«Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В. И. Ульянова (Ленина)»
(СПБГЭТУ "ЛЭТИ")

Направление	27.04.04 - Управление в технических системах
Программа	Управление и информационные технологии в технических системах
Факультет	КТИ
Кафедра	АПУ

К защите допустить
Зав. кафедрой Шестопалов М Ю.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА

Тема: ПРИМЕНЕНИЕ МЕТОДОВ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ В ЗАДАЧАХ УПРАВЛЕНИЯ

Студент _____ Шпаковская И.И.

Руководитель д.т.н., профессор _____ Душин С.Е.

Консультанты к.т.н., доцент _____ Белаш О.Ю.

к.т.н., доцент _____ Власенко С.В.

д.э.н., доцент _____ Сулейманкадиева А.Э.

Санкт-Петербург
2021

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Утверждаю
Заф. кафедры АПУ
_____ Шестопалов М. Ю.
«___»_____2021 г.

Студентка Шпаковская И.И. Группа 5391
Тема работы: Применение методов обучения с подкреплением в задачах управления
Место выполнения ВКР: кафедра АПУ Санкт-Петербургского государственного электротехнического университета «ЛЭТИ»
Исходные данные (технические требования): Необходимо проанализировать методы обучения с подкреплением на предмет применения алгоритмов для задачи разработки систем управления.
Содержание ВКР: Основные положения обучения с подкреплением, Методы обучения с подкреплением в задаче управления, Применение методов обучения с подкреплением в задаче разработки регуляторов для сложных систем управления.
Перечень отчетных материалов: пояснительная записка, презентация
Дополнительные разделы: Составление бизнес-плана по коммерциализации результатов НИР магистранта

Дата выдачи задания «___»_____2021 г.
Дата предоставления ВКР к защите «___»_____2021 г.

Студентка _____ Шпаковская И.И.

Руководитель д.т.н., профессор _____ Душин С.Е.

КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Утверждаю
Заф. кафедры АПУ
_____ Шестопалов М. Ю.
«___»_____2021 г.

Студентка Шпаковская И.И. Группа 5391

Тема работы: Применение методов обучения с подкреплением в задачах управления.

№ п/п	Наименование работ	Срок выполнения
1	Обзор литературы по теме работы	08.02 – 20.02
2	Изучение общей терминологии и методов области обучения с подкреплением	20.02 – 09.03
3	Изучение программных продуктов, предоставляемых MATLAB	09.03 – 25.03
4	Анализ схемы алгоритмов обучения с подкреплением, сравнение с базовыми адаптивными схемами	25.03 – 26.04
5	Реализация регуляторов на основе обучения с подкреплением для математического маятника, модели распространения опухоли и производства пенициллина	26.04 – 10.05
6	Разработка бизнес-плана по коммерциализации проекта	10.05 – 12.05
7	Оформление пояснительной записки	12.05 – 25.05

Студентка _____ Шпаковская И.И.

Руководитель д.т.н., профессор _____ Душин С.Е.

РЕФЕРАТ

Пояснительная записка 70 стр., 20 рисунков, 14 таблиц,

Ключевые слова: обучение с подкреплением, оптимальное управление, динамическое программирование, приближенное динамическое программирование, адаптивное управление, оптимально-адаптивный регулятор, нейродинамическое программирование

Тема выпускной квалификационной работы: Применение методов обучения с подкреплением в задачах управления

Объект исследования — методы обучения с подкрепленим это то, на что направлен процесс познания (индивид, коллектив, общность людей, сфера деятельности и т.п.). Предмет исследования — определить связь методами обучения с подкреплением с адаптивными и оптимальными подходами в теории автоматического управления.

Цель исследования — проанализировать современные методы искусственного интеллекта, а именно метод обучения с подкреплением, на предмет применимости для решения задач автоматизированного управления сложных систем.

В данной работе изложена сущность современного метода машинного обучения – обучения с подкреплением для задачи автоматизированного управления сложных систем. Представленны общая терминология и методы обучения с подкреплением для задач управления. Показаны методы обучения с подкреплением с позиции решения задачи приблеженного динамического программирования. Показано сходство косвенных адаптивных регуляторов с методами обучения с подкреплением на базе моделей и сходство структуры прямых адаптивных регуляторов с безмодельными методами обучения с подкреплением. Реализованы регуляторы на базе алгоритмов глубокого обучения с подкреплением для перевернутого маятника, роста опухоли и производства пенициллина.

ABSTRACT

Keywords: reinforcement learning, optimal control, dynamic programming, approximate dynamic programming, adaptive control, optimal-adaptive controller, neurodynamic programming

The subject of the graduate qualification work is: Application of reinforcement learning for control systems problems

The object of the study is reinforcement learning methods - this is what the cognition process is directed to (an individual, a collective, a community of people, a field of activity, etc.). The subject of the research is to determine the relationship between reinforcement learning methods and adaptive and optimal approaches in the theory of automatic control.

The purpose of the study is to analyze modern methods of artificial intelligence, namely, the reinforcement learning method, for applicability for solving problems of automated control of complex systems.

This paper outlines the essence of the modern method of machine learning - reinforcement learning for the problem of automated control of complex systems. General terminology and reinforcement learning methods for management tasks are presented. Reinforcement learning methods from the position of solving the approximate dynamic programming problem are shown. The similarity of indirect adaptive controllers with model-based reinforcement learning methods and the similarity of the structure of direct adaptive controllers with modelless reinforcement learning methods are shown. Implemented deep-learning reinforcement-based controllers for inverted pendulum, tumor growth, and penicillin production.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяют следующие термины с соответствующими определениями:

АСУ ТП – Автоматизированная система управления технологическим процессом

ДП – динамическое программирование

МППР – Марковский процесс принятия решения

ИИ – искусственный интеллект

ОУ – объект управления

RL – обуение с подкреплением (англ. Reinforcement learning)

MPC – управление с прогнозирующими моделями (англ. Model Predictive Control)

PI – алгоритм итерации по стратиегиям (англ. Policy Iteration)

VI – алгоритм итерации по ценности (англ. Value Iteration)

ВВЕДЕНИЕ

Классические и современные методы теории автоматического управления направленны на разработку систем управления на основе математического и компьютерного моделирования объекта, путем описания физики процессов, как правило, нелинейных динамических объектов. В то же время, с учетом скорости развития вычислительных ресурсов, возникает вопрос о возможности применения накопленных об объекте данных для разработки и уточнения систем регулирования. Такие области как адаптивное, нейросетевое, нечеткое управление и идентификация позволяют не ограничиваться представлением объекта в форме пространства состояний или дифференциальных уравнениях, а учитывают тот факт, что объект представлен набором данных.

Разработка адаптивных регуляторов характерна тем, что процесс подстройки параметров регулятора возможен в режиме онлайн, то есть, используя данные, измеренные в реальном времени. Адаптивные регуляторы могут удовлетворять некоторым условиям оптимальности, но в то же время, адаптивные контроллеры, как правило, не проектируются как оптимальные в смысле минимизации заданного функционала качества. Так же адаптивные регуляторы требуют исследования структуры объекта управления. Косвенные адаптивные регуляторы используют методы идентификации, чтобы сначала определить параметры системы, а затем использовать полученную модель для нахождения оптимального управления. Оптимальное управление обычно проектируются в автономном режиме (режиме off-line) путем решения уравнений Гамильтона–Якоби–Беллмана, решение которых часто затруднительно.

Возникает вопрос о возможности применения современных методов, которые способны решать задачи управления с учетом оптимизации функционала, адаптируясь к изменением параметров объекта управления.

В большинстве современных исследований по обучению машин, распознаванию изображений и искусственным нейронным сетям рассматривается подход обучения с учителем, цель которого - обобщать, располагая лишь фиксированным набором данных, с ограниченным количеством примеров. Тогда как существует подход, называемый *обучение с подкреплением*, где в основе лежит идея об активном взаимодействии со средой методом проб и ошибок, с целью определения последовательности действий, максимизирующую некоторую награду. Другими словами, алгоритмы обучения с подкреплением построены на

идее, что эффективные управляющие значения должны запоминаться с помощью сигнала подкрепления для повторного использования.

Большая часть теории, лежащей в основе обучении с подкреплением, основана на предположении гипотезы вознаграждения, которая вкратце утверждает, что все цели и задачи агента могут быть объяснены одним скаляром, называемым вознаграждением. Более формально гипотеза вознаграждения представлена далее: все, что мы подразумеваем под целями и задачами, можно хорошо представить как максимизацию ожидаемого значения совокупной суммы полученного скалярного сигнала (называемого вознаграждением).

Актуальность исследования заключается в интересе со стороны сообщества инженеров по автоматизации в использовании новых подходов на основе алгоритмов машинного обучения с применением накопленных данных.

Объект исследования — методы обучения с подкрепленим это то, на что направлен процесс познания (индивид, коллектив, общность людей, сфера деятельности и т.п.).

Предмет исследования — определить связь методами обучения с подкреплением с адаптивными и оптимальными подходами в теории автоматического управления.

Цель исследования – проанализировать современные методы искусственного интеллекта, а именно метод обучения с подкреплением, на предмет применимости для решения задач автоматизированного управления сложных систем.

В соответствии с целью исследования, определяются следующие **задачи работы**:

- Проанализировать терминологию и математический аппарат области обучения с подкреплением, свойственный сообщесту искусственного интеллекта.
- Формализовать терминологию, методов и алгоритмов области обучения с подкреплением в рамках теории автоматического управления.
- Привести сравнительный анализ концепции разработки регуляторов на основе обучения с подкрепленим с методами адаптивного, оптимального управления.
- Провести ряд программных экспериментов для сложных объектов управления. Качественно установить применимость регулятора на основе обучения с подкреплением.

Выдвигается следующая **гипотеза** – обучения с подкреплением напрямую связанно с теорией оптимального управления, поэтому терминология и методы обучения с подкреплением напрямую связанны с оптимальным управлением. Общая структурная схема регулирования на базе обучения с подкреплением соотвествует известным в теории адаптивного управления подходам. Методы обучения с подкреплением позволяют получать качество регулирования сложными техническими системами на том же уровне, что и численные методы оптимального управления.

Теоретическая и методологическая база исследования. Теоретической основой выпускной квалификационной работы послужили исследования Дмитрия Бертсекаса в области динамического программирования и оптимального управления, а так же работы Ричарда Саттона и Эндрю Барто в области обучения с подкреплением. Практическая часть работы выполнялась на основании научных статей по разработке математических моделей в дифференциальных уравнениях, а так же открытых интернет источников по реализации алгоритмов обучения с подкреплением на языке Python и MATLAB. При подготовке ВКР были использованы материалы таких учебных дисциплин, как «Оптимальные системы управления», «Современные методы теории управления», «Адаптивные системы управления», «Нейросетевые системы управления», «Коммерциализация результатов научных исследований и разработок».

1 ОСНОВНЫЕ ПОЛОЖЕНИЯ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Цель главы – дать общее представление о терминах и алгоритмах обучения с подкреплением. В параграфе 1.1 приводится актуальность изучения новой области инженерами в области автоматизации. В параграфе 1.2 приведены основные положения обучения с подкреплением в терминологии свойственной для специалистов в области искусственного интеллекта, именно на базе введенной в данном параграфе терминологии формулируются методы и алгоритмы обучения с подкреплением в параграфе 1.6.

1.1 Актуальность обучения с подкреплением

Обучение с подкреплением (англ. Reinforcement learning, RL) – это широкая область, объединяющая исследователей из самых разных областей: искусственный интеллект (ИИ), управление, робототехника, исследования операций, экономика, нейробиология. По данной теме было опубликовано множество книг и обзорных статей представляющих самые разные области: ИИ, где классический учебник – это учебник Саттона и Барто (1998) [1] со вторым изданием [2], но также и другие [3; 4]; теория управления [5]; оптимальное управление [6; 7]; робототехника [8]; Некоторые исследования сосредоточены на конкретных задачах RL: как методы градиента стратегии, аппроксимация функций, байесовские формулировки RL, иерархический RL, многоагентные подходы, глубокий RL, безопасный RL и так далее.

В то же время, необходимо упомянуть о том, что количество приложений методов RL для разработки систем управления техническими объектами мало в сравнении с количеством и масштабам приложений в области разработки рекомендательных систем, игр, обработки информации и исследований операций.

С одной стороны сложность применения к техническим системам затруднена в силу сложности обеспечения безопасности. В отличие от симуляции, в физическом мире действия имеют реальные последствия. В результате любой алгоритм, развернутый в реальных системах, должен отвечать нормам безопасности и качества. Безопасность в известных средах давно рассматривается и формализуется сообществами контроля и формальных методов, где можно синтезировать стратегии контроля, соответствующие заданной спецификации. Все методы определения устойчивости в области управления направлены на работу с моделями объектов. Тогда как при применении RL нет необходимо-

сти в математическом моделировании объетка. основаны на известной модели системы.

Другая сложность RL для специалистов в области управления – особенность терминологии и акцента на данные, которые привычны специалистам в области ИИ. Несмотря на эти и другие сложности, наблюдается рост исследований методов RL со стороны инженеров систем управления. Это подтверждается увеличением количества докладов и статьей, опубликованных в профильных журналах по автоматическому управлению (рис.1.1). Международная федерация по автоматическому управлению на конгрессе в 2020 году (IFAC-V 2020) вынесла на пленарное заседание доклад «Reinforcement Learning for Process Control and Beyond» (автор Jay H. Lee), что говорит о важности и влиятельности области обучения с подкреплением.

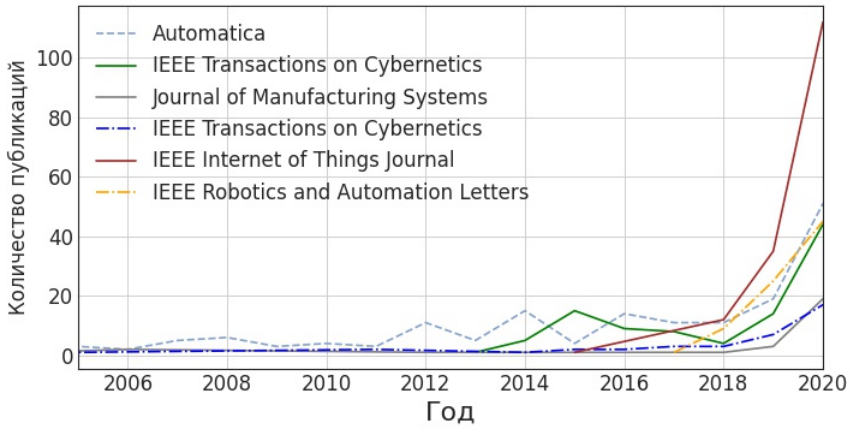


Рисунок 1.1 — Тендеции количества публикаций на тему RL в специализорованных журналах по управлению в технических системах

Учитывая существующий запрос на разработку интеллектуальных систем управление, что связано с возрастающей сложностью объектов управления, можно предполагать, что регуляторы с применением обучения с подкреплением станут неотъемлемой частью любого автоматизированного технологического процесса.

1.2 Терминология обучения с подкреплением

Обучение с подкреплением – это класс методов машинного обучения, которые основанны на взаимодействии алгоритма со средой и изменения стратегии действия или политики управления на основе стимулов, полученных в ответ на действия алгоритма, с целью получения желаемого результата. Цель применения методов RL — определить последовательность входных сигналов, которая

обеспечивает желаемую работу динамической системы, начиная с минимальных знания о работе системы.

Агент – интеллектуальная сущность (система/робота/алгоритм) принимающая решения, взаимодействует с объектом, который называется окружением или средой (англ., environment). Агент и окружение взаимодействуют на каждом шаге последовательности $t = 0, 1, 2, \ldots$ При этом окружение задается, зависящим от времени состоянием $S_t \in S$, где S – множество всех возможных состояний. Агенту в каждый момент времени в общем случае доступно только некоторое наблюдение (англ., observation) текущего состояния среды. На основании наблюдений состояния агент выполняет процедуру выбора действия $A_t \in A(S_t)$, $A(S_t)$ – множество действий доступных агенту в состоянии S_t. Процедура выбора действий агентом называются стратегией или политикой (англ., policy) и обозначается как π_t, описывая вероятность выбора действия $A_t = a$ в состоянии $S_t = S$. По результатам применения действия $A(S_t)$ в среде, на вход агента поступает численная награда $R_{t+1} \in R$ (англ. reward) и новое состояние среды $A(S_{t+1})$. Методы RL определяют способ выбора стратегии агентом в результате полученного опыта.

Необходимо уточнить, что понимается под термином окружающая среда – это все то, что не является агентом. Среда принимает текущее состояние и действие агента в качестве входных данных и возвращает вознаграждение агента и состояние.

Марковские процессы. Одна из структур для RL основана на марковских процессах принятия решений (МППР). Задача RL удовлетворяет условию Марковости – процесс зависит только от текущего состояния и не зависит от всей предыдущей истории. МППР позволяет формализовывать основные элементы RL, такие как функции ценности, награды, а далее основные алгоритмы RL. Многие задачи принятия решений могут быть сформулированы как МППР, включая системы управления с обратной связью. МППР представляет собой четверку (S, A, P, r), где:

S – конечное пространство состояний;

A – конечное пространство действий;

P – функция переходов, определяющая вероятность перейти в состояние s' из состояния s посредством действия a;

r – функция награды. При условии любого состояния s и действия a, вероятность каждого возможного следующего состояния равна:

$$p_{ij}(a,t) = P\{s_{t+1} = j | s_t = i, a_t = a\}$$

На рис. 1.2 представлена общая структурная схема RL в терминах МППР.

