

Выполним схему предложенного аналогового процессора в соответствии с (4) в программе схемотехнического моделирования OrCAD. Для моделирования интеграторов и дифференциаторов дробного порядка были использованы модели фрактальных элементов. Полученные результаты представлены на рис. 3.

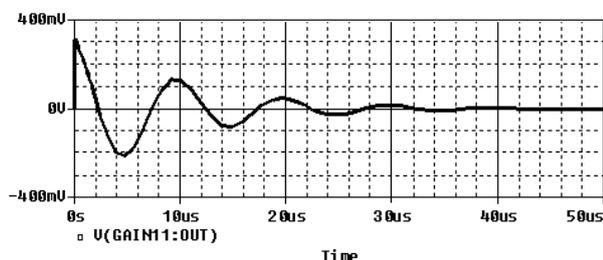


Рис. 3. Результаты работы аналогового процессора, смоделированного в программе OrCad в соответствии с (4)

Сравнивая результаты численного решения и схемотехнического моделирования, видим, что они полностью совпадают. Это говорит о том, что предложенный алгоритм действительно позволяет создавать аналоговые процессоры, способные решать дифференциальные уравнения дробного порядка.

Так как алгоритм достаточно формализован, такие процессоры можно реализовать с помощью ПАИС (ФРАА), в которых реализованы интеграторы и дифференциаторы дробного порядка на основе

операционных усилителей и фрактальных элементов, выполненных по интегральной технологии.

Библиографические ссылки

1. Лебедеко Е. Аналоговые вычислители: жизнь до и во время эпохи Цифры // КомпьютерраOnline. – URL: <http://www.computerra.ru/vision/668463/> (дата обращения: 21.05.2013).
2. Генератор электрических колебаний на основе элементов с фрактальным импедансом / К. О. Максимов, А. А. Потапов, П. А. Ушаков и др. // Радиолокация и радиосвязь : V Всерос. науч.-техн. конф., 21–25 нояб. 2011 г., Москва : докл. – М. : Изд-во ИРЭ РАН, 2011. – С. 54–58.
3. Fractional Order Systems: Modeling and Control Applications / R. Caponetto, G. Dongola, L. Fortuna, I. Petráš // World Scientific Publishing Co. Pte. Ltd., 2010. 195 p. – URL: http://www.worldscientific.com/doi/pdf/10.1142/9789814304207_fmatter (дата обращения: 21.05.2013).
4. Левин Л. Методы решения технических задач с использованием аналоговых вычислительных машин. – М. : Мир, 1966. – 415 с.
5. Xue D., Chen Y. Solving Applied Mathematical Problems with MATLAB. – CRC Press, 2010. – 433 p. – URL: http://mniyazicankaya.files.wordpress.com/2013/02/ebooksclub-org_solving_applied_mathematical_problems_with_matlab-1.pdf (дата обращения: 21.05.2013).
6. Ушаков П. А. Методы анализа и синтеза многослойных неоднородных RC-элементов с распределенными параметрами и устройств на их основе : автореф. дис. ... д-ра техн. наук: 05.13.05. – Ижевск : Изд-во ИжГТУ, 2009. – 35 с. – URL: <http://vak.ed.gov.ru/common/img/uploaded/files/vak/announcements/techn/2009/09-02/UshakovPA.doc> (дата обращения: 21.05.2013).

P. A. Ushakov, DSc in Engineering, Professor, Kalashnikov Izhevsk State Technical University
A. V. Shadrin, Master's degree student, Kalashnikov Izhevsk State Technical University

Circuit simulation of analogue processor for solution of fractional order differential equations

At present there are prerequisites to create hybrid computers in which the universal analogue processor can be used, allowing to solve differential equations of any fractional order in real-time mode. The computing circuit of such a processor implemented as a circuitry model in program OrCAD is offered. The result of solving the fractional order differential equation by means of the designed model completely coincided with the result of solving this equation by Grunwald - Letnikov method.

Keywords: hybrid computer, fractional order differential equations, analogue processor, integrator of the fractional order, differentiator of the fractional order

Получено: 25.04.13

А. С. Шаура, кандидат технических наук, ассистент;
Ижевский государственный технический университет
имени М. Т. Калашникова
Д. С. Кочурова, инженер-программист
ОАО «Ижевский радиозавод»

РЕШЕНИЕ ЗАДАЧИ ТРАЕКТОРНОГО УПРАВЛЕНИЯ МОБИЛЬНЫМИ РОБОТАМИ С ПОМОЩЬЮ ПАРАЛЛЕЛЬНОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА

С использованием средств .NET Framework 4.0 реализован параллельный генетический алгоритм, с помощью которого решена задача оптимального управления движением мобильных устройств в вязкой жидкости по заданной траектории. В качестве мобильных устройств рассмотрены системы, обладающие возможностью самопродвижения в вязкой жидкости за счет возвратно-поступательных колебаний внутренней массы. Проведено исследование эффективности разработанного алгоритма в зависимости от размерности задачи и количества ядер процессора.

