

## И н ф о р м а т и к а   и   с и с т е м ы   у п р а в л е н и я

УДК 519.71

И.В. Дахно, Т.А. Галаган

### **ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В РАБОТЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ПОДБОРА ПАРАМЕТРОВ РОБАСТНОЙ СИСТЕМЫ УПРАВЛЕНИЯ СУЩЕСТВЕННО НЕСТАЦИОНАРНЫМ ОБЪЕКТОМ**

*В статье рассмотрена возможность использования технологии параллельных вычислений для работы генетического алгоритма, который позволяет подбирать конкретные значения коэффициентов алгоритма робастного регулятора, полученных на этапе синтеза робастной системы в виде некоторых числовых интервалов. Реализация данной технологии производилась в прикладном пакете MATLAB на трех рабочих станциях компьютерной сети.*

*Ключевые слова: параллельные вычисления, робастный регулятор, система управления существенно нестационарным объектом, генетический алгоритм.*

### **APPLICATION OF THE PARALLEL CALCULATIONS IN THE GENETIC ALGORITHM FOR SELECTION OF ROBUST CONTROL SYSTEM FOR ESSENTIALLY NON-STATIONARY OBJECT PARAMETERS**

*The article discusses the possibility of using parallel computing technology for the genetic algorithm, which allows you to select the specific value of coefficients in the algorithm of robust controller obtained in stages of the synthesis in the form of numeric intervals. Implementation of this technology made in the application package MATLAB on three workstations.*

*Key words: parallel computing, robust control, control system of essentially non-stationary object, genetic algorithm.*

#### **Введение**

Широкое применение параллельных вычислений в различных областях современных научных исследований обусловлено главным образом двумя факторами. Первый из них вызван появлением новых областей знаний и усложнением математических моделей, требующих ресурсоемких расчетов, которые в ряде случаев могут быть выполнены только на базе высокопроизводительной техники, с помощью методов параллельных или распределенных вычислений. Второй заключается в повсеместном распространении параллельных компьютерных систем.

Синтез робастных систем управления предполагает поиск алгоритма управления, сохраняющего выходные характеристики системы и сигнал ошибки в заданных допустимых пределах, несмотря на наличие в ней некоторых неопределенностей. Робастные алгоритмы управления существенно нестационарными объектами, полученные на основе критерия гиперустойчивости в работе [1], как правило, содержат коэффициенты, значения которых синтезированы в виде диапазонов. Конкретные же значения подбираются в результате достаточно длительных вычислительных экспериментов. Это обусловило использование здесь генетического алгоритма и для увеличения скорости его работы параллельных вычислений.

### Постановка задачи

Робастная система управления синтезирована в работе [1] в виде:

$$\begin{cases} \frac{dx(t)}{dt} = A(t, \xi)x(t) + B(t, \xi)u(t) + f_{\xi}(t), \\ y(t) = L^T x(t), z = g^T y(t), \end{cases} \quad (1)$$

$$x(t) \in R^n, y(t) \in R^m, z(t) \in R, g \in R^m; u \in R.$$

$$A(t) = A(t, \xi), \quad B(t) = B(t, \xi) \quad (2)$$

$$A_* x_* + B_* r_* = 0, \quad y_* = L^T x_*, \quad r_* = g^T y_* = const. \quad (3)$$

$$A(t) = A_* + B_* \beta^T(t) L^T, \quad B(t) = B_*(1 + \alpha(t)). \quad \alpha(t) > 0, \quad (4)$$

$$u(t) = (\gamma_1 |r_*| + \gamma_2 y^T(t) y(t)) \text{sign}(r_* - g^T y(t)), \quad (5)$$

$$\gamma_1 > 0, \quad \gamma_2 = \gamma_1 \sup_t \|\beta(t)\|^2 \quad (6)$$

$$\lim_{t \rightarrow \infty} (x_* - x(t)) = \lim_{t \rightarrow \infty} \varepsilon(t) = 0. \quad (7)$$

Здесь (1) – объект, функционирующий в условиях априорной неопределенности (2); (3) – не-явный эталон; (4) – условия структурного согласования объекта и эталона; (5) – робастный регулятор с коэффициентами, удовлетворяющими условиям (6); (7) – цель управления при действии на объект внешних ограниченных возмущений.