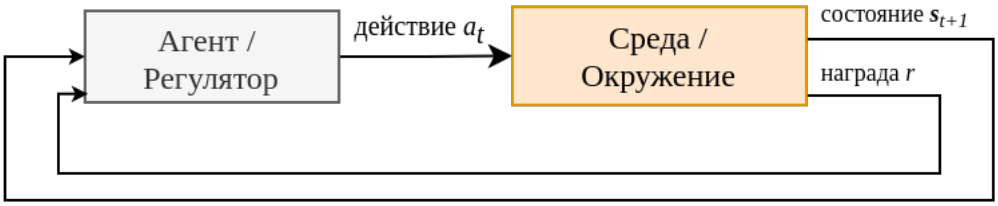


Рисунок 1.2 — Общая структура обучения с подкреплением

В задаче обучения с подкреплением, цель агента формализуется в сигнале награды, которую получает агент от среды на каждом временном шаге. Использование сигнала награды для формализации цели – одна из отличительных характеристик задачи обучения с подкреплением. Алгоритмы RL оптимизируют ту награду, которая была предоставлена ему на входе, при этом принимая истинной «гипотезы награды», которая утверждает, что любую интеллектальную задачу можно определить (задать) при помощи функции награды. Но, как несложно догадаться, на практике дизайн функции награды оказывается сложной задачей и качество работы алгоритмы, напрямую зависит от способности формализовать цель управления.

Пусть последовательность наград после временного шага t, записывается как r_t, r_{t+1}.... Тогда в рамках задачи обучения с подкреплением, необходимо максимизировать ожидаемую награду R_t, которая определена, как сумма всех последующих наград: Тогда в рамках задачи обучения с подкреплением, необходимо максимизировать ожидаемую награду R_t, которая определена как сумма всех последующих наград:

$$R_t = r_t + r_{t+1} + r_{t+2} + \ldots$$

Такое выражение удовлетворительно для задач с конечным временным шагом, В системах с конечным количеством шагов взаимодействия, вводится понятие разбиения взаимодействие агента и окружения на последовательности – эпизоды. Каждый эпизод заканчивается особым терминальным состоянием, за которым следует переход к заданному начальному состоянию или к выбору начального состояния из распределения начальных состояний. Среда называется эпизо-

дичной, если для любой стратегии процесс взаимодействия гарантированно завершается не более чем за некоторое конечное число шагов.

Для ряда задач управления характерно продолжительное действие – взаимодействие агента и окружения не разбивается на эпизоды. В этом случае ожидаемая награда, которую необходимо максимизировать, может достигать бесконечности. Для решения таких задач вводится коэффициент обесценивания (дисконтирования), тогда награда формируется следующим образом:

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots + \gamma^{k-1} r_{t+k-1}$$

где $\gamma \in (0, 1]$ – коэффициент дисконтирования.

Предполагается, что на каждом шаге с вероятностью $1 - \gamma$ взаимодействие обрывается, и итоговым результатом агента станет та награда, которую он успел собрать до прерывания.Это обеспечивает приоритет получению награды в ближайшее время перед получением той же награды через некоторое время. Математически смысл дисконтирования, во-первых, в том, что данный коэффициент позволяет гарантировать ограниченность оптимизируемого функционала, а во-вторых, выполнение условий некоторых теоретических результатов, которые явно требуют $\gamma < 1$.

Учитывая терминалогию приведенную выше, задача RL для заданного МППР может быть сформулирована как поиск стратегии π^*, максимизирующей среднюю дисконтированную суммарную награду.

$$J(\pi) = E_\pi \sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \rightarrow \max_\pi \tag{1.1}$$

В основе многих алгоритмов RL лежит понятие функции ценности и функции ценности действия. Это в некотором роде «обобщение» функционала 1.1, варьируя начальное состояние. *Функция ценности* (англ. Value Function) или оценочная функция состояния $V^\pi(s)$ показывает сколько набирает в среднем агент из состояния s_t при стратегии π. Функция ценности определяется для отдельных стратеги π и равна сумме наград:

$$V^\pi(s) = E_\pi\{R_t | s_t = s\} = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \middle| s_t = s \right] \tag{1.2}$$

где E_π – ожидаемое значение награды R_t при действии агента в соответсвии со стратегией π на каждом t.

Функция ценности действия $Q^{\pi}(s,a)$ (англ. Value Action) для стратегии π характеризует сколько набирает в среднем агент из состояния s_t после выполнения действия a.

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s, a_t = a\} = E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \middle| s_t = s, a_t = a\right]$$

Функцию ценности и функцию ценности действий также называют V-функцией и Q-функцией, соответственно.

Исходя из задания V-фунции (1.2) задача RL формулируется как определение политики $\pi(s,a)$, которая максимизирует награду:

$$\pi^*(s,a) = \arg\max_{\pi} V_t^{\pi}(s) = \arg\max_{\pi} E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \middle| s_t = s\right]$$

Политика $\pi^*(s,a)$ называется оптимальной политикой, и соответствующее оптимальное значение V-функции задается как:

$$V_t^*(s) = \max_{\pi} V_t^{\pi}(s) = \max_{\pi} E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \middle| s_t = s\right]$$

Легко заметить, что для максимизации $V_t^{\pi}(s_0)$, где s_0 – стартовое состояние, необходимо промаксимизировать $V_t^{\pi}(s)$. То есть задача имеет подзадачи эквивалентной структуры. Ряд методов RL основывается на рекурсивном свойстве V-функции. Тоже верно и для Q-функции. Это означает то, что для этих функций может быть записано управнение Беллмана, которое выражает отоношение между значениями функций в текущем состояний и значением в последующих состояний.

1.3 Историческая справка

Термин «подкрепление»(англ. reinforcement) унаследован из поведенческой психологии, а именно из работ физиолога И.П. Павлова. Здесь подкрепление обозначает награду или наказание за результат, который зависит как от принятых решений так и от внешних, в общем случае, не контролируемых воздействий. Под обучением здесь понимается поиск способов достичь желаемого результата методом проб и ошибок, то есть попытки решить задачу и использовать накопленный опыта для усовершенствования своей стратегии выбора действий в будущем.

Специалисты в области обучения с подкреплением на раннем этапе истории выделяют два основных направления развития. Одно направление связано

с обучением методом проб и ошибок. Второе направление связано с задачей оптимального управления и решением ее с помощью функций стоимости и динамического программирования.

Главной психологической идеей, которая используется в обучении с подкреплением, является метод проб и ошибок, предложенный ученым и философом Александром Бэном в 1855 году. Ученый объясняет возникновение произвольных движений – вводит представление о спонтанной активности нервной системы. Когда какое-либо движение более одного раза совпадает с состоянием удовольствия, то некоторая сила духа устанавливает между ними ассоциации. На базе идеи метода проб и ошибок начиная с 1930 годов было сделано ряд автоматов, демонстрирующий этот подход. Самая ранняя демонстрация — машина Томаса Росса 1933-ого года, которая могла пройти простейший лабиринт и запомнить последовательность переключателей. В 1952 году Клод Шеннон продемонстрировал лабиринт с роботом-мышью, который передвигался по лабиринту с помощью трех колес и магнита с обратной стороны лабиринта, он мог запомнить путь по лабиринту, исследуя его тем самым методом проб и ошибок. Появление таких электро-механических машин открыло путь к написанию компьютерных программ, способных к разным типам обучения, некоторые из которых были способны к обучению методом проб и ошибок.

Помимо психологического подхода, независимо развивался подход математического программирования. Термин «оптимальное управление» появился в конце 1950-х годов и применялся для описания задачи проектирования устройства управления, которое должно было максимизировать заданную характеристику поведения динамической системы во времени. Один из подходов к решению этой задачи был разработан в середине 1950-х годов Ричардом Беллманом и другими учеными путем обобщения теории Гамильтона–Якоби, созданной в XIX веке. В этом подходе понятия состояния динамической системы и функции стоимости, используются для вывода функционального уравнения – уравнение Беллмана. Класс методов решения задач оптимального управления путем решения уравнения Беллмана называется динамическим программированием [9]. В работе [10] описана дискретная стохастическая версия задачи оптимального управления, известная под названием «марковский процесс принятия решений» (МППР, англ. MDP). В работе Ховарда Роналда [11] предложен метод итерации по стратегиям для МППР. Все это – существенные элементы, лежащие в основе теории и алгоритмов обучения с подкреплением. Общепризнанно, что дина-

мическое программирование – единственный практически применимый способ решения общих стохастических задач оптимального управления. Большим недостатков динамического программирования является «проклятием размерности», т. е. требования к вычислительной мощности растут экспоненциально с ростом числа переменных состояния. Тем не менее динамическое программирования гораздо более эффективно и распространено, чем любой другой общий метод.

Развитие оптимального управления и обучения с подкреплением происходило независимо, так как данные области ставят перед собой разные цели. Так же влияет тот факт, что динамическое программировании представляется как пакетный метод вычислений, который сильно зависит от наличия точной модели системы и аналитических решений уравнения Беллмана. К тому же простейшая форма динамического программирования – вычисление, происходящее в обратном направлении по времени, поэтому трудно понять, как его можно применить в процессе обучения, который по необходимости протекает в прямом направлении.

Некоторые из первых работ по динамическому программированию, содержат в себе и идеи обучения например [12]. В работе [13] приводятся явные аргументы в пользу более тесной связи между динамическим программированием и методами обучения и доказывается, что динамическое программирование имеет прямое отношение к пониманию работы нейронов и когнитивных механизмов.

Явная связь методов динамического программирования с обучением для дискретных стахостических систем отражена в работе Криса Уоткинса 1989 года [14], в которой обучение с подкреплением изложено с позиций формализма МППР. Методы оптимального управления и обучения с подкреплением активно разрабатываются многими исследователями, в особенности Димитрием Бертсеркасом и Джоном Цициклисом, которые предложили термин «нейродинамическое программирование», описывающий комбинацию динамического программирования с нейронными сетями [6]. Еще один термин, широко употребляемый в настоящее время, -– «приближенное (адаптимное) динамическое программирование».

С одной стороны, почти все традиционные методы требуют полного знания об управляемой системе, кажется не совсем верно утверждать, что они так же относятся к обучению с подкреплением. С другой стороны, многие алгоритмы динамического программирования инкрементные и итеративные.

Как и методы обучения, они постепенно приходят к правильному ответу путем последовательных приближений.

1.4 Примеры применения обучения с подкреплением

Первоначально методы RL применялись только к простым задачам, но применение глубоких нейронных сетей позволило применять RL для систем другого уровня сложности. Сейчас RL применяется в самых разных областях: робототехника, финансы, автономные транспортные средства, медицина и здравоохранения, оптимизация процессов и обнаружение неисправностей. Ниже рассмотрены основные области применения RL:

- Промышленная робототехника – активно развивающаяся область применения RL, поскольку является естественным внедрением этой парадигмы в практику [15]. Например, использование глубокого обучения и обучения с подкреплением позволяет обучать роботов, способных захватывать различные объекты - даже те, которые не видны во время обучения. Или другой пример – обучение работа повторять команды за человеком, выполняя перемещение объектов [16]. Данные приложения могут быть использованы при сборке продуктов на сборочной линии.
- Здравоохранение. RL в здравоохранении относится к методам динамического лечения (англ. Dynamic Treatment Regimes), так как алгоритмы RL позволяют находить решения для оптимального лечения пациентам в каждый момент времени. Например, вход алгоритма – набор клинических наблюдений и оценок пациента, выход - варианты лечения для каждого этапа.[17]
- Обработка естественного языка. RL активно применяется при решении таких задач, как построение диалоговых систем [18], резюмированние и сокращение текстов [19] машинный перевод и др.
- Проектирование архитектуры глубоких нейронных сетей. С одной стороны, применение RL для подбора архитектуры – вычислительно затратное решение, но такой подход позволяет создавать лучшие архитектуры глубоких нейронных сетей.
- Автономные транспортные средства. Алгоритмы RL используются как для организации дорожного движения - автономные светофоры, так и для решения задач автономного вождения, например, оптимизация

траектории движения, динамическое определение пути, смена полосы движения, парковка [20].

– Оптимизация производственных процессов. RL применяется для прогнозирования технического обслуживания, диагностики оборудования режиме реального времени и управлять производственной деятельностью, а так же для оптимизации энергопотребления.[21] Например, компания Royal Dutch Shell применяет RL в задачах по разведке и бурению. Алгоритмы RL, обученные на исторических данных бурения, а также дообученные на физических моделях, используются для управления газовыми буровыми установками при их движении по геологической среде.

1.5 Классификация алгоритмов обучения с подкреплением

В данном разделе приведена основополагающая классификация алгоритмов RL. На рис.1.3 представлена лишь одна из возможных таксономий алгоритмов RL. Одна из наиболее важных классификаций основана на наличии доступа или возможность исследования агентом модели среды.

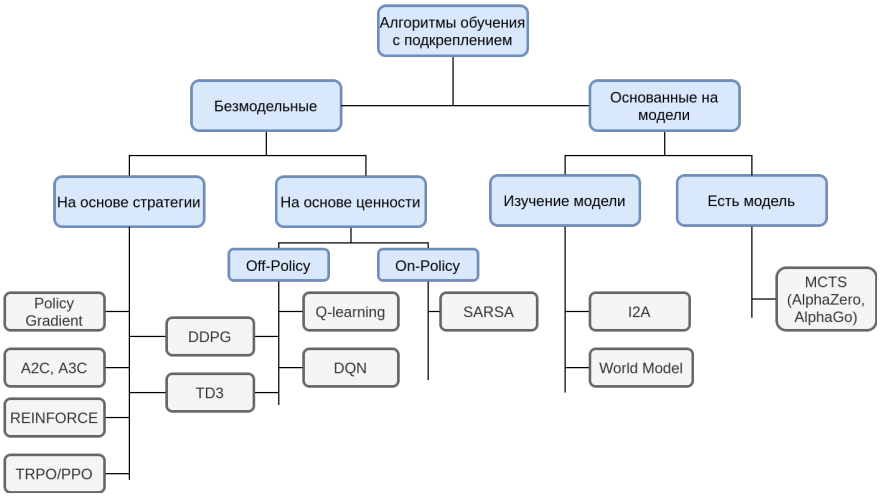


Рисунок 1.3 — Классификация алгоритмов Rl

Алгоритмы модельно-ориентированного обучения (англ. model-based RL) на первом этапе решают задачу идентификации системы либо используют уже составленную модель для решения задач управления. Безмодельное обучение (англ. model-free RL) направленно на прямой поиск управления на основе наблюдений и действий. Другими словами, безмодельные методы направлены на решение задач управления путем исследования системы и улучшения стратегий на основе прошлых вознаграждений и состояний. Модельные алгоритмы

RL имеют такой же недостаток, как и классические подходы в управлении – сложность получения достоверной модели среды (объекта).

Безмодельные методы в свою очередь делятся на два подхода: оптимизация политики и на основе функции ценности. Алгоритмы градиента стратегии (англ. Policy-based) , использую оценки градиента функционала по параметрам политики, что означает, что каждое обновление использует данные, полученные на последней политики. Алгоритмы на основе ценности (англ. value-based) позволяют получать стратегию неявно через теорию оценочных функций. В наиболее часто встречающейся постановке для аппроксимации функции применяются глубокие нейронные сети и другие современные подходы, которые позволяют справиться с высокой дисперсией и общей неустойчивостью. Стоит отметить, что алгоритмы градиента стратегии применимы для сред с непрерывной областью действий, тогда как алгоритмы функций ценности применяются для дискретных систем. Поскольку такие алгоритмы используют целевую функцию основанную на уравнении Беллмана. Модельные и безмодельные алгоритмы не лишены недостатков, поэтому ведется активное развитие гибридных безмодельных алгоритмов RL.

Наколенные данные от эксперта, то есть записи взаимодействия со средой некоторой стратегии, не обязательно оптимальной, могут упростить задачу поиска условного оптимального решения ряда алгоритмов RL. Исходя из способности алгоритма использовать опыт взаимодейтсвия со средой произвольной стратегии, выделяют on-policy и off-policy алгоритмы. Алгоритм off-policy способен использовать для обучения опыт взаимодействия произвольной стратегии. То есть появляется возможность проводить очередной шаг обучения на произвольных траекториях, сгенерированных произвольными стратегиями. Для алгоритма on-policy на очередной итерации требуется опыт взаимодействия конкретной, предоставляемой самим алгоритмом, стратегии. В то же время, накопленные данные от эксперта могут быть использованы для инициализации on-policy алгоритмов. Важно, что off-policy алгоритмы способны на данных произвольного эксперта условно сойтись к оптимуму при достаточном объёме и разнообразии экспертной информации, не требуя дополнительного взаимодействия со средой.

1.6 Методы и алгоритмы обучения с подкреплением

Динамическое программирование. Динамическое программирование (ДП) – один из фундаментальных модельных методов в RL, который основан на урав-

нении Беллмана. Базовыми алгоритмами ДП являются итерирование стратегий (англ., Policy iteration, PI) и итерирование ценности (англ., Value iteration, VI). Алгоритм PI во время одной итерации вычисляет функцию награды для текущей стратегии, потом улучшает эту стратегию, то есть оценивание и улучшение стратегии выполняется циклически, тогда как в алгоритме VI – оценивание и улучшение стратегии выполняется за одно обновление. В литературе по RL процесс вычисления функции ценности состояния V^{π} для произвольной стратегии π называют оценкой стратегии (англ. Policy Evaluation) или задачей прогнозирования.Функция ценности для стратегии вычисляется для того, чтобы далее с помощью нее найти новую, улучшенную стратегию. Процесс формирования новой, улучшающей исходную, стратегии называется улучшением стратегии (англ., Policy Improvement). Так, вычисленное значение функции ценности позволяет улучшить стратегию, в частостии, жадной стратегией:

$$\pi' = argmax_a Q_{\pi}(s,a) = argmax_a \sum_{s',r} p(s'|s,a)[r + \gamma V_p i(s')]$$

Алгоритм итерация по стратегиям представлен на рис.1.4. Алгоритм итерация по ценности представлен на рис.1.5.

