

Быстров А.В., Карпенко А.П., Козлова О.Г., Федин В.А.  
МГТУ им. Н.Э.Баумана

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

# Постановка задачи определения области достижимости динамической системы (ОДЛА)

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

$X = X(t)$  –  $n$ -мерный вектор фазовых переменных

$U = U(t)$  –  $m$ -мерный вектор управлений

$F(t, X, U)$  –  $n$ -мерная вектор-функция

$X^0$  –  $n$ -мерный вектор начальных условий

$X \in D_X$  – множество допустимых значений вектора фазовых переменных

$U \in D_U$  – множество допустимых управлений

Среди компонентов  $X = (x_1, x_2, \dots, x_n)$  выделим  $v \leq n$  переменных, образующих  $v$ -мерный вектор  $Y = (x_1, x_2, \dots, x_v)$

Область достижимости в момент времени  $T$

$$G(X^0, T) = \{ \tilde{Y}(T) / U(t) \in D_U, X(t) \in D_X \}$$

# Математическая модель динамики летательного аппарата 6-го порядка

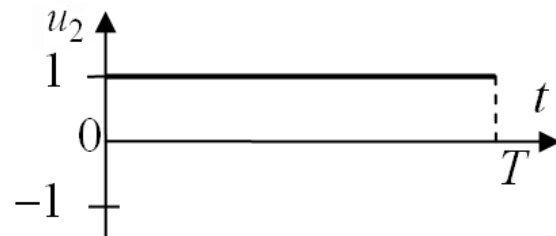
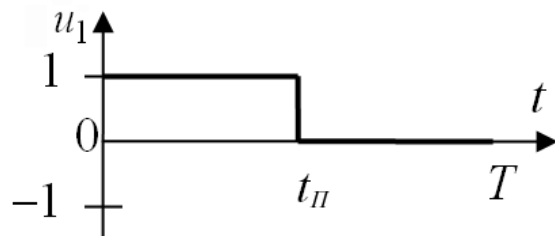
$$\left\{ \begin{array}{l} \dot{V} = g \cdot (n_T - \sin \Theta) \\ \dot{\Theta} = \frac{g}{V} \cdot (n \cdot \cos \gamma_c - \cos \Theta) \\ \dot{\Psi} = \frac{-g \cdot n \cdot \sin \gamma_c}{V \cdot \cos \Theta} \\ \dot{X} = V \cdot \cos \Theta \cdot \cos \Psi \\ \dot{Y} = V \cdot \sin \Theta \\ \dot{Z} = -V \cdot \cos \Theta \cdot \sin \Psi \end{array} \right.$$

$V$  – скорость летательного аппарата  
 $\Theta$  – угол наклона траектории  
 $\Psi$  – угол поворота траектории  
 $Y$  – высота летательного аппарата  
 $n_T$  – тангенциальная перегрузка  
 $n$  – нормальная перегрузка  
 $\gamma_c$  – скоростной угол крена  
 $g$  – ускорение свободного падения

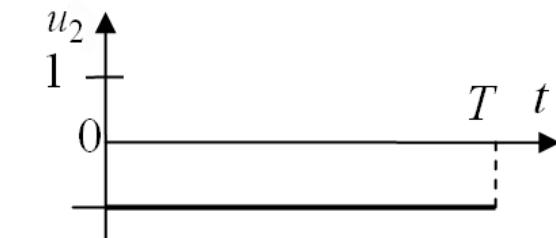
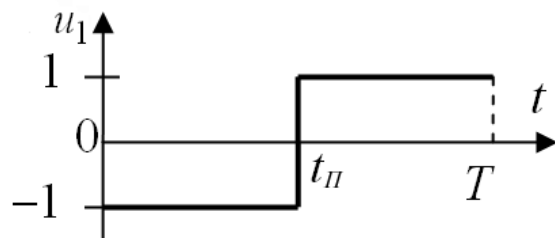
$\mathbf{U} = (n_T, n, \gamma_c)$  – вектор управления

# Структуры граничных управлений при фиксированном угле крена

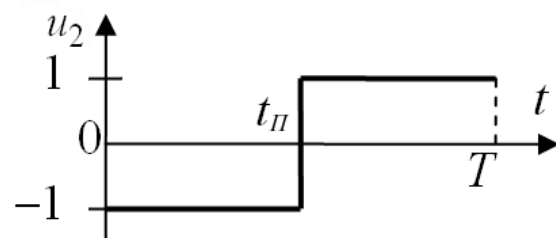
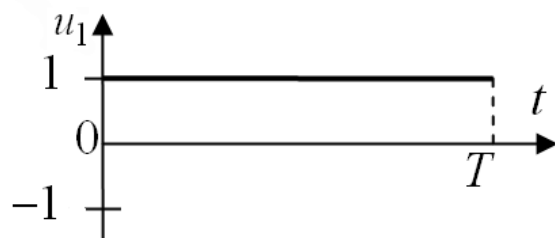
а) для дальней границы



