УДК 517.977.1 EDN: YKJZQV

ПОСТРОЕНИЕ УПРАВЛЕНИЯ НА ОСНОВЕ АЛГОРИТМА ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Д.Д. Девяткин[1] devyatkindd@student.bmstu.ru
А.В. Юрченков[2] alexander.yurchenkov@yandex.ru

[1] **МГТУ им. Н.Э. Баумана, Москва, Российская Федерация**
[2] **ИПУ РАН, Москва, Российская Федерация**

Аннотация

Работа посвящена построению управления на основе алгоритма обучения с подкреплением для непрерывной системы и его сравнению с классическим методом дискретно-непрерывного управления. Дискретно-непрерывное управление расширяет классические методы, позволяя изменять управляющий сигнал внутри интервала дискретизации. Это повышает точность, однако требует знания параметров системы, что ограничивает его применение в условиях неопределенности. В качестве более современного и адаптивного метода рассмотрен подход на основе данных с использованием алгоритма off-policy Q-learning, который не требует априорной идентификации модели и знания точных параметров управляемого объекта, а обучается непосредственно на измеренных данных. Показано, что последовательность коэффициентов усиления имеет передел, при этом каждый элемент последовательности будет стабилизировать замкнутую систему. Разработанный алгоритм управления обладает свойством робастности. Проведено численное моделирование для системы двойного интегратора, демонстрирующее эффективность обоих методов, а также эксперимент с воздействием шума на модель. Выполнены анализ и сравнение обоих алгоритмов. Практическая часть реализована на языке программирования Python с использованием общедоступных библиотек NumPy, SciPy, Matplotlib и Seaborn

Ключевые слова

Дискретно-непрерывное управление, Q-learning, обучение с подкреплением, линейные системы

Поступила 12.05.2025
Принята 29.06.2025
© Автор(ы), 2026

Введение. В современной теории управления решение задачи построения точных математических моделей динамических систем представляет собой сложную и трудоемкую проблему. В большинстве случаев для

упрощения анализа и синтеза систем управления допускаются различные приближения, включая пренебрежение определенными силами и нелинейностями. Однако такие упрощения могут приводить к значительным отклонениям динамики построенной модели от реального поведения объекта управления, что в свою очередь снижает эффективность разработанных алгоритмов управления и даже может привести к неустойчивости замкнутой управлением системы.

При рассмотрении задач управления в дискретном времени можно выделить два подхода. Первый — управление с нулевым порядком задержки (zero-order hold (ZOH)), который удерживает управляющее воздействие постоянным в течение каждого интервала выборки. Второй — дискретно-непрерывное управление (discrete-continuous (D/C)), который позволяет изменять управление внутри одного интервала выборки. Оба подхода улучшают качество управления и расширяют возможности управления системой [1]. Однако в последнее время все большее внимание уделяется методам управления, основанным на данных, которые позволяют минимизировать влияние погрешностей моделирования и адаптироваться к неопределенностям [2, 3]. Здесь под данными понимаются измеренные данные о состоянии системы и входных управляющих воздействиях. Эти данные о системе используются для построения стабилизирующего управляющего воздействия.

В последние годы значительно преобладают новые методы управления, среди которых выделяется подход прямого управления на основе данных (direct data-driven control) [4–6]. Семейство таких методов предполагает проектирование стабилизирующего регулятора непосредственно на основе экспериментальных данных без необходимости явной идентификации модели системы [7]. Такой подход к решению задачи открывает новые возможности для построения адаптивных и робастных систем управления, что особенно актуально в условиях неопределенности и сложных динамических возмущений.

Настоящая работа посвящена сравнению метода управления динамическими системами на основе данных с методом дискретно-непрерывного управления. Рассмотрены основные теоретические аспекты подходов, их преимущества и ограничения, а также практическая реализация на примере конкретных задач управления.

Постановка задачи. Рассмотрим линейную динамическую систему, описываемую уравнением состояния

$$x_{k+1} = Ax_k + Bu_k, \tag{1}$$

где $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ — матрицы системы; $x_k \in \mathbb{R}^{\mathrm{n}}$ — вектор состояния системы в момент времени k; $u_k \in \mathbb{R}^{\mathrm{m}}$ — вектор управляющего воздействия.

В работе рассмотрены два подхода к построению управления: 1) дискретно-непрерывное (D/C) управление, в котором управляющий сигнал может изменяться внутри интервала дискретизации в соответствии с заданным законом; 2) управление на основе алгоритма off-policy Q-learning, который позволяет обучать регулятор непосредственно на основе накопленных данных без необходимости явной идентификации системы.

Цель работы — показать эффективность управления, построенного с использованием алгоритма Q-learning над методом дискретно-непрерывного управления, доказать свойство стабилизируемости замкнутой системы на любом шаге вычисления оптимального коэффициента усиления и ее устойчивости к изменению параметров. Для этого проведено численное моделирование, включающее в себя оценку качества стабилизации, влияния шумов на динамику системы и сравнительный анализ норм управляющих воздействий.

Дискретно-непрерывное управление. Методы дискретного управления традиционно опираются на концепцию, при которой управляющий сигнал остается постоянным на протяжении каждого интервала дискретизации. Однако метод дискретно-непрерывного управления расширяет эту концепцию, что позволяет управляющему сигналу изменяться внутри каждого интервала по заранее заданному закону. Это приводит к улучшению характеристик системы по сравнению с классическим дискретным управлением.

Рассмотрим линейную динамическую систему, описываемую уравнением состояния

$$\dot{x} = Ax + Bu,\ t_0 = 0,\ \ x(0) = x_0, \tag{2}$$

где $x \in \mathbb{R}^n$; $u \in \mathbb{R}^m$; $A \in \mathbb{R}^{n \times n}$; $B \in \mathbb{R}^{n \times m}$. Если управление u можно представить в виде взвешенной суммы базисных функций, которые удовлетворяют некоторому дифференциальному уравнению, тогда уравнение из (1) в дискретном времени имеет вид:

$$x((k+1)T) = \widetilde{A}x(kT) + \overline{BH}\nu(kT),$$

где $x((k+1)T)$ — вектор состояния системы в момент времени $(k+1)T$; T — период дискретизации; $\widetilde{A} = e^{AT}$; $\nu = \left[\nu_1, \nu_2, \ldots, \nu_s\right]$ — вектор состояний управляющего D/C-сигнала $u(t)$;

$$\overline{BH} = \int_{kT}^{(k+1)T} e^{A(t-\tau)} B\overline{H}(\tau) e^{\overline{D}(\tau - T)} d\tau,$$

$\overline{H} \in \mathbb{R}^{r \times s}$, $\overline{D} \in \mathbb{R}^{s \times s}$ — известные матрицы [8].

