



Math-Net.Ru

Общероссийский математический портал

Д. К. Андрейченко, А. А. Ерофтиев, Д. В. Мельничук, Распараллеливание параметрического синтеза по схеме «портфель задач» на основе технологии MPI, *Изв. Саратов. ун-та. Нов. сер. Сер.: Математика. Механика. Информатика*, 2015, том 15, выпуск 2, 222–228

DOI: 10.18500/1816-9791-2015-15-2-222-228

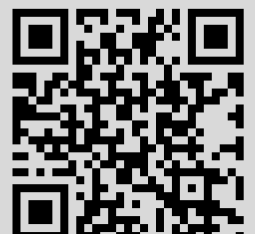
Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 18.97.14.82

5 декабря 2024 г., 03:41:45



ИНФОРМАТИКА

УДК 629.78

РАСПАРАЛЛЕЛИВАНИЕ ПАРАМЕТРИЧЕСКОГО СИНТЕЗА ПО СХЕМЕ «ПОРТФЕЛЬ ЗАДАЧ» НА ОСНОВЕ ТЕХНОЛОГИИ MPI

Д. К. Андрейченко¹, А. А. Ерофтиев², Д. В. Мельничук³

¹Доктор физико-математических наук, профессор кафедры математического обеспечения вычислительных комплексов и информационных систем, Саратовский государственный университет им. Н. Г. Чернышевского, kр_andreichenko@renet.ru

²Аспирант факультета компьютерных наук и информационных технологий, Саратовский государственный университет им. Н. Г. Чернышевского, sparxxx.at@gmail.com

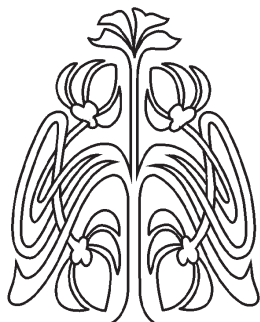
³Студент факультета компьютерных наук и информационных технологий, Саратовский государственный университет им. Н. Г. Чернышевского, meldm007@gmail.com

Предложен прототип схемы распараллеливания «портфель задач» стандартными средствами технологии MPI применительно к задачам параметрического синтеза управляемых комбинированных динамических систем. Выполнен параметрический синтез системы стабилизации подвижного объекта управления. Показана эффективность предложенного алгоритма динамической балансировки вычислительной нагрузки.

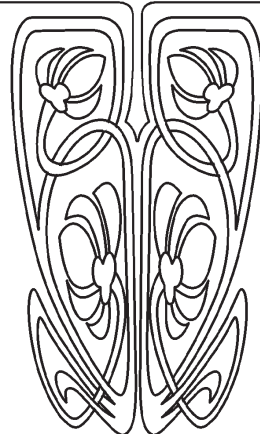
Ключевые слова: управляемые комбинированные динамические системы, MPI, параллельное программирование.

ВВЕДЕНИЕ

В математическом моделировании при реализации параллельных алгоритмов и программ часто требуется выполнять динамическую балансировку вычислительной нагрузки между процессорами. При этом обычно используется схема «портфель задач», также называемая схемой «менеджер – исполнители»: задачи поочередно запускаются на исполнение на свободных в текущий момент времени процессорах и выполняются на них параллельно. Средства поддержки наборов задач реализованы в технологии OpenMP 3 (директива `omp task`) и библиотеках Microsoft Concurrency Runtime. Однако возможности распараллеливания, предлагаемые данными программными средствами, ограничены в пределах одного процесса операционной системы, что делает невозможным их использование на кластерных системах с распределенной памятью. При этом в современном стандарте MPI [1], положенном в основу межпроцессного взаимодействия на кластерах, поддержка наборов задач не предусмотрена. Целью данной работы является создание прототипа схемы распараллеливания «портфель задач» стандартными средствами технологии MPI применительно к задачам параметрического синтеза управляемых комбинированных динамических систем.



