

**Моделирование движения динамических объектов
по достижению целей в 3D пространстве**

В.В. Афонин

Национальный исследовательский

Мордовский государственный университет им. Н.П. Огарёва,

430005, г. Саранск, ул. Большевистская, д. 68, Россия

Резюме. Цель. В работе рассматривается задача моделирования движения динамического объекта в заданные точки 3D пространства, включая цели на какой-либо функциональной траектории некоторой внешней функции $f(x)$. Решение задачи заключается в расчете управления, управляющего воздействия, входящего в математическое описание обыкновенных дифференциальных уравнений, заставляющего решение дифференциальных уравнений достигнуть заданных целей в 3D пространстве. **Метод.** Используются численные методы решения обыкновенных стационарных дифференциальных уравнений. В частности, метод Рунге-Кутты 4-го порядка, который входит в состав ряда сред и языков программирования, включая Python. Рассматриваются способы и приемы формирования трехмерного пространства и вывода динамического объекта в заданные или произвольные его точки. **Результат.** На модельных примерах линейных и нелинейных динамических объектах показывается возможность вывода их из начальной точки вектора состояния в достаточно произвольные точки или цели 3D пространства. При этом модельные объекты могут быть асимптотически устойчивыми или нет. **Вывод.** Предлагаемые методы и приемы дают возможность перевода динамических объектов управления в заданные точки 3D пространства. В ряде случаев вывода динамических объектов в заданные цели 3D пространства возникает необходимость изменения параметров моделей динамических объектов, как линейных, так и нелинейных.

Ключевые слова: стационарные динамические объекты, внешняя функция $f(x)$, цели в 3D пространстве, индекс вектора состояния $X(t)$, управление $U(t)$, Python

Для цитирования: В.В. Афонин. Моделирование движения динамических объектов по достижению целей в 3D пространстве. Вестник Дагестанского государственного технического университета. Технические науки. 2026;53(2):36-51. DOI:10.21822/2073-6185-2026-53-2-36-51

Simulation of the movement of dynamic objects to achieve goals in 3D space

V.V. Afonin

N.P. Ogarev National Research Mordovia State University,

68/1 Bolshevistskaya St., Saransk 430005, Russia

Abstract. Objective. The paper considers the problem of modeling the movement of a dynamic object to given points in 3D space, including targets on some functional trajectory of some external function $f(x)$. The solution to the problem lies in calculating the control, the control action included in the mathematical description of ordinary differential equations, forcing the solution of differential equations to achieve the specified goals in 3D space. **Method.** Numerical methods for solving ordinary stationary differential equations are used. In particular, the fourth-order Runge-Kutta method, which is included in a number of programming environments and languages, including Python. The article examines methods and techniques for forming a three-dimensional space and displaying a dynamic object at specified or arbitrary points. **Result.** Using model examples of linear and nonlinear dynamic objects, the possibility of deriving them from the

initial point of the state vector to fairly arbitrary points or targets in 3D space is demonstrated. Model objects may be asymptotically stable or not. **Conclusion.** Methods and techniques make it possible to transfer dynamic control objects to specified points in 3D space. In a number of cases of outputting dynamic objects to specified targets in 3D space, it becomes necessary to change the parameters of dynamic object models, both linear and nonlinear.

Keywords: stationary dynamic objects, external function $f(x)$, goals in 3D space, state vector index $X(t)$, control $U(t)$, Python.

For citation: V.V. Afonin. Simulation of the movement of dynamic objects to achieve goals in 3D space. Herald of Daghestan State Technical University. Technical Sciences. 2026;53(2):36-51. (In Russ) DOI:10.21822/2073-6185-2026-53-2-36-51

Введение. Задачи моделирования различных процессов встречается во многих отраслях науки и техники [1-4]. Как правило, в теории автоматического управления рассматриваются модели объектов по характеру дифференциальных уравнений, описывающих динамические объекты управления [5-7]. Во многих практических случаях для таких объектов ставится задача их перевода из некоторого начального состояния в заданную точку или в некоторую точку цели внешней функциональной кривой. Например, внешние траектории могут отображать движение летательных и беспилотных летательных аппаратов [8, 9], а также роботов [10, 11]. Перевод объекта управления из одного состояния в некоторое другое может быть рассмотрен также с помощью методов теории оптимального управления [12-14], когда на управляющие воздействия накладываются ограничения.

Далее рассматриваются динамические объекты, математические модели которых представляются в виде обыкновенных стационарных дифференциальных уравнений [15, 16]. В работе переменные состояния и управляющие воздействия принимаются без каких-либо ограничений. На модельной внешней непрерывной функции $f(x)$ фиксируются маркеры цели, к которым необходимо обеспечить движение динамического объекта [17]. В данном случае будет предполагаться, что движения осуществляется в прямоугольной декартовой системе координат трехмерного пространства.

Постановка задачи. В качестве примеров динамических объектов рассматриваются как известные из библиографических источников [13, 14], так и некоторые модельные объекты, математическое описание которых задаются обыкновенными дифференциальными уравнениями с постоянными коэффициентами. Первые два примера взяты из источника [13, с. 102, с. 143], в которых рассматриваются нелинейные динамические объекты управления 3-го и 2-го порядков. Математическое описание объекта первого примера из [13, с. 102] имеет следующий вид:

$$\frac{dx_1(t)}{dt} = b_1 u(t) - a_{11} x_1(t); \quad \frac{dx_2(t)}{dt} = a_{21} x_1; \quad \frac{dx_3(t)}{dt} = a_{32} (x_2(t) - k x_2^2(t)) - x_3(t), \quad (1)$$

где постоянные коэффициенты $b_1, a_{11}, a_{21}, a_{32}, k$ принимаются равные единице.

В соответствии с [13] требуется найти такое управление в (1), чтобы за минимальное время перевести объект (1) из начального вектора состояния в точку, принадлежащую множеству стационарных состояний, которое, как следует из (1) определяется уравнением

$$x_3 = x_2 - x_2^2. \quad (2)$$

Далее, будет показан результат без претензии на управление по быстродействию, но выполняющий задачу перевода объекта вида (1) на кривую (2), которая будет представлена в 3D пространстве. Следующий пример из [13, с. 143] дан с математическим описанием двух параллельно соединённых нелинейных звеньев:

$$\frac{dx_1(t)}{dt} = b_1 u(t) - k_1 x_1(t) + k_2 x_1^2(t); \quad \frac{dx_2(t)}{dt} = b_2 u(t) - k_3 x_2(t) + k_4 x_2^2(t), \quad (3)$$

где постоянные коэффициенты: $b_1 = b_2 = 1$; $k_1 = k_3 = 1$; $k_2 = k_4 = 1$, $u(t)$ – управление, подлежащее определению.

Как отмечается в [13], объект (3) описывает процессы с рециклом. Здесь для него ставится задача перевода из ненулевого начального состояния в точки стационарной кривой, уравнение которой задано в неявном виде:

$$x_1^2 - x_1 + x_2 - x_2^2 = 0. \quad (4)$$

Следующий модельный объект основан на примере из [18], но здесь он описывается системой нелинейных дифференциальных уравнений 4-го порядка:

$$\begin{aligned} \frac{dx_1(t)}{dt} &= \sin(x_2(t)) + u(t); \quad \frac{dx_2(t)}{dt} = -x_1(t)x_2(t)x_3(t)x_4(t); \\ \frac{dx_3(t)}{dt} &= -x_1(t) - x_2(t)x_3(t)\text{sign}(x_1(t)); \quad \frac{dx_4(t)}{dt} = x_1(t). \end{aligned} \quad (5)$$

Для нелинейного динамического объекта (5) ставится задача его перевода из нулевого начального состояния в некоторые точки достаточно произвольных функциональных кривых $f(x)$. Характер систем (1), (3), (5) относится к классу нелинейных динамических систем линейных по управлению, которые называются аффинными системами [19].

В рассмотрение введём и линейные стационарные динамические объекты вида

$$\frac{dX(t)}{dt} = AX(t) + BU(t), \quad (6)$$

где $X(t)$ – n -мерный вектор состояния, $U(t)$ – m -мерный вектор управления, A – постоянная матрица состояния размера $n \times n$, B – постоянная матрица входа размера $n \times m$. Сначала зададим два модельных объекта.