На каждом интервале дискретизации управление $u(t)$ изменяется во времени в соответствии с выражениями

$$u(t) = \overline{H}\nu(t), \quad kT \le t < (k+1)T, \tag{3}$$

$$\nu(t) = e^{\overline{D}(t-kT)}\nu(kT). \tag{4}$$

Рассмотрим функционал качества, который используется для построения линейно-квадратичного регулятора в непрерывном времени:

$$J(x,u) = \frac{1}{2}\left(x^{\mathrm{T}}(NT)Sx(NT) + \int_0^{NT} x^{\mathrm{T}}Qx + u^{\mathrm{T}}Ru \; dt \right), \tag{5}$$

где $S = S^{\mathrm{T}}$, $Q = Q^{\mathrm{T}}$, $R = R^{\mathrm{T}}$, $S > 0$, $Q > 0$, $R > 0$, причем неравенства понимаются в смысле положительной определенности матрицы; N — число интервалов дискретизации. Поскольку рассматривается задача поиска D/C-управления на дискретном времени, функционал (5) принимает вид:

$$J_{D/C} = \frac{1}{2}\left(x^{\mathrm{T}}(NT)\, Sx(NT) + \sum_{k=0}^{N-1}\left[x^{\mathrm{T}}(kT)Qx(kT) + \int_{kT}^{(k+1)T} u^{\mathrm{T}}Ru\, dt \right]\right). \tag{6}$$

Вынесем слагаемое с управлением из-под знака суммы в (6) и подставим в него (3) и (4), в результате получим:

$$\int_0^{NT} u^{\mathrm{T}}Ru\, dt = \sum_{k=0}^{N-1} \nu^{\mathrm{T}}(kT)\mathbf{R}\nu(kT),$$

где

$$\mathbf{R} = \frac{1}{2}\int_{kT}^{(k+1)T} e^{\overline{D}^{\mathrm{T}}(\tau - kT)} \overline{H}^{\mathrm{T}} R \overline{H} e^{\overline{D}(\tau - kT)} d\tau.$$

Рассмотрим случай, когда $u(t) = \mathrm{const}$ на каждом интервале $[kT, (k+1)T)$, где $k = 0, \ldots, N-1$. Следовательно, $\overline{D} = 0$, $\overline{H} = I$ и $\mathbf{R} = R$.

Теперь необходимо найти оптимальную последовательность $\nu^* = \left[\nu(0), \nu(T), \ldots, \nu(NT)\right]$, которая минимизирует функционал $J_{D/C}$. Для текущей задачи поиска управления оптимальная последовательность выражается [9, 10] как

$$\nu^*(kT) = -\left[R + \overline{BH}^{\mathrm{T}} \overline{P}(k+1) \overline{BH} \right]^{-1} \overline{BH}^{\mathrm{T}} \overline{P}(k+1) \widetilde{A}(kT), \qquad (7)$$

где $\overline{P}$ — величина, удовлетворяющая разностному уравнению Риккати [11]

$$\overline{P}(kT) = Q + \widetilde{A} \overline{P}((k+1)T) \cdot \left[I + \overline{BH} R^{-1} \overline{BH}^{\mathrm{T}} \overline{P}((k+1)T) \right]^{-1} \widetilde{A},$$

$$\overline{P}(NT) = S, \ k = 0, \ldots, N-1.$$

Такой метод управления показывает эффективность, однако требует знания точных параметров системы, таких как матрицы A, B, что может быть затруднено в реальных приложениях.

Алгоритм off-policy Q-learning (Q-обучение). Одним из перспективных подходов в области управления на основе данных является использование методов машинного обучения, которые позволяют обучать управляющие алгоритмы непосредственно на основе накопленных данных о поведении системы. В частности, методы обучения с подкреплением (reinforcement learning (RL)) [12] продемонстрировали высокую эффективность в управлении сложными нелинейными объектами без необходимости точного знания их математической модели. Обучение с подкреплением представляет собой набор алгоритмов, которые в последние годы активно используются как инструмент управления для стабилизации динамических систем. Например, алгоритм Q-learning (далее Q-обучение) [13] является качественным методом обучения с подкреплением ввиду интуитивно понятного формирования и свойства сходимости. Кроме того, алгоритм Q-обучения является off-policy-методом, т. е. текущая оценка оптимального регулятора никогда не применяется к системе в процессе обучения. Вместо этого агент обучается на данных, собранных при использовании произвольного возбуждающего ввода (persistently exciting input). Алгоритм off-policy Q-обучения основан на применении фундаментальной леммы Виллемса, которая позволяет представить систему в виде линейной комбинации данных, собранных при воздействии устойчиво возбуждающего входного сигнала.

Прежде чем описывать алгоритм поиска управления, введем важные обозначения и понятия. Рассмотрим дискретную линейную систему вида (1). Для последовательности состояний длиной N введем понятие матрицы Ганкеля глубины L:

$$H_L(x_{[0,\,N-1]}) = \begin{bmatrix} x_0 & x_1 & \dots & x_{N-L} \\ x_1 & x_2 & \cdots & x_{N-L-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{L-1} & x_L & \dots & x_{N-1} \end{bmatrix}.$$

Определение 1. *Входной сигнал* $\{u_k\}_{k=0}^{N-1}$ *называют постоянно существующим порядка L, если* $N \geq (m+1)L-1$ *и* $\operatorname{rank} H_L(u_{[0,N-1]}) = mL$.