НАУЧНЫЙ
ОТДЕЛ





1. УСТОЙЧИВОСТЬ И ПАРАМЕТРИЧЕСКИЙ СИНТЕЗ УПРАВЛЯЕМЫХ КДС

Математические модели ряда современных технических систем, представляющие собой системы обыкновенных дифференциальных уравнений и связанных с ними посредством условий связи и граничных условий уравнений в частных производных при соответствующих начальных условиях, будем далее называть комбинированными динамическими системами (КДС) [2]. КДС характеризуются входной и выходной вектор-функциями $\mathbf{x}(t) = (x_1(t), \dots, x_{N_x}(t))^T$ и $\mathbf{y}(t) = (y_1(t), \dots, y_{N_y}(t))^T$, $\mathbf{x} : \mathbb{R} \rightarrow \mathbb{R}^{N_x}$, $\mathbf{y} : \mathbb{R} \rightarrow \mathbb{R}^{N_y}$. Модельные уравнения управляемых КДС зависят от параметров обратных связей $\mathbf{p} = (p_1, \dots, p_{N_p})^T \in \mathbb{R}^{N_p}$. После линеаризации и выполнения прямого интегрального преобразования Лапласа по времени t $f(t) \rightarrow \tilde{f}(\lambda) = \int_0^\infty f(t)e^{-\lambda t} dt$ динамическая модель КДС представляется в виде матрицы передаточных функций $\Phi(\lambda, \mathbf{p})$:

$$\tilde{\mathbf{y}}(\lambda) = \Phi(\lambda, \mathbf{p})\tilde{\mathbf{x}}(\lambda), \quad \Phi(\lambda, \mathbf{p}) = [\Phi_{kj}(\lambda, \mathbf{p})] = [Q_{kj}(\lambda, \mathbf{p})/D(\lambda, \mathbf{p})],$$

$$D(\bar{\lambda}, \mathbf{p}) = \overline{D(\lambda, \mathbf{p})}, \quad Q_{kj}(\bar{\lambda}, \mathbf{p}) = \overline{Q_{kj}(\lambda, \mathbf{p})} \quad k = 1, 2, \dots, N_y, \quad j = 1, 2, \dots, N_x.$$

Здесь характеристический $D(\lambda, \mathbf{p})$ и возмущающие $Q_{kj}(\lambda, \mathbf{p})$ квазимногочлены аналитичны по λ при $\text{Re } \lambda > \sigma_0$, $\sigma_0 \in (-\infty, 0)$, если математические модели элементов КДС с распределенными по пространству параметрами учитывают малую, но конечную диссипацию энергии. Обобщенная степень [3] $n \in \mathbb{R}$ характеристического квазимногочлена $D(\lambda, \mathbf{p})$ находится из условия

$$\lim_{\lambda \rightarrow \infty} \lambda^{-n} D(\lambda, \mathbf{p}) = C_a(\mathbf{p}), \quad 0 < |C_a(\mathbf{p})| < \infty, \quad \text{Re } \lambda > -\infty,$$

а проверка принадлежности параметров обратных связей \mathbf{p} области устойчивости $\Omega_{st} \subset \mathbb{R}^{N_p}$ сводится к проверке условия

$$\mathbf{p} \in \Omega_{st} \iff \Delta_{0 \leq \omega < \infty} \arg D(i\omega, \mathbf{p}) = n\pi/2, \quad (1)$$

которое следует из теорем об устойчивости КДС [2, 3]. При использовании оптимизации шага изменения частоты ω , (1) представляет собой быстрый алгоритм проверки устойчивости КДС.

Под параметрическим синтезом понимают процедуру выбора значений параметров обратных связей в пределах области устойчивости $\mathbf{p} \in \Omega_{st}$, обеспечивающих требуемое качество переходных процессов. Параметрический синтез основан на среднеквадратическом приближении вещественной частотной характеристики проектируемой системы к желаемой вещественной частотной характеристике [3] и реализуется на основе минимизации функции $F : \mathbb{R}^{N_p} \rightarrow \mathbb{R}$:

$$F(\mathbf{p}) \rightarrow \min, \quad F(\mathbf{p}) = \begin{cases} \left(\|R_A(0, \mathbf{p}_0)\|^{-2} + \|R_A(0, \mathbf{p})\|^{-2} \right) \int_0^\infty f(\omega, \mathbf{p}) d\omega, & \mathbf{p} \in \Omega_{st}, \\ +\infty, & \mathbf{p} \notin \Omega_{st}. \end{cases} \quad (2)$$

