

Система поддержки принятия решений при выборе параллельного аппаратно-программного комплекса для построения областей достижимости летательного аппарата

А.В. Быстров, А.П. Карпенко, О.Г. Козлова, В.А. Федин

МГТУ им. Н.Э.Баумана

Приводится постановка задача приближенного построения области достижимости динамической системы. Рассматриваются численные методы решения этой задачи на кластерных вычислительных системах, графических процессорных устройствах и нейросетевых ускорителях. Обсуждаются основные принципы разработки системы поддержки принятия решений, предназначенной для оптимального проектирования аппаратно-программного комплекса, обеспечивающего параллельное построение границ области достижимости летательного аппарата с заданной точностью и за заданное время.

1. Введение

Для приближенного построения области достижимости летательного аппарата (ОДЛА) могут быть использованы методы на основе многократного интегрирования исходной модельной системы обыкновенных дифференциальных уравнений (ОДУ), близкие алгоритмы, использующие полиномиальную или нейросетевую аппроксимацию правых частей этих уравнений, алгоритмы приближенного построения аппроксимаций границ ОДЛА и т.д. [1].

Указанные методы могут быть реализованы на многопроцессорных вычислительных системах с общей или распределенной памятью, системах на основе графических процессорных устройств, а также на нейросетевых ускорителях [2,3]. При этом каждому из методов может быть поставлено в соответствие несколько алгоритмов, каждый из которых может быть несколькими способами отображен на архитектуру каждой из рассматриваемых вычислительных систем. В силу многообразия вариантов состава целевого аппаратно-программного комплекса, реализующего построение ОДЛА на борту летального аппарата, возникает задача разработки программной системы для автоматизированного синтеза этого комплекса. В работе данная система рассматривается, как система поддержки принятия решений (СППР).

Полагается, что СППР функционирует на инструментальном гетерогенном вычислительном комплексе, который включает в себя все указанные классы вычислительных систем. Это дает возможность корректной оценки времени решения задачи, а также позволяет выполнять на инструментальном комплексе отладку целевых программ.

2. Постановка задачи

Рассмотрим динамическую систему

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

где $X = X(t)$ – n -мерный вектор фазовых переменных системы, $U = U(t)$ – m -мерный вектор управлений, X^0 – n -мерный вектор начальных условий, $t \in [0, T]$, $F(t, X, U)$ – n -мерная вектор-функция. Задано множество допустимых управлений D_U , так что $U \in D_U \subset L_U[0, T]$, где $L_U[0, T]$ – некоторое пространство m -мерных функций, определенных на интервале $[0, T]$, например, пространство кусочно-постоянных функций [4].

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

Областью достижимости $D = D(T, X^0)$ системы (1) назовем множество всех возможных значений вектора $Y(T)$, которые принимаются на решениях системы (1) при начальных условиях X^0 и выполнении условия $U \in D_U$. Ставится задача приближенного построения области достижимости D .

В области достижимости D может находиться одна или несколько, вообще говоря, динамических запрещенных областей $\Omega_i(t)$, $i \in [1: \kappa]$. Очевидно, что в результате область достижимости D трансформируется. Сохраним за трансформированной препятствиями область достижимости прежнее обозначение D .

Для решения поставленной задачи, очевидно, достаточно построить границу соответствующей области достижимости. В некоторых случаях могут быть известны множества управлений $D_U^\Gamma \in L_U^\Gamma[0, T] \subseteq L_U[0, T]$, которые приводят систему (1) на границу Γ области достижимости D [4]. В этом случае задача сводится к построению границы Γ . Здесь $L_U^\Gamma[0, T]$ - некоторое подпространство функционального пространства $L_U[0, T]$.

2. Методы решения задачи

Метод мультифиниша. Покроем множество D_U некоторой сеткой с узлами U_1, U_2, \dots, U_M . Поставим в соответствие системе (1) совокупность M систем ОДУ с указанными управлениями:

$$\left\{ \begin{array}{l} \dot{X}_1 = F(t, X_1, U_1), X_1(0) = X^0, \\ \dots \\ \dot{X}_M = F(t, X_M, U_M), X_M(0) = X^0. \end{array} \right. \quad (2)$$

Тогда схему приближенного построения области достижимости D методом мультифиниша можно представить в следующем виде.