Далее представлены основные положения фундаментальной леммы Виллемса, которая состоит из двух утверждений и является теоретической основой для разработки алгоритма управления.

Лемма 1 [14]. *Пусть входной сигнал* $\{u_k\}_{k=0}^{N-1}$ *является постоянно существующим порядка* $n+L$ *для системы (1), тогда*

$$\operatorname{rank}\left(\begin{bmatrix} H_1(x_{[0,N-L]}) \\ H_L(x_{[0,N-1]}) \end{bmatrix}\right) = Lm+n.$$

Теорема 1 [14]. *Последовательность* $\{x_k\}_{k=0}^{N-1}$ *при постоянно существующей последовательности* $\{u_k\}_{k=0}^{N-1}$ *порядка* $n+L$ *является траекторией системы (1) тогда и только тогда, когда* $\exists \alpha \in \mathbb{R}^{N-L+1}$:

$$\begin{bmatrix} H_L(x_{[0,N-1]}) \\ H_L(u_{[0,N-1]}) \end{bmatrix} = \alpha \begin{bmatrix} \overline{x} \\ \overline{u} \end{bmatrix},$$

где $\overline{x} = \left[\overline{x}_0^{\mathrm{T}}, \dots, \overline{x}_{L-1}^{\mathrm{T}}\right]^{\mathrm{T}}$; $\overline{u} = \left[\overline{u}_0^{\mathrm{T}}, \dots, \overline{u}_{L-1}^{\mathrm{T}}\right]^{\mathrm{T}}$.

Лемма 1 и теорема 1 показывают, что каждую возможную траекторию $\left\{\overline{u_k}, \overline{x_k}\right\}_{k=0}^{N-1}$ управляемой системы (1) можно выразить как линейную комбинацию одной устойчиво существующей траектории, полученной из системы. Таким образом, эти результаты позволяют представить управляемую линейную систему на основе данных.

Перейдем к поиску управляющего сигнала с использованием алгоритма Q-обучения. Отличие текущего алгоритма от классического [15] заключается в том, что вследствие леммы Виллемса алгоритм становится off-policy-методом, т. е. нет необходимости применять текущую оценку оптимальности входного воздействия. В общем случае цель алгоритма Q-обучения — определить решение задачи линейно-квадратичного регу-

лятора (LQR) без какого-либо знания параметров модели. Здесь задача управления LQR формулируется определением текущего вклада в функционал качества входа u_k в состоянии x_k как:

$$c(x_k, u_k) = x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k, \tag{8}$$

где $Q \in \mathbb{R}^{n \times n}$; $R \in \mathbb{R}^{m \times m}$; $Q, R > 0$. Итоговое значение на бесконечном горизонте:

$$J(x, u) = \sum_{k=0}^{+\infty} c(x_k, u_k). \tag{9}$$

Основная цель — определить последовательность $\{u_k\}_{k=0}^{N-1}$, которая минимизирует (9). Управление будем искать в виде $u_k = -K^* x_k$, где K^* — статический коэффициент усиления, который интерпретируется как оптимальная политика в терминах задачи обучения с подкреплением. Далее оценим функционал (9), начиная с момента времени k, с использованием коэффициента K:

$$V^K(x_k) = \sum_{t=k}^{+\infty} (x_t^{\mathrm{T}} Q x_t + u_t^{\mathrm{T}} R u_t)\big|_{u_t = -Kx_t} = \sum_{t=k}^{+\infty} x_t^{\mathrm{T}} (Q + K^{\mathrm{T}} R K) x_t.$$

Определена функция стоимости V, конкретный вид функции при выборе коэффициента K обозначен как V^K. По аналогии определяется Q-функция для коэффициента K как значение функционала при произвольном управлении u_k в момент времени k, затем $u_t = -Kx_t$, начиная с момента $t = k+1$:

$$\mathbb{Q}^K(x_k, u_k) = c(x_k, u_k) + V^K(x_{k+1}). \tag{10}$$

Отметим, что Q-функция и функция стоимости V связаны соотношением

$$\mathbb{Q}^K(x_k, -Kx_k) \triangleq V^K(x_k). \tag{11}$$

Для линейных систем вида (1) функцию стоимости V можно выразить как

$$V^K(x_k) = x_k^{\mathrm{T}} P^K x_k, \tag{12}$$

где $P^K > 0$ — решение дискретного уравнение Ляпунова.

Для Q-функции в момент времени k и $k + 1$ существует связь, которую можно получить из (10) и (11): $\mathbb{Q}^K(x_k, u_k) = c(x_k, u_k) + \mathbb{Q}^K(x_{k+1}, -Kx_{k+1})$. Объединяя (1), (8), (10)–(12), получаем

$$\mathbb{Q}^K(x_k,u_k)=c(x_k,u_k)+V^K(x_{k+1})=$$
$$=x_k^{\mathrm{T}}Qx_k+u_k^{\mathrm{T}}Ru_k+V^K(x_{k+1})=$$
$$=x_k^{\mathrm{T}}Qx_k+u_k^{\mathrm{T}}Ru_k+(Ax_k+Bu_k)^{\mathrm{T}}(P^K)^{\mathrm{T}}(Ax_k+Bu_k)=z_k^{\mathrm{T}}\Theta^K z_k, \qquad (13)$$

где

$$z_k=\left[x_k^{\mathrm{T}},u_k^{\mathrm{T}}\right];$$

$$\Theta^K=\begin{bmatrix}Q+A^{\mathrm{T}}P^KA & A^{\mathrm{T}}P^KB\\ B^{\mathrm{T}}P^KB & R+B^{\mathrm{T}}P^KB\end{bmatrix}=\begin{bmatrix}\Theta_{xx} & \Theta_{ux}^{\mathrm{T}}\\ \Theta_{ux} & \Theta_{uu}\end{bmatrix}.$$

С использованием соотношения (13) уравнение (12) имеет вид

$$z_k^{\mathrm{T}}\Theta^K z_k=z_k^{\mathrm{T}}\overline{Q}z_k+\zeta_{k+1}^{\mathrm{T}}\Theta^K\zeta_{k+1}, \qquad (14)$$

где $\overline{Q}=\mathrm{diag}(Q,R)$; $\zeta_{k+1}=\left[x_{k+1}^{\mathrm{T}},u_{k+1}^{\mathrm{T}}\right]^{\mathrm{T}}$ — расширенное состояние.

