

УДК 519.6

Е. М. Воронов, А. П. Карпенко,
О. Г. Козлова, В. А. Федин

ЧИСЛЕННЫЕ МЕТОДЫ ПОСТРОЕНИЯ ОБЛАСТИ ДОСТИЖИМОСТИ ДИНАМИЧЕСКОЙ СИСТЕМЫ

Рассмотрена задача приближенного построения области достижимости динамической системы. Исследованы подходы к решению этой задачи на основе последовательных и параллельных методов мультифиниша, аппроксимации векторного поля соответствующей системы обыкновенных дифференциальных уравнений, а также на основе нейронных сетей. В качестве примера рассмотрено применение указанных методов для приближенного построения границы области достижимости летательного аппарата.

E-mail: apkarpenko@mail.ru

Ключевые слова: динамическая система, область достижимости, аппроксимация векторного поля, нейронная сеть, летальный аппарат.

Во многих приложениях, использующих модели исследуемых объектов в виде обыкновенных дифференциальных уравнений (ОДУ), возникает задача построения области достижимости соответствующей динамической системы. Например, такая задача возникает при решении проблемы траекторной безопасности летательного аппарата [1], при решении близкой задачи о посадке вертолета на подвижный носитель [2]. Важной особенностью задачи построения области достижимости является то, что ее часто приходится решать в режиме реального масштаба времени.

Аналитически построить области достижимости удается лишь в простейших случаях, не представляющих практического интереса, поэтому при рассмотрении реальных ситуаций приходится использовать численные методы, т.е. строить некоторую дискретную аппроксимацию области достижимости или ее границ.

Численное построение области достижимости можно выполнить путем многократного интегрирования модельной системы ОДУ при различных управлениях (метод мультифиниша). В практических задачах типичной является ситуация, когда основные вычислительные затраты при построении области достижимости методом мультифиниша обусловлены затратами на интегрирование указанной системы ОДУ. Поэтому ускорить вычисления можно за счет параллельного интегрирования модельной системы ОДУ.

К параллельным методам интегрирования ОДУ относятся блочные методы, а также методы на основе геометрической схемы декомпозиции системы ОДУ.

Методы первого класса основаны на использовании одношаговых и многошаговых блочных формул интегрирования ОДУ, например формулы Адамса–Башфорта [3]. В вычислительной практике обычно используют блочные формулы не выше четырехточечных. Поэтому потенциальный параллелизм этой группы методов невелик.

Идея методов второго класса состоит в разбиении модельной системы ОДУ на подсистемы, число которых равно числу используемых процессоров вычислительной системы, и интегрировании каждой подсистемы на своем процессоре. В силу, как правило, относительно невысокой размерности вектора фазовых координат исследуемой динамической системы данные методы также принципиально не могут обеспечить значительное ускорение.

В настоящей работе полагается, что используемая многопроцессорная вычислительная система (МВС) представляет собой *однородную* MIMD-систему с общей или распределенной памятью [4]. Поскольку метод мультифиниша использует интегрирование совокупности систем модельных ОДУ, естественным в данном случае является распараллеливание путем разделения этой совокупности на наборы, число которых равно числу процессоров в используемой МВС. В работе рассматривается проблема балансировки загрузки МВС при распараллеливании в методе мультифиниша по указанной схеме, получены оценки эффективности параллельных вычислений.

Даже при вычислениях на однопроцессорной ЭВМ значительное ускорение при построении области достижимости можно обеспечить, аппроксимируя векторное поле модельной системы ОДУ [2]. При этом область допустимых значений вектор-функции, соответствующей правым частям системы ОДУ, покрывается некоторой сеткой. Предварительно во всех узлах этой сетки вычисляются значения указанной вектор-функции и сохраняются в памяти ЭВМ. При интегрировании модельной системы ОДУ используются эти значения или их подходящая аппроксимация. В работе получены оценки объема требуемой памяти, а также оценки ускорения данного метода.

Очевидной является идея использовать метод мультифиниша с заменой модельной системы ОДУ ее некоторой нейросетевой аппроксимацией. Однако данный метод не дает кардинального сокращения времени построения области достижимости. Поэтому в настоящей работе используется другой подход к построению области достижимости с использованием нейронных сетей — подход, основанный на построении нейронной сети, выходами которой являются непосредственно

координаты точек, принадлежащих области достижимости или ее границе.

В качестве примера рассмотрена задача построения области достижимости для летательного аппарата, описываемого системой ОДУ шестого порядка. Рассмотрено решение задачи с помощью метода мультифиниша, метода аппроксимации векторного поля этой системы ОДУ, а также нейросетевого метода.

Постановка задачи. Рассмотрим динамическую систему

$$\dot{X} = F(t, X, U), \quad X(0) = X^0, \quad (1)$$

где $X = X(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$ — n -мерный вектор фазовых переменных системы; $U = U(t) = (u_1(t), u_2(t), \dots, u_m(t))^T$ — m -мерный вектор управлений; $F(t, X, U) = (f_1(t, X, U), f_2(t, X, U), \dots, f_n(t, X, U))^T$ — n -мерная вектор-функция; $X^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$ — n -мерный вектор начальных условий, $t \in [0, T]$. На вектор фазовых переменных X и вектор управления U наложены ограничения

$$X \in D_X, \quad U \in D_U \subset L_U[0, T], \quad (2)$$

где $L_U[0, T]$ — некоторое пространство m -мерных функций, определенных на интервале $[0, T]$, например, пространство функций, интегрируемых с квадратом на этом интервале.

Среди фазовых переменных x_1, x_2, \dots, x_n выделим $\nu \leq n$ переменных; не ограничивая общности, положим, что эти переменные образуют ν -мерный вектор $Y(t) = (x_1(t), x_2(t), \dots, x_\nu(t))^T$.

