УДК 004.896
DOI: 10.26102/2310-6018/2026.56.5.003

Методика верификации функции вознаграждения для обучения политик локомоции четвероногого робота

А.С. Героев[1]✉, О.М. Гергет[1], А.В. Башкирова[1], А.А. Фильченков[2]

[1]Институт проблем управления имени В.А. Трапезникова РАН, Москва, Российская Федерация
[2]Московский политехнический университет, Москва, Российская Федерация

Резюме. В статье предложен подход к моделированию функции вознаграждения путем последовательного тестирования ее функциональных компонент. Некорректные функциональные компоненты могут привести к тому, что максимальное значение результирующей функции перестанет соответствовать желаемому целевому поведению робота. Для решения этой проблемы, а также предварительной оценки самой функции была предложена методика верификации, позволяющая проводить систематическую проверку как отдельных компонент функции вознаграждения, так и их весовых коэффициентов до начала длительного и ресурсоемкого обучения политики. Методика включает в себя формирование набора желательных и нежелательных сценариев поведения робота для последующей оценки изменения функции вознаграждения и ее функциональных компонент. Предложен двухуровневый метод тестирования: на первом уровне тестируются отдельные функциональные компоненты, отвечающие за соблюдение желаемых критериев движения робота, таких как сохранение целевой скорости, сохранение целевой устойчивости корпуса, сохранение целевой высоты корпуса и т. д. на предмет их монотонного убывания в нежелательных состояниях. На втором уровне тестируется результирующая функция взвешенной суммы этих компонент, чтобы убедиться, что дисбаланс весов не приводит к росту награды при потере устойчивости, падении или движению с нежелательной скоростью в нежелательном направлении. Особое внимание уделяется тесту на соответствие желательному состоянию – сценарию идеального прямолинейного движения, который позволяет выявить «некорректные» наборы коэффициентов, при которых штрафующие компоненты доминируют даже в идеальных условиях. Экспериментальная проверка проведена на модели робота Unitree Go1 в среде PyBullet. Результаты подтверждают, что предложенные тесты эффективно выявляют ошибки в реализации компонент и дисбаланс весов, что существенно повышает надежность процесса обучения и сокращает временные затраты на разработку.

Ключевые слова: обучение с подкреплением, окружение четвероногого робота, интеллектуальный агент, пространство состояний, пространство действий, функция вознаграждения, локомоция.

Для цитирования: Героев А.С., Гергет О.М., Башкирова А.В., Фильченков А.А. Методика верификации функции вознаграждения для обучения политик локомоции четвероногого робота. *Моделирование, оптимизация и информационные технологии.* 2026;14(5). URL: https://moitvivt.ru/ru/journal/article?id=2272 DOI: 10.26102/2310-6018/2026.56.5.003

Reward function verification methodology for training locomotion policies of a quadruped robot

A.S. Geroev[1]✉, O.M. Gerget[1], A.V. Bashkirova[1], A.A. Filchenkov[2]

[1]V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, the Russian Federation
[2]Moscow Polytechnic University, Moscow, the Russian Federation

Abstract. This article proposes an approach to reward function modeling through sequential testing of its functional components. Incorrect functional components can lead to the maximum value of the resulting function no longer corresponding to the desired robot behavior. To address this issue and to preliminarily evaluate the function itself, a verification method was proposed that allows for the systematic verification of both individual reward function components and their weighting coefficients before beginning time-consuming and resource-intensive policy training. The method involves generating a set of desirable and undesirable robot behavior scenarios for subsequent evaluation of the reward function and its functional components. A two-level testing method is proposed: at the first level, individual functional components responsible for maintaining desired robot motion criteria, such as maintaining target speed, maintaining target body stability, maintaining target body height, etc., are tested for monotonic decrease in undesirable states. At the second level, the resulting function of the weighted sum of these components is tested to ensure that weight imbalances do not lead to increased reward during instability, falls, or movement at an undesirable speed in an undesirable direction. Particular attention is paid to testing for compliance with the desired state – a scenario of ideal linear motion – which helps identify "incorrect" sets of coefficients where penalizing components dominate even under ideal conditions. Experimental validation was conducted on a Unitree Go1 robot model in the PyBullet environment. The results confirm that the proposed tests effectively identify component implementation errors and weight imbalances, significantly increasing the reliability of the training process and reducing development time.

Keywords: reinforcement learning, environment of a quadruped robot, intelligent agent, state space, action space, reward function, locomotion.

For citation: Geroev A.S., Gerget O.M., Bashkirova A.V., Filchenkov A.A. Reward function verification methodology for training locomotion policies of a quadruped robot. *Modeling, Optimization and Information Technology.* 2026;14(5). (In Russ.). URL: https://moitvivt.ru/ru/journal/article?id=2272 DOI: 10.26102/2310-6018/2026.56.5.003

Введение

В настоящее время для моделирования поведения роботов в условиях стохастического окружения широко применяют нейросетевые политики, обучаемые при помощи алгоритмов обучения с подкреплением (RL) [1]. Агент RL взаимодействует с окружением: на каждом шаге получает вектор состояния s_t, генерирует действие a_t и получает скалярную награду r_t. Задача агента – максимизировать ожидаемую сумму будущих наград:

$$J(\pi) = E_\pi[\sum_{t=0}^{T} \gamma^t r_t], \quad (1)$$

где π – политика, $\gamma \in [0,1)$ – коэффициент дисконтирования.