Определим алгоритм поиска управления. По определению Q-функция оценивает, насколько «хорошим» является выполнение действия u_k в состоянии x_k. С точки зрения будущих результатов это делается через призму минимизации затрат, поэтому оптимальное управление находится минимизацией Q-функции: $u_k^*=\arg\min\mathbb{Q}^K(x_k,u_k)$, где u_k^* — оптимальное управление. Найдем управление, при котором достигается минимум функционала $\mathbb{Q}^k$, с помощью поиска экстремума через производную:

$$u_k^*=-\Theta_{xx}^{-1}\Theta_{ux}x_k, \qquad (15)$$

где $\Theta_{xx}=Q+A^{\mathrm{T}}P^KA$; $\Theta_{ux}=B^{\mathrm{T}}P^KB$.

На основе уравнений (14) и (15) можно разработать off-policy итерационный алгоритм, предназначенный для решения задачи LQR, который строится только на измеренных данных системы (1):

procedure

1. Собрать выборку $\left\{\overline{u_k},\overline{x_k}\right\}_{k=0}^{N-1}$, где $N\geq m(n+1)+n$ и данные сгенерированы произвольным воздействием порядка $n+1$.

2. Собрать $m+n$ векторов $z_{k_j}=\left[x_{k_j}^{\mathrm{T}},u_{k_j}^{\mathrm{T}}\right]^{\mathrm{T}}$, $j=1,\ldots,m+n$, которые являются линейно независимыми.

3. Инициализировать итератор $i=0$, ε малым числом и коэффициент K^0 так, что $A+BK^0$ была устойчива.

4. Повторять пока $K^{i+1} - K^i > \varepsilon$:

a. Решить систему уравнений относительно Θ^{i+1}:

$$z_{k_j}^{\mathrm{T}} \Theta^{i+1} z_{k_j} = z_{k_j}^{\mathrm{T}} \overline{Q} z_{k_j} + \zeta_{i,k_j+1}^{\mathrm{T}} \Theta^{i+1} \zeta_{i,k_j+1},$$

где $\zeta_{i,k+1} = \left[x_{k+1}^{\mathrm{T}}, -(K^i x_{k+1})^{\mathrm{T}} \right]^{\mathrm{T}}$.

b. Обновить коэффициент усиления $K^{i+1} = \left(\Theta_{uu}^{i+1} \right)^{-1} \Theta_{ux}^{i+1}$.

c. Установить значение итератора $i = i+1$ и перейти к шагу 4.

end procedure

Такой алгоритм поиска управления показывает универсальность, так как не требует знаний о параметрах системы в силу того, что управление строится по измеренным данным.

Докажем свойства сходимости алгоритма. Продемонстрируем, что коэффициент усиления K на каждой итерации является стабилизирующим.

Теорема 2. *Если на i-й итерации коэффициент усиления K^i стабилизирует систему (1), то K^{i+1} также стабилизирует систему (1).*

◄ Предположим, что коэффициент усиления K^i стабилизирует систему (1). Докажем, что коэффициент K^{i+1} также стабилизирует систему (1). Рассмотрим уравнение (14) на i-й итерации, которое имеет единственное решение для итерации $i+1$ [16], что гарантирует вычисление K^{i+1}, $\mathbb{Q}^{K_i}(x,u) = z_k^{\mathrm{T}} \Theta^{i+1} z_k$ и означает

$$\zeta_{i+1,k}^{\mathrm{T}} \Theta^{i+1} \zeta_{i+1,k} \le z_k^{\mathrm{T}} \Theta^{i+1} z_k, \ \forall u_k \colon \zeta_{i+1,k} = z_k \big|_{u_k = -K^{i+1} x_k}.$$

В частности, $\zeta_{i+1,k+1}^{\mathrm{T}} \Theta^{i+1} \zeta_{i+1,k+1} \le \zeta_{i,k+1}^{\mathrm{T}} \Theta^{i+1} \zeta_{i,k+1}$. Из уравнения (14) имеем

$$z_k^{\mathrm{T}} \Theta^{i+1} z_k \ge z_k^{\mathrm{T}} \overline{Q} z_k + \zeta_{i+1,k+1}^{\mathrm{T}} \Theta^{i+1} \zeta_{i+1,\,k+1} =$$
$$= z_k^{\mathrm{T}} \overline{Q} z_k + z_k^{\mathrm{T}} \Phi_{i+1}^{\mathrm{T}} \Theta^{i+1} \Phi_{i+1} z_k,$$

где

$$\Phi_{i+1} = \begin{bmatrix} A & B \\ -K_{i+1}A & -K_{i+1}B \end{bmatrix}.$$

Выражение верно для всех линейно независимых векторов, следовательно, и для всех z_k. Тогда

$$-\overline{Q} \geq \Phi_{i+1}^{\mathrm{T}} \Theta^{i+1} \Phi_{i+1} - \Theta^{i+1}.$$

Известно, что $-\overline{Q} < 0$ и K^i стабилизируют систему по условию, тогда $\Theta^{i+1} > 0$. Из неравенства следует, что все собственные значения Φ_{i+1} находятся внутри единичного круга по критерию Ляпунова. Это означает устойчивость замкнутой системы $A - BK^{i+1}$. ►

Докажем, что алгоритм Q-обучения сходится к оптимальному решению задачи.

Теорема 3. *Если K^i стабилизирует систему на каждой итерации, то*

$$\lim_{i \to \infty} \Theta^i = \Theta^*, \quad \lim_{i \to \infty} K^i = K^*,$$

где K^* — оптимальный коэффициент усиления.