Первый динамический объект выражается следующими матрицами A и B :

$$A = \begin{bmatrix} 0,44 & 0,89 & 0,22 & 0,17 & 0,13 & 0,11 \\ 0,08 & 0,16 & 0,79 & 0,07 & 0,05 & 0,04 \\ 0,03 & 0,06 & 0,09 & 0,81 & 0,03 & 0,03 \\ 0,01 & 0,03 & 0,04 & 0,06 & 0,84 & 0,02 \\ 384,89 & 700,43 & 856,20 & 841,90 & 738,63 & 627,71 \\ -461,84 & -840,47 & -1027,37 & -1010,20 & -886,24 & -752,15 \end{bmatrix}; \quad A \rightarrow 6 \times 6;$$

$$B = \begin{bmatrix} 2 & 6 & 1 \\ -3 & 5 & -3 \\ 3 & -2 & 1 \\ 6 & -3 & 7 \\ 2 & 5 & 2 \\ 6 & 0 & -1 \end{bmatrix}; \quad B \rightarrow 6 \times 3. \quad (7)$$

Линейный объект с матрицами (7) является изначально асимптотически устойчивым и полностью управляемым по алгебраическому критерию Калмана. Следующий объект типа (6) имеет матрицы

$$A = \begin{bmatrix} 10,1 & 8,7 & 72,01 \\ -21,21 & 6,4 & -25 \\ 1,9 & 5,3 & 9,9 \end{bmatrix}; \quad B = \begin{bmatrix} -19,53 & -3,3 \\ 35,5 & -10,213 \\ 87,7 & 72,53 \end{bmatrix}. \quad (8)$$

Объект (8) асимптотически неустойчив и не полностью управляем по критерию Калмана. Алгебраический критерий управляемости по Калману можно найти в [5, 6, 7].

К объектам с описанием (6) относятся элементарные звенья стационарных систем управления [6, 7]. Например, в [20, с. 46] рассматриваются колебания систем со многими

степенями свободы. В некоторой степени по аналогии с примерами из [20], будет рассмотрена задача синтеза управления для четырёх устойчивых и неустойчивых колебательных звеньев, соединённых параллельно. Математическое описание таких звеньев хорошо известно, поэтому здесь не приводится.

Тестовые модели внешних функциональных кривых $f(x)$ примем следующими:

$$f(x) = 5,3|\cos(0,15x) - 0,15|x^{-0,03x} + e^{(-0,0535x)} + 1,13; x \in [0; 245,5]; \quad (9)$$

$$f(x) = \frac{5,3\sin(x)}{x}; x \in [-25; 25,5]. \quad (10)$$

Первая задача будет заключаться в представлении (9), (10) в трёхмерном пространстве, например с помощью модуля `matplotlib` Python. Графические возможности модуля `matplotlib` и языка программирования Python достаточно обширны [21-23].

Методы исследования. Опираясь на [17], где решение дифференциальных уравнений выполнялось в символьном виде, в данном исследовании предлагается перейти к традиционным численным методам интегрирования обыкновенных стационарных дифференциальных уравнений. В частности методы Рунге-Кутты 4-го порядка, которые входят в библиотеку `scipy (odeint)` Python, `MATLAB (ode45)`, `C#` с DLL-библиотекой `MathNet.Numerics.dll (RungeKutta.FourthOrder)`. Они были выбраны, исходя из их свойств, изложенных, например в [15, 16]. Предполагается, что тестовые дифференциальные уравнения не являются жёсткими (*non-stiff differential equations*).

Для представления внешней функции $f(x)$ движения какого-то объекта в трёхмерном пространстве принято, что координатная сетка плоскости $ХОУ$ квадратная, исходя из диапазона изменения аргумента $f(x)$. Соответственно, числовые значения по оси аппликат будут соответствовать значениям самой функции $f(x)$ или стационарных кривых (2), (4), (9), (10). Например, для (10) пусть задан список или массив x , список $f(x) - z$, тогда, следуя [22, 23]: приведём строчку кода на Python:

```
plt.figure(figsize = (6, 4), dpi=130)
ax = plt.axes(projection = '3d')
ax.plot3D(x, x, z, 'k') # или plt.plot(x, x, z, 'black')
ax.view_init(10, -45)
# 10 – eval (возвышение), -45 – azim (градус поворота вокруг оси аппликат).
```

Конечно, должен быть подключен модуль `matplotlib`, позволяющий дополнить цвет, толщину и стиль линии графика, а также шрифты подписей на диаграмме. Сетка $ХОУ$ может быть и прямоугольной.

Внешняя функция может быть построена сравнительно для небольшого количества точек. В тоже время, для численного интегрирования с помощью `odeint` требуется указывать количество итераций `Node`, число которых может отличаться от числа точек для внешней функции, поэтому величина `Node` задаётся отдельно, поскольку она входит, например, в

$$\text{хopti} = \text{np.linspace}(0, \text{dina}, \text{Node}, \text{dtype} = \text{float}, \text{endpoint}=\text{True}) \quad (11)$$

В выражении кода (11) целочисленное значение `Node` задаётся пользователем. Префикс `np` – от названия модуля `numpy` Python. Массив (11) задаёт направление интегрирования дифференциальных уравнений – интервал численного интегрирования от 0 до значения `dina`. Соответственно, обращение к числовому методу интегрирования обыкновенных дифференциальных уравнений может быть задано в виде

$$yf = \text{odeint}(\text{fun_dxdt}, y0, \text{хopti}, \text{args}=(\text{um}, 1), \text{rtol}=1\text{e-}11, \text{atol}=1\text{e-}12) \quad (12)$$

В (12) `fun_dxdt` – пользовательская функция (метод) с описанием дифференциальных уравнений динамического объекта, `y0` – вектор начальных условий, кортеж `args` передаёт входные параметры в метод `fun_dxdt`. Значения `rtol`, `atol` задают относительную и абсолютную точность вычислений. Возвращаемое значение `yf` (с любым допустимым обозначением

для Python) содержит в себе список решения дифференциальных уравнений по каждой из переменных состояния исследуемого динамического объекта.

Значение $dina$ в (11) рассчитывается как расстояние между двумя точками в трёхмерном пространстве. Например, по формуле из [24]:

$$dina = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \quad (13)$$

где $x_1, x_2, y_1, y_2, z_1, z_2$ – координаты двух точек $M_1(x_1, y_1, z_1), M_2(x_2, y_2, z_2)$ в трёхмерном пространстве.

Формула (13) будет применяться для определения абсолютной ошибки Δ достижения цели в 3D пространстве. Можно организовать итерационный процесс достижения цели в 3D относительно допустимой ошибки Δ . При этом на каждой итерации поиска предлагается следующий приём, позволяющий связывать значения маркеров цели с расчётом абсолютной ошибки, вычисляемой по формуле (13). По аналогии с [17] на втором этапе поиска решения задачи выполним только численное интегрирование в виде

$$yopti = \text{odeint}(\text{fun1_dxdt}, y0, \text{хоpti}, \text{args}=(Us, 1), \text{rtol}=1e-11, \text{atol}=1e-12) \quad (14)$$

Для этого определим следующий массив:

$$to = \text{np.linspace}(0, 1, \text{Node}, \text{dtype} = \text{float}, \text{endpoint}=\text{True}) \quad (15)$$

Считая, что маркеры цели x_p, y_p, z_p вводятся относительно нулевой точки 3D, то тогда используем параметрическое уравнение прямой в пространстве [23], т.е.

$$\begin{aligned} xfix &= x_p * to \\ yfix &= y_p * to \\ zfix &= yopti[-1, kix] \end{aligned} \quad (16)$$

В (16) $zfix$ принимает последнюю строку решения для индекса kix вектора состояния $X(t)$. Тогда абсолютная ошибка $erka$ по формуле (13) будет вычисляться программно средствами Python в виде

$$erka = \text{np.sqrt}((xfix[-1] - x_p)**2 + (yfix[-1] - y_p)**2 + (zfix - z_p)**2) \quad (17)$$

Если значение (17) не превосходит допустимую ошибку Δ , то выполняется построение движения динамического объекта (ДО) по выражению

$$\text{ax.plot3D}(xfix, yfix, yopti[:, kix], c = 'r', lw = 1, label = \text{'Траектория ДО'}) \quad (18)$$

В (18) цвет траектории красный: $c = 'r'$. Толщина линии $lw = 1$.