Здесь \mathbf{p}_0 — параметры обратных связей в момент старта параметрического синтеза,

$$f(\omega, \mathbf{p}) = \|R_A(\omega, \mathbf{p}) - R_A(0, \mathbf{p})R_A^*(\omega)\|^2 + c_1 \|R_A'(\omega, \mathbf{p}) - R_A(0, \mathbf{p})R_A^{*'}(\omega)\|^2 + c_2 \|R_A''(\omega, \mathbf{p}) - R_A(0, \mathbf{p})R_A^{*''}(\omega)\|^2, \quad () = d()/d\omega,$$

$$R_A(\omega, \mathbf{p}) = [R_{A\nu j}(\omega, \mathbf{p})], \quad R_A^*(\omega) = \text{diag} \{R_{A_j}^*(\omega)\},$$

$$R_{A\nu j}(\omega, \mathbf{p}) = \begin{cases} \text{Re } \Phi_{\nu j}(i\omega, \mathbf{p}), & A_j(\mathbf{p}) \neq 0, \\ \sqrt{1 + \omega^2} \text{Re } [\Psi_{\nu j}(i\omega, \mathbf{p})], & A_j(\mathbf{p}) = 0, \end{cases}$$

$$R_{A_j}^*(\omega) = \begin{cases} (1 - (t_0\omega)^2)/(1 + (t_0\omega)^4), & A_j(\mathbf{p}) \neq 0, \\ \sqrt{1 + \omega^2}(1 - (t_0\omega)^2)/(1 + (t_0\omega)^4), & A_j(\mathbf{p}) = 0, \end{cases}$$

$$A_j(\mathbf{p}) = \left[\sum_{\nu=1}^{N_y} |\Phi_{\nu j}(0, \mathbf{p})|^2 \right]^{1/2}, \quad \Psi_{\nu j}(\lambda, \mathbf{p}) = \Phi_{\nu j}(\lambda, \mathbf{p})/\lambda, \quad \nu = 1, 2, \dots, N_y, \quad j = 1, 2, \dots, N_x.$$

Целевая функция (2) является негладкой, и для ее минимизации используется безградиентный метод Нелдера – Мида.



3. ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ПАРАМЕТРИЧЕСКОГО СИНТЕЗА И УПРАВЛЕНИЕ НАБОРАМИ ЗАДАЧ НА ОСНОВЕ ТЕХНОЛОГИИ MPI

Пусть $0 = \omega_0 < \omega_1 < \omega_2 < \dots < \omega_N < \omega_{N+1}$ — некоторые точки на числовой полуоси $\omega \geq 0$, тогда в (2)

$$\int_0^{\infty} f(\omega, \mathbf{p}) d\omega = \sum_{j=0}^N \int_{\omega_j}^{\omega_{j+1}} f(\omega, \mathbf{p}) d\omega + \int_{\omega_{N+1}}^{\infty} f(\omega, \mathbf{p}) d\omega. \quad (3)$$

Заметим, что частотная характеристика КДС характеризуется узкими высокими пиками, что вынуждает применять при вычислении отдельных слагаемых в (3) адаптивный метод численного интегрирования Гаусса. Более того, вычисление подынтегрального выражения в (3) требует решения некоторой вспомогательной линейной краевой задачи, которое для различных диапазонов изменения параметра λ реализуется на основе различных алгоритмов. По указанным причинам невозможно априорно определить характерное время вычисления отдельных слагаемых в (3).

Распараллеливание параметрического синтеза основано на том [3], что вычисление каждого из слагаемых в (3) выделяется в отдельную подзадачу, и создается набор задач [4]. Задачи из созданного набора поочередно запускаются на свободных в текущий момент времени процессорах и выполняются на них параллельно. Как следует из (3), при распараллеливании параметрического синтеза информационные связи между отдельными задачами минимальны, и, следовательно, согласно [5] теоретически можно реализовать схему динамической балансировки вычислительной нагрузки «портфель задач» или «менеджер-исполнители» стандартными средствами MPI.