◄ Покажем, что последовательность Θ^i монотонно сходится к Θ^*. При этом $\Theta^{i+1} = \overline{Q} + \Phi_i^{\mathrm{T}} \Theta^{i+1} \Phi_i$. Следовательно, разность текущего и следующего решений можно записать как

$$\Theta^{i+1} - \Theta^i = \Phi_i^{\mathrm{T}} (\Theta^{i+1} - \Theta^i) \Phi_i + \Psi_i^{\mathrm{T}} \Theta^i \Psi_i,$$

где

$$\Psi_i = \begin{bmatrix} 0 & 0 \\ (K^i - K^{i-1})A & (K^i - K^{i-1})B \end{bmatrix}.$$

Далее легко получить выражение

$$\Theta^{i+1} - \Theta^i = \sum_{j=0}^{\infty} (\Phi_i^j)^{\mathrm{T}} \Psi_i^{\mathrm{T}} \Theta^i \Psi_i \Phi_i^j \geq 0,$$

правая часть которого положительно определенная.

Если аналогично выполнить для оптимального решения Θ^*, то $\Theta^i \geq \Theta^{i+1} \geq \Theta^*$, так как правая часть выражения положительно определенная. Очевидно, что Θ^* является неподвижной точкой и такая точка единственная. В соответствии с этим у последовательности $\{\Theta^i\}_{i=0}^{\infty}$ существует предел, равный Θ^*. Коэффициент усиления K^i также имеет предел, вычисляемый по формуле $K^{i+1} = (\Theta_{uu}^i)^{-1} \Theta_{ux}^i$. ►

Докажем, что найденное управление обладает свойством робастности.

Теорема 4. *Пусть* $\{\chi_k\}_{k=0}^{N-1} = \{x_k + \omega_k\}_{k=0}^{N-1}$ *зашумленное состояние системы (1) при постоянно существующем сигнале* $\{u_k\}_{k=0}^{N-1}$ *порядка* $n+1$. *Если шумовой сигнал* $\{\omega_k\}_{k=0}^{N-1}$ *такой, что*

$$\| H_1(\omega_{[0,N-1]}) \|_2 < \sigma_{\min}\left(\begin{bmatrix} H_1(x_{[0,N-1]}) \\ H_1(u_{[0,N-1]}) \end{bmatrix}\right),$$

то

$$\operatorname{rank}\left(\begin{bmatrix} H_1(\chi_{[0,N-1]}) \\ H_1(u_{[0,N-1]}) \end{bmatrix}\right) = m+n,$$

что гарантирует стабилизацию системы (1).

◄ Рассмотрим возмущенную матрицу

$$M + \Delta M = \begin{bmatrix} H_1(x) \\ H_1(u) \end{bmatrix} + \begin{bmatrix} H_1(\omega) \\ 0 \end{bmatrix}.$$

Матрица теряет полный ранг, если ее минимальное сингулярное число становится нулем $\sigma_{\min}(M+\Delta M) = 0$. По неравенству Вейланда $\left|\sigma_{\min}(M+\Delta M) - \sigma_{\min}(M)\right| \le \|\Delta M\|_2$. Следовательно, чтобы матрица не теряла полный ранг, достаточно потребовать $\sigma_{\min}(M+\Delta M) > 0$. Из неравенства Вейланда $\sigma_{\min}(M) - \|\Delta M\|_2 > 0$, что эквивалентно условию

$$\| H_1(\omega) \|_2 < \sigma_{\min}\left(\begin{bmatrix} H_1(x) \\ H_1(u) \end{bmatrix}\right). \text{ ►}$$

Численное моделирование. Для моделирования алгоритмов использован язык программирования Python [17, 18], который является распространенным программным средством для реализации научных вычислений и содержит большое число библиотек и фрэймворков, удобных в использовании и ускоряющих расчеты. При моделировании использованы библиотеки NumPy — для векторизации вычислений [19], SciPy — для численных методов линейной алгебры [20], Matplotlib и Seaborn — для визуализации [21].

В качестве системы, для которой будет проводиться моделирование, рассмотрена модель двойного интегратора

$$\begin{cases} \dot{x}_1 = x_2; \\ \dot{x}_2 = u, \end{cases} \quad x_0 = \begin{bmatrix} 20 \\ 160 \end{bmatrix}. \tag{16}$$

Матрицы системы:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Сначала стабилизируем систему (16) с помощью D/C-управления, для чего получим ее дискретный аналог. В результате это приведет к виду матриц реализации в дискретном времени:

$$\widetilde{A} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \widetilde{B} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}.$$

Далее найдем оптимальную последовательность управлений $\nu^* = [\nu(0), \nu(T), \ldots, \nu(NT)]$, которая минимизирует $J_{D/C}$. Поиск такой последовательности осуществляется с использованием (7). Поиск каждого нового значения $\nu(kT)$ проводится после получения нового состояния. Моделирование выполняется при $T = 5$, $R = 5$ и $Q = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix}$. Визуализация последовательности $\nu^*(t)$ и состояний $x_1(t), x_2(t)$ показана на рис. 1.

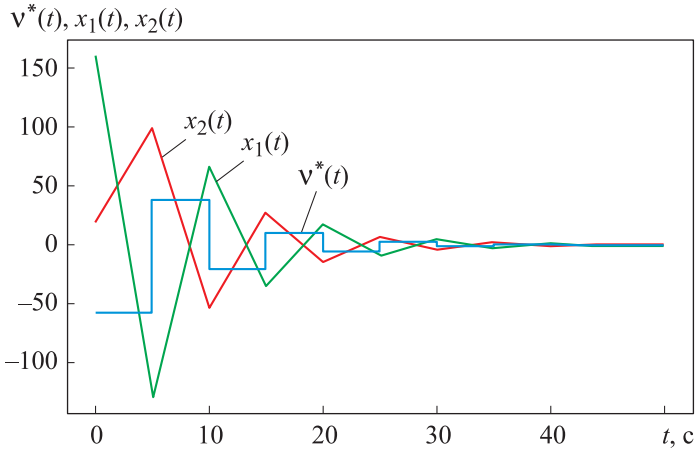

Рис. 1. Стабилизация системы с использованием D/C-управления

Перейдем к построению управления с помощью алгоритма Q-обучения. При моделировании принято $T = 5$, $R = 2$, $Q = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$. Результаты

моделирования (управления $u^*(t)$ и состояния $x_1(t), x_2(t)$) приведены на рис. 2.