Областью достижимости $D_Y = D_Y(T, X^0)$ системы (1) назовем множество всех возможных значений вектора $Y(T)$, которые достигаются на решениях системы (1) при начальных условиях X^0 и выполнении ограничений (2) на фазовые переменные и управления.

Ставим следующую *прямую задачу*: при заданных векторе начальных условий X^0 и конечном времени T построить область достижимости D_Y системы (1). Также рассмотрим *обратную задачу*: при заданных векторе начальных условий X^0 , конечном времени T и точке $Y(T)$, принадлежащей области достижимости D_Y , найти управление $U \in D_U$, переводящее систему (1) в эту точку.

Для решения прямой задачи, очевидно, достаточно построить границу Γ_Y области достижимости D_Y . В некоторых случаях удастся найти множество допустимых управлений $D_U^{\Gamma} \subseteq D_U$, принадлежащих классу управлений $L_U^{\Gamma}[0, T] \subseteq L_U[0, T]$, которые приводят систему (1) на эту границу [1]. В этом случае прямая задача сводится к построению границы Γ_Y .

Сделаем следующие допущения. Во всех случаях при интегрировании системы ОДУ (1) используется алгоритм с постоянным шагом интегрирования $\Delta t = T/K$, где K — число шагов интегрирования.

Каждый шаг интегрирования требует l вычислений значений функции $F(t, X, U)$, вычислительная сложность которой (число арифметических операций, необходимых для однократного вычисления значения этой функции) не зависит от аргументов t, X, U и равна C_F . Вычислительная сложность затрат на реализацию алгоритма интегрирования на одном шаге интегрирования равна $C_I = C_I(l)$.

Метод мультифиниша. Покроем множество D_U некоторой сеткой с узлами $U_i = U_i(t) = \{u_{i,j}(t), i \in [1 : M], j \in [1 : m]\}$, где M — общее число узлов сетки. Поставим в соответствие системе (1) совокупность M систем ОДУ с указанными управлениями:

$$\begin{aligned} \dot{X}_1 &= F(t, X_1, U_1), & X_1(0) &= X^0; \\ \dots\dots\dots & & & \\ \dot{X}_M &= F(t, X_M, U_M), & X_M(0) &= X^0. \end{aligned} \quad (3)$$

Тогда схему приближенного решения прямой задачи методом мультифиниша можно представить в следующем виде.

1. Путем интегрирования совокупности систем ОДУ (3) находим множество точек $\{Y_i(T), i \in [1 : M]\}$, представляющее собой дискретную аппроксимацию области D_Y .

2. Сохраняем полученные наборы значений $(U_i, Y_i(T))$ в памяти используемой ЭВМ.

3. Во множестве $\{Y_i(T)\}$ находим граничные точки $\{Z_{i_j}(T), i_j \in [1:\zeta]\}$, представляющие собой дискретную аппроксимацию границы Γ_Y области достижимости; ζ — число граничных точек, $\zeta \leq M$.

4. Сохраняем в памяти ЭВМ соответствующие наборы значений $(U_{i_j}, Z_{i_j}(T))$.

5. На основе множества граничных точек $\{Z_{i_j}(T)\}$ строим подходящую непрерывную аппроксимацию $\tilde{\Gamma}_Y$ границы Γ_Y .

Задача построения дискретной или непрерывной аппроксимации границы Γ_Y представляет собой классическую задачу вычислительной геометрии — задачу построения оболочки облака точек. Методы решения этой задачи хорошо разработаны [5]. Известно много программных систем, реализующих такие методы [6]. Поэтому в настоящей работе рассматривается только шаг 1 приведенной схемы приближенного решения прямой задачи.

Обратная задача в методе мультифиниша может быть решена по следующей схеме.

1. Во множестве точек $\{Y_i(T)\}$ находим точку $Y_*(T)$, которая в используемой метрике является ближайшей к заданной точке $Y(T)$.

2. В качестве искомого управления принимаем управление из набора $(U_{i_j}, Z_{i_j}(T))$, соответствующее точке $Y_*(T)$.

Схема распараллеливания вычислений при приближенном решении прямой задачи методом мультифиниша с использованием NN -процессорной МВС имеет вид:

1) процессор $P_i, i \in [1 : NN]$, получает от $host$ -процессора векторы управлений $U_{(i-1)r+1}, U_{(i-1)r+2}, \dots, U_{ir}$;

2) интегрируя при каждом из этих управлений систему ОДУ (3), процессор находит координаты множества точек $\{Y_j(T), j \in [(i-1)r+1 : ir]\}$;

3) процессор P_i передает координаты полученных точек $host$ -процессору;

4) $host$ -процессор строит сначала дискретную, а затем непрерывную аппроксимацию границы Γ_Y области достижимости.

Здесь $r = [M/NN]$ — число векторов управления, обрабатываемых процессором P_i ; $[*]$ — символ ближайшего большего целого.

Отметим, что при большом числе узлов M задача построения дискретной и непрерывной аппроксимации границы Γ_Y области достижимости может иметь высокую вычислительную сложность и требовать распараллеливания.

При параллельном построении множества достижимости по рассмотренной схеме проблема балансировки загрузки МВС не возникает. При построении границы Γ_Y области достижимости D_Y ситуация может быть иной.

Равномерная декомпозиция точек переключения. Положим, что одну из границ множества достижимости формируют управления класса $L_U^\Gamma[0, T]$, в которых все компоненты вектора управления, кроме одной, постоянны, а компонента $u_1(t)$, например, имеет одну точку переключения:

$$u_1^\Gamma(t, t^S) = \begin{cases} 1, & t \in [0, t^S], \\ -1, & t \in (t^S, T], \end{cases} \quad u_2^\Gamma(t) = \text{const}_2, \dots, u_m^\Gamma(t) = \text{const}_m.$$

Здесь $t^S \in [0, T]$ — момент времени, в который происходит переключение управления $u_1^\Gamma(t)$.