Применительно к задаче параметрического синтеза управляемых КДС был предложен и реализован алгоритм динамической балансировки вычислительной нагрузки по схеме «портфель задач» («менеджер-исполнители») на основе стандартных операций MPI. Отдельный процесс MPI с рангом 0 назначается менеджером, остальные процессы — исполнителями. Процессы-исполнители получают от процесса-менеджера сообщения, в зависимости от тега которых либо обновляются данные, необходимые для вычислений, либо производятся сами вычисления, либо завершается работа процесса-исполнителя при завершении работы параллельной программы. В начале работы алгоритма динамической балансировки вычислительной нагрузки все процессы-исполнители полагаются свободными. Далее, пока не будет исчерпан набор пар значений — границ интегрирования $\omega_0, \omega_1; \omega_1, \omega_2; \dots; \omega_N, \omega_{N+1}; \omega_{N+1}, \infty$, повторяются следующие действия: 1) неблокирующая пересылка (операция MPI_Isend) границ интегрирования ω_{j-1}, ω_j из процесса-менеджера в свободные процессы-исполнители; 2) блокирующий прием (операция MPI_Recv) границ интегрирования ω_{j-1}, ω_j процессом-исполнителем; 3) неблокирующая пересылка (операция MPI_Isend) рассчитанного значения $\int_{\omega_{j-1}}^{\omega_j} f(\omega, \mathbf{p}) d\omega$ из процесса-исполнителя в процесс-менеджер; 4) неблокирующий прием рассчитанных значений в процессе-менеджере из процессов-исполнителей (операция MPI_Irecv); 5) блокирующее ожидание завершения приема (операция MPI_Waitany) рассчитанного значения из процесса-исполнителя в процесс-менеджер, после успешного выполнения которого процесс-исполнитель снова полагается свободным. После отправки последнего значения нижней границы ω_{N+1} полубесконечного интервала интегрирования в процесс-исполнитель выполняется блокирующее ожидание приема рассчитанных значений от завершающих текущую работу процессов-исполнителей.

4. МОДЕЛЬНАЯ ЗАДАЧА О СТАБИЛИЗАЦИИ ПОДВИЖНОГО ОБЪЕКТА УПРАВЛЕНИЯ

Рассмотрим задачу о минимизации ошибки системы угловой стабилизации подвижного объекта управления, а именно угла отклонения β_1 ракеты от вертикали, обусловленного воздействием входного возмущения в виде поперечной силы F_e , с учетом упругих деформаций корпуса ракеты. В предположении, что отклонения стабилизируемого объекта от вертикали достаточно малы, линеаризованные модельные уравнения движения КДС в безразмерных переменных и параметрах принимают вид

$$\begin{aligned} J_0 \ddot{\beta}_0 &= -p_1 \dot{\beta}_0 - p_2 \beta_0 + \mathbf{S}(\beta_1 + \beta_2), & m_1 \ddot{y}_1 &= (1 + m_1 + m_2) \beta_0 + P_1 - F_e, \\ J_0 \ddot{\beta}_0 + J_1 \ddot{\beta}_1 &= M_1, & m_2 [(1 + a) \ddot{\beta}_1 + \ddot{y}_1 + \ddot{y}_2] &= P_2 + a_x m_2 \beta_2, \end{aligned}$$



$$\begin{aligned}
 J_2(\ddot{\beta}_1 + \ddot{\beta}_2) &= M_2 - aP_2, \mathbf{S}(\cdot) = p_3 d(\cdot)/dt + p_4 \cdot (\cdot) + p_5 \int_0^t (\cdot) dt, \\
 \ddot{u} + u'''' + \gamma \dot{u}'''' + a_x[(m_2 + (1-x))u']' &= -\ddot{y}_1 - x\ddot{\beta}_1, \quad (\cdot)' = \partial(\cdot)/\partial x, \\
 u(0, t) = 0, \quad u'(0, t) = 0, \quad u(1, t) = y_2(t), \quad u'(1, t) &= \beta_2(t), \\
 M_1 = u''(0, t) + \gamma \dot{u}''(0, t), \quad P_1 = -u'''(0, t) - \gamma \dot{u}'''(0, t), \\
 M_2 = -u''(1, t) - \gamma \dot{u}''(1, t), \quad P_2 = u'''(0, t) + \gamma \dot{u}'''(0, t), \\
 \beta_0(0) = \beta_1(0) = \beta_2(0) = \dot{\beta}_0(0) = \dot{\beta}_1(0) = \dot{\beta}_2(0) = y_1(0) = y_2(0) = \dot{y}_1(0) = \dot{y}_2(0) &= 0, \\
 u(x, 0) = \dot{u}(x, 0) = 0.
 \end{aligned}$$

Здесь точкой сверху обозначено дифференцирование по времени t , входная вектор-функция содержит лишь одну компоненту $\mathbf{x}(t) = \{F_e(t)\}$, выходная вектор-функция суть $\mathbf{y}(t) = (\beta_1(t), \beta_2(t))^T$, а параметры обратных связей $\mathbf{p} = (p_1, p_2, p_3, p_4, p_5)^T$ требуется выбрать так, чтобы наиболее эффективно подавлять ошибку системы стабилизации. Матрица передаточных функций содержит лишь один столбец $\Phi(\lambda, \mathbf{p}) = (\Phi_2^*(\lambda, \mathbf{p}), \Phi_3^*(\lambda, \mathbf{p}))^T$. Здесь компоненты вспомогательного вектора $\Phi^* = (\Phi_1^*, \dots, \Phi_5^*)^T$ являются решением системы линейных алгебраических уравнений:

$$\sum_{j=1}^5 \psi_{kj}(\lambda) \Phi_j^* = -\delta_k^2, \quad k = \overline{1, 5},$$

