Моделирование и анализ данных
2026, *16*(1), 125—140.
https://doi.org/10.17759/mda.2026160108
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2026, *16*(1), 125—140.
https://doi.org/10.17759/mda.2026160108
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

ЧИСЛЕННЫЕ МЕТОДЫ | NUMERICAL METHODS

Научная статья | Original paper

УДК 519.6:004.8

Обзор современных методов обучения с подкреплением для решения задач оптимального управления динамическими системами

В.Н. Пановский
ФГБОУ ВО «Московский авиационный институт
(национальный исследовательский университет)»
Москва, Российская Федерация
✉ panovskiy.v@yandex.ru

Резюме

Контекст и актуальность. Задачи синтеза оптимального управления для нелинейных динамических систем с ограничениями и неопределённостями остаются вычислительно сложными, особенно в аэрокосмических приложениях. Обучение с подкреплением рассматривается как практический инструмент построения обратной связи и/или ускорения планирования, когда применение классических методов затруднено. **Цель**. Систематизировать классы алгоритмов для задач оптимального управления и выделить критерии выбора подхода под конкретную постановку. **Гипотеза**. Практическая применимость обеспечивается корректной постановкой и учётом требований к непрерывности управления, данным, безопасности и робастности; наиболее эффективны комбинированные решения. **Методы и материалы**. Выполнен обзор и сравнительный анализ семейств разных алгоритмов обучения с подкреплением. **Результаты**. Для непрерывного управления базовыми остаются actor–critic, а альтернативы повышают выборочную эффективность, но чувствительны к ошибкам модели и покрытия данных. **Выводы**. Наиболее перспективны гибридные архитектуры, сочетающие обучение с подкреплением с базовыми регуляторами и обеспечивающие контролируемое соблюдение ограничений. Выбор метода должен определяться не только качеством, но и безопасностью, робастностью и стоимостью вычислений.

Ключевые слова: обучение с подкреплением, теория управления, динамические системы, кибернетика

Для цитирования: Пановский, В.Н. (2026). Обзор современных методов обучения с подкреплением для решения задач оптимального управления динамическими системами. *Моделирование и анализ данных, 16*(1), 125—140. https://doi.org/10.17759/mda.2026160108

Overview of modern reinforcement learning methods for solving problems of optimal control of dynamical systems

V.N. Panovskiy
Moscow Aviation Institute, Moscow, Russian Federation
✉ panovskiy.v@yandex.ru

Abstract

Context and relevance. The tasks of synthesizing optimal control for nonlinear dynamic systems with constraints and uncertainties remain computationally challenging, especially in aerospace applications. Reinforcement learning is considered a practical tool for building feedback and/or accelerating planning when classical methods are difficult to apply. **Objective**. To systematize classes of algorithms for optimal control tasks and identify criteria for selecting an approach for a specific problem. **Hypothesis**. Practical applicability is ensured by correct formulation and consideration of requirements for control continuity, data, safety, and robustness; combined solutions are most effective. **Methods and materials**. A review and comparative analysis of families of different reinforcement learning algorithms was performed. **Results**. Actor-critic remains the basis for continuous control, while alternatives increase selective efficiency but are sensitive to model and data coverage errors. **Conclusions**. The most promising are hybrid architectures that combine reinforcement learning with basic controllers and ensure controlled compliance with constraints. The choice of method should be determined not only by quality, but also by safety, robustness, and computational cost.

Keywords: reinforcement learning, control theory, dynamical systems, cybernetics

For citation: Panovskiy, V.N. (2026). Overview of modern reinforcement learning methods for solving problems of optimal control of dynamical systems. *Modelling and Data Analysis, 16*(1), 125—140. (In Russ.). https://doi.org/10.17759/mda.2026160108

Введение

В современной теории управления и вычислительной математике достаточно большое внимание уделяется задаче синтеза (и численного построения) оптимального

управления динамическими системами, функционирующими при наличии неопределённостей и неполноты информации. Прикладная значимость подобных постановок особенно высока в задачах авиационной и ракетно-космической техники, где требуется строить управления, обеспечивающие стабилизацию, наведение, минимизацию времени/расхода ресурса и выполнение терминальных ограничений при сложной динамике и жёстких ограничениях на исполнительные органы. На этом фоне заметный интерес в последние годы вызывает применение методов обучения с подкреплением (reinforcement learning, RL) как альтернативы (или дополнения) к классическим подходам оптимального управления, прежде всего в ситуациях, когда математическая модель известна неполно, либо вычислительная стоимость решения слишком высока.