1) С помощью того ли иного метода интегрирования совокупности систем (2) находим множество точек $\{Y_i(T)\}$, представляющее собой дискретную аппроксимацию области D . Запоминаем полученные наборы значений $(U_i, Y_i(T))$; $i \in [1: M]$.

2) Во множестве $\{Y_i(T)\}$ находим граничные точки $\{Z_j(T)\}$, представляющие собой дискретную аппроксимацию Γ_d границы Γ области достижимости. Запоминаем соответствующие наборы значений $(U_j, Z_j(T))$; $j \in [1: \zeta]$.

3) На основе точек $\{Z_j(T)\}$ строим подходящую непрерывную аппроксимацию Γ_c границы Γ .

Модифицированный метод мультифиниша использует комбинацию рассмотренного метода мультифиниша с методом аппроксимации векторного поля системы (1), т.е. подходящую аппроксимацию вектор-функции $F(t, X, U)$.

Общая схема метода имеет следующий вид.

1). Покрываем множество $[0, T] \times D_X \times D_U$ некоторой сеткой $\Delta = \{t_a, x_{i,b}, u_{k,c}\}$ с узлами $(t_a, x_{i,b}, u_{k,c})$; $i \in [1: n]$, $k \in [1: m]$.

2). Во всех узлах сетки Δ вычисляем значения функции $F(t_a, X_b, U_c)$ и запомним эти значения.

3). Строим некоторую функцию $\tilde{F}(t, X, U)$, аппроксимирующую функцию $F(t, X, U)$.

4) По схеме метода мультифиниша интегрируем совокупность систем ОДУ (2), используя в качестве требуемых значений функции $F(t, X, U)$ значения функции $\tilde{F}(t, X, U)$. Строим дискретную Γ_d , а затем непрерывную аппроксимацию Γ_c границы Γ .

Рассматривается два класса методов построения функции $\tilde{F}(t, X, U)$ - методы на основе полиномиальной МНК-аппроксимации функции $F(t, X, U)$ и методы нейросетевой аппроксимации этой функции [5]. Отметим, что из числа методов первого класса в алгоритмическом отношении удобно использовать композицию одномерных полиномов (метод Брандона) [6].

Для интегрирования совокупности систем ОДУ (2) предполагается использование явных, неявных, а также соответствующих нейросетевых методов интегрирования.

Для построения непрерывной границы Γ_c имеется в виду использование полиномиальной или нейросетевой аппроксимацию границы Γ_d .

Далее полагается, что известен класс управлений, приводящих систему (1) на границу области достижимости Γ , и что этот класс представляет собой релейные управления с не более, чем одной точкой переключения каждого из компонентов вектора $U(t)$ [4].

3. Распараллеливание вычислений

В методе мультифиниша при приближенном построении границы Γ естественным образом выделяется два этапа:

1) интегрирование системы ОДУ (2) и построение дискретной аппроксимации Γ_d границы области достижимости Γ ;

2) построение непрерывной аппроксимации Γ_c этой границы.

Этап 1. Для решения задачи первого этапа рассматривается использование многопроцессорной вычислительной системы с распределенной памятью (вычислительного кластера), графического процессорного устройства (ГПУ) с архитектурой CUDA, а также нейросетевой вычислительной системы (НСВС).

Схемы распараллеливания вычислений первого этапа и особенности отображения метода мультифиниша на архитектуры вычислительного кластера и ГПУ рассмотрены в наших работах [2,3]. В этих же работах приведены соответствующие результаты исследования эффективности распараллеливания.

Этап 2. Построение границы Γ_c также может быть выполнено на вычислительных кластерах, ГПУ и на НСВС. Использование с этой целью НСВС рассмотрено в наших работах [7, 8].

В модифицированном методе мультифиниша аналогично выделяется три этапа.

1) построение функции $\tilde{F}(t, X, U)$, аппроксимирующей функцию $F(t, X, U)$;

2) интегрирование системы (2) и построение дискретной аппроксимации Γ_d границы области достижимости Γ с использованием функции $\tilde{F}(t, X, U)$ вместо функции $F(t, X, U)$;

3) построение непрерывной аппроксимации Γ_c этой границы.

Среди указанных этапов новым, по сравнению с этапами метода мультифиниша, является только первый этап. На этом этапе полиномиальная МНК-аппроксимация функции $F(t, X, U)$ может быть реализована на вычислительных кластерах и ГПУ, а нейросетевая аппроксимация – на указанных системах, а также на НСВС.