Покроем интервал $[0, T]$ равномерной сеткой с шагом $\Delta t^S = T/M$ и узлами $t_i^S, i \in [1 : M]$. Положим, что шаг Δt^S кратен шагу Δt , так что $\Delta t^S = q\Delta t$, где $q = K/M \geq 1$ — целое число. При этом естественно принять $U_i^\Gamma = (u_1^\Gamma(t, t_i^S), u_2^\Gamma, \dots, u_m^\Gamma)^\top$. Таким образом, схема распараллеливания в данном случае имеет вид, показанный на рис. 1.

В соответствии с этой схемой на процессоре P_1 выполняется интегрирование системы ОДУ (3) при управлениях $U_1^\Gamma, \dots, U_r^\Gamma$, на процессоре P_2 — при управлениях $U_{r+1}^\Gamma, \dots, U_{2r}^\Gamma$ и далее до процессора P_{NN} , который выполняет интегрирование при управлениях $U_{(NN-1)r+1}^\Gamma, \dots, U_M^\Gamma$.

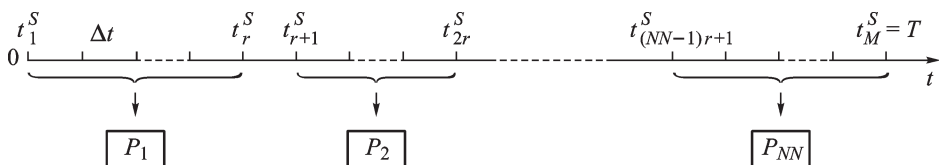


Рис. 1. Схема распараллеливания при равномерной декомпозиции точек переключения

Вычисления на процессоре P_i организуем следующим образом.

Шаг 0 (этап разгона). Исходя из начальных условий X^0 , выполняем интегрирование системы (3) при управлении $(1, u_2^\Gamma, u_3^\Gamma, \dots, u_m^\Gamma)$ от момента времени 0 до момента времени t_{ir-1}^S и сохраняем в памяти ЭВМ значения компонент векторов $X(t_{(i-1)r+1}^S) = X_{(i-1)r+1}^0$, $X(t_{(i-1)r+2}^S) = X_{(i-1)r+2}^0, \dots, X(t_{ir-1}^S) = X_{ir-1}^0$.

Шаг 1. Исходя из начальных условий $X_{(i-1)r+1}^0$, выполняем интегрирование системы (3) при управлении $(-1, u_2^\Gamma, u_3^\Gamma, \dots, u_m^\Gamma)$ от момента времени $t_{(i-1)r+1}^S$ до момента времени T .

Шаг 2. Исходя из начальных условий $X_{(i-1)r+2}^0$, выполняем интегрирование системы (3) при управлении $(-1, u_2^\Gamma, u_3^\Gamma, \dots, u_m^\Gamma)$ от момента времени $t_{(i-1)r+2}^S$ до момента времени T .

.....

Шаг r. Исходя из начальных условий X_{ir-1}^0 , выполняем интегрирование системы (3) при управлении $(-1, u_2^\Gamma, u_3^\Gamma, \dots, u_m^\Gamma)$ от момента времени t_{ir-1}^S до момента времени T .

Замечание. В рассмотренной схеме каждый из последующих процессоров повторяет этап разгона своего предыдущего процессора и только затем выполняет интегрирование системы (3) на своем интервале $[t_{(i-1)r}^S, t_{ir-1}^S]$. Если вычислительные затраты на интегрирование системы (3) велики, то может оказаться целесообразной более тонкая организация вычислений на этапе разгона — однократное интегрирование системы (3) на интервале $[0, T]$ при управлении $(1, u_2^\Gamma, u_3^\Gamma, \dots, u_m^\Gamma)$ на одном процессоре и передача необходимых начальных условий $X_1^0, X_2^0, \dots, X_M^0$ остальным процессорам вычислительной системы. Однако, если $r \gg 1$, что можно считать типичной ситуацией, выигрыш от такой организации вычислений не может быть существенным.

В нашей работе [7] показано, что при равномерной декомпозиции точек переключения время последовательного решения задачи определяется выражением

$$\tau_1 = t_{cat}q(lC_F + C_I) \sum_{j=1}^M (M - (j - 1)) \approx \frac{1}{2} t_{cat}q(lC_F + C_I) M^2,$$

а время параллельного решения задачи — выражением

$$\tau_{NN} = \sum_{j=1}^r t_{cal}q(M-j)(lC_F + C_I) \approx t_{cal}q(lC_F + C_I) \frac{M^2}{NN}.$$

Здесь и далее t_{cal} — время выполнения арифметической операции на одном процессоре используемой МВС.

Утверждение 1. Ускорение S параллельного метода на основе равномерной декомпозиции точек переключения оценивается величиной

$$S \approx \frac{NN}{2}.$$

Из утверждения 1 следует, что равномерная декомпозиция точек переключения не может обеспечить ускорение, превышающее 50% максимального потенциально возможного ускорения, равного NN . Лучшей балансировки загрузки МВС можно добиться за счет неравномерной декомпозиции точек переключения.

Неравномерная декомпозиция точек переключения. Положим, что процессор P_i выполняет интегрирование системы ОДУ (3) при управлениях $U_a^\Gamma, \dots, U_b^\Gamma$, а процессор P_{i+1} — при управлениях $U_{b+1}^\Gamma, \dots, U_c^\Gamma$, $1 \leq a < b < c \leq M$. Потребуем, чтобы времена решения задачи процессорами P_i, P_{i+1} были равны, т.е. чтобы выполнялось равенство $T_i = T_{i+1}$, $i \in [1 : (NN - 1)]$. В работе [7] показано, что для этого величины a, b, c должны удовлетворять рекуррентному уравнению

$$c^2 - 2(M+1)c + (a^2 - 2b^2 - 2Ma + 2(2M+1)b) = 0. \quad (4)$$

Если для процессора P_1 задать величины $a = a_1 = 1, b = b_1$, то из выражения (4) последовательно легко найти управления, которые должны обрабатывать процессоры P_2, P_3, \dots, P_{NN} .