Обучение с подкреплением представляет собой класс методов машинного обучения, в которых агент, взаимодействуя со средой, последовательно выбирает действия и по наблюдаемым состояниям и скалярному сигналу вознаграждения обучается стратегии (политике) поведения, максимизирующей математическое ожидание накопленного (дисконтированного либо конечно горизонтного) выигрыша. Каноническая формализация опирается на марковский процесс принятия решений и тесно связана с идеями динамического программирования Беллмана, где оптимальная политика выражается через оптимальную функцию ценности (Bellman, 1957). Исторически становление RL как самостоятельного направления связывают с работами 1980-х годов по обучающемуся управлению и адаптивным элементам (Barto–Sutton–Anderson), далее — с развитием методов обучения по значениям (в частности, Q-learning) и градиентных методов поиска политики (семейство REINFORCE) (Barto, Sutton, Anderson, 1983). Новый этап начался с «глубокого» обучения с подкреплением, когда аппроксимация функций ценности/политики нейросетями позволила перейти к высокоразмерным наблюдениям и сложным нелинейным объектам; характерным ориентиром здесь стала работа по DQN, показавшая возможность обучения управлению непосредственно по сенсорным данным (Mnih et al., 2015).

Для задач непрерывного управления (типичных для динамических систем) в качестве де-факто стандартов вычислительного эксперимента и практической настройки часто рассматриваются методы семейства actor–critic: TRPO (Trust Region Policy Optimization) и PPO (Proximal Policy Optimization) как устойчивые on-policy схемы оптимизации политики и SAC (Soft Actor Critic) / TD3 (Twin Delayed DDPG) / DDPG (Deep Deterministic Policy Gradients) как off-policy методы, обеспечивающие более высокую выборочную эффективность на непрерывных действиях (Schulman et al., 2015). При этом значительная часть прикладных постановок задач оптимального управления естественным образом является ограниченной (по состояниям, управлениям, ресурсам, безопасным областям), что стимулировало развитие constrained RL (например, подход CPO (Constrained Policy Optimization) и дальнейшие вариации), ориентированных на явное соблюдение ограничений в процессе обучения и применения политики (Achiam et al., 2017). Важно подчеркнуть, что термин «State of the Art» в RL носит прикладной характер: в зависимости от класса задач (on-policy/off-policy, дискретные/непрерывные действия, наличие модели, требования к робастности

и безопасности) «лучшие» методы различаются, однако перечисленные семейства составляют основу современных прикладных решений и служат базой для многочисленных модификаций.

Перенос методов RL на задачи оптимального управления обычно выполняется через следующую концептуальную схему: динамическая система рассматривается как среда, вектор состояния (или наблюдения) формирует вход агента, управление интерпретируется как действие, а функционал качества — как суммарное вознаграждение. Таким образом, задача минимизации функционала оптимального управления приводится к задаче максимизации ожидаемой суммарной награды. При наличии точной модели и возможности генерировать траектории ключевым становится вопрос выбора:

- модель-свободное обучение (политика/ценность напрямую по данным),
- модель-ориентированные методы (обучение/уточнение модели и планирование по ней),
- гибридные схемы, где RL используется для адаптации параметров/стоимостей/приближений в связке с MPC (Model predictive control).

В частности, современная линия работ рассматривает RL как часть общего аппарата приближённого динамического программирования и связывает его с методами MPC и итеративной оптимизации в единой концептуальной рамке (Bertsekas, 2024). Отдельное направление составляют подходы, «встраивающие» физические априорные знания и ограничения (модели, инварианты, законы сохранения) в процесс обучения — как способ повысить выборочную эффективность и переносимость решений на реальный объект.

Практическая привлекательность RL для задач поиска оптимального управления особенно заметна на примерах сложных аэрокосмических постановок, где требуется строить управление в условиях неопределённостей и ограничений. Так, для задач стабилизации спутника и интеллектуального управления системой ориентации предложены различные варианты глубокого RL, демонстрирующие возможность формирования стабилизирующих стратегий при внешних возмущениях и неполной параметрической информации (Ma et al., 2018). Для задач оптимального по времени управления солнечным парусом (включая робастные постановки с неопределённостями оптических параметров и возмущениями) показано применение PPO-подобных алгоритмов для синтеза политики, сопоставляющей оптимальную ориентацию паруса текущему динамическому состоянию (Пантелеев, Пановский, 2016; Bianchi et al., 2025). Подобные результаты иллюстрируют общий тренд: RL используется либо как прямой генератор управления (политика как регулятор), либо как механизм ускорения/аппроксимации решения вложенной задачи оптимизации, возникающей в классических схемах планирования и управления (Bertsekas, 2024).

