

А. Д. Обухов, д-р техн. наук, доц., obuhov.art@gmail.com,
А. О. Назарова, студент, nazarova.al.ol@yandex.ru,
Тамбовский государственный технический университет

Метод управления на основе технологий компьютерного зрения и машинного обучения для адаптивных систем*

Рассматривается проблема организации процесса управления в адаптивных системах, в которых требуется обеспечить сохранение оптимального состояния системы при изменении внешних условий. Анализ существующих подходов к решению данной задачи показал большую перспективность синергетического эффекта от использования технологий машинного обучения и компьютерного зрения. Проведен системный анализ процесса управления с использованием данных технологий, формализованы его основные объекты, поставлена задача исследования. Для ее решения предложен метод, новизна которого заключается в применении технологий машинного обучения и компьютерного зрения для распознавания и получения сжатого представления о состоянии наблюдаемой среды, объектов наблюдения и управления, а также в унификации процесса выбора управляющей команды на основе трех подходов (системы правил, классифицирующей нейронной сети, машинного обучения с подкреплением). Все этапы метода формализованы, возможность использования технологий машинного обучения (нейронных сетей) для их реализации теоретически обоснована. Практическая значимость разработанного метода заключается в возможности автоматизации деятельности человека-оператора в сложных адаптивных системах за счет использования технологий машинного обучения и компьютерного зрения. Метод апробирован на примере системы управления адаптивной беговой платформой. Проведены экспериментальные исследования для оценки работоспособности метода, его производительности и точности работы при определении состояния объектов наблюдения с использованием технологий компьютерного зрения. В результате работы была доказана высокая эффективность предложенного подхода. Использование технологий компьютерного зрения и машинного обучения позволило не только осуществлять управление адаптивной беговой платформой, но и корректно определять критические ситуации (падение и резкую остановку человека), что повышает безопасность работы системы управления, расширяет ее функциональность в области мониторинга состояния окружающей среды и объектов наблюдения.

Ключевые слова: адаптивные системы, процесс управления, компьютерное зрение, машинное обучение, нейронные сети, распознавание объектов

Введение

Разработка адаптивных систем сопряжена с необходимостью обеспечения их корректного функционирования при различных условиях внешней среды. Это требует реализации гибких алгоритмов управления отдельными компонентами, обладающими высокой стабильностью и надежностью. Вопросам обеспечения адаптивности посвящены работы многих исследователей, так как в условиях повсеместной автоматизации производства, промышленности и в повседневной жизни необходимо осуществить замещение человека-оператора на сравнимую по качеству принятия управленческих решений программную систему.

Адаптивность программной системы может быть достигнута посредством различных механизмов: циклов обратной связи, рефлексии, сервис-ориентированной архитектуры, агентов,

машинного обучения [1]. Анализ данной области выявил множество подходов, относящихся как непосредственно к разработке адаптивных систем, так и к классификации неопределенных внешних объектов и воздействий среды, которые способны привести адаптивную систему к сбою. В работах [2, 3] рассматриваются современные архитектурные подходы к решению проблемы неопределенности, их классификация, влияние на качество работы адаптивных систем. В работе [4] внимание сосредоточено на аспектах проектирования самоадаптивных систем, а также проведена систематизация подходов к самоавтоматизации в условиях неопределенности. Понятие самоадаптации автором разбито на семь "волн", и седьмая волна (адаптация на основе опыта) фокусируется на методах машинного обучения как на средстве улучшения реализации адаптирующейся программной системы, так как растущие масштабы и сложность программного обеспечения требует более гибких и интеллектуальных алгоритмов.

Для отслеживания состояния объектов окружающей среды необходимо применение

*Статья подготовлена при финансовой поддержке Министерства науки и высшего образования РФ в рамках гранта Президента РФ МК-857.2022.1.6.

соответствующих технологий, которые должны позволить получить своевременную и достаточную информацию о воздействиях на адаптивную систему. Если система достаточно изолирована от внешней среды и структурно проста, то решение данной задачи возможно за счет включения в среду большого числа датчиков и сенсоров, отслеживающих состояние, положение и характеристики ключевых объектов наблюдения [5]. Однако в целом такие условия невозможны для широкого класса систем: объекты наблюдения могут постоянно меняться, появляться и выходить из зоны функционирования адаптивной системы, сама эта зона может быть непостоянной и подвижной (например, в системах автопилотирования, видеонаблюдения и так далее) [6]. Но даже в относительно изолированных системах, например, при разработке систем захвата движений, существуют определенные сложности при использовании датчиков и трекеров, приводящие к ограничению свободы движения и перемещения человека в пространстве [7]. Сложности возникают и при распознавании нескольких человек одновременно на большой площади, так как системы трекинга в основном ограничены довольно небольшой зоной.

Все вышесказанное приводит к необходимости использования более универсальных подходов, применимых для решения широких классов задач при наличии в области наблюдения произвольного числа объектов различных категорий. Перспективным инструментом для реализации подобных алгоритмов является использование технологий компьютерного зрения и машинного обучения [8]. В настоящее время они уже успешно применяются при реализации модулей управления для различных адаптивных систем: автопилотов [9], систем умного дома с распознаванием лиц и жестов [10], систем поддержки принятия врачебных решений [11], робототехнических комплексов [12], систем дополненной и виртуальной реальности [13].

Таким образом, дальнейшее совершенствование алгоритмов управления в адаптивных системах является актуальной задачей, имеющей важное значение для развития экономики, промышленности и обеспечения безопасности в различных отраслях. Целью исследования является разработка метода управления, позволяющего имитировать действия человека-оператора в адаптивной системе, что позволит достигнуть высоких показателей эффективно-

сти ее работы. Под эффективностью будем понимать некоторое значение оценочной функции среды наблюдения. Для достижения поставленной цели проведен системный анализ процессов в предметной области, после чего формализованы основные этапы предлагаемого подхода и, наконец, проведена его апробация и проверка в ходе экспериментальных исследований.