1. Инициализация $V(s) \in \mathcal{R}$ и $\pi(s) \in \mathcal{A}(s)$ для всех $s \in \mathcal{S}$
2. Оценка стратегии Пока $\Delta < \theta$
  $v := V_k(s)$
  $V_{k+1}(s) := \sum_{s',r} p(s'|s,\pi(a))[r + \gamma V_k(s')]$
  $\Delta := \max(\Delta, |v - V_{k+1}(s)|)$
Конец цикла
3. Улучшение стратегии
Стратегия устойчива:=истинно
Цикл по $s \in \mathcal{S}$
  $b := \pi(s)$
  $\pi(s) = argmax_a Q_{\pi}(s,a) = argmax_a \sum_{s'r} p(s'|s,a)[r + \gamma V_{k+1}(s)]$
  Если $b \neq \pi(s)$ тогда
  Стратегия устойчива:=ложно
Конец цикла
Если Стратегия устойчива тогда
  выход
иначе переход к шагу 2

Рисунок 1.4 — Алгоритм итерирование стратегии

Недостатком методов ДП является то, что они требуют вычислений в каждом состоянии. В случае, если множество состояний велико, это может

1. Инициализация $V_0(s)$ произвольно для всех $s \in \mathcal{S}$
2. Оценка стратегии
Пока $\Delta < \theta$
$v := V_k(s)$
$V_{k+1}(s) := \sum_{s',r} p(s'|s,\pi(a))[r + \gamma V_k(s')]$
$\Delta := \max(\Delta, |v - V_{k+1}(s)|)$
Конец цикла
3. Улучшение стратегии
$\pi(s) = argmax_a \sum_{s'r} p(s'|s,a)[r + \gamma V_{k+1}(s)]$

Рисунок 1.5 — Алгоритм итерирование ценности

потребовать больших вычислительных ресурсов. Асинхронные алгоритмы ДП обновляют оценки значений только подмножества состояний, а не всего набора на каждой итерации оценки политики. Такие алгоритмы обновляют значения состояний в любом порядке, используя значения других доступных состояний. Как было сказано ранее, PI состоит из двух чередующихся взаимодействующих процессов: оценивание стратегии, которое обеспечивает совместимость функции ценности с текущей стратегией и улучшение стратегии, который делает стратегию жадной к текущей функции ценности. В VI между двумя улучшениями стратегии только одна итерация оценивания стратегии. В асинхронных методах ДП процесс оценивания и учлучшение стратегии разбивается на более крупные чередующиеся итерации.

Одни из главных недостатков ДП – экспоненциальное возрастание сложности вычислений с увеличением числа состояний («Проклятье размерности») и необходимость модели объекта. Несмотря на это, идеи оценки стратегии и итерирования по стратегии лежат в основе почти всех алгоритмов ОП.

В отличии от ДП методы Монте-Карло (МК) не предполагают полного знания о модели. Такие методы предполагают наличие данных – набор состояний, действий и наград, полученных при взаимодействии с средой. Для применения методов МК требуется, чтобы задача была эпизодическая, так как методы МК инкриментны на уровне эпизодов, а не на уровне шагов. В отличии от ДП оценка для каждого состояния в методах МК – независимы. Подробно не будет останавливаться на методах МК и ДП.

Обучение на основе временных различий. Обучение на основе временных различий (англ., Temporal Difference, TD) или TD-обучение – это метод обучения с подкреплением без использования моделей, где агент обучается на каждом

отдельном действии, которое он предпринимает и при этом обновляет знания агента на каждом временном шаге (действии), а не в каждом эпизоде.TD-методы совмещают в себе идеи методов МК и ДП. Коррекция в простейшем TD-методе производится путем улучшения ценности на небольшую величину в направлении оптимального значения:

$$V(s_t) = V(s_t) + \alpha[r + \gamma V(s_{t+1}) - V(s_t)]$$

где α – параметр, который определяет степень изменения ценности состояния при каждом обновлении. Если $a = 0$, то ценность состояния не изменяется. Если же $a = 1$, то ценность состояния будет равна $r + \gamma V(s_{t+1})$ – старая ценность затирается. К алгоритмам TD-методов относятся: Q-learning, SARSA, R-learning, методы исполнитель-критик

На рис.1.6 – 1.7 представлены алгоритмы Q-learning, SARSA. Алгоритм Q-learning – это табличный безмодельный off-policy алгоритм RL. Это метод основан на временных разностей для вычисления оптимальной Q-функции с $\varepsilon-$жадной стратегией исследования, то есть агент выбирает случайное действие с вероятностью ε, но использует известное лучшее действие. Алгоритм наследует от TD-обучения характеристики одношагового обучения, такие как возможность обучаться на каждом шаге и способность обучаться на опыте, не имея модели окружающей среды. Q-learning отличается от SARSA прежде всего тем, что это алгоритм с разделенной стратегией. Разделенная стратегия означает, что обновление производится независимо от того, какая стратегия использовалась для накопления опыта, то есть алгоритмы с разделенной стратегией могут использовать прежний опыт для улучшения стратегии. Стратегия, которая применяется для улучшение стратегии – целевая, а стратегия для взаимодействия с окружающей средой – поведенческая. Алгоритм состоит из следующих шагов: 1. Инициализация Q-таблицы с нулевыми значениями, то есть в начальный момент времени все стратегии равновероятны и равноценны. 2. Выбор действия с наибольшей ценностью. 3. Отправка на вход среды выбранного действие, на выходе получаем вознаграждение. 4. Обновление Q-таблицы с учетом полученного вознаграждения. Гиперпараметрами алгоритма являются $\alpha \in (0,1]$ – параметр экспоненциального сглаживания, $\varepsilon > 0$ – параметр исследования.

Разница между этими двумя алгоритмами в том, что SARSA выбирает действие, соответствующее той же текущей политике, и обновляет его Q-значения, тогда как Q-обучение выбирает жадное действие, то есть действие, которое

Инициализация $Q(s,a)$ произвольно для всех $s \in \mathcal{S}$, $\alpha \in \mathcal{A}$
Цикл по эпизодам
  Инициализация s
  Цикл по шагам эпизодов k
  Выбор a_k: с вероятностью ε принимаем $a_k \sim Uniform(\mathcal{A})$
  иначе $a_k = argmaxQ(s_k, a_k)$
  Выполнение действие a_k
  Нахождение r_k, s_{k+1}
  Обновление
  $Q(s,a) \leftarrow Q(s,a) + \alpha[r_k + \gamma \max_{a_{k+1}} Q(s_{k+1}, a_{k+1}) - Q(s_k,a_k)]$
  Конец цикла
Конец цикла

Рисунок 1.6 — Алгоритм Q-learning

Инициализация $Q(s,a)$ произвольно для всех $s \in \mathcal{S}$, $\alpha \in \mathcal{A}$
Цикл по эпизодам
  Считывание s_0, находим $a_k \sim Uniform(\mathcal{A})$
  Цикл по шагам эпизодов k
  Нахождение r_k, s_{k+1}
  Выбор a_{k+1}: с вероятностью ε принимаем $a_{k+1} \sim Uniform(\mathcal{A})$
  иначе $a_{k+1} = argmaxQ(s_{k+1}, a_{k+1})$
  Выполненине действия a_k
  Обновление $Q(s,a) \leftarrow Q(s,a) + \alpha[r_k + \gamma Q(s_{k+1}, a_{k+1}) - Q(s_k,a_k)]$
  Конец цикла
Конец цикла

Рисунок 1.7 — Алгоритм SARSA

дает максимальное значение Q для состояния, то есть оно следует оптимальной политике.

Апроксимация V-функции.

Главным недостатком перечисленных выше методов является необходимость хранить в памяти большие объемы данных при увеличении сложности объекта, а так же сложность работы с непрерывным пространством действий. Необходимо ввести аппроксимацию V-функции (Q-функции), что позволит представить функцию в ограниченной области определения, располагая памятью фиксированного объема.Применение аппроксимации функций позволяет заменить пространство состояний набором признаков, порождаемых по исходным со- стояниям

Основная идея аппроксимации функций – воспользоваться набором признаков для оценки значений V-функции (Q-функции). Есть несколько способов

отобразить признаки на значения функции, например линейная аппроксимация, решающие деревья, алгоритм ближайших соседей, искусственные нейронные сети. В случайно линейной аппроксимации функция ценности состояний записывается в виде взвешенной суммы признаков. Как можно ожидать, нейронные сети используются чаще других подходов. В частности, используются глубокие нейронные сети (ГНС).

Вывод по главе 1. В ходе анализа и изучения области обучения с подкреплением, сделаны следующие выводы:

- МППР – нотация для представления задачи обучения с подкреплением;
- Функция ценности и функция ценности действия – основополагающие термины для дальнейших исследований;
- Динамическое программирование рассматривается как один из главных способов обучения агента;
- Область управления и область обучения с подкреплением связаны математической базой, а именно уравнение Беллмана;
- Примемения методов обучения с подкреплением распространено в области рекомендательных систем и игр;

2 МЕТОДЫ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ В ЗАДАЧЕ УПРАВЛЕНИЯ

Обучение с подкреплением предлагает мощные алгоритмы разработки оптимального управления для систем с нелинейностями, со сложной неизвестной стахостической динамикой. Данная глава охватывает подходы RL с точки зрения инженера по управлению. В этой главе приводятся объяснения, как апроксимация позволяет использовать RL с непрерывными состояниями и управляющими воздействиями. Цель данной главы – преодоление рассогласованости терминологии управления и RL.

2.1 Связь нотаций

Анализ динамических систем с использованием механики Лагранжа, Гамильтона, позволяет получить описание системы в форме нелинейных ОДУ или разностных уравнений. В первом случае объект управления описывается уравнением вида:

$$\dot{x} = f(x, u),$$

где $x(t) \in X \subseteq R^n$, $u(t) \in U \subseteq R^m$ – управление.

Показатель качества (критерий оптимальности), по которому проектируется система управления, задается в виде:

$$J = \int_0^{t_f} L(t, x(t), u(t))dt + h(x(t_f), t_f),$$

где t_f – конечное время.

Такое представление системы, в отличии от МППР, имеет непрерывное пространство состояний, непрерывный вход, а так же непрерывность во времени.

Во втором случае система либо дискретна по своей природе, либо приводится к дискретной форме:

$$x_{k+1} = f(x_k, u_k), k = 0, 1,.., N,$$

При этом выполняется свойство марковости, так как состояние в момент времени $k + 1$ зависит только от состояния и входов в момент времени k. При этом x_k и u_k могут принадлежать конечному или счетному множеству, а так же континууму. Показатель качества для дискретных систем имеет вид:

$$J = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k), \tag{2.1}$$

где $g_N(x_N)$ – терминальные (конечные) значения, N – число временных шагов (этапов, стадий)

В некоторых задачах важно получить представление об оптимальном процессе при длительном, практически неограниченном процессе. Отдельно выделяют задачи оптимального управления с бесконечным временем, когда $t_f \to \infty$ или $N \to \infty$. Особенность таких задач связана с тем, что показатель качества представляет собой бесконечную сумму или несобственный интеграл. Одни из наиболее распространенных подходов для задач с бесконечным временем является введение дисконтирования:

$$J = \int_0^{\infty} L(x(t), u(t))dt$$

или в дискретном случае:

$$J = \sum_{k=0}^{\infty} \gamma^k r(x_k,u_k)$$

где $0 < \gamma \leqslant 1$ - коэффициент дисконтирования. Введение коэффициента дисконтирования гарантирует сходимость ряда или интеграла в большинстве

случаев. Физический смысл — штраф или награда в будущем имеют меньшую значимость, чем текущая. Кроме того, введение дисконтирования уменьшает влияние неточности модели на показатель качества.

2.2 Переход между нотациями

Рассмотрим переход от разностных уравнений к МППР и обратно. Пусть задан МППР:

$$p_{ij}(u,k) = P\{x_{k+1} = j | x_k = i, u_k = u\}$$

Соответствующая дискретная система будет $s_{k+1} = w_k$, где

$$P\{w_k = j | x_k = i, u_k = u\} = p_{ij}(u_k)$$

В обратную сторону, пусть задана дискретная система:

$$x_{k+1} = f(x_k, u_k, w_k) \tag{2.2}$$

где $w_k \sim P_k(w_k|x_k,u_k)$ – известно. Тогда

$$p_{ij}(u,k) = P_k\{W_k(i,u,j)|x_k = i, u_k = u\} \tag{2.3}$$

где $W_k(i,u,j) = w|j = f_k(i,u,w)$

Как видно, в обучении с подкреплением обычно в общем случае рассматриваются стохастические системы, однако, для простоты, в данной работе ограничимся рассмотрением только детерминированных систем.

2.3 Дискретные системы

Классическое динамическое программирование.Все задачи ДП направлены на работу динамических систем с дискретным временем. В детерминированных системах x_{k+1} определено. Задача ДП рассматривает динамические системы с дискретным временем вида:

$$x_{k+1} = f_k(x_k, u_k), \tag{2.4}$$

где $k = 0, 1,.., N-1$.

Цель – найти управление, минимизирующее показатель качества:

$$J(x_0; u_0,\ldots,u_{N-1}) = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k), \tag{2.5}$$

Тогда оптимальное значение функционала записывается как:

$$J^*(x_0) = \min_{\substack{u_k \in U_k(x_k) \\ k=0,\ldots,N-1}} J(x_0; u_0,\ldots,u_{N-1}), \tag{2.6}$$

При решении задачи (2.6) методом ДП, поиск последовательности $J_N^*(x_N)$, $J_{N-1}^*(x_{N-1})$,..., $J_0^*(x_0)$ осуществляется следующим образом:

$$J_N^*(x_N) = g_N(x_N) \text{ для всех } x_N$$

$$J_k^*(x_k) = \min_{u_k \in A_k(x_k)} [g_k(x_k, u_k) + J_{k+1}^*(f_k(x_k, u_k))] \text{ для всех } x_k$$

После этого, зная $J_N^*(x_N)$, $J_{N-1}^*(x_{N-1})$..., $J^*(x)$, можно последовательно найти u_0, ..., u_{N-1}:

$$u_k^* \in \arg\min_{u_k \in U_k(x_k^*)} [g_0(a_k^*, x_k) + J_{k+1}^*(f_k(x_k^*, u_k))],$$

$$x_{k+1}^* = f_k(x_k^*, u_k^*),$$

Таким образом, для нахождения u_0, ..., u_{N-1} необходимо вычисление всех $J_k^*(x_k)$. На практике вычисление J_k^* с помощью ДП занимает большое количество времени, поскольку количество x_k и k может быть очень большим. Главным недостатком метода является «проклятие размерности» — сложность вычислений возрастает с увеличением размерности задачи. Помимо этого формирование управляющего воздействия методом ДП протекает не в режиме реального времени (режим off-line). С ростом сложности системы возникает необходимость хранения огромного количества данных, увеличение доли шумов. Решение задачи оптимального управления методом ДП, предполагает знание модели объекта, в следствии чего качество регулятора зависит напрямую от качества построения математической модели объекта. Отсутствие универсального алгоритма, который был бы пригоден для решения всех задач рассматриваемого класса. Алгоритмы ДП объединены общей идеей, но в каждом конкретном случае должны формироваться применительно к специфике прикладной задачи, поэтому отсутствие универсального алгоритма – еще один недостаток методов оптимального управления, таких как ДП.

2.4 Бесконечное время и методы

Рассмотрим стационарную систему $x_{k+1} = f(x_k, u_k)$ на бесконечном интервале времени:

$$J = \sum_{k=0}^{\infty} g(x_k, u_k) \tag{2.7}$$

Необходимо найти закон управления $\mu(x)$, который является также стационарным, что значительно упрощает синтез и реализацию. С другой стороны,

в случае рассмотрения систем на бесконечном интервале времени мы не можем итеративно двигаться назад, начиная с терминального состояния, как мы это делали в случае конечного времени. Обозначим как $J_\mu(x_0)$ значение функционала при законе управления μ и начальном условии x_0 . Тогда:

$$J_\mu(x_k) = \sum_{t=k}^{\infty} g(x_t, \mu(x_t)) = g(x_k,\mu(x_k)) + J_\mu(x_{k+1}), J_\mu(0) = 0$$

произведена замена бесконечного суммирования в 2.7 на решение разностного уравнения. Отсюда можно получить следующий алгоритм решения рассматриваемой задачи, называемый итерации по ценности (VI):

$$J_0(x) = 0$$
$$J^*_{k+1}(x) = \min_{u_k \in U(x)} [g_k(x_k, u_k) + J^*_k(f_k(x_k, u_k))], k = 0, 1, \ldots$$

Рассмотренный выше алгоритм для конечного случая — тоже VI. Тот же самый алгоритм, переписанный в более привычном для RL виде:

Итерация по ценности:

1. Инициализация. Выбор произвольного закона управления $\mu_0(x)$, $k = 0$ и $J_0(s)$
2. Шаг policy evaluation (оценка политики).

$$J_{k+1}(x) := g(x, \mu_k(x)) + J_k(f(x,u))$$

3. Шаг policy improvement (обновление политики). Обновление закона управления

$$\mu_{k+1}(x) = \arg\min_{\mu(\cdot)}(g(x, \mu_k(x)) + J_{k+1}(f(x,u)))$$

Другой алгоритм – итерация стратегии (PI):

1. Инициализация. Выбор произвольного закона управления $\mu_0(x)$, $k = 0$ и $J_0(s)$
2. Шаг policy evaluation (оценка политики).

$$J_{\mu_{k+1}}(x) = g(x, \mu_k(x)) + J(f(x,\mu_k(x)))$$

3. Шаг policy improvement (обновление политики). Обновление закона управления

$$\mu_{k+1}(x) = \arg\min_{\mu(\cdot)}(g(x, \mu_k(x)) + J(f(x,\mu_k(x))))$$

Рассмотрим вариант PI алгоритма, в котором шаг оценки стратегии производится неточно, в частности, алгоритм начинается с некоторого J_0 и генерирует последовательность пар функций стоимости и закона управления J_k, μ_k следующим образом: учитывая J_k, мы генерируем μ_k в соответствии с:

$$\mu_k(x) = \arg\min_{\mu(\cdot)}(g(x,u) + J_k(f(x,u)))$$

и тогда получаем J_{k+1} при $m_k \geqslant 1$

$$J_{\mu_{k+1}}(x_0) = J_k(x_{m_k}) + \sum_{t=0}^{m_k-1} g(x_t,\mu_k(x_t)))$$

где x_t – последовательность, сгенерированная с использованием μ_k и начиная с x_0, m_k - произвольные положительные целые числа. При $m_k = 1$ алгоритм эквивалентен алгоритму VI, а частный случай $m_k = \infty$ – алгоритму PI. Такой алгоритм в RL называется Обобщенная итерация стратегий (Generalized Policy Iteration, GPI) – одна итерация решения m-шагового уравнения Беллмана чередуется с шагом обновления политики). Алгоритм GPI при любом m сходится к оптимальной стратегии и оптимальной оценочной функции.