Для маркеров цели допустимо использовать следующий код:

$$\begin{aligned} pxyz &= \text{ax.plot3D}([x_p], [y_p], [z_p], label=\text{'Маркеры цели'}) \\ \text{plt.setp}(pxyz, \text{linestyle}='-', \text{linewidth}=0.5, \text{marker}='v', \text{markersize}=3, \\ &\text{markeredgecolor} = 'b', \text{markerfacecolor}='c') \end{aligned} \quad (19)$$

Маркеры по (19) могут находиться на внешней функции или стационарной кривой, а также случайным образом в трёхмерном пространстве. Для расположения динамического объекта в 3D пространстве:

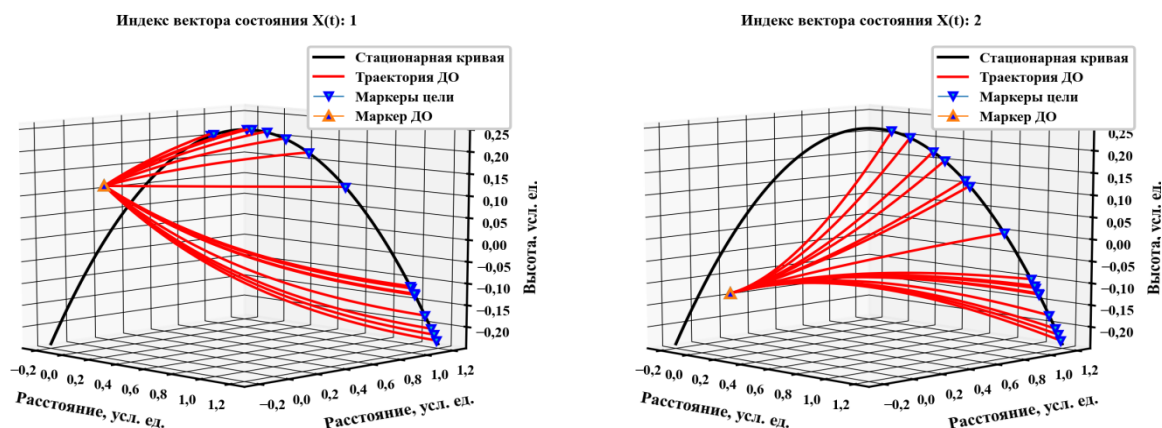
$$\begin{aligned} \text{dar} &= \text{plt.plot}([xfix[0]], [yfix[0]], [yopt2i[0, kix]], label=\text{'Маркер ДО'}) \\ \text{plt.setp}(\text{dar}, \text{linestyle}='-', \text{linewidth}=0.5, \text{marker}='^', \text{markersize}=3.5, \text{markeredgecolor} = 'r', \text{markerfacecolor}='b') \end{aligned} \quad (20)$$

Обсуждение результатов. Перейдем к результатам модельных экспериментов.

Для примера (1), (2) примем следующие значения начального вектора состояния:

$$x_1(0) = 0,123; x_2(0) = -0,123; x_3(0) = 0,053; \quad (21)$$

Результат решения задачи (1), (2) показан на рис. 1, рис. 2.



а) б)
Рис. 1 - Диаграммы вывода ДО в точки стационарной кривой
Fig. 1 - Diagrams of the output of DO at the points of the stationary curve

Из сохранённых текстовых результатов выделим следующие:

Номер переменной состояния: 1

Координаты цели (x, y, z):
[0.3815, 0.3815, 0.23594]

Абсолютная ошибка достижения цели:
 $\max \Delta = 0.0049049802332035985$

Управление U, выводящее ДО до цели:
 $U = [0.3877972253348239]$

Оценка времени T достижения цели:
 $T = 0.040000200271606445$, с.

Номер переменной состояния: 2

Координаты цели (x, y, z):
[1.1945, 1.1945, -0.23234]

Абсолютная ошибка достижения цели:
 $\min \Delta = 0.0007332997805385433$

Управление U, выводящее ДО до цели:
 $U = [-0.23593178785671937]$

Оценка времени T достижения цели:
 $T = 0.18000030517578125$, с.

Результаты достижения целей, показанные на рис. 1 а и б, практически одинаковые. Показана максимальная ошибка $\max \Delta$ и минимальная – $\min \Delta$.

Как видно из рис. 1, рис. 2, переменные состояния №1, №2 более управляемые в плане достижения целей на стационарной кривой вида (2). Программная реализация тестового примера (1, 2) была выполнена на языке программирования Python в среде IDLE Shell 3.8.9 на компьютере с операционной памятью в 4 Гб.

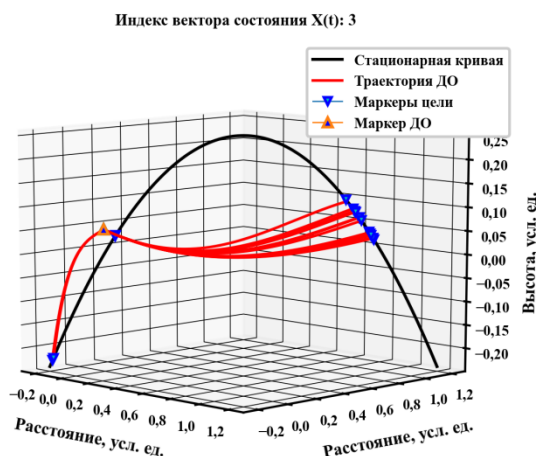


Рис. 2 - Диаграмма вывода ДО по переменной состояния №3 в точки стационарной кривой
Fig. 2 - Diagram of the output of the DO for the state variable No. 3 to the points of the stationary curve

Сохранённые результаты для индекса вектора состояния № 3:

Координаты цели (x, y, z):
[0.8655, 0.8655, 0.11641]

Абсолютная ошибка достижения цели:

$\max \Delta = 0.004779951386761863$

Управление U, выводящее ДО до цели:

$U = [1.2754083234069766]$

Оценка времени T достижения цели:
 $T = 5.190007209777832$, с.

Координаты цели (x, y, z):
[0.0425, 0.0425, 0.04072]

Абсолютная ошибка достижения цели:

$\min \Delta = 0.0006428294171631557$

Управление U, выводящее ДО до цели:

$U = [1.032172590020873]$

Оценка времени T достижения цели:
 $T = 5.690007925033569$, с.

Результаты решения задачи (3), (4) показаны на рис. 3 для 1-й переменной состояния (а) и для 2-й (б).

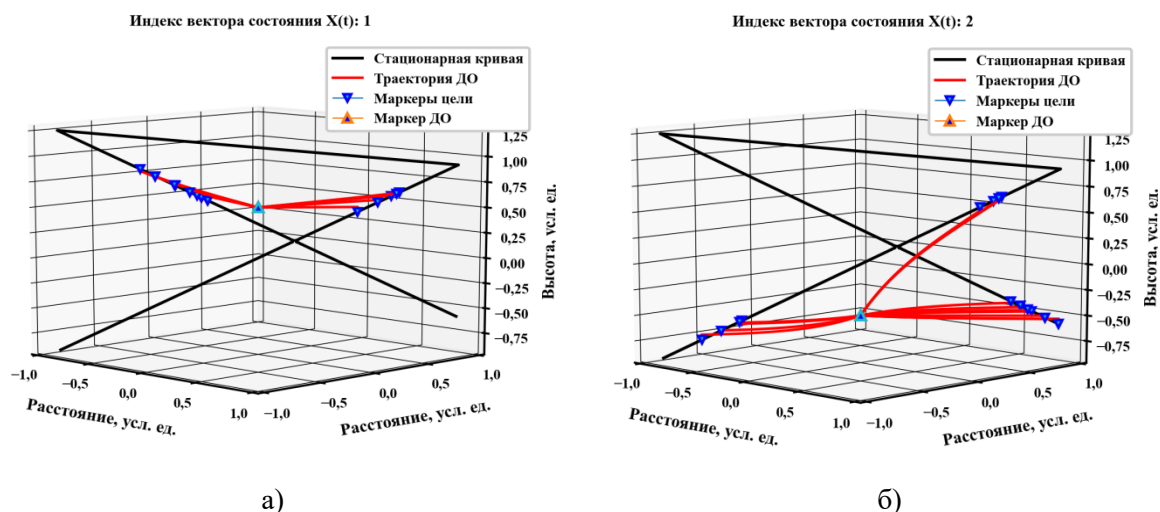


Рис. 3 - Вывод динамического объекта с рециклом на стационарную кривую
Fig. 3 - Output of a dynamic object with recycling to a stationary curve