Системный анализ предметной области

На первом этапе осуществим системный анализ и формализацию основных компонентов предметной области. Пусть задана некоторая среда функционирования адаптивной системы, для которой необходимо реализовать систему управления. Среда может принимать различные состояния $e_t \in E$, где E — множество всех состояний среды. Под состоянием среды понимается некоторая конечная последовательность значений параметров среды, однозначно определяющая ее ключевые характеристики.

Определение 1. Под *состоянием среды* будем понимать графическую информацию, полученную из кадра $f_t \in F$ в некоторый момент времени t из множества всех отслеживаемых моментов T (или из некоторой последовательности кадров $F_a = \{f_t | t_1 \leq t \leq t_2\}$, $F_a \subseteq F$):

$$e_t \rightarrow \{f_t\}, t \in T. \quad (1)$$

Кадру соответствует кортеж его характеристик:

$$f_t \rightarrow \langle pix_t, sz_x, sz_y \rangle, \quad (2)$$

где pix_t — множество значений пикселей кадра; sz_x, sz_y — размер кадра по оси OX и OY , таким образом, кадр задается трехмерным массивом значений пикселей размером $sz_x \times sz_y \times ch$, где ch — число цветовых каналов.

Каждому состоянию среды также соответствует некоторое множество объектов, которые присутствуют на кадре (объекты наблюдения X) либо которые воздействуют на среду (объекты управления O). Получим:

$$e_t \rightarrow \langle X_t, O_t \rangle, X_t \subseteq X, O_t \subseteq O. \quad (3)$$

Определение 2. Под *объектом наблюдения* $x_i \in X$ понимается распознанная некоторым методом MR сущность v_i среды e_t , обладающая

определенными свойствами и характеристиками, поведением и функциональностью:

$$MR : f_t \rightarrow \{v_i : (\forall i)(x_i \in X_t)\}. \quad (4)$$

Каждый объект наблюдения определяется кортежем:

$$x_i = \langle s_j, k_n, v_i \rangle, \quad (5)$$

где s_j — состояние объекта наблюдения x_i , включающее некоторую конечную последовательность значений параметров, характеризующих объект: $s_j \rightarrow \langle s_{j,1}, s_{j,2}, \dots, s_{j,m} \rangle$; k_n — категория (тип) объекта наблюдения x_i ; v_i — визуальное представление объекта наблюдения x_i в кадре f_t .

В соответствии с работами [14–16] в настоящее время предпочтительнее с точки зрения производительности и точности работы использование в качестве исходных данных не всего изображения f_y , а некоторого его сжатого представления z_t , обладающего намного меньшей размерностью: $|z_t| \ll |sz_x \times sz_y \times ch|$. Такое представление содержит основные признаки среды, но его использование накладывает ограничения на возможность осуществления отображения (4). Поэтому для того, чтобы, с одной стороны, сохранить возможность извлечения объектов наблюдения, а с другой, значительно уменьшить размерность исходных данных, перейдем от представления (5) к следующей форме представления объектов наблюдения:

$$x_i = \langle s_j, k_n, zv_i \rangle, \quad (6)$$

$$NN_{AE}(v_i) = v_i \xrightarrow{\text{Encoder}} zv_i \xrightarrow{\text{Decoder}} v_i,$$

где $NN_{AE}(v_i)$ — автоэнкодер, принимающий на вход и выходящий графическую информацию об объекте наблюдения и формирующий сжатое представление признаков zv_i , обладающее намного меньшей размерностью $|zv_i| \ll |v_i|$.

Определение 3. Под объектом управления (агентом) $o_m \in O$ будем понимать сущность, которая осуществляет взаимодействие со средой, объектами наблюдения и может быть формализована в форме объекта наблюдения, если он присутствует в рамках среды как распознанная сущность, либо в форме без графического представления, если он не принадлежит кадру.

Формализуем представление объекта управления как выбор из двух возможных форм:

$$o_m = \begin{cases} \langle s_j, k_n, v_m \rangle, & \text{если } (\exists MR(f_t) \wedge \exists o_m \in O_t); \\ \langle s_j, k_n \rangle, & \text{в противном случае.} \end{cases} \quad (7)$$

Таким образом, можно сказать, что в рассматриваемой среде задана пара (марковская цепь) $\langle S, P \rangle$, где S — множество состояний объектов среды, а P — множество вероятностей переходов:

$$P = \{p(s_{t+1} | s_t) : (t \in \{0, 1, \dots\}), (s_t, s_{t+1} \in S)\}.$$

Данные переходы существуют и для объектов наблюдения, и для объектов управления. Для осуществления оптимального управления объектом $o_m \in O$ необходимо выбрать из множества управляющих воздействий (действий) A те, что позволяют достигнуть желаемого результата. Чтобы формализовать достижимость результата для объекта управления, необходимо задать функцию оценки состояния среды (награды) $Q : S \times A \rightarrow \mathbb{R}$, ставящую в соответствие текущим состояниям объектов некоторое численное значение, принимающее максимальные значения при их оптимальном состоянии и минимальные — в противном случае. Расчетная формула функции зависит от конкретной предметной области и ключевых свойств объектов среды.

Обозначим MC — метод управления, отвечающий за формирование необходимых управляющих воздействий для компенсации возможных негативных факторов и сохранения максимальных значений целевой функции наград Q . Метод управления в простейшей форме можно представить как множество различных траекторий изменения состояний среды под воздействием множества действий:

$$MC : S_t \rightarrow (\langle S_t, a_t \rangle, \langle S_{t+1}, a_{t+1} \rangle, \langle S_{t+2}, a_{t+2} \rangle, \dots). \quad (8)$$

Иными словами, метод управления должен формировать такую траекторию изменения состояния среды, при которой функция наград будет стремиться к максимуму. В рассматриваемой предметной области, таким образом, реализуется марковский процесс принятия решений, который можно представить в виде четверки $\langle S, A, P, Q \rangle$. Получим схему управления, показанную на рис. 1.