2.5 Приближенное динамическое программирование

Эти методы широко распространены под названием приблежённое или адаптивное динамическое программирование [22] или нейродинамическое программирование [bertsekas1996neuro].

Для преодоления проклятия размерности применяют различные аппроксимации. Аппроксимировать можно J_k^* (аппроксимация in value space) и или непосредственно управление (аппроксимация in policy space). Здесь важно привести еще одну важную классификацию в зависимости от то, когда вычисляется управление: оффлайн — вычисление управления производится до начала процесса управления, онлайн — вычисляем управление непосредственно в процессе эксплуатации. Так, аппроксимация in policy space — оффлайн метод, аппроксимация in value space — в основном онлайн метод.

В случае аппроксимации «in policy space» мы ищем управление из заданного параметрически семейства $\mu_k(x_k, r_k)$, где r_k – параметры. Если осуществили аппроксимацию, то управление легко посчитать: $u_k = \widetilde{\mu}_k(x_k, r_k)$, то есть такой подход можно использовать для аппроксимации известного закона с целью удобного онлайн использования. В общем случае собираются пары (x_k^s, u_k^s), $s = 1,\ldots, q$,

такие что u s_k — хорошее управление для данного x_k^s . Далее определяются параметры, например, методом наименьших квадратов.

В случае аппроксимации «in value space» аппроксимируем J_k^* функцией $\widetilde{J}_k$. Тогда управление находится алгоритмом ДП, минимизируя функционал на конечном горизонте плюс аппроксимация оптимальной будущей стоимости $\widetilde{J}$. Значение $\widetilde{J}$ может быть получено разными способами: симуляция методами Монте-Карло, эвристики и др.

Методы нахождения $\widetilde{J}_k$ можно условно классифицировать на 4 группы:

1. *Аппроксимация задачи*. Нахождение функции $\widetilde{J}_k$ в более простой задаче. Вводится ряд допущений – уменьшение размера пространста состояния, не учитываются нелинейности и т.д. Частный случай – агрегация;
2. *Приближенная онлайн оптимизация*. Здесь применяются окононные алгоритмы: (Алгоритм Rollout algorithms, Model Predictive Control и т.д);
3. *Параметрическая аппроксимация*. Параметризуем $\widetilde{J}_k(x_k, r_k)$ некоторой параметрической функцией, где параметры определены используя данные и, например, нейронные сети;
4. *Агрегация*. Эта группа методов обычно предполагает разбиение пространства состояний. Более того, агрегация может применяться вместе с методами (1-3) и служить начальным приближением для решения задачи другим методом.

В данной работе остановимся на 2 и 3 подходе.

Приближенная онлайн оптимизация, метод MPC

Идея метод состоит в том, чтобы не параметризовать политику управления параметрами W (как это будет далее показано в RL), а вместо этого оптимизировать входные данные управления u_k непосредственно на конечном горизонте T. Чтобы учесть усеченный горизонт, мы можем использовать функцию ценности при некотором законе управлении, чтобы чтобы приблизить стоимость за пределами T шагов. Таким образом, политика управления определяется выражением

$$\mu_(x_k) = \arg\min_{\mu(\cdot)}(g(x_k, u) + J(f(x,\mu_k(x))))$$

Модельное прогнозирующее управление (англ. model predictive control, MPC) – метод управления который позволяет расчитывать управляющее воздействия на основе математической модели объекта, прогнозируя значения

переменных состояния и выхода, решая задачи оптимизации в реальном времени, при этом учитывая ограничения (рис.2.1). Регулятор минимизирует ошибку между предсказанным и фактическим значением по горизонту управления. В отличии от классических подходов управления MPC позволяет управлять близко к ограничениям, что ведет к более устойчивой работе.

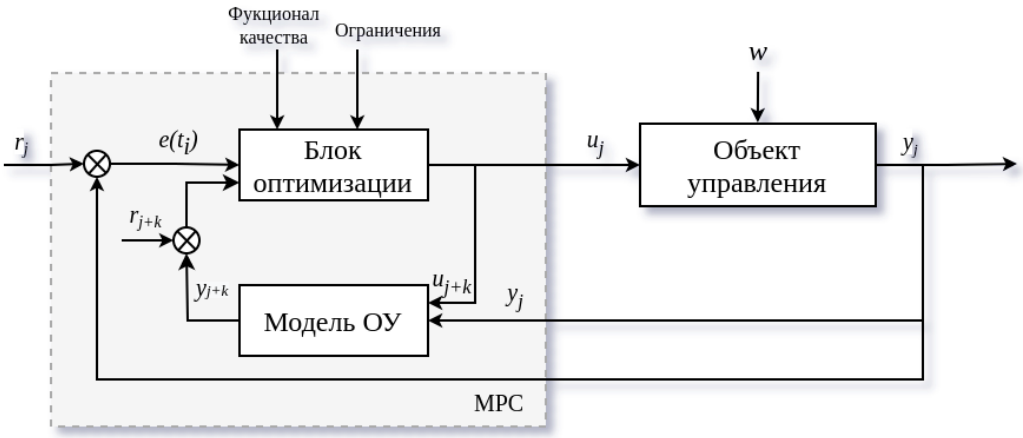


Рисунок 2.1 — Структурная схема системы управления с прогнозирующими моделями

Регулятора на основе MPC получили широкое распространение в силу развития вычислительных мощностей. Отметим ключевые моменты, которые отличают подход MPC и RL: а) Так как на каждом шаге решается задача оптимизации, качества управления во многом будет зависеть от точности модель процесса. В то время как регуляторы RL не требуют какого-либо предварительного доступа к модели процесса: б) Промышленные системы управления подвержены эксплуатационным ограничениям, которые должны всегда соблюдаться. MPC устраняет эти ограничения, явно включая их в задачу оптимизации, решаемую на каждом этапе. В контроллере RL ограничения могут быть встроены непосредственно в функцию перенастройки или реализованы посредством отсечения градиента. Второй подход более мягкий – нарушение ограничения может произойти во время обучения, но оно не приветствуется из-за огромного штрафа в сигнале вознаграждения. Надежные механизмы для безопасной работы полностью обученной системы и, действительно, для безопасной работы во время онлайн-обучения, являются открытой задачей для RL регулирования. в) Адаптация: со временем меняются параметры систем. Поддержание производительности требует реагирования как на изменения, так и на неизвестные нарушения. Традиционные системы на основе MPC включают механизмы для выявление несоответствия модели и объекта, но изменение параметров будет

сопровождаться повторной идентификацией модели процесса. Этот процесс, который требует одновременной оценки состояния и параметров, может быть сложным и дорогостоящим. Некоторые недавние варианты MPC, такие как робастный MPC и стохастический MPC, учитывают неопределенности модели, встраивая их непосредственно в задача оптимизации. Регуляторы на основе RL автоматически корректируют параметры в сетях актеров и исполнителях. Это обеспечивает регулятор RL свойством самонастройки, так что процесс остается субоптимальным по отношению к выбранному критерию вознаграждения.

Рассмотрим *параметрическую аппроксимацию*. Для этого представим оценку функции стоимости в следующем виде:

$$\widetilde{J}_{\mu}(x) = W^T \varphi(x)$$

где базисный вектор $\varphi(x) = [\varphi_1(x), \varphi_2(x), \ldots \varphi_L(x)]$, $W \in \mathrm{R}^L$ –вектор настраиваемых параметров. Для наивной настройки весов W требуется для каждого x_k вычислять $\widetilde{J}_{\mu}(k)$, то есть вычисления производятся оффлайн. Для устранения этого недостатка вводится понятие временного различия (англ, Temporal Differences, TD):

$$e_k = g(x_k, \mu(x_k)) + J_{\mu}(x_{k+1}) - J_{\mu}(x_k)$$

Если $e_k = 0$ для всех x_k , то это просто уравнение Беллмана. TD ошибка представляет собой ошибку между предсказанной и действительной наградой за действие. Это равенство должно выполняться для всех x_k в любой момент времени k, поэтому можно записать онлайн версии рассмотренных ранее алгоритмов.

Онлайн алгоритм итерации по стратегии (онлайн PI)

1. *Инициализация*. Выбор любого допустимого закона управления $\mu_0(x_k)$
2. *Шаг оценки стратегии*. Оценка параметров W_{j+1}:

$$W_{j+1}^T(\varphi(x_k) - \gamma\varphi(x_{k+1})) = r(x_k, h_j(x_k))$$

3. *Шаг улучшения стратегии*. Обновление закона управления:

$$\mu_{j+1}(x_k) = \arg\min_{\mu(\cdot)}(g(x_k, \mu(x_k)) + W_{j+1}^T \varphi(x_{k+1}))$$

Онлайн алгоритм итерации по ценности. (онлайн VI)

1. *Инициализация*. Выбор любого закона управления $\mu_0(x_k)$, не обязательно допустимой.

2. *Шаг оценки стратегии*. Оценка параметров W_{j+1}:

$$W_{j+1}^T \varphi(x_k) = g(x_k, \mu_j(x_k)) + W_j^T \gamma \varphi(x_{k+1}) \tag{2.8}$$

3. *Шаг улучшения стратегии*. Обновление закона управления:

$$\mu_{j+1}(x_k) = \arg\min_{\mu(\cdot)}(g(x_k, \mu(x_k)) + W_{j+1}^T \varphi(x_{k+1}))$$

В обоих алгоритмах на шаге policy evaluation, параметры W_{j+1} можно искать методом наименьших квадратов. Однако для этого требуется обновление закона управления производить не раньше, чем через L шагов с этим законом. Так как $W_{j+1} \in R^L$, нужно не менее L уравнений для оценки этого вектора. В момент времени x_{k+1} имеем набор $(x_k, \mu_(x_k), x_{k+1}, g(x_k, \mu(x_K)))$, то есть одно уравнение. Однако обычно вектор параметров W_{j+1} ищут рекурсивным методом наименьших квадратов или градиентным спуском до сходимости, а затем обновляют закон управления.

На шаге обновления закона управления необходимо осуществлять поиск функции, что плохо, поэтому необходимо параметризовать. Параметризовав шаг 3,получаем схему исполнитель-критик.

Часто реализация обучения с подкреплением осуществляется с использованием двух НС: одна в качестве критика, а другая в качестве исполнителя (рисунок 1). В этой системе управления критик и исполнитель настраиваются в режиме онлайн с использованием наблюдаемых данных. Критик и исполнитель настраиваются последовательно – веса одной НС остаются постоянными, а веса другой настраиваются до сходимости. Эта процедура повторяется до тех пор, пока обе НС не сойдутся. Затем регулятор определяет оптимальное значение управления в режиме онлайн. Таким образом, это онлайн оптимально-адаптивная система управления, в которой параметры аппроксимированного функционала настраиваются онлайн, обеспечивая сходимость к оптимальному управлению.

2.6 Q-фактор

На данном этапе получили онлайн версии с параметризацией, но в описанных выше алгоритмах, даже если мы знаем J, требуется знать функцию $x_{k+1} = (x_k, u_k)$. То есть все еще требуется построение модели. Поэтому вводим понятие Q-фактора.

Можно ввести следующее обозначение:

$$Q_k^*(x_k, u_k) = g_k(x_k, u_k) + J_{k+1}^*(f_k(x_k, u_k))$$

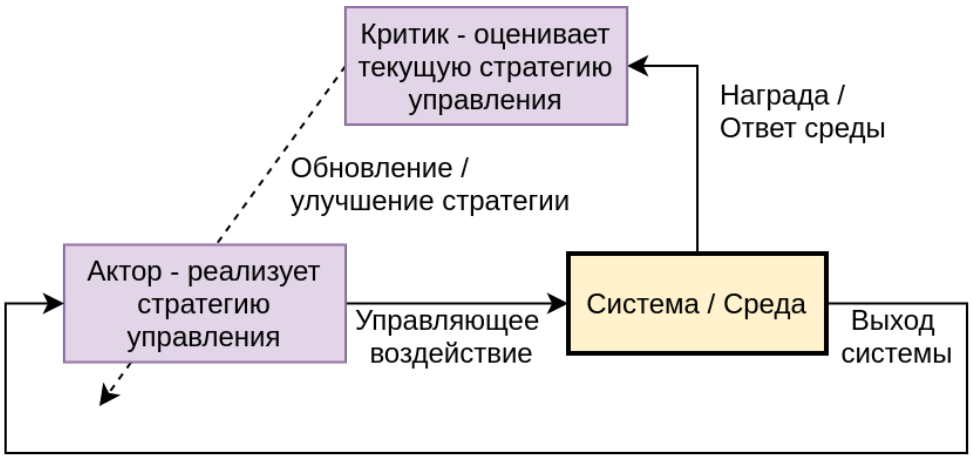


Рисунок 2.2 — Структурная схема системы управления с прогнозирующими моделями

Тогда

$$u_k^* \in \underset{u_k \in U_k(x_k^*)}{\arg\min} Q_k^*(x_k^*, u_k)$$

Оптимальная стоимость записфвается как:

$$J_k^*(x_k) = \min_{u_k \in U_k(x_k^*} Q_k^*(x_k, u_k)$$

Тогда алгоритм ДП можно переписать через Q-фактор:

$$Q_k^*(x_k, u_k) = g_k(x_k, u_k) + \min_{u_{k+1} \in U_{k+1}(f_k(x_k, u_k))} Q_{k+1}^*(f_k(x_k, u_k), u_{k+1})$$

J имеет преимущество, что если система меняется, то работает (online replanning).

Оптимальные методы управления обычно направлены на поиск закона управления в автономном режиме при условии наличия базовых динамических моделей. Когда лежащие в основе модели недоступны или известны только частично, используются подходы адаптивного управления в онлайн режиме [23]. Таким образом, можно сказать, что онлайн-методы RL – это методы адаптивного оптимального управления. Так как RL направлено на поиск оптимального (субоптимальныого) закона управления в режиме онлайн с использованием измерений в реальном времени без знания модели системы.

Анализ устойчивости оптимальных и адаптивных методов управления имеет решающее значение в потенциально опасных приложениях, например, при взаимодействии человека и робота, автономная робототехника или управление электростанциями. При разработке оптимальных и адаптивных законов управления на первом этапе выбираются детерминированные настройки. Впоследствии,

потенциальные неопределенности (шумы и возмущения) которые не учтнены в детерминированной настройке, исследуются с использованием различных инструментов устойчивости, чтобы сделать вывод об устойчивости, например, о локальной, асимптотической, экспоненциальной устойчивости.

В отличие от стандартных подходов в управлении, которые с самого начала ставят требования полной устойчивости, подходы RL требуют дополнительных гарантий стабильности и надежности. Стоит отметить, что здесь нас интересует устойчивость с точки зрения управления, то есть устойчивость замкнутой системы в результате управления. Тогда как в ИИ, термин устойчивость относится к сходимости алгоритмов обучения (к асимптотическому поведению с точки зрения управления). Это подчеркивает философское различие между искусственным интеллектом и областью автоматического управления. Исследователи ИИ сосредотачиваются на производительности с точки зрения совокупного вознаграждения, где вознаграждение может иметь любое значение и рассматривается как часть задачи. Алгоритмически это означает, что учитывается только сходимость (качественная / асимптотическая или количественная через скорости сходимости) процесса обучения к почти оптимальному решению, в то время как допустимые границы в процессе обучения, которые необходимы для обеспечения устойчивости замкнутого контура, не учитываются. Иногда это допустимо в силу характера некоторых приложений ИИ (например, для обработки видео или настольных игр). Цели инженеров по управлению направлены на соблюдение устойчивости, так что даже при использовании оптимального управление главная - и часто единственная - роль фунции стоимости заключается в уточнениях требований устойчивости, например, в стандартных подходах к MPC.

2.7 Обучение с подкреплением для непрерывных систем

Для систем с непрерывным временем применение методов обучения с подкреплением сложнее, чем для систем с дискретным временем.

Рассмотрим нелинейную динамическую систему с непрерывным временем:

$$\dot{x} = f(x) + g(x)u$$

с состоянием $x(t) \in R^n$, управляющим входом $u(t) \in R^m$, Предполагается, что система стабилизируема на Ω – существует непрерывное управляющее воздействие $u(t)$ такое, что замкнутая система асимптотически устойчива на Ω.

Задана мера производительности или функции стоимости, связанной с политикой управления обратной связью $u = \mu(x)$ как:

$$J^{\mu}(x(t)) = \int_{t}^{\infty} r(x(\tau),u(\tau))d\tau$$

где награда $r(x,u) = Q(x) + u^T Ru$ и $Q(x)$ – положительно определенная. Уравнение Беллмана определяется на основе Гамильтонина:

$$H(x, \mu(x), \nabla J^{\mu}) = r(x, \mu(x)) + (\nabla J^{\mu})^T (f(x) + g(x)\mu(x)))$$

где ∇J^{μ} - градиент функции стоимости J_m по отношению к x Отметим проблемы с непрерывными системами: сравним Гамильтониан для непрерывных систем Беллмана с Гамильтонианом для дискретных систем. Первый содержит полную динамику системы $f(x) + g(x)u$, а Гамильтониан для дискретных систем - нет. Это означает, что нет возможности использовать уравнение Беллмана в качестве основы для обучения с подкреплением, если не известна полная динамика.

Было проведено несколько исследований обучения с подкреплением, где применялся метод Эйлера для дискретизации уравнения Беллмана

$$0 = r(x,\mu(x)) + (\nabla V^{\mu})^T (f(x) + g(x)\mu(x)) = r(x, \mu(x)) + V^{\mu}$$

$$0 = r(x_k, u_k) + \frac{V^{\mu}(x_{k+1} - V^{\mu}(x_k)}{T} =$$
$$= \frac{r_S(x_k, u_k}{T} + \frac{V^{\mu}(x_{k+1} - V^{\mu}(x_k)}{T}$$

с периодом T так, чтобы $t = kT$. Награда для дискретной формы $r_S(x_k, u_k) = r(x_k, u_k)T$ задается через умножение на период T.