где δ_k^j — символ Кронекера,

$$\begin{aligned}
 \psi_{11}(\lambda) &= J_0 \lambda^3 + p_1 \lambda^2 + p_2 \lambda, & \psi_{12}(\lambda) &= -(p_3 \lambda^2 + p_4 \lambda + p_5), & \psi_{14}(\lambda) &\equiv 0, & \psi_{15}(\lambda) &\equiv 0, \\
 \psi_{13}(\lambda) &= -(p_3 \lambda^2 + p_4 \lambda + p_5), & \psi_{21}(\lambda) &= -(1 + m_1 + m_2), & \psi_{22}(\lambda) &= -(1 + \gamma \lambda) \lambda^2 P_1^{(3)}(\lambda), \\
 \psi_{23}(\lambda) &= -(1 + \gamma \lambda) P_1^{(4)}(\lambda), & \psi_{24}(\lambda) &= [m_1 - (1 + \gamma \lambda) P_1^{(1)}(\lambda)], & \psi_{25}(\lambda) &= -(1 + \gamma \lambda) P_1^{(2)}(\lambda), \\
 \psi_{31}(\lambda) &= J_0 \lambda^2, & \psi_{32}(\lambda) &= \lambda^2 [J_1 - (1 + \gamma \lambda) M_1^{(3)}(\lambda)], & \psi_{33}(\lambda) &= -(1 + \gamma \lambda) M_1^{(4)}(\lambda), \\
 \psi_{34}(\lambda) &= (1 + \gamma \lambda) M_1^{(1)}(\lambda), & \psi_{35}(\lambda) &= -(1 + \gamma \lambda) M_1^{(2)}(\lambda), & \psi_{41}(\lambda) &= 0, \\
 \psi_{42}(\lambda) &= \lambda^2 [m_2(1 + a) - (1 + \gamma \lambda) P_2^{(3)}(\lambda)], & \psi_{43}(\lambda) &= -a_x m_2 - (1 + \gamma \lambda) P_2^{(4)}(\lambda), \\
 \psi_{44}(\lambda) &= [m_2 - (1 + \gamma \lambda) P_2^{(1)}(\lambda)], & \psi_{45}(\lambda) &= m_2 \lambda^2 - (1 + \gamma \lambda) P_2^{(2)}(\lambda), \\
 \psi_{51}(\lambda) &= 0, & \psi_{52}(\lambda) &= \lambda^2 [J_2 + (1 + \gamma \lambda)(a P_2^{(3)}(\lambda) - M_2^{(3)}(\lambda))], \\
 \psi_{53}(\lambda) &= J_2 \lambda^2 + (1 + \gamma \lambda)[a P_2^{(4)}(\lambda) - M_2^{(4)}(\lambda)], \\
 \psi_{54}(\lambda) &= (1 + \gamma \lambda)[a P_2^{(1)}(\lambda) - M_2^{(1)}(\lambda)], & \psi_{55}(\lambda) &= (1 + \gamma \lambda)[a P_2^{(2)}(\lambda) - M_2^{(2)}(\lambda)],
 \end{aligned}$$