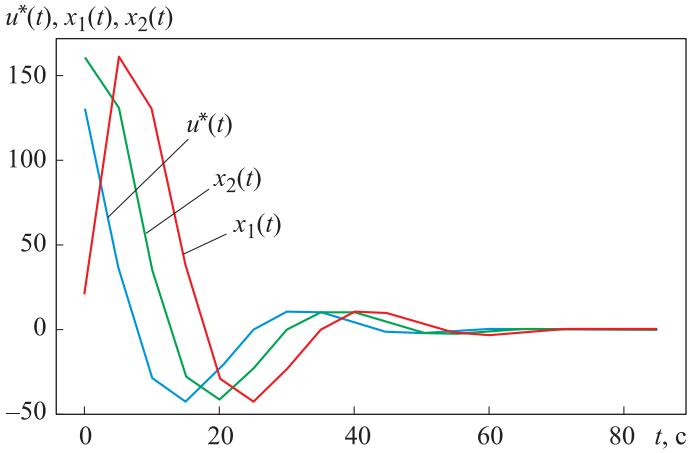

Рис. 2. Стабилизация системы с помощью алгоритма Q-обучения

D/C-управление стабилизирует систему быстрее, чем алгоритм Q-обучения. В первом случае амплитуда меньше, но наблюдается более высокая колебательность. Если сравнивать L2-норму управляющего воздействия, то для D/C-управления она меньше: $\|\nu\|_2 = 73{,}25$, $\|u\|_2 = 142{,}90$.

Рассмотрим случай, когда параметры системы претерпевают малое изменение. Найдем стабилизирующее управление для исходной системы с помощью двух алгоритмов. Далее к матрицам A и B аддитивно добавим шум $\varepsilon \sim \mathcal{N}(0, 0{,}01)$ и будем применять найденное управление к такой системе. Оценим результаты моделирования. Результат моделирования для D/C-управления показан на рис. 3, *а*. Время стабилизации с помощью D/C-управления и колебательность системы увеличились. Значение L2-нормы $\|\nu\|_2 = 153{,}68$.

Для алгоритма Q-обучения результаты моделирования приведены на рис. 3, *б*. Время стабилизация не изменилось. Значения стабилизирующего управления претерпели небольшое изменение. Значение L2-нормы $\|u\|_2 = 148{,}85$.

Согласно результатам моделирования, алгоритм Q-обучения имеет преимущество в том, что нет необходимости знать параметры системы при поиске стабилизирующего управления. Это означает, что появилась возможность представлять систему через данные, избегая этапа явной идентификации модели. Подход на основе D/C-управления строит оптимальное управление для системы с точными параметрами, что позволяет

находить более оптимальное управление. Однако алгоритм Q-обучения более устойчив к малым изменениям параметров системы, а также требует решения меньшего числа уравнений и операций по сравнению с методами на основе линейных матричных неравенств.

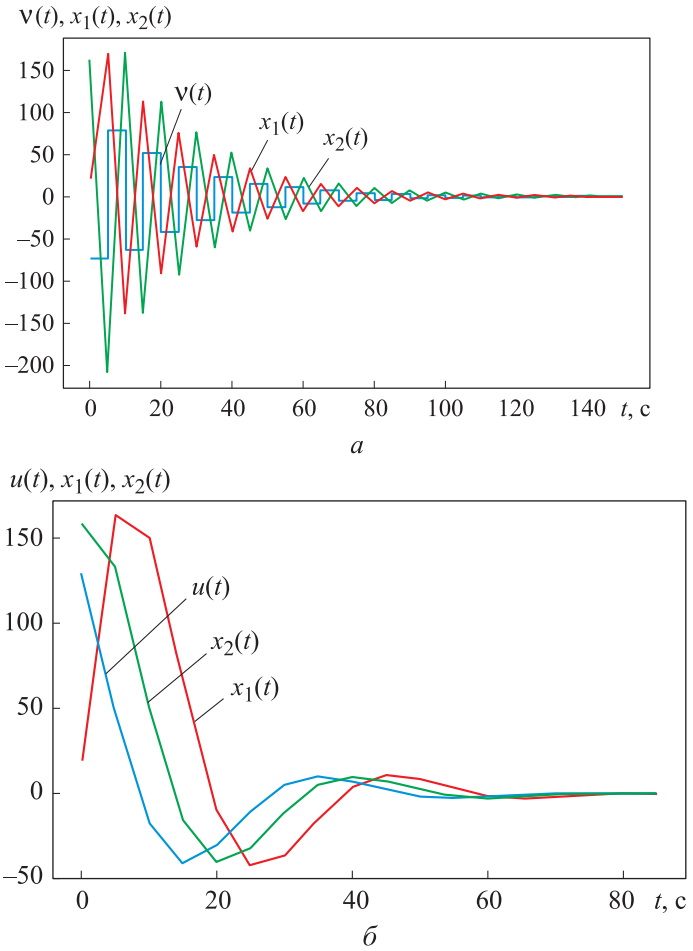

Рис. 3. Стабилизация системы с шумом с помощью D/C-управления (*а*) и алгоритма Q-обучения (*б*)

Несмотря на продемонстрированные преимущества как D/C-управления, так и подхода на основе off-policy Q-обучения, оба метода имеют существенные ограничения, которые важно учитывать при практической реализации. Например, D/C-управление требует численного решения уравнения Риккати на каждом шаге построения LQR, что является трудоемкой задачей. При больших размерностях системы это приводит к высокой вычислительной нагрузке. Сложность алгоритма составляет $O(n^3)$.

Ограничением алгоритма также является наличие матриц A и B, так как во многих прикладных задачах они неизвестны. При переходе к методу Q-обучения следует учитывать требование к объему данных: корректность работы алгоритма напрямую зависит от полноты и качества обучающей выборки. Для получения линейно независимых векторов z_k необходимо собрать достаточно длинную и возбуждающую входную последовательность, что может быть затруднено в условиях ограниченных наблюдений. Сложность алгоритма $O(n^5)$.

