

## МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОГО УПРАВЛЕНИЯ РОБОТОТЕХНИЧЕСКИМИ И МЕХАТРОННЫМИ СИСТЕМАМИ. ОБЗОР

Зайцева Ю.С.<sup>1</sup>

(Институт проблем машиноведения РАН, Санкт-Петербург)

Искусственный интеллект стремительно завоевывает все больше областей для применения, успешно справляясь с задачами, поставленными инженерами-проектировщиками. Закономерно, что сложные задачи управления требуют новых современных решений. В этом обзоре представлены не только последние разработки по управлению мехатронными системами, но и показана связь методов классической теории автоматического управления и машинного обучения. Читателю кратко описаны такие известные методы классической теории управления как оптимизация, адаптация, нечеткая логика, на фундаменте которых расположились искусственные нейронные сети и обучение с подкреплением, а также представлены последние достижения по применению интеллектуального управления для ряда актуальных задач в различных областях техники. Проведенный анализ литературы показывает, что будущие исследования направлены на все большую степень автоматизации и автономии объектов управления, а их свойства и характер функционирования должны приблизиться к человеческим очертаниям интеллекта.

Ключевые слова: искусственный интеллект, машинное обучение, нейронные сети, оптимизация, адаптация.

### 1. Введение

С точки зрения автоматике стратегию интеллектуального управления можно трактовать достаточно гибко. Эта концепция отличается от классического управления с обратной связью тем, что динамически постулируется в режиме реального времени, пока система выполняет полезные функции. Так, классическое адаптивное управление может рассматриваться как интеллектуальное, если закон управления или стратегия управления системой постоянно обновляются. С точки зрения классификации такие системы можно отнести к пограничным [1]. При анализе литературы, посвященной ин-

---

<sup>1</sup>Юлия Сергеевна Зайцева, к.т.н. (juliazaytsev@gmail.com).

теллектуальному управлению мехатронными системами, хорошо прослеживается развитие методов, берущих начало из вычислительной математики, переходящих затем в область программирования, обработки и анализа большими массивами данных, и, в последнее время активно применяющихся в области управления физическими инженерными системами. Взрывной рост литературы приходится на начало 2010-х годов и не перестает уменьшаться.

В робототехнике и мехатронике применяют различные подходы и направления искусственного интеллекта, такие как машинное обучение, искусственные нейронные сети (НС), нечеткая логика, эволюционные вычисления и генетические алгоритмы. В область машинного обучения входят такие методы как классическое обучение (с учителем и без учителя), ансамблевые методы, обучение с подкреплением, НС и глубокое обучение. Алгоритмы интеллектуального управления используют различные оптимизационные, статистические и вероятностные методы для извлечения полезных закономерностей из произведенных действий и анализа больших наборов данных [2]. Если классическое обучение и ансамбли в основном применяют для обработки и классификации наборов данных, то НС и обучение с подкреплением подходят для применения в области автоматического управления, планирования и логистики [3, 4, 5]. При этом НС особенно эффективны при работе со сложными данными и невыделенными признаками.

Целью данного обзора является описание наиболее используемых современных интеллектуальных методов управления и работ с их применением в области мехатроники, выявление тенденций и открытых вопросов в интеллектуальном управлении.

## 2. Методы интеллектуального управления.

Хотя интеллектуальное управление выделяют в отдельную область, новые концепции, такие как, например, нейронные сети сами по себе нежизнеспособны. Чтобы применять

их в составе контура управления приходится задействовать различные математические методы, лежащие в основе теории классического автоматического управления. Таким образом, интеллектуальное управление является комбинированным методом, в котором активно задействованы методы автоматического регулирования, при этом объект управления способен обучаться на предыдущем своем опыте, что дает ему возможность выполнять эту задачу с каждым разом лучше. Рассмотрим наиболее широко используемый методы современной теории управления, используемые в интеллектуальном подходе. Некоторые из них уже давно известны и не требуют подробного описания, поэтому больше внимания в этой работе уделяется обучению с подкреплением и нейронным сетям.

- Методы адаптивного управления. Адаптивное управление, так же как оптимальное, имеет хорошо разработанную теоретическую базу и строгое математическое обоснование [6]. В рамках классической теории автоматического управления этот метод стал отправной точкой к интеллектуальному управлению, так как синтез адаптивных регуляторов обеспечивает заданное качество работы системы при изменении ее параметров и учета влияния окружающей среды, а принцип адаптации также лежит в основе интеллектуального управления. Однако за свою историю развития принцип адаптации прошел эволюцию от самонастраивающихся регуляторов до адаптивного обучения [7]. Однако, как показано в [8], наличие теории не гарантирует ее применение на практике, особенно в реальных условиях построения и эксплуатации систем управления.
- Методы оптимизации. Исторически оптимизационные методы появились значительно раньше, чем машинное обучение, и использовались для нахождения экстремума функции [9]. Теория оптимальности легла в основу большинства задач машинного обучения, которые в

общем виде можно сформулировать как минимизацию некоторого функционала  $J$  по некоторому параметру  $\theta$ :  $J(\theta) \rightarrow \min_{\theta \in X}$ . Вид минимизируемой величины зависит от метода машинного обучения. Так, например, в случае задачи классификации или регрессии минимизируют ошибку предсказания на имеющейся выборке, в теории обучения с подкреплением находят максимальную выгоду от действий объекта. Достигнуть эту цель можно с помощью какого-либо поискового алгоритма. Таким образом, у математической оптимизации появилось множество видов, методов и приложений.

- Искусственные нейронные сети. История развития искусственных нейронных сетей переживала взлеты и падения, пока к началу 90-х гг. не сформировался их теоретический фундамент. НС Структура нейрона описывается функцией:  $R^n \rightarrow R$  [10]. У нейрона есть  $n$  входов  $x_i$  с весом  $w_i$ , на который умножается сигнал, проходящий по связи. После этого взвешенные сигналы  $x_i w_i$  направляются в сумматор. Сумма взвешенных нейронов находится как:  $net = \sum_{i=1}^n w_i x_i = w^T x$ . Далее эта взвешенная сумма проходит через функцию активации  $\phi(net)$ , которая преобразует взвешенную сумму в число, подаваемое на выход нейрона. Простая схема работы нейрона показана на рис. 1, где в качестве функции активации используется функция единичного скачка.

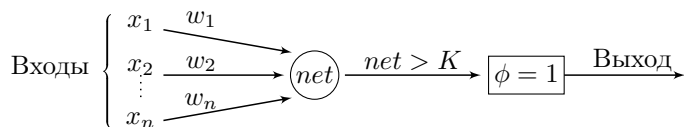


Рис. 1. Структура нейрона

Задача обучения сводится к определению таких архитектуры и весов, при которых ошибка обучения стремится к минимуму для всех возможных обучающих выборок.

Процесс обучения нейронной сети - это поиск набора весов, преобразующих входной сигнал в нужный выходной. Поскольку функция ошибки обучения может иметь произвольный вид, обучение НС в общем случае описывается в терминах многоэкстремальной невыпуклой задачей оптимизации.

Применение НС эффективно для управления в нелинейных и зашумленных системах, обеспечивает адаптивность, а после обучения НС может работать в режиме реального времени. Непрерывно развивающиеся структуры и свойства НС направлены на преодоление имеющихся недостатков. Так, эвристические подходы могут привести к неоднозначным решениям и тупиковым ситуациям, требуют подготовки обучающей выборки. Весомые для робототехники недостатки заключаются в продолжительных временных затратах на выполнение обучения, однозначно непредсказуемый результат обучения увеличивает риск для управления дорогостоящими объектами, реализация НС на данный момент возможна в виде сверхбольших интегральных схем.

- Методы нечеткой логики. Аппарат нечеткой логики, предложенный Л. Заде, базируется на понятии нечеткого множества как объекта с функцией принадлежности элемента ко множеству, принимающей значения на интервале  $[0,1]$  [11, 12]. Нечеткие логические рассуждения оказались возможным представить в виде НС. Для этого роль функции активации нейрона должна выполнять функция принадлежности, связи между нейронами – передача сигнала, а логические t-нормы и k-нормы представляют как специальные виды нейронов. На данный момент разработано много нейро-нечетких сетей, которые приближенно описываются универсальной формой аппроксиматоров [13]. Их характерной чертой является человекоподобный стиль рассуждений и коннекционист-

ская структура НС.

- Обучение с подкреплением. В концепции обучения с подкреплением существуют понятия агента и среды. Агент в процессе взаимодействия с неопределенной динамической средой изучает так называемую оптимальную политику принятия решений, отображающую состояние агента после выполнения оптимальных действий [14, 15, 16]. Схематически процесс обучения с подкреплением изображена на рис. 2.

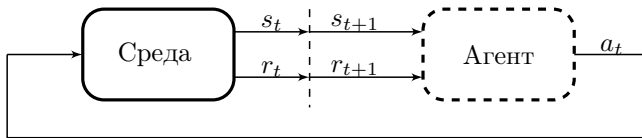


Рис. 2. Структура обучения с подкреплением

В [17] описан общий подход к разработке политики управления для различных роботов, сред и манипулятивных задач. Среда может быть как физической, так и виртуальной, реализованной в виде компьютерной программы. В работе [18] авторами продемонстрирована эффективность обучения с подкреплением на примере перевернутого маятника, управляемого ПИД-регулятором, реализованного виртуально. Важным аспектом во взаимодействии агента и среды является режим реального времени [19]. Агент получает наблюдение реакции среды в режиме реального времени, когда предпринимаются случайные или неоптимальные действия, и оценку этих действий в виде награды. Итеративный процесс обучения с подкреплением обычно состоит из двух шагов: оценка политики и последующее ее улучшение. В литературе понятие обучения с подкреплением упоминается под несколькими альтернативными названиями: приближенное динамическое программирование (approximate dynamic programming) [20],

нейродинамическое программирование (neuro-dynamic programming) [21], адаптивный критик (adaptive critic) [22]. Вербос (Werbos) использовал широкий термин «приближенное динамическое программирование обучения с подкреплением» [23]. Приближенное динамическое программирование в широком смысле включает в себя все доступные с вычислительной точки зрения инструменты для нахождения наиболее точного возможного решения уравнения Беллмана, как это требуется для оптимальности, связанной с динамическим программированием. С точки зрения теории управления обучение с подкреплением можно рассматривать как адаптивное оптимальное управление, которое обеспечивает адаптивный регулятор, сходящийся к оптимальному [24].

Общая стохастическая система дискретного времени в задаче обучения с подкреплением, или управления с обратной связью, выражается в виде одношаговой марковской модели, в которой полностью наблюдаемая система  $X_{t+1} = F(X_t, u_t, w_t)$  развивается на основе  $u_t$ , случайная величина  $w_t$  представляет собой неопределенность или возмущение. Цель в каждый момент времени  $t$  состоит в том, чтобы найти политику  $\pi$ , указывающую, какое решение следует принять при заданном состоянии, чтобы минимизировать (или максимизировать) сумму всех штрафов (или вознаграждений):

$$\min_{\pi(\cdot)} E \left[ \sum_{\tau=t}^{\infty} \gamma^{\tau-t} U(X_{\tau}, u_{\tau} = \pi(X_{\tau})) \mid X_t \right],$$

где  $0 < \gamma \leq 1$  – коэффициент дисконтирования,  $U$  – функция стоимости.