б) для ближней границы

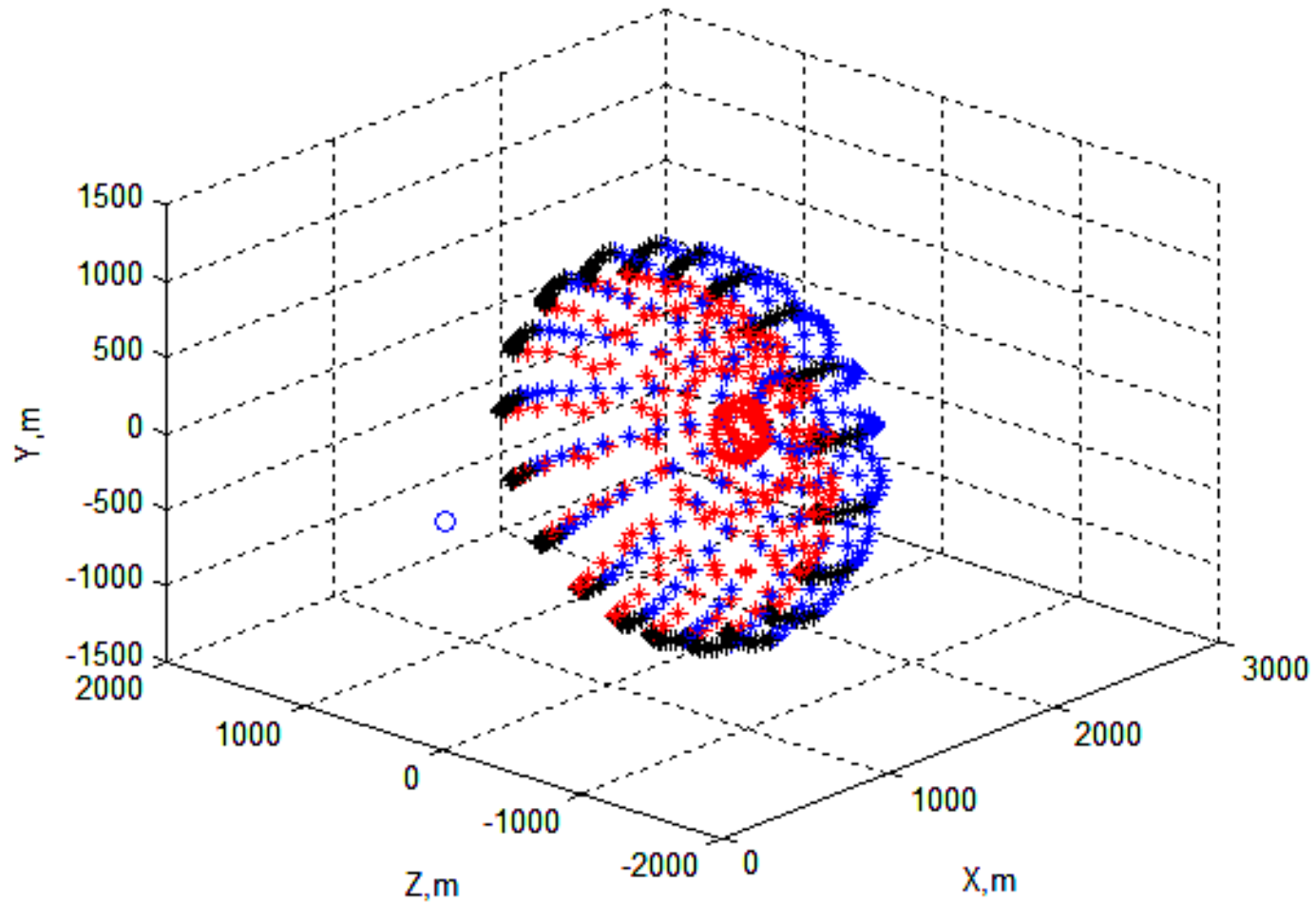


в) для боковой границы



$$n^{\max} \cdot u_1 = n, \quad n_T^{\max} \cdot u_2 = n_T, \quad |u_1| \leq 1, \quad |u_2| \leq 1, \quad \gamma = const$$