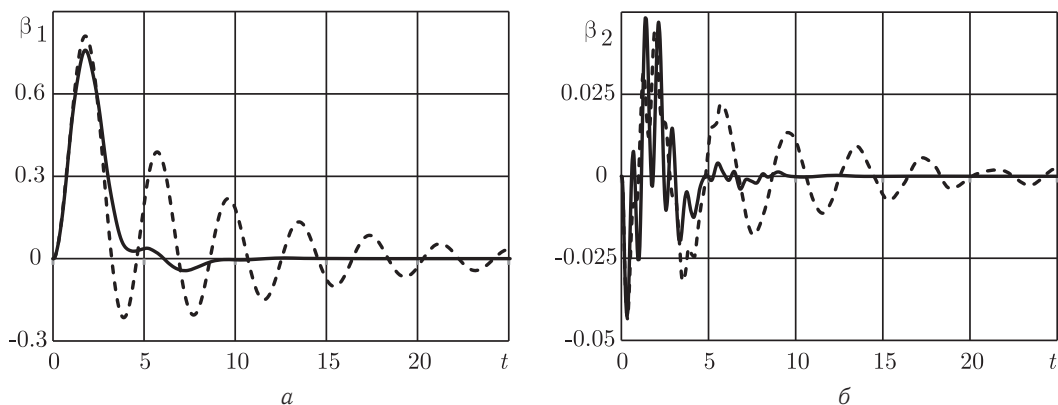
а функции $P_k^{(j)}(\lambda)$, $M_k^{(j)}(\lambda)$, $k = 1, 2$, $j = \overline{1, 4}$, находятся при помощи решения следующих вспомогательных линейных краевых задач:

$$\begin{aligned}
 (1 + \gamma \lambda) u''''(x, \lambda) + a_x[(m_2 + 1 - x)u'(x, \lambda)]' + \lambda^2 u(x, \lambda) &= -\delta_j^1 - \delta_j^3 x, \\
 u(0, \lambda) = 0, \quad u'(0, \lambda) = 0, \quad u(1, \lambda) = \delta_j^2, \quad u'(1, \lambda) &= \delta_j^4 \\
 M_1^{(j)}(\lambda) = u''(0, \lambda), \quad M_2^{(j)}(\lambda) = -u''(1, \lambda), \\
 P_1^{(j)}(\lambda) = -u'''(0, \lambda), \quad P_2^{(j)}(\lambda) = u'''(1, \lambda), \quad j = \overline{1, 4}.
 \end{aligned} \tag{4}$$

Характеристический квазимногочлен КДС суть $D(\lambda) = \det[\psi_{kj}(\lambda)]$, $k, j = \overline{1, 5}$, а его обобщенная степень $n = 9$.

Аналогично [6] при умеренных значениях λ линейная краевая задача (4) решалась численно на основе проекционного метода Галеркина с использованием в качестве базисных функций ортогональных полиномов Чебышева 1-го рода. При $|\lambda| \gg 1$ применялось асимптотическое интегрирование на основе приближения Вентцеля – Крамерса – Бриллюэна. Проверка отсутствия особенностей функций $P_k^{(j)}(\lambda)$, $M_k^{(j)}(\lambda)$, $k = 1, 2$, $j = \overline{1, 4}$, в правой комплексной полуплоскости (λ) и вблизи мнимой оси аналогично [6] выполнялась на основе принципа аргумента.

На рисунке, *а* для системы стабилизации с характерными параметрами $J_0 = 0.02$, $m_1 = 0.3$, $J_1 = 0.07$, $m_2 = 0.2$, $J_2 = 0.05$, $a = 0.167$, $a_x = 1$, $\gamma = 0.01$ представлены переходные функции – реакции на входное возмущение в форме единичного скачка Хевисайда. Переходные функции вычислялись на основе алгоритма численного обращения одностороннего интегрального преобразования Лапласа [7]. Пунктиром показаны переходные функции до выполнения параметрического синтеза, им соответствуют значения параметров обратных связей $p_1 = 3$, $p_2 = 8$, $p_3 = 6$, $p_4 = 10$, $p_5 = 3$. Сплошной линией показаны переходные функции после выполнения параметрического синтеза; им соответствуют значения параметров обратных связей $p_1 = 0.9569$, $p_2 = 2.017$, $p_3 = 2.397$, $p_4 = 1.960$, $p_5 = 0.8877$. Как следует из представленных на рисунке, *б* результатов, выполнение параметрического синтеза позволяет существенно улучшить качество переходных процессов за счет подавления ошибок системы стабилизации.