Математическая формулировка такой последовательной задачи принятия решений называется Марковским процессом принятия решений (МПП) [25]. Сформулированная задача МПП решается именно с помощью динамического программирования, в котором функция ценности

(стоимости), оценивающая сумму всех будущих затрат при оптимальной политике принятия решений, вводится в следующей временной рекурсивной форме (оптимальное уравнение Беллмана):

$$(1) \quad V^*(X_t) = \min_{\pi(\cdot)} \{U(X_t, \pi(X_t)) +$$

$$(2) \quad \gamma E[V^*(X_{t+1}) \mid X_t, \pi(X_t)]\}.$$

Соответствующая (2) оптимальная политика получается путем решения следующей задачи оптимизации:

$$(3) \quad \pi^*(X_t) = \arg \min_{\pi(\cdot)} \{U(X_t, \pi(X_t)) +$$

$$(4) \quad \gamma E[V^*(X_{t+1}) \mid X_t, \pi(X_t)]\}.$$

Точное решение (2) нельзя получить для большинства практических задач из-за вычислительной сложности, связанной с огромными размерами пространства состояний и действий, что называется «проклятием размерности». Приближенное динамическое программирование обучения с подкреплением было создано как практический метод развития политики принятия решений в направлении оптимальной политики в режиме онлайн с использованием обратной связи. Уравнение Беллмана заменяется уравнением рекурсии, а функция значения итеративно обновляется в зависимости от ошибки прогнозирования (temporal difference). Для представления функции ценности или политики действия используется какая-то аппроксимация, например, параметрическая модель, нейронная сеть, усреднение ближайшего соседа, ряд Тейлора и т.д. Основная идея приближенного динамического программирования заключается в приближенном вычислении функции качества, однако из-за случайного выбора начальных весов ему нужно много раз учиться сходиться. Решение этой проблемы предло-



жено в работе [26], где строится эквивалент между начальным действием НС и ПИД-регулятором, в результате чего ПИД-регулятор может управлять выбором начальных весов.

Если динамика модели системы неизвестна, то применяют так называемое Q-обучение, при котором вводится функция ценности с аргументом состояние-действие [27]:

$$(5) \quad Q^*(X_t, u_t) = U(X_t, u_t) + \gamma \min_{u'} E[Q^*(X_{t+1}, u_t)],$$

$$(6) \quad \pi^*(X_t, u_t) = \arg \min_{u(\cdot)} Q^*(X_t, u_t).$$

Важный вопрос в обучении заключается в том, какое решение должен принять агент, чтобы быстро изучить новую информацию, которая определяет политику действий. Сходимость Q-функции (5) выполняется только тогда, когда все действия имеют ненулевую вероятность быть выбранными политикой принятия решений, т.е. с точки зрения обучения, решения должны приниматься с целью получения более ценной информации о модели и неопределенности. С другой стороны, чтобы добиться хороших результатов, следует выбрать лучший вариант, исходя из текущих знаний.

Вероятностный вывод для обучения это эффективный подход, в котором ставится задача байесовской оценки с регрессией гауссовского процесса, дает полностью детерминированный приближительный вывод для оценки политики, что делает его эффективным с точки зрения вычислений [28]. Для этого подхода требуется сценарий конкретной задачи. Если агенту ставится новая цель, то модель нужно обучать заново. В [28] вероятностный вывод для обучения расширен для случая настройки регулятора с линейной обратной связью. Для этого к урав-

нению Беллмана добавляется улучшенная политика:

$$\pi(s) = \arg \max_{a \in A(s)} \int P_{s,s'}^a [R_{s,s'}^a + \gamma V(s')] ds',$$

где  $P_{s,s'}^a$  – вероятность перехода в состояние  $s'$  при выполнении действия  $a$ ,  $V(s')$  – функция ценности. Предложенная структура обучения проверена на системе управления квадрокоптером без заданной динамики и показала сходимость обучения в зависимости от числа итераций.

В [29] для решения классической задачи оптимального управления и обучения с подкреплением описан метод динамического программирования Гауссовских процессов и алгоритм обучения с подкреплением на основе приближенной функции ценности. Ожидаемую ценность в работе предлагается вычислять как:

$$U(\tilde{x}) = \rho E_V [V_k^*(\tilde{x}) | X_k] + \frac{\beta}{2} \log(\text{var}_V [V_k^*(\tilde{x}) | X_k]),$$

где  $\tilde{x} \in \tilde{X}_k$ ,  $\tilde{x}$  – вектор состояния,  $\rho$ ,  $\beta$  – весовые коэффициенты,  $E_V$  – нормальное распределение,  $\text{var}_V$  – дисперсия. Функция ценности состояния  $V_k^*$  является аргументом функций математического ожидания и ковариации.

### 3. Применение интеллектуальных методов для решения различных задач управления

#### 3.1. ЗАДАЧИ СТАБИЛИЗАЦИИ И ПРОГРАММНОГО УПРАВЛЕНИЯ

Задачи стабилизации и программного управления подразумевают наличие обратной связи в контуре. Обычно вектор состояния системы недоступен для измерения, поэтому управление строится по доступному измеряемому выходу. Стандартными задачами мехатроники и робототехники являются такие задачи как стабилизация скорости и отслеживание

траектории. Эти переменные достаточно легко поддаются измерению на выходе системы. В [30] рассмотрено управление квадрокоптером методом машинного обучения, в котором политика  $\pi_\theta$  параметризована  $\theta$  и дифференцируема по параметрам. Целевой функционал  $J(\pi_\theta)$  дифференцируем по  $\theta$ , т.е. оптимизируется градиентным методом (policy gradient method). Для этого используется формула градиента средней награды по параметрам стратегии и ее стохастические оценки. Наиболее распространенный вид оценщика градиента записывается в следующей форме:

$$\hat{g} = \hat{E}_t[\nabla_\theta \log_{\pi_\theta}(a_t|s_t)\hat{A}_t],$$

где  $\hat{A}(s_t, a_t) = Q(s_t, a_t) - \hat{V}(s_t)$  – оценка функции преимущества на временном шаге  $t$ ,  $\hat{E}_t$  – эмпирическое среднее по конечному набору выборок, полученное путем чередования процессов генерации выборок и улучшения политики  $\pi_\theta$ . В данной задаче обучения с подкреплением динамическая модель неизвестна, возможно недифференцируема, и которую необходимо обучить, что приведет к увеличению дисперсии оценок градиента. В связи с чем, авторы разработали более надежный метод оптимизации политики путем монотонного повышения производительности агента. Для этого предложена следующая целевая функция, после дифференцирования которой получают (??):

$$J^{CLIP}(\theta) = \hat{E}_t[\min(r(\theta)\hat{A}_t, \text{clip}(r(\theta), 1 - \epsilon, 1 + \epsilon))],$$

где  $\epsilon$  – гиперпараметр,  $r(\theta) = \pi_\theta/\pi_{\theta_{old}}$ . Функция награды выбрана как можно проще в виде:

$$r_t(e_{xt}, e_{yt}, e_{zt}) = a - \sqrt{e_{xt}^2 + e_{yt}^2 + e_{zt}^2},$$

где  $e_{xt}$ ,  $e_{yt}$ ,  $e_{zt}$  – рассогласование позиций квадрокоптера по 3-м координатам,  $a$  – постоянная величина, используемая для обеспечения того, чтобы квадрокоптер получал награду в ограниченной области полета.

В [31] рассмотрено энтропийно-адаптивное обучение с подкреплением ходьбе мобильного робота-треноги с вибрационным приводом с мягкой мембраной. Для такого типа объектов затруднена разработка регулятора с обратной связью и сбор данных для обучения, поэтому авторы используют безмодельный метод обучения. Регулятор определяется как политика нормального распределения Гаусса:  $f_\phi(s_t) = (\mu_t, \sigma_t)$ , где  $\mu_t$  и  $\sigma_t$  – среднее и стандартное отклонение,  $\phi$  – параметр регулятора. Функция  $f_\phi$  задается как НС. Политика действий описывается как  $N(a_t; f_\phi(s_t))$ . Функция награды предлагается в виде:

$$r(s_t) = -d_t - \delta\theta_t + c,$$

где  $\delta\theta_t$  – угловая разность между текущим и желаемым положением,  $d_t$  – среднеквадратическая ошибка между текущим положением робота и конечным положением,  $c$  – коэффициент. Оптимальная политика действий для максимизации энтропии определяется как:

$$\pi_\alpha^* = \arg \max_{\pi} E_{\tau \sim P, \pi} \left[ \sum_{t=0}^{\infty} \gamma^t (\hat{r}(s_t, a_t) + \alpha H(\pi(\cdot | s_t))) \right],$$

где  $\hat{r}(s, a) = E_{s' \sim P(\cdot | s, a)} [r(s')]$ , температура энтропии  $\alpha \in [0, \infty)$ ,  $H(\pi_\phi(\cdot | s)) = E_{\alpha \sim \pi_\phi} [-\log(\pi_\phi(a | s))]$ . Согласно используемому адаптивному мягкому алгоритму актер-критик необходимо минимизировать функционалы для каждой функции ценности методом стохастического градиентного спуска.

В том случае, когда цель управления меняется во времени, описанный выше подход обучения с подкреплением не применим. Решение этой проблемы предложено в [32], где управлять набором состояние-действие-вознаграждение для обучения имитации постоянной цели в каждом наборе. Такой подход был проверен на системе управления скоростью с различными заданными значениями на модели и реальном транспортном средстве.

Важным подходом в построении интеллектуальных систем являются байесовские статистические методы [33]. Исходя из последних данных по изучению мозга человека появилась гипотеза о том, что мозг может использовать байесовские внутренние модели во время определенных типов сенсомоторного обучения, чтобы оптимизировать выполнение конкретной задачи, и возникающая активность определенных нейронных ансамблей может быть смоделирована как совместные байесовские процессы. Эти концепции обработки нейронных сигналов могут применяться в приложениях от реабилитационной инженерии до искусственного интеллекта.

В [34] предложен новый подход упреждающего управления роботом в частично наблюдаемой среде методами машинного обучения, состоящих из нескольких шагов. В начале вычисляется эпизодическая память  $E$ , состоящая из временной последовательности  $n$  событий  $e$  и цели  $g$ , которую робот преследовал во время каждого эпизода:  $E = \{(e_1, e_2, \dots, e_n), g\}$ . Эпизоды разделены на основе целей, которые используются в качестве ключей для извлечения соответствующих эпизодов из памяти. Событие состоит из наблюдения  $o = \{z_1, z_2, \dots, z_m\}$ , полученных от  $m$ -датчиков измерений, действия, выполняемые мотором  $b = \{\sigma_1, \sigma_2, \dots, \sigma_\beta\}$  и награды  $r$ :  $e = \{o, b, r\}$ . Цель робота  $g$  задается последовательностью  $\{z_1^g, z_2^g, \dots, z_m^g\}$ . Упреждающее поведение робота вычисляется с помощью выборки событий, воспоминания эпизода, сопоставления событий и выбора поведения. На каждом этапе используется разное машинное обучение: временное разностное обучение (TD learning), обучение на примерах, рекурсивная байесовская оценка и МПР. Функция полезности  $i$ -ого события предложена в виде:

$$U(e_i) = r_i - \sum_{e' \in E} p(e' | b_{i+1}, e_i) U(e'),$$

где  $p$  – вероятность перехода. Лучшая стратегия действий за-

писывается как:

$$b^* = \arg \max_b \frac{1}{|\Gamma^+(b)|} \sum_{E \in \Gamma^+(b)} \sum_{e' \in E} p(e' | b, \hat{e}_{[E]}) U(e'),$$

где функция  $\Gamma^+(b)$  возвращает подмножество эпизодов, содержащее совпавшие события.