Заключение. Проведено сравнение двух подходов к управлению динамическими системами: D/C-управления и off-policy Q-обучения. D/C-управление позволяет улучшить качество управления за счет изменения управляющего сигнала в рамках одного интервала дискретизации и поиска управления на основе вектора состояния, однако требует точного знания параметров системы. В то же время алгоритм off-policy Q-обучения, основанный на методах обучения с подкреплением, позволяет обучаться на данных о состоянии системы и входных управляющих воздействиях без явной идентификации модели, что делает его более универсальным и применимым в условиях неопределенности.

ЛИТЕРАТУРА

[1] Johnson C.D., Abdel-Haleem M. Optimal discrete-continuous control for the linear-quadratic regulator problem. *Proc. 28th Southeastern Symposium on System Theory*, 1996, pp. 184–188. DOI: https://doi.org/10.1109/SSST.1996.493495

[2] Cheng S., Quilodrán-Casas C., Ouala S., et al. Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review. *CAA J. Autom. Sin.*, 2023, vol. 10, iss. 6, pp. 1361–1387. DOI: https://doi.org/10.1109/JAS.2023.123537

[3] Мыльников Л.А., Гергель Н.А., Кычкин А.В. и др. Использование динамических предиктивных моделей для управления техническими системами с инертностью. *Вестник ПНИПУ. Электротехника, информационные технологии, системы управления,* 2018, № 26, с. 77–91. EDN: XUEDGP

[4] Van Waarde H.J., Eising J., Trentelman H.L., et al. Data informativity: a new perspective on data-driven analysis and control. *IEEE Trans. Autom. Control*, 2020, vol. 65, iss. 11, pp. 4753–4768. DOI: https://doi.org/10.1109/TAC.2020.2966717

[5] Berberich J., Romer A., Scherer C., et al. Robust data-driven state-feedback design. *Proc. ACC*, 2020, pp. 1532–1538.
DOI: https://doi.org/10.23919/ACC45564.2020.9147320

[6] Dorfler F., Coulson J., Markovsky I. Bridging direct and indirect data-driven control formulations *via* regularizations and relaxations. *IEEE Trans. Autom. Control*, 2023, vol. 68, iss. 2, pp. 883–897. DOI: https://doi.org/10.1109/TAC.2022.3148374

[7] Yang Y., Guo Z., Xiong H., et al. Data-driven robust control of discrete-time uncertain linear systems *via* off-policy reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 2019, vol. 30, iss. 12, pp. 3735–3747.
DOI: https://doi.org/10.1109/TNNLS.2019.2897814

[8] Johnson C.D. A new discrete-time state model for linear dynamical systems with continuously-varying control/disturbance inputs. *Proc. of the 1994 Southeastern Symposium on System Theory (SSST)*, 1994, pp. 523–527.
DOI: https://doi.org/10.1109/SSST.1994.287821

[9] Ogata K. Discrete-time control systems. Prentice-Hall, 1987.

[10] Kuo B.C. Digital control systems. Oxford Univ. Press, 1992.

[11] Dorato P., Levis A.H. Optimal linear regulators: the discrete-time case. *EEE Trans. Autom. Control,* 1971, vol. 16, iss. 6, pp. 613–620.
DOI: https://doi.org/10.1109/TAC.1971.1099832

[12] Sutton R.S., Barto A.G. Reinforcement learning: an introduction. MIT Press, 2018.

[13] Watkins J., Dayan P. Q-learning. *Mach. Learn.*, 1992, vol. 8, no. 3, pp. 279–292.
DOI: https://doi.org/10.1007/BF00992698

[14] Willems J.C., Rapisarda P., Markovsky I., et al. A note on persistency of excitation. *Syst. Control Lett.*, 2005, vol. 54, iss. 4, pp. 325–329.
DOI: https://doi.org/10.1016/j.sysconle.2004.09.003

[15] Bradtke S.J., Ydstie B.E., Barto A.G. Adaptive linear quadratic control using policy iteration. *Proc. ACC*, 1994, vol. 3, pp. 3475–3479.
DOI: https://doi.org/10.1109/ACC.1994.735224

[16] Lopez V.G., Alsalti M., Muller M.A. Efficient off-policy Q-learning for data-based discrete-time LQR problems. *IEEE Trans. Autom. Control*, 2023, vol. 68, iss. 5, pp. 2922–2933. DOI: https://doi.org/10.1109/TAC.2023.3235967

[17] Любанович Б. Простой Python. СПб., Питер, 2020.

[18] Бэрри П. Изучаем программирование на Python. М., Эксмо, 2020.

[19] Idris I. NumPy beginner's guide. Packt Publ., 2015.

[20] Нуньес-Иглесиас Х., Уолт Ш., Дэшноу Х. Элегантный SciPy. М., ДМК Пресс, 2018.

[21] Абдрахманов М.И. Python. Визуализация данных. Devpractice.ru, 2020.

Девяткин Даниил Дмитриевич — аспирант кафедры «Математическое моделирование» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Юрченков Александр Викторович — канд. физ.-мат. наук, старший научный сотрудник лаборатории № 1 «Динамические информационно-управляющие системы» им. Б.Н. Петрова ИПУ РАН (Российская Федерация, 117997, Москва, ул. Профсоюзная, д. 65).