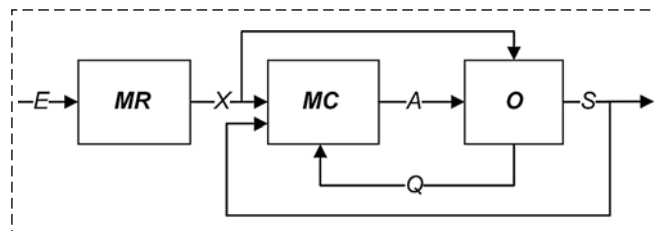


Рис. 1. Схема управления адаптивной системой
Fig. 1. Adaptive system control scheme

На основе проведенного анализа осуществим постановку задачи исследования в формализованной форме.

Формализованная постановка задачи исследования

Необходимо для текущего состояния среды $e_t \in E$, включающей множество объектов наблюдения X_t и управления O_t , каждому из которых соответствует некоторая категория $k_n \in K$ и состояние $s_j \in S$, на основе метода *МС* определить управляющее воздействие $a_k \in A$, при котором целевая функция оценки среды достигает экстремума $Q \rightarrow \max$.

Для решения поставленной задачи предлагается метод управления *МС*, основанный на технологиях машинного обучения и компьютерного зрения и позволяющий автоматизировать следующие процессы: поиск и распознавание категорий объектов ($X \rightarrow K$); оценку состояний объектов ($X \rightarrow S$, $O \rightarrow S$) и определение множества векторов признаков ($X \rightarrow ZV$); выявление свойств среды e_t (нахождение вектора признаков $E \rightarrow Z$); выбор управляющего воздействия $a_k \in A$; оценку состояния среды Q .

Формализация метода управления на основе технологий компьютерного зрения и машинного обучения

Рассмотрим формализованное представление основных этапов предлагаемого метода.

1. Получение информации в момент времени $t \in T$ о среде e_t путем извлечения кадра f_t или некоторого множества кадров ($\{f_t\}$).

2. Реализация процедуры распознавания *MR* объектов наблюдения. Процедура распознавания *MR* основана на технологиях компьютерного зрения и нейронных сетях, что позволяет сегментировать кадр на области, соответствующие объектам заданных категорий. Данная процедура может быть реализована как за счет уже существующей предварительно обученной нейронной сети, так и путем обучения новой сети на собранном наборе данных. Для того, чтобы определить все составляющие процедуры распознавания *MR* для каждого объекта наблюдения, выполним следующие шаги:

2.1. Определение визуального представления для каждого объекта наблюдения $x_i \in X_t$ (сегментирование кадра):

$$NN_r : f_t \rightarrow v_i; v_i = \{h_{ij}\}, \quad (9)$$

где $\{h_{ij}\}$ — множество ключевых точек объекта наблюдения, распознанных нейронной сетью распознавания NN_r , в том числе весь набор пикселей, принадлежащих объекту x_i .

2.2. Получение сжатого представления о среде и объектах наблюдения:

$$NN_{AE}(e_t) \rightarrow z_t; \quad (10)$$

$$NN_{AE}(v_i) \rightarrow zv_i. \quad (11)$$

2.3. Определение категории объектов с применением нейронных сетей или иных технологий классификации NN_k :

$$NN_k : zv_i \rightarrow k_n. \quad (12)$$

2.4. Определение состояния объекта с помощью алгоритма классификации или нейронной сети NN_s :

$$NN_s : \langle zv_i, k_n \rangle \rightarrow s_j. \quad (13)$$

2.5. Оценка общего состояния среды как совокупности состояний объекта, существующих в ней в момент времени $t \in T$:

$$S_t = \{s_j : (\forall i)(\langle x_i, s_j \rangle \in e_t)\}. \quad (14)$$

3. Выбор управляющей команды. Процедура выбора управляющей команды может осуществляться по трем различным направлениям, в результате чего определяется метод управления *МС*.

3.1. Управление на основе правил относительно объекта управления. Введем систему продукционных правил следующих видов:

— относительно состояний объектов:

$$\begin{aligned} &\text{ЕСЛИ } \{ \langle \text{состояние объекта управления} \\ &\quad o_m = s_a \rangle \} \\ &\{ \text{И/ИЛИ } \langle \text{состояние объекта наблюдения} \\ &\quad x_i = s_b \rangle \} \\ &\text{ТО } \langle \text{выполнить действие } a_k \rangle; \quad (15) \end{aligned}$$

— относительно состояния всей среды:

$$\begin{aligned} &\text{ЕСЛИ } \{ \langle \text{состояние среды } e_t = S_t \rangle \} \\ &\text{ТО } \langle \text{выполнить действие } a_k \rangle. \quad (16) \end{aligned}$$

Данный подход требует привлечения экспертов для формирования системы правил, однако в ряде технических систем такой подход обеспечивает наибольшую прозрачность и

надежность работы системы управления. Недостатком подхода является некорректная работа системы в случае появления состояний, не попадающих ни под одно из правил.

3.2. Управление на основе классифицирующей нейронной сети для выбора оптимальной команды:

$$NN_c : \{s_j : (s_j \in S_t)\} \rightarrow a_k. \quad (17)$$

Для реализации данного подхода требуется большая выборка размеченных данных для обучения нейронной сети, охватывающих все возможные состояния объектов наблюдения с соответствующими им действиями системы управления. Недостаточная точность нейронной сети может привести к выбору ошибочного действия, а ее переобучение — к некорректной работе на неизвестных сочетаниях состояний объектов.