Ключевые слова: оптимальное управление движением, самопродвижение, генетический алгоритм, параллельные вычисления

Введение

Задачи управления автономными мобильными устройствами, в частности обеспечение движения по определенным траекториям, осуществление перемещения между заданными точками, являются актуальными и особенно востребованы при поиске новых способов передвижения и разработки самоходных роботов. Сложность управления такими системами заключается в отсутствии однозначных явных зависимостей между управляющими параметрами и траекториями движения.

Поиск оптимального управления движением сопряжен со значительными вычислительными затратами, но, несмотря на широкое распространение многопроцессорных систем, предоставляемые ими возможности достаточно редко используются при разработке алгоритмов и программ для решения прикладных задач. Во многом это связано с тем, что организация параллельных вычислений посредством большинства современных технологий (OpenMP, MPI, а также CUDA и др.) сопряжена со значительными сложностями алгоритмического характера. Программный код, выполняемый в параллельных потоках, намного сложнее соответствующей реализации последовательно выполняемых инструкций для решения аналогичной задачи, хуже поддается анализу и отладке, вследствие чего в большей степени может стать источником ошибок или проявлений некорректного поведения программы.

Вышедшая в 2010 г. новая версия .NET Framework 4.0 известной платформы от Microsoft включила в себя библиотеку параллельных задач Task Parallel Library (TPL), что во многом упростило программирование для многопроцессорных и распределенных систем. TPL представляет собой набор открытых типов и API-интерфейсов в пространствах имен System.Threading и System.Threading.Tasks. Она динамически масштабирует степень параллелизма для наиболее эффективного использования всех доступных процессоров. Начиная с .NET Framework 4.0 библиотека параллельных задач является предпочтительным способом создания многопоточного и параллельного кода для работы в среде Windows.

Параллельный генетический алгоритм

Параллельное развитие множества особей в рамках одной популяции или одновременное развитие нескольких параллельных популяций в генетических алгоритмах обуславливает «естественность» и удобство их распараллеливания, что является характерной особенностью и достоинством эволюционных методов.

Существует два принципиальных подхода к построению параллельных генетических алгоритмов (ПГА) [1]. Первый подход включает так называемые «островные» генетические алгоритмы, которые предполагают одновременное развитие нескольких относительно независимых популяций, между ко-

торыми через заданное число поколений происходит обмен лучшими особями.

Для второго подхода характерна схема «мастер – рабочий». При таком подходе развивается единственная популяция, и цикл развития этой популяции осуществляется на процессоре, который называется «мастер». На «рабочих» процессорах осуществляются вычисления, связанные с оценкой особей. Работа по оценке особей может осуществляться параллельно, и эффективность параллельных вычислений в данном случае тем больше, чем сложнее процедура оценки особей, что особенно актуально при моделировании, когда нет явных выраженных формулами зависимостей для вычисления приспособленности.

В представленной работе была реализована схема «мастер – рабочий» следующим способом. На «рабочие» процессоры, помимо вычислений оценки особей, передавалась и вычислительная нагрузка, связанная с формированием следующего поколения: выбор родительской пары, скрещивание, мутация. Таким образом, распараллеливанию подвергся весь цикл генерации новой популяции:

```

ПолучитьНовоеПоколение(...)
{
    НоваяПопуляция[0] ← СохранитьЭлиту();

    for (int i=1; i<РазмерПопуляции; i++)
    {
        ВыбратьРодителей(P1, P2, СтараяПопуляция);
        НоваяПопуляция[i] ← Скрестить(P1, P2);
        Мутировать(НоваяПопуляция[i]);
        ОценитьОсобь(НоваяПопуляция[i]);
    }
}

```

Распараллеливание представленного цикла осуществлено с помощью методов библиотеки Task Parallel Library из .NET Framework 4.0.

Генетический алгоритм реализован с сохранением элиты – лучшая особь без изменений вносится в новую популяцию, остальные особи популяции получают в результате работы операторов скрещивания и мутации.

Использовались следующие параметры генетического алгоритма: размер популяции – 100; вероятность скрещивания – 0,8; вероятность мутации – 0,2; отбор – турнирный; оператор скрещивания выбирался случайным образом из Bin1, Bin2, BLX- α и Fit [2]; мутация реализует случайное изменение координат особи.

Для оценки эффективности параллельного генетического алгоритма при решении задач управления рассмотрим задачу управления траекторным движением автономного мобильного объекта, которая заключается в нахождении таких значений управляющего параметра, которые обеспечивали бы перемещение управляемого объекта по заданной траектории.