Рассмотрим для примера ситуацию, когда $M = 1000, b_1 = 100$. Из уравнения (4) вытекает, что в данном случае можно обеспечить приблизительно равномерную загрузку шести процессоров (таблица). Отметим, что для процессора P_6 число точек переключения вычислено по остаточному принципу и не удовлетворяет уравнению (4).

Как показано в работе [7], при неравномерной декомпозиции точек переключения время последовательного решения задачи можно оценить выражением

$$\begin{aligned} \tau_1 &= t_{cal}q(lC_F + C_I) \left(M - 1 + \sum_{i=1}^M (M - i) \right) = \\ &= t_{cal}q(lC_F + C_I) (M - 1) (1 + 0,5M), \end{aligned}$$

а время параллельного решения — выражением

$$\tau_{NN} = t_{cal}q(lC_F + C_I) ((M + 1)b_1 - 0,5b_1^2).$$

**Пример неравномерной декомпозиции точек
переключения**

Процессор P_i	t_a^S	t_b^S	$t_b^S - t_a^S$
P_1	1	100	99
P_2	101	212	111
P_3	213	342	129
P_4	343	504	151
P_5	505	756	251
P_6	757	1000	243

Таким образом, имеем утверждение 2.

Утверждение 2. Ускорение S параллельного метода на основе неравномерной декомпозиции точек переключения оценивается величиной

$$S = \frac{(M-1)(1+0,5M)}{(M+1)b_1 - 0,5b_1^2} \approx \frac{M^2}{2Mb_1 - b_1^2}.$$

Из утверждения 2 следует, что в условиях рассмотренного примера ускорение можно записать как

$$S \approx \frac{1000^2}{2 \cdot 1000 \cdot 100 - 100^2} \approx 5,26.$$

Заметное отличие ускорения от максимально возможного (равного 6) объясняется тем, что по причинам, указанным ранее, загрузка последнего процессора P_6 далека от оптимальной.

Явно решить рекуррентное уравнение (4) не удастся. Поэтому число точек переключения b_1 , обрабатываемых процессором P_1 , придется подбирать таким образом, чтобы вычислительная загрузка всех процессоров, включая последний, была примерно одинакова. При этом критерий окончания перебора удобно строить, исходя из того, что при несбалансированности загрузки процессора P_{NN} корни уравнения (4) оказываются комплексными.

Метод на основе аппроксимации векторного поля системы ОДУ. Идея данного метода состоит в следующем.

1. Предварительно множество $[0, T] \times D_X \times D_U$ покрываем некоторой сеткой Ω с узлами $(t_\alpha, x_{j,\beta}, u_{k,\gamma})$, где $j \in [1 : n]$, $k \in [1 : m]$, $\alpha \in [1 : N^t]$, $\beta \in [1 : N_j^X]$, $\gamma \in [1 : N_k^U]$.

2. Предварительно во всех узлах сетки Ω вычисляем значения функции $F(t, X, U)$ и сохраняем в памяти ЭВМ.

3. Тем или иным методом численно интегрируем систему ОДУ (3) при управлении $U_i, i \in [1 : M]$, используя в качестве требуемых значе-

ний функции $F(t, X, U)$ соответствующие интерполяции или аппроксимации сохраненных в памяти ЭВМ значений этой функции.

Здесь N_j^X — число узлов сетки Ω по фазовой переменной x_j ; N_k^U — число узлов той же сетки по компоненте u_k вектора управления.

Ускорение вычислений в данном методе достигается за счет того, что этапы 1 и 2 выполняются предварительно, так что при выполнении этапа 3 не требуется вычислять значения правых частей системы ОДУ (3).

Для того чтобы использовать весь потенциал рассматриваемого метода, значения функции $F(t, X, U)$ в узлах сетки Ω следует хранить в оперативной памяти компьютера. Примем, что длина вещественного числа равна 4 байтам. Пусть также $N^X = \prod_{j=1}^n N_j^X$, $N^U = \prod_{k=1}^m N_k^U$. Тогда с учетом того факта, что общее число узлов сетки Ω равно $N^t N^X N^U$, имеем утверждение 3.

Утверждение 3. Требуемый объем памяти ЭВМ для хранения значений функции $F(t, X, U)$ в узлах сетки Ω равен

$$W = 4nN^t N^X N^U \text{ байт.}$$

Положим, что $n = 5$, $m = 3$; $N^t = 10$; $N_j^X = 10$, $j \in [1 : n]$; $N_k^U = 10$, $k \in [1 : m]$. Из утверждения 3 следует, что при этом требуемый объем памяти равен

$$W = 4 \cdot 5 \cdot 10^9 = 2 \cdot 10^{10} \text{ байт} \approx 20 \text{ Гбайт.}$$

В простейшем случае в качестве значения $F(t, X, U)$ используется значение этой функции в ближайшем узле сетки Ω . Такая схема вычислений применима при построении области D_Y в грубом приближении и означает использование кусочно-постоянной интерполяции функции $F(t, X, U)$.

В работе [7] показана справедливость следующего утверждения.

Утверждение 4. Ускорение S последовательного метода на основе кусочно-постоянной интерполяции функции $F(t, X, U)$ равно

$$S \approx \frac{lC_F + C_I}{C_I} \approx l \frac{C_F}{C_I} \gg 1. \quad (5)$$

Дальнейшего уменьшения времени решения задачи можно добиться путем организации параллельных вычислений. Схема распараллеливания при этом принципиально не отличается от схемы, рассмотренной ранее.

Кусочно-линейная интерполяция функции $F(t, X, U)$. При том же числе узлов сетки Ω большую точность вычислений можно обеспечить путем использования линейных интерполяций компонент функции $F(t, X, U)$. При этом рассмотренная ранее идея последовательного

Ускорение *последовательного* метода на основе кусочно-линейной интерполяции функции $F(t, X, U)$ определяется так же, как в методе на основе кусочно-постоянной интерполяции.