3.3. Управление на основе максимизации функции оценки и машинного обучения с подкреплением основано на использовании Q-обучения. Данный подход основан на анализе вознаграждения R системы управления вплоть до окончания работы в момент времени w :

$$R = \sum_{t=0}^w r_t. \quad (18)$$

Система управления максимизирует вознаграждение в каждый момент времени t , однако на практике будущие вознаграждения учитываются с поправочными уменьшающими коэффициентами $\gamma < 1$. В теории Q-обучения для определения полного дисконтированного будущего вознаграждения, начиная от текущего момента времени t и до конца работы системы w , используется следующая формула:

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots + \gamma^{w-t} r_w = r_t + \gamma R_{t+1}. \quad (19)$$

Для выбора оптимальной команды необходимо максимизировать значение Q-функции, которая определяется как максимальное дисконтированное будущее вознаграждение при выполнении команды a_k в состоянии S_i :

$$Q(S_i, a_k) = \max(R_{t+1}). \quad (20)$$

Для реализации метода управления на основе Q-обучения необходимо реализовать нейронную сеть, аппроксимирующую выражение (20):

$$NN_q : \langle S, A \rangle \rightarrow R. \quad (21)$$

Для обучения сети NN_q необходимо собрать большой объем данных о среде и поведении объектов в ней либо в реальной системе, либо в ее имитации (абстрактной модели). В процессе функционирования нейронная сеть NN_q для всех возможных состояний S_t будет выбирать действие из множества A таким образом, чтобы в результате его выполнения достигалась наибольшая награда R_{t+1} .

4. Выполнение выбранной управляющей команды a_k . Выбор команды a_k в момент времени $t \in T$ приводит к изменению среды:

$$a_k : S_t \rightarrow S_{t+1}, X_t \rightarrow X_{t+1}, O_t \rightarrow O_{t+1}. \quad (22)$$

5. Определение обратной связи FB — разницы между текущим и следующим состоянием среды:

$$FB_t = \left\langle \begin{array}{l} \Delta S_t = \{s_j : (\forall j)(s_j \notin S_t, s_j \in S_{t+1})\}, \\ \Delta X_t = \{x_i : (\forall i)(x_i \notin X_t, x_i \in X_{t+1})\}, \\ \Delta O_t = \{o_m : (\forall m)(o_m \notin O_t, o_m \in O_{t+1})\} \end{array} \right\rangle. \quad (23)$$

6. Оценка состояния системы. Если используется Q-обучение, то для оценки можно использовать формулу (19). Иначе можно адаптировать данную формулу и привести ее к общему виду функции оценки, которая зависит от нового состояния среды и полученной обратной связи и накапливается с течением времени:

$$Q(S_t, FB_t) = \sum_{i=0}^t q_i. \quad (24)$$

Таким образом, необходимо максимизировать значения функции (24) путем выбора команд, обеспечивающих наибольшее значение q_i . Значение q_i может быть задано на основе оценки характеристик объектов наблюдения, среды или использования иных расчетных формул.

Для успешного применения предложенного метода для решения задач анализа и обработки информации рассмотрим обоснование возможности осуществления отображений (9)—(13), (17), (21). Введем следующие обозначения:

G_i — i -е отображение из набора (9)—(13), (17), (21);

X_i — входные данные отображения G_i , имеющие форму $NX_i = \{nx_{ij}\}$, где $|NX_i|$ — число измерений формата входных данных, nx_{ij} — размер данных по j -му измерению;

Y_i — выходные данные отображения G_i , имеющие форму $NY_i = \{ny_{ij}\}$, где $|NY_i|$ — число измерений формата выходных данных, ny_{ij} — размер данных по j -му измерению;

NN_i — нейронная сеть, реализующая отображение G_i .

Проанализируем формат входных и выходных данных для каждого отображения G_i :

1) X_1 — множество пикселей (кадр), $|NX_1| = 3$, Y_1 — множество координат пикселей, $|NY_1| = 2$;

2) X_2 — множество пикселей (кадр или их последовательность), $|NX_2| \in \{3,4\}$, Y_2 — вектор, $|NY_2| = 1$;

3) X_3 — множество координат пикселей, $|NX_3| = 3$, Y_3 — вектор, $|NY_3| = 1$;

4) X_4 — вектор, $|NX_4| = 1$, Y_4 — категория (целое число), $|NY_4| = 1$;

5) X_5 — кортеж из вектора и категории (целого числа), $|NX_5| = 2$, Y_5 — состояние (целое число), $|NY_5| = 1$;

6) X_6 — множество состояний, вектор, $|NX_6| = 1$, Y_6 — команда (целое число), $|NY_6| = 1$;

7) X_7 — кортеж из множества состояний и действий, $|NX_7| = 1$, Y_7 — вектор значений вознаграждения для каждого действия, $|NY_7| = 1$.

Для обоснования возможности выполнения всех отображений G_i с применением нейронных сетей NN_i сформулируем следующую теорему.

Теорема 1. *Каждому $|NX_i|$ -мерному вектору входных данных X_i , заданному в отображении G_i , можно поставить в соответствие $|NY_i|$ -мерный вектор выходных данных Y_i , полученный в ходе преобразования X_i нейронной сетью NN_i , с отклонением от $|NY_i|$ -мерного вектора требуемых значений Y_i^* не более, чем на ε при условии, что отображение $X_i \rightarrow Y_i$ непрерывно.*