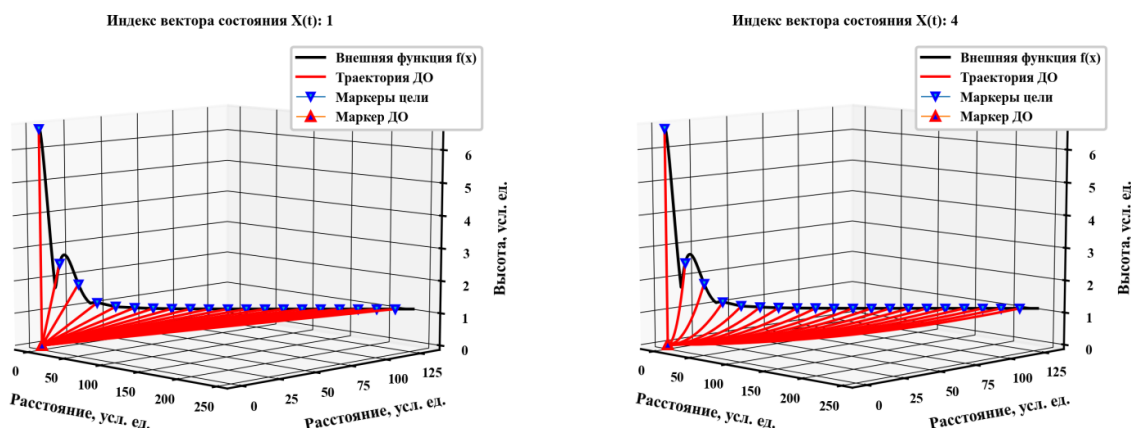
Основные показатели численного интегрирования (3) представлены ниже:

Допустимое значение $\Delta = 0.05$; Параметр ode = 1000; Значения начального вектора состояния: $x_1(0) = 0.5$; $x_2(0) = -0.5$;	Допустимое значение $\Delta = 0.05$; Параметр ode = 1000; Начальный вектор состояния: $x_1(0) = 0.5$; $x_2(0) = -0.5$;
Координаты цели (x, y, z): [0.45668, 0.45668, 0.45673000000000001] Абсолютная ошибка достижения цели: $\max \Delta = 0.04808310850446684$ Управление U, выводящее ДО до цели: $U = [0.25607695797610475; 0.25607695797610475]$ Оценка времени T достижения цели: $T = 1.7316029071807861$, с.	Координаты цели (x, y, z): [-0.73778, -0.73778, -0.7378] Абсолютная ошибка достижения цели: $\max \Delta = 0.049644406359140913$ Управление U, выводящее ДО до цели: $U = [-1.185734331982073; -1.185734331982073]$ Оценка времени T достижения цели: $T = 1.1700019836425781$, с.
Координаты цели (x, y, z): [0.63648, 0.63648, 0.63642000000000001] Абсолютная ошибка достижения цели: $\min \Delta = 0.000554262686415341$ Управление U, выводящее ДО до цели: $U = [0.3682437757965719; 0.3682437757965719]$ Оценка времени T достижения цели: $T = 1.7784030437469482$, с.	Координаты цели (x, y, z): [-0.64872, -0.64872, -0.64871] Абсолютная ошибка достижения цели: $\min \Delta = 0.002524875495493628$ Управление U, выводящее ДО до цели: $U = [-1.092639737075871; -1.092639737075871]$ Оценка времени T достижения цели: $T = 1.762803316116333$, с.
Номер переменной состояния: 1	Номер переменной состояния: 2

Как видно из приведенных результатов, управление, выводящее ДО к цели стационарной кривой имеет кусочно-постоянное значение. Параметр ode используется для решателя дифференциальных уравнений odeint модуля scipy Python.

Для следующего примера (5) принята прямоугольная координатная плоскость XOY в 3D пространстве. Объект с математической моделью (5) является нелинейным объектом 4-го порядка с одним входным управляющим воздействием.

Результаты решения синтеза управления для объекта (5) представлены на рис. 4 для 1-го индекса вектора состояния (а) и для 4-го (б). На рис. 4 представлены диаграммы решения задачи только для 1, 4 переменных состояния.



а)

б)

Рис. 4 - Вывод динамического нелинейного объекта 4-го порядка в точки внешней функции $f(x)$
Fig. 4 - Output of a dynamic nonlinear object of the 4th order to the points of the external function $f(x)$

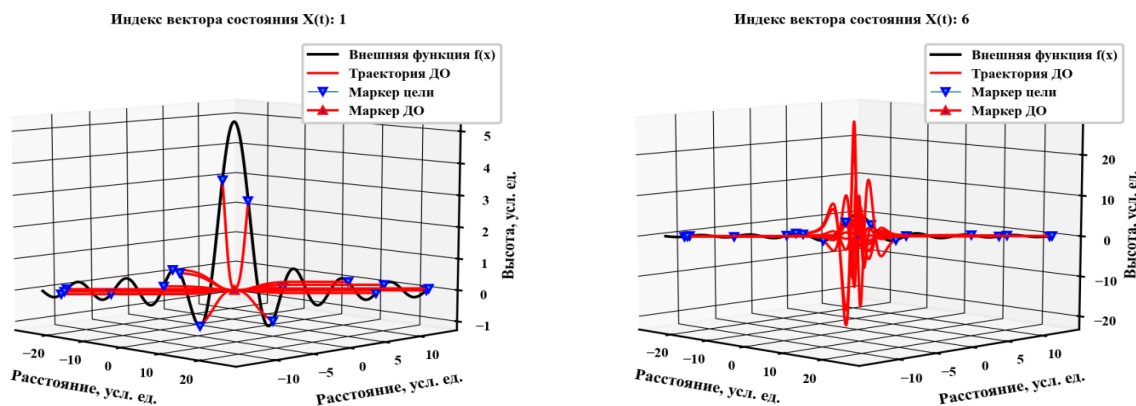
Результаты решения задачи (5) практически идентичны для переменных состояния 1, 3, 4. По переменной состояния 2 объект не управляем.

Ниже приведены характерные результаты синтеза:

Для переменной состояния 1 вектора состояния Допустимое значение $\Delta = 0.005$; Параметр ode = 1000; Границы (ax, bx) по оси абсцисс: ax = 0.0; bx = 245.5; Границы (ay, by) по оси ординат: ay = 0.0; by = 130.115; Координаты цели (x, y, z): [0.0, 0.0, 6.635] Абсолютная ошибка Δ достижения цели: min$\Delta = 0.0$ Управление U, выводящее ДО до цели: U = [1.0] Оценка времени T достижения цели: T = 0.04680013656616211, с. Координаты цели (x, y, z): [122.99599198396794, 65.18787575150301, 1.131387532433282] Абсолютная ошибка Δ достижения цели: max$\Delta = 4.440892098500626e-16$ Управление U, выводящее ДО до цели: U = [0.00812734082335044] Оценка времени T достижения цели: T = 0.23400020599365234, с.	Для переменной состояния 4 вектора состояния Допустимое значение $\Delta = 0.005$; Параметр ode = 1000; Границы (ax, bx) по оси абсцисс: ax = 0.0; bx = 245.5; Границы (ay, by) по оси ординат: ay = 0.0; by = 130.115; Координаты цели (x, y, z): [0.0, 0.0, 6.635] Абсолютная ошибка Δ достижения цели: min$\Delta = 0.0$ Управление U, выводящее ДО до цели: U = [0.30143180105501133] Оценка времени T достижения цели: T = 0.04680013656616211, с. Координаты цели (x, y, z): [122.99599198396794, 65.18787575150301, 1.131387532433282] Абсолютная ошибка Δ достижения цели: $\Delta = 4.440892098500626e-16$ Управление U, выводящее ДО до цели: U = [0.00011676577117097365] Оценка времени T достижения цели: T = 0.20280051231384277, с.
--	--

Далее приводятся результаты анализа и синтеза линейных динамических объектов в соответствии с (6), (7), (8). Для устойчивого и полностью управляемого динамического объекта с математическим описанием (7) результаты представлены на рис. 5, где приводятся результат управления по 1-й переменной состояния (а) и по 6-й переменной (б).

На рис. 5 приведены диаграммы достижения цели на внешней функции для двух индексов вектора состояния. Для остальных индексов задача достижения целей с высокой точностью также выполняется.