Тестирование ПГА на решении задачи траекторного управления

Для обеспечения движения системы по заданной траектории $Y(t)$ необходимо решать задачу оптимального управления типа Лагранжа. Движение системы описывается обыкновенными дифференциальными уравнениями:

$$\frac{dx}{dt} = f(t, x, u), \quad (1)$$

где $u = u(t)$ – управляющий параметр. В качестве критерия оптимальности примем минимальное значение среднеквадратического отклонения траектории системы $r(t)$ от заданной $Y(t)$:

$$J(u) = \int_0^T |r(t) - Y(t)| dt \rightarrow \min. \quad (2)$$

Рассмотрим одномерный случай, когда движение системы определяется уравнением

$$\frac{dx}{dt} = -\frac{x}{4} + \frac{u(t)}{t+1}.$$

В качестве заданной выберем траекторию, представленную на рис. 1, на временном отрезке $t \in [0, 10]$.

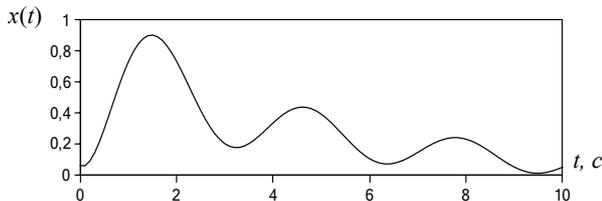


Рис. 1. Заданная траектория движения

При решении задачи с помощью генетического алгоритма будем считать, что управление кусочно-постоянно. Для этого разобьем отрезок $t \in [0, 10]$ на N частей и положим, что на каждом i -м временном участке управление постоянно и равно u_i . Тогда размерность вектора управления U , особи генетического алгоритма, равна N .

Траектория движения системы и соответствующее управление после первой итерации при $N = 50$ показаны на рис. 2 и рис. 3 соответственно.

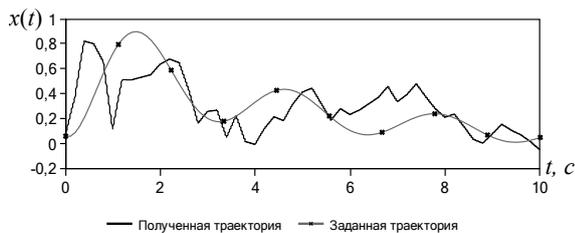


Рис. 2. Траектория движения после первой итерации

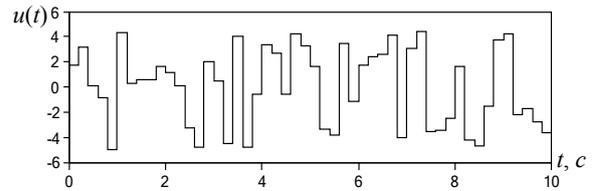


Рис. 3. Управление движением после первой итерации

Изменение целевой функции в процессе решения оптимизационной задачи при $N = 50$ показано на рис. 4, а вид траектории движения объекта при оптимальном управлении – на рис. 5.

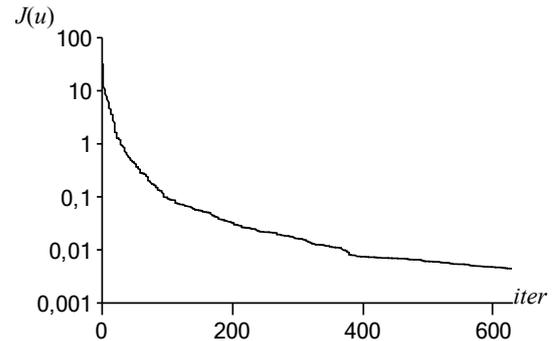


Рис. 4. Изменение целевой функции в процессе итераций

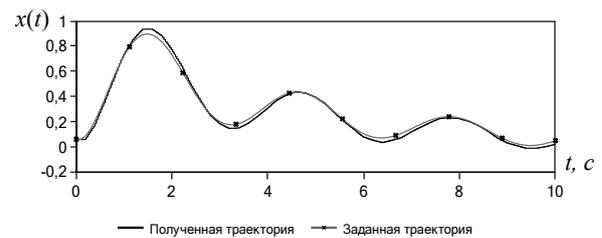


Рис. 5. Траектория движения управляемого объекта

Изменение значения оптимальной управляющей функции $u(t)$ во времени показано на рис. 6.

