

*Мишин Павел Александрович, студент*

*ФГБОУ ВО «Национальный исследовательский*

*Мордовский государственный университет им. Н. П. Огарёва»,*

*Россия, г. Саранск*

*Мишина Полина Андреевна, студент*

*ФГБОУ ВО «Национальный исследовательский*

*Мордовский государственный университет им. Н. П. Огарёва»,*

*Россия, г. Саранск*

## **АНАЛИЗ ТОЧНОСТИ ВЫЧИСЛЕНИЯ ОПТИМАЛЬНОГО РЕГУЛЯТОРА НА ОСНОВЕ СООТНОШЕНИЯ ОПТИМАЛЬНОСТИ В СИСТЕМЕ MATLAB И ЯЗЫКЕ PYTHON**

**Аннотация:** В данной статье представлен аналитический контроль решения задачи оптимальной стабилизации для линейных стационарных объектов со скалярным управлением. Произведён анализ точности вычисления оптимального регулятора по евклидовой норме, на примере электропривода, в системе MATLAB и языке Python и сделаны соответствующие выводы.

**Ключевые слова:** оптимальная стабилизация, соотношение оптимальности, функционал качества,  $LQR$ -регулятор, пространство состояний, MATLAB, Python.

**Annotation:** This article presents an analytical control of solving the optimal stabilization problem for linear stationary objects with scalar control. The analysis of the accuracy of calculating the optimal regulator according to the Euclidean norm, using the example of an electric drive, in the MATLAB system and Python, and the corresponding conclusions are made.

**Key words:** optimal stabilization, optimality ratio, quality functional, LQR

controller, state space, MATLAB, Python.

## **Введение**

Промышленные электромеханические комплексы всегда устойчивы по всем переменным состояниям, только в том случае, когда они являются элементами разомкнутых систем управления.

Для одной и той же устойчивой системы существует множество стабилизирующих воздействий, и необходимо выбирать среди них оптимальное с точки зрения критерия (функционала) качества. Такое управление называется оптимальным.

В данной работе будет рассматриваться задача оптимальной стабилизации для линейного стационарного объекта со скалярным управлением. Некоторые результаты решения задач оптимальной стабилизации отображены в работах [1; 2; 3; 4].

## **Материалы и методы**

При решении задачи стабилизации математическая модель линейной стационарной системы может быть представлена в следующем виде:

$$\begin{aligned}\frac{dX(t)}{dt} &= AX(t) + Bu, \\ Y(t) &= CX(t),\end{aligned}\tag{1}$$

где  $X(t)$  –  $n$ -мерный вектор состояния,  $u$  – скалярное управление,  $A, B$  – постоянные матрицы действительных чисел размерностей  $n \times n$ ,  $n \times 1$ , соответственно,  $Y(t)$  – выход системы,  $C$  – постоянная матрица действительных чисел размера  $m \times n$  ( $m \leq n$ ).

Решая задачу о стабилизации, необходимо обеспечить асимптотическую устойчивость заданного движения, а также обеспечить минимизацию следующих функционалов:

$$J_1 = \int_0^{\infty} (X^T(t)QX(t) + u^T R^T u) dt \rightarrow \min,\tag{2}$$

$$J_2 = \int_0^{\infty} (Y^T(t)QY(t) + u^T R^T u) dt \rightarrow \min, \quad (3)$$

где  $Q, R$  – диагональные положительно-определенные матрицы размера  $n \times n, r \times r$  соответственно,  $T$  – символ транспонирования. Функционал (2) является критерием минимума расхода энергии на управление; критерий максимального быстродействия – функционал (3). В силу того, что у нас скалярное управление, матрица  $R$  будет представляться положительным числом.

Решение задачи (1) представляется в форме линейной обратной связи по переменным состояния  $x$  или по переменным  $y$  [3; 6]

$$u = -K_p X(t), u = -K_p Y(t), \quad (4)$$

где  $K_p = R^{-1} B^T P$  – матрица коэффициентов оптимального регулятора размера  $r \times n$ , на значения которой ограничения не наложены, а матрица  $P$  размера  $n \times n$  должна удовлетворять матричному нелинейному алгебраическому уравнению Риккати [8]

$$PA + A^T P - PBR^{-1}B^T P + Q = 0. \quad (5)$$

Основная доля вычислений оптимального управления связана с решением уравнения (5). Выражение (4) обеспечивает минимизацию функционалов (2), (3) [7].