Просьба ссылаться на эту статью следующим образом:

Девяткин Д.Д., Юрченков А.В. Построение управления на основе алгоритма обучения с подкреплением. *Вестник МГТУ им. Н.Э. Баумана. Сер. Естественные науки*, 2026, № 1 (124), с. 32–50. EDN: YKJZQV

SYNTHESIS A CONTROL SYSTEM BASED ON A REINFORCEMENT LEARNING ALGORITHM

D.D. Devyatkin[1] devyatkindd@student.bmstu.ru
A.V. Yurchenkov[2] alexander.yurchenkov@yandex.ru

[1] **BMSTU, Moscow, Russian Federation**
[2] **ICS RAS, Moscow, Russian Federation**

Abstract

This article is dedicated to developing a control strategy based on a reinforcement learning algorithm for a continuous system and comparing it with the classical method of discrete-continuous control. Discrete-continuous control extends classical methods by allowing the control signal to vary within the sampling interval, which improves accuracy; however, it requires knowledge of system parameters, limiting its applicability under uncertainty. As a more modern and adaptive approach, a data-driven method using the off-policy Q-learning algorithm is considered. This method doesn't require a priori model identification or precise knowledge of the system parameters, as it learns directly from measured data. It is shown that the sequence of gain coefficients converges, and each element of the sequence stabilizes the closed-loop system. The developed control algorithm exhibits robustness. Numerical simulations were carried out for a double integrator system, confirming the effectiveness of both methods. Additionally, an experiment was conducted to evaluate the impact of measurement noise on model performance. A comparative analysis of the two algorithms is presented. The practical implementation was done in Python using open-source libraries such as NumPy, SciPy, Matplotlib and Seaborn

Keywords

Discrete-continuous control, Q-learning, reinforcement learning, linear systems

Received 12.05.2025
Accepted 29.06.2025
© Author(s), 2026

REFERENCES

[1] Johnson C.D., Abdel-Haleem M. Optimal discrete-continuous control for the linear-quadratic regulator problem. *Proc. 28th Southeastern Symposium on System Theory*, 1996, pp. 184–188. DOI: https://doi.org/10.1109/SSST.1996.493495

[2] Cheng S., Quilodrán-Casas C., Ouala S., et al. Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review. *CAA J. Autom. Sin.*, 2023, vol. 10, iss. 6, pp. 1361–1387. DOI: https://doi.org/10.1109/JAS.2023.123537

[3] Mylnikov L.A., Gergel N.A., Kychkin A.V., et al. Dynamic prediction model in control systems of technic processes with inertia. *Vestnik PNIPU. Elektrotekhnika, informatsionnye tekhnologii, sistemy upravleniya* [PNRPU Bulletin. Electrical Engineering, Information Technology, Control Systems], 2018, no. 26, pp. 77–91 (in Russ.).
EDN: XUEDGP

[4] Van Waarde H.J., Eising J., Trentelman H.L., et al. Data informativity: a new perspective on data-driven analysis and control. *IEEE Trans. Autom. Control*, 2020, vol. 65, iss. 11, pp. 4753–4768. DOI: https://doi.org/10.1109/TAC.2020.2966717

[5] Berberich J., Romer A., Scherer C., et al. Robust data-driven state-feedback design. *Proc. ACC*, 2020, pp. 1532–1538.
DOI: https://doi.org/10.23919/ACC45564.2020.9147320

[6] Dorfler F., Coulson J., Markovsky I. Bridging direct and indirect data-driven control formulations *via* regularizations and relaxations. *IEEE Trans. Autom. Control*, 2023, vol. 68, iss. 2, pp. 883–897. DOI: https://doi.org/10.1109/TAC.2022.3148374

[7] Yang Y., Guo Z., Xiong H., et al. Data-driven robust control of discrete-time uncertain linear systems *via* off-policy reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 2019, vol. 30, iss. 12, pp. 3735–3747.
DOI: https://doi.org/10.1109/TNNLS.2019.2897814

[8] Johnson C.D. A new discrete-time state model for linear dynamical systems with continuously-varying control/disturbance inputs. *Proc. of the 1994 Southeastern Symposium on System Theory (SSST),* 1994, pp. 523–527.
DOI: https://doi.org/10.1109/SSST.1994.287821

[9] Ogata K. Discrete-time control systems. Prentice-Hall, 1987.

[10] Kuo B.C. Digital control systems. Oxford Univ. Press, 1992.

[11] Dorato P., Levis A.H. Optimal linear regulators: the discrete-time case. *EEE Trans. Autom. Control,* 1971, vol. 16, iss. 6, pp. 613–620.
DOI: https://doi.org/10.1109/TAC.1971.1099832

[12] Sutton R.S., Barto A.G. Reinforcement learning: an introduction. MIT Press, 2018.

[13] Watkins J., Dayan P. Q-learning. *Mach. Learn.*, 1992, vol. 8, no. 3, pp. 279–292. DOI: https://doi.org/10.1007/BF00992698

[14] Willems J.C., Rapisarda P., Markovsky I., et al. A note on persistency of excitation. *Syst. Control Lett.*, 2005, vol. 54, iss. 4, pp. 325–329.
DOI: https://doi.org/10.1016/j.sysconle.2004.09.003

[15] Bradtke S.J., Ydstie B.E., Barto A.G. Adaptive linear quadratic control using policy iteration. *Proc. ACC*, 1994, vol. 3, pp. 3475–3479.
DOI: https://doi.org/10.1109/ACC.1994.735224

[16] Lopez V.G., Alsalti M., Muller M.A. Efficient off-policy Q-learning for data-based discrete-time LQR problems. *IEEE Trans. Autom. Control*, 2023, vol. 68, iss. 5, pp. 2922–2933. DOI: https://doi.org/10.1109/TAC.2023.3235967

[17] Lubanovic B. Introducing Python. O'Reilly Media, 2019.

[18] Barry P. Head first Python. O'Reilly Media, 2016.

[19] Idris I. NumPy beginner's guide. Packt Publ., 2015.

[20] Nunez-Iglesias J., van der Walt S., Dashnow H. Elegant SciPy. O'Reilly Media, 2017.

[21] Abdrakhmanov M.I. Python. Vizualizatsiya dannykh [Python. Data visualization]. Devpractice.ru Publ., 2020.

Devyatkin D.D. — Post-Graduate Student, Department of Mathematical Simulation, BMSTU (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Yurchenkov A.V. — Cand. Sc. (Phys.-Math.), Senior Researcher, Laboratory No. 1 of B.N. Petrov Dynamic Information and Control Systems, ICS RAS (Profsoyuznaya ul. 65, Moscow, 117997 Russian Federation).

Please cite this article in English as:

Devyatkin D.D., Yurchenkov A.V. Synthesis a control system based on a reinforcement learning algorithm. *Herald of the Bauman Moscow State Technical University, Series Natural Sciences*, 2026, no. 1 (124), pp. 32–50 (in Russ.). EDN: YKJZQV