Интеллектуальное производство электроэнергии стремительно развивается и привлекает внимание исследователей. Классические методы становятся недостаточными в этой области из-за растущих масштабов систем и различными неопределенностями. Авторы [35] видят альтернативу этим методам в расцветающей эре машинного обучения и управления на основе данных. Задачи управления, требующие учета стохастических внешних воздействий, тяжело решаются классическими методами, имеют сложные структуры управления, не всегда удовлетворяющие качеству работы [36, 37]. К таким устройствам относятся ветрогенераторы, преобразующие энергию ветра в механическую энергию. Для них методы интеллектуального управления особенно актуальны и эффективны. В [37] исследуется проблема управления шагом ветряной турбины, зависящим от нелинейной динамики ветроприемника, связи переменных и неопределенностей окружающей среды. Для ее решения предлагается улучшенное по скорости гибридное обучение, состоящее из ПИД-регулятора и обучающимся наблюдателем. Такая структура управления оказалась эффективнее чем нечеткий регулятор и НС, и позволяет снизить погрешность выходной мощности на 41%. Архитектура

управления описывается следующими выражениями:

$$\begin{aligned}
 s_t &= f_{se}(P_{err}(t_i), P_{err}(t_{i-1}), v_w(t_{i-1})), \\
 T_{s_{t-1}, a_{t-1}}(t_i) &= f_{pu}(r_t, T_{s_{t-1}, a_{t-1}}(t_{i-1})), \\
 \theta_R(t_i) &= f_{as}(T_{s_t}(t_i), \epsilon(t_i), eW(t_i)), \\
 \theta_{ref}(t_i) &= \theta_R(t_i) - \theta_{PID}(t_i), \\
 [\epsilon(t_i), eW(t_i)] &= \begin{cases} f_{LO}(P_{err}(t_i), P_{err}(t_{i-1}) \dots P_{err}(t_{i-N})) \\ [\epsilon(t_{i-1}), eW(t_{i-1})], \end{cases} \\
 [T(t_i), bT(t_i)] &= \begin{cases} f_{LOT}(P_{err}(t_i), P_{err}(t_{i-1}) \dots P_{err}(t_{i-N})) \\ [T(t_{i-1}), bT(t_{i-1})]. \end{cases}
 \end{aligned}$$

где  $P_{err}$  – ошибка между заданной и выходной мощностью,  $v_w$  – скорость ветра,  $f_{se}$  – функция оценщика состояния,  $f_{pu}$  – функция обновления политики,  $f_{as}$  – функция выбора действия,  $f_{LO}$  – функция, которую реализует обучающий наблюдатель для обновления пары  $[\epsilon, eW]$ ,  $f_{LOT}$  – функция, используемая обучающим наблюдателем для обновления таблицы  $T$ ,  $N$  – число итераций,  $bT$  – наилучшая политика. Состояние системы оценивается по ошибке мощности, скорости ветра и их производных. Функция награды определена в виде сигнум-функции, знак которой зависит от знака производной от ошибки мощности. Обучающийся наблюдатель представляет собой модуль стратегии управления, который получает на вход ошибку выходной мощности и политику действий. Он генерирует наилучшую политику и настраивает входные параметры. Модуль выбора действий выбирает случайное действие с вероятностью  $\epsilon$  и жадное действие с вероятностью  $1 - \epsilon$  для исключения локальных минимумов.

НС справляются с решением широкого класса задач, в которых может присутствовать полная неопределенность как динамики агента, так среды. Пример управления такой сложной нелинейной системы высокого порядка, описываемой уравнениями в частных производных, рассмотрен в [38]. Авторами представлена структура прогнозирующего управления на основе модели глубокого обучения. В [39] показано

выполнение процесса лазерной сварки при одновременном использовании НС для предсказания качества сварочного шва по обработанным видеоизображениям и зависящей от нее подаваемой мощности лазера. НС актер-критик следует стохастической политике управления  $\pi(a|s)$ , выбранной из распределения Гаусса  $N(\mu, \sigma^2)$ , где  $\mu$  и  $\sigma$  являются линейными комбинациями вектора веса и вектора признаков. За меру вознаграждения выбрана оценка целостности сварочного шва:

$$(7) \quad r = -0.5 + 1/[1 + \exp(|d^* - d|)],$$

где  $d$  и  $d^*$  – достигнутая и заданная глубина сварки соответственно. Сигмоидная форма функции (7) ограничена в диапазоне  $[-0.5, 0]$  для обеспечения сходимости алгоритма. Критик при этом оценивает производительность алгоритма, после чего происходит обновление функции ценности и весов.

В [40] авторами представлена система управления автономным колесным мобильным роботом (autonomous wheeled mobile robot), состоящая из адаптивного нейросетевого контура управления скоростью движения и самообучающегося контура отслеживания траектории с учетом динамических ограничений. Трехслойная НС, состоящая из актера, критика и теневого критика, в которой каждый скрытый нейрон имеет гиперболическую тангенциальную функцию активации [41]:

$$(8) \quad \bar{y}_j(n) = a \tanh(b \sum_{i=0}^I w_{ji}(n) \bar{x}_i(n)), \bar{x}_0(n) = 1, (a, b) > 0,$$

Каждый выходной нейрон имеет линейную функцию активации:

$$(9) \quad \bar{z}_k(n) = c \sum_{j=0}^J w_{kj}(n) \bar{y}_j(n), \bar{y}_0(n) = 1, c > 0$$

Помимо функций активации НС содержит такие компоненты их частных производных как две функции чувствительности ( $\partial \bar{z}_k(n)/\partial w_{kj}(n)$ ,  $\partial \bar{z}_k(n)/\partial w_{ji}(n)$ ) и функцию Якобиана ( $\partial \bar{z}_k(n)/\partial x_i(n)$ ). Задача НС состоит в прогнозировании состояния модели объекта управления и вычислении значения



Якобианов. Для отслеживани заданной скорости используется следующая функция полезности:

$$(10) \quad U(n) = 0.25(v(n) - v_d(n))^2 + 0.25(\omega(n) - \omega_d(n))^2,$$

где  $v_d$ ,  $\omega_d$  – заданная линейная и угловая скорость соответственно. Для достижения цели управления необходимо найти веса НС, соответствующие минимальным текущим и будущим значениям (10), для чего требуется минимизировать функционал вида:

$$J(n) = \sum_{k=0}^{\infty} \eta^k U(n+k) = U(n) + \eta J(n+1),$$

где  $\eta$  – коэффициент дисконтирования,  $0 < \eta \leq 1$ . Обучение НС проводится методом градиентного спуска:

$$\Delta \omega_{km}(n) = \alpha \frac{\partial J(n)}{\partial w_{km}(n)},$$

$$\Delta w_{sm}(n) = \beta \frac{\partial E(n)}{\partial w_{sm}(n)},$$

где  $\alpha$ ,  $\beta$  – скорость обучения,  $w_{km}$  – вес  $m$ -ого нейрона, связанный с  $k$ -ым выходом актера,  $E$  – ошибка для коррекции НС теневого критика,  $w_{sm}$  – вес  $m$ -ого нейрона, связанный с  $s$ -ым выходом теневого критика. Механизм самообучения построен путем идентификации специальной обратной скоростной и угловой модели робота. После вычисления ошибки инверсии корректируются веса в обратной скоростной модели. Таким образом, робот получил управление без регулятора, обучающих выборок и логических правил управления.

Много исследований с применением НС посвящено автоматизации пилотируемых и беспилотных авиационных систем за счет реализации адаптивных стратегий управления полетом [42, 43]. Особенно это актуально для автономной работы в сложных атмосферных условиях и системных отказах. Методы машинного обучения позволяют повысить точность

управления и реагировать на неизвестные нелинейные условия полета. Как показано в [43] НС в таком случае могут использоваться для реализации обучаемой системы управления полетом. Большие группы нейронных элементов управления обучаются требуемому поведению на основе измеренных данных. Качество обучения может затем оцениваться статистическими методами.

Благодаря своим способностям к обучению НС хорошо подходят для решения сложных задач нелинейного управления и отказоустойчивости. В [44] предложен подход нелинейного адаптивного управления полетом с линеаризацией обратной связи и использованием онлайн обучения в скользящем режиме. Алгоритм обучения в скользящем режиме основан на теории управления переменной структурой и рассматривает НС и ее обучение как процесс управления. Такой подход позволяет преодолеть летательному аппарату негативное влияние нелинейностей в динамике и внешних возмущений. В [45] рассмотрена программируемая логическая интегральная схема типа FPGA на скользящем режиме на основе рекуррентной вейвлет-нейронной сети для управления линейного ультразвукового двигателя. Для оценки нелинейных функций в модели системы, параметров системы и внешних возмущений используется НС. Алгоритм адаптивного обучения НС в режиме реального времени получен с использованием теоремы устойчивости Ляпунова. Высокая эффективность алгоритма проверена для задач позиционирования и отслеживания. Адаптивный скользящий режим на основе метода опорных векторов с нечеткими правилами для повышения быстродействия и точности управления двухзвенным роботом-манипулятором описан в [46]. Возможности этого метода позволяют хорошо обрабатывать нелинейные сигналы и имеют способность к обобщению, которая используется для уменьшения дребезга при управлении в скользящем режиме. Параметры регулятора оптимизированы с помощью гибридного алгоритма обучения, сочетающего в себе метод наименьших квадратов и генетиче-

ский алгоритм.

В [47] рассмотрен подход к проектированию ПИД-регулятора для системы с запаздыванием с помощью нового предиктора Смита на основе НС, что делает регулятор адаптивным к изменяющимся параметрам системы. В [48, 49] для разработки регулятора для сильно нелинейной системы управления используются спайковые НС, которые используют временные последовательности импульсов и могут выполнять более быстрые и сложные вычисления. Регуляторы на основе спайковых НС способны к онлайн обучению и самонастройки при переходе от моделирования к реальному миру. Такие структуры НС имеют меньше нейронов и нейронных связей по сравнению с предыдущими поколениями НС. Обучение спайковой НС в рамках данной работы проводилось с использованием эволюционных алгоритмов. Эффективность предложенного подхода проверена на модели движения роботизированной руки.