Генетический алгоритм осуществляется для подбора набора значений коэффициентов  $\gamma_1, \gamma_2$  в алгоритме вида (5) в виде числовых интервалов (6).

Одним из показателей эффективности работы генетического алгоритма является его скорость, которая оценивается временем, затраченным на его выполнение до достижения критерия останова. Данный показатель стало возможно улучшить путем распараллеливания его выполнения на нескольких рабочих станциях, которыми могут служить как ядра процессора, так и отдельные вычислительные машины.

### Применения генетического алгоритма

Схема работы генетического алгоритма представлена на рис. 1.

Методика применения генетического алгоритма для подбора значений коэффициентов робастного регулятора включает следующие шаги.

1. *Создание начальной популяции.* Популяция – набор из 20 хромосом, представляющих собой некоторые значения коэффициентов робастного регулятора. Каждая хромосома состоит из двух существенных чисел. Формирование начальной популяции может быть осуществлено либо на основе ранее экспериментально полученных значений, либо задаваться произвольным образом (во втором случае поиск решения существенно замедлится).



Рис. 1. Схема работы генетического алгоритма.

2. *Оценивание приспособленности.* В качестве функции приспособленности каждой хромосомы выступает числовой параметр – ошибка управления, поступающий на его вход. Значения этого параметра являются результатом работы Simulink-модели системы управления (1) – (7). Ранжирование хромосом осуществляется в соответствии со значениями их функций приспособленности: особь, имеющая наименьшее значение функции приспособленности, получает наивысший ранг, равный единице.

3. *Отбор.* Первоначально из популяции изымаются и переходят в следующее поколение потомки, которые имеют наилучшие значения функции приспособленности. Они не будут участвовать в скрещивании. Количество таких особей можно задать произвольно, по умолчанию оно равно трем. Дальнейший отбор производится на основе метода рулетки. В результате особи, имеющие наибольший ранг, могут попасть в родительский пул по несколько раз; особи, имеющие низкий ранг, также попадут в нее, но только один раз.

4. *Применение генетических операторов.* Скрещивание реализуется путем попарного обмена хромосомами между случайно сформированными парами родителей. А особи, имеющие наихудшее значение функции приспособленности, подвергаются мутации. Она заключается в суммировании каждой хромосомы родителя со случайно сгенерированным числом из распределения Гаусса. Таким образом, хромосома потомка будет равна сумме хромосомы родителя и произведения некоторого множителя на случайное число из диапазона от 1 до родительского значения. Множителем в данном выражении определяется степень мутации.

5. *Условие завершения алгоритма.* Окончание работы происходит при выполнении одного из двух условий: а) достигнуто конечное число итераций; б) значение ошибки, полученное на предыдущем шаге, совпадает со значением ошибки управления на текущем шаге.

### Организация параллельных вычислений

Реализация распараллеливания работы генетических алгоритмов возможна на основе несколько подходов. Первый подход основан на создании одной из существующих моделей параллельных генетических алгоритмов. Такими моделями могут быть [2]:

модель «master-slave», особенность которой заключается в синхронной работе популяций, обладающих общим адресным пространством, причем мастер (управляющий процесс) осуществляет развитие популяции, а подчиненные занимаются расчетом целевой функции;

модель «fine-grained», имеющая управляющий процесс и сильно зависящих от него подчиненных, работающих в асинхронном режиме при использовании общих ресурсов;

модель «coarse-grained», в которой каждый процесс моделирует свою популяцию, используя собственное адресное пространство, а взаимодействие популяций определяется топологией ее сети.

Второй подход предполагает применение библиотек моделей параллельного программирования – таких как MPI, OpenMP, PVM, Pthreads, обеспечивающих распараллеливание вычислений между некоторыми кластерами или рабочими станциями.