Доказательство. На первом этапе убедимся, что отображение $X_i \rightarrow Y_i$ удовлетворяет данным условиям. Изложенный выше анализ X_i и Y_i показывает, что каждое отображение G_i является непрерывным: кадру или набору пикселей всегда можно поставить в соответствие некоторый вектор (для отображений G_1, G_2, G_3), вектору — значение категории (для G_4, G_5, G_6) или вознаграждения (для G_7). Предположим, что это не так. Тогда получим, что для отображений G_1, G_2, G_3 невозможно выполнить преобразование пикселей таким образом, чтобы получить некоторый вектор. Однако принцип работы сверточных нейронных сетей и алгоритмов сжатия изображений позволяет осуществить такую свертку или сжатие для любых изображений. Если для отображений G_4, G_5, G_6 входному вектору x_{ij} нельзя поставить ни одну категорию $y_{ij} \in Y_i$, то множество Y_i может быть расширено дополнительной категорией y_{uj} , соответствующей x_{ij} ("неизвестное" состо-

яние, "пустая" команда и так далее). Наконец, для любого сочетания состояний и команд отображения G_7 может быть задана награда, в том числе нулевая, если такое сочетание не присутствовало в исходной выборке. Если $X_i \rightarrow Y_i$ задано табличными значениями, оно может быть заменено сплайном некоторой степени, что обеспечивает непрерывность каждого G_i .

На основе доказательства теорем Хехт—Нильсена и Колмогорова—Арнольда [17] возможно аппроксимировать многомерную функцию $X_i \rightarrow Y_i$ с помощью некоторой нейронной сети (минимум трехслойной) с конечным числом элементов для любой размерности векторов $|NX_i|$ и $|NY_i|$. Теорема Цыбенко гласит, что при условии непрерывности отображения G_i его можно аппроксимировать нейронной сетью с ошибкой, не превышающей ε . Таким образом, мы можем успешно применять нейронные сети NN_i для каждого отображения G_i .

Практическое применение метода при реализации системы управления беговой платформой

Рассмотрим пример апробации разработанного метода при реализации системы управления беговой платформой. Общий вид экспериментальной установки представлен на рис. 2. Используемая в эксперименте беговая платформа обладает следующими характеристиками: длина полотна = 1,5 м, ширина полотна = 0,9 м, диапазон регулировки скорости полотна = [0,01; 3,5] м/с [18]. Для реализации системы управления использовали камеру с максимальным разрешением кадра 1920×1080 пикселей и записью до 60 кадров в секунду. Необходимо организовать регулирование скорости адаптивной беговой платформы с использованием технологий машинного обучения и компьютерного зрения на основе разработанного метода.

На первом этапе соберем последовательность кадров с камеры, используя библиотеку *OpenCV*. В результате получена информация о движении человека в течение 17 026 кадров, в процессе которых наблюдаемый объект стоял, двигался, ходил и падал. Обозначим состояния объектов наблюдения следующим образом: для человека x_1 — остановка ($s_{1,1}$), движение ($s_{1,2}$), падение ($s_{1,3}$); для беговой платформы x_2 — размеры дорожки ($s_{2,1}$), относительное положение человека на дорожке в диапазоне от 0 до 1 ($s_{2,2}$), скорость беговой дорожки ($s_{2,3}$).



Рис. 2. Распознанные объекты: беговая платформа и человек на ней
 Fig. 2. Recognized objects: the running platform and the person on it

Для распознавания объектов используем два инструмента. Так как беговая платформа и камера зафиксированы относительно друг друга, то для распознавания платформы на кадре можно использовать класс *TrackerKCF* из набора *OpenCV-Tracker*, позволяющий отслеживать с некоторой точностью фрагмент изображения с высокой производительностью. Для распознавания человека используем предварительно обученную нейронную сеть *MediaPipe Pose*, отслеживающую 33 ключевые точки, а также позволяющую сегментировать человека на кадре. Результат второго этапа (сегментирование основных объектов наблюдения) представлен на

рис. 2. Таким образом, для двух основных объектов посредством технологий компьютерного зрения и нейронных сетей получены их сжатые представления — 33 точки для человека и 4 точки для дорожки (записаны в состоянии $s_{2,1}$), что значительно сокращает объем обрабатываемых данных.

У каждого из объектов наблюдения задана только одна категория, поэтому необходимости в проведении этапа 2.3 нет. Далее в соответствии с этапом 2.4 осуществим классификацию состояния человека на три состояния: остановка ($s_{1,1}$), движение ($s_{1,2}$), падение ($s_{1,3}$). Множество записанных кадров разделено на три группы, соответствующие этим состояниям: 7347 кадров для состояния $s_{1,1}$, 7273 кадров — для $s_{1,2}$, 2406 — для $s_{1,3}$. Для решения задачи классификации обучена нейронная сеть NN_s из трех слоев, на вход передается вектор из пяти записей по 33 точки с двумя координатами (последовательность поз человека), на выход — метка состояния. В качестве функции потерь используется перекрестная энтропия, оптимизатор — *Adam*. После обучения итоговая точность классификации состояния составила 88 %, на тестовых данных — до 71 %. Процесс обучения отражен на рис. 3.

Результат работы нейронной сети представлен на рис. 4 (см. вторую сторону обложки). Нейронная сеть успешно фиксирует в ходе анализа последовательности из пяти кадров состояние человека и может использоваться для получения соответствующего управляющего воздействия. Ее можно применять для фиксации состояния человека (его резкой остановки или падения), что дополнительно повышает безопасность использования беговой платформы. Данный момент особенно актуален при