# Вид общей границы области достижимости ЛА



# Схема метода построения ОДЛА

- 1)  $\{U_i | U_i \in D_U, i \in [0, M]\}$  – некоторая сетка, на основе которой генерируется совокупность систем ОДУ:

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

- 2)  $\{Y_i(T)\}$  – дискретная аппроксимация области достижимости  $G$
- 3)  $\{Z_i(T)\}$  – дискретная аппроксимация  $\Gamma_d$  границы  $\Gamma$  области достижимости
- 4)  $\Gamma_c$  – непрерывная аппроксимация границы  $\Gamma$  области достижимости

# Варианты методов построения ОДЛА

## Алгоритмы:

- без аппроксимации правых частей,
- с использованием аппроксимация полиномами,
- с использованием нейросетевой аппроксимации.

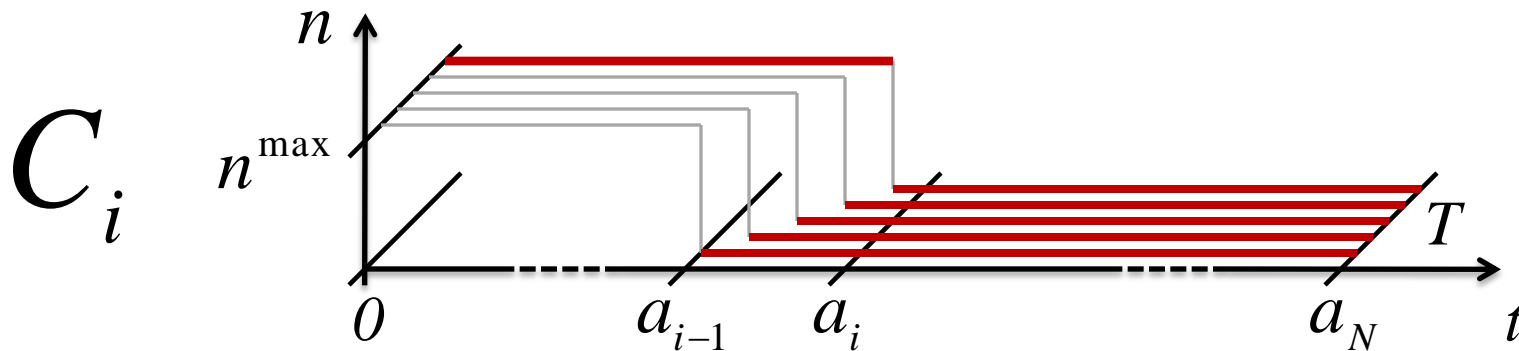
## Общие параметры алгоритмов:

- метод интегрирования,
- шаг интегрирования (требуемая точность интегрирования),
- шаг дискретизации времени управления.

## Способ оптимизации отображения на архитектуру:

- равномерная декомпозиция точек переключения (для SIMD систем),
- неравномерная декомпозиция точек переключения (для MIMD систем).

# Схема распараллеливания для MIMD систем



$$P_i : [a_{i-1} : a_i], i = 1, 2, \dots, N$$

$$a_0 = 0$$

$$a_N = M$$

$$C_i = a_i + \sum_{j=a_{i-1}}^{a_i} (M - j)$$

$M$  – число точек переключения

$N$  – число процессоров

$P_i$  –  $i$ -й процессор,  $i=1, 2, \dots, N$

$a_{i-1}$  – индекс первой точки переключения на  $i$ -м процессоре

$C_i$  – объем вычислений на  $i$ -м процессоре



## Балансировка загрузки при $M \gg N$

$$C_i = a_i + \sum_{j=a_{i-1}}^{a_i} (M - j) \approx a_i + (M - \frac{(a_i + a_{i-1})}{2}) \cdot (a_i - a_{i-1})$$

Система уравнений:  $N+1$  уравнений,  $N+1$  неизвестных:  $a_i, (i = 0, 1, \dots, N)$