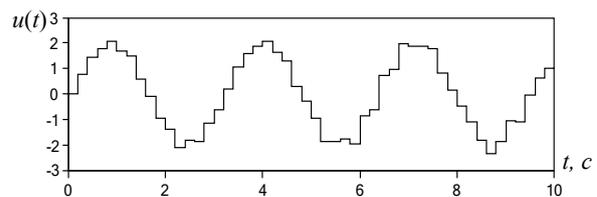


Рис. 6. Управление движением объекта

Результаты решения задачи были получены для различных значений $N \in \{10, 30, 50, 100, 200, 500, 1000\}$. В табл. 1 представлено время выполнения 600 итераций T_K и значение функционала $J(u)$, усредненные по 10 запускам генетического алгоритма в зависимости от размерности вектора управления U и количества используемых ядер процессора K .

Основными характеристиками при оценке параллельных алгоритмов являются ускорение параллельного алгоритма S_K и эффективность параллельного алгоритма E_K .

Таблица 1. Зависимость времени вычисления и невязки от размерности задачи и количества используемых ядер

N	Количество ядер процессора							
	K=1		K=4		K=8		K=32	
	T ₁ , с	J(u)	T ₄ , с	J(u)	T ₈ , с	J(u)	T ₃₂ , с	J(u)
10	1,97	0,0004	1,12	0,0004	0,86	0,0005	0,93	0,0003
30	5,02	0,0033	2,59	0,0037	1,71	0,0034	1,41	0,0033
50	8,06	0,0057	3,57	0,0049	2,59	0,0047	1,90	0,0045
100	15,50	0,0053	6,83	0,0066	4,78	0,0067	3,23	0,0059
200	30,51	0,0078	14,16	0,0080	8,94	0,0088	5,74	0,0072
500	77,26	0,0110	35,62	0,0108	22,82	0,0111	14,95	0,0116
1000	160,74	0,0173	76,87	0,0162	51,92	0,0155	39,02	0,0192

Ускорение алгоритма S_K показывает, во сколько раз применение параллельности уменьшает время решения задачи по сравнению с последовательным алгоритмом, и определяется величиной:

$$S_K = \frac{T_1}{T_K}, \quad (3)$$

где T_1 – время выполнения алгоритма на одном процессоре; T_K – время выполнения того же алгоритма на K процессорах.

Эффективность параллельного алгоритма определяется величиной:

$$E_K = \frac{S_K}{K}, \quad (4)$$

где S_K – ускорение параллельного алгоритма; K – количество процессоров в системе. Из того, что $S_K \leq K$, следует ограничение сверху на величину эффективности: $E_K \leq 1$.

Для реализованного параллельного генетического алгоритма зависимость значений ускорения S_K и эффективности E_K от размерности задачи N и количества процессоров в системе K приведены в табл. 2.

Таблица 2. Зависимость ускорения и эффективности параллельного алгоритма от размерности задачи и количества используемых ядер

N	Количество ядер процессора					
	K=4		K=8		K=32	
	S_K	E_K	S_K	E_K	S_K	E_K
10	1,76	0,44	2,28	0,28	2,11	0,07
30	1,94	0,48	2,92	0,37	3,54	0,11
50	2,26	0,56	3,11	0,39	4,24	0,13
100	2,27	0,57	3,24	0,40	4,79	0,15
200	2,15	0,54	3,41	0,43	5,31	0,17
500	2,17	0,54	3,38	0,42	5,17	0,16
1000	2,09	0,52	3,10	0,39	4,12	0,13

Графики представленных в табл. 1 и 2 зависимостей S_K и E_K от количества ядер системы для $N = 10$, $N = 100$ и $N = 1000$ показаны на рис. 7.

Исходные данные для зависимостей, представленных в табл. 2 и на рис. 7, получены на системах, имеющих в составе процессора 4, 8 и 32 ядра, из которых 2, 4 и 16 ядер соответственно являются физическими. Полученные зависимости ускорения и эффективности параллельного алгоритма от количества ядер процессора хорошо согласуются с известными законами.

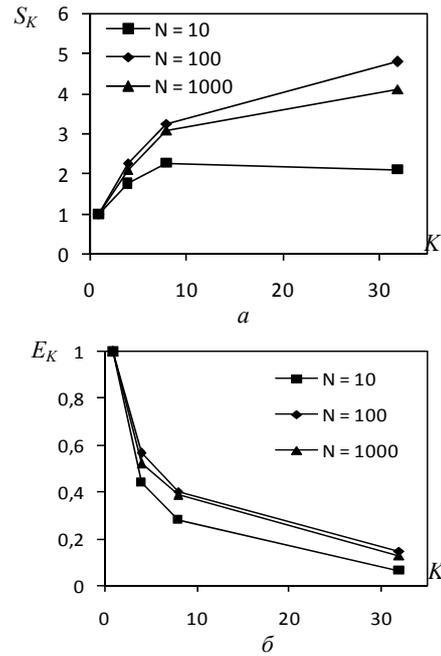


Рис. 7. Зависимость характеристик параллельного генетического алгоритма от количества ядер процессора: а – ускорение S_K ; б – эффективность E_K

Задача управления мобильным роботом

Рассматривается плоская постановка задачи управления движением в вязкой жидкости автономного объекта (мобильного робота), обладающего свойствами самопродвижения. Управляемый объект имеет форму кругового цилиндра массой M . Движение начинается из состояния покоя. Самопродвижение объекта является следствием возвратно-поступательного перемещения внутренней материальной точки массой m по заданному закону $\rho(t) = (\xi(t), \eta(t))^T$.