Кусочно-линейная аппроксимация функции $F(t, X, U)$. Известной проблемой интерполяции многомерных функций является невозможность использования произвольных сеток. Поэтому в вычислительной практике обычно используют не интерполяцию функции $F(t, X, U)$, а ее аппроксимацию на основе метода наименьших квадратов (МНК).

Аналогично предыдущему пункту методике линейной аппроксимации функции $F(t, X, U)$ рассмотрим на примере i -й компоненты этой функции $f_i(t, X, U) = f(t, X, U)$. Для кусочно-линейной аппроксимирующей функции в этом случае имеем следующую СЛАУ относительно неизвестных коэффициентов $a_{J,0}, a_{J,1}, a_{J,2}, \dots, a_{J,n+m+1}$:

$$\begin{aligned} a_{J,0} + (A_J, V_{J,1}) &= f(V_{J,1}); \\ a_{J,0} + (A_J, V_{J,2}) &= f(V_{J,2}); \\ &\dots\dots\dots \\ a_{J,0} + (A_J, V_{J,d}) &= f(V_{J,d}), \end{aligned}$$

где d — число используемых узлов ячейки Ω_J ; $(n+m+2) < d \leq 2^{n+m+1}$.

В соответствии с методикой МНК решение этой системы получаем в виде [7]

$$\mathbf{A}_J = (\mathbf{V}_J^T \mathbf{V}_J)^{-1} \mathbf{V}_J^T \mathbf{f}_J, \quad (8)$$

где $\mathbf{A}_J = (a_{J,0}, a_{J,1}, \dots, a_{J,n+m+1})^T$ — $(n+m+2)$ -мерный вектор; $\mathbf{f}_J = (f(V_{J,1}), f(V_{J,2}), \dots, f(V_{J,d}))^T$ — d -мерный вектор; $\mathbf{V}_J = (V_{J,1}, V_{J,2}, \dots, V_{J,d})^T$ — $d \times (n+m+2)$ - матрица, которая в качестве столбцов имеет $(n+m+2)$ -мерные векторы $V_{J,i} = (1, t_J, x_{1,J}, \dots, x_{n,J}, u_{1,J}, \dots, u_{m,J})^T$. Важно, что, кроме вырожденного случая, когда все точки $f(V_{J,1}), f(V_{J,2}), \dots, f(V_{J,d})$ лежат в одной гиперплоскости, решение СЛАУ (8) существует и единственно.

Объем памяти ЭВМ, необходимый для хранения коэффициентов $a_{J,0}, a_{J,1}, a_{J,2}, \dots, a_{J,n+m+1}$ кусочно-линейной аппроксимации функции $F(t, X, U)$ на сетке Ω , определяется утверждением 5. Ускорение *последовательного* метода на основе кусочно-линейной аппроксимации функции $F(t, X, U)$ определяется утверждением 4.

Большое число вариантов рассмотренного подхода к кусочно-линейной аппроксимации функции $F(t, X, U)$ можно получить, привлекая результаты теории планирования эксперимента. Так, каждый узел сетки Ω можно использовать в качестве центральной точки для построения регрессионного плана эксперимента первого порядка, удовлетворяющего критерию D -оптимальности, A -оптимальности, E -оптимальности и т.д. [8]. СЛАУ (8) при этом сохраняет свою структуру.

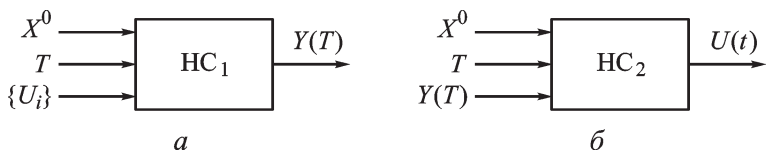


Рис. 2. Нейросетевая аппроксимация области достижимости

Замечание. Повысить точность аппроксимации функции $F(t, X, U)$ можно, используя квадратичную аппроксимацию. При этом естественно ориентироваться на планы эксперимента второго порядка, такие как центрально-композиционные планы, композиционные планы Хартли и Вестлейка [9]. В алгоритмическом отношении для аппроксимации удобно использовать композицию одномерных полиномов (метод Брандона) [10].

Нейросетевые методы. Для аппроксимации многомерных функций чаще всего используют многослойные нейронные сети прямого распространения (многослойные персептроны) [11]. В качестве аппроксиматора функций могут использоваться также радиально-базисные нейронные сети. При прочих равных условиях аппроксимация этими сетями требует значительно большего числа нейронов, чем аппроксимация многослойными персептронами. Поэтому радиально-базисные нейронные сети в работе не рассматриваются.

Для решения прямой и обратной задач построения области достижимости используем нейронные сети $НС_1$, $НС_2$ соответственно (рис. 2). Входами нейронной сети $НС_1$ являются вектор начальных условий X^0 , конечный момент времени T и совокупность управлений $\{U_i, i \in [1 : M]\}$, а выходами — вектор $Y(T)$. Нейронная сеть $НС_2$ имеет в качестве входов величины X^0 , T , $Y(T)$, а в качестве выходов — вектор $U(t) \in D_U$.

В случае, когда известно множество допустимых управлений $D_U^\Gamma \subseteq D_U$, приводящих систему (3) на границу области достижимости Γ_Y , могут быть использованы сети, аналогичные сетям $НС_1$, $НС_2$. Отличие состоит в том, что здесь управления $\{U_i\}$ должны принадлежать множеству D_U^Γ , а вектор $Y(T)$ — границе Γ_Y множества достижимости. В случае сложной топологии границы Γ_Y может оказаться необходимым использовать различные нейронные сети для аппроксимации разных фрагментов этой границы.