а) б)
**Рис. 5 - Пример вывода линейного динамического объекта 6-го порядка
в точки внешней функции $f(x)$**

Fig. 5 - Example of output of a 6th order linear dynamic object to the points of an external function $f(x)$

Ниже приведены некоторые результаты синтеза для линейного стационарного динамического объекта 6-го порядка:

Число строк матрицы состояния A: 6. Число столбцов матрицы состояния A: 6
Собственные значения/числа матрицы состояния A исходного динамического объекта:
[(-8.249995703567727+0j), (-1.163534742827491+1.7704121492694718j),
(-1.163534742827491-1.7704121492694718j), (-1.1266865997094229+0j),
(-0.5331241055339142+0.4006883238317334j), (-0.5331241055339142-0.4006883238317334j)]
Исходный динамический объект (ДО) асимптотически устойчив.
Пара (A, B) динамического объекта полностью управляема по критерию Калмана.
Допустимое значение ошибки $\Delta = 0.005$;
Детерминант исходной матрицы A: $|A| = 18.555140340180024$
Детерминант $|A_{\text{преобр}}|$ преобразованной матрицы A: $|A_{\text{преобр}}| = 18.555140340180024$
Параметр ode = 2000;
Границы (a, b) функции $f(x)$: a = -25; b = 25.5;

Множитель μ к A: 1;
Номер переменной состояния: 1
Число решений для ДО: 17

Координаты цели (x, y, z):

[14.873747494989985,
7.883086172344693,
0.2639604188833907]

Абсолютная ошибка $\min \Delta$ достижения цели:

$\min \Delta = 4.9404924595819466e-15$

Управление U, выводящее ДО до цели:

$U = [0.0005215325239339573;$
 $0.0005215325239339573;$
 $0.0005215325239339573]$

Оценка времени T достижения цели:

$T = 1.1856019496917725$, с.

Координаты цели (x, y, z):

[1.8186372745491006,
0.9638777555110234, 2.825222474871464]

Абсолютная ошибка $\max \Delta$ достижения цели:

$\max \Delta = 1.7053025658242404e-12$

Управление U, выводящее ДО до цели:

$U = [0.01132057875764698;$
 $0.01132057875764698; 0.01132057875764698]$

Оценка времени T достижения цели:

$T = 0.35880064964294434$, с.

Множитель μ к A: 1;
Номер переменной состояния: 6
Число решений для ДО: 17

Координаты цели (x, y, z):

[-21.862725450901802,
-11.587244488977955,
0.031047052483193092]

Абсолютная ошибка $\min \Delta$ достижения цели:

$\min \Delta = 7.794806466954185e-14$

Управление U, выводящее ДО до цели:

$U = [-0.001981698683962924;$
 $-0.001981698683962924;$
 $-0.001981698683962924]$

Оценка времени T достижения цели:

$T = 0.468000888244629$, с.

Координаты цели (x, y, z):

[-1.5210420841683359,
-0.806152304609218, 3.4801412623026935]

Абсолютная ошибка $\max \Delta$ достижения цели:

$\max \Delta = 3.9659608930264767e-10$

Управление U, выводящее ДО до цели:

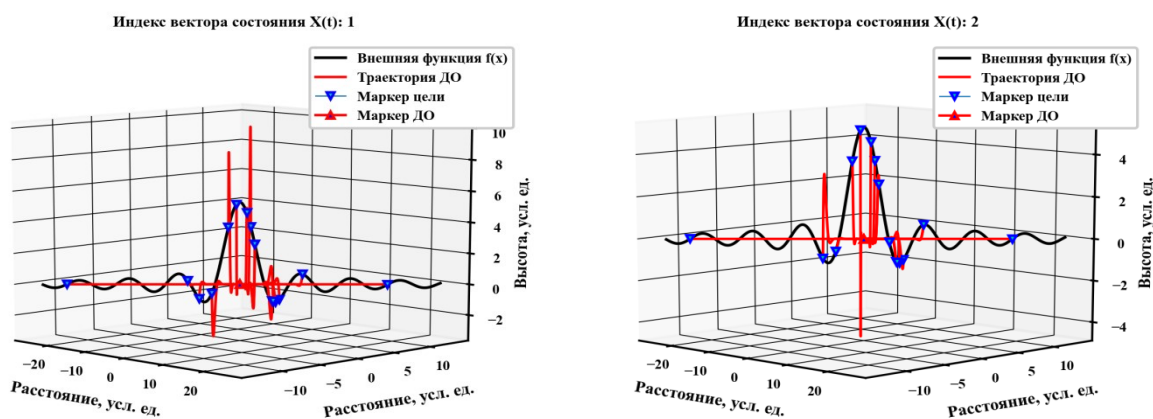
$U = [-0.030377731338165927;$
 $-0.030377731338165927; -0.030377731338165927]$

Оценка времени T достижения цели:

$T = 1.8720033168792725$, с.

Множитель $0 < \mu \leq 1$ к матрице состояния A приводит к ее изменению, что возможно в некоторых случаях допустимо. И это позволяет решать задачу достижения предопределенных целей. Для устойчивого и полностью управляемого рассмотренного динамического объекта 6-го порядка приемлемый множитель, равный единице.

Пример (8) представляет собой линейный объект 3-го порядка, который асимптотически неустойчивый и не полностью управляемый. Программные исследования показали, что объект управляем по всем переменным состояния объекта, который не полностью управляемый. Ранг пары (A, B) равен 2, что указывает на неполноту управляемости, поскольку динамический объект 3-го порядка. При множителе $\mu < 1$ характер вывода динамического объекта в заданные цели функциональной внешней траектории заметно изменяется. Результаты решения задачи с объектом 3-го порядка и матрицами (8) представлены на рис. 6 для переменных 1, 2 индекса вектора состояния (a, b), рис. 7 для переменной состояния 3 вектора состояния с различными (a, b) значениями множителя μ к матрице состояния A . Следует отметить, что по формальным признакам достижения целей не полностью управляемый объект 3-го порядка под действием рассчитанных управляющих воздействий достигает некоторых целей при $\mu = 1$. Уменьшение множителя позволяет расширить множество достигаемых целей на тестовой внешней функции (10).

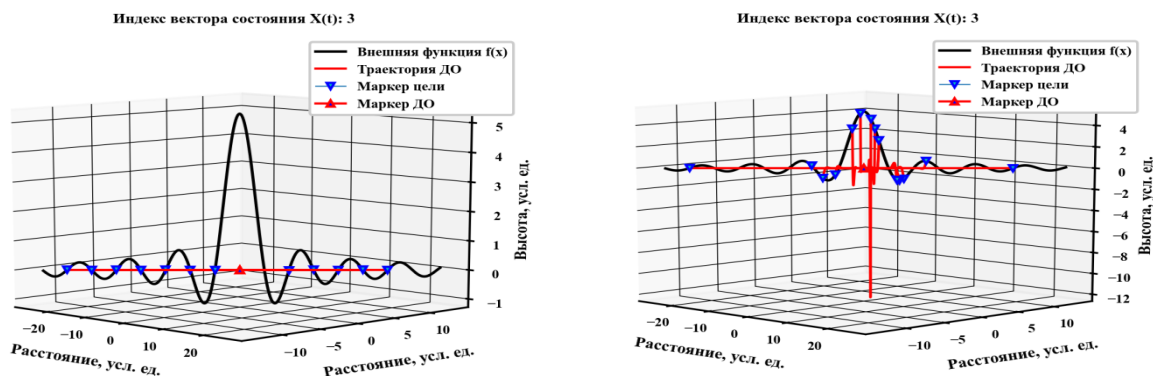


а) $\mu = 0,2$

б) $\mu = 0,2$

Рис. 6 - Управление линейным динамическим объектом 3-го порядка с множителем $\mu=0,2$ к матрице состояния A

Fig. 6 - Control of a 3rd order linear dynamic object with a multiplier $\mu = 0.2$ to the state matrix A



а) $\mu = 1$

б) $\mu = 0,2$

Рис. 7 - Управление линейным динамическим объектом 3-го порядка с различными множителями к матрице состояния A

Fig. 7 - Control of a 3rd order linear dynamic object with different multipliers to the state matrix A

Общий вывод по управлению неустойчивым и не полностью управляемым по критерию Калмана линейным объектом 3-го порядка: предложенный прием расчета управляющих воздействий позволяет управлять в той или иной степени в зависимости от величины числового множителя к матрице состояния A .

Следующий пример касается устойчивых колебательных звеньев с одним входным управлением. Приняты следующие параметры звеньев: $k = 1,23$ (статический коэффициент передачи), $T = 1,2$ (постоянная времени), $\xi = 0,0123$ (относительный коэффициент демпфирования). Два результата перевода такого динамического объекта в точки внешней траектории приведены на рис. 8.