Движение системы рассматривается относительно двух систем координат (рис. 8) – неподвижной (x, y) и связанной с телом подвижной (ξ, η) .

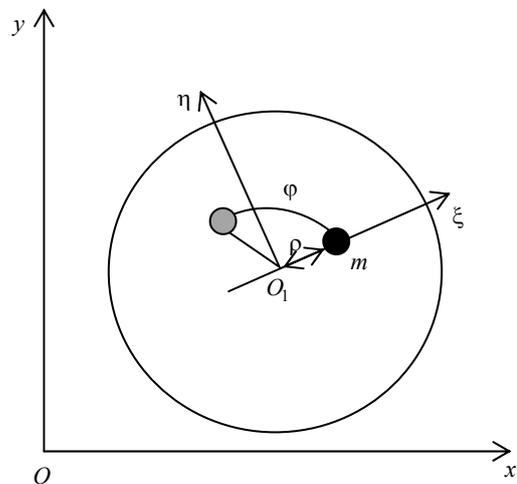


Рис. 8. Взаимное расположение систем координат и управляемого объекта

Для плоской постановки уравнения движения управляемого объекта в жидкости представлены в [3]:

$$\begin{aligned} \frac{dP_1}{dt} - \omega P_2 &= F_1 \\ \frac{dP_2}{dt} + \omega P_1 &= F_2 \\ \frac{dK}{dt} + u_b P_2 - v_b P_1 &= G \\ \frac{dx}{dt} &= u_b \cos(\varphi_0) - v_b \sin(\varphi_0) \\ \frac{dy}{dt} &= u_b \sin(\varphi_0) + v_b \cos(\varphi_0) \\ \frac{d\varphi_0}{dt} &= \omega \end{aligned} \quad (5)$$

$$P_1(0) = P_2(0) = K(0) = x(0) = y(0) = 0,$$

где $P_1 = (A_1 + m)u_b + m(\dot{\xi}(t) - \omega\eta(t))$ и $P_2 = (A_2 + m)v_b + m(\dot{\eta}(t) + \omega\xi(t))$ – проекции вектора импульса в подвижной системе координат; $K = B\omega - m(u_b\eta - v_b\xi) - m(\dot{\xi}(t) - \omega\eta)\eta + m(\dot{\eta}(t) + \omega\xi)\xi$ – кинетический момент; $\mathbf{F} = (F_1, F_2)$ – вязкая сила, действующая со стороны жидкости на тело; G – момент вязкой силы; $A_1 = M + \lambda_1$, $A_2 = M + \lambda_2$ и $B = J + \lambda_3$, причем M и J – масса тела и момент инерции без учета материальной точки, а λ_1 , λ_2 , λ_3 – коэффициенты присоединенных масс; ω – угловая скорость вращения; $\mathbf{U}_b = (u_b, v_b)$ – вектор скорости движения тела; φ_0 – угол поворота подвижной системы координат относительно неподвижной.

Управлять движением объекта можно путем поворота оси, вдоль которой внутренняя материальная точка совершает возвратно-поступательные колебания, на некоторый угол φ относительно оси ξ подвижной системы координат. За управляющую функцию примем величину угла направления перемещения внутренней точки, равного сумме углов φ и φ_0 : $\theta = \varphi + \varphi_0$.

Поскольку решение задачи оптимального управления движением мобильного робота требует многократного вычисления значения вязкой силы $\mathbf{F} = (F_1, F_2)$, получаемой из решения системы уравнений Навье – Стокса, используем аппроксимацию вязкой силы зависимостью от основных параметров движения в явном виде [4]:

$$\mathbf{F} = q_1 \mathbf{A}^{k-1} + q_2 \frac{C_D |\mathbf{U}_b| \mathbf{U}_b}{2} \quad (6)$$

$$C_D = \frac{q_3}{\text{Re}} (1 + q_4 \text{Re}^\gamma), \quad \mathbf{A}^{k-1} = \frac{\mathbf{U}_b^{k-1} - \mathbf{U}_b^{k-2}}{\Delta t}$$

$$q_1 = -3,2, \quad q_2 = 0,57, \quad q_3 = 7, \quad q_4 = 0,6, \quad \gamma = 0,87.$$

В качестве закона перемещения внутренней материальной точки выберем следующую зависимость:

$$\xi(t) = \begin{cases} \rho \sin^2(\omega_0 \tau), & \tau < \frac{\pi}{2\omega_0} \\ \rho \cos^2\left(\tau - \frac{\pi}{2\omega_0}\right), & \tau \geq \frac{\pi}{2\omega_0} \end{cases}, \quad \tau = t - \left\lfloor \frac{t}{T_0} \right\rfloor, \quad (7)$$

$$T_0 = \frac{\pi}{2} \left(1 + \frac{1}{\omega_0}\right),$$

$$\eta(t) = 0,$$

где операция $\left\lfloor \frac{t}{T_0} \right\rfloor$ соответствует взятию целой части, $\omega_0 = 4$.