Метод мультифиниша можно считать частным случаем модифицированного метода мультифиниша, когда $\tilde{F}(t, X, U) = F(t, X, U)$. Поэтому будем далее говорить только о последнем из этих методов и называть его просто «метод мультифиниша».

4. Основные функции системы поддержки принятия решений

Для приближенного построения границ ОДЛА может быть использована иерархическая совокупность численных методов, представленная на рисунке 1. Здесь полагается, что «Методы построения границы Γ_d » включают в себя и методы интегрирования систем ОДУ (2).



Рис. 1. Иерархия методов приближенного построения границы ОДЛА

Обозначим $M_1 = \{M_{1,i}\}$ - рассматриваемую совокупность методов аппроксимации функции $F(t, X, U)$. Аналогичные обозначения M_2, M_3 введем для совокупностей методов построения границ Γ_d, Γ_c соответственно.

Каждому из методов M_1, M_2, M_3 можно поставить в соответствие некоторую совокупность алгоритмов $A_{i,j} = \{A_{i,j,k}(\mathbf{P}_{i,j,k})\}$, где $A_{i,j,k}$ - k -й алгоритм, реализующий метод $M_{i,j}$, а $\mathbf{P}_{i,j,k}$ - вектор свободных параметров этого алгоритма. В случае алгоритма $A_{1,j,k}$, например, этот вектор может включать в себя порядок аппроксимирующего полинома; в случае алгоритма $A_{2,j,k}$ - число узлов сетки, покрывающей множество D_U ; в случае алгоритма $A_{3,j,k}$ - также порядок аппроксимирующего полинома.

Назовем проектируемую бортовую вычислительную систему «целевой», а вычислительные системы, на которых функционирует рассматриваемая СППР, «инструментальными». В качестве вариантов реализации целевой ЭВМ рассмотрим следующие классы систем: вычислительный кластер $C_1 = \{C_1(\mathbf{P}_1)\}$, ГПУ $C_2 = \{C_2(\mathbf{P}_2)\}$ и НСВС $C_3 = \{C_3(\mathbf{P}_3)\}$, где $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ - векторы свободных параметров соответствующих ЭВМ (например, число процессоров для вычислительного кластера, число мультипроцессоров для ГПУ, число нейрончиков для НСВС).

Множество вариантов отображения алгоритма $A_{i,j,k}(\mathbf{P}_{i,j,k})$ на архитектуру системы $C_l(\mathbf{P}_l)$ обозначим

$$E(A_{i,j,k}(\mathbf{P}_{i,j,k}), C_l(\mathbf{P}_l)) = \{E_m(A_{i,j,k}(\mathbf{P}_{i,j,k}), C_l(\mathbf{P}_l))\}.$$

Задача оптимального проектирования аппаратно-программного комплекса, обеспечивающего построение областей достижимости летательного аппарата за заданное время с заданной точностью, состоит в многокритериальном выборе методов $M_{1,i^*}, M_{2,j^*}, M_{3,k^*}$, соответствующих алгоритмов $A_{1,i^*,j^*}(\mathbf{P}_{1,i^*,j^*}^*), A_{2,k^*,j^*}(\mathbf{P}_{2,k^*,j^*}^*), A_{3,m^*,n^*}(\mathbf{P}_{3,m^*,n^*}^*)$, целевой ЭВМ $C_{q^*}(\mathbf{P}_{q^*}^*)$ и, наконец, в выборе вариантов отображения указанных алгоритмов на архитектуру выбранной ЭВМ.

Процесс решения задачи оптимального проектирования является итерационным и включает в себя несколько вложенных циклов:

- выбор методов $M_{1,i}, M_{2,j}, M_{3,k}$;
- выбор соответствующих алгоритмов $A_{1,i,l}, A_{2,j,m}, A_{3,k,n}$;
- выбор значений параметров $\mathbf{P}_{1,i,l}, \mathbf{P}_{2,j,m}, \mathbf{P}_{3,k,n}$ этих алгоритмов;
- выбор класса целевой вычислительной системы C_q ;
- выбор значений вектора параметров \mathbf{P}_q и, тем самым, целевой системы $C_q(\mathbf{P}_q)$;
- выбор вариантов отображения алгоритмов $A_{1,i,l}(\mathbf{P}_{1,i,l}), A_{2,j,m}(\mathbf{P}_{2,j,m}), A_{3,k,n}(\mathbf{P}_{3,k,n})$ на вычислительную систему $C_q(\mathbf{P}_q)$;
- построение аппроксимации границы Γ_c ОДЛА на инструментальной вычислительной системе класса C_q с помощью выбранных методов, алгоритмов и их параметров;
- оценка точности полученной аппроксимации;
- в случае неудовлетворительной точности, последовательные попытки изменить значения параметров алгоритмов, выбрать другой алгоритм и, наконец, - другой метод;
- оценка времени решения задачи на системе $C_q(\mathbf{P}_q)$;
- в случае неудовлетворительного времени решения задачи, последовательные попытки изменить значения параметров системы и класса системы.

Отметим, что для корректной оценки времени приближенного построения ОДЛА на целевом аппаратно-программном комплексе, в инструментальный вычислительный комплекс требуется включить вычислительный кластер, ГПУ и НСВС. Гетерогенность инструментального вычислительного комплекса позволяет одновременно проводить на нем отладку целевого программного обеспечения.

Для вычислительного кластера методика оценки времени решения задачи на целевой ЭВМ рассмотрена в нашей работе [2]. Методика предполагает получение аналитической зависимости этого времени от параметров вычислительной системы и последующее экспериментальное уточнение указанной зависимости.

Разрабатываемая СППР призвана облегчить лицу, принимающему решения (ЛПР), или группе лиц, принимающих решения (ГПР), решение указанной задачи оптимального проектирования. Основной задачей, которую должна решать эта СППР, является многокритериальная оценка вариантов параллельного аппаратно-программного комплекса. В качестве критериев качества решения при этом могут использоваться вес вычислительной системы, ее габариты, энергопотребление, стоимость и т.д.

5. Оценка точности аппроксимации границ ОДЛА

Для вычисления значений указанных выше критериев оптимальности нужно, прежде всего, оценить точность аппроксимации границы Γ области достижимости границей Γ_c с помощью выбранных методов и алгоритмов. Аналитическая оценка этой точности не удастся. Поэтому предлагается определение этой точности с помощью вычислительного эксперимента на соответствующей вычислительной системе инструментального комплекса.

Рассмотрим прежде частный случай, который определяется следующими соглашениями [4].

1) Имеет место плоское движение ЛА, которое определяется компонентой $u_1(t) = \gamma_c = const$ вектора управления $U(t)$, где γ_c - угол наклона траектории; $\gamma_c \in [0, \pi]$.

2) Известна структура оптимальных управлений $u_2(t), u_3(t)$, приводящих на границы ОДЛА. Точнее говоря, дальняя граница достигается при управлениях

$$u_2 = \begin{cases} \pm 1, & t \leq \tau_f, \\ 0, & t > \tau_f, \end{cases} \quad u_3 = const = 1,$$

ближняя граница - при управлениях

$$u_2 = \begin{cases} \mp 1, & t \leq \tau_n, \\ \pm 1, & t > \tau_n, \end{cases} \quad u_3 = \text{const} = -1,$$

а боковая граница – при управлениях

$$u_2 = \text{const} = \pm 1, \quad u_3 = \begin{cases} \pm 1, & t \leq \tau_s, \\ \mp 1, & t > \tau_s, \end{cases}$$

Здесь $\tau_f \in [0, T]$ – момент времени переключения управления $u_2 = u_2(t)$ при построении дальней границы ОДЛА, $\tau_n \in [0, T]$ – аналогичный момент времени при построении ближней границы, $\tau_s \in [0, T]$ – момент времени переключения управления $u_3 = u_3(t)$ при построении боковой границы ОДЛА.

Положим, что тем или иным способом для управлений U_1, U_2, \dots, U_M получены наборы точек $\{Z_i^f(T)\}$, $\{Z_k^n(T)\}$, $\{Z_l^s(T)\}$, представляющие собой, соответственно, дискретные аппроксимации Γ_d^f , Γ_d^n , Γ_d^s дальней, ближней и боковой границ ОДЛА. И пусть на этой основе построены некоторые непрерывные аппроксимации Γ_c^f , Γ_c^n , Γ_c^s указанных границ.