Переходные функции β_1 (*а*) и β_2 (*б*) системы угловой стабилизации

5. ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ РЕАЛИЗАЦИИ СХЕМЫ «ПОРТФЕЛЬ ЗАДАЧ» СРЕДСТВАМИ MPI

Для оценки эффективности предложенного алгоритма динамической балансировки вычислительной нагрузки проводилось сравнение характерных затрат машинного времени при выполнении параметрического синтеза в последовательной версии программы, параллельной версии на основе стандартных средств управления наборами задач технологии OpenMP и параллельной версии на основе технологии MPI. Тестирование выполнялось на двух компьютерах: с процессором Intel i5 2450M (2 физических ядра и 4 виртуальных ядра за счет использования технологии Hyper Threading) и ОС Windows 7 x64 и с процессором Intel i7 3610QM (4 физических ядра и 8 виртуальных ядер) и ОС Windows 8.1 x64. Последовательная и параллельные версии программы разработаны с использованием компиляторов Microsoft Visual C++ 2013, Microsoft Visual C# 2013 и пробной версии Intel C++ 2014 (поддерживает стандарт OpenMP 3.1). Также использовались пробные версии библиотек поддержки высокопроизводительных вычислений Intel Math Kernel Library 11.1 и IMSL 7.0. В качестве объектно ориентированной оболочки для функций BLAS и LAPACK из состава IMKL использовалась свободно распространяемая библиотека LAPACK++. В качестве реализации стандарта MPI 2.2 использовались входящие в состав свободно распространяемого пакета Microsoft HPC Pack 2012 SP1 библиотеки Microsoft MPI и соответствующие клиентские утилиты. Сравнение характерных затрат машинного времени (в секундах) при выполнении параметрического синтеза ранее рассмотренной системы угловой стабилизации в последовательной и параллельных версиях программы приведено в табл. 1 и 2.

Таблица 1

Время параметрического синтеза, с. Intel i5 2450M

Номер теста	1	2	3	4	5
Последовательная	487	480	502	496	483
OpenMP	231	243	232	231	231
MPI, 3 процесса	269	273	261	272	264
MPI, 5 процессов	241	242	242	241	242



Таблица 2

Время параметрического синтеза, с. Intel i7 3610QM

Номер теста	1	2	3	4	5
Последовательная	497	497	498	501	502
OpenMP	138	140	138	139	138
MPI, 5 процессов	127	125	124	127	124
MPI, 9 процессов	117	116	117	116	116

Как следует из представленных результатов, предложенный алгоритм динамической балансировки вычислительной нагрузки стандартными средствами MPI наиболее эффективен, если число процессов-исполнителей совпадает с числом виртуальных ядер процессора. На процессоре Intel i5 2450M (см. табл. 1) эффективность предложенного алгоритма сопоставима с эффективностью стандартных средств технологии OpenMP для создания наборов задач и управления ими. На процессоре Intel i7 3610QM (см. табл. 2) эффективность предложенного алгоритма превосходит эффективность стандартных средств OpenMP.

Библиографический список

1. MPI: A Message-Parsing Interface Standard 3.0. September 21, 2012. URL: <http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf> (дата обращения: 18.12.2014).
2. Андрейченко Д. К., Андрейченко К. П. К теории комбинированных динамических систем // Изв. РАН. Теория и системы управления. 2000. № 3. С. 54–69.
3. Андрейченко Д. К., Андрейченко К. П. Моделирование, анализ и синтез комбинированных динамических систем: учеб. пособие. Саратов: ООО Изд. дом «Райт-Экспо», 2013. 144 с.
4. Эндрюс Г. Р. Основы многопоточного, параллельного и распределенного программирования. М.: Изд. дом «Вильямс», 2003. 512 с.
5. Гергель В. П. Теория и практика параллельных вычислений. М.: Интернет-ун-т информ. технологий: БИНОМ. Лаб. знаний, 2007. 423 с.
6. Андрейченко Д. К., Андрейченко К. П., Кононов В. В. К устойчивости системы угловой стабилизации вращающегося упругого стержня под действием продольного ускорения // Изв. РАН. Теория и системы управления. 2013. № 5. С. 12–25. DOI 10.7868/S0002338813030025.
7. Андрейченко Д. К. Эффективный алгоритм численного обращения интегрального преобразования Лапласа // Журн. вычисл. матем. и матем. физ. 2000. Т. 40, № 7. С. 1030–1044.

Parallelization of Parametric Synthesis by «Problems Portfolio» Scheme Based on MPI Technology

D. K. Andreichenko, A. A. Eroftiev, D. V. Melnichuk

Saratov State University, 83, Astrakhanskaya st., 410012, Saratov, Russia, kp_andreichenko@renet.ru, sparxxx.at@gmail.com, meldm007@gmail.com

We propose a prototype of a parallelization scheme «problems portfolio» by standard means of MPI technology with application to problems of parametric synthesis of controlled hybrid dynamical systems. Parametric synthesis of the stabilization system of a controlled mobile object was performed. Efficiency of the proposed algorithm of dynamical balancing of computational load is shown.