Пример. Рассмотрим летательный аппарат, уравнения движения центра масс которого в нормальной земной системе координат $Oxyz$ описываются системой нелинейных дифференциальных уравнений:

$$\begin{aligned}
\dot{v} &= g(u_T - \sin \Theta); \\
\dot{\Theta} &= g/v(u \cos \gamma_C - \cos \Theta); \\
\dot{\Psi} &= -gu \sin \gamma_C/v \cos \Theta; \\
\dot{x} &= v \cos \Theta \cos \Psi; \\
\dot{y} &= v \sin \Theta; \\
\dot{z} &= -v \cos \Theta \sin \Psi,
\end{aligned}
\tag{9}$$

где v — скорость летательного аппарата; Θ — угол наклона траектории; Ψ — угол поворота траектории; y — высота летательного аппарата; u_T — тангенциальная перегрузка; u — нормальная перегрузка; γ_C — скоростной угол крена; g — ускорение свободного падения [1].

Управлениями летательного аппарата являются тангенциальная перегрузка, нормальная перегрузка и скоростной угол крена, так что $U = (u_T, u, \gamma_C)^T$. На управления наложены ограничения

$$\begin{aligned}
u_T^{\min} \leq u_T \leq u_T^{\max}; \quad u_T^{\min} = -1,6; \quad u_T^{\max} = 0,6; \\
|u| \leq u^{\max}; \quad u^{\max} = 8; \\
|\gamma_C| \leq \pi.
\end{aligned}$$

Дальняя, ближняя и боковая границы области достижимости системы (9) формируются управлениями, принадлежащими классу кусочно-постоянных управлений с одной точкой переключения [1]. Ограничимся рассмотрением дальней границы области достижимости. Для каждого допустимого угла крена $\gamma_C(t) = \text{const}$ структура управлений, формирующих эту границу, приведена на рис. 3.

Метод мультифиниша. Для экспериментального исследования эффективности метода мультифиниша разработана МРІ-программа [4], реализующая этот метод. Эксперименты выполнены на виртуальном кластере, созданном с помощью программной системы VMware [12] и функционирующим под управлением свободно распространяемой операционной системы Ubuntu [13]. Результаты экспериментов приведены на рис. 4, из которого следует, что при числе процессоров от 2 до 10 достигается ускорение, близкое к расчетному.

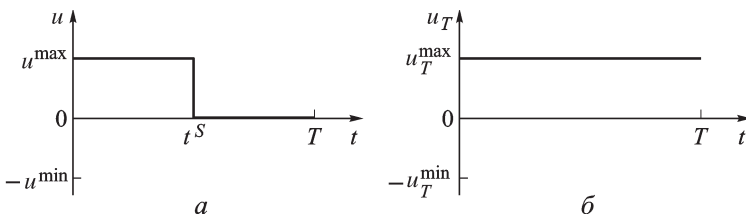


Рис. 3. Структура управлений, формирующих дальнюю границу области достижимости

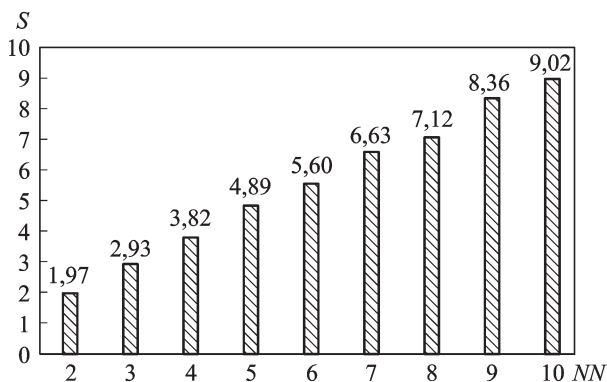


Рис. 4. Ускорение метода мультифиниша

Приведенные на рис. 4 экспериментальные результаты получены при следующих значениях параметров задачи: $v(0) = 330$ м/с; $\Theta(0) = 0$; $\Psi(0) = 0$; $x(0) = 0$; $y(0) = 10000$ м; $z(0) = 0$; $T = 15$ с.

Аппроксимация векторного поля системы ОДУ. Рассмотрим плоскую задачу (9), когда $\Psi(t) \equiv 0$, $t \in [0, T]$, $T = 30$ с. Предположим, что область фазового пространства D_X системы (9) определяется неравенствами

$$1,6v_{зв} \leq v(t) \leq 1,9v_{зв}; \quad -90^\circ \leq \Theta(t) \leq 90^\circ; \quad \Psi(t) \equiv 0;$$

$$0 \leq x(t) \leq 5500 \text{ м}; \quad 0 \leq y(t) \leq 2700 \text{ м}; \quad z(t) = 0.$$

Здесь $v_{зв} = 331,46$ м/с — скорость звука в воздухе. Принимаются следующие начальные условия летательного аппарата: $v(0) = 132,584$ м/с; $\Theta(0) = 0$; $\Psi(0) = 0$; $x(0) = 0$; $y(0) = 1000$ м; $z(0) = 0$; $\gamma_c(0) = 0$.

В соответствии с методикой покроем интервал времени $[0, T]$ равномерной сеткой с 20 узлами. Указанную область D_X фазового пространства системы (9) покроем равномерной сеткой так, что число шагов этой сетки равно 18, 20, 20, 18 по измерениям v , Θ , x , y соответственно. Наконец, в качестве моментов времени t^S переключения управления u рассмотрим $M = 30$ узлов равномерной сетки, покрывающей интервал времени $[0, T]$ (рис. 3, а).

Численные эксперименты выполнены для кусочно-постоянной аппроксимации компонент функции $F(t, X, U)$ на указанной сетке (рис. 5).

Эксперименты показали, что метод обеспечивает погрешность в определении расстояния от начала координат до дальней границы области достижимости, не превышающую 4%. В качестве точного положения этой границы принималось положение, полученное путем численного интегрирования системы (9) при соответствующем управлении.