Аналитический контроль решения задачи оптимальной стабилизации с векторным управлением в системе MATLAB представлен в работе [1]. В данной статье рассматривается аналитический контроль со скалярным управлением (4).

Общая формула, используемая как для скалярного, так и для векторного управления имеет вид [4]:

$$R = B^T S^T Q S B + (E + B^T S^T K_p^T) R (E + K_p S B), \quad (6)$$

где  $S = (A - BK_p)^{-1}$ ,  $E$  – единичная матрица размера  $r \times r$ . В данном случае, имея скалярное управление, матрица  $E$  будет представлена единицей.

### Результаты эксперимента

Эксперименты проводились в системе MATLAB и языке Python. В MATLAB существует библиотечная функция *lqr* (*linear-quadratic regulator*), из

пакета Optimization Toolbox, позволяющая найти оптимальный регулятор  $K_p$ .

В Python существует функция `control.lqr`, однако использование `control.lqr` возможно при условии установки модуля `slycot`. В качестве альтернативы, была использована библиотека функций линейной алгебры `scipy.linalg`.

```
from numpy import*
from scipy.linalg import*
def lqr(A,B,Q,R):
    S = matrix(solve_continuous_are(A, B, Q, R))
    Kp = matrix(inv(R)*(B.T*S))
    P = eig(A-B*K)[0]
    return Kp, S, P
```

Листинг 1. Реализация функции *lqr* на языке Python

Функция `solve_continuous_are` предназначена для решения матричного нелинейного алгебраического уравнения Риккати (5). Для того чтобы вернуть матрицу из объекта, подобного массиву, или из строки используется функция `matrix`. При прямом обращении матрицы используется функция `inv`. Для вычисления собственных значений квадратной матрицы существует функция `eig`.

Несмотря на то, что в Python и MATLAB синтаксис использования функции `lqr` практически не отличается, нас будет интересовать точность вычислений.

В качестве примера рассмотрен синтез *LQR*-регулятора для электромеханического объекта [9], заданного в пространстве состояний соответствующими матрицами *A*, *B*, *C* и *D*

$$A = \begin{bmatrix} -100 & 0 & 0 \\ 143.678 & -16.667 & -195.402 \\ 0 & 1.046 & 0 \end{bmatrix}; B = \begin{bmatrix} 2300 \\ 0 \\ 0 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; D = 0.$$

В роли переменных состояния выступают:  $x_1$  – напряжение преобразователя,  $x_2$  – ток двигателя,  $x_3$  – угловая скорость,  $c^{-1}$ . Данная система называется тиристорным преобразователем-двигателем постоянного

тока независимого возбуждения (ТП-ДПТ НВ):  $P_{\text{ном}} = 3$  кВт;  $U_{\text{ном}} = 220$  В;  $I_{\text{ном}} = 147$  А;  $\omega_0 = 169$  с<sup>-1</sup>. Вектор начальных состояний выглядит следующим образом

$$x_0 = [220, 147, 169].$$

Принимаем значения элементов матрицы  $R$  равными 84, а для диагональной матрицы  $Q - 0.01$

$$R = [84], Q = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}.$$

В языке Python функция *lqr* будет возвращать кортеж. После того как значение  $K_p$  получено, необходимо вычислить евклидову норму, воспользовавшись функцией *norm*, которая входит в библиотеку *scipy.linalg*

```
[Kp, S, P]=lqr(A, B, Q, R)
```

```
nm = norm(Kp)
```

Листинг 2. Синтез *LQR*-регулятора и вычисление евклидовой нормы матрицы  $K_p$  в языке Python

В системе MATLAB мы будем использовать библиотечную функцию *lqr* для синтеза *LQR*-регулятора и *norm* для вычисления евклидовой нормы.

```
[Kp, S, P]=lqr(A, B, Q, R, 0);
```

```
nm = norm(Kp);
```

Листинг 3. Синтез *LQR*-регулятора и вычисление евклидовой нормы матрицы  $K_p$  в системе MATLAB

Вычислим значение евклидовой нормы матрицы  $K_p$  для нескольких значений матрицы  $R$  в MATLAB и Python и проанализируем полученные результаты.