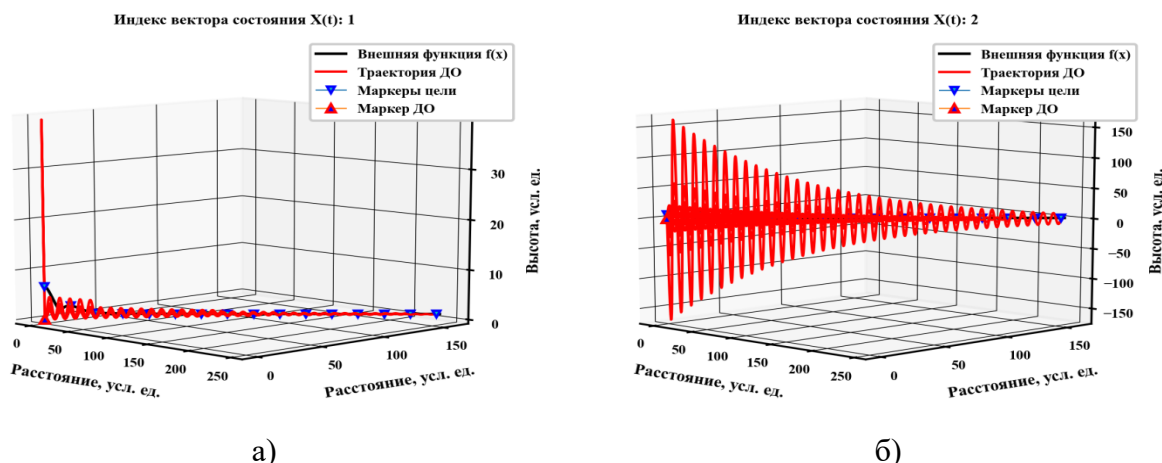


Рис. 8 - Диаграммы (а, б) управления объектом, состоящим из параллельного соединения 4-х устойчивых колебательных звеньев

Fig. 8 - Diagrams (a, b) of control of an object consisting of a parallel connection of 4 stable oscillatory links

Некоторые текстовые результаты в соответствии с рис. 8 (а, б):

Координаты цели (x, y, z): [178.59018036072143, 116.08361723446896, 1.1300708762872256] Абсолютная ошибка Δ достижения цели: $\Delta = 2.6116886431282182e-11$ Управление U, выводящее ДО до цели: $U = [0.9209914462752539;$ 0.9209914462752539; 0.9209914462752539; 0.9209914462752539] Номер переменной состояния: 1	Координаты цели (x, y, z): [178.59018036072143, 116.08361723446896, 1.1300708762872256] Абсолютная ошибка Δ достижения цели: $\Delta = 2.46219933330849e-10$ Управление U, выводящее ДО до цели: $U = [9.785054009442925;$ 9.785054009442925; 9.785054009442925; 9.785054009442925] Номер переменной состояния: 2
---	---

Из рис. 8 для 1-й переменной состояния можно предположить, что при технической реализации могут быть недопустимы выбросы переменных состояния. Уменьшение выбросов возможно при увеличении постоянных времени. Но, как показали модельные эксперименты, задача перевода объекта в точки какой-либо внешней траектории практически всегда разрешима. Уменьшение выбросов может быть достигнуто в случае неустойчивых колебательных звеньев, когда для них принимаются постоянные звенья выше некоторого значения.

На рис. 9 приведены диаграммы для звеньев, соединённых параллельно, у которых постоянные времени равны 17,2 усл. ед. В соответствии с результатами на рис. 9 можно указать, что неустойчивые линейные динамические объекты могут быть более предпочтительными при конструировании и моделировании систем управления для синтеза управления, обеспечивающего вывод динамического объекта в заданные точки 3D пространства.

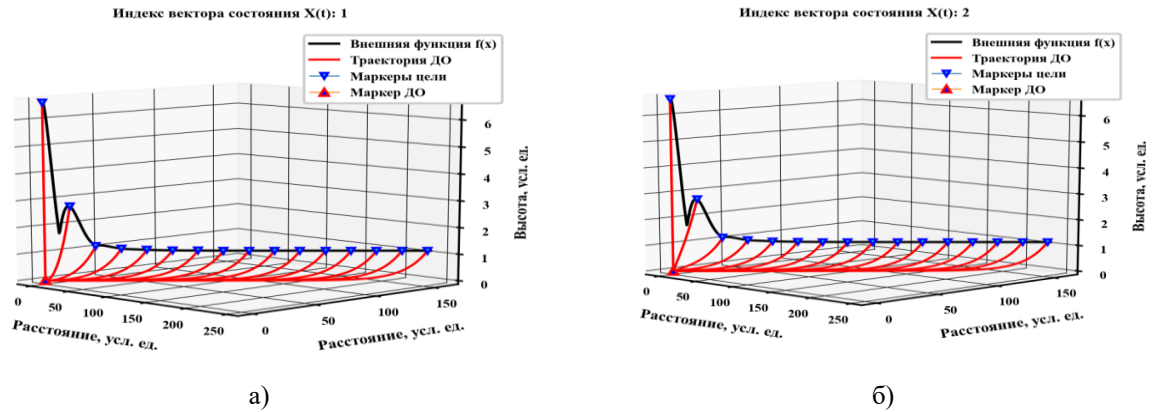


Рис. 9 - Диаграммы (а, б) управления объектом, состоящим из параллельного соединения 4-х неустойчивых колебательных звеньев
 Fig. 9 - Diagrams (a, b) of control of an object consisting of a parallel connection of 4 unstable oscillatory links

Здесь приводится программный код построения движения устойчивого колебательного объекта 4-го порядка к заданным целям в 3D пространстве:

```
import math # +++
from math import fabs, cos, pow, exp, sqrt
# import numpy
import numpy as np
import scipy
from scipy.integrate import odeint
import matplotlib
import matplotlib.pyplot as plt
## Локаль - региональные стандарты
import locale
locale.setlocale(locale.LC_NUMERIC, "rus")
plt.rcParams()
plt.rcParams['axes.formatter.use_locale'] = True
deta = chr(916) # греческая буква Дельта

def fun_fx(x):
    fx = 5.3 * fabs(cos(0.15 * x) - 0.15) * (pow(x, -0.03 * x)) +
    exp(-0.0535 * x) + 1.13
    return fx

def fun_dxdx(x, t, u):
    dx = [x[1], -x[0] / (1.2 * 1.2) - 2.0 * 0.0123 * x[1] / 1.2 + 1.23 * u[0] / (1.2 * 1.2), x[1],
    -x[0] / (1.2 * 1.2) - 2.0 * 0.0123 * x[1] / 1.2 + 1.23 * u[0] / (1.2 * 1.2), x[1], -x[0] / (1.2 * 1.2)
    - 2.0 * 0.0123 * x[1] / 1.2 + 1.23 * u[0] / (1.2 * 1.2), x[1], -x[0] / (1.2 * 1.2) - 2.0 * 0.0123 *
    x[1] / 1.2 + 1.23 * u[0] / (1.2 * 1.2)]
    return dx

def fun3d(xak, yak):
    ## Diagram
    Nx = len(xak)
    Node = 1000 # параметр ode
    to = np.linspace(0, 1, Node, dtype = float, endpoint=True)

    plt.rcParams['font.family'] = 'Times New Roman'
    plt.rcParams['figure.autolayout'] = False
    plt.rcParams['xtick.labelsize'] = 6.5

    Wd = 1366 # ширина экрана дисплея
    Hd = 768 # высота -----
    fw = math.ceil(Wd * 0.003)
    fh = math.ceil(Hd * 0.0036)
    fg = plt.figure("Тест", figsize = (fw, fh), dpi=150)

    ax = plt.axes(projection = '3d')
    plt.axis('on')
    plt.grid(True)
    ax.grid(which="major", color='black', linewidth=0.5, linestyle='-', alpha=0.53)

    plt.rcParams['figure.constrained_layout.use'] = True
    fg.set_tight_layout(True)
    fg.tight_layout()
```

```

## Приближение цифровой разметки к осям
ax.tick_params(axis='x', which='major', pad=-3.5)
ax.tick_params(axis='y', which='major', pad=-3.5)
ax.tick_params(axis='z', which='major', pad=-3.5)
## Ширина линий координатной сетки
ax.xaxis._axinfo["grid"]['linewidth'] = 0.25
ax.yaxis._axinfo["grid"]['linewidth'] = 0.25
ax.zaxis._axinfo["grid"]['linewidth'] = 0.25
## Цвет rgb
ax.xaxis._axinfo["grid"]['color'] = [0,0,0]
ax.yaxis._axinfo["grid"]['color'] = [0,0,0]
ax.zaxis._axinfo["grid"]['color'] = [0,0,0]
mno = 0.65
yy = np.linspace(0, xak[-1]*mno, Nx, endpoint=True)#, retstep=True, axis=0)
ax.plot3D(xak, yy, yak, 'k', linewidth = 1.125, label='Внешняя функция')
ax.view_init(10, -45)
ax.set_facecolor('1') # 1 - белый

## Контейнер лист маркеров цели
xyz = [178.59018036072143, 116.08361723446896, 1.1300708762872256]
rx0 = 0; kix = 1 # номер переменной состояния
y0 = [0.0]*8 # начальный вектор состояния

xp = xyz[0] # маркер цели по оси абсцисс
yp = xyz[1] # маркер цели по оси ординат
zp = xyz[2] # маркер цели по оси аппликат

## Расстояние от начала координат до маркера цели
dina = np.sqrt(xp**2 + yp**2 + zp**2)

хоpti = np.linspace(rx0, dina, Node, dtype = float, endpoint=True)

U = [9.785054009442925]*4 ## из результатов моделирования

yopti = odeint(fun_dxdt, y0, хopti, args=(U, ), rtol=1e-11, atol=1e-12)
xfix = xp*to
yfix = yp*to
zfix = yopti[:, kix]
erka = sqrt((xp-xfix[-1])**2 +(yp-yfix[-1])**2 +(zp-zfix[-1])**2)

ax.plot3D(xfix, yfix, yopti[:, kix], c = 'r', lw = 0.72, label = 'Траектория Д0')

pxyz = ax.plot3D([xp], [yp], [zp], label=f"Маркер цели")
plt.setp(pxyz, linestyle='-', linewidth=0.5, marker='v', markersize=3, markeredgcolor = 'b', markerfacecolor='c')

dar = plt.plot([xfix[0]], [yfix[0]], [yopti[0, kix]], label="Маркер Д0")
plt.setp(dar, linestyle='-', linewidth=0.5, marker='^', markersize=3.5, markeredgcolor = 'r', markerfacecolor='b')

plt.legend(loc='upper right', frameon=True, shadow=False, prop={'size': 6.0}, facecolor='white', framealpha=1)
ax.set_xlabel("Расстояние, усл. ед.", fontsize = 8.0, labelpad=-2.0)
ax.set_ylabel("Расстояние, усл. ед.", fontsize = 8.0, labelpad=-2.0)
ax.set_zlabel("Высота, усл. ед.", fontsize = 8.0, labelpad=-6.0)

plt.title(f"Индекс вектора состояния X(t): {kix + 1}\n\nАбсолютная ошибка {deta} достижения цели: {deta} = {erka}", fontsize = 7.5)
ax.xaxis._axinfo["grid"]['linewidth'] = 0.45
ax.yaxis._axinfo["grid"]['linewidth'] = 0.45
ax.zaxis._axinfo["grid"]['linewidth'] = 0.45

plt.show()
if __name__ == '__main__':
a = 0.0 # левая граница функции f(x)
b = 245.5 # правая граница функции f(x)
n = 10000 # количество итераций построения графика внешней функции

xak = np.linspace(a, b, n, endpoint=True)
yak = []
for x in xak:
    yak.append(fun_fx(x))

fun3d(xak, yak)
print("Test 3d")

