как будут загружены отдельные узлы вычислительной установки. Поэтому динамические программы предпочтительнее любых других типов.

4. Программная реализация алгоритма на языке программирования T++

Динамическое распараллеливание имеет ряд преимуществ не только для случая, когда вопросы организации параллельного счета решаются во время ис-

полнения программы, но и для выравнивания нагрузки в гетерогенных и/или меняющихся со временем параллельных вычислительных системах, а также для задач, обладающих гранулами различной тяжести. При этом, использование неявных конструкций распараллеливания вычислений позволяет легко переносить программы между различными платформами параллельных вычислений - многопроцессорными системами, вычислительными кластерами, метакластерными системами и т.п.

T-система – система параллельного программирования, реализующая концепцию автоматического динамического распараллеливания программ. Это - оригинальная российская разработка, которая была начата в Институте программных систем РАН [4-6]. T-система автоматически (без участия программиста) выполняет распараллеливание фрагментов кода в программе, планировку вычислений, синхронизацию параллельных фрагментов кода, обмен данными между фрагментами программы и распределение данных по различным узлам кластера. Причем, эти действия определяются и выполняются в динамике, во время исполнения программы (а не планируются заранее, в статике, во время компиляции).

T-система предоставляет язык программирования T++ (очень простой параллельный диалект C++), который предназначен для эффективной реализации динамического распараллеливания. Он позволяет смешивать чистую функциональную и императивную мощь C++ на уровне T-вызовов.

Программная реализация параллельного алгоритма улучшения управления для дискретных динамических систем (T-программа) была успешно применена к исследованию имитационной модели маневров безопасной нештатной посадки вертолета с определением границы безопасной зоны. Результаты для наиболее жесткого из рассмотренных сценариев нештатной ситуации уже после трех итераций позволили сделать вывод о повышении границы опасной зоны на 15% против начального приближения при сохранении качественного характера динамики управлений и состояния.

Вычисления проводились на суперкомпьютере семейства «СКИФ», расположенном в ИПС РАН, для 256 различных наборов параметров метода улучшения для начальной программы управления, при нарушении требуемых ограничений соответствующей траекторией. В результате работы программы удалось уменьшить значение целевого функционала, удовлетворив при этом всем ограничениям. Время расчетов, при переходе на параллельный вариант, значительно сократилось.

Для оценки эффективности распараллеливания программы проведен запуск программы на различном числе узлов и замер времени работы в каждом случае. Полученные данные представлены в таблице 1. Они позволяют сделать вывод об эффективном распараллеливании указанного класса алгоритмов. А именно, хорошо виден почти идеальный (линейный) эффект распараллеливания, что и ожидалось с самого начала при формировании алгоритма.

Таблица 1. Анализ эффективности параллельной реализации метода улучшения.

Число узлов: n	1	3	5	7	9	11	13	15
Время работы программы: t_n , с	1029,85	351,99	218,83	159,60	130,71	110,29	93,69	90,10
Ускорение: t_n/t_1	1	2,93	4,71	6,45	7,88	9,34	10,99	11,43

5. Заключение

Предложенный итерационный метод улучшения начального управления может быть непосредственно применен к задачам улучшения управления для дискретных динамических систем, и для непрерывных динамических систем (после их предварительной дискретизации с достаточно малым шагом) независимо от способа представления исходной модели – явного, в аналитической форме, или неявного, в форме компьютерной программы.

Метод естественным образом ориентирован на параллельную программно-алгоритмическую реализацию. Соответствующая параллельная программа (Т-программа) была успешно применена в прикладной задаче исследования маневров вертолета в нештатной ситуации. Показана высокая эффективность распараллеливания указанного класса алгоритмов.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, проект № 08-01-00274-а.

Список литературы

1. Гурман В.И., Квоков В.Н., Ухин М.Ю. Приближенные методы оптимизации управления летательным аппаратом // *АиТ*. 2008. № 4. С. 191-201.
2. Гурман В.И. Принцип расширения в задачах управления. М. Наука: Физматлит, 1985.
3. Гурман В.И., Трушкова Е.А., Ухин М.Ю. Улучшение управления, реализующего скользящий режим // *АиТ*. 2008. № 3. С. 161-171.
4. Абрамов С.М., Есин Г.И., Загоровский И.М., Матвеев Г.А., Роганов В.А. Принципы организации отказоустойчивых параллельных вычислений для решения вычислительных задач и задач управления в Т-Системе с открытой архитектурой (OpenTS) // Международная конференция «Программные системы: теория и приложения». Переславль-Залесский, октябрь 2006. М.: Наука-Физматлит. Т. 1. С. 257-264.
5. Абрамов С.М., Загоровский И.М., Коваленко М.Р., Матвеев Г.А., Роганов В.А. Миграция от MPI к платформе OpenTS: эксперимент с приложениями PovRay и ALCMD. // Международная конференция «Программные системы: теория и приложения». Переславль-Залесский, октябрь 2006. М.: Наука-Физматлит. Т. 1. С. 265-275.
6. Абрамов С.М., Кузнецов А.А., Роганов В.А. Кроссплатформенная версия Т-системы с открытой архитектурой // Труды Международной научной конференции «Параллельные вычислительные технологии (ПаВТ '2007)». Челябинск, 29 января - 2 февраля 2007 г. Челябинск: Изд. ЮУрГУ. Т. 1. С. 115-121.