Будем решать задачу оптимального управления движением из $A(x_A, y_A)$ в точку $B(x_B, y_B)$ при различной начальной ориентации мобильного робота по траектории, имеющей минимальное отклонение от прямолинейной. Траектория движения робота характеризуется точками (x_c, y_c) , являющимися координатами тела в конце каждого периода T_0 . Управляющей функцией является угол θ поворота вектора $\rho(t)$, поэтому функционал (2) в задаче оптимального управления принимает вид:

$$J(\theta) = \int_0^T [y_c(t) - x_c(t)]^2 dt \rightarrow \min \quad (8)$$

при условии минимального отклонения конечной точки траектории $(x_c(T), y_c(T))$ от заданной точки $B(x_B, y_B)$:

$$|(x_c(T), y_c(T)) - B| \rightarrow \min. \quad (9)$$

Для решения задачи (8) – (9) с помощью параллельного генетического алгоритма весь временной отрезок $[0, T]$ был разбит на N частей, предполагая, что на каждом i -м временном участке управление постоянно и равно θ_i . Таким образом, каждая особь генетического алгоритма представляла собой вектор значений управляющей функции $\mathbf{U} = \{\theta_i\}$, $i = \overline{1, N}$. Для вычисления значений целевой функции решалась система дифференциальных уравнений (5) методом Рунге – Кутты четвертого порядка. Из первых трех уравнений системы определялись значения P_1 , P_2 и K , на основании которых вычислялись компоненты вектора скорости тела u и v , а также угловая скорость вращения ω . Из трех последних уравнений системы определялись точки (x_c, y_c) траектории движения тела.

Результаты решения задачи управления роботом

Заданная траектория движения объекта определяется начальной и конечной точками $A(0, 0)$ и $B(1, 1)$

соответственно. Решение ищется на временном отрезке $t \in [0, 100]$. Управление считается постоянным на отрезке T_0 . Размерность вектора управления U равна 50 при $\omega_0 = 4$. Начальная ориентация робота определяется углом $\varphi_0 = 1$. Изменение функционала (8) в процессе итерационного поиска оптимального управления показано на рис. 9.

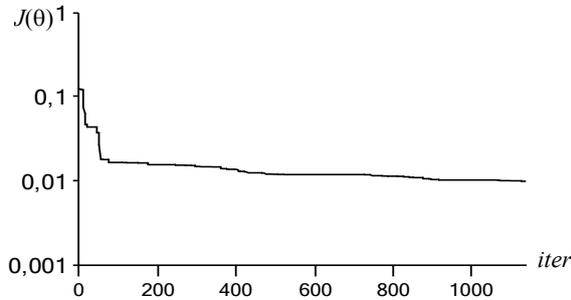


Рис. 9. Изменение целевой функции в процессе итераций поиска оптимального управления движением тела

Траектория движения объекта из начальной точки в конечную, полученная в результате решения задачи оптимального управления, показана на рис. 10.

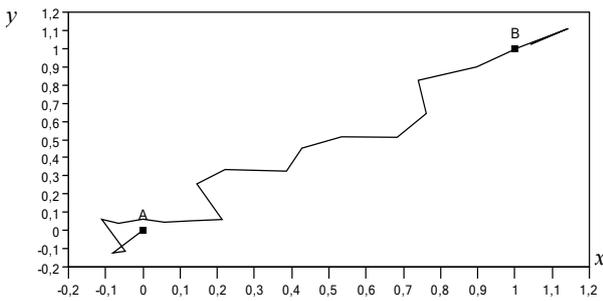


Рис. 10. Траектория движения системы из точки A в точку B

Изменение значения оптимальной управляющей функции во времени, соответствующее представленной траектории, приведено на рис. 11.

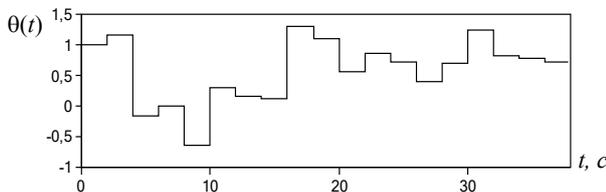


Рис. 11. Управление движением системы

Процесс изменения координат робота во времени при движении по оптимальной траектории представлен на рис. 12 и 13.