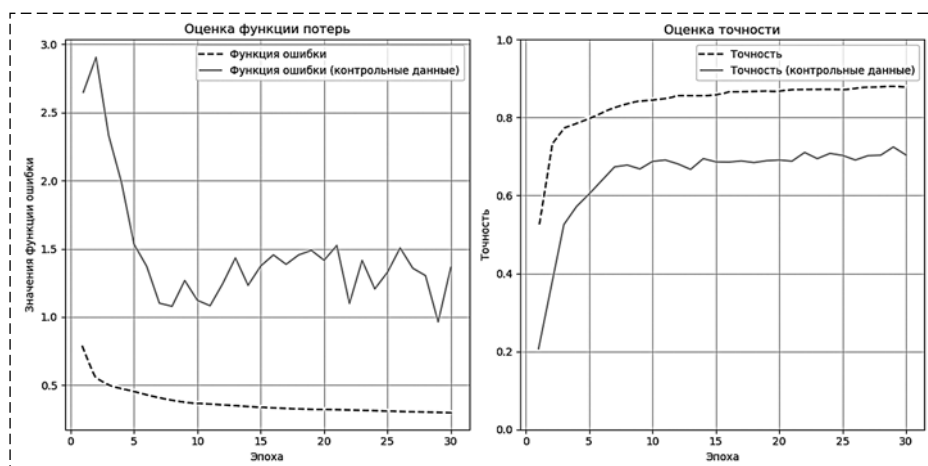


Рис. 3. Процесс обучения нейронной сети классификации поз человека
 Fig. 3. The process of training a neural network for the classification of human poses

применении технологий компьютерного зрения несмотря на то, что в процессе взаимодействия с беговой дорожкой человек находится в страховке.

Оценим скорость распознавания состояния человека. Камера захватывает кадры с частотой 60 кадров в секунду, для распознавания позы используется последовательность из пяти кадров. Таким образом, время распознавания составляет минимум 83 мс плюс время обработки

кадра (30 мс на кадр). Поза может быть распознана через 233 мс или ранее (так как нейронная сеть может обработать позу раньше, чем зафиксирует пять кадров), что является адекватной скоростью реакции системы.

Под состоянием объекта "дорожка" $s_{2,2}$ в данном эксперименте будем понимать положение человека относительно дорожки в диапазоне от 0 до 1, где нулю соответствует левый край полотна, а 1 — правый. Определение этого состояния можно осуществить путем вычисления положения центральной точки силуэта человека относительно платформы. После завершения этапа 2 будет выполнена обработка информации для распознавания, классификации и позиционирования объектов в кадре. Если человека находится в состоянии $s_{1,1}$ (остановка) или $s_{1,3}$ (падение), то состояние дорожки игнорируется. Результат определения позиции человека на дорожке представлен на рис. 5 (см. вторую сторону обложки).

Далее в соответствии с этапом 3 необходимо выбрать управляющую команду на основе состояния объектов. При этом ввиду небольшого числа объектов воспользуемся подходом, изложенным на этапе 3.1. Пусть задано три класса команд: "стоп", "ускорение", "замедление". Введем правила:

- если объект пользователя находится в состоянии $s_{1,2}$, и состояние платформы $s_{2,2} > 0,5$, то выполнить действие "ускорение";
- если объект пользователя находится в состоянии $s_{1,2}$, и состояние платформы $s_{2,2} < 0,5$, то выполнить действие "замедление";
- если объект пользователя находится в состоянии $s_{1,1}$ или $s_{1,3}$, то выполнить действие "стоп".

В качестве обратной связи системы управления используется текущая скорость платформы $s_{2,3}$, которая изменяется на следующей итерации в зависимости от управляющего воздействия. В качестве функции оценки среды используем текущее отклонение человека относительно центральной точки (положение 0,5 на платформе). Будем использовать следующую формулу:

$$Q = 1 - \frac{\sum_{t=0}^T |s_{2,3}(t) - 0,5|}{T}, \quad (25)$$

где $s_{2,3}(t)$ — состояние дорожки, характеризующее относительное положение человека, в момент времени t . График изменения положения человека в ходе испытания представлен на рис. 6.

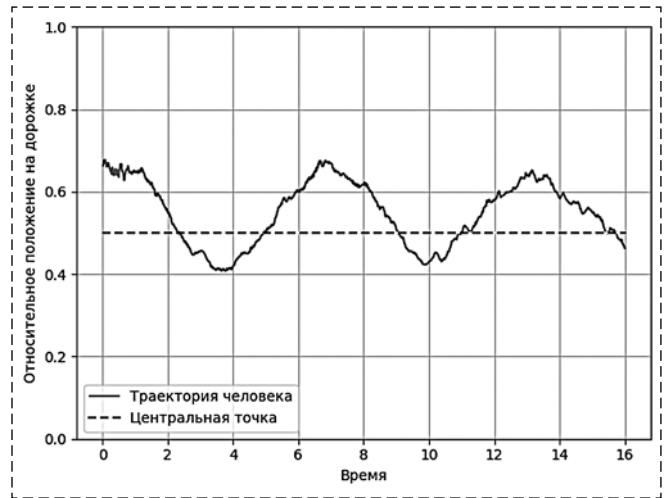


Рис. 6. Пример определения позиции человека

Fig. 6. The example of determining the position of a person

При движении в течение 16 с получено значение функции наград, равное 0,92, таким образом, отклонение от эталонной траектории составило всего 8 %, что можно считать допустимым результатом, сравнимым со значениями, полученными при участии человека-оператора (0,85...0,95). Таким образом, разработанный метод был успешно апробирован в программном обеспечении системы управления беговой платформы.

Заключение

В ходе анализа предметной области адаптивных систем выявлена необходимость в разработке обобщенного подхода к организации процесса управления с применением технологий компьютерного зрения. В результате системного анализа формализованы ключевые составляющие процесса управления и поставлена задача исследования: необходимо определить набор управляющих воздействий для каждого состояния среды наблюдения на основе выбранной целевой функции оценки. Для решения поставленной задачи предложен метод, основанный на технологиях машинного обучения и компьютерного зрения, формализованы его этапы, представлено теоретическое обоснование его реализации. Новизна метода заключается в применении технологий машинного обучения и компьютерного зрения для распознавания и получения сжатого представления о состоянии наблюдаемой среды, объектов наблюдения и управления, унификации процесса выбора управляющей команды на основе трех подходов (система правил, клас-

сифицирующая нейронная сеть, машинное обучение с подкреплением). Представленный метод позволяет автоматизировать деятельность человека-оператора в адаптивных системах.