Нейросетевая аппроксимация. В данном случае начальный угол поворота траектории $\Psi(0)$ и начальные координаты $x(0)$, $y(0)$, $z(0)$ ле-

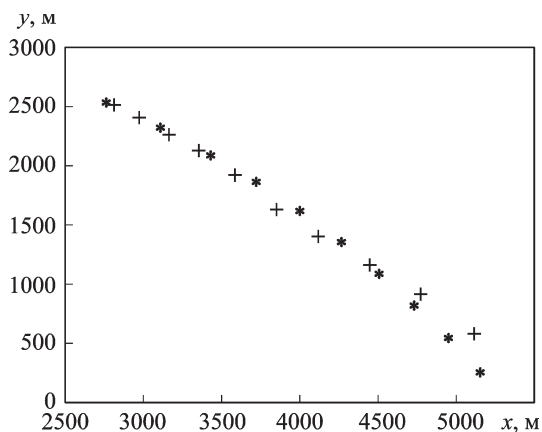


Рис. 5. Дальняя граница области достижимости (“плоская” задача):

* — точная граница; + — приближенная граница

тательного аппарата приняты равными нулю. Так что исходное положение летательного аппарата определяется начальной скоростью $v(0)$ и начальным углом наклона траектории $\Theta(0)$: $1,6v_{зв} \leq v(0) \leq 1,9v_{зв}$; $-\frac{\pi}{8} \leq \Theta(0) \leq \frac{\pi}{8}$. Варьируемыми параметрами являются точки переключения t^S нормальной перегрузки u и значения скоростного угла крена γ_C . Полагается, что $t^S \in [0, T]$, где $T \in [5, 15]$ с. Поскольку область достижимости симметрична относительно плоскости Oxy , принято, что $\gamma_C \in [0, \pi]$.

Для указанной задачи были построены нейронные сети $НС_1$, $НС_2$. Во всех случаях обучающая выборка состояла из 4500 элементов, из которых 20% случайно выбранных точек использовались в качестве проверочных для определения момента окончания обучения. Начальные веса и пороговые значения нейронов устанавливались в соответствии с алгоритмом инициализации Нгуен-Видроу (Nguyen-Widrow) [11]. Обучение каждой сети выполнялось в среднем 10 раз с различными начальными весами и пороговыми значениями, на основе чего выбиралась сеть с минимальным значением ошибки. Для исследования точности обученной сети использовалась тестовая выборка, содержащая более 20000 элементов. Все эксперименты выполнены в программной среде MatLab 6 [11].

Сеть $НС_1$. Входной слой сети $НС_1$ состоит из пяти элементов, соответствующих величинам $v(0)$, $\Theta(0)$, T , γ_C , t^S . Сеть содержит два скрытых слоя, содержащих одинаковое число нейронов с функцией активации в виде гиперболического тангенса. Выходной слой сети состоит из трех нейронов с линейной функцией активации, соответствующих координатам $x(T)$, $y(T)$, $z(T)$ конечного положения летательного аппарата.

Исследование показало, что при числе нейронов в сети, равном 22, максимальная относительная ошибка определения нейронной сетью

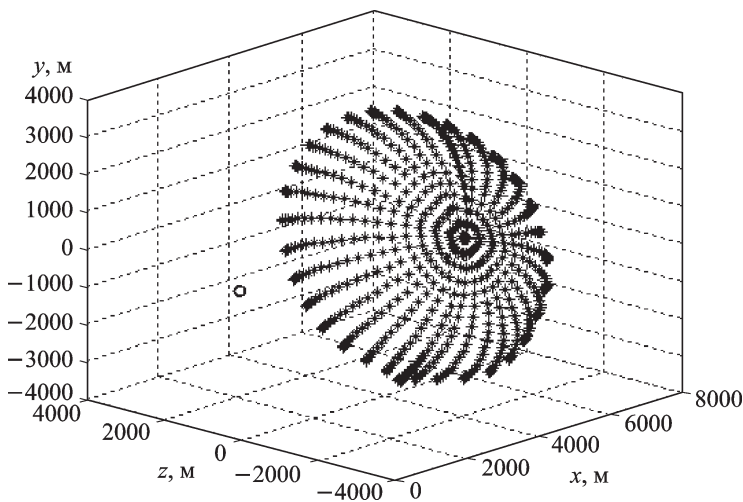


Рис. 6. Нейросетевая аппроксимация дальней границы области достижимости

HC_1 координат точек границы области достижимости не превышает 12 %. С ростом числа нейронов эта ошибка быстро уменьшается и при 40 нейронах не превышает 4 %. Результаты исследования приведены на рис. 6, на котором показана аппроксимация дальней границы области достижимости нейронной сетью HC_1 с 32 нейронами для случая $T = 15$ с, $v(0) = 1,9v_{зв}$, $\Theta(0) = \pi/10$.

Сеть HC_2 аналогична сети HC_1 . Входной слой сети HC_2 состоит из пяти элементов, соответствующих величинам $v(0)$, $\Theta(0)$, T , α , β , где α , β — углы, задающие требуемое направление в пространстве. Выходной слой состоит из двух нейронов, соответствующих положению точки переключения t^S и значению скоростного угла крена γ_C .

Для оценки точности аппроксимации сети определяются координаты точек границы области достижимости, соответствующие управлению, получаемому на выходе нейронной сети, и сравниваются с эталонными координатами этих точек, вычисленными путем интегрирования системы ОДУ (9).

Исследование показало, что максимальная относительная ошибка определения нейронной сетью HC_2 координат точек границы области достижимости при 22 нейронах в сети не превышает 9 %, а при 40 нейронах составляет ~ 3 %.

Детально результаты исследования метода нейросетевой аппроксимации границы области достижимости летательного аппарата приведены в работах [14, 15].