```

Результат по предложенному коду показан на рис. 10.

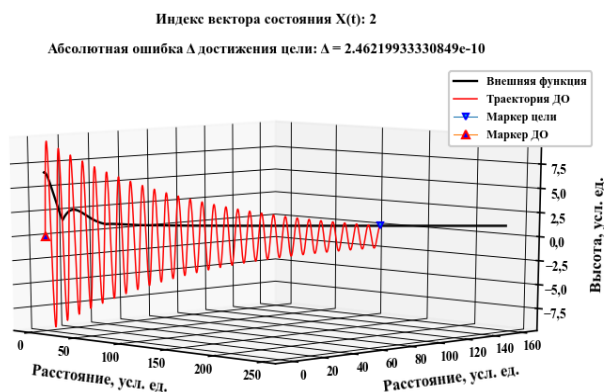


Рис. 10 - Диаграмма с выделенным графиком решения устойчивого колебательного динамического объекта 4-го порядка
Fig. 10 - Diagram with highlighted graph of the solution of a stable 4th-order oscillatory dynamic object

Как видно из рис. 10, по известному значению управляющего воздействия и числовым значениям цели (маркеров цели) программное решение задачи достаточно оперативно реализуется.

Вывод. Анализ и синтез систем автоматического управления - одна из актуальных задач теоретических и прикладных исследований. Как известно, нелинейные динамические системы управления имеют особенно большое разнообразие по сравнению с линейными стационарными системами [4-6, 12].

В данной работе в плане продолжения работы [17] предложено численное эвристическое решение задачи вывода динамических объектов в заданные цели трёхмерного пространства. Особенностью решения является то, что управляющие воздействия имеют кусочно-постоянные значения, что позволяет достаточно надёжно обеспечить их техническую реализацию. Решение выполнено на языке программирования Python.

Полученные результаты подтвердили, что в случае нелинейных динамических объектов нельзя быть уверенным распространить предложенный подход синтеза на все типы нелинейных систем управления. Однако для линейных объектов решение практически достижимо как для многомерных (MIMO), так и для одномерных (SISO) динамических объектов управления.

Изложенные результаты с графической визуализацией показывают как частный характер для нелинейных динамических объектов, так и в некоторой степени общий характер для линейных стационарных динамических объектов при решении задачи их вывода в точки 3D пространства на основе предлагаемого подхода.

Библиографический список:

1. Смирнова С.А. Комплекс имитационного моделирования систем адаптивно-робастного управления многорежимными объектами с переключениями. Информатика и системы управления. 2023. № 3(77). С. 29-40.
2. Т.Г. Гасанов, Э.З. Батманов, М.Р. Гусейнов, М.Т. Муталибов. Математическое моделирование угловых перемещений базовой машины двухконсольного дождевального агрегата. Вестник Дагестанского государственного технического университета. Технические науки. 2023; 50(2):41-47. DOI:10.21822/2073-6185-2023-50-2-41-47
3. Коваленко О.Ю., Афонин В.В. Программный синтез светодиодного модуля // Вестник Федерального государственного образовательного учреждения высшего профессионального образования «Московский государственный агроинженерный университет имени В.П. Горячкина». 2009. № 2 (33). С. 52-54.
4. А.К. Юсупов, Х.М. Муселемов, Р.И. Вишталов. Нелинейные модели балок с переменной жёсткостью. Вестник Дагестанского государственного технического университета. Технические науки. 2025;52(2):235-246. DOI:10.21822/2073-6185-2025-52-2-235-246