Метод апробирован при реализации системы управления адаптивной беговой платформой, проведена оценка его точности, быстродействия и эффективности. Полученные практические результаты подтвердили работоспособность метода. Применение технологий компьютерного зрения в рамках метода позволило успешно распознавать с высокой точностью (до 88 %) текущую позу человека, что может быть использовано для оценки его состояния и предотвращения опасных для здоровья пользователя ситуаций.

Проведенные экспериментальные исследования позволили также выявить ограничения используемых технологий компьютерного зрения: необходимость разделения программного кода на отдельные потоки выполнения для обеспечения высокого уровня производительности; высокая вероятность искажения объекта наблюдения в кадре при его быстром перемещении, что затрудняет работу нейронных сетей; определенные требования к оборудованию (разрешение и частота кадров видеопотока). Решением данных проблем может быть использование высокочастотных камер (от 60 и более кадров в секунду) с большим разрешением кадра (от 1920×1080 пикселей), обеспечивающих более низкую вероятность получения смазанных и нечетких изображений. Таким образом, при использовании технологий компьютерного зрения необходимо учитывать сильные и слабые стороны данного инструмента, качество видеопотока, так как эффективность работы алгоритмов в этой области напрямую зависит от исходных данных.

Список литературы

1. Raibulet C., Arcelli Fontana F., Caretoni S. A preliminary analysis of self-adaptive systems according to different issues // *Software Quality Journal*. 2020. Vol. 28, N. 3. P. 1213–1243.

2. Weyns D., Bencomo N., Calinescu R., Camara J., Ghezzi C., Grassi V., Tamburrelli G. Perpetual assurances for self-adaptive systems, *Software Engineering for Self-Adaptive Systems III*, Assurances. Springer, Cham. 2017. P. 31–63.

3. Mahdavi-Hezavehi S., Avgeriou P., Weyns D. A classification framework of uncertainty in architecture-based self-adaptive systems with multiple quality requirements // *Managing Trade-Offs in Adaptable Software Architectures*. Morgan Kaufmann, 2017. P. 45–77.

4. Weyns D. *An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective*. John Wiley & Sons. 2020.

5. Casadei R., Fortino G., Pianini D., Russo W., Savaglio C., Viroli M. Modelling and simulation of opportunistic IoT services with aggregate computing // *Future Generation Computer Systems*. 2019. Vol. 91. P. 252–262.

6. Chen Y., Li W., Sakaridis C., Dai D., Van Gool L. Domain adaptive faster r-cnn for object detection in the wild // *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. P. 3339–3348.

7. Colyer S. L., Evans M., Cosker D. P., Salo A. I. A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system // *Sports medicine-open*. 2018. Vol. 4, N. 1. P. 1–15.

8. Lyons N. Deep Learning-based Computer Vision Algorithms, Immersive Analytics and Simulation Software, and Virtual Reality Modeling Tools in Digital Twin-driven Smart Manufacturing // *Economics, Management, and Financial Markets*. 2022. Vol. 17, N. 2. P. 67–81.

9. Ge M., Zhang Y. Visual Autopilot Decision System Based on Deep Learning, 3D Imaging Technologies—Multi-dimensional Signal Processing and Deep Learning // Springer, Singapore. 2021. P. 329–335.

10. Oudah M., Al-Naji A., Chahl J. Hand gesture recognition based on computer vision: a review of techniques // *Journal of Imaging*. 2020. Vol. 6, N. 8. P. 73.

11. González Izard S., Sánchez Torres R., Alonso Plaza Ó., Juanes Méndez J. A., García-Peñalvo F. J. Nextmed: automatic imaging segmentation, 3D reconstruction, and 3D model visualization platform using augmented and virtual reality // *Sensors*. 2020. Vol. 20, N. 10. P. 2962.

12. Kazemian A., Yuan X., Daytalab O., Khoshnevis B. Computer vision for real-time extrusion quality monitoring and control in robotic construction // *Automation in Construction*. 2019. Vol. 101. P. 92–98.

13. Yang L. L., Huang J., Feng T. I. A. N., Hong-An W. A. N. G., Guo-Zhong D. A. I. Gesture interaction in virtual reality // *Virtual Reality & Intelligent Hardware*. 2019. Vol. 1, N. 1. P. 84–112.

14. Cheng Z., Sun H., Takeuchi M., Katto J. Deep convolutional autoencoder-based lossy image compression // *2018 Picture Coding Symposium (PCS)*. IEEE. 2018. P. 253–257.

15. Zhou L., Cai C., Gao Y., Su S., Wu J. Variational autoencoder for low bit-rate image compression // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018. P. 2617–2620.

16. Tsimpouris E., Tsakiridis N. L., Theocharis J. B. Using autoencoders to compress soil VNIR—SWIR spectra for more robust prediction of soil properties // *Geoderma*. 2021. Vol. 393. P. 114967.

17. Бутырский Е. Ю., Кувалдин И. А., Чалкин В. П. Аппроксимация многомерных функций // *Научное приборостроение*. 2010. Т. 20, № 2. С. 82–92.

18. Krasnyanskiy M. N., Obukhov A. D., Dedov D. L. Control System for an Adaptive Running Platform for Moving in Virtual Reality // *Automation and Remote Control*. 2022. Vol. 83, N. 3. P. 355–366.