Дискретизированное уравнение Беллмана имеет тот же вид, что и дискретное уравнение Беллмана . Следовательно, могут применяться все вышеописанные методы обучения с подкреплением. *Алгоритм итерации по стратегии для непрерывных систем*

– *Инициализация*. Выбор любой допустимой политики $\mu(0)(x)$.
– *Шаг оценки стратегии*. Решение $V^{\mu(i)}(x(t))$:

$$V^{\mu(i)}(x(t)) = \int_{t}^{t+T} r(x(s), \mu(i)(x(s))ds + V^{\mu(i)}(x(t+T)) \qquad (2.9)$$

где $V^{\mu(i)}(0) = 0$

– *Шаг улучшение стратегии*. Определение улучшенной стратегии:

$$\mu^{i+1} = \arg\min_{u}[H(x, u, \nabla V_x^{\mu^{(i)}})]$$

Алгоритм итерации по ценности для непрерывных систем

– *Инициализация*.Выбор любой политики управления $\mu^(0)(x)$, необязательно допустимой.

– *Шаг оценки стратегии*.

$$V^{\mu^(i)}(x(t)) = \int_t^{t+T} r(x(s), \mu^(i)(x(s))ds + V^{\mu^(i+1)}(x(t+T)) \qquad (2.10)$$

– *Шаг улучшение стратегии*.

$$\mu^{i+1} = \arg\min_{u}[H(x, u, \nabla V_x^{\mu^{(i)}})]$$

Стоит обратить внимание, что ни один алгоритм не требует знания динамики системы. То есть они работают для частично неизвестных систем.

2.8 Обучение с подкреплением и адаптивное управление

В настоящее время наблюдаются существенные отличия в терминологии области теории автоматического управления и обучения с подкреплением. Так в одном случае используется терминология, связанная с искусственным интеллектом - максимизация функции, ценность, награда, тогда как в случае ДП стандартным является терминология из области ТАУ - минимизация функции, стоимость, затраты (табл.2.1).

Таблица 2.1 — Термины RL и ТАУ

Обучение с подкреплением	Теория управления
Агент	Алгоритм принятия решения, регулятор
Действие	Управляющее воздействие
Среда	Объект управления (система)
Награда	Противоположна стоимости
Функция ценности	Противоположна функции стоимости (функционал качества)
Максимизация функции ценности	Минимизация функции стоимости

На рис.2.3 показано две структурные схемы, первая из которых своейственна в рамках теории управления, а вторая – науке ИИ. Стоит отметить, что в терминологии ИИ в физический мир (окружение) относят все элементы и сигналы, не относящиеся к алгоритму регулятора – объект упраления, исполнительный механизм, датчики, задающие устройства, шум и т.д.

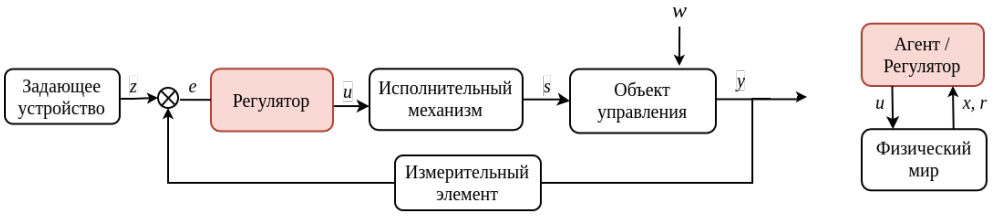


Рисунок 2.3 — Структурная схема системы управления

Связь адаптивного управления и RL

Косвенное адаптивное управление, включает блок идентификации, который принимает на вход управляющее воздействие и действительный выход с объекта управления, расчитывает новые параметры, чтобы обновить параметры управления.Данный подход похож на методы RL на основе модели. (рис.2.4)

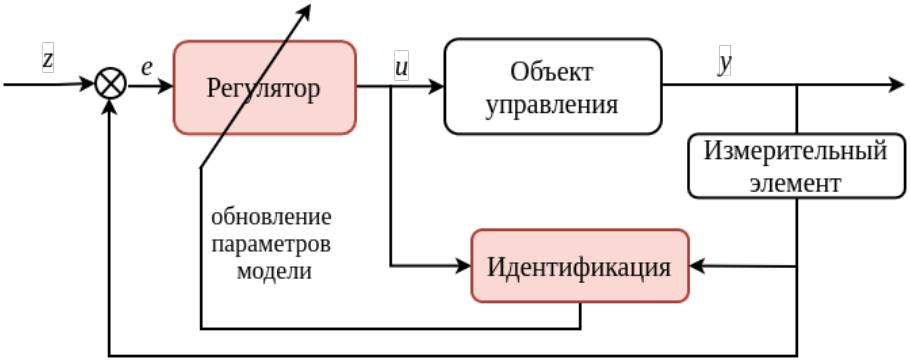


Рисунок 2.4 — Косвенное адаптивное управление или модельный RL

В этом случае идентификация представлена предсказывающим блоком и блоком оценки (рис.2.5). На вход блока оценки поступает предсказанный сигнал и действительный сигнал, в случае, если различие между сигналами превосходит некоторого заданного порога, критик инициирует обновление параметров блока предсказателя. Данный подход аналогичн имитационному обучению, где цель RL формируется не на основе функции награды, а на основе экспертных около-оптимальных данных. В прямом адаптивном управлении явно не

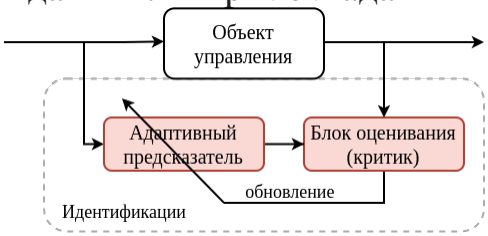


Рисунок 2.5 — Идентификация или имитационное обучение

задается модель системы (рис.2.6), а используется специальный механизм для сравнения желаемого значения сигнала на выходе и действительно, при этом Прямое адаптивное управление соответсвует безмодельному RL. Рассмотрим один из основополагающих алгоритмов в RL – итеративное обучение (рис.2.7).

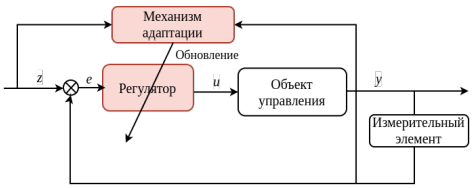


Рисунок 2.6 — Прямое адаптивное управление

Представим этот метод, используя две одинаковые замкнутые системы. Моделирование выполняется на разных итерациях. Первая система запускается на перой итерации, накапливая последовательность рассогласования, генерирует сигнал ошибки, подает его на вход регулятора, который генерирует поправку, на основе минимизации наблюдаемой ошибки. Поправка суммируется с управляющим сигналом регулятора второй система на другом шаге итерации. Такой процесс может выполнятся на каждой итерации с целью подавления помех в замкнутом контуре и минимизации ошибки управления.

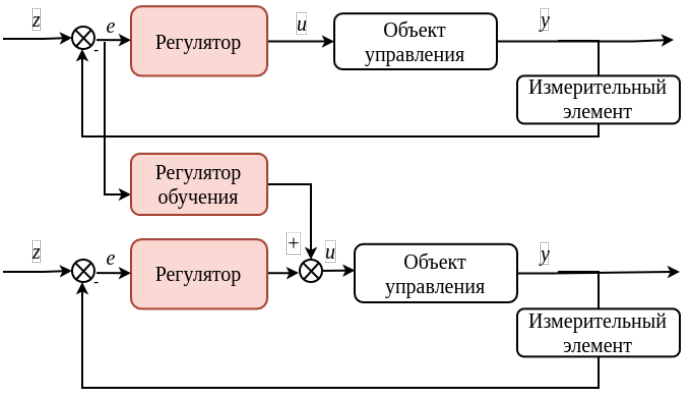


Рисунок 2.7 — Итеративное обучение

2.9 Открытые задачи обучения с подкреплением

Обучение с подкреплением доказало свою ценность в ряде ограниченных задач, но большую часть исследований в области RL часто трудно использовать в реальных системах из-за ряда допущений, которые редко выполняются на практике [24]. Ниже перечислены основные нерешенные задачи в области RL, которые необходимо решить, чтобы активно использовать метолды RL для управления технических систем:

- Настройка регулятора на реальной системе на ограниченных образцах.
- Многомерные непрерывные пространства состояний и действий.
- Ограничения безопасности, которые никогда или, для определенных объектов, редко должны нарушаться.
- Частично наблюдаемые системы, альтернативно рассматриваются как нестационарные или стохастические.

– Сложность формализации функционала качества (функции награды), в частности, для многоагентных систем.
– Интерпретируемость результатов синтеза регуляторов на базе RL.
– Расчет управляющее воздействие на частоте работы системы.
– Большие или неизвестные задержки в исполнительных или измерительных механизмах системы.

В отличие от большинства исследований в области RL, в реальных системах нет отдельной среды настройки и оценки. Все данные для обучения поступают из реальной системы, регулятор должен работать достаточно хорошо и действовать безопасно на протяжении всей настройки (обучения). Для многих систем это означает, что исследование должно быть ограничено, в результате поступающие данные будут иметь низкую дисперсию – небольшая часть пространства состояний может быть исследована. Кроме того, поскольку часто существует только один экземпляр системы, подходы, которые создают экземпляры сотен или тысяч сред для сбора большего количества данных для распределенного обучения, не могут быть применены. В случае, если существуют собранные автономные данные по системе, в большинстве случаев они не содержат необходимое количество и охват, которые необходимы существующим алгоритмам RL. Итерации обучения в реальной системе могут занять много времени, так как есть некоторые системы с большим периодом управления – от одного часа до нескольких месяцев, а ответный сигнал может составлять порядка месяцев (например, управление гидролитосферными процессами, онлайн-реклама, эффективность применения лекарственных препаратов). Даже в случае высокочастотных задач управления алгоритм обучения должен быстро настраиваться на потенциальных ошибках без необходимости повторять их несколько раз, прежде чем исправлять их. Таким образом, для обучения в реальной системе требуется, чтобы алгоритм был эффективным и производительным.

В области обучении с подкреплением выделают отдельное направление связанное с разработкой безопасных алгоритмов RL. Безопасное обучение с подкреплением можно определить как процесс разработки закона управления при максимизации функцианала, при обеспечении соблюдение ограничений безопасности во время процессов настройки и эксплуатации. Как правило, выделяют два подхода к безопасному обучению с подкреплением. Первый основан на модификации критерия оптимальности. Второй основан на модификации процесс обучения путем включения внешних знаний, дополнительных ограни-

чений. Может применяется резервный регулятор, который в случае, если закон управления нарушает ограничения безопасности, взять на себя управление. Это своего рода алгоритмическое резервирование управления (рис.2.8)

Рисунок 2.8 — один из вариантов

Есть ряд успешных работ, которые изучают варианты использования функции Ляпунова для доказательства безопасности управления [25].

Еще одним важным аспектом реальных систем является то, что они принадлежат и управляются людьми, которые требуют понимания алгоритмов управления. По этой причине интерпретируемость закона управления важна для реальных задач. Особенно в случаях, когда функция имеет альтернативный и неожиданный подход к управлению. В случае ошибок алгоритма управления важно иметь возможность апостериорно понять причину ошибки. Есть ряд работ направленных на исследование данной проблемы, например, с применением предметно-ориентированного языка программирования.

Вывод по главе 2. В ходе анализа и изучения связи между обучением с подкреплением и управлением, сделаны следующие выводы:

- Обучение с подкреплением – это параметрический метод аппроксимации функции ценности для решения задачи приближенного динамического программирования;
- Обучение с подкреплением, как и приближенное динамическое программирование способно справиться с проблемой проклятья размерности;
- Динамическое программирование рассматривается как один из главных способов обучения агента;
- Модельное обучение с подкреплением соответствует косвенному адаптивному управлению, безмодельное обучение с подкреплением соответствует прямому адаптивному управлению;
- Главные нерешенные задачи обучения с подкреплением – это описание функции награды и обеспечение безопасности как во врмея настройки регулятора, так и во время эксплуатации;

3 ПРИМЕНЕНИЕ МЕТОДОВ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ В ЗАДАЧЕ РАЗРАБОТКИ РЕГУЛЯТОРОВ ДЛЯ СЛОЖНЫХ СИСТЕМ УПРАВЛЕНИЯ

В данной главе будут рассмотрены практические примеры применения алгоритмов RL для задачи управления обратного маятника и сложных систем, которые описываются моделями типа «хищник-жертва»: развитие опухоли, производство пенициллина, развитие ВИЧ-инфекции.

3.1 Задача управления обратным маятником

Рассмотрим следующую математическую модель перевернутого маятника:

$$\ddot{\vartheta}_t = -0.01\dot{\vartheta}_t + 9.8 \sin \vartheta_t - U_t \cos \vartheta_t$$

где $\vartheta_t \in \mathbb{R}$ – угол наклона маятника; $U_t \in \mathcal{U}$ – крутящий момент приложенный к маятнику в момент времени t. Параметры модели маятника: $m = 1$ кг; $l = 1$ м; $g = 9.8$ м/с²; $\mu = 0.01$; Пространство действий может быть задано как $\mathcal{U} = [-u_{max}, u_{max}] \subset \mathbb{R}$, при ограничении $u_{max} = 5$ Н $\cdot$ м. Динамика системы может быть представлена как:

$$f_d(x) = \begin{bmatrix} x_2 \\ 9.8 \sin x_1 - 0.01 x_2 \end{bmatrix}, F_c(x) = \begin{bmatrix} 0 \\ -\cos x_1 \end{bmatrix}$$

где $x = [x_1, x_2]^T \in \mathbb{R}^2$. При моделировании устанавливаем коэффициент дисконтирования равный $\gamma = 0.1$ и шаг по времени $\Delta t = 10$ мс.

Цель управления – качнуть вверх и, в конечном итоге, установить маятник в вертикальное положение $2\pi k$ для некоторого $k \in \mathbb{Z}$ при ограничении крутящего момента $|U_t| \leqslant u_{max}$.

Нахождение оценки политики V_i на каждой итерации i производится аппроксимируя линейную функцию:

$$V_i(x) \approx V(x; \theta_i) = \theta_i^T \varphi(x),$$

где $\theta_i \in \mathbb{R}^L$ - веса и $\varphi(x)$ признаки с $L = 121$

Поскольку для шага улучшения политики требуется дифференцируемая функция, выбрана радиально-базисные функция в качестве признаков $\varphi(x)$, следовательно, j-я компонента вектора признаков $\varphi(x)$ задается как:

$$\varphi_j(x) = \exp(-(x - c_j)^T \Sigma^{-1} (x - c_j))$$

где Σ^{-1} – весовая матрица, а c_j - центральные точки радиально базисных функций.

Рассматривается фунция вознаграждения r, заданная формулами:

$$r(x,u) = \mathfrak{r}(x) - \mathfrak{c}(u)$$

где $\mathfrak{r}(x)$ и $\mathfrak{c}(u)$ для непрерывного управления, применяя алгоритм DPI, имеют следующий вид:

$$\begin{aligned}
\mathfrak{r}(x) &= \cos x_1 \\
\mathfrak{c}(u) &= \lim_{v->u} \int_0^v (s^T)^{-1}(\mathfrak{u}) \cdot \Gamma du \\
s(\mathfrak{u}) &= u_{\max} \tanh(\mathfrak{u}/u_{\max})
\end{aligned}$$

При этом функция s задает ограничения модели. Г - положительно определенная матрица. Для решения задачи управления, был использован следующий метод обучения с подкреплением: *Дифференциальная оценка по стратегиям* (англ. Differential Policy Iteration, DPI) — алгоритм оценивает и улучшает стратегию, начиная от начальной допустимой стратегии π_0 и до тех пор, пока оценка и стратегия не сойдутся. На этапе оценки стратегии производится расчет дифференциального уравнения Беллмана, чтобы получить функцию стоимости $v_i = v_{\pi_{i-1}}$ для последней стратегии π_{i-1}:

$$\alpha \cdot v_i(x) = h(x, \pi_{i-1}(x), \nabla v_i(x)) \forall x \in X \tag{3.1}$$

Далее рассчитанное значение v_i используется в улучшении стратегии, то есть для получения следующей стратегии π_i путем минимизации гамильтониана:

$$\pi_i(x) \in \arg\min_{u \in U} h(x,u, \nabla v_i(x)) \forall x \in X \tag{3.2}$$

Стоит отметить, что при использовании алгоритма дифференциальной оценки по стратегиям необходимо иметь формализованную модель ОУ (на первом шаге алгоритма рассчитывается гамильтониан). В работе [26] представлен схожий алгоритм, который позволяет отказаться от использования модели объекта, который в рамках данной работы рассмотрен не был. Но в дальнейшем представляет особый интерес, так как данные методы направлены на работу в непрерывном времени и пространстве. Результат работы регулятора для перевернутого маятника при ограничениях представлен на рис.3.1.