Одновременно с этим необходимо отметить ряд принципиальных вызовов, которые в значительной степени определяют текущую повестку исследований и практики применения RL к оптимальному управлению:

- для реальных динамических объектов сбор траекторий дорог и ограничен требованиями безопасности; следовательно, «чисто» модель-свободные методы

нередко оказываются неприемлемыми без симуляции, переноса обучения и/или использования модели (Haarnoja et al., 2018),

- в классическом управлении устойчивость и ограничения задаются явно; в RL эти свойства не «появляются автоматически» и требуют специальных постановок (constrained RL, safe exploration, барьерные/штрафные функции, shield-подходы, комбинирование с MPC) (Achiam et al., 2017),
- политика, обученная в модели, может деградировать при несовпадении динамики, шумов и ограничений; практические решения используют доменную рандомизацию, робастные критерии, адаптацию и физически информированные ограничения
- для задач оптимального управления (особенно задач быстродействия с терминальными множествами) некорректная форма награды приводит к «нецелевому» поведению и трудной настройке; это одна из основных прикладных проблем при переходе от функционала к reward-сигналу,
- глубокие RL-алгоритмы чувствительны к гиперпараметрам, распределениям начальных условий и случайности; устойчивые схемы (PPO/SAC и др.) нивелируют это зависимость, но не устраняют её полностью (Schulman et al., 2017).

Несмотря на описанные сложности, направление RL всё прочнее входит кибернетику и показывает отличные результаты, что делает это направление крайне перспективным для дальнейшего изучения. Далее в статье будут рассмотрены основные классы алгоритмов на базе обучения с подкреплением.

Модель-свободные методы по функциям ценности

Модель-свободные методы (где процесс обучения ведётся методом проб и ошибок без явного построения модели динамики) по функциям ценности (value-based) подходы опираются на принцип оптимальности и уравнение Беллмана (Bellman, 1957) и реализуют приближённое динамическое программирование: вместо явного решения уравнения Беллмана строится аппроксимация функции ценности $V(x)$ или функции действия-ценности $Q(x,u)$ после чего управление выбирается по правилу максимума/минимума. В задачах оптимального управления это означает, что при дискретизации времени и (часто) при дискретизации множества управлений U минимизация функционала приводится к вычислению функции ценности («cost-to-go» function) и выбору действия, оптимального в смысле ожидаемого будущего выигрыша (или стоимости).

Классический алгоритм Q-learning уточняет оценку Q по выборкам переходов (x,u,r,x') и в табличной постановке имеет строгие результаты сходимости. Для реальных динамических систем табличная схема быстро становится неприемлемой из-за роста размерности пространства состояний и требований к сетке. Поэтому используются аппроксимации (линейные и нелинейные), а также fitted-схемы, где аппроксиматор Q_θ подгоняется по «целевым» значениям функции Беллмана на батчах (пучках) траекторий (Sutton, Barto, 2018). Существенный практический прогресс связан с глубокими аппроксиматорами и инженерными приёмами стабилизации обучения.

Для прикладного управления показательна линия DQN (Deep Q Network): введение буфера воспроизведения и целевой сети позволяет частично снять неустойчивость, возникающую при сочетании аппроксимации и бутстрэпинга (Mnih et al., 2015). Далее появлялись модификации, повышающие качество и стабильность: Double DQN снижает переоценку действий, а распределительные варианты (distributional RL) моделируют не только математическое ожидание, но и распределение возврата. Однако даже при таких улучшениях ключевым ограничением остаётся необходимость работы с дискретным множеством действий: при переходе к непрерывному управлению требуется либо квантизация U, либо иной класс алгоритмов.

Достоинства:

- off-policy обучение позволяет многократно переиспользовать накопленные данные, что критично при дорогой симуляции,
- решение формируется через Q-функцию, что удобно при наличии дискретных режимов и переключений (гибридные системы),
- при умеренной квантизации управления метод даёт прямой механизм получения регулятора.

Недостатки:

- квантизация управления быстро становится вычислительно неприемлемой и ухудшает точность,
- устойчивость обучения чувствительна к масштабу вознаграждений и распределению стартовых состояний,
- ограничительные условия (на состояние и управление) не соблюдаются автоматически и требуют дополнительных средств контроля.

С практической точки зрения value-based методы целесообразно рассматривать прежде всего в задачах, где действие естественно дискретно (выбор режима, конфигурации, последовательности операций), либо когда допускается ограниченная квантизация управляющего воздействия. В непрерывных задачах оптимального управления они чаще выступают как компонент комбинированных схем, дополняющих непрерывный регулятор дискретным модулем принятия решений.

Градиентные методы оптимизации политики

В отличие от value-based подходов, методы оптимизации политики (policy gradient, on-policy) параметризуют управление непосредственно как $u = \pi_\theta(x)$ (или стохастически $u \sim \pi_\theta(\cdot|x)$) и оптимизируют параметры θ по градиенту ожидаемого суммарного вознаграждения. Отправной точкой служит алгоритм REINFORCE, где градиент цели выражается через логарифмическую производную вероятности действий (Williams, 1992). Для задач оптимального управления эта постановка удобна тем, что работа ведётся сразу с непрерывными управляющими воздействиями (через непрерывные распределения или параметризацию детерминированного регулятора), а ограничения на управление могут быть учтены на уровне параметризации (насыщение, проекция, ограничение дисперсии и т.п.).