A Control Method Based on Computer Vision and Machine Learning Technologies for Adaptive Systems

A. D. Obukhov, obuhov.art@gmail.com, A. O. Nazarova, nazarova.al.ol@yandex.ru,
Tambov State Technical University, Tambov, 392000, Russian Federation

Corresponding author: Obukhov Artem D., Dr. of Tech. Sc., Associate Professor,
Tambov State Technical University, Tambov, 392000, Russian Federation,
e-mail: obuhov.art@gmail.com

Abstract

We consider the problem of organizing the control process in adaptive systems, in which it is required to ensure the preservation of the optimal state of the system when external conditions change. The analysis of existing approaches to its solution showed great promise in the synergistic effect of using machine learning and computer vision technologies. A system analysis of the management process using these technologies has been carried out. Its primary objects have been formalized, and the research task has been set. To solve it, a method is proposed, the novelty of which lies in the usage of machine learning and computer vision technologies for recognizing and obtaining a compressed idea of the state of the observed environment, objects of observation and control. And also, the choice of the control team was unified, based on three approaches: a system of rules, a neural network with classification, and machine learning with reinforcement. All stages of the method are formalized, and the possibility of using machine learning technologies (neural networks) for their implementation is theoretically substantiated. The practical significance of the developed method lies in the possibility of automating the activities of a human operator in complex adaptive systems through the use of machine learning and computer vision technologies. The method was tested on the example of an adaptive running platform control system. Experimental studies have been carried out to assess the efficiency of the method, its performance and accuracy of work in determining the state of objects of observation using computer vision technologies. The result of the work is the proven high efficiency of the proposed approach. The usage of computer vision and machine learning technologies made it possible not only to control the adaptive running platform but also to determine critical situations (falling or sudden stop of a person), which increases the safety of the control system, expands its functionality in monitoring the state of the environment and objects of observation.

Keywords: adaptive systems, process management, computer vision, machine learning, neural networks, object recognition

Acknowledgements: This article was prepared with the financial support of the Ministry of Science and Higher Education of the Russian Federation under the grant of the President of the Russian Federation MK-857.2022.1.6.

For citation:

Obukhov A. D., Nazarova A. O. A Control Method Based on Computer Vision and Machine Learning Technologies for Adaptive Systems, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2023, vol. 24, no. 1, pp. 14–23.

DOI: 10.17587/mau.24.14-23

References

1. Raibulet C., Arcelli Fontana F., Carettoni S. A preliminary analysis of self-adaptive systems according to different issues, *Software Quality Journal*, 2020, vol. 28, no. 3, pp. 1213–1243.
2. Weyns D., Bencomo N., Calinescu R., Camara J., Ghezzi C., Grassi V., Tamburrelli G. Perpetual assurances for self-adaptive systems, *Software Engineering for Self-Adaptive Systems III. Assurances*. Springer, Cham, 2017, pp. 31–63.
3. Mahdavi-Hezavehi S., Avgeriou P., Weyns D. A classification framework of uncertainty in architecture-based self-adaptive systems with multiple quality requirements, *Managing Trade-Offs in Adaptable Software Architecture*, Morgan Kaufmann, 2017, pp. 45–77.
4. Weyns D. An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective, John Wiley & Sons, 2020.
5. Casadei R., Fortino G., Pianini D., Russo W., Savaglio C., Viroli M. Modelling and simulation of opportunistic IoT services with aggregate computing, *Future Generation Computer Systems*, 2019, vol. 91, pp. 252–262.
6. Chen Y., Li W., Sakaridis C., Dai D., Van Gool L. Domain adaptive faster r-cnn for object detection in the wild, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3339–3348.
7. Colyer S. L., Evans M., Cosker D. P., Salo A. I. A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system, *Sports medicine-open*, 2018, vol. 4, no. 1, pp. 1–15.
8. Lyons N. Deep Learning-based Computer Vision Algorithms, Immersive Analytics and Simulation Software, and Virtual Reality Modeling Tools in Digital Twin-driven Smart Manufacturing, *Economics, Management, and Financial Markets*, 2022, vol. 17, no. 2, pp. 67–81.
9. Ge M., Zhang Y. Visual Autopilot Decision System Based on Deep Learning, *3D Imaging Technologies—Multi-dimensional Signal Processing and Deep Learning*, Springer, Singapore, 2021, pp. 329–335.
10. Oudah M., Al-Naji A., Chahl J. Hand gesture recognition based on computer vision: a review of techniques, *Journal of Imaging*, 2020, vol. 6, no. 8, pp. 73.
11. González Izard S., Sánchez Torres R., Alonso Plaza Ó., Juanes Méndez J. A., García-Peñalvo F. J. Nextmed: automatic imaging segmentation, 3D reconstruction, and 3D model visualization platform using augmented and virtual reality, *Sensors*, 2020, vol. 20, no. 10, 2962 p.
12. Kazemian A., Yuan X., Davtalab O., Khoshnevis B. Computer vision for real-time extrusion quality monitoring and control in robotic construction, *Automation in Construction*, 2019, vol. 101, pp. 92–98.
13. Yang L. I., Huang J., Feng T. I. A. N., Hong-An W. A. N. G., Guo-Zhong D. A. I. Gesture interaction in virtual reality, *Virtual Reality & Intelligent Hardware*, 2019, vol. 1, no. 1, pp. 84–112.
14. Cheng Z., Sun H., Takeuchi M., Katto J. Deep convolutional autoencoder-based lossy image compression, *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 253–257.
15. Zhou L., Cai C., Gao Y., Su S., Wu J. Variational autoencoder for low bit-rate image compression, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2617–2620.
16. Tsimpouris E., Tsakiridis N. L., Theocharis J. B. Using autoencoders to compress soil VNIR–SWIR spectra for more robust prediction of soil properties, *Geoderma*, 2021, vol. 393, 114967 p.
17. Butyrskij E. Yu., Kuvaldin I. A., Chalkin V. P. Approximation of multidimensional functions, *Nauchnoe Priborostroenie*, 2010, vol. 20, no. 2, pp. 82–92 (in Russian).
18. Krasnyanskiy M. N., Obukhov A. D., Dedov D. L. Control System for an Adaptive Running Platform for Moving in Virtual Reality, *Automation and Remote Control*, 2022, vol. 83, no. 3, pp. 355–366.