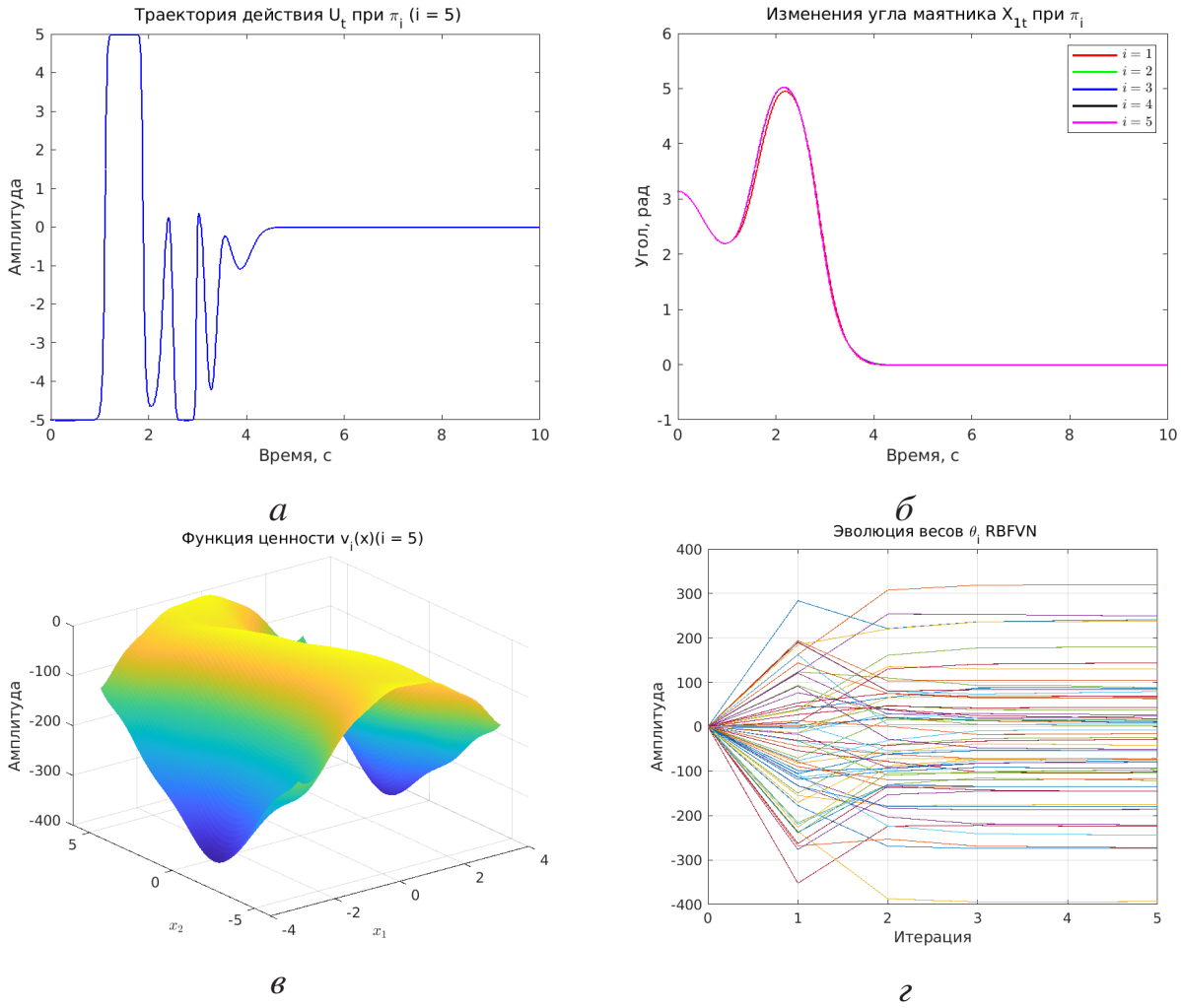


Рисунок 3.1 — Применение алгоритма DPI для задачи управления обратным маятником; *а* – управляющее воздействие; *б* – изменение угла наклона маятника; *в* – изменение функционала качества; *г* – график изменения весов

3.2 Управление производством пенициллина

Модель биореактора описывает поведение двух видов, которые соревнуются за один субстрат. Реализация систем автоматизации для биореакторов ограничена сложностью синтеза регулятора для систем с большим количеством переменных. Так же автоматизация таких систем ограничена сложностью описания внешних воздействий (например учет воздействия кислорода и углекислого газа на систему), поэтому возникает интерес реализации оптимально-адаптивных регуляторов для таких систем.

В качестве примера рассмотрена модель производства пенициллина периодического действия [27]. Модель была основана на следующих предположениях: 1) рост клеток ограничен субстратом (глюкозой) и кислородом, 2) вся биомасса способна расти и синтезировать пенициллин, 3) образование продукта подавляется субстратом, и 4) требования к техническому обслуживанию постоянны

[28]. Модель системы описывается системой однородных дифференциальных уравнений:

$$\begin{aligned}
\dot{X} &= \mu(S)X - \frac{u_{\text{inp}}}{V}X, \\
\dot{S} &= -\frac{\mu(S)X}{Y_{\text{x}}} - \frac{vX}{Y_{\text{p}}} + \frac{u_{\text{inp}}}{V}(S_{\text{in}} - S), \\
\dot{P} &= vX - \frac{u_{\text{inp}}}{V}P, \\
\dot{V} &= u_{\text{inp}},
\end{aligned} \tag{3.3}$$

где удельная скорость роста продукта:

$$\mu(S) = \frac{\mu_m S}{K_{\text{m}} + S + (S^2/K_{\text{i}})},$$

где X – концентрация биомассы (г/л), S – концентрация субстрата (г/л), P – концентрация продукта (г/л), V – объем биомассы в биореакторе. Управляющим входом является u – скорость подачи субстрата (г/(л ч)). Параметры модели представлены в табл.3.1

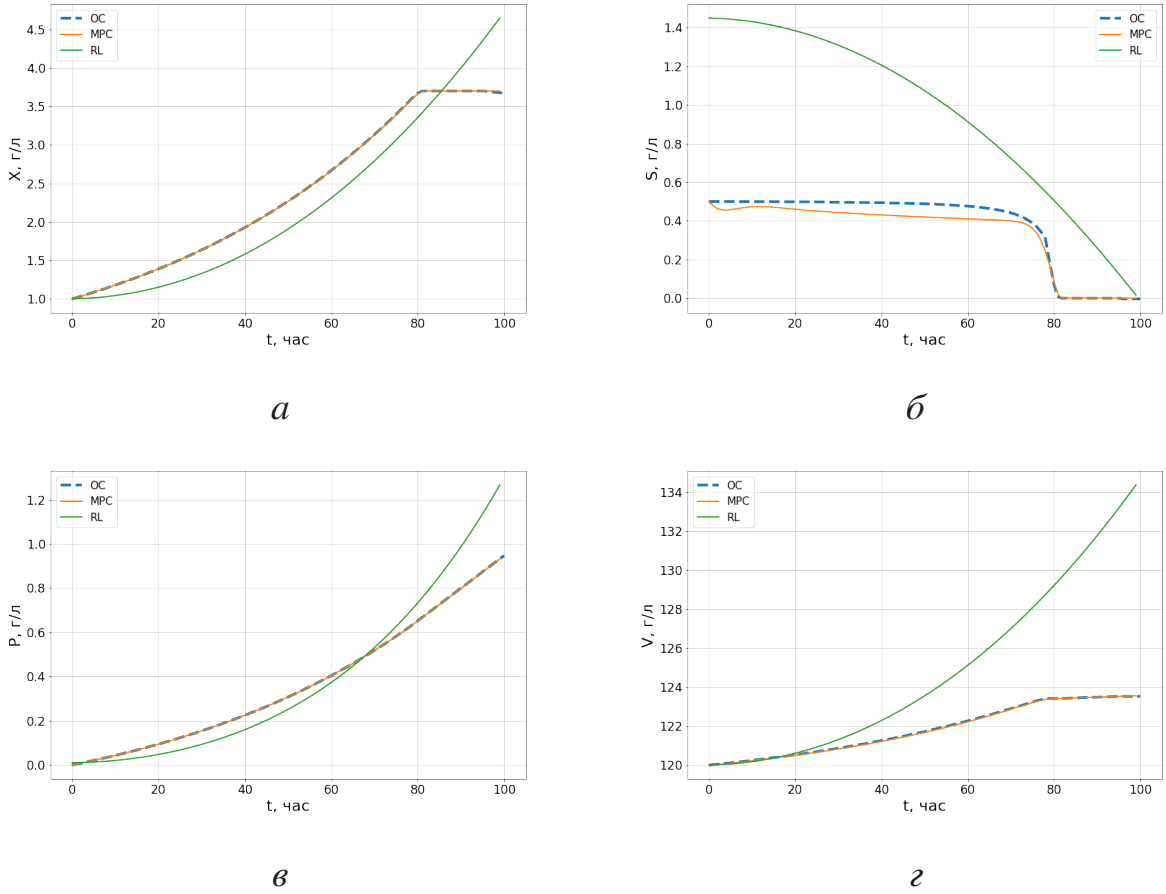
Таблица 3.1 — Параметры модели производства пенициллина

Параметр	Описание параметра	Величина	Единица измерения
S_{in}	концентрация субстрата в корме	0.2	л / (г/л)
Y_x	доходность биомассы на единицу массы субстрата	0.2	л / (г / ммоль O2)
Y_p	доходность продукта на единицу массы субстрата	1.2	л /((г / ммоль O2)
μ_m	максимальная удельная скорость продукта d	0.02	1/ч
v	удельная скорость роста биомассы	0.004	1/ч
K_m	константа Моно	0.05	г/л
K_i	константа ингибирования	5.0	г/л

Цель управления – получение максимальной концентрации продукта P. Задача оптимального управления может быть сформулирована следующим образом:

$$\begin{aligned}
&J = \min_{u,t_f}\Big(\int_0^{t_f} P(\tau)d\tau\Big) \\
&\mathbf{x} = [X\ S\ P\ V], \mathbf{x}(0) = [1.0\ 0.5\ 0.01\ 120.0]^T \\
&\dot{\mathbf{x}} = f(\mathbf{x},u) \\
&0 \leqslant u \leqslant 0.2,\ \ 0 \leqslant X \leqslant 3.7,\ \ 0 \leqslant P \leqslant 3.0 \\
&0 \leqslant V \leqslant 125,\ \ S \geqslant 0
\end{aligned} \tag{3.4}$$

Для данного объекта управления было реализовано три типа регулятора: оптимальный регулятор, полученный численными методами, регулятор онлайн-оптимизации MPC и регулятор на основе обучения с подкреплением – Глубокие детерминированные градиенты политики (англ. Deep Deterministic Policy Gradient, DDPG). Используя пакетный модуль OpenOCL, реализованный на языке MATLAB найдено численное решение оптимального управления для модели производства пенициллина. При разработке системы управления методом RL функция награды имела следующий вид: $r = lg(P_k/P_{k-1}) - 50 * [X > 3.7] - 50 * [P > 3.0] - 50 * [V > 125]$. Результаты моделирования представлены на рис.3.2



а *б*

в *г*

Рисунок 3.2 — Моделирование управления производства пенициллина: *а* – изменения концентрации биомассы, *б* – изменения концентрации субстрата, *в* – изменения концентрации продукта, *г* – изменения заполненного объема в биореакторе

Для данного процесса важным критерием качества является конечное значение концентрации пенициллина в реакторе P. Различие траекторий управляющих воздействий рис.3.3 обосновывается не удовлетворительным выбором функции награды, которое предполагается составлять вместе с экспертом.

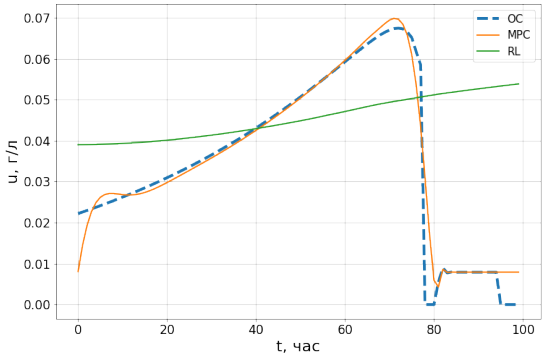


Рисунок 3.3 — Управляющее воздействие, а именно скорость подачи субстрата в реактор

3.3 Управление ростом опухоли

Рак – это общее название группы заболеваний, которые включают повторяющееся и неконтролируемое деление и распространение аномальных клеток. Эти аномальные ткани называются опухолями. Ранняя диагностика и эффективное лечение повышают выживаемость больных. Оптимальный график лечения и доза лекарства варьируются в зависимости от стадии опухоли, веса пациента, уровней лейкоцитов (иммунных клеток), сопутствующего заболевания и возраста пациента. Таким образом, правильное планирование и персонализация химиотерапевтического лечения жизненно важны для снижения уровня смертности. Чтобы вычислить оптимальную политику управления и вознаграждение, требуется математическая модель, которая показывает воздействие химиотерапевтического препарата в динамике. Реалистичная модель должна учитывать рост опухоли, реакцию иммунной системы человека на рост опухоли и влияние химиотерапевтического лечения на иммунные клетки, нормальные клетки и рост опухоли.

Одна из основных проблем, связанных с изучением рака как динамической системы, состоит в том, что как и любое другое заболевание, известен своей сложной, нелинейной и неопределенной механикой действия. Следовательно, математические модели в дифференциальных уравнениях не в состоянии учесть все вариации в динамике пациента, поэтому поставлена задача разработки регулятора на основе обучения с подкреплением.

В работе [29] представлена фармакологическая модель химиотерапии рака, заданная нелинейной системой из 4-х детерминированных однородных

дифференциальных уравнений:

$$\begin{aligned}
\dot{T} &= r_1 T(1 - b_1 T) - c_2 IT - c_3 TN - a_2(1 - e^{-M})T, \\
\dot{N} &= r_2 N(1 - b_2 N) - c_4 TN - a_3(1 - e^{-M})N, \\
\dot{I} &= s + \frac{\rho IT}{\alpha + T} - c_1 IT - d_1 I - a_1(1 - e^{-M})I, \\
\dot{M} &= u(t) - d_2 M
\end{aligned} \tag{3.5}$$

где $T(t)$ – концентрация раковых клеток, $N(t)$ – концентрация нормальных клеток $I(t)$ – концентрация иммунных клеток в крови (лейкоцитов), и $M(t)$ – концентрация химиотерапевтического препарата в крови. Управляющее воздействие $u(t)$ соответствует скорости ввода препарата (мг/(л день)). Параметры модели представлены в табл.3.2.

При разработке схемы лечения важно оптимизировать количество используемого лекарства, чтобы регулировать побочные эффекты химиотерапии, поскольку часто иммунная система пациента ослабевает и становится склонной к опасным для жизни инфекциям, что снижает ее способность искоренить рак. В литературе рассматривается два случая: основной и подготовительный – несколько нереалистичный случай, в котором цель состоит в том, чтобы искоренить рак независимо от состояния популяции остальных клеток. В обоих случаях начальное условие было одинаковым: [0,7 1 1 0] Т.

Задача оптимального управления для упрощенного случая, может быть сформулирована следующим образом:

$$\begin{aligned}
J &= \min_{u,t_f}\left(\int_0^{t_f} T(\tau)d\tau\right) \\
\mathbf{x} &= [T\ N\ I\ M], \quad \mathbf{x}(0) = [0.7\ 1\ 1\ 0]^T \\
\dot{\mathbf{x}} &= f(\mathbf{x},u) \\
0 &\leqslant u \leqslant 10
\end{aligned}$$

Тогда как такую задачу в терминах RL, опираясь на гипотезу о награде сформировать в виде функции вознаграждения: $R = -dt \cdot T$. В этом случае dt включено в вознаграждение только для того, чтобы упростить сравнение с функционалом качества, приведенного в постановке задачи оптимального управления. Решением такого случае станет подача максимального значения управляющего воздействия на вход системы.

Рассмотрим реальный случай. Чтобы гарантировать безопасность пациента во время лечения, к постановке (3.3) добавлены дополнительные ограничения

Таблица 3.2 — Параметры модели опухоли

Параметр	Описание параметра	Величина	Единица измерения
a_1	скорость фракционного уничтожения иммунных клеток	0.2	л / (мг день)
a_2	скорость фракционного уничтожения опухолевых клеток	0.3	л / (мг день)
a_3	скорость фракционного уничтожения нормальных клеток	0.1	л / (мг день)
b_1	взаимная несущая способность опухолевых клеток	1	1 / клетка
b_2	взаимная несущая способность нормальных клеток	1	1 / клетка
c_1	срок конкуренции иммунных клеток (конкуренция между опухолевыми и иммунными клетками)	1	1 / (клетка день)
c_2	срок конкуренции опухолевых клеток (конкуренция между опухолевыми и иммунными)	0.5	1 / (клетка день)
c_3	срок конкуренции опухолевых клеток (конкуренция между нормальными и опухолевыми)	1	1 / (клетка день)
c_4	срок конкуренции нормальных клеток конкуренция между нормальными и опухолевыми)	1	1 / (клетка день)
d_1	Темп гибели иммунных клеток	0.2	1 / день
d_2	Скорость распада вводимого медицинского препарата	1	1 / день
r_1	Скорости роста опухолевых клеток (на единицу)	1.5	1 / день
r_2	Скорости роста нормальных клеток (на единицу)	1	1 / день
s	Скорость притока иммунных клеток	0.33	клетка / день
α	Скорость иммунного порога (порогового входа)	0.3	клетка
ρ	Скорость иммунного ответа	0.01	1 / день

состояния. В частности $N(t) \geqslant 0{,}4$ и $I(t) \geqslant 0{,}4$ Для задачи RL награда будет равна:

$$R = dt \cdot (-T - 0.5 \cdot [N < 0.4] - 0.5 \cdot [I < 0.4])$$

На рисунке приведены графики моделирования при оптимальном управления, регулирования на основе MPC и регулятора на основе алгоритма обучения с подкреплением расширения глубоких детерминированных градиентов поли-

тики (англ. Twin Delayed Deep Deterministic Policy Gradients, TD3). Результаты моделирования при регулировании представленны на рис.3.4 и рис.3.5.