Схема метода оценки точности аппроксимации, к примеру, границы области достижимости Γ^f границей Γ_c^f имеет следующий вид.

1) В прямоугольнике $[0, T] \times [0, \pi]$ генерируем K псевдослучайных равномерно распределенных точек $(\tau_{F,1}, \gamma_{c,1}), (\tau_{F,2}, \gamma_{c,2}), \dots, (\tau_{F,k}, \gamma_{c,k})$ и определяем соответствующие управления $u_1(t) = \gamma_{c,i}$, $u_2(\tau_{f,i})$, $u_3 = 1$, $i \in [1: K]$.

2) При каждом из указанных управлений интегрируем систему ОДУ (1), находим декартовы координаты точки $Z_i^f(T)$, а затем ее полярные координаты $(\varphi_i^f(T), \chi_i^f(T), \rho_i^f(T))$, где $\varphi_i^f(T)$, $\chi_i^f(T)$, $\rho_i^f(T)$ – азимут, угол места и расстояние от начала координат до точки $Z_i^f(T)$ соответственно.

3) В направлении $(\varphi_i^f(T), \chi_i^f(T))$ определяем расстояние $\tilde{\rho}_i^f(T)$ от начала координат до точки на границе Γ_c^f .

4) Вычисляем величину $\varepsilon_i^f(T) = |\rho_i^f(T) - \tilde{\rho}_i^f(T)|$ – погрешность аппроксимации границы ОДЛА в направлении $(\varphi_i^f(T), \chi_i^f(T))$.

5) В качестве искомой оценки точности аппроксимации принимаем, например, оценку математического ожидания погрешности или оценку максимальной погрешности:

$$\bar{\varepsilon}^f(T) = \frac{1}{K} \sum_{i=1}^K \varepsilon_i^f(T); \quad \varepsilon_{\max}^f(T) = \max_{i \in [1:K]} \varepsilon_i^f(T).$$

6. Пример

Рассмотрим летальный аппарат, движение центра масс которого в земной неподвижной системе координат $0x_r y_r z_r$ описывается системой нелинейных дифференциальных уравнений (3), где v – скорость летательного аппарата; Θ – угол наклона траектории; Ψ – угол поворота траектории; x_r, y_r, z_r – геометрические координаты центра масс летательного аппарата (y_r – высота); $u_1 = \gamma_c$ – скоростной угол крена, $\gamma_c \in [0, \pi]$; $n_T = n_T^{\max} u_2$ – тангенциальная перегрузка, $|n_T| \leq n_T^{\max}$, $|u_2| \leq 1$; $n = n^{\max} u_3$ – нормальная перегрузка, $|n| \leq n^{\max}$, $|u_3| \leq 1$; g – ускорение свободного падения [4].

Движение летательного аппарата рассматривается на интервале времени $t \in [0, T]$. Заданы начальные условия $v(0) = v^0$, $\Theta(0) = \Theta^0$, $\Psi(0) = \Psi^0$, $x_r(0) = x_r^0$, $y_r(0) = y_r^0$, $z_r(0) = z_r^0$.

$$\begin{cases} \dot{v} = g \cdot (n_T^{\max} u_2 - \sin \Theta), \\ \dot{\Theta} = g/v \cdot (n^{\max} u_3 \cos u_1 - \cos \Theta), \\ \dot{\Psi} = -gn^{\max} u_3 \cdot \sin u_1 / v \cdot \cos \Theta, \\ x_r = v \cdot \cos \Theta \cdot \cos \Psi, \\ y_r = v \cdot \sin \Theta, \\ z_r = -v \cdot \cos \Theta \cdot \sin \Psi. \end{cases} \quad (3)$$

Положим, что относительно управлений u_1, u_2, u_3 действуют соглашения, указанные в предыдущем разделе.

Построение области достижимости летательного аппарата, описываемого системой ОДУ (3), выполнено на ГПУ NVIDIA GeForce 9600 GT. В качестве среды разработки системы использована Microsoft Visual Studio 2008 и технология OpenCL. Схему параллельных вычислений покажем на примере построения дальней границы ОДЛА. Покроем прямоугольник $[0, T] \times [0, \pi]$ равномерной сеткой Δ_f с $N_1 \times N_2 = M$ узлами $(\tau_{f,i}, \gamma_{c,j})$, где $i \in [1: N_1], j \in [1: N_2]$. Разобьем узлы этой сетки на $N_{blocs} = N_2 \frac{N_1}{L}$ блоков по L узлов в каждом и поставим в соответствие каждому из блоков поток с идентификатором (k, j) , где $k = iL$. Таким образом, потоку с идентификатором (k, j) соответствуют значения $\tau_f = \tau_{f,l}, \tau_{f,(l+1)}, \dots, \tau_{f,(l+L)}$ и значение $\gamma_c = \gamma_{c,j}$. Здесь $l = kL + 1$.