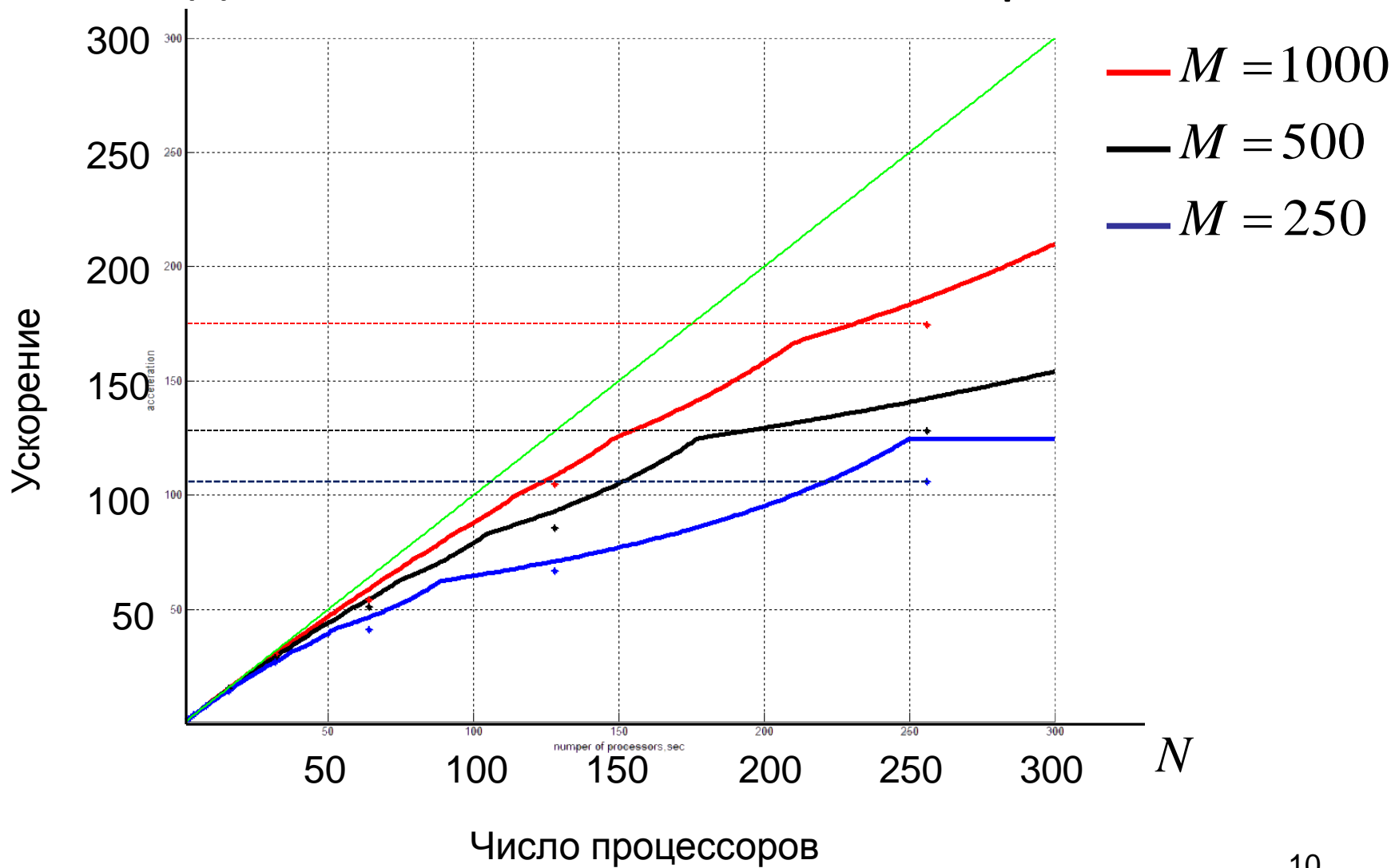
$$\begin{cases} a_i + \frac{2 \cdot M - (a_i + a_{i-1})}{2} \cdot (a_i - a_{i-1}) = \\ = a_{i+1} + \frac{2 \cdot M - (a_{i+1} + a_i)}{2} \cdot (a_{i+1} - a_i); i = 1, 2, \dots, N - 1 \\ a_0 = 0 \\ a_N = M \end{cases}$$

## Балансировка загрузки при $M \sim N$

$$C_i = a_i + \sum_{j=a_{i-1}}^{a_i} (M - j)$$

Дискретная задача оптимизации:  $a_1, \dots, a_{N-1} : \max_{i=1, \dots, N} C_i \rightarrow \min$

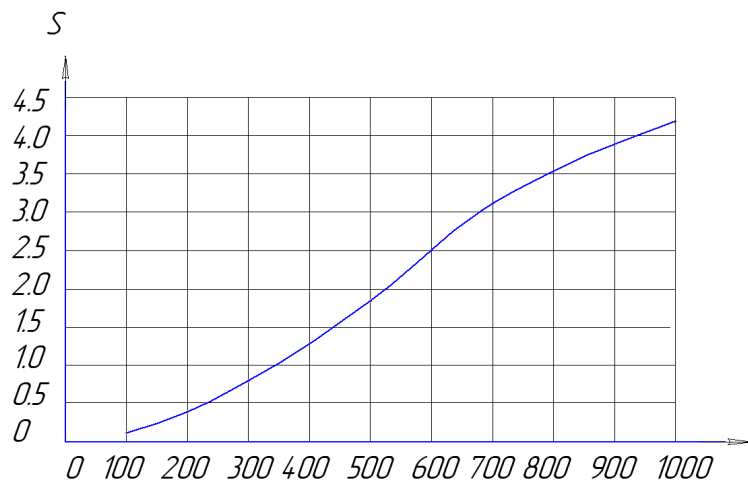
# Некоторые результаты для вычислительных кластеров