Ключевой практической проблемой policy gradient является большая дисперсия оценок градиента и, как следствие, неустойчивость обучения. Типичный выход — использование базисной функции (baseline) и построение оценки преимущества $A^{\pi}(x,u)$, что приводит к семейству actor–critic (Sutton, Barto, 2018). Для снижения дисперсии дополнительно применяют методы многошаговой оценки и сглаживания, например, GAE (Generalized Advantage Estimation), позволяющий управлять компромиссом «смещение–дисперсия» в оценке преимущества. В контексте оптимального управления это особенно важно при длинных горизонтах и при разреженных наградах (терминальные требования), когда «сырая» оценка градиента быстро деградирует.

Среди современных on-policy методов наиболее распространены TRPO и PPO. TRPO определяет шаг обновления как задачу максимизации суррогатной цели при ограничении на KL-расхождение (дивергенция Кульбака-Лейблера) между новой и старой политикой, тем самым контролируя «размер» обновления и снижая риск деградации. PPO заменяет жёсткое ограничение на более простую клиппированную суррогатную функцию, что делает алгоритм существенно проще в реализации и, как правило, достаточно устойчивым на широком классе задач непрерывного управления (Schulman et al., 2017). Для инженерных постановок это означает наличие относительно «надёжной» базовой процедуры, которую можно использовать как для обучения регулятора с нуля, так и для доводки политики, полученной иным способом.

Достоинства:

- естественная работа с непрерывными управлениями и параметрическими ограничениями на u,
- относительно высокая устойчивость современных on-policy схем (особенно PPO) при корректной нормировке сигналов,
- возможность прямого обучения «регулятора» без явного решения уравнений, следующих из условий оптимальности.

Недостатки:

- низкая выборочная эффективность: данные, полученные старой политикой, ограниченно пригодны для обучения новой,
- чувствительность к форме вознаграждения и к распределению стартовых состояний,
- отсутствие встроенных гарантий соблюдения ограничений и устойчивости без специальных модификаций.

В результате on-policy policy gradient методы целесообразно использовать либо при наличии быстрого и достоверного симулятора, либо как этап «дошлифовки» политики, предварительно полученной иным способом (модель-ориентированным планированием или имитационным обучением) (Sutton, Barto, 2018).

Actor–Critic для непрерывного управления

Для большинства задач оптимального управления динамическими системами характерны кусочно-непрерывные управляющие воздействия. Поэтому значительная

часть прикладных исследований опирается на actor–critic методы, где «актор» задаёт политику $u = \pi_\theta(x)$, а «критик» аппроксимирует $Q_\phi(x,u)$ или $V_\phi(x)$ и используется для построения направления улучшения политики (Sutton, Barto, 2018). С помощью θ и ϕ задаются параметры актора и критика соответственно. В off-policy варианте траектории могут собираться не текущей политикой, а некоторым поведением (behavior policy); переходы (x,u,r,x') сохраняются в replay-буфере и многократно переиспользуются при обучении, что повышает выборочную эффективность.

Наиболее известной линией является детерминированный градиент политики: DPG и его глубокая реализация DDPG (Deep Deterministic Policy Gradients), где политика детерминирована, а градиент вычисляется через критик (Lillicrap et al., 2016). Детерминированная политика удобна в непрерывном управлении, однако практическая реализация чувствительна к ошибкам аппроксимации критика и к корреляции данных. TD3 (Twin Delayed DDPG) вводит два критика, задержку обновления актора и сглаживание целевого действия, что снижает переоценку Q-функции и повышает устойчивость. Альтернативой является стохастический подход, где оптимизируется не только ожидаемая награда, но и энтропия политики; наиболее распространённый представитель — SAC (Haarnoja et al., 2018).

Достоинства:

- высокая выборочная эффективность за счёт off-policy данных и replay-буфера,
- естественная работа с непрерывным управлением без дискретизации U,
- возможность обучения на смешанных наборах траекторий, полученных при разных возмущениях и начальных условиях, что полезно при построении робастных стратегий.

Недостатки:

- качество политики тесно связано с качеством критика; при ошибках аппроксимации возможны переоценка Q и деградация обучения,
- чувствительность к нормировкам, масштабу вознаграждений, параметрам исследовательского шума и распределению стартовых состояний,
- ограничения на состояние/управление не обеспечиваются автоматически и требуют специальных средств (проекции, барьерные конструкции, constrained RL).

С практической точки зрения полезно различать детерминированные и стохастические политики. Детерминированные методы (DDPG/TD3) нередко дают высокое качество при тонкой настройке, но могут быть менее робастны к смене условий из-за «жёсткости» отображения $x \to u$.