На рисунке 2 в качестве примера изображена область достижимости летательного аппарата (3) при двух стационарных сферических препятствиях Ω_1, Ω_2 , построенная при начальных условиях $Z=1000$ м, $X=0, Y=0, V=330$ м/с, $\Theta=0, \Psi=0$ и $T=10$ с. Соответствующие области запрещенных управлений (управлений, которые порождают траектории летательного аппарата, пересекающие хотя бы одно из препятствий) приведены на рисунке 3.

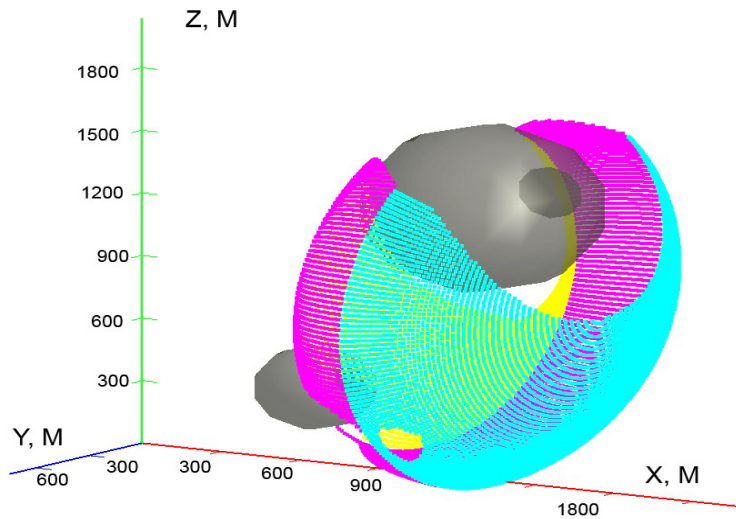


Рис. 2. Область достижимости летательного аппарата при двух стационарных сферических препятствиях

Для ситуации, когда в области достижимости летательного аппарата (3) имеется одно стационарное сферическое препятствие, выполнено исследование производительности параллельных вычислений. Некоторые результаты исследований представлены на рисунке 4, который показывает, что при большом числе узлов сетки Δ_f ускорение вычислений достигает пяти (по сравнению с вычислениями на центральном процессоре системы).

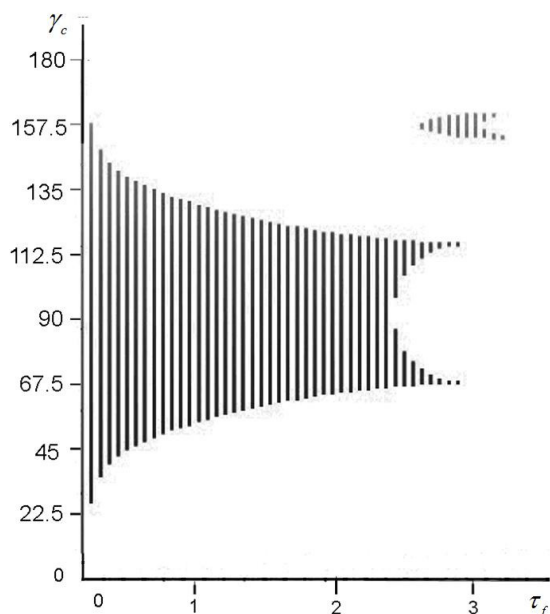


Рис. 3. Области запрещенных управлений для задачи с двумя стационарными сферическими препятствиями