# Схема построения ОДЛА на ГПУ

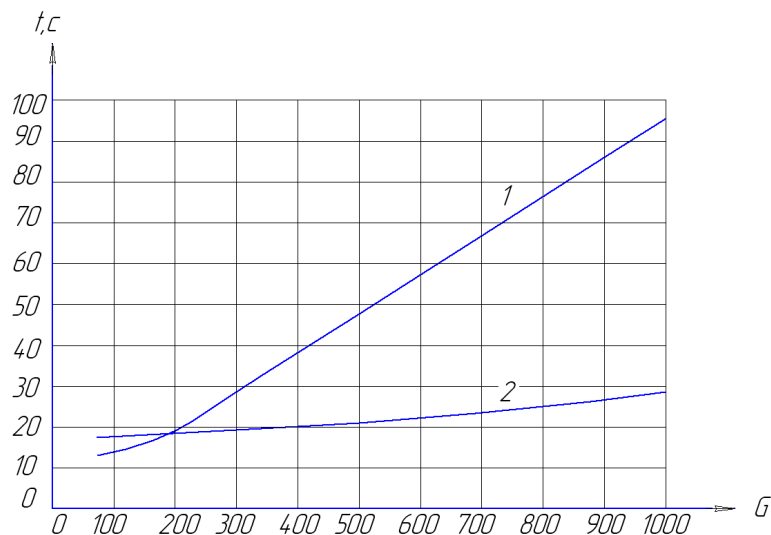
- Покроем прямоугольник  $[0, T] \times [0, \pi]$  равномерной сеткой  $\Delta_f$  с  $N_1 \times N_2 = M$  узлами  $(t_{fi}, \gamma_{cj})$ , где  $i \in [1: N_1]$   $j \in [1: N_2]$
- Разобьем узлы сетки на  $N_{blocs} = N_2 \frac{N_1}{L}$  блоков по  $L$  узлов
- Поставим в соответствие блоку  $(k, j)$  набор из управлений  $[(t_{f[L*k+1]}, \gamma_{cj}), (t_{f[L*k+2]}, \gamma_{cj}), \dots, (t_{f[L*(k+1)]}, \gamma_{cj})]$

# Некоторые результаты построения ОДЛА на ГПУ



Пример результатов ускорения вычислений для модели динамики ЛА 6-го порядка

График зависимости ускорения вычислений от числа точек переключения управлений.



Зависимость времени выполнения программы от числа точек переключения управлений.

1 - для вычислений на CPU,  
2 - на GPU.

# Построение границы ОДЛА с аппроксимацией правых частей ДУ полиномами

- 1) Сетка  $\Omega = \{t_a, x_{i,b}, u_{k,c} \mid t_a \in [0, T], x_{i,b} \in D_X, u_{k,c} \in D_U\};$   
 $i \in [1, n], k \in [1, m], a \in [1, N_t], b \in [1, N_X], c \in [1, N_U].$

$N_t \cdot N_X^n \cdot N_U^m$  – общее число узлов сетки, пропорционально  
требуемому объему памяти

- 2)  $F(t_a, x_{i,b}, u_{k,c})$  – значения функции  $F(t, X, U)$  в узлах сетки  $\Omega$
- 3)  $\dot{X} = \tilde{F}(t, X, U)$ , где  $\tilde{F}$  – интерполяция или аппроксимация  
значений  $F(t_a, x_{i,b}, u_{k,c})$

# Виды аппроксимаций правых частей ДУ полиномами

1. Линейная интерполяция. Базисные функции:

$$f_0=1, f_1=x.$$

2. Полиномиальная аппроксимация. Базисные функции:

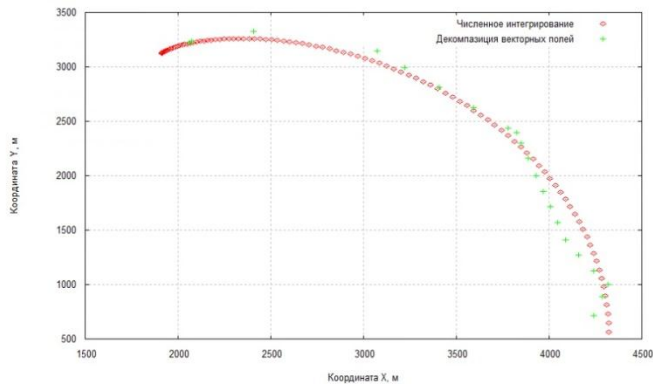
- $f_i = x^i$ .
- Полиномы Чебышева.
- Полиномы вида:

$$F_i = \sum_{j_1=0}^l ((x^1)^{j_1} \times (\dots \times (\sum_{j_n=0}^l (x^n)^{j_n} \times (\sum_{j_{n+1}=0}^l (u^1)^{j_{n+1}} \times$$

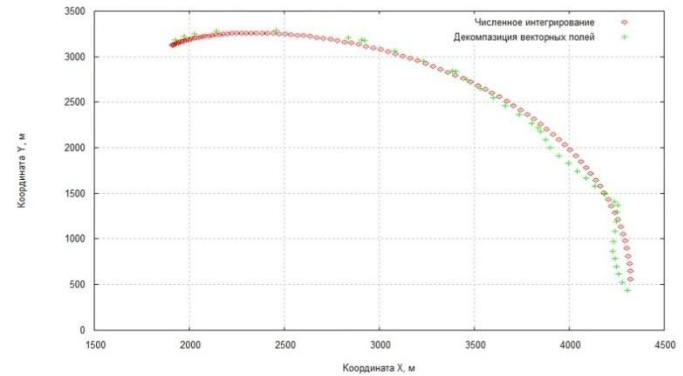
$$\times (\dots \times (\sum_{j_{n+k}=0}^l (\varphi_{j_1 \dots j_n j_{n+1} \dots j_{n+k}} \times (u^k)^{j_{n+k}})))))))).$$

# Примеры построения дальней границы ЛА с использованием аппроксимации правых частей

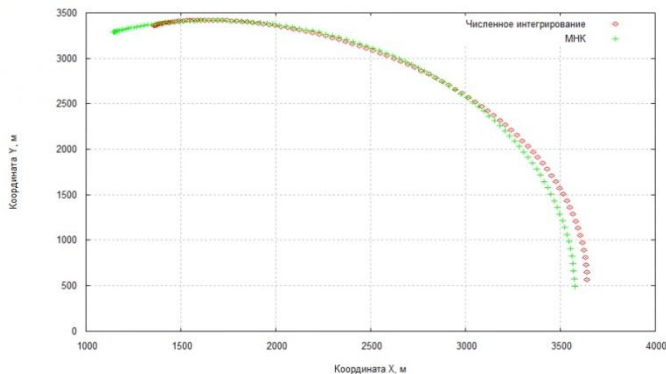
## Кусочно-постоянная интерполяция



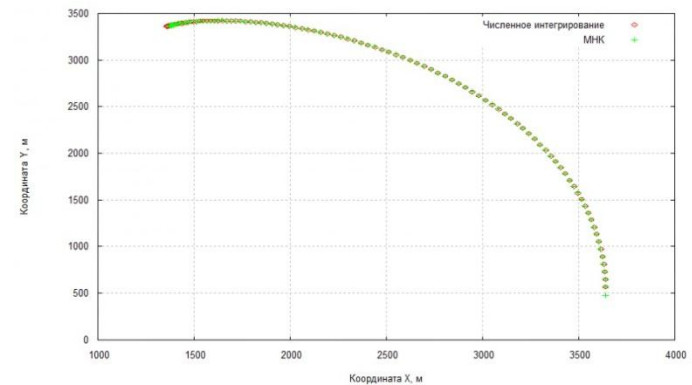
## Кусочно-линейная интерполяция



## Аппроксимация полиномами 2-го порядка



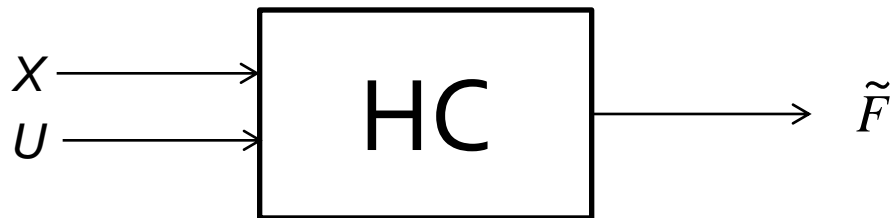
## Аппроксимация полиномами 3-го порядка



# Построение границы ОДЛА с аппроксимацией правых частей ДУ нейронными сетями

1) Из множества  $D_X \times D_U$  генерируется случайная выборка пар вида  $(x_i, u_i)$ ,  $i \in [1, N]$  и вычисляются соответствующие значения функции  $F_i = F(x_i, u_i)$ .