В данной работе распараллеливание генетического алгоритма осуществляется применением встроенной в пакет MATLAB модели OpenMP. За счет идеи инкрементального распараллеливания OpenMP идеально подходит для разработчиков, желающих быстро распараллелить свои вычислительные программы с большими параллельными циклами, что и требуется в данном исследовании.

Для выполнения вычислительного эксперимента использовалась схема подключения аппаратного обеспечения, приведенная на рис. 2.

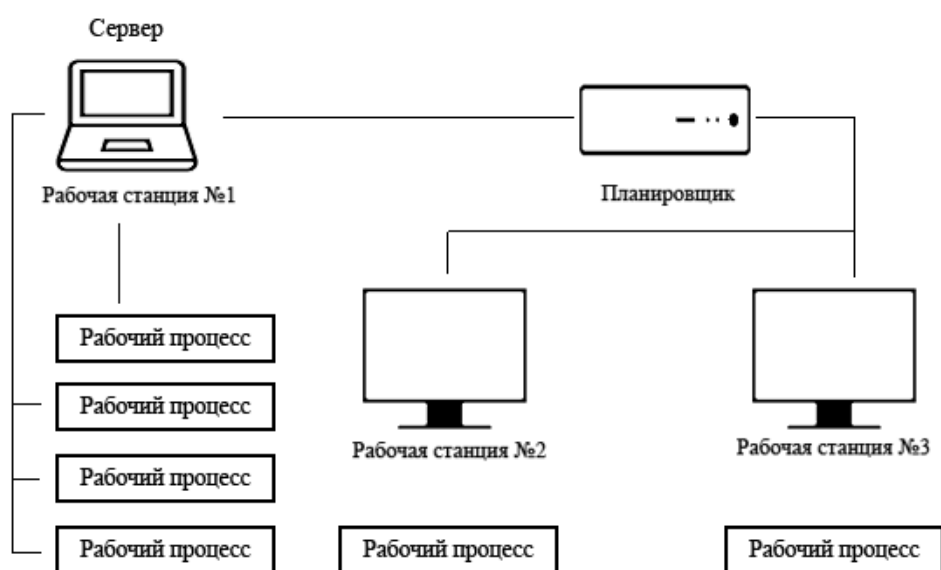


Рис.2. Схема используемой конфигурации оборудования.

В качестве клиента может выступать любой компьютер, подключенный к сети и имеющий доступ к серверу через планировщик, что является обязательным. В качестве планировщика в нашем случае выступает отдельное устройство. На узле планировщика рабочий вычислительный процесс не запускается. Он просто выполняет функции координатора работы остальных рабочих станций, на которых можно запустить по одному процессу. Запуск нескольких процессов на одном узле имеет смысл только в случаях, когда он является многопроцессорным или многоядерным. В данном эксперименте использовалась только одна многоядерная станция, выполняющая роль основного сервера.

Технические характеристики узлов использованной системы:

рабочая станция № 1: ноутбук, процессор Intel Core i5 2.5GHz (3.0GHz on boost), L2 cache 6MB, 4 ядра, 8 логических процессоров, 8GB RAM, SSD 120GB, Win 7 x64;

рабочая станция № 2: стационарный персональный компьютер, процессор Intel Pentium 4 670 3.8GHz, L2 cache 2MB, 1 ядро, 2 логических процессора, 4GB RAM, 500HDD, WIN XP x32;

рабочая станция № 3: стационарный персональный компьютер, AMD Athlon64 3200+ 2.2GHz, L2 cache 512KB, 1 ядро, 2 логических процессора, 4GB RAM, 250HDD, WIN XP x32;

планировщик: спутниковый ресивер DREAMBOX 7025+ на основе ОС Linux. ATI Xilleon 226 300MHz, 128MB RAM, 120HDD, Enigma2 (Linux).