### 3.2. НАСТРОЙКА РЕГУЛЯТОРА.

В промышленности распространено применение ПИД-, адаптивных и нечетких регуляторов. Методы адаптивного управления позволяют изменять как структуру, так и параметры регулятора в зависимости от изменения параметров объекта управления или внешних возмущений. Историческая перспектива методов адаптивного управления и обучения представлена в обзорной работе [7]. Часто структура регуляторов фиксирована и настройка требуется только их параметров. Известны методы настройки регулятора с использованием описания динамики системы, однако в реальности их трудно получить, так как требуются глубокие знания о системе или обширные измерения в потенциально разомкнутом контуре. В первых работах в этой области был описан алгоритм настройки ПИД-регулятора по переходной характеристике модели объекта и по амплитуде и периоду автоколебаний замкнутой системы [50, 51]. Позднее были предложены адаптивные ПИД-регуляторы с алгоритмом идентифициру-

ющего адаптивного управления, коэффициенты которых настраиваются на основе оценок параметров модели [52, 53]. В некоторых случаях, особенно, когда система неустойчива, возможны только измерения по обратной связи. В связи с чем, их настройка становится трудоемким и неэффективным процессом в меняющихся условиях работы системы, поэтому прибегают к автоматизированным методам, которые позволяют быстро определять параметры регулятора в соответствии с поставленной задачей без участия человека. Работы в этой области начались с самонастраивающихся структур. В [54] предложен многопараметрический самонастраивающийся регулятор для управления технологическим процессом термопластавтоматов. Модель системы задана стохастической дискретного времени:

$$(11) \quad A(q^{-1})y(t) = B(q^{-1})u(t - d - 1) + C(q^{-1})e(t),$$

где  $y(t)$  – вектор выхода размерности  $p$ ,  $u(t)$  – вектор входа размерности  $s$ ,  $e(t)$  – белый гауссовский шум размерности  $p$ ,  $q^{-1}$  – оператор обратного сдвига,  $(q^{-1})y(t) = y(t - 1)$ ,  $d$  – чистое запаздывание. Используя процедуру самонастраивающегося прогноза с  $k$ -шаговой рекурсией модель (11) записывается в виде:

$$\hat{y}(t + k|t) = \sum_{i=1}^{n_a} \hat{A}_i \hat{y}(t + k - i|t) + \sum_{i=d}^{n_b} \hat{B}_i u(t + k - i) + \sum_{i=k}^{n_c} \hat{C}_i \hat{e}(t + k - i),$$

где  $k = 1, 2, \dots, d$ ,  $\hat{A}_i$ ,  $\hat{B}_i$ ,  $\hat{C}_i$  – оценки матриц системы (11). Тогда задача оптимизации записывается как:

$$J = \left\| D_0 \hat{B}_d u(t) + \hat{L}(t) \right\|_{Q_1}^2 + \left\| G_0 u(t) + \sum_{i=1}^r G_i u(t - 1) \right\|_{Q_2}^2.$$

Таким образом, проблема стохастической оптимизации переходит в проблему детерминированной оптимизации: 20

$\partial J / \partial u(t) = 0$ , где  $u(t)$  – выход самонастраивающегося регулятора:

$$u(t) = -[(D_0 \hat{B}_d)^T Q_1 D_0 \hat{B}_d + G_0^T Q_2 G_0]^{-1} \times \\ [(d_0 \hat{B}_d)^T Q_1 \hat{L}(t) + G_0^T Q_2 \sum_{i=1}^r G_i u(t-1)].$$

Результаты моделирования также указывают на то, что для удовлетворения требований к более сложным машинам следует использовать обучающийся регулятор с дополнительными функциями управления и координации.

Несмотря на свою простоту, ПИД-регулятор не теряет своей эффективности и продолжает использоваться в современных системах управления, а настройка его параметров производится теперь с помощью машинных методов обучения. В [55] для настройки ПИД-регулятора системы управления статического компенсатора реактивной мощности, подключенного к системе преобразования энергии ветра, применяются эвристический метод поиска гармонии (harmony search).

В [56] рассмотрена автоматизированная настройка ПИД-регулятора для системы подачи топлива в автомобиле. В работе сравниваются два метода настройки: безопасное активное обучение и байесовской оптимизации. Безопасность подразумевает ввод ограничений на такие параметры как, например, перерегулирование, которое может привести к потере устойчивости системы и аварийному режиму работы. Активное обучение предполагает минимизацию функции потерь, которая зависит только от параметров регулятора, ограничивает большую ошибку управления и колебательные управляющие

действия:

$$J = \max(J_1, J_2), \quad J \in [0, 1],$$

$$J_1 = \frac{1}{n} \sum_{k=1}^n (1 - \exp(-\frac{\theta_1}{2}(y(k) - r(k))^2)),$$

$$J_2 = \frac{1}{n} \sum_{k=1}^n (1 - \exp(-\frac{\theta_2}{2}(u(k) - \bar{u}(k))^2)),$$

где  $r(k)$ ,  $u(k)$ ,  $y(k)$  – входной сигнал с дискретным временем, выход регулятора и выход объекта управления соответственно,  $\theta_{1,2}$  – настраиваемые параметры,  $\bar{u}$  – скользящее среднее величины сигнала на выходе регулятора. Для обучения дискриминантной модели определяется следующая функция риска  $h$ :

$$h = \min(h_1, h_2),$$

$$h_1 = \frac{\max_{k \in [1, n]}(y(k)) - y_{\max}}{y_{\min \max} - y_{\max}}, \quad h_2 = 1 - \frac{J}{J_{\max}},$$

где  $y_{\max}$  – максимально допустимое выходное значение,  $y_{\min \max}$  – наименьшее ожидаемое максимальное выходное значение,  $J_{\max}$  – максимально допустимое значение функции потерь. Цель задачи байесовской оптимизации состоит в нахождении глобального минимума или максимума функционала при больших затратах на его оценку. В этой работе для минимизации выбран нижний доверительный предел (Lower Confidence Bound):

$$(12) \quad LCB(x) = \mu_{f_*}(x) - k\sigma_{f_*}(x),$$

где  $\mu_{f_*}$  – прогнозируемое значение для модели  $f$ ,  $k$  – настраиваемый параметр,  $\sigma_{f_*}$  – прогнозируемое стандартное отклонение для регрессии. Для обеспечения выборки безопасных для системы параметров авторы предложили комбинированный подход, заключающийся в использовании Байесовской оптимизации с дискриминантной моделью  $g$ . Отличие этого

подхода от активного обучения состоит в ограничении нижнего доверительного предела (12) оцениваемым риском:

$$x_{i+1} = \operatorname{argmin} LCB(x_*) \quad \text{при} \quad \mu_{g^*}(x_*) - \nu\sigma_{g^*}(x_*) \geq 0.$$

Оба подхода показали хорошие результаты и имеют дальнейшие перспективы улучшения. Метод безопасной байесовской оптимизации также использовался для настройки ПИ-регулятора системы управления квадроторами в [57]. В работе предложен алгоритм вычисления следующей оценки функционала с ограничениями на область безопасных параметров:

$$\begin{aligned} J(a_n) &= C(a_n) - 0.95C(a_0), \\ C(a_n) &= - \sum_k = 0^N x_k^T Q x_k + R u_k^2, \end{aligned}$$

где  $a_n$  – параметры регулятора,  $x$  – вектор состояния,  $Q$ ,  $R$  – положительно полуопределенные матрицы,  $N$  – временной горизонт прогнозирования,  $u$  – вход системы,  $n$  – номер итерации.

В [58] на основе метода МПР проведена настройка параметров двигателя внутреннего сгорания в режиме реального времени для достижения оптимального режима работы при любом стиле вождения.

В [59] вместо метаэвристического алгоритма предложен более эффективный метод опорных векторов для обучения и построения модели, прогнозирующей оптимальные параметры резонансного виброгасителя. Однако, адекватность этих моделей сильно зависит от количества и точности используемых для обучения баз данных. Хорошие результаты показал генетический алгоритм при настройке ПИД-регулятора автопилота [60]. Для моделирования и прогнозирования динамических характеристик релейно-реактивной машины в работе [61] предложено использовать адаптивную нечеткую нейронную систему на основе радиальных базисных функций. В исследуемой системе доступно косвенному измерению только

фазовое потокосцепление, по которому рассчитывается фазовый крутящий момент. Обучение машины знанию электромагнитных характеристик проводится с помощью иерархического самоорганизующего алгоритма.

Направление другой области искусственного интеллекта для синтеза и настройки ПИД-регулятора рассмотрено в [62, 63, 64, 65]. В [65] предложена структура планирования траектории с помощью обучения с подкреплением нечеткого регулятора. Модель динамики робота и регулятора описывается нечеткими правилами:

$$\begin{aligned} &\text{если } x_1(t) = M_1^i \text{ и } x_2(t) = M_2^i \text{ и } x_n = M_n^i, \\ &\text{тогда } \dot{x}(t) = A_i x(t) + B_i u(t), \\ &\quad u(t) = K_j x(t), \\ &\quad i = 1, 2, \dots, r \quad j = 1, 2, \dots, r, \end{aligned}$$

где  $x(t)$  – вектор состояния для  $i$ -ого правила,  $M_j^i$  – нечеткое множество,  $u(t)$  – вектор входа,  $r$  – число нечетких правил,  $A_i, B_i$  – матрицы,  $K_j$  – коэффициент обратной связи для  $j$ -ой линейной подсистемы. Для обучения регулятора используется итеративный алгоритм Q-обучения, благодаря которому агент выбирает оптимальную последовательность действий МПР. Цель алгоритма оценить каждое значение Q-функции для каждой пары состояние-действие  $(s_t, a_t)$ . Q-функция обновляется по следующему закону:

$$Q(s_t, a_t) = r_t + \gamma Q(s_{t+1}, a),$$

где  $r_t$  – вознаграждение за выполненное действие,  $\gamma$  – коэффициент дисконтирования. В зависимости от области препятствий  $S_d$ , желаемого положения  $S_t$  и безопасной области  $S_s$  функция вознаграждения имеет вид:

$$r = \begin{cases} 100, & S = S_t, \\ 0 & S \in S_s, S \neq S_t, \\ -100, & S \in S_d. \end{cases}$$



ПИД-регулятор, построенный на основе метода нечеткой логики с Гауссовой функцией принадлежности показал свою эффективность при управлении шаговым двигателем в работе [62]. Нечеткая структура регулятора позволяет адаптировать его усиление к изменениям в системе. Аналогичный подход использовался в [64] для активного магнитного подшипника, в котором дополнительно проведена многокритериальная оптимизация центров треугольных функций принадлежности и коэффициента усиления с помощью генетического алгоритма. В [63] применяется нечеткая модель Сугэно (Sugeno) нулевого порядка, параметры которой настраиваются в автономном режиме с помощью алгоритма гибридного обучения, который представляет собой комбинацию метода наименьших квадратов и обратного распространения ошибки. Компьютерное моделирование на примере системы стабилизатора напряжения показало, что нечеткая модель хорошо справляется с устойчивостью системы при малых сигналах.