Key words: controlled hybrid dynamical systems, MPI, parallel programming.

References

1. MPI: A Message-Parsing Interface Standard 3.0. September 21, 2012. Available at: <http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf> (Accessed 18, December, 2014).
2. Andreichenko D. K., Andreichenko K. P. On the theory of hybrid dynamical systems. *Journal of Computer and Systems Sciences International*, 2000, vol. 39, no 3, pp. 383–398.
3. Andreichenko D. K., Andreichenko K. P. *Modelirovanie, analiz i sintez kombinirovannykh dinamicheskikh sistem. Uchebnoe posobie* [Modeling, analysis and synthesis of combined dynamical systems. Tutorial]. Saratov, Izdatel'skii Dom «Rait-Ekspo», 2013. 144 p. (in Russian).
4. Andrews G. R. *Foundations of Multithreaded, Parallel and Distributed Programming*. Addison Wesley, 2000, 664 pp.
5. Gergel V. P. *Teoriya i praktika paralelnykh*



vychisleniy [Theory and practice of parallel computing] Moscow, Internet-Un-t Inform. Tekhnologii, BINOM, Lab. znanii, 2007. 423 с.

6. Ahdreichenko D. K., Andreichenko K. P., Kononov V. V. On the Stability of the Angular Stabilization System of the Rotating Rod under Longitudinal Acceleration. *Journal of Computer*

and System Sciences International, 2013, vol. 52, no 5, pp. 686–699. DOI: 10.1134/S1064230713030027.

7. Andreichenko D. K. An Efficient Algorithm for Numerical Inversion of the Laplace Transform. *Comput. Math. Math. Phys.*, 2000, vol. 40, no 7, pp. 1030–1044.

УДК 519.7

ВОССТАНОВЛЕНИЕ ГРАФА С ПОМЕЧЕННЫМИ ВЕРШИНАМИ ПЕРЕМЕЩАЮЩИМСЯ ПО НЕМУ МОБИЛЬНЫМ АГЕНТОМ

С. В. Сапунов

Кандидат физико-математических наук, научный сотрудник, Институт прикладной математики и механики НАН Украины, Донецк, sapunov_sv@yahoo.com

Рассматривается задача построения автономным мобильным агентом топологической модели своей операционной среды. Модель среды представляет собой связный неориентированный граф с помеченными вершинами. В работе предложен полиномиальный алгоритм восстановления и разметки графа среды для коллектива из агента-исполнителя и агента-вычислителя.

Ключевые слова: графы с помеченными вершинами, мобильный агент, восстановление графа.

ВВЕДЕНИЕ

Помеченные графы широко применяются в информатике для описания и моделирования разнообразных вычислительных процессов. В этом контексте наиболее изучены конечные оргграфы с помеченными дугами (LTS [1], взвешенные автоматы [2], конечные автоматы). Тем не менее существует множество вычислительных процессов, естественным образом представляемых графами с помеченными вершинами (в программировании, робототехнике [3], верификации моделей [4]). В данной работе исследуется модель информационной системы как системы взаимодействующих объектов: агента и его операционной среды [5]. Операционная среда представляется в виде топологической модели, т. е. графа с помеченными вершинами.

Одной из центральных задач анализа операционной среды является построение ее карты, т. е. восстановление графа среды [3]. Причем построенная карта должна быть пригодной для дальнейшей навигации агентов. В данной работе под навигацией понимается перемещение агента (робота, поисковой программы и т. п.) из текущей области его операционной среды в заданную ее область. Будем считать карту пригодной для навигации, если на основании описания пути по карте агент, получающий минимум локальной информации о среде, может пройти этот путь в среде.

1. ПОСТАНОВКА ЗАДАЧИ

Мобильный агент установлен в произвольную вершину неизвестного ему графа, все вершины которого не помечены или, что то же самое, помечены одной и той же меткой. Требуется разработать алгоритм перемещений агента по графу и разметки пройденных вершин, который позволяет восстановить граф. Причем полученная разметка должна способствовать навигации агента по графу.

К настоящему времени известно довольно много методов восстановления графов [6]. Однако эти методы, как правило, предполагают разметку всех вершин графа одной и той же меткой, что неприемлемо для последующей навигации. Тривиальным методом решения поставленной задачи является восстановление графа с разметкой каждой его вершины уникальной меткой. В работе предложен алгоритм, в котором на основе анализа структуры графа уменьшается число различных меток.