5. Поляк Б.Т., Щербakov П.С. Трудные задачи линейной теории управления. Некоторые подходы к решению // Автоматика и телемеханика. 2005. Вып. 5. С. 7-46.
6. Ким Д.П. Теория автоматического управления: учебник и практикум для вузов / Д.П. Ким. – Москва: Издательство Юрайт, 2025. – 276 с.
7. Методы классической и современной теории автоматического управления: учебник в 5-ти т. Т. 1: Математические модели, динамические характеристики и анализ систем автоматического управления / Под общ. ред. К.А. Пупкова. – 2-е изд., пере-раб. и доп. – М.: Изд-во МГТУ, 2004. – 656 с.
8. Соколов С.В., Охотников А.Л. Робастное позиционирование беспилотных объектов с использованием спутниковых измерений и данных цифровой модели пути // Мехатроника, автоматизация, управление. 2024; 25(7): С. 372-379. – <https://doi.org/10.17587/mau.25.372-379>
9. Деменков Н.П., Цзоу К. Метод прогнозирования траектории на основе графовой модели для беспилотного вождения // Автоматизация. Современные технологии. 2022. Т. 76. № 9. С. 408-413.
10. Man K., Damasio A. Homeostasis and soft robotics in the design of feeling machines // Nature machine intelligence. 2019. Vol. 1. P. 446-452.
11. Shahzad M.M., Saeed Z., Akhtar A., Munawar H., Yousaf M.H., Baloach N.K., Hussain F. A review of swarm robotics in a NutShell // Drones. 2023. Vol. 7, No. 4. P. 269.
12. Р.В. Гусейнов, М.Р. Гусейнова, К.А. Алиева. Некоторые вопросы многокритериальной оптимизации параметров сложных систем. Вестник Дагестанского государственного технического университета. Технические науки. 2023; 50(2): 67-75. DOI:10.21822/2073-6185-2023-50-2-67-75
13. Олейников В.А. Оптимальное управление технологическими процессами в нефтяной и газовой промышленности. – Л.: Недра, 1982. – 216 с.
14. Хорошавин В.С. Особое оптимальное управление нелинейными объектами: [монография] / В.С. Хорошавин, А.В. Зотов. – Киров: Вятский государственный университет, 2019. – 208 с.
15. Пантелеев А.В. Обыкновенные дифференциальные уравнения. Практический курс / Ф.В. Пантелеев, А.С. Якимова, К.А. Рыбаков. – М.: Логос, 2010. – 384 с.
16. Зайцев В.Ф. Обыкновенные дифференциальные уравнения. В 2 частях. Ч. 1: справочник для вузов / В.Ф. Зайцев, А.Д. Поляник. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2024. – 385 с. – (Высшее образование).
17. Афонин В.В. Моделирование движения линейных объектов управления к внешней функциональной траектории. Вестник Дагестанского государственного технического университета. Технические науки. 2025;52(1):49-56. <https://doi.org/10.21822/2073-6185-2025-52-1-49-56>
18. Афонин В.В. Вывод нелинейного объекта 3-го порядка на заданное движение // Вестник Мордовского университета, серия «Физико-математические науки». – Саранск : Изд-во Мордов. ун-та, 2010. – 136 с. – С. 52-54.
19. Краснощеченко В.И. Декомпозиция уравнений нелинейных аффинных систем управления и ее приложение к синтезу регуляторов. Мехатроника, автоматизация, управление. 2024;25(1):3-12. <https://doi.org/10.17587/mau.25.3-12>.
20. Шакирзянов Р.А., Шакирзянов Ф.Р. Динамика и устойчивость сооружений: Учебное пособие. Казань: Издательство Казанского государственного архитектурно-строительного университета, 2015. 120 с.
21. Самое полное руководство по разработке на Python в примерах от сообщества Stack Overflow. – Москва : Издательство АСТ, 2024. – 672 с.
22. Васильев Ю.А. Python для datascience / Ю.А. Васильев. – СПб.: Питер, 2023. – 272 с.
23. Ильичев В.Ю., Гридчин Н.В. Визуализация масштабируемых 3d-моделей с помощью модуля Matplotlib для Python. // Системный администратор. 2020. № 12 (217). С. 86-89.
24. Выгодский М.Я. Справочник по высшей математике. М.: Изд-во АСТ. 2022. 704 с.

References:

1. Smirnova S.A. Simulation complex for adaptive-robust control systems of multi-mode objects with switching. Computer Science and Control Systems. 2023;3(77):29-40. (In Russ)
2. T.G. Gasanov, E.Z. Batmanov, M.R. Huseynov, M.T. Mutalibov. Mathematical modeling of angular displacements of the base machine of a two-console sprinkling unit. Herald of Daghestan State Technical University. Technical Science. 2023; 50(2): 41-47. DOI:10.21822/2073-6185-2023-50-2-41-47 (In Russ)
3. Kovalenko O.Yu., Afonin V.V. Software synthesis of a LED module // Bulletin of the Federal State Educational Institution of Higher Professional Education "Moscow State Agroengineering University named after V.P. Goryachkin". 2009; 2(33):52-54. (In Russ)
4. A.K. Yusupov, Kh.M. Muselemov, R.I. Vishtalov. Nonlinear models of beams with variable stiffness. Herald of the Daghestan State Technical University. Technical Sciences. 2025;52(2): 235-246. (In Russ) DOI:10.21822/2073-6185-2025-52-2-235-246. (In Russ)
5. Polyak B.T., Shcherbakov P.S. Complex problems of linear control theory. Some approaches to solution // Automation and Telemechanics. 2005; 5:7-46. (In Russ)

6. Kim D.P. Automatic control theory: textbook and practical training for universities / D.P. Kim. – Moscow: Yurait Publishing House, 2025; 276 p. (In Russ)
7. Methods of classical and modern theory of automatic control: Textbook in 5 volumes. Volume 1: Mathematical models, dynamic characteristics and analysis of automatic control systems / Under the general editorship of K.A. Pupkov. – 2nd ed., revised and enlarged. – Moscow: Publishing house of Moscow State Technical University, 2004; 656 p. (In Russ)
8. Sokolov S.V., Okhotnikov A.L. Robust positioning of unmanned objects using satellite measurements and digital path model data. *Mechatronics, automation, control*. 2024; 25(7):372–379. DOI.org/10.17587/mau.25.372-379. (In Russ)
9. Demenkov N.P., Zou K. Trajectory prediction method based on a graph model for unmanned driving. *Automation. Modern technologies*. 2022;76(9):408-413.
10. Man K., Damasio A. Homeostasis and soft robotics in the design of feeling machines. *Nature machine intelligence*. 2019;1:446–452.
11. Shahzad M.M., Saeed Z., Akhtar A., Munawar H., Yousaf M.H., Baloach N.K., Hussain F. A review of swarm robotics in a NutShell. *Drones*. 2023; 7(4):269.
12. R.V. Guseynov, M.R. Guseynova, K.A. Alieva. Some issues of multicriterial optimization of parameters of complex systems. *Bulletin of the Dagestan State Technical University. Technical sciences*. 2023; 50(2): 67-75. DOI:10.21822/2073-6185-2023-50-2-67-75 (In Russ)
13. Oleynikov V.A. Optimal control of technological processes in the oil and gas industry. – L.: Nedra, 1982. 216 p. (In Russ)
14. Khoroshavin V.S. Special optimal control of nonlinear objects: [monograph] / V.S. Khoroshavin, A.V. Zotov. – Kirov: Vyatka State University, 2019; 208 p. (In Russ)
15. Panteleev A.V. Ordinary Differential Equations. Practical Course / F.V. Panteleev, A.S. Yakimova, K.A. Rybakov. – M.: Logos, 2010; 384 p. (In Russ)
16. Zaitsev V.F. Ordinary Differential Equations. In 2 parts. Part 1: Handbook for Universities / V.F. Zaitsev, A.D. Polyaniy. - 2nd ed., corrected. and additional. - Moscow: Yurait Publishing House, 2024; 385 p. - (Higher education). (In Russ)
17. Afonin V.V. Modeling the motion of linear control objects to an external functional trajectory. *Herald of Dagestan State Technical University. Technical Sciences*. 2025;52(1):49-56. (In Russ) DOI.org/10.21822/2073-6185-2025-52-1-49-56
18. Afonin V.V. Derivation of a third-order nonlinear object for a given motion. *Bulletin of the Mordovian University, series "Physical and Mathematical Sciences"*. – Saransk: Publishing house of the Mordovian University, 2010;136: 52-54. (In Russ)
19. Krasnoshechenko V.I. Decomposition of Equations of Nonlinear Affine Control Systems and its Application to the Synthesis of Regulators. *Mekhatronika, Avtomatizatsiya, Upravlenie*. 2024;25(1):3-12. (In Russ) <https://doi.org/10.17587/mau.25.3-12>. (In Russ)
20. Shakirzyanov R. A., Shakirzyanov F. R. Dynamics and Stability of Structures: Textbook. Kazan: Publishing House of Kazan State University of Architecture and Civil Engineering, 2015. 120 p. (In Russ)
21. The most comprehensive guide to Python development with examples from the Stack Overflow community. – Moscow: AST Publishing House, 2024. 672 p. (In Russ)
22. Vasiliev Yu. A. Python for datascience / Yu. A. Vasiliev. – St. Petersburg: Peter, 2023. 272 p. (In Russ)
23. Pilychev V.Yu., Gridchin N.V. Visualization of scalable 3D models using the Matplotlib module for Python. *System administrator*. 2020;12 (217):86-89. (In Russ)
24. Vygodsky M.Ya. Handbook of higher mathematics. M.: Publishing house AST. 2022. 704 p. (In Russ)

Сведения об авторе:

Виктор Васильевич Афонин, кандидат технических наук, доцент, доцент кафедры измерительных и инфокоммуникационных технологий; vvafonin53@yandex.ru.

Information about the author:

Viktor V. Afonin, Cand. Sci. (Eng.), Assoc. Prof.; Assoc. Prof., Department of Measurement and Infocommunication Technologies; vvafonin53@yandex.ru.

Конфликт интересов/Conflict of interest.

Автор заявляет об отсутствии конфликта интересов/The author declare no conflict of interest.

Поступила в редакцию/Received 10.12.2025.

Одобрена после рецензирования/Revised 27.02.2026.

Принята в печать/Accepted for publication 26.04.2026.