### 3.3. ЗАДАЧИ ИДЕНТИФИКАЦИИ.

В [66] представлен подход к идентификации линейной системы, описанной разностными уравнениями, с использованием алгоритма пчелиного роя. Задача идентификации сформулирована как задача оптимизации, целью которой является получение модели и набора параметров, для которых ошибка прогнозирования между измеренными выходными данными объекта и модели будет минимальной. Результаты работы алгоритма продемонстрированы на модели двигателя постоянного тока. В [67] используется относящийся к эвристическим алгоритм конкуренции колоний (colonial competitive) для настройки параметров ПИД-регулятора системы управления испарителя, при этом минимизируется величина абсолютной квадратичной ошибки слежения системы. Показывается эффективность алгоритма по сравнению с методом Циглера-Никольса (Ziegler-Nichols) и генетическим алгоритмом.

Тенденция развития интеллектуальных методов управ-

ления направлена на комбинацию различных методов. Так, в [68] для идентификации параметров робота используются фильтр Калмана, рекуррентная НС, метод градиентного спуска, а устойчивость синтезируемой системы управления оценивается функцией Ляпунова. Динамика робота описывается уравнениями Эйлера-Лагранжа. Декартовы силы принимаются за выходы системы, а положение  $X$ , скорость  $\dot{X}$  и ускорение  $\ddot{X}$  – за компоненты вектора входного регрессора. Для задачи обратной динамической идентификации рассматриваются три самообучающиеся НС прямого действия с многоуровневой структурой нейронов. Они включают в себя один скрытый слой и достаточное количество нелинейных нейронов и выходной слой с линейной функцией активации. Функция качества  $J$  для управления правилами обновления весов НС представляет собой среднеквадратичное значение четырех ошибок:

$$\begin{aligned}
 J &= e_1^T e_1 + e_5^T e_5, \text{ при } e_2 = e_3 = e_4 = 0, \\
 e_{1i} &= \tau_i - \hat{M}_i \ddot{X} - \hat{C}_i \dot{X} - \hat{G}_i, \quad i \in \{1, 2, \dots, n\}, \\
 e_{2i} &= \hat{M}_{ii} - 2\hat{C}_{ii}, \quad i \in \{1, 2, \dots, n\}, \\
 e_{3m} &= \hat{M}_{ji} - 2\hat{M}_{ij} - 2(\hat{C}_{ji} + \hat{C}_{ij}), \quad i \neq j, \quad m \in \{1, 2, \dots, (n^2 - n)/2\}, \\
 e_{4i} &= |\hat{M} - \lambda I_{n \times n}|, \quad \lambda = \text{eig}(\hat{M})e^{\lambda_0/t}, \quad i \in \{1, 2, \dots, n\}, \\
 e_5 &= X(k) - (T_s \dot{\hat{X}}(k) + \hat{X}(k - 1)), \\
 \hat{X}(k) &= T_s \hat{\dot{X}}(k) + \hat{X}(k - 1), \\
 \hat{\dot{X}}(k) &= \hat{M}^{-1}(k - 1)(\hat{\tau}(k) - \hat{C}(k - 1) - \hat{G}(k - 1)),
 \end{aligned}$$

где  $\hat{M}$ ,  $\hat{C}$ ,  $\hat{G}$  – оценки компонентов матриц инерции, кориолисовой и центробежной, гравитационной составляющих соответственно,  $e_5$  – ошибка прогнозирования,  $\hat{\tau}$  – оценка вектора приложенных сил,  $T_s$  – частота выборок,  $\lambda > 0$  – параметр, обеспечивающий сходимость алгоритма во времени,  $\lambda$  – вектор корней характеристического уравнения системы,  $I_{n \times n}$  – единичная матрица. НС для  $M$ ,  $C$ ,  $G$  настраиваются по следу-

ющему принципу:

$$\begin{aligned}\hat{M}_{ij} &= W_M^2 f(W_M^1 u), \quad u = [X, I_{n \times n}]^T, \\ \hat{M}_i &= [M_{i1}, \dots, M_{ij}, \dots, M_{in}]^T, \quad i, j = \{1, 2, \dots, n\},\end{aligned}$$

где  $W_M^1, W_M^2$  – веса скрытого и выходного слоя НС. Авторы рассматривают два алгоритма обучения: на основе обратного распространения ошибки и дискретной функции Ляпунова. В работе приведено доказательство того, что следующая функция Ляпунова гарантирует глобальную устойчивость системы:

$$(13) \quad \begin{aligned}V(k) &= \sum_i \sum_j \|W_j^i\|_F^2, \\ i &\in \{o, h\}, \quad j \in \{M, C, G\}.\end{aligned}$$

### 3.4. ЗАДАЧИ ОПТИМИЗАЦИИ.

Распространенной задачей оптимизации является поиск какого-либо экстремального значения параметра системы. Так, например, в [69] целью разработки регулятора является стабилизация энергетической системы, которое выражается в обеспечении максимального демпфирования следующим функционалом:  $J = \max(\min(\xi_{ij}))$ , где  $i$  – корни характеристического уравнения системы,  $j$  – рабочие точки. В качестве регулятора энергетической системы используется стандартный регулятор :

$$K(s) = K_p \left( \frac{sT_w}{1 + sT_w} \right) \left( \frac{1 + sT_1}{1 + sT_2} \right) \left( \frac{1 + sT_3}{1 + sT_4} \right),$$

где  $T_1, T_2, T_3, T_4$  – постоянные времена,  $K_p$  – коэффициент. Поиск этих параметров произведен методом популяционного инкрементного обучения, сочетающем в себе алгоритм оптимизации и оценки распределения.

В [70] динамическая модель мягкого робота описана с помощью рекуррентной НС, которая задает траекторию движения робота в виде функции:

$$x_{i+1} = f(x_i, x_{i-1}, \tau_i), \quad i = 1 \dots k, k = t_f/dt,$$

где  $x_i, x_{i-1}, x_{i+1}$  – текущее, предыдущее и следующее положение манипулятора,  $\tau_i$  – управляющие силы электрокабелей,  $f$  – обученное сопоставление НС,  $dt$  – шаг времени. Оптимальная стратегия действий манипулятора получена из решения следующей задачи нелинейной оптимизации:

$$\Pi(t)^* = \min_{\tau} \|x^* - x\|^2 + \sum_k \tau_k^T R \tau_k,$$

при  $0 \leq \tau_k^m \leq \tau_{max}^m$ ,

где  $m$  – число манипуляторов,  $x^*, x$  – желаемая и фактическая координата положения робота. Задача (??) решается численно с помощью градиентного метода последовательного квадратичного программирования (SQP) [71].

Алгоритмы безградиентного поиска получили широкое распространение из-за своей универсальности для любых задач оптимизации, в том числе для НС, так как не используют градиент функции и не предполагают ее дифференцируемости [72]. Их особенность состоит в том, что полученное решение задачи оптимизации является приемлемым, а не оптимальным. В последнее время стали популярными различные биоинспирированные решения, идеи которых заимствованы у природы [73]. К ним относятся эволюционные алгоритмы, алгоритмы роя и колонии [74, 75], популяционные [76] др. Известен также алгоритм летучей мыши, относящийся к роевому интеллекту, основанный на эхолокации. В [77] этот алгоритм используется для идентификации параметров модели синхронного двигателя с постоянными магнитами в режиме реального времени. Для этого минимизируют среднюю квадратичную ошибку наблюдения за вектором состояния системы. Роевой алгоритм кукушки применяется для настройки ПИД-регулятора в системе управления электродвигателем постоянного тока [78] и устройства тиристорно-управляемой продольной компенсации [79]. После сравнения роевого алгоритма с эвристическим, первый оказался эффективнее.

В [80] предложен подход оптимального управления на основе алгоритма опорных векторов для минимизации энерго-

потребления при ходьбе двуногих роботов в условиях неизвестной динамической модели системы и небольших размерах выборки данных. Новый регулятор встроен в оптимальный регулятор и направлен на минимизацию связанной с энергией функции затрат с ограничениями на углы сочленений робота. Функционал энергии выглядит как:

$$J_{EE} = \int_0^T \frac{1}{2} \tau^T \tau dt, \quad \text{при } \tau = g(\Theta),$$

где  $\Theta$  – вектор обобщенных координат,  $g(\cdot)$  параметризована НС. Функция качества в методе опорных векторов в квадратичной форме имеет вид:

$$J_{SVM} = \min \frac{1}{2} w^T w + \frac{1}{2} C \sum_{i=1}^N \xi_i^2, \quad \text{при } \tau_i = w^T \phi(\Theta_i) + \xi_i,$$

где  $w$  – вектор весовых коэффициентов,  $\xi_i$  – положительная переменная,  $N$  – количество обучающих выборок,  $C$  – штрафной коэффициент,  $\phi(\cdot)$  – функция отображения входного пространства в пространство признаков более высокого порядка. Результирующий функционал состоит из суммы  $J_{EE}$  и  $J_{SVM}$ .

В [81] проведена многокритериальная оптимизация параметров ПИД-регулятора по улучшенному алгоритму поиска «учитель-ученик», исключающего попадания значений функции в локальный минимум. Для этого фаза ученика больше не ограничивается двумя учащимися, и сочетается с дополнительной фазой, а также добавлена фаза заблокированного устройства, в которой блокируются параметры близкие к несовместимым целям. Это приводит к тому, что каждая цель не может конфликтовать с другими. В [82] проведен сравнительный анализ между результатами оптимизации параметров ПИД-регулятора системы управления двигателем постоянного тока, полученных с помощью алгоритма пчелиной колонии, метода роя частиц и «учитель-ученик», последний из которых показал лучшие результаты.

### 3.5. ЗАДАЧИ ИТЕРАТИВНОГО ОБУЧЕНИЯ.

Одной из форм искусственного интеллекта является машинное обучение, которое позволяет системе обучаться на основе данных, записанных в памяти, а не путем программирования в явном виде [83]. По результатам обработки алгоритмами наборов обучающих данных создаются более точные модели. Это позволяет обучать модели как до их использования, так и постоянно. Итеративный процесс обучения моделей, в отличие от человека, всегда позволяет улучшать типы связей между элементами данных независимо от их сложности или большого объема. После завершения обучения модели она может использоваться для продолжения обучения уже в режиме реального времени.

Повторяющееся управление изначально было сформулировано в работе [84] для задачи преодоления повторяющихся возмущений с известным периодом, характерных при вращении двигателей, и получил дальнейшее развитие для случаев, когда период возмущений неизвестен или меняется во времени. Задача повторяющегося управления формулируется следующим образом. Рассматривается асимптотически устойчивая система дискретного времени:

$$\begin{aligned} A(z^{-1})y(k) &= z^{-d}B(z^{-1})[u(k) + w(k)], \\ A(z^{-1}) &= 1 + a_1z^{-1} + \dots + a_nz^{-n}, \\ B(z^{-1}) &= b_0 + b_1z^{-1} + \dots + b_mz^{-m}, \end{aligned}$$

где  $u$ ,  $y$ ,  $w$  – вход, выход и возмущение соответственно,  $z^{-1}$  – задержка на один временной шаг,  $d$  – чистое запаздывание. Возмущение  $w$  повторяется с периодом  $N$ :

$$(1 - z^{-N})w(k) = 0.$$

Целью закона управления является асимптотическое стремление ошибки управления к нулю:  $\lim_{k \rightarrow \infty} e(k) = 0$ . Регулятор может быть синтезирован различными способами, наиболее

популярным и изученным из которых является метод на основе принципа внутренней модели, который состоит в построении независимых блоков идентификации параметров гармоник [85]. В [86] представлены некоторые фундаментальные проблемы разработки регуляторов на основе повторяющегося управления (repetitive control) для обработки периодических возмущений, возникающих в электромеханических системах. Преодоление параметрической неопределенности возмущений предлагается с помощью инструментов адаптивного управления,  $\mu$ -синтеза, Q-параметризацию, антивиндап компенсацию.