Заключение. В работе рассмотрены три метода построения области достижимости динамической системы — метод мультифиниша, метод на основе аппроксимации векторного поля соответствующей системы ОДУ, а также метод на основе нейронных сетей. Для первых

двух методов предложены параллельные алгоритмы, ориентированные на вычислительные системы класса MIMD. Эффективность всех рассмотренных методов исследована на примере построения границы области достижимости летательного аппарата.

Проведенное исследование показало перспективность практического использования всех рассмотренных методов. Тот или иной метод может быть рекомендован в зависимости от вычислительной сложности правых частей системы ОДУ, описывающей исследуемую динамическую систему, а также от располагаемых вычислительных ресурсов.

Кроме рассмотренных методов интерес представляет использование для приближенного построения области достижимости нечетких множеств. Отметим также целесообразность исследования метода, основная идея которого состоит в следующем: для представительного множества начальных условий динамической системы тем или иным образом строим совокупность областей достижимости; границу каждой области аппроксимируем подходящей функцией; функциональную зависимость коэффициентов этой функции от начальных условий аппроксимируем, например, с помощью нейронной сети. Можно ожидать, что такой метод позволит получить приемлемую точность аппроксимации множества достижимости для широкого набора начальных условий.

Авторы благодарят А.Г. Трофимова за идею использования нейронных сетей для аппроксимации границ области достижимости, а также К.О. Вишневецкого и Д.С. Кулеша за проведение вычислений с использованием методов мультифиниша и аппроксимации векторного поля.

СПИСОК ЛИТЕРАТУРЫ

1. Воронев Е. М., Карпунин А. А. Алгоритм оценки границ области достижимости летательного аппарата с учетом тяги // Вестник МГТУ. Сер. "Приборостроение". – 2007. – № 4(69). – С. 81–99.
2. Гурман В. И., Квоков В. И., Ухин М. Ю. Приближенные методы оптимизации управления летальным аппаратом // Автоматика и телемеханика. – 2008. – № 4. – С. 191–201.
3. Ортега Дж., Пул У. Введение в численные методы решения дифференциальных уравнений / Пер. с англ., под ред. А.А. Абрамова. – М.: Наука, 1986. – 288 с.
4. Воеводин В. В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2004. – 608 с.
5. Скворцов А. В. Обзор алгоритмов построения триангуляции Делоне // Вычислительные методы и программирование. – 2002. – Т. 3. – С. 14–39.
6. Directory of Computational Geometry Software [Электронный ресурс]. /<http://www.geom.uiuc.edu/software/cglist/>
7. Воронев Е. М. Численные методы построения области достижимости динамической системы [Электронный ресурс] // Электронное научно-техническое издание: наука и образование. – 2010. – № 1. <http://www.technomag.edu.ru>

8. Карпенко А. П. Аппроксимация функции предпочтений лица, принимающего решения, в задаче многокритериальной оптимизации. 1. Методы на основе планов первого порядка [Электронный ресурс] // Электронное научно-техническое издание: наука и образование. – 2008. – № 3. <http://www.technomag.edu.ru>
9. Карпенко А. П. Аппроксимация функции предпочтений лица, принимающего решения, в задаче многокритериальной оптимизации. 2. Методы на основе планов второго порядка [Электронный ресурс] // Электронное научно-техническое издание: наука и образование. – 2008. – № 3. <http://www.technomag.edu.ru>
10. Хохлов С. Ф., Школа О. И. Применение метода Брандона для обработки экспериментальных данных // Вопросы химии и химической технологии. – 1973. – Вып. 28. – С. 204–207.
11. Медведев В. С., Потемкин В. Г. Нейронные сети. Matlab 6. – М.: Диалог-МИФИ, 2001. – 496 с.
12. VM ware [Электронный ресурс]. <http://www.vmware.com/>
13. U b u n t u [Электронный ресурс]. <http://www.ubuntu.com/>
14. Козлова О. Г. Нейросетевая аппроксимация границы области достижимости летательного аппарата [Электронный ресурс] // Электронное научно-техническое издание: наука и образование. – 2009. – № 7. <http://www.technomag.edu.ru/doc/129990.html>
15. Козлова О. Г. Нейросетевая аппроксимация границы области достижимости летательного аппарата. Трехмерный случай [Электронный ресурс] // Электронное научно-техническое издание: наука и образование. // – 2009. – № 8. <http://www.technomag.edu.ru/doc/130282.html>

Статья поступила в редакцию 28.01.2010

Евгений Михайлович Воронов родился в 1940 г., окончил в 1963 г. МВТУ им. Н.Э. Баумана и в 1969 МГУ им. М.В. Ломоносова. Д-р техн. наук, профессор кафедры “Системы автоматического управления” МГТУ им. Н.Э. Баумана. Автор 150 научных работ в области теории управления и ее приложений.



Ye.M. Voronov (b. 1940) graduated from the Bauman Moscow Higher Technical School in 1963 and the Lomonosov Moscow State University in 1969. D. Sc. (Eng.), professor of “Automatic Control Systems” department of the Bauman Moscow State Technical University. Author of 150 publications in the field of control theory and its applications.

Анатолий Павлович Карпенко родился в 1947 г., окончил МВТУ им. Н.Э. Баумана в 1975 г. Автор 141 научной работы в области информационных технологий.

A.P. Karpenko (b. 1947) graduated from the Bauman Moscow Higher Technical School in 1975. Author of 141 publications in the field of information technologies.

Оксана Геннадьевна Козлова родилась в 1984 г., окончила МВТУ им. Н.Э. Баумана в 2009 г. Автор пяти научных работ в области информационных технологий.

O.G. Kozlova (b. 1984) graduated from the Bauman Moscow Higher Technical School in 2009. Author of 5 publications in the field of information technologies.

Владимир Андреевич Федин родился в 1986 г., окончил МВТУ им. Н.Э. Баумана в 2008 г. Автор пяти научных работ в области информационных технологий.

V.A. Fedin (b. 1986) graduated from the Bauman Moscow Higher Technical School in 2008. Author of 5 publications in the field of information technologies.