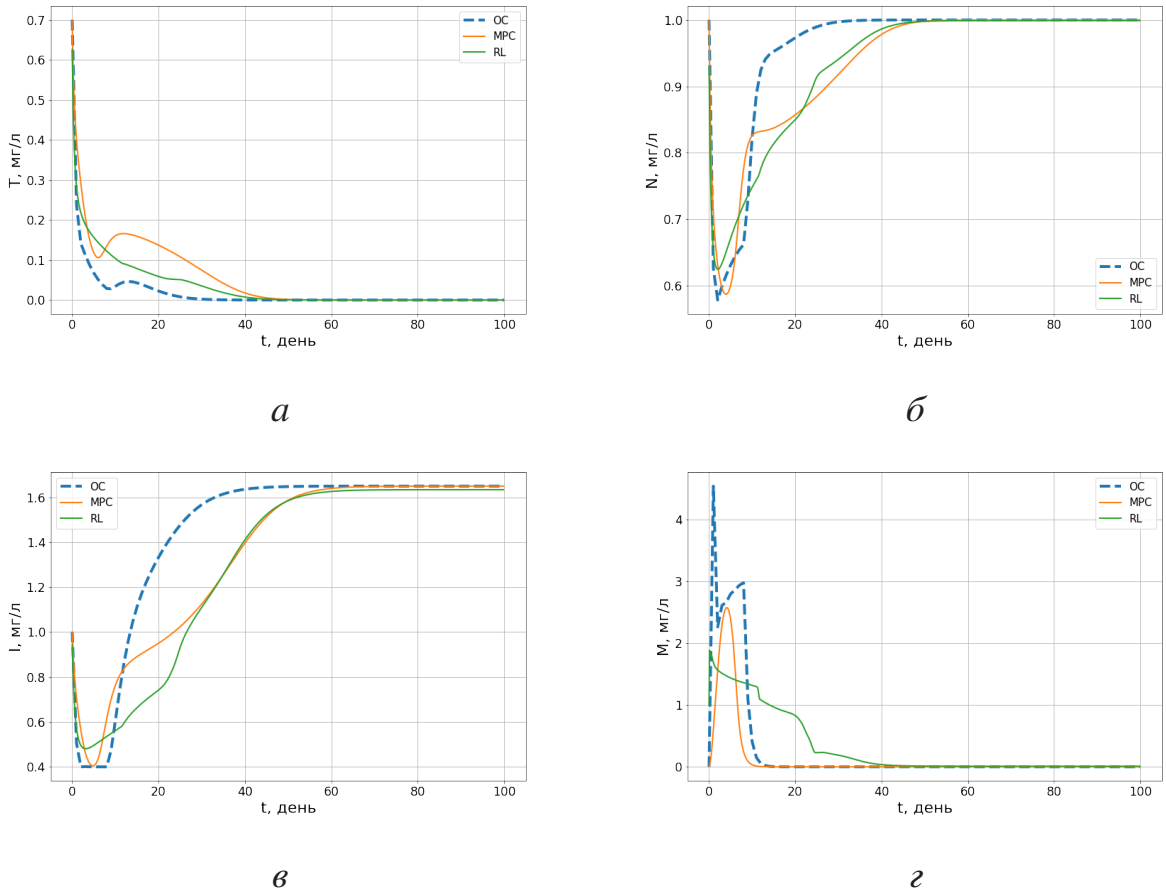


Рисунок 3.4 — Моделирование управления ростом опухоли: *а* – изменение концентрации раковых клеток, *б* – изменение концентрации нормальных клеток, *в* – изменение концентрации иммунных клеток в крови (лейкоцитов), *в* изменение концентрации химиотерапевтического препарата в крови

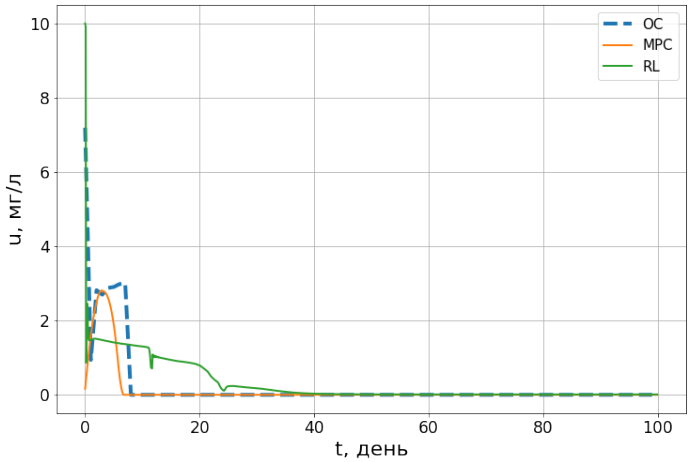


Рисунок 3.5 — Управляющее воздействие, представленное скоростью ввода препарата

Вывод по главе 3. В ходе проведения модельных экспериментов и разработки регуляторов для сложных объектов было установлено:

- Эксперимент по разработке регулятора для управления производством пенициллина показал, что качество работы безмодельных алгоритмов обучения с подкреплением сильно зависит от дизайна функционала качества, что только подтверждает указанную проблему в главе 2;
- Для задачи управления перевернутым маятников регулятор на основе модельного алгоритма обучения с подкреплением учел ограничения, которые были указаны при разработке функционала качества;
- Регулятор на основе обучения с подкрепления для задачи управления роста опухоли показал такие же высокие качества управления как оптимальный регулятор и регулятор на основе онлайн оптимизации MPC;
- Проведенные эксперименты подтверждают применимость методов обучения с подкреплением для задач управления сложными объектами;

4 РАЗРАБОТКА БИЗНЕС-ПЛАНА ПО КОММЕРЦИАЛИЗАЦИИ ПРОЕКТА

4.1 Описание проекта

Резюме. Бизнес-план посвящен оценки рентабельности оказания услуги по разработке алгоритмов управления сложных технических систем с применением методов обучения с подкреплением. Потенциальные заказчики услуги – промышленные предприятия и компании в области разработки и проектирования автоматизированных систем управления технологическим процессом (АСУ ТП). Для существующих подходов в управлении отдельными техническими системами на рынке АСУ ТП характерна высокая стоимость и длительный цикл разработки. Поэтому предлагается использовать методы обучения с подкреплением, с целью уменьшить время разработки, без потери качеств управления.

В первый квартал планируется реализация и внедрение двух алгоритмов автоматизации, стоимость каждого – 700 000 рублей. В последующие кварталы планируется объем – 3 единиц услуг, по 500 000 рублей каждая. В связи со спецификой технологии оказываемых услуг, необходимо проводить активную кампанию по продвижению услуги на рынке, что подразумевает затраты на рекламу в размере 90 600 рублей без НДС в первый год оказания услуг. Бизнес-план составлен на прогнозный период 1 год. Срок организации бизнеса составляет 1 месяц, предполагается, что со 2-го месяца проект начнет приносить доход. В качестве расчетов принят календарный год с мая по апрель. Для расчета инвестиционной привлекательности проекта использовались следующие допущения:

- организация использует общую систему налогообложения;
- налог на добавленную стоимость (НДС) – 20%;
- отчисления работодателя с доходов сотрудника – 30%;
- налог на прибыль – 20%.

Полученные показатели инвестиционной привлекательности проекта с учетом ставки дисконтирования (20 %):

- сумма инвестиций (I) составляет 500 000 рублей;
- чистая текущая стоимость проекта (NPV) составляет 1 830 000 рублей;
- срок окупаемости проекта составит 1 квартал.

Полученные показатели указывает на экономическую целесообразность осуществления проекта с учетом ограничений и допущений в бизнес-плане.

Описание продукции.Описание услуги: Разработка алгоритмов интеллектуальных систем управления сложных технологических объектов с применением методов обучения с подкреплением.

Цифровизация промышленности и, как следствие, автоматизация производственных процессов - современный тренд, движение в сторону которого, обеспечивает предприятиям высокую гибкость в формировании бизнес-моделей и широкий охват потенциальной клиентской базы посредством интеграции киберфизических систем и интернета вещей в производственный процесс. В основе внедрения новых технологий лежит стремление к комплексному повышению эффективности и созданию условий для успешной работы предприятия. Промышленные компании сталкиваются с задачами снижения издержек, сокращения сроков вывода новой продукции на рынок, улучшения эффективности всех процессов, поскольку требования со стороны потребителей данной услуги каждый год растут. Поэтому, со стороны производства возникает запрос на автоматизацию производственных процессов, с применением современных технологий, которые смогут обеспечить адаптивность, отказоустойчивость и оптимальность процессов. Компания Forrester указывает в своем отчете, что 20% компаний создадут инновационные цифровые подразделения в ближайшие годы. То есть ряд промышленных предприятий будет пробовать реализовывать и внедрять новые технологии автоматизации самостоятельно. Существует ряд компаний, которые уже давно на рынке и готовы внедрять надежные системы автоматизации, с помощью традиционных подходов. Компании, которые в настоящее время поставляют системы автоматического управления, при синтезе регуляторов, по большей части, используют классические подходы теории автоматического управления. Тогда как наука шагнула вперед и теперь существуют лучшие подходы в разработке алгоритмов управления технических систем.

Обращаясь к традиционным подходам в управлении сложных технических систем при создании систем автоматического управления необходимо иметь точную математическую модель объекта управления (ОУ). Во многих реальных задачах построение такой модели либо невозможно, либо требует проведения трудоёмких исследований. При этом параметры ОУ могут изменяться в широких пределах в процессе функционирования системы, либо иметь большой разброс значений от образца к образцу. В таких случаях регуляторы с постоянными настройками не всегда могут обеспечить требуемое качество работы системы.

Если обратиться к области машинного обучения, то существует ряд методов, которые могут помочь в задаче управления технических систем. Обучение с подкреплением – группа методов, при которых алгоритм учится выполнять задачу, многократно взаимодействуя с симулятором динамической системой. Это происходит без прямого вмешательства человека и без необходимости программировать алгоритм для выполнения конкретной задачи.

В рамках данного проекта предлагается услуга по разработке алгоритмов управления на основе методов обучения с подкреплением и внедрение их в микроконтроллеры.

Технические характеристики предоставляемой услуги будет:

1. Алгоритм регулирования на высокоуровневом языке программирования;
2. Подключения к оборудованию по промышленным протоколам передачи данных;
3. Визуализация работы алгоритма, создание автоматизированных отчетов;
4. Алгоритм конвертации кода на промышленный язык программирования.

Самая большая проблема рынка автоматизации производства – это первоначальная стоимость инвестиций, необходимых для проектирования, выполнения и установки автоматизированной системы, а так же малое колличество подготовленных специалистов на стыке разных дисциплин. Предполагается, что предоставляя услугу по разработке алгоритмов управления с использованием обучения с подкреплением, можно в разы ускорить и удешевить процесс разработки регуляторов, увечив качество управления за счет адаптации к системе. Так же сложности возникают, если на предприятии уже имеется оборудование автоматизации отдельного производителя, предлагаемая услуга не привязана к модели контроллеров, датчиков или исполнительных механизмов. То есть является универсальным решением для АСУ ТП.

Анализ рынка сбыта. По данным аналитиков компании Fortune Business Insights, к 2026 году рынок промышленной автоматизации вырастет до суммы в $310 миллиардов, при совокупном среднегодовом темп роста около 8,5%. Оценивая основные экономические характеристики отрасли, отметим, что объем российского рынка АСУ ТП в 2020 году составил 58,7 млрд рублей, подсчитали аналитики компании J’son & Partners Consulting [30]. Рынок находится на рас-

тущем этапе развития, следовательно и сегменту по разработке программного обеспечения присущ рост. На рынке заметна нехватка мощностей, что вызвано стимуляцией цифровизации промышленности со стороны государства. Ожидается, что усиленное внимание к повышению производительности и стремление к устранению опасных ручных процессов с привлечением человека – станет основными движущими силами в ближайшие несколько лет. Отдельно стоит отметить, что пандемия коронавируса уже стала одним из факторов роста рынка автоматизации.

На рынке АСУ ТП можно выделить минимум 2 сегмента по цене. Так как сложно найти информацию по средней цене, затрачиваемой предприятиями на автоматизацию, стоит отметить, что рассматриваемая услуга будет направлена на предприятия среднего и ниже среднего ценового сегмента. Например, такие компании как ПАО «СИБУР Холдинг», ПАО «Газпром» могут позволить себе полноценный комплекс автоматизации от ключевых игроков рынка (Siemens, Schneider Electric и т.д.), которые помимо классических решений регулирования, так же предлагают современные адаптивно-оптимальные подходы. Но в данном случае существует привязка к оборудованию. Данных проект направлен на заказчиков, которые имеют оборудования автоматизации от разных производителей и при этом доход таких компаний относится к средним показателям по рынку автоматизации.

Потенциальные заказчики услуги:

- Промышленные предприятия среднего ценового сегмента;
- Компании в области разработки и проектирования АСУ ТП среднего ценового сегмента.

Высокие затраты на установку и техническое обслуживание, а также отсутствие квалифицированных специалистов являются одними из ограничений в этой отрасли. Услуга охватывает все сегменты предприятий по отраслям: нефтегазовая, металлургическая и горнодобывающая, фармацевтическая, целлюлозно–бумажная, автомобильная, химическая и т.д. Для любых производственных компаний, занимающихся данной деятельностью, важнейшим фактором выживаемости в конкурентной борьбе является постоянное обновление проектов в соответствии с современными стандартами и технологиями.

Лидеры рынка АСУ ТП неизменны на протяжении многих лет, их решения проверены временем и используются на многих предприятиях. Стабильность позиций во многом связана с тем, что производители предлагают законченную

инфраструктуру, и с точки зрения эксплуатации клиентам выгодно иметь оборудование одного вендора, так как это облегчает поиск и замену оборудования и комплектующих, а также подготовку обслуживающего персонала. Ключевыми игроками на рынке автоматизации являются ABB, General Electric, Honeywell Solutions, Emerson Electric, Siemens AG, Schneider Electric, Mitsubishi Electric Corporation, Rockwell Automation и Yokogawa Electric.

Предложений о разработки алгоритмов с помощью методов обучения с подкреплением в свободных источниках лидирующих компаний в автоматизации нет. Но, предполагается, что лидеры области предлагают такие услуги для предприятий высокого ценового сегмента, и стоимость данной услуги – велика. Помимо этого ключевые игроки готовы работать с оборудованием автоматизации только собственной поставки, что не всегда удобно для компаний низкого и среднего ценового сегмента.

По охватываемым областям промышленности рынок систем автоматизации подразделяется на следующие сегменты: аэрокосмический и оборонный, автомобильный, химический, энергетический, продуктовый, здравоохранение, металлургия, нефтегазовый.

В промышленном производстве обучение с подкреплением предлагается использовать в процессах, где требуются сложные навыки принятия решений и регулирования, особенно, когда необходимо справляться с изменениями в динамической среде. Например, в процессе эксплуатации параметры оборудования могут изменятся. Другой пример применения алгоритмов обучения с подкреплением – это разработка алгоритмов управления, используя цифровые двойники. Исследователями из лаборатории промышленного искусственного интеллекта Hitachi America разработали виртуальный цех как двумерную матрицу и использовали алгоритмы обучения с подкреплением для многократного взаимодействия с этой виртуальной средой. По результатам моделирования, исследователи смогли определить лучшую настройку для повышения производительности работы цеха и сокращения задержек поставок.

Анализ конкурентов. Прямых аналогов в выделенном сегменте рынка нет. Потенциальными конкурентами выступают научные лабораторий автоматизации при университетах. Например, лаборатория систем автоматизированного проектирования (САПР) в Санкт-Петербургском Политехническом Университете Петра Великого. Основным преимуществом которой, является наличие учебных стендов, которые могут понадобиться на этапе тестирования алгоритмов. Но

данная лаборатория не специализируется в современных методах машинного обучения, поэтому предполагается, что угроза появления конкурента низкая. Проведен конкурентный анализ по модели Партера. Результаты представлены в таблице 4.1.

Таблица 4.1 — Конкурентный анализ по модели Партера

Критерий	Вывод
Оценка конкурентоспособность товара компании и уровня конкуренции на рынке	средний уровень угрозы со стороны товаров-заменителей
Оценка уровня внутриотраслевой конкуренции	Низкий уровень внутриотраслевой конкуренции
Оценка угрозы входа новых игроков	Высокий уровень угрозы входа новых игроков
Оценка угрозы ухода потребителей (рыночная власть покупателя)	Низкий уровень угрозы ухода клиентов
Оценка угрозы для Вашего бизнеса со стороны поставщиков	Низкий уровень влияния поставщиков

Будут применены следующие стратегии повышении конкурентноспособности: дифференциация продукта – уникальность продукта и его высокое качество и особый подход, своевременное реагирование на потребности рынка – опережение конкурентов во времени за счет универсальности системы реализации алгоритмов управления.

4.2 План маркетинга

План маркетинга включает в себя план продаж, товарную политику, ценовую политику и сбытовую политику и рекламные мероприятия.

План продаж. С учетом роста рынка, на основе метода экспертных оценок сформирован прогнозный план продаж табл. 4.2. Ожидаемый объем продаж и цена услуги установлена исходя из высокого спроса и оценки эксперта в области автоматизации.

Товарная политика. Предлагаемый комплекс продуктов: программный код на высокоуровневом языке, техническая документация, алгоритм конвертации модели на язык программирования промышленных контроллеров (стандарта МЭК (IEC 61131-3)), а так же система визуализации качества регулирования. Так же удовлетворены условия качества продукта – продукт полностью соответствует

Таблица 4.2 — План продаж

Показатели	Квартал				Всего
	I	II	III	IV	
Разработка алгоритма управления					
Ожидаемы объем продаж, ед.	2	3	3	3	10
Цена с НДС, т.р.	700	500	500	500	-
Выручка с НДС, т.р.	1 400	1 500	1 500	1 500	5 900
Нетто-выручка (без НДС), т.р.	1 120	1 200	1 200	1 200	4 720
Сумма НДС, т.р	280	300	300	300	1 180

современным стандартам программного обеспечения, имеет сертифицированный уровень защиты информации. Дизайн и товарный знак продукта будут уточнены в процессе разработки. Предполагаемое техническое обслуживание включено в стоимость – обращение в техническую поддержку за консультациями по работе системы, помощь с первоначальной установкой на программируемый логический контроллер (ПЛК), демонстрация режимов. Гарантийное обслуживание: компания не несет ответственности за проблемы в работе системы, вызванные аппаратными сбоями, но помощь по восстановлению работы системы после аппаратных сбоев включена в стоимость.

Ценовая политика. Метод ценообразования: постоянная базовая составляющая. Цена продукции получена с учетом экспертной оценки: 500 000 р + 100 000 р, в случае дополнительных ограничений. В первый квартал добавленная надбавка по рекомендации специалиста, с целью обеспечить окупаемость продукта. Скидки не предусмотрены. Условия платежа: единоразовый платеж либо рассрочка на 2 месяца. Формы оплаты: банковский перевод, онлайн-платеж. Сроки и условия предоставления кредита: оплата в кредит не предусмотрена.

Сбытовая политика и рекламные мероприятия. С учетом специфики компании возможен только один канал сбыта – прямые продажи. Исходя из данной сбытовой политики, спланированы расходы на рекламную кампанию. Одна из главных целей рекламных мероприятий - это завоевание доли рынка и повышение значимости оказываемых услуг.

С целью привлечения новых заказчиков и повышения имиджа компании рекламная деятельность будет проводиться в специализированных печатных изданиях (не реже 2 раз в год), в сети Интернет (путем разработки и продвижения собственного сайта), налаживание коммуникативных связей на специализи-

рованных конференциях и выставках. В будущем предполагается добавить рекламную деятельность в форме участия в разнообразных массовых мероприятиях в качестве спонсора. Данные по расходам на рекламные мероприятия приведены в табл. 4.3.