Тесно связанным с повторяющимся управлением является итеративное обучающее управление [87, 88, 89]. Итеративное обучающее управление – это метод интеллектуального управления отслеживанием для систем, работающих в повторяющемся режиме, применяемый с целью повышения точности систем. Для него выделяется два процесса: повторяющееся выполнение однородных действий и их последовательная корректировка. Примерами таких систем являются роботы в промышленных конвейерных системах, наносистемы, химические периодические процессы, испытательные установки надежности и др. Итерационное обучение в общем виде можно описать как систему из линейных уравнений состояния, одно из которых представляет динамику системы во времени, второе – динамику итерационного обучения [90, 91]:

$$\begin{cases} x(t+1, k) = Ax(t, k) + Bu(t, k), \\ y(t, k) = Cx(t, k), \end{cases}$$

где  $t$  – дискретное время,  $t = 0, 1, \dots, T$ ;  $k$  – номер шага обучения,  $k = 0, 1, \dots$ ;  $u(t, 0) = 0$ ,  $x \in R^n$  – вектор состояния,  $x(0, k) = x_0$ ;  $y \in R^p$  – выходной вектор,  $u \in R^m$  – вектор управления,  $A, B, C$  – матрицы.

Алгоритм итерационного обучения состоит в том, что на каждом шаге измеряемые переменные системы записываются в память программы. На каждом следующем шаге эти переменные обрабатываются для достижения заданного качества

критерия управления. Цель управления состоит в достижении вектором выходных параметров  $y$  заданного значения  $y_r$  с установленной точностью. В связи с чем, закон итеративного обучающего управления с коррекцией на каждом шаге записывается в виде:

$$u(t, k) = u(t, k - 1) + \Delta u(t, k - 1),$$

где  $\Delta u$  – корректирующая добавка к входной переменной системы, выбираемая из условия, что ошибка

$$(14) \quad e(t, k) = y_r(t) - y(t, k),$$

стремится к нулю при  $k \rightarrow \infty$ , при этом  $u_\infty = \lim_{k \rightarrow \infty} u_k(t)$  ограничено по абсолютной величине.

В [92, 93] предложен подход к итеративному обучающему управлению нейропротезом, при котором формируется закон изменения интенсивности электростимуляции в каждый момент времени движения. Угол наклона стопы  $\alpha$  оценивается в режиме реального времени с помощью гироскопов и акселерометров. В начале каждого следующего шага человека определяют рассогласование угла наклона стопы  $e_j$  как поэлементное отклонение между измеренной угловой траекторией  $a_j$  и первыми  $n_j$  отсчетами соответствующей эталонной траектории  $r_\alpha$ :

$$e_j = r_\alpha(1 : n_j) - \alpha_j.$$

В [94] представлен метод отказоустойчивого управления на основе итеративного управления обучением контура управления током для восстановления производительности многофазных приводов с постоянными магнитами в условиях отказа с обрывом цепи. Преимущество такого подхода заключается в том, что не требуется проводить обнаружение и диагностику неисправностей, а достаточно только измерения крутящего момента. Таким образом, итеративное управление обучением дает исчерпывающую информацию о системе и обеспечивает устойчивость к неопределенностям модели. Для движения системы облачных колесных роботов по заданной траектории и одновременной транспортировке груза разработана



гибкая схема управления отслеживанием траектории посредством итеративного управления обучением и проведен анализ устойчивости системы [95]. В [96] рассмотрено управление итеративным обучением под руководством человека для задач слежения за траекторией, когда регулятор получает данные от действий человека-оператора. В [89] описаны новые результаты по разработке динамических итеративных законов управления обучением с учетом влияния шума.

В [5] рассмотрена итеративная процедура планирования процесса фрезерования, для которой необходимо знать технологические параметры станка и геометрические параметры обработки детали для формирования траектории движения инструмента станка. Отклонение инструмента от заданной траектории является существенной проблемой, требующей постоянной проверки и планирования процесса фрезерования. Авторами предложено следующее решение, уменьшающее ошибку обработки до 50%: с помощью методов машинного обучения прогнозируют ошибку между моделью обрабатываемой формы и измерениями реальной поверхности. Для этого используют статистический метод опорных векторов, а набор данных из предыдущих процессов используется в качестве набора обучающих данных. В [97] представлены результаты большого исследования по самооптимизирующимся измерительным инструментам, предназначенного для разработки и проверки инновационного производственного оборудования и систем управления, основанных на понимании, оценке и контроле производственных характеристик крупных станков с помощью самообучающегося регулятора, способного управлять большим количеством измеренных датчиками данных. Предлагаемая архитектура использует метод опорных векторов и аппарат нечеткой логики для сравнения собранных данных и фактическим текущим состоянием процесса.

### 3.6. ЗАДАЧИ ПРОГНОЗИРОВАНИЯ.

К отдельному классу задач относятся задачи прогнозирования состояния системы, для чего требуется обширный

сбор и последующая обработка информации от измерительных датчиков. В [98] предлагается использовать технологию интернета вещей (internet of things, IoT) в производстве, которая позволяет проводить аналитику данных для получения полезной информации. Прогнозирование возможных сбоев в работе продольно-резательной машины выполняется с помощью методологии анализа временных рядов, называемой авторегрессией скользящего среднего. Часто в системах предусмотрены превентивные меры, как например, аварийное отключение нагрузки при возникновении скачков рабочей частоты в энергосистеме. В [99] эта проблема решается с помощью обучения модели прогнозирования по заранее подготовленной базе данных и дальнейшего применения в режиме реального времени. Актуальной проблемой для автоматизированных обрабатывающих станков является износ инструмента. На примере контролирования удаления материала с помощью роботизированной ленточной шлифовальной машины в [100] предложен подход создания модели процесса шлифования, способной отслеживать изменения в рабочих условиях и точно прогнозировать удаление материала. Основная идея состоит в том, чтобы построить исходную модель на основе регрессии опорных векторов на основе данных шлифования в качестве обучающих выборок. Обученная модель модифицируется в соответствии с данными измерений в реальном времени, и на ее основе рассчитываются параметры управления машиной. Прогнозирование геометрической формы оказалось необходимым компонентом управления мягкими роботами. Интеллектуальное моделирование процесса токарной обработки на основе НС, направленное на прогнозирование набора характеристик качества обработки для известных комбинаций параметров резания описано в [101]. Наилучшие результаты показали рекуррентные НС и на основе пчелиной колонии. В [102] по измерениям пьезорезистивных датчиков предложена основа для прогноза трехмерной конфигурации мягкого робота с помощью глубокого обучения. В [103] по-

казано применение машинных методов обучения для оценки состояния забойного насоса и последующей его диагностике. Данные собирались по динамометру, измеряющему нагрузку на насосную штангу, и обрабатывались с помощью специализированного пакета программ для разработки и обеспечения работы в реальном времени систем сбора, обработки информации об объекте мониторинга. Таким образом, выполнение прогноза на основе машинного обучения является современным эффективным способом предотвращения аварийных ситуаций на производстве, снижении стоимости обслуживания и улучшения общего производственного процесса.

Помимо классификации работ по методам применения машинного обучения, существует ряд обзорных статей, написанных по принципу применения методов в какой-либо области техники. В обзорной статье [104] рассматриваются некоторые идеи и методы машинного обучения, которые можно использовать для разработки орбитального и пространственного управления космическими аппаратами. Авторами справедливо замечено, что работ по поиску оптимальной траектории управления очень мало по сравнению с работами, посвященными НС. Публикации разделены по принципу применяемого метода – обучение с учителем и обучение с подкреплением. Также среди обзорных публикаций можно выделить работы, в которых методы машинного обучения применяются к конкретной области техники: управление двуногими роботами [105], электроэнергетическими системами [106], вращающимися электрическими машинами [107], вибрацией конструкций [108], управление автономными транспортными средствами [109], в различных областях механики [110], управление подвеской транспортных средств [111].

#### 4. Заключение.

Присутствующий в мире тренд перехода к высокотехнологичным видам организации производства отражается в научных публикациях по применению интеллектуальных мето-

дов в электро-механических системах. Методы искусственного интеллекта позволяют достичь цели и решить задачи управления мехатронными системами, которые раньше казались невыполнимыми, при этом с большей вычислительной эффективностью и простотой реализации. Сложность задач управления существенно нелинейными, неопределенными или многоагентными системами в условиях воздействия внешней среды требуют индивидуального подхода к решению конкретной задачи, для чего на данный момент уже предложено большое количество инструментов. Эффективность этих алгоритмов для обучения сложных систем устанавливается, как правило, только при проведении практического эксперимента. Развитие самих алгоритмов направлено не только на повышение скорости и точности обучения, но и на все большую независимость от жестко заданной человеком стратегии обучения и ее адаптации к различным целям. Разработчики заимствуют идеи из природы, пытаясь воспроизвести поведение живых организмов в своих алгоритмах. Будущие исследования ставят перед собой задачу, которая в международных публикациях получила название «обучение для обучения» (learning-to-learn) [112], когда агенту приходится самому подбирать стратегию обучения и регулировать метопараметры этой стратегии.

#### Литература

1. Vepa R. Review of techniques for machine learning of real-time control strategies // Intelligent systems engineering. 1993. Vol. 2, no. 2. P. 77–90.
2. Mitchell T. Machine Learning WCB. Boston, MA: McGraw-Hill, 1997.
3. Mosheiov G. Scheduling problems with a learning effect // European Journal of Operational Research. 2001. Vol. 132, no. 3. P. 687–693.

4. Gurel S., Akturk M. Scheduling preventive maintenance on a single CNC machine // *International Journal of Production Research*. 2008. Vol. 46, no. 24. P. 6797–6821.
5. Dittrich M.-A., Uhlich F., Denkena B. Self-optimizing tool path generation for 5-axis machining processes // *CIRP Journal of Manufacturing Science and Technology*. 2019. Vol. 24. P. 49–54.
6. Фомин В.Н., Фрадков А.Л., Якубович В.А. . Адаптивное управление динамическими объектами. Москва: Наука, 1981. С. 447.
7. Annaswamy A. M., Fradkov A. L. A historical perspective of adaptive control and learning // *Annual Reviews in Control*. 2021. Vol. 52. P. 18–41. URL: <https://www.sciencedirect.com/science/article/pii/S1367578821000894>
8. Перельман И.И. Анализ современных методов адаптивного управления с позиций приложения к автоматизации технологических процессов // *Автомат. и телемех.* 1991. С. 3–32.
9. Андриевский Б.Р., Фрадков А.Л. Метод скоростного градиента и его приложения // *Автомат. и телемех.* 2021. С. 3–72.
10. Николенко С., Кадулин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. «Питер», 2018. С. 93–123.
11. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. Москва: Мир, 1976. С. 168.
12. Новак В. Математические принципы нечеткой логики. Москва: Физматлит, 2006. С. 352.
13. Cybenko G. Approximation by superpositions of a sigmoidal function // *Mathematics of control signals and systems*. 1989. Vol. 2, no. 4. P. 303–314.
14. Lee J., Shin J., Realff M. Machine learning: Overview of the recent progresses and implications for the process systems engineering field // *Computers and Chemical Engineering*.