Анализируя представленную траекторию (рис. 10) и соответствующее ей управление (рис. 11), можно сделать вывод, что при выбранном в работе законе перемещения внутренней массы (7) движение тела происходит преимущественно в направлении, противоположном оси колебаний внутренней точки. Однако углом поворота оси колебаний направление движения тела однозначно не определяется (рис. 12 и

13), что и является главной сложностью при управлении движением рассматриваемых систем.

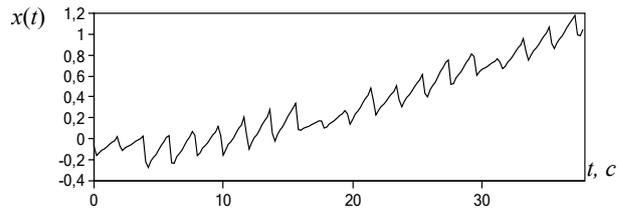


Рис. 12. Изменение координаты робота $x(t)$ при движении по оптимальной траектории

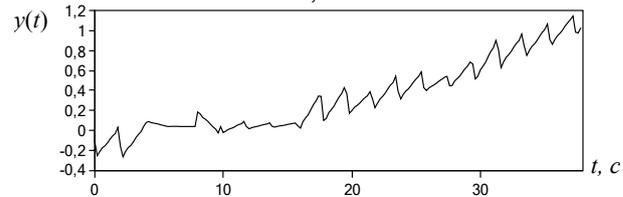


Рис. 13. Изменение координаты робота $y(t)$ при движении по оптимальной траектории

Траектории движения из точки $A(0, 0)$ в точку $B(1, 1)$ при различных углах φ_0 начальной ориентации робота приведены на рис. 14 и 15.

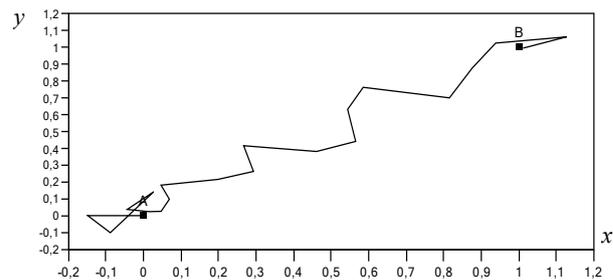


Рис. 14. Траектория движения при $\varphi_0 = 0$

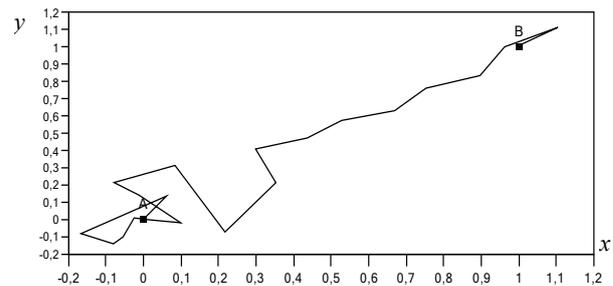


Рис. 15. Траектория движения при $\varphi_0 = -2$

Из рис. 10, 14 и 15 хорошо заметно, что наибольшие сложности возникают на начальном участке траектории при выходе из точки A, когда управляемый объект пытается выйти на прямую, соединяющую начальную и конечную точки движения. После этого движение системы осуществляется с минимальными отклонениями от прямолинейного.

Время выполнения 600 итераций и показатели эффективности S_K и E_K параллельного генетического алгоритма, рассчитанные по формулам (3) и (4), представлены в табл. 3 и 4.

Таблица 3. Зависимость времени вычисления и невязки от размерности задачи и количества используемых ядер

N	Количество ядер процессора							
	K=1		K=4		K=8		K=32	
	T_1, c	$J(u)$	T_4, c	$J(u)$	T_8, c	$J(u)$	T_{32}, c	$J(u)$
10	38	1,19	14	1,18	11	1,18	9	1,18
30	92	0,30	37	0,16	22	0,26	19	0,32
50	157	0,01	61	0,70	36	0,57	26	0,53
100	293	0,70	115	0,58	68	0,54	46	0,59
200	566	0,61	224	0,71	129	0,43	86	0,65
500	1 367	1,19	546	0,59	308	0,54	206	0,67

Таблица 4. Зависимость ускорения и эффективности параллельного алгоритма от размерности задачи и количества используемых ядер

N	Количество ядер процессора					
	K=4		K=8		K=32	
	S_K	E_K	S_K	E_K	S_K	E_K
10	2,60	0,65	3,42	0,43	4,18	0,13
30	2,48	0,62	4,22	0,53	4,85	0,15
50	2,57	0,64	4,41	0,55	6,04	0,19
100	2,54	0,64	4,31	0,54	6,37	0,20
200	2,53	0,63	4,37	0,55	6,58	0,21
500	2,51	0,63	4,44	0,55	6,64	0,21

Графики представленных в табл. 3 и 4 зависимостей S_K и E_K от количества ядер процессора для $N = 10$, $N = 100$ и $N = 500$ показаны на рис. 16.