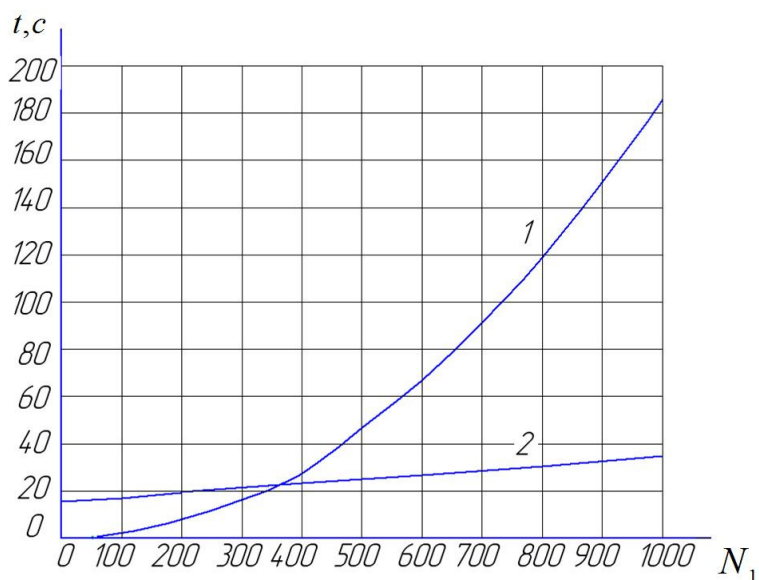


Рис. 4. Время построения дальней границы ОДЛА в функции числа узлов N_1 при $N_2 = 256$:
1 – GPU; 2 - CPU

Заключение

В работе изложены основные принципы построения системы поддержки принятия решений, предназначенной для оптимального проектирования аппаратно-программного комплекса, обеспечивающего параллельное построение границ области достижимости летательного аппарата с заданной точностью и за заданное время.

В качестве целевой вычислительной системы предполагается использование кластерной системы, системы на основе графических процессорных устройствах или системы на основе и нейросетевых ускорителей. Инструментальная вычислительная система полагается гетерогенной, включающей в себя указанные типы параллельных вычислительных систем.

Рассмотрена иерархическая совокупность численных методов приближенного построения границ ОДЛА, включающая в себя различные методы аппроксимации векторного поля

модельной системы обыкновенных дифференциальных уравнений, методы построения дискретной аппроксимации границы ОДЛА, а также методы построения непрерывной аппроксимации этой границы.

В настоящее время ведутся работы по реализации рассматриваемой системы поддержки принятия решений.

Авторы благодарят д.т.н., проф. МГТУ им. Н.Э.Баумана Воронова Е.М. и к.т.н. Карпунина А.А. за постановку задачи приближенного построения границ области достижимости летательного аппарата и плодотворные многократные обсуждения возможных подходов к ее решению.

Литература

1. Воронов Е.М., Карпенко А.П., Козлова О.Г., Федин В.А. Численные методы построения области достижимости динамической системы // Вестник МГТУ. Сер. "Приборостроение". 2010. №2(79). С. 3-19.
2. Воронов Е. М., Карпенко А. П., Федин В. А. Параллельное построение множества достижимости высокоманевренного летательного аппарата методом "мультифиниша" // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции / ЮУрГУ. Челябинск: ЮУрГУ, 2010. С. 113-120.
3. Витюков Ф.А., Домашнев В.К., Карпенко А.П., Федин В.А. Построение области достижимости динамической системы на NVIDIA и AMD графических процессорах // Научный сервис в сети Интернет: суперкомпьютерные центры и задачи 2010: Труды международной суперкомпьютерной конференции / МГУ. Москва: МГУ, 2010. С. 635-641.
4. Воронов, 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.
5. Саймон Хайкин. Нейронные сети: полный курс. 2-е издание. Вильямс, 2006. С. 281-286.
6. Хохлов, С.Ф., Школа О.И. Применение метода Брандона для обработки экспериментальных данных // Вопросы химии и химической технологий. 1973. Вып. 28. С. 204-207.
7. Козлова О.Г. Нейросетевая аппроксимация границы области достижимости летательного аппарата. 1. Двухмерный случай // Наука и образование: электронное научно-техническое издание. 2009. №7.
URL <http://technomag.edu.ru/doc/129990.html> (дата обращения: 11.12.2010).
8. Козлова О.Г. Нейросетевая аппроксимация границы области достижимости летательного аппарата. 2. Трехмерный случай // Наука и образование: электронное научно-техническое издание. 2009. №8.
URL <http://technomag.edu.ru/doc/130282.html> (дата обращения: 11.12.2010).