2018. Vol. 114. P. 111–121.
15. Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction. 2nd edition. Cambridge, MA: The MIT Press, 2018. URL: <http://incompleteideas.net/book/the-book-2nd.html>.
  16. Buoniu L., Babuska R., De Schutter B., Ernst D. Reinforcement learning and dynamic programming using function approximators. 2010. P. 1–271.
  17. Kurrek P., Jocas M., Zoghلامي F. et al. AI motion control - A generic approach to develop control policies for robotic manipulation tasks // Proc. International Conference on Engineering Design, ICED. Vol. 2019-August. 2019. P. 3561–3570.
  18. Kim J.-B., Lim H.-K., Kim C.-M. et al. Imitation reinforcement learning-based remote rotary inverted pendulum control in openflow network // IEEE Access. 2019. Vol. 7. P. 36682–36690.
  19. Menghal P., Laxmi A. Real time simulation: A novel approach in engineering education // ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology. Vol. 1. 2011. P. 215–219.
  20. Powell W. Approximate Dynamic Programming: Solving the Curses of Dimensionality. 2nd edition. John Wiley and Sons, 2011.
  21. Bertsekas D., Tsitsiklis J. Neuro-dynamic programming: an overview. Paper presented at the Decision and Control // Proc. 34th IEEE Conference. 1995.
  22. Prokhorov D., Wunsch D. Adaptive critic designs // IEEE Trans. Neural Netw. 1997. Vol. 8, no. 5. P. 997–1007.
  23. Werbos P. J. Reinforcement Learning and Approximate Dynamic Programming (RLADP)—Foundations, Common Misconceptions, and the Challenges Ahead // Reinforcement Learning and Approximate Dynamic Programming for Feedback Control. John Wiley and Sons, Ltd, 2012. P. 1–30. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118453988.ch1>.

- URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118453988>.
24. Lewis F., Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control // *IEEE Circuits Syst. Mag.* 2009. Vol. 9, no. 3.
  25. Puterman M. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 2014.
  26. Sun J., Liu F., Si J. et al. An improved approximate dynamic programming and its application in SVC control // *Dianji yu Kongzhi Xuebao/Electric Machines and Control*. 2011. Vol. 15, no. 5. P. 95–102.
  27. Watkins C. *Learning from Delayed Rewards*. England: University of Cambridge, 1989.
  28. Yoo J. Gaussian process model based reinforcement learning // *Journal of Institute of Control, Robotics and Systems*. 2019. Vol. 25, no. 8. P. 746–751.
  29. Deisenroth M., Rasmussen C., Peters J. Gaussian process dynamic programming // *Neurocomputing*. 2009. Vol. 72, no. 7-9. P. 1508–1524.
  30. Lopes G., Ferreira M., Da Silva Simoes A., Colombini E. Intelligent control of a quadrotor with proximal policy optimization reinforcement learning // *Proceedings - 15th Latin American Robotics Symposium, 6th Brazilian Robotics Symposium and 9th Workshop on Robotics in Education, LARS/SBR/WRE 2018*. 2018. P. 509–514.
  31. Kim J., Hong M., Lee K. et al. Learning to Walk a Tripod Mobile Robot Using Nonlinear Soft Vibration Actuators with Entropy Adaptive Reinforcement Learning // *IEEE Robotics and Automation Letters*. 2020. Vol. 5, no. 2. P. 2317–2324.
  32. Puccetti L., Kopf F., Rathgeber C., Hohmann S. Speed Tracking Control using Online Reinforcement Learning in a Real Car // *2020 6th International Conference on Control, Automation and Robotics, ICCAR 2020*. 2020. P. 392–399.
  33. Poon C.-S. Sensorimotor learning and information processing by Bayesian internal models // *Annual International*

- Conference of the IEEE Engineering in Medicine and Biology - Proceedings. Vol. 26 VI. 2004. P. 4481–4482.
34. Endo Y. Anticipatory robot control for a partially observable environment using episodic memories // Proceedings - IEEE International Conference on Robotics and Automation. 2008. P. 2852–2859.
  35. Sun L., You F. Machine Learning and Data-Driven Techniques for the Control of Smart Power Generation Systems: An Uncertainty Handling Perspective // Engineering. 2021. Vol. 7, no. 9. P. 1239–1247.
  36. Tomin N., Kurbatsky V., Guliyev H. Intelligent control of a wind turbine based on reinforcement learning // 2019 16th Conference on Electrical Machines, Drives and Power Systems, ELMA 2019 - Proceedings. 2019.
  37. Sierra-Garcia J. E., Santos M., Pandit R. Wind turbine pitch reinforcement learning control improved by PID regulator and learning observer // Engineering Applications of Artificial Intelligence. 2022. Vol. 111. P. 104769. URL: <https://www.sciencedirect.com/science/article/pii/S0952197622000598>
  38. Bieker K., Peitz S., Brunton S. et al. Deep model predictive flow control with limited sensor data and online learning // Theoretical and Computational Fluid Dynamics. 2020. Vol. 34, no. 4. P. 577–591.
  39. Gunther J., Pilarski P., Helfrich G. et al. Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning // Mechatronics. 2016. Vol. 34. P. 1–11.
  40. Lin W.-S., Yang P.-C. Adaptive critic motion control design of autonomous wheeled mobile robot by dual heuristic programming // Automatica. 2008. Vol. 44, no. 11. P. 2716–2723.
  41. Haykin S. Neural networks: a comprehensive foundation. 1999 // Mc Millan, New Jersey. 2010. P. 1–24.
  42. Calise A., Rysdyk R. Nonlinear adaptive flight control using



- neural networks // IEEE Control Systems Magazine. 1998. Vol. 18, no. 6. P. 14–25.
43. Kruger T., Mossner M., Axmann J. et al. Adaptive flight control for unmanned aircraft using a stable neural network observer // AIAA Infotech at Aerospace 2010. 2010.
  44. Kruger T., Schnetter P., Placzek R., Vorsmann P. Fault-tolerant nonlinear adaptive flight control using sliding mode online learning // Neural Networks. 2012. Vol. 32. P. 267–274.
  45. Lin F.-J., Hung Y.-C., Chen S.-Y. Field-programmable gate array-based intelligent dynamic sliding-mode control using recurrent wavelet neural network for linear ultrasonic motor // IET Control Theory and Applications. 2010. Vol. 4, no. 9. P. 1511–1532.
  46. Zhu D., Mei T., Luo M. Adaptive sliding mode control for robots based on fuzzy support vector machines // 2009 IEEE International Conference on Mechatronics and Automation, ICMA 2009. 2009. P. 3469–3474.
  47. Wang Y., Sun F., Liu J. et al. Application of Smith predictor based on single neural network in cold rolling shape control // Chinese Journal of Mechanical Engineering (English Edition). 2009. Vol. 22, no. 2. P. 282–286.
  48. Perez J., Cabrera J., Castillo J., Velasco J. Bio-inspired spiking neural network for nonlinear systems control // Neural Networks. 2018. Vol. 104. P. 15–25.
  49. Perez-Pena F., Lenero-Bardallo J., Linares-Barranco A., Chicca E. Towards bioinspired close-loop local motor control: A simulated approach supporting neuromorphic implementations // Proceedings - IEEE International Symposium on Circuits and Systems. 2017.
  50. Ziegler J., Nichlos N. Optimum settings for automatic controllers // Trans. ASME. 1942. Vol. 64. P. 759–768.
  51. Ziegler J., Nichlos N. Process lags in automatic-control circuits // Trans. ASME. 1943. Vol. 65, no. 5. P. 433–443.
  52. Nishikava Y., Sannomiya N., Ohta T. e. a. A method

- for auto-tuning of PID control parameters // *Automatica*. 1984. Vol. 20. P. 321–332.
53. Astrom K., Hagglund T. *Advanced PID Control*. Res. Triangle Park, North Carolina: ISA, 2006.
54. Dong C., Tseng A. A multivariable self-tuning controller for injection molding machines // *Computers in Industry*. 1989. Vol. 13, no. 2. P. 107–122.
55. Keshta H., Ali A., Saied E., Bendary F. Application of Static Var Compensator (SVC) with PI Controller for Grid Integration of Wind Farm Using Harmony Search // *International Journal of Emerging Electric Power Systems*. 2016. Vol. 17, no. 5. P. 555–566.
56. Schillinger M., Hartmann B., Skalecki P. et al. Safe Active Learning and Safe Bayesian Optimization for Tuning a PI-Controller // *IFAC-PapersOnLine*. 2017. Vol. 50, no. 1. P. 5967–5972. 20th IFAC World Congress.
57. Berkenkamp F., Schoellig A., Krause A. Safe controller optimization for quadrotors with Gaussian processes // *Proceedings - IEEE International Conference on Robotics and Automation*. Vol. 2016-June. 2016. P. 491–496.
58. Malikopoulos A., Papalambros P., Assanis D. A learning algorithm for optimal internal combustion engine calibration in real time // *2007 Proc. ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2007*. Vol. 6 PART A. 2008. P. 91–100.
59. Etedali S., Mollayi N. Cuckoo Search-Based Least Squares Support Vector Machine Models for Optimum Tuning of Tuned Mass Dampers // *International Journal of Structural Stability and Dynamics*. 2018. Vol. 18, no. 2.
60. Manocha A., Sharma A. Three axis aircraft autopilot control using genetic algorithms : An experimental study // *2009 IEEE International Advance Computing Conference, IACC 2009*. 2009. P. 171–174.
61. Liang D., Ding W. Modelling and predicting of a switched