Таблица 1. Значение евклидовой нормы матрицы оптимального регулятора

№ п/п	R	$\ K_p\ _2$	
		MATLAB	Python
1	65	0.016918102016306134	0.01691810201630633
2	77	0.015185331800310582	0.015185331800310634
3	84	0.014347961762431749	0.014347961762431841
4	88	0.013914547593136584	0.013914547593136446
5	106	0.01227758579374554	0.012277585793745266

Как мы видим, имеются отличия в расчете евклидовой нормы  $K_p$ , начиная с 16 знака после десятичной точки, что никак не влияет на управление системы.

Для значений матрицы  $R$ , приведенных в таблице 1, произведем аналитический контроль решения задачи оптимальной стабилизации, используя выражение (6). Затем, для этих же матриц  $R$ , рассчитаем относительную погрешность по евклидовой норме и сравним полученные результаты. Общая формула расчета относительной погрешности, для MATLAB и Python, выглядит следующим образом:

$$\text{abs}(\text{norm}(R) - \text{norm}(R_o) / \text{norm}(R)) * 100,$$

где  $R_o$  – левая часть выражения (6).

Таблица 2. Относительная погрешность по евклидовой норме матрицы  $R$ , определённая на основе соотношения оптимальности

№ п/п	R	$\delta, \%$	
		MATLAB	Python
1	65	3.7741052513623918	3.7741052513624798
2	77	3.6004347160218524	3.6004347160218892
3	84	3.5057112101626218	3.5057112101626724
4	88	3.4536853027339638	3.453685302733915
5	106	3.2371782939259006	3.2371782939257665

Проанализировав результаты в таблице 2, можно сделать вывод о том, что

погрешность при аналитическом контроле в MATLAB и Python практически одинаковая. Отличия наблюдаются лишь в 13 знаке после десятичной точки, что является несущественным.

### **Выводы**

На основании вышеизложенного можно сделать следующие выводы:

1. Использование пользовательской функции *lqr* в языке Python позволяет достичь нам оптимального управления, практически совпадающего с результатами при использовании системы MATLAB.

2. Погрешность при аналитическом контроле решения задачи оптимальной стабилизации в MATLAB и Python имеет разницу в 13 знаке числа, после десятичной точки.

Таким образом, при изучении соответствующего раздела теории управления, имеется возможность отказаться от лицензионной программы MATLAB и использовать язык Python.

### **Библиографический список:**

1. Мишин П. А., Немчинова П. А. АНАЛИТИЧЕСКИЙ КОНТРОЛЬ РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОЙ СТАБИЛИЗАЦИИ СТАЦИОНАРНОГО ОБЪЕКТА С ВЕКТОРНЫМ УПРАВЛЕНИЕМ //E-Scio. – 2022. – №. 4 (67). – С. 231-238.

2. Мишин П. А., Немчинова П. А. АНАЛИТИЧЕСКИЙ КОНТРОЛЬ РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОЙ СТАБИЛИЗАЦИИ СТАЦИОНАРНОГО ОБЪЕКТА С ВЕКТОРНЫМ УПРАВЛЕНИЕМ //МАТЕМАТИКА И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ. – 2022. – С. 377-378.

3. Афонин В. В. Аналитический контроль решения задачи оптимальной стабилизации стационарного объекта со скалярным управлением // Вестник Мордовского университета. – 1998. – №3-4. – С. 122-123.

4. Афонин В. В., Мурюмин С. М. Обратные задачи оптимальной стабилизации со скалярным управлением // Инженерные технологии и системы.

– 2017. – Т. 27, №. 4. – С. 504-517.

5. Афонин В. В., Мурюмин С. М. Соотношения оптимальности в линейно-квадратичной задаче управления // Журнал Средневолжского математического общества. – 2014. – Т. 16, № 2. – С. 118-120.

6. Малкин И. Г. Теория устойчивости движения. – М.: Наука, 1966. – 532 с.

7. Бесекерский В. А., Попов Е. П. Теория систем автоматического регулирования. – Спб.: Профессия, 2003. – 752 с.

8. Petersen I. R., Hollot C. V. A Riccati equation approach to the stabilization of uncertain linear systems // Automatica. – 1986. – Т. 22, №. 4. – С. 397-411.

9. Кибардин В. В., Ковалева О. А., Язев В. Н. Критерии оптимального управления и lqr-оптимизация в электроприводе // Вестник Красноярского государственного аграрного университета. – 2015. – №. 12. – С. 61-73.