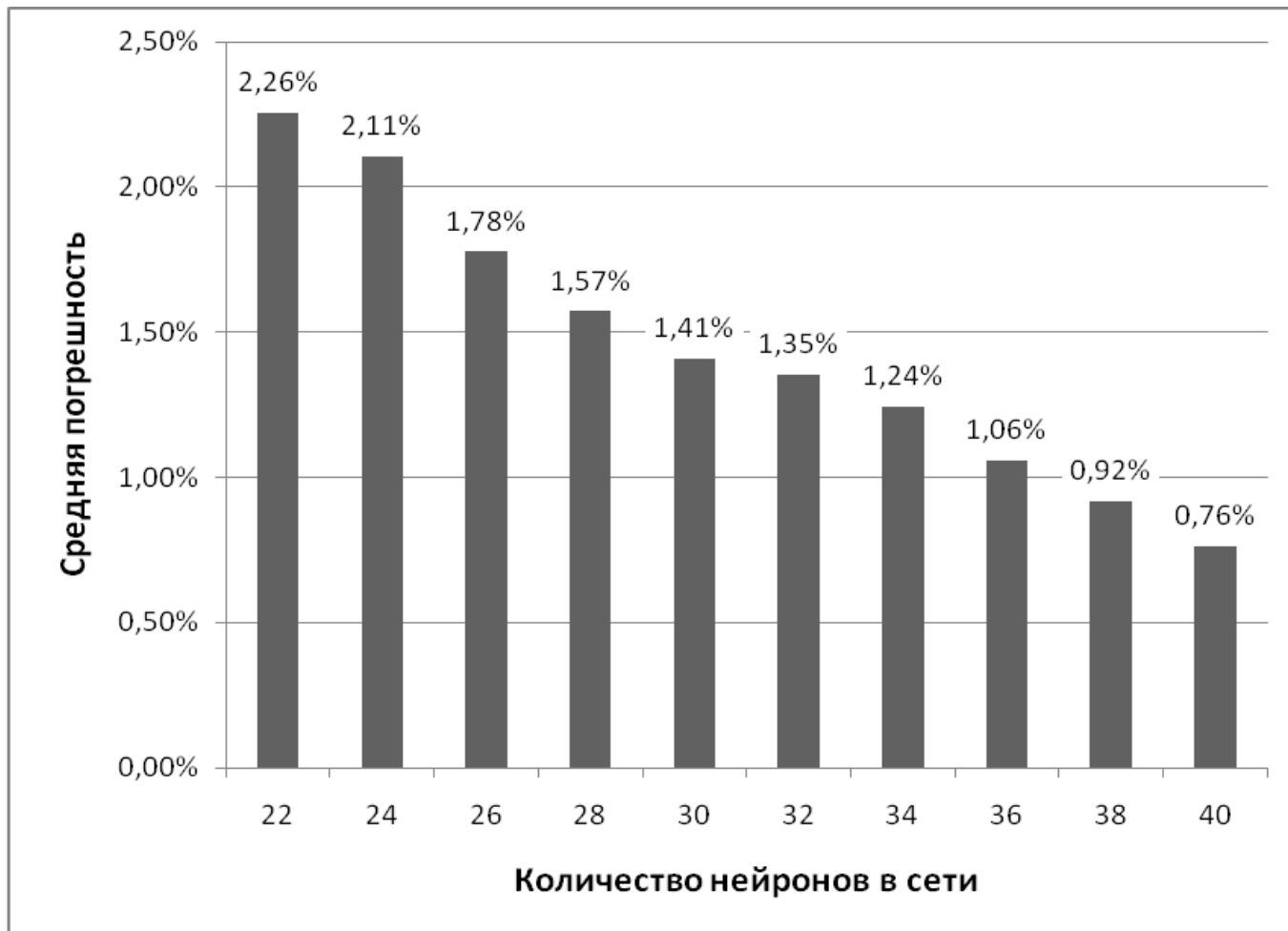
2) На основе этой выборки обучается нейронная сеть вида:



3) При интегрировании динамической системы  $\dot{X}(t) = F(X, U)$  вектор-функция  $F(X, U)$  заменяется своим приближением  $\tilde{F}(X, U)$ .