Несмотря на то, что в качестве планировщика выбрано нестандартное решение, со своей задачей он справился достаточно хорошо, при этом не занимая ресурсы сервера и рабочих станций, на которых он мог бы располагаться. Каждый рабочий процесс, получив от него задачу на исполнение, выполнял ее, затем возвращал результат и получал новую задачу. При завершении рабочими процессами всех задач менеджер приступал к управлению следующей группой задач с доступными рабочими процессами.

В качестве планировщика использовался The MathWorks. Каждая машина с запущенным рабочим процессом имела MDCE-службу, которая восстанавливает рабочий процесс и менеджер задач в случае неисправности машины [3].

Типичная сессия параллельных вычислений для клиента включала следующие шаги: получение ссылки на менеджер задач при помощи функций MATLAB; формирование задачи, в которую будут записаны некоторые подзадачи; формирование подзадачи – сегмента основной задачи; постановка задачи в очередь на исполнение в менеджере задач; получение результатов работы; удаление задачи и освобождение ресурсов памяти.

### Анализ полученных результатов

Моделирование замкнутой системы управления проводилось со следующими значениями параметрами объекта и эталона:

$$A(t) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a1(t) & a2(t) & a3(t) \end{pmatrix}, \quad b_s = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad L^T = \begin{pmatrix} 0 & 1 & 0.5 \\ 0.25 & 1 & 0 \end{pmatrix}, \quad f = \begin{pmatrix} 0 \\ 0 \\ f_n \end{pmatrix}, \quad g^T(2 \ 3), \quad (8)$$

$$a1(t) = -5 + \sin 6t, \quad a2(t) = -1 + 4 \sin 6t + 2 \cos 4t, \quad a3(t) = -4 + 2 \cos 4t. \quad (9)$$

На рис. 3 приведены результаты моделирования в однопоточном режиме. Слева – с коэффициентами регулятора, подобранными в работе [1], справа – результат работы генетического алгоритма, запущенного в одном потоке. Значения числовых коэффициентов в алгоритме управления в результате работы генетического алгоритма соответственно равны 418.752 и 16.44, а подобранные вручную – 100 и 10.

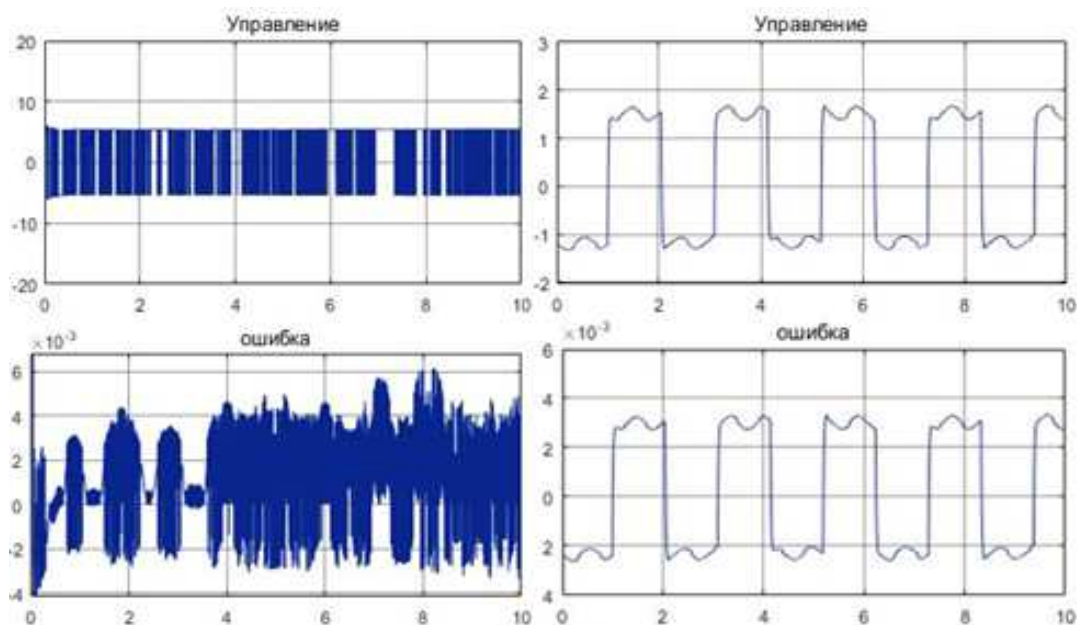


Рис. 3. Результаты моделирования робастной системы управления (1) – (9) в однопоточном режиме.