Стохастические методы (SAC) обычно демонстрируют более корректное исследование и большую устойчивость к разреженным наградам (терминальные условия, минимальное время), поскольку энтропийная регуляризация препятствует преждевременному «схлопыванию» политики в узкую область управлений.

В результате off-policy actor–critic методы представляют собой один из наиболее практичных инструментов синтеза управления в высокоразмерных системах при наличии симулятора и возможности накопления данных.

Модель-ориентированное обучение с подкреплением

Модель-ориентированные методы исходят из идеи: если удаётся построить (или уточнить) модель динамики $\hat{f}$ и, при необходимости, модель вознаграждения $\hat{r}$, то синтез управления можно выполнять как задачу планирования по модели. Для оптимального управления это естественно, поскольку классические методы (динамическое программирование, MPC) опираются на модель. Отличие model-based RL состоит в том, что модель может быть неизвестна полностью, и тогда она восстанавливается и уточняется по данным взаимодействия агента со средой (или с имитационной моделью).

Ранним примером data-efficient model-based поиска политики служит подход PILCO, использующий вероятностную модель динамики и оптимизацию политики по ожидаемой стоимости (Sutton, Barto, 2018).

На практике распространены два типовых сценария:

- **Обучение модели + MPC/планирование.** Сначала по данным строится прогнозная модель переходов (часто нейросетевая или ансамблевая), затем на каждом шаге решается короткогоризонтная задача оптимизации (shooting, CEM (Cross-Entropy Method), iLQR (iterative Linear Quadratic Regulator) и др.), а в контуре управления используется «уходящий горизонт» (receding horizon). Характерные представители этой линии — PETS (Probabilistic Ensembles with Trajectory Sampling) и родственные ансамблевые схемы, где неопределённость модели учитывается при планировании (Chua et al., 2018)
- **Обучение политики по синтетическим траекториям.** Модель используется для генерации «мнимых» переходов (Dyna-подобная идея), после чего политика обучается как в model-free RL, но на расширенном датасете. Важный практический момент — ограничение длины синтетических роллаутов (симуляций), чтобы не накапливать ошибку модели.

Классический риск model-based RL — смещение из-за ошибки модели (model bias): даже малая систематическая ошибка в $\hat{f}$ может приводить к накоплению ошибки на горизонте и, как следствие, к неверному выбору управления. В инженерных задачах это проявляется особенно резко при сильной нелинейности и при наличии «опасных» областей фазового пространства. Типичный ответ — ансамбли моделей и явный учёт неопределённости, а также ограничение горизонта синтетического моделирования. В MBPO (Model-Based Policy Optimization) эта идея реализована как генерация коротких роллаутов модели с последующим off-policy обучением, что обеспечивает хороший компромисс между выборочной эффективностью и смещением.

Достоинства:

- высокая выборочная эффективность и возможность обучения при малом числе реальных траекторий,
- более прямое включение ограничений и терминальных условий через планировщик/MPC,
- естественная возможность использовать структурные знания о физике (параметрические модели + обучаемые поправки).

Недостатки:

- риск смещения из-за ошибки модели и деградации при переносе,
- необходимость решать вложенную задачу оптимизации в реальном времени, что повышает вычислительную сложность,
- чувствительность к параметризации модели и к качеству датасета.

В задачах поиска оптимального управления динамическими системами model-based RL часто выступает как компромисс между «чистым» RL и классическим MPC: RL обеспечивает адаптацию модели/стоимости по данным, а MPC даёт контролируемое поведение и удобный механизм учёта ограничений. При наличии грубой физической модели эффективны гибриды вида $\dot{x} = f(x,u,p) + \Delta_\theta(x,u)$, где Δ_θ обучается по данным и компенсирует немоделируемые эффекты (Sutton, Barto, 2018).

Обучение по фиксированному датасету

В ряде инженерных приложений (и особенно в аэрокосмических задачах) онлайн-эксплорация недопустима: ошибки управления могут приводить к выходу из допустимой области, повреждению аппарата или срыву миссии. В таких условиях естественным становится offline (batch) RL, где обучение выполняется по фиксированному набору траекторий, а взаимодействие со средой в процессе обучения отсутствует. Датасет может быть получен либо по результатам эксплуатации, либо в симуляции, либо с помощью классических методов оптимального управления (например, методами прямой дискретизации, принципом максимума, MPC), что позволяет получить богатый набор квазиоптимальных траекторий.

Ключевая трудность offline RL — распределительный сдвиг: обучаемая политика может предлагать действия, которые в датасете практически не встречались, и тогда оценка Q становится недостоверной (ошибка экстраполяции). В динамических системах это приводит к особенно опасным эффектам: небольшая ошибка оценки на одном шаге может «увести» систему в область, где данные отсутствуют, после чего качество управления резко деградирует. Поэтому большинство успешных offline-алгоритмов вводит явные ограничения на область действий, в которой допускается улучшение политики, либо делает оценку Q консервативной по отношению к действиям вне поддержки данных.