- reluctance motor drive using radial basis function network-based adaptive fuzzy system // IET Electric Power Applications. 2009. Vol. 3, no. 3. P. 218–230.
62. Elsodany N., Rezeka S., Maharem N. Adaptive PID control of a stepper motor driving a flexible rotor // Alexandria Engineering Journal. 2011. Vol. 50, no. 2. P. 127–136.
  63. Mitra P., Maulik S., Chowdhury S., Chowdhury S. Anfis based automatic voltage regulator with hybrid learning algorithm // Proc. Universities Power Engineering Conference. 2007. P. 397–401.
  64. Chen H.-C. Optimal fuzzy pid controller design of an active magnetic bearing system based on adaptive genetic algorithms // Proc. 7th International Conference on Machine Learning and Cybernetics, ICMLC. Vol. 4. 2008. P. 2054–2060.
  65. Wen S., Hu X., Lv X. et al. Q-learning trajectory planning based on Takagi–Sugeno fuzzy parallel distributed compensation structure of humanoid manipulator // International Journal of Advanced Robotic Systems. 2019. Vol. 16, no. 1.
  66. Ercin O., Coban R. Identification of linear dynamic systems using the artificial bee colony algorithm // Turkish Journal of Electrical Engineering and Computer Sciences. 2012. Vol. 20, no. SUPPL.1. P. 1175–1188.
  67. Atashpaz Gargari E., Hashemzadeh F., Rajabioun R., Lucas C. Colonial competitive algorithm: A novel approach for PID controller design in MIMO distillation column process // International Journal of Intelligent Computing and Cybernetics. 2008. Vol. 1, no. 3. P. 337–355.
  68. Agand P., Shoorehdeli M. A., Khaki-Sedigh A. Adaptive recurrent neural network with Lyapunov stability learning rules for robot dynamic terms identification // Engineering Applications of Artificial Intelligence. 2017. Vol. 65. P. 1–11. URL: <https://www.sciencedirect.com/science/article/pii/S0952197617301574>
  69. Folly K. Parallel Pbil Applied to Power System Controller

- Design // Journal of Artificial Intelligence and Soft Computing Research. 2013. — 07. Vol. 3.
70. Thuruthel T., Falotico E., Renda F., Laschi C. Learning dynamic models for open loop predictive control of soft robotic manipulators // Bioinspiration and Biomimetics. 2017. Vol. 12, no. 6.
71. Fletcher R. Practical Methods Of Optimization. 1st edition. Chichester: Wiley, 1987.
72. Поляк Б.Т. Введение в оптимизацию. Москва: Наука, 1983. С. 384.
73. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Москва: МГТУ им. Н.Э. Баумана, 2014. С. 446.
74. Balochian S., Baloochian H. Social mimic optimization algorithm and engineering applications // Expert Systems with Applications. 2019. Vol. 134. P. 178–191.
75. Kumar S., Kumar A., Shankar G. Crow search algorithm based optimal dynamic performance control of SVC assisted SMIB system // 2018 20th National Power Systems Conference, NPSC 2018. 2018.
76. Abualigah L. Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications // Neural Computing and Applications. 2020. Vol. 32, no. 16. P. 12381–12401.
77. Rahimi A., Bavafa F., Aghababaei S. et al. The online parameter identification of chaotic behaviour in permanent magnet synchronous motor by Self-Adaptive Learning Bat-inspired algorithm // International Journal of Electrical Power and Energy Systems. 2016. Vol. 78. P. 285–291.
78. Kishnani M., Pareek S., Gupta R. Optimal Tuning of PID controller by Cuckoo Search via Levy flights // 2014 International Conference on Advances in Engineering and Technology Research, ICAETR 2014. 2014.
79. Sethi R., Panda S., Sahoo B. Cuckoo search algorithm based optimal tuning of PID structured TCSC controller //

- Smart Innovation, Systems and Technologies. 2015. Vol. 31. P. 251–263.
80. Wang L., Liu Z., Chen C. et al. Support vector machine based optimal control for minimizing energy consumption of biped walking motions // International Journal of Precision Engineering and Manufacturing. 2012. Vol. 13, no. 11. P. 1975–1981.
81. Xiao L., Zhu Q., Li C. et al. Application of modified teaching-learning algorithm in coordination optimization of TCSC and SVC // Communications in Computer and Information Science. 2014. Vol. 483. P. 44–53.
82. Shouran M., Habil M. Tuning of PID Controller Using Different Optimization Algorithms for Industrial DC Motor // 2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2021. 2021. P. 756–759.
83. Hurwitz J., Kirsch D. Machine Learning for Dummies. Ibm limited edition edition. Hoboken, NJ: John Wiley and Sons, Inc., 2018.
84. Inoue T., Nakano M., Iwai S. High accuracy control of a proton synchrotron magnet power supply // Proc. 8th IFAC world congress, XX. 1981. P. 216–221.
85. Tomizuka M. Zero Phase Error Tracking Algorithm for Digital Control // Journal of Dynamic Systems, Measurement, and Control. 1987. — 03. Vol. 109, no. 1. P. 65–68. [https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/109/1/65/5778350/65\\_1.pdf](https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/109/1/65/5778350/65_1.pdf). URL: <https://doi.org/10.1115/1.3143822>.
86. Tomizuka M. Dealing with periodic disturbances in controls of mechanical systems // Annual Reviews in Control. 2008. Vol. 32, no. 2. P. 193–199.
87. Uchiyama M. Formation of High-Speed Motion Pattern of a Mechanical Arm by Trial // Transactions of the Society of Instrument and Control Engineers. 1978. Vol. 14, no. 6. P. 706–712.

88. Li W., Maier P., Enge H. Self-learning control applied to vibration control of a rotating spindle by piezopusher bearings // Proc. Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering. 2004. Vol. 218, no. 3. P. 185–196.
89. Hladowski L., Galkowski K., Rogers E. Further results on dynamic iterative learning control law design using repetitive process stability theory // 2017 10th International Workshop on Multidimensional (nD) Systems, nDS 2017. 2017.
90. Kurek J., Zaremba M. Iterative Learning Control Synthesis Based on 2-D System Theory // IEEE Transactions On Automatic Control. 1993. Vol. 38, no. 1. P. 121–125.
91. Pakshin P., Emelianova J., Emelianov M. Iterative Learning Control Design for Multiagent Systems Based on 2-D Models // Automation and Remote Control. 2018. Vol. 79, no. 6. P. 1040–1056.
92. Seel T., Werner C., Raisch J., Schauer T. Iterative learning control of a drop foot neuroprosthesis - Generating physiological foot motion in paretic gait by automatic feedback control // Control Engineering Practice. 2016. Vol. 48. P. 87–97.
93. Seel T., Werner C., Schauer T. The adaptive drop foot stimulator – Multivariable learning control of foot pitch and roll motion in paretic gait // Medical Engineering and Physics. 2016. Vol. 38, no. 11. P. 1205–1213.
94. Mohammadpour A., Mishra S., Parsa L. Iterative learning control for fault-tolerance in multi-phase permanent-magnet machines // Proc. American Control Conference. 2013. P. 5929–5934.
95. Li J., Wang S., Wang J. et al. Iterative learning control for a distributed cloud robot with payload delivery // Assembly Automation. 2021. Vol. 41, no. 3. P. 263–273.
96. Warriar R., Devasia S. Iterative Learning from Novice Human Demonstrations for Output Tracking // IEEE Trans-

- actions on Human-Machine Systems. 2016. Vol. 46, no. 4. P. 510–521.
97. Silvestri M., Pedrazzoli P., Boer C., Rovere D. Compensating high precision positioning machine tools by a self learning capable controller // Proc. 11th International Conference of the European Society for Precision Engineering and Nanotechnology, EUSPEN 2011. Vol. 2. 2011. P. 121–124.
  98. Kanawaday A., Sane A. Machine learning for predictive maintenance of industrial machines using IoT sensor data // Proc. IEEE International Conference on Software Engineering and Service Sciences, ICSESS. Vol. 2017-November. 2018. P. 87–90.
  99. Dai Y., Xu Y., Dong Z. et al. Real-time prediction of event-driven load shedding for frequency stability enhancement of power systems // IET Generation, Transmission and Distribution. 2012. Vol. 6, no. 9. P. 914–921.
  100. Yixu S., Hongbo L., Zehong Y. An adaptive modeling method for a robot belt grinding process // IEEE/ASME Transactions on Mechatronics. 2012. Vol. 17, no. 2. P. 309–317.
  101. Shettigar A., Patel G., Chate G. et al. Artificial bee colony, genetic, back propagation and recurrent neural networks for developing intelligent system of turning process // SN Applied Sciences. 2020. Vol. 2, no. 4.
  102. Truby R., Santina C., Rus D. Distributed proprioception of 3d configuration in soft, sensorized robots via deep learning // IEEE Robotics and Automation Letters. 2020. Vol. 5, no. 2. P. 3299–3306.
  103. Sharaf S., Bangert P., Fardan M. et al. Beam pump dynamometer card classification using machine learning // SPE Middle East Oil and Gas Show and Conference, MEOS, Proceedings. Vol. 2019-March. 2019.
  104. Shirobokov M., Trofimov S., Ovchinnikov M. Survey of machine learning techniques in spacecraft control design // Acta Astronautica. 2021. Vol. 186. P. 87–97.

105. Wang S., Chaovalitwongse W., Babuska R. Machine learning algorithms in bipedal robot control // *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*. 2012. Vol. 42, no. 5. P. 728–743.
106. Glavic M. (Deep) Reinforcement learning for electric power system control and related problems: A short review and perspectives // *Annual Reviews in Control*. 2019. Vol. 48. P. 22–35.
107. Dineva A., Mosavi A., Ardabili S. et al. Review of soft computing models in design and control of rotating electrical machines // *Energies*. 2019. Vol. 12, no. 6.
108. Li Z., Adeli H. Control methodologies for vibration control of smart civil and mechanical structures // *Expert Systems*. 2018. Vol. 35, no. 6.
109. Di X., Shi R. A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning // *Transportation Research Part C: Emerging Technologies*. 2021. Vol. 125.
110. Patel A., Ramaiya K., Bhatia C. et al. Artificial Intelligence: Prospect in Mechanical Engineering Field—A Review // *Lecture Notes on Data Engineering and Communications Technologies*. 2021. Vol. 52. P. 267–282.
111. Mozaffari A., Chenouri S., Qin Y., Khajepour A. Learning-based vehicle suspension controller design: A review of the state-of-the-art and future research potentials // *eTransportation*. 2019. Vol. 2.
112. Lansdell B., Kording K. Towards learning-to-learn // *Current Opinion in Behavioral Sciences*. 2019. P. 45–50.



## METHODS OF INTELLIGENT CONTROL OF ROBOTIC AND MECHATRONIC SYSTEMS. A SURVEY

Iuliia Zaitceva, Institute for Problems of Mechanical  
Engineering of RAS, St.Petersburg, Cand. Sc. (St.Petersburg,  
Bolshoi prospect V.O., 61, (812)321-47-78).

Abstract: Artificial intelligence is rapidly conquering more and more areas for application, successfully coping with the tasks set by design engineers. Naturally, complex control tasks require new modern solutions. This review presents not only the latest developments in the control of the mechatronic systems but also shows the connection between the methods of the classical theory of automatic control and machine learning. The reader briefly describes such well-known methods of classical control theory as optimization, adaptation, and fuzzy logic, on the foundation of which artificial neural networks and reinforcement learning are located, and also presents the latest achievements in the application of intelligent control for several topical problems in various fields of technology. The analysis of the literature shows that future research is aimed at an increasing degree of automation and autonomy of processes, and their properties and nature of functioning should approach the human outlines of intelligence.

Keywords: artificial intelligence, machine learning, neural networks, optimization, adaptation.

УДК 681.51  
ББК 3-30

Статья представлена к публикации  
членом редакционной коллегии ...

Поступила в редакцию ...

Дата опубликования ...