Таблица 4.3 — Расходы на маркетинговые и рекламные мероприятия

Статья	Сумма, рубл.
Реклама в журнале «Control Engineering Россия»	45 000
Реклама в журнале «Автоматизация в промышленности»	20 000
Создание и продвижение сайта компании	20 000
Печать буклетов и брошюр	5 600
Всего	90 600

Предполагаемая величина суммарных расходов на рекламу в первый год составит 90 600 рублей, в последующие годы прогнозируется увеличение расходов на рекламную компанию на 8% в год с целью стимулирования спроса на услуги. Так же в последующие годы планируется проведение исследования на предмет целесообразности открытия филиалов в других населенных пунктах, проведение маркетинговых исследований с целью получения анализа о востребованности услуги, мониторинг клиентской базы и усиленный комплекс коммуникативных мероприятий.

4.3 План производства

Ввиду того, что компания занимается оказанием услуг по разработке и внедрению программного обеспечения, то принципиальной необходимости в удобном местоположении офисов и лабораторий отсутствует. Проект предусматривает аренду одного офисного помещения с мебелью. Средняя цена аренды офисного помещения с мебелью в 40 м.^{2} в Санкт-Петербурге стоит около 30 000 рублей в месяц. 360 000 рублей в год (288 000 + 72 000 руб. НДС).

В таблице 4.4 представлены расходы на материалы для осуществления проекта по предоставлению данной услуги за первый год.

На коммунальные услуги (электроэнергия, отопление, водопровод, мобильная связь и пр.) планируется потратить 250 000 руб. за первый год реализации услуг (200 000 + 50 000 руб. НДС). В дни, когда необходимо осуществлять выезд на объект к заказчику возникает необходимость посуточной аренды легкового транспортного средства. На аренду автомобиля планируется потратить 100 000 руб. за первый год реализации услуг (80 000 + 20 000 руб. НДС).

Таблица 4.4 — Потребность в расходных материалах за первый год

№	Наименование	Кол-во	Сумма с НДС, руб.	Сумма без НДС, руб	Сумма НДС, руб.
1	Бумага офисная, 500 л., А4	4	1 000	800	200
2	Ручка шариковая, синяя, упаковка 8 шт.	3	560	448	112
3	Папка - регистратор, А4	4	760	608	152
Итого в год			2 320	1 856	464
Итого в месяц			193	155	39

В таблице 4.5 приведены планируемые затраты на приобретения оборудования. При этом доставка оборудования предоставляется магазинам бесплатно.

Таблица 4.5 — Затраты на приобретение оборудования

№	Наименование	Кол-во	Цена руб./шт.	Итого, руб.
1	Монитор Asus VS247NR, 1920x1080, LED, черный	2	9 490	18 980
2	MicroXperts [C300-05] W7PRO персональный компьютер, Intel Core i5-4460, RAM 8Gb, HDD 1Tb, DVD±RW	2	44 560	89 120
3	SVEN Standard 310 Combo, клавиатура USB + оптическая мышь USB, черный	2	990	1 980
4	ИБП CyberPower UT650E	1	3 000	3 000
5	Лазерное МФУ Xerox WorkCentre, А4, Сетевое, USB 2.0, принтер/копир/сканер	1	12 000	12 000
6	Коммутатор Cisco SB SG100D-08-EU	1	3 900	3 900
Всего				128 980

Согласно учетной политики организации, амортизация рассчитывается линейным способом, по основному оборудованию, по формуле:

$$A = S \cdot K,$$

где – размер месячных амортизационных отчислений; S – первичная стоимость имущества; K – норма амортизации. Норма амортизации рассчитывается по формуле:

$$K = \frac{1}{T} \cdot 100\%,$$

где T – срок полезного использования, указанный производителем оборудования. В таблице 4.6 приведены значения годовых амортизационных отчислений.

Таблица 4.6 — Амортизационные отчисления (годовые)

№	Наименование	Срок полезного использования	Норма амортизации	Сумма амортизационных отчислений, руб.
1	Монитор Asus VS247NR	10	10	1 898
2	Персональный компьютер MicroXperts W7PRO Intel Core i5-4460	10	10	8 912
3	Клавиатура USB + оптическая мышь USB	5	20	396
4	ИБП CyberPower UT650E	10	10	300
5	Лазерное МФУ Xerox WorkCentre	10	10	1 200
6	Коммутатор Cisco SB SG100D-08-EU	10	10	390
Всего				13 096

Годовая амортизация с основных средств составит 13 100 рублей, тогда ежемесячная амортизация составит 1090 рублей.

Инвестиционные затраты. Произведена оценка общих инвестиционных затрат, равная суммарной потребности в инвестициях на создание предприятия (инвестиционные затраты на основные средства и предпроизводственные расходы) и потребности в инвестициях для текущей деятельности (оборотные активы, необходимые для формирования начальных товарно-материальных запасов и др.). Результаты представлены в табл.4.7.

Таблица 4.7 — Перечень необходимого оборудования

№	Наименование	Способ получения	Стоимость без НДС, руб.	Стоимость вкл. НДС, руб.	Сумма НДС, руб.	Период получения
1	Основное оборудование	Покупка	103 184	128 980	25 796	май 2021
2	Транспортное средство	Аренда	80 000	100 000	20 000	май 2021
Итого в год			183 184	228 980	36 637	
Итого в месяц			15 265	19 082	3 053	

Процесс предоставления услуги может быть описан в 5 шагов, представленных в таблице 4.8.

Для реализации проекта необходима линейная организационная структура, которая идеально отвечает вызовам рынка, так как оперативно реагирует на изменения. Для выполнения всех трудовых функций на перспективу ближайших

Таблица 4.8 — Характеристика производственных операций

№	Наименование выполняемых операций	Наименование используемого оборудования	Объем продукции на выходе	Кол-во занятых чел.
1	Анализ процессов оборудования. Анализ целесообразности применения методов обучения с подкреплением	Персональный компьютер, аренда автомобиля	1	2
2	Реализация алгоритма управления в режиме «обучение»	Персональные компьютер	1	1
3	Запуск алгоритма управления в режиме «обучение» на оборудовании заказчика	Персональные компьютер, аренда автомобиля	1	1
4	Тестовая конвертация алгоритма к языку МЭК	Персональные компьютер	1	1
5	Подготовка документации	Персональные компьютер, принтер	1	1

пяти лет достаточно трех человек: главный инженер, ведущий Data Science-специалист, младший инженер, при условии, что в обязанности главного инженера включены управленческие функции. В таблице 4.9 представлены затраты на оплату труда рабочих, считая управленческие затраты включенными в основную заработную плату главного инженера.

Таблица 4.9 — Затраты на оплату труда персонала

Должность	Кол-во человек	Заработная плата	Отчисления на социальные нужды	Итог (з/п + отчисления), руб
Главный инженер	1	80 000	24 000	104 000
Ведущий Data Science-специалист	1	70 000	21 000	91 000
Младший инженер	1	10 000	3 000	13 000
Итого в месяц		160 000	48 000	208 000
Итого в год		1 920 000	576 000	2 496 000

В результате расчетов, общий фонд оплаты труда (ФОТ) за год составит 1 920 000 рублей, социальные отчисления 576 000 рублей.

4.4 Финансовый план

Определим себестоимость усредненной услуги как отношение суммы материальных затрат, без учета НДС, к объему производства за год:

$$\text{Себестоимость услуги} = \frac{\Sigma_{\text{расходы}}}{V_{\text{производства}}} = \\ = \frac{1856 + 1920000 + 200000 + 350000}{10} = 247186 \text{ рублей}$$

Развитие проекта предполагается за счет заемных средств путем оформления кредита. Наиболее оптимальным вариантом для работы компании станет оформление кредита, например, в ПАО Банк «ФК Открытие». Сумма кредита составляет 500 000 руб. исходя из первоначальных затрат в проект (единовременные затраты на оборудование и необходимый запас денежных средств на первый период для осуществления текущей деятельности). Ставка по кредиту составит 5.5%, срок кредита – 24 месяцев, ежемесячный платеж равен 22 048 рублей. При данных условиях привлечения денежных средств переплата по кредиту составит 29 152 рубля, выплаты за весь срок кредита составит 529 152 руб.

С учетом приведенных расходов в разделе план производства и динамики роста объема продаж, при условии сохранения стоимости и себестоимости услуг, построен прогноз доходов и расходов на 4 квартала с целью определения финансового результата проекта, таблица 4.10. В данной таблице приведены доходы и расходы организации без НДС во избежание искажения показателей управленческого учета.

В работе рассчитана чистая текущая стоимость проекта (NPV – Net Present Value), как разность дисконтированных денежных потоков поступлений и платежей, производимых в процессе реализации проекта за весь инвестиционный период. Значения приведены в таблице 4.11.

Вычислен дисконтированный период окупаемости инвестиций (срок возврата):

$$T_{\text{ок}} = x + \frac{NVP_x}{\text{ЧДД}_{x-1}} = 2 + \frac{339.31}{593.23} = 2.57(\text{месяца})$$

где x – последний месяц/год, когда NVP < 0; NVP_x – значение NVP в этом месяце/году; ЧДД_{x-1} – значение ЧДД в следующем периоде; T_{ok} – срок окупаемости. Очевидно, что чем меньше период возврата инвестиций, тем более экономически привлекательным является проект. В нашем случае NVP в первом же квартале

Таблица 4.10 — План прибылей и убытков на 4 квартала

Показатели, тыс. руб	Квартал				Всего
	I	II	III	IV	
1. Выручка от реализации	1 120	1 200	1 200	1 200	4 720
2. Себестоимость	494.37	741.56	741.56	741.56	2472
3. Затраты	891.74	891.74	891.74	891.74	3 567
3.1. Затраты на материалы	0.46	0.46	0.46	0.46	1.86
3.2. Амортизация	3.27	3.27	3.27	3.27	13.1
3.3. Затраты на оплату труда с отч.	624	624	624	624	2 496
3.4. Общепроизводственные затраты	122	122	122	122	488
3.4.1. Аренда помещения	72	72	72	72	288
3.4.2. Коммунальные услуги	50	50	50	50	200
3.5. Транспортные расходы	20	20	20	20	80
4. Валовая прибыль (1-3)	228.26	308.26	308.26	308.26	1 153
5. Коммерческие затраты	22.65	22.65	22.65	22.65	91
6. Прибыль от продаж (4-5)	205.61	285.61	285.61	285.61	1 062
7. Выплаты по кредиту	66.14	66.14	66.14	66.14	264
8. Прибыль до налогообл. (6-7)	139.47	219.47	219.47	219.47	798
9. Налог на прибыль, 20%	27.89	43.89	43.89	43.89	159
9. Чистая (нераспр.) прибыль	111.58	175.58	175.58	175.58	638

имеет положительное значение, поэтому потребовалось дополнительно рассчитать значение NVP для первых трех месяцев предоставления услуги. Первые 2 месяца NVP имеет отрицательное значения.

Чистый дисконтированный поток за четыре квартала с учетом дисконтирования под 20% составит 1 830 тыс. руб. Положительное значение NPV свидетельствует о целесообразности принятия решения о финансировании и реализации проекта. При сравнении нескольких инвестиционных вариантов показателя внутренней рентабельности проекта (IRR) служит критерием отбора более эффективного варианта. На данном этапе нет необходимости рассчитывать данный параметр. Так же стоит отметить, что по полученным расчетам окупаемость проекта наступает в 1 квартале, что в действительности очень тяжело осуществимо, но так как расчет производился в учебных целях, работа подтверждает экономическую целесообразность оказания услуги по разработке алгоритмов управления с использованием методом обучения с подкреплением.

Таблица 4.11 — План прибылей и убытков на 4 квартала

Показатели, тыс. руб	Квартал			
	I	II	III	IV
1. Поступление денежных средств	1 900	1 500	1 500	1 500
1.1. Поступление денежных средств от продажи продукции	1 400	1 500	1 500	1 500
1.2. Поступление денежных средств от кредита	500	0	0	0
2. Производственные и общехозяйственные расходы	624.65	624.65	624.65	624.65
2.1. Оплата труда	480	480	480	480
2.2. Оплата общепроизводственных расходов	122	122	122	122
2.3. Оплата коммерческих рассходов	22.65	22.65	22.65	22.65
2.4. Транспортные расходы	20	20	20	20
3. Покупка оборудования	103.184	0	0	0
4. Уплата налогов	171.89	187.89	187.89	187.89
4.1. Отчисления на соц.нужды	144	144	144	144
4.2. Налог на прибыль	27.89	43.89	43.89	43.89
5. Всего отток денежных средств (2+3+4)	905.72	812.54	812.54	812.54
6. Погашение кредита	66.14	66.14	66.14	66.14
7. Чистый денежный поток (1-5-6)	924.14	621.32	621.32	621.32
8. Дисконтированный денежный поток (20%)	739.31	431.47	359.56	300.15
9. Дисконтированный денежный поток нарастающим итогом (NVP)	739.31	1 170.78	1 530,34	1 830.49

Вывод по главе 4. В ходе составления бизнес-плана по коммерциализации проекта был сделан вывод о целесообразности оказания услуги разработки регуляторов на базе методов обучения с подкреплением для малого и среднего бизнеса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Introduction to reinforcement learning. T. 135 / R. S. Sutton, A. G. Barto [и др.]. — MIT press Cambridge, 1998.

2. *Sutton R. S.*, *Barto A. G.* Reinforcement learning: An introduction. — MIT press, 2018.

3. Reinforcement learning: A survey / M. L. Littman, A. W. Moore [и др.] // Journal of artificial intelligence research. — 1996. — Т. 4, № 1. — С. 237—285.

4. *Otterlo M. van* Reinforcement learning: State-of-the-Art. — Springer Berlin Heidelberg, 2012.

5. *Lewis F. L.*, *Vrabie D.*, *Vamvoudakis K. G.* Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers // IEEE Control Systems Magazine. — 2012. — Т. 32, № 6. — С. 76—105.

6. *Bertsekas D. P.*, *Tsitsiklis J. N.* Neuro-dynamic programming. — Athena Scientific, 1996.

7. *Powell W. B.* Approximate Dynamic Programming: Solving the curses of dimensionality. T. 703. — John Wiley & Sons, 2007.

8. A survey on policy search for robotics / M. P. Deisenroth, G. Neumann, J. Peters [и др.] // Foundations and trends in Robotics. — 2013. — Т. 2, № 1—2. — С. 388—403.

9. *Bellman R.* Dynamic Programming. — 1-е изд. — Princeton, NJ, USA: Princeton University Press, 1957.

10. *Bellman R.* A Markovian decision process // Journal of mathematics and mechanics. — 1957. — Т. 6, № 5. — С. 679—684.

11. *Howard R. A.* Dynamic Programming and Markov Processes. — Cambridge, MA: MIT Press, 1960.

12. *Bellman R.*, *Dreyfus S.* Functional approximations and dynamic programming // Mathematical Tables and Other Aids to Computation. — 1959. — С. 247—251.

13. *Werbos P. J.* Building and Understanding Adaptive Systems: A Statistical/Numerical Approach to Factory Automation and Brain Research // IEEE Transactions on Systems, Man, and Cybernetics. — 1987. — Т. 17, № 1. — С. 7—20. — DOI 10.1109/TSMC.1987.289329.

14. *Watkins C. J. C. H.* Learning from delayed rewards. — 1989.

15. *Kober J.*, *Bagnell J. A.*, *Peters J.* Reinforcement learning in robotics: A survey // The International Journal of Robotics Research. — 2013. — Т. 32, № 11. — С. 1238—1274.

16. Meta Learning via Learned Loss / S. Bechtle [и др.] // International Conference on Pattern Recognition, ICPR, Italy, January 10-15, 2021. — 2021.

17. *Yu C.*, *Liu J.*, *Nemati S.* Reinforcement learning in healthcare: A survey // arXiv preprint arXiv:1908.08796. — 2019.

18. Deep reinforcement learning for dialogue generation / J. Li [и др.] // arXiv preprint arXiv:1606.01541. — 2016.

19. *Paulus R.*, *Xiong C.*, *Socher R.* A deep reinforced model for abstractive summarization // arXiv preprint arXiv:1705.04304. — 2017.

20. Exploring applications of deep reinforcement learning for real-world autonomous driving systems / V. Talpaert [и др.] // arXiv preprint arXiv:1901.01536. — 2019.

21. Optimization of global production scheduling with deep reinforcement learning / B. Waschneck [и др.] // Procedia Cirp. — 2018. — Т. 72. — С. 1264—1269.

22. *Werbos P.* Approximate dynamic programming for realtime control and neural modelling // Handbook of intelligent control: neural, fuzzy and adaptive approaches. — 1992. — С. 493—525.

23. Adaptive control: algorithms, analysis and applications / I. D. Landau [и др.]. — Springer Science & Business Media, 2011.

24. *Dulac-Arnold G.*, *Mankowitz D.*, *Hester T.* Challenges of real-world reinforcement learning // arXiv preprint arXiv:1904.12901. — 2019.

25. A Lyapunov-based Approach to Safe Reinforcement Learning / Y. Chow [и др.]. — 2018. — arXiv: 1805.07708 [cs.LG].

26. *Lee J. Y.*, *Sutton R. S.* Integral Policy Iterations for Reinforcement Learning Problems in Continuous Time and Space // CoRR. — 2017. — Т. abs/1705.03520. — arXiv: 1705.03520. — URL: http://arxiv.org/abs/1705.03520.

27. *Bajpai R.*, *Reuss M.* A mechanistic model for penicillin production // Journal of Chemical Technology and Biotechnology. — 1980. — Т. 30, № 1. — С. 332—344.

28. *Patnaik P. R.* Penicillin fermentation: mechanisms and models for industrial–scale bioreactors // Critical reviews in microbiology. — 2001. — Т. 27, № 1. — С. 25—39.

29. Nonlinear dynamics of immunogenic tumors: parameter estimation and global bifurcation analysis / V. A. Kuznetsov [и др.] // Bulletin of mathematical biology. — 1994. — Т. 56, № 2. — С. 295—321.

30. *Consulting J. P̃.* Экономический эффект от цифровизации отраслей реального сектора экономики в России. Требования к сетям связи нового поколения. — январь 2021.