В результате значение ошибки рассогласования объекта и эталона уменьшилось всего в полтора раза, но алгоритм управления принял вид, реализуемый на практике.

Затраченные ресурсы выполненной задачи в однопоточном режиме: размер одной популяции – 20 хромосом, общее количество популяций – 860, количество итераций – 44, время выполнения – 108 мин.

На рис. 4 приведены результаты моделирования с использованием технологии параллельного вычисления. В данной сессии используем все те же параметры, размеры и количество популяций, что и при однопоточном вычислении. Значения коэффициентов, подобранные в этом режиме работы генетического алгоритма, равны соответственно 411.42 и 19.24.

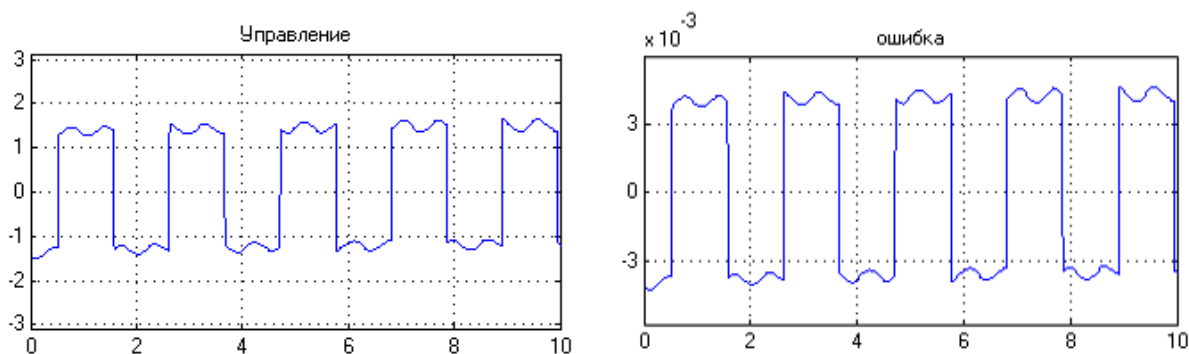


Рис. 4. Результаты моделирования робастной системы управления (1) – (9) с применением параллельных вычислений.

Затраченные ресурсы выполненной задачи в режиме параллельных вычислений: размер одной популяции – 20 хромосом, общее количество популяций – 860, количество итераций – 44, время выполнения – 19.5 мин.

Результаты эксперимента показывают, что применение параллельных вычислений не вносит корректировок в результат, но сокращает время выполнение данной задачи более чем в пять раз.

На рис. 5 представлен график работы трех рабочих станций в течение 7 мин.