Типичные представители: BCQ (Batch-Constrained Deep Q-Learning) ограничивает действия вблизи поведения, представленного в датасете (через генеративную модель действий) (Fujimoto, Meger, Precup, 2019); BEAR (Bootstrapping Error Accumulation Reduction) и родственные методы используют регуляризацию по расхождению между обучаемой и поведенческой политикой (Kumar et al., 2019); CQL (Conservative Q-Learning) добавляет консервативный член, занижающий оценки Q для действий вне поддержки данных, тем самым снижая риск переоценки (Kumar et al., 2020). Более поздние методы (например, IQL) строят обновления, устойчивые к неточностям оценки поведения, что упрощает применение в задачах непрерывного управления (Kostrikov, Nair, Levine, 2022).

Отдельного упоминания требует проблема оценки качества политики без запуска в реальной среде (off-policy evaluation). Классические оценки на основе важностного взвешивания теоретически корректны, но на длинных горизонтах обладают большой дисперсией; на практике применяются смешанные (doubly-robust) оценки и модельные аппроксимации. Для задач оптимального управления это означает необходимость иметь отдельный стенд валидации (симулятор/цифровой двойник) и контролировать «область применимости» обучаемой политики

Достоинства:

- отсутствие необходимости опасной онлайн-эксплорации,
- возможность использовать «наследованные» данные: траектории, полученные классическими оптимальными регуляторами, демонстрации экспертов, результаты численного решения задач оптимального управления,
- удобная интеграция с инженерным жизненным циклом: обучение → верификация на тестовых сценариях → внедрение.

Недостатки:

- высокая зависимость результата от полноты и качества датасета,
- сложность корректной оценки качества без дополнительных предположений,
- риск деградации при выходе за область, покрытую данными, особенно при сильных нелинейностях и разреженных терминальных наградах.

В задачах поиска оптимального управления offline RL часто выступает как «мост» между классическими методами и глубоким RL: сначала формируется датасет траекторий (например, методами оптимального управления или MPC), затем обучается политика обратной связи, обеспечивающая быстрый онлайн-расчёт управления и допускающая формальную верификацию на ансамбле тестовых сценариев.

Робастное RL и работа с неопределённостями

Даже при корректном алгоритме обучения возникает типичная инженерная проблема: политика, обученная в одном наборе условий, деградирует при изменении параметров объекта, возмущений и шумов измерений. Для задач оптимального управления это критично, поскольку реальная динамика почти всегда отличается от расчётной. Робастное RL рассматривает различные модели неопределённости и стремится синтезировать политику, устойчивую к вариациям среды.

С теоретической точки зрения естественной рамкой является robust MDP (Markov Decision Process), где переходы принадлежат неопределённому множеству, а оптимизация проводится по «наихудшему случаю» или по заданному распределению неопределённостей. Для непрерывных динамических систем аналогичная идея реализуется через случайные параметры модели (масса, моменты инерции, коэффициенты сопротивления и т.п.) и внешние возмущения. На практике наиболее распространена domain randomization: в процессе обучения параметры среды рандомизируются в заданном диапазоне, а политика оптимизируется по среднему качеству, что повышает переносимость на реальные условия (Tobin et al., 2017).

Более «жёсткие» варианты робастификации используют minimax-постановки и адверсариальные (намеренные) возмущения. В EPOpt обучение проводится на подмножестве наиболее «тяжёлых» сценариев (нижний квантиль по награде), что приближает критерий плохого случая и повышает надёжность. В адверсариальном RL вводится внешний агент-возмущатель, ухудшающий динамику, а обучаемая политика должна компенсировать худшие воздействия (Pinto et al., 2017). Отдельный класс составляют риско-чувствительные критерии, позволяющие управлять «хвостами» распределения качества, что важно для миссий с высокой ценой редких отказов.

С инженерной точки зрения полезны и более прикладные приёмы:

- обучение ансамбля политик и выбор управления по принципу «согласия» (или с учётом дисперсии),
- совмещение робастного обучения с идентификацией параметров «в контуре» и адаптацией,
- введение в наблюдения явных оценок параметров (масса, момент инерции) или скрытых переменных, что переводит часть неопределённости в задачу оценивания состояния.

Эти приёмы не дают строгих гарантий, но часто существенно уменьшают деградацию качества.

Достоинства:

- повышение переносимости и устойчивости к дрейфам параметров,
- возможность учёта неопределённостей без явного вывода робастных условий устойчивости,
- совместимость с базовыми RL-алгоритмами (PPO/SAC/TD3) как «надстройка».

Недостатки:

- рост вычислительных затрат: для покрытия диапазона параметров требуется существенно больше траекторий,
- риск излишней консервативности, когда политика теряет качество на номинальной модели,
- отсутствие гарантий, если реальные неопределённости выходят за обучающий диапазон.