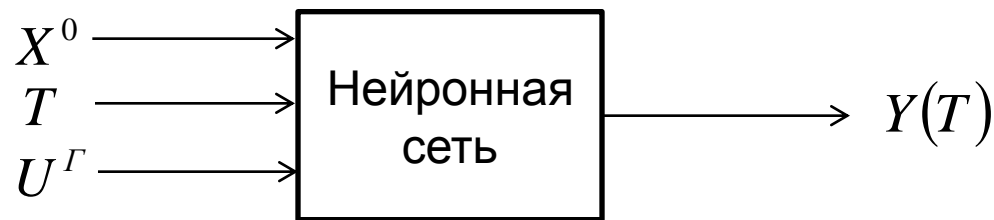


# Погрешности построения границы области достижимости с аппроксимацией правых частей ДУ нейронными сетями

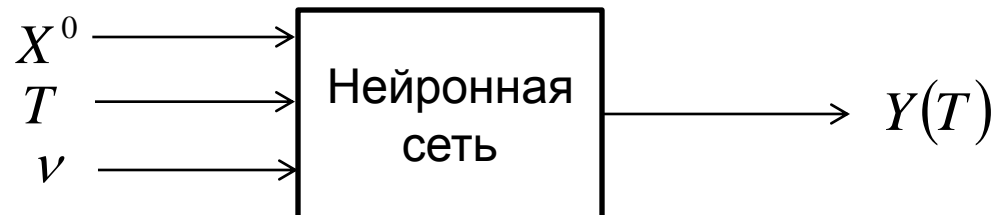


# Методы нейросетевой аппроксимации границ области достижимости

- 1) Определение точки на границе ОД по заданному оптимальному управлению:



- 2) Определение точки на границе ОД по заданному направлению:



# Система поддержки принятия решений, основные понятия

$\mathbf{M}_i = \{M_{i,j}\}$  метод построения ОД динамической системы

$\mathbf{A}_{i,j} = \{A_{i,j,k}(\mathbf{P}_{i,j,k})\}$  алгоритмы, реализующие метод  $M_{i,j}$

$\mathbf{P}_{i,j,k}$  вектор свободных параметров алгоритма  $A_{i,j,k}$

$\mathbf{C}_q = \{C_{q,r}(\mathbf{P}_{q,r})\}$  класс вычислительных систем

$\mathbf{P}_{q,r}$  вектор свободных параметров вычислительной системы  $C_{q,r}$

$\mathbf{E}(A_{i,j,k}(\mathbf{P}_{i,j,k}), C_{q,r}(\mathbf{P}_{q,r})) = \{E_m(A_{i,j,k}(\mathbf{P}_{i,j,k}), C_{q,r}(\mathbf{P}_{q,r}))\}$

множество вариантов отображения алгоритма  $A_{i,j,k}(\mathbf{P}_{i,j,k})$

на архитектуру системы  $C_{q,r}(\mathbf{P}_{q,r})$

# Использующиеся в системе зависимости, полученные экспериментально

- Построение ОДЛА интегрированием

- Без аппроксимации

**точность / число точек переключения управления**

- На вычислительном кластере

**скорость / число процессоров / число точек переключения**

- На графическом процессоре

**скорость / число потоков / число точек переключения**

- С аппроксимацией полиномами

**точность / плотность используемой сетки для каждого типа аппроксимации**

**скорость / число точек переключения для каждого типа аппроксимации**

- С аппроксимацией нейронными сетями

**точность / число нейронов для каждого типа НС**

**скорость / число точек переключения для каждого типа НС**

- Для построения ОДЛА нейронными сетями:

**точность / число нейронов**

# Система поддержки принятия решений, краткое описание работы

Вход:

- динамическая система  $\dot{X}(t) = F(X, U)$
- требования к точности и быстродействию
- предпочтительный класс  $C_q$

Процесс:

- выбор  $C_{q,r}(\mathbf{P}_{q,r})$   $A_{i,j,k}(\mathbf{P}_{i,j,k})$
- проверка точности и скорости
- варьирование по  $j, k, r$

Результат работы :

получение набора  $E^*(A_{i,j^*,k^*}(\mathbf{P}_{i,j^*,k^*}), C_{q,r^*}(\mathbf{P}_{q,r^*}))$

позволяющего строить область достижимости с нужной точностью за требуемое время или указание на невыполнимость поставленных требований к системе.

# Результаты

- Разработаны методы приближенного параллельного построения области достижимости высокоманевренного летательного аппарата
- Сформирована структура системы поддержки принятия решений при создании аппаратно-программного комплекса для быстрого построения ОДЛА
- Проведены серии компьютерных экспериментов, подтверждающие работоспособность разработанных методов