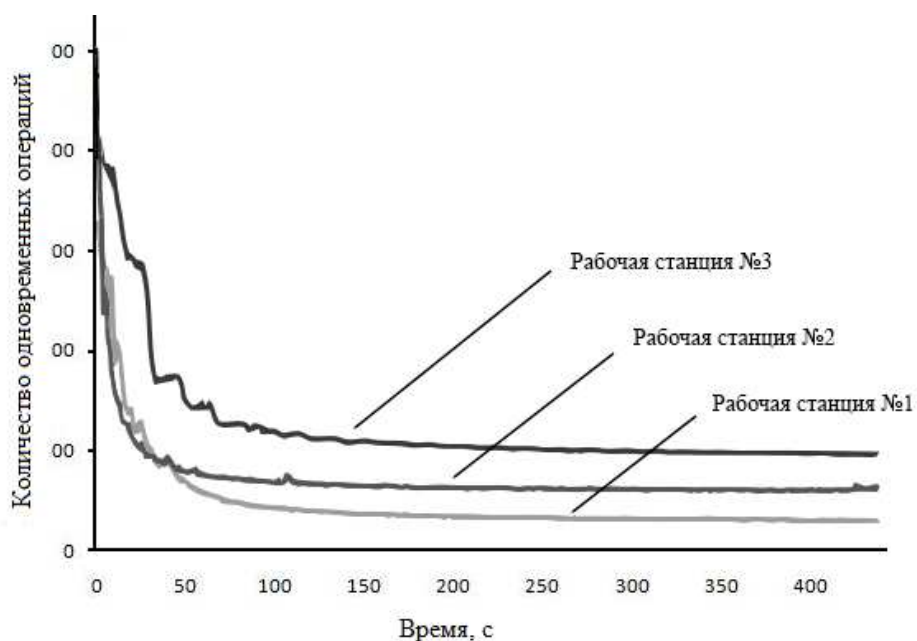


Рис. 5. График производительности параллельной вычислительной системы при выполнении генетического алгоритма.

Таким образом, внедрение технологии параллельных вычислений позволило сократить время работы генетического алгоритма для подбора коэффициентов робастной системы управления. Подо-

бранные значения коэффициентов позволяют улучшить качество управления динамическим существенно нестационарным объектом.

В дальнейшем предполагается использовать описанную в работе методику для робастных систем с явным и явно-неявным эталоном, а также в системах с запаздыванием по состоянию.

1. Еремин, Е.Л., Галаган, Т.А., Семичевская, Н.П. Нелинейное робастное управление нестационарными объектами. – Благовещенск: Амурский гос. ун-т, 2006. – 185 с.
2. Кныш, Д.С., Курейчик, В.М. Параллельный генетический алгоритм с нечетким оператором миграции // Искусственный интеллект. – Донецк, 2010. – № 3. – С. 73-80.
3. Оленев, Н.Н. Параллельное программирование в Matlab и его приложения. – М.: ВЦ РАН, 2007. – 120 с.
4. Филатова, Е.С., Филатов, Д.М. Применение параллельных вычислений в задачах многокритериальной оптимизации и их реализация в среде MATLAB // Наука и современность. – Новосибирск, 2015. – 282 с.
5. Мунасыпов, Р.А., Ахмеров, К.А., Ахмеров, К.А. Методика оптимизации нечеткого регулятора с помощью генетических алгоритмов // Фундаментальные исследования. – 2015. – № 2-15. – С. 3275-3280.

УДК 004.42

С.Г. Самохвалова, С.В. Худяков

## ИНФОРМАЦИОННАЯ СИСТЕМА МОНИТОРИНГА ТЕМПЕРАТУРЫ И ВЛАЖНОСТИ В ПОМЕЩЕНИИ

*В работе приведены данные о существующих системах, достоинства и недостатки данных систем, технические характеристики выбранной аппаратной части, а также продемонстрирована работа разработанной системы.*

*Ключевые слова: микроконтроллер, Arduino, датчик температуры и влажности, скетч.*

## INFORMATION SYSTEM FOR INDOOR TEMPERATURE AND HUMIDITY MONITORING

*The article deals with data on existing systems, their advantages and disadvantages, technical characteristics of the selected hardware. Here we demonstrated operation of the developed system.*

*Key words: microcontroller, Arduino, temperature and humidity sensor, sketch.*

### Введение

Контроль и мониторинг внешних воздействий – таких как температура, влажность, загрязненность воздуха, электромагнитное излучение и т.д. – являются на сегодняшний день важной частью технологического, а также эксплуатационного процессов.

При постоянно увеличивающемся спросе и соответственно растущей как пищевой, так и промышленной продукции необходимо свести к минимуму либо вовсе оградить ее от внешних воздействий. Сегодня для решения данного вопроса проектируются и реализуются полностью автоматические или автоматизированные системы для мониторинга и контроля внешних воздействий.