В целом, робастные надстройки следует рассматривать как обязательный элемент практического применения RL в управлении сложными объектами, где модель неизбежно является приближённой.

Гибридные схемы RL

Практика применения RL к задачам оптимального управления показывает, что «чистые» схемы обучения редко используются в изоляции. Гораздо чаще эффективными оказываются гибридные подходы, в которых RL-модуль встраивается в классическую архитектуру управления и решает одну из вспомогательных задач: уточнение модели, настройка критерия, формирование опорного управления,

компенсация немоделируемых эффектов. Такая гибридизация снижает риски обучения и упрощает верификацию.

Один из наиболее распространённых вариантов — связка RL и MPC. Здесь MPC обеспечивает явный учёт ограничений и предсказуемое поведение, а RL используется для:

- обучения модели динамики,
- приближённого вычисления стоимости-к-идти/терминального функционала,
- настройки весов в критерии,
- ускорения решения вложенной оптимизационной задачи (например, через тёплый старт или параметризацию управляющего профиля) (Mayne, 2014).

В прикладном смысле MPC выступает «страховкой», ограничивающей действия обучаемого модуля внутри допустимой области по состояниям и управлениям.

Второй характерный класс — residual RL, когда к базовому (часто линейному или MPC) регулятору добавляется обучаемая поправка: $u = u_{base}(x) + \Delta u_\theta(x)$. Это позволяет использовать известные свойства базового регулятора (устойчивость в окрестности, ограниченность управления), а RL-модуль компенсирует немоделируемые нелинейности и возмущения. Аналогично, RL может применяться для адаптивной коррекции параметров регулятора или фильтра состояния, не заменяя всю систему управления.

Третья важная линия — связка имитационного обучения и RL. При наличии траекторий оптимального управления (полученных численным решением или экспертом) можно обучить политику методом поведенческого клонирования, а затем улучшить её RL-алгоритмом. Для снижения накопления ошибок используется DAgger-подобная схема агрегирования данных (Ross, Gordon, Bagnell, 2011), а для обучения по демонстрациям без явной модели поведения — методы типа GAIL (Generative Adversarial Imitation Learning) (Ho, Ermon, 2016). Для задач оптимального управления это особенно удобно: демонстрации позволяют быстро «подвести» политику к области приемлемого поведения, после чего RL уже оптимизирует тонкие характеристики (минимизация времени, расхода ресурса и т.п.).

Достоинства:

- повышение надёжности и предсказуемости за счёт базового регулятора/MPC,
- снижение требований к данным и ускорение обучения,
- более простая инженерная валидация (можно тестировать RL-модуль как надстройку).

Недостатки:

- необходимость аккуратного согласования уровней (чтобы RL-поправка не разрушала свойства базовой системы),
- рост сложности программной реализации,
- риск «замыкания» на узкой области, заданной базовым регулятором, если требуется принципиально иное поведение.

В результате гибридные схемы представляются наиболее перспективными для реальных задач оптимального управления, поскольку позволяют сочетать вычислительную гибкость RL с формальными и проверяемыми элементами классической теории управления.

Заключение

Рассмотренные классы методов обучения с подкреплением формируют достаточно целостный инструментарий для решения прикладных задач поиска оптимального управления динамическими системами — от синтеза регуляторов с обратной связью до построения стратегий планирования при сложных терминальных условиях и жёстких ограничениях. Модель-свободные подходы (value-based, on-policy и off-policy actor–critic) позволяют получать управляющие политики без явного решения уравнений, следующих из условий оптимальности и потому особенно привлекательны при труднодоступных градиентах и сложной нелинейной динамике. Вместе с тем практика показывает, что ключевые инженерные ограничения — безопасность, робастность и переносимость — требуют специальных надстроек (constrained/safe RL, робастное обучение, offline-схемы) и тщательной верификации на ансамбле сценариев. Наиболее перспективным направлением для реальных систем управления представляются гибридные архитектуры, сочетающие RL с MPC и классическими регуляторами и подходами (Пантелеев, Бортаковский, 2016): они позволяют сохранить проверяемые элементы теории управления и одновременно использовать преимущества обучения по данным при наличии неопределённостей и немоделируемых эффектов.