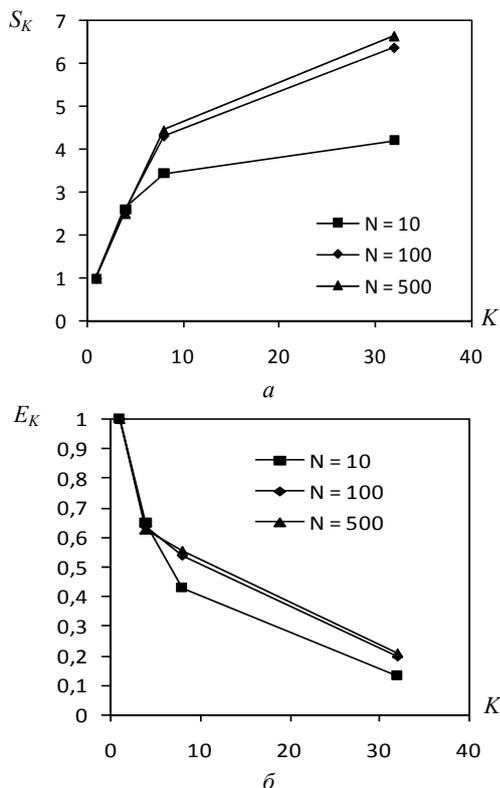


Рис. 16. Зависимость характеристик параллельного генетического алгоритма от количества ядер процессора: a – ускорение S_K ; b – эффективность E_K

Применение генетического алгоритма в этом случае дает большее ускорение и является более эффективным, чем в случае, представленном в табл. 2 и на рис. 7. Задача (5)–(9) имеет большую сложность, а вычисление значения целевой функции (8) является более трудоемким и требует большего числа операций, вследствие чего меньшее влияние оказывает время передачи данных, а распределение нагрузки между ядрами процессора – большой выигрыш.

Заключение

Анализ результатов проведенных численных исследований показывает, что реализация генетических алгоритмов с использованием параллельных вычислений при проведении скрещивания и оценки особей позволяет уменьшить время решения задачи. При этом целесообразность распараллеливания тем выше, чем более сложной является процедура вычисления целевой функции. Полученные зависимости оценок эффективности параллельного генетического алгоритма согласуются с общими законами, известными для параллельных и распределенных вычислений.

Решенная с помощью параллельного генетического алгоритма задача оптимального управления движением в вязкой жидкости автономных мобильных объектов подтверждает возможность неограниченного перемещения таких систем путем самопродвижения за счет возвратно-поступательного перемещения внутренней массы.

Проведенные исследования позволяют говорить об эффективности применения генетических алгоритмов для решения сложных задач оптимизации, в том числе и когда явная зависимость целевой функции от значений управляющих параметров неизвестна.

Библиографические ссылки

1. Иванов Д. Е. Масштабируемый параллельный ГА построения идентифицирующих последовательностей для современных многоядерных ВС // Управляющие системы и машины. – 2011. – № 1. – С. 25–32.
2. Тенев В. А. Применение генетических алгоритмов с вещественным кроссовером для минимизации функций большой размерности // Интеллектуал. системы в пр-ве. – 2006. – № 1. – С. 93–107.
3. Рамоданов С. М., Тенев В. А. Движение тела с переменной геометрией масс в безграничной вязкой жидкости // Нелинейная динамика. – 2011. – Т. 7, № 3. – С. 635–648.
4. Ветчанин Е. В., Тенев В. А. Управляемое движение тела в жидкости при возвратно-поступательном перемещении внутренней материальной точки // Интеллектуал. системы в пр-ве. – 2011. – № 2. – С. 62–72.

* * *

A.S. Shaura, PhD in Engineering, Assistant, Kalashnikov Izhevsk State Technical University
D.S. Kochurova, Software engineer, Izhevsk Radio Plant JSC

Solution of the problem of mobile robot trajectory control by means of parallel genetic algorithm

In the paper we represent the solution of optimal control of mobile robot moving along the assigned trajectory in viscous fluid by means of the parallel genetic algorithm. It is implemented by .NET Framework 4.0. Systems are considered as mobile devices with the possibility of self-propulsion in a viscous fluid due to back-and-forth oscillations of the internal mass. The effectiveness of the developed algorithm depending on the problem dimension and the number of processor cores is investigated.

Keywords: optimal motion control, self-propulsion, genetic algorithm, parallel computations

Получено: 23.05.13