Список источников / References

1. Пантелеев, А.В., Пановский, В.Н. (2016). Прикладное применение интервального метода взрывов для поиска оптимального программного управления солнечным парусом. *Вестник НПО им. С.А. Лавочкина, 4, 110—117*.
   Panteleev, A.V., Panovskiy, V.N. (2016). Application of interval explosion method for generation of optimal program control of solar sail. *S.A. Lavochkin NGO Bulletin, 4, 110—117* (In Russ).
2. Пантелеев, А.В., Бортаковский, А.С. (2016) Теория управления в примерах и задачах. *ИНФРА-М, Москва*.
3. Panteleev, A.V., Bortakovskiy, A.S.(2016) Control Theory in Examples and Problems. *INFRA-M, Moscow*. (In Russ).
4. Achiam J. et al. (2017). Constrained Policy Optimization. https://doi.org/10.48550/arXiv.1705.10528
5. Barto, A.G., Sutton, R.S., Anderson, C.W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. SMC*. https://doi.org/10.1109/TSMC.1983.6313077
6. Bellman, R. (1957). Dynamic Programming. *Princeton University Press*.
7. Bertsekas, D.P. (2024). Model Predictive Control and Reinforcement Learning. https://doi.org/10.48550/arXiv.2406.00592
8. Bianchi, C. et al. (2025) Robust solar sail trajectories using proximal policy optimization. https://doi.org/10.1016/j.actaastro.2024.10.065
9. Chua, K. et al. (2018) Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. *NeurIPS*. https://doi.org/10.48550/arXiv.1805.12114

10. Fujimoto, S., Meger, D., Precup, D. (2019). Off-Policy Deep Reinforcement Learning without Exploration. *ICML*. https://doi.org/10.48550/arXiv.1812.02900
11. Haarnoja, T. et al. (2018) Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. https://doi.org/10.48550/arXiv.1801.01290
12. Ho, J., Ermon, S. (2016). Generative Adversarial Imitation Learning. *NeurIPS*. https://doi.org/10.48550/arXiv.1606.03476
13. Kostrikov, I., Nair, A., Levine, S. (2022). Offline Reinforcement Learning with Implicit Q-Learning. *ICLR*. https://doi.org/10.48550/arXiv.2110.06169
14. Kumar, A. et al. (2019). Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. *NeurIPS*. https://doi.org/10.48550/arXiv.1906.00949
15. Kumar, A. et al. (2020). Conservative Q-Learning for Offline Reinforcement Learning. *NeurIPS*. https://doi.org/10.48550/arXiv.2006.04779
16. Lillicrap, T.P. et al. (2016). Continuous control with deep reinforcement learning. *ICLR*. https://doi.org/10.48550/arXiv.1509.02971
17. Ma, Z. et al. (2018). Reinforcement Learning-Based Satellite Attitude Stabilization. https://doi.org/10.3390/s18124331
18. Mayne, D.Q. (2014) Model Predictive Control: Recent Developments and Future Promise. *Automatica*. https://doi.org/10.1016/j.automatica.2014.10.128
19. Mnih, V. et al. (2015). Human-level control through deep reinforcement learning. *Nature*. https://doi.org/10.1038/nature14236
20. Pinto, L. et al. (2017). Robust Adversarial Reinforcement Learning. *ICML*. https://doi.org/10.48550/arXiv.1703.02702
21. Ross, S., Gordon, G., Bagnell, D. (2011) A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. *AISTATS*. https://doi.org/10.48550/arXiv.1011.0686
22. Schulman, J. et al. (2015) Trust Region Policy Optimization. https://doi.org/10.48550/arXiv.1502.05477
23. Schulman, J. et al. (2017). Proximal Policy Optimization Algorithms. https://doi.org/10.48550/arXiv.1707.06347
24. Sutton, R.S., Barto A.G. (2018). Reinforcement Learning: An Introduction (2nd ed.). *MIT Press*
25. Tobin, J. et al. (2017). Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. *IROS*. https://doi.org/10.48550/arXiv.1703.06907
26. Williams, R.J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning.

Информация об авторах

Валентин Николаевич Пановский, кандидат физико-математических наук, доцент кафедры № 805 «Математическая кибернетика», институт № 8 «Компьютерные науки и прикладная математика», ФГБОУ ВО «Московский авиационный институт (национальный исследовательский университет)», Москва, Российская Федерация, ORCID: https://orcid.org/0009-0007-1708-8984, e-mail: panovskiy.v@yandex.ru

Information about the authors

Valentin N. Panovskiy, Candidate of Sciences (Physics & Math), assistant professor of department № 805, «Mathematical Cybernetics», Institute № 8 «Computer Science and Applied Mathematics», Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0009-0007-1708-8984, e-mail: panovskiy.v@yandex.ru

Вклад авторов

Пановский В.Н. — идеи исследования; детализация и структурирование обзора, написание и оформление рукописи.

Contribution of the authors

Valentin N. Panovskiy — research ideas; detailing and structuring the review, writing and formatting the manuscript.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest

The authors declare no conflict of interest.

Поступила в редакцию 15.02.2026 — Received 2026.02.15
Поступила после рецензирования 20.02.2026 — Revised 2026.02.20
Принята к публикации 21.02.2026 — Accepted 2026.02.21
Опубликована 31.03.2026 — Published 2026.03.31