В контексте робототехники окружением является сам робот: численные показания системы датчиков формируют вектор состояний, а действия представляют собой вектора управляющих воздействий на приводы. Для безопасного поиска удовлетворительных политик обучение проводится в физическом симуляторе. Однако последующий перенос обученной политики на реального робота зачастую сопряжен с проблемой sim-to-real gap – расхождением динамик симуляционной и реальной среды [2]. Для повышения робастности политики к изменениям динамики применяют подход sim-to-sim: политику, обученную в одном симуляторе, дообучают или тестируют в другом [3, 4]. Однако даже при переносе между симуляторами необходимо быть уверенным в корректности самого окружения и функции вознаграждения.

Актуальность исследования. Функция вознаграждения является ключевым элементом, определяющим поведение обучаемого агента. Некорректные допущения при ее проектировании могут устойчиво закрепить ошибочные стратегии: агент оптимизирует функцию вознаграждения, а не желаемое поведение напрямую. Даже

небольшие дефекты в сигнале награды способны привести к тому, что агент выучит политику прихода в нежелательные состояния, которые максимизируют формальную метрику, но не соответствуют реальным целям управления. Поэтому необходимы систематические методы проверки (верификации) того, что функция вознаграждения согласованно и монотонно реагирует на изменение качества поведения робота [2, 5].

В данной работе предлагается методика тестирования функции вознаграждения для окружения четвероногого робота, включающая:

– определение желательных и нежелательных состояний;

– покомпонентную верификацию функциональных составляющих награды;

– верификацию итоговой взвешенной функции при различных наборах весовых коэффициентов;

– тест на соответствие функции вознаграждения желательному состоянию при идеальном прямолинейном движении.

Материалы и методы

Постановка задачи

Вектор состояния и функция вознаграждения. Окружение формирует вектор наблюдений $s_t \in R_n$, включающий в себя положение и ориентацию корпуса, линейные и угловые скорости центра масс робота, углы и скорости движимых шарниров, а также сенсорные сигналы контакта стоп с опорной поверхностью:

$$s_t = \left[q_t^j, \dot{q_t^j}, \tau_t, q_t, \omega_t, c_t\right], \tag{2}$$

где $q_t^j \in R^{12}$ – углы сочленений по три на каждую из четырех конечностей $q_i^j \in [-\pi/2\,;\pi/2]$ рад; $\dot{q_t^j}$ – угловые скорости сочленений $\dot{q_t^j} \in (-21,+21)$ рад/с; $\tau_t \in R^{12}$ – моменты на приводах; $\tau_t \in [-40{,}40]$ Н·м (ограничение привода); $q_t \in R^4$ – ориентация корпуса в виде единичного кватерниона, $\| qt \| = 1$; $\omega_t \in R^3$ – угловые скорости корпуса (крен, тангаж, рыскание); $\omega_i \in (-100,+100)$ рад/с; $c_t \in \{0{,}1\}^4$ – бинарные признаки контакта каждой из четырех стоп с опорной поверхностью ($c_i = 1$ – стопа в контакте, $c_i = 0$ – нет).

Итоговая функция вознаграждения рассматривается как взвешенная сумма компонент:

$$R_t = \sum_i \omega_i \cdot R_i(s_t, a_t), \tag{3}$$

где ω_i – весовые коэффициенты, а R_i – функциональные компоненты, отвечающие за различные аспекты поведения. Конкретная функция для математической модели робота Unitree Go1 имеет вид:

$$R_t = R_{vel} + R_{yaw} + R_{post} + R_{height} + R_{energy} + R_{\Delta a} + R_{contact} + R_{stab} + R_{survive} + \\ + R_{tail} + R_{progress}. \tag{4}$$

Компоненты определяются следующим образом:

$$R_{vel} = -k_v \left[(v_x - v_x^*)^2 + \left(v_y - v_y^*\right)^2\right] + kv_x, \tag{5}$$

$$R_{yaw} = -k_\omega(\omega_z - \omega_z^*)^2, \tag{6}$$

$$R_{post} = -k_{post}(\phi^2 - \theta^2), \tag{7}$$

$$R_{height} = -k_h(h - h^*)^2, \tag{8}$$

$$R_{energy} = -k_\tau \sum_i \tau_i^2, \tag{9}$$

$$R_{\{\Delta a\}} = -k_{\Delta a}^2 \sum_i \left(a_{t,i} - a_{t-1,i}\right)^2, \tag{10}$$

$$R_{contact} = k_c \frac{1}{N} \sum_i c_i, \tag{11}$$

$$R_{stab} = k_\eta \eta, \tag{12}$$

$$R_{survive} = k_s, \tag{13}$$

$$R_{fall} = \begin{cases} -1, |\phi| > \phi_{max} \text{ или } |\theta| > \theta_{max} \\ 0, \text{иначе} \end{cases}, \tag{14}$$

$$R_{progress} = k_p \cdot \Delta x. \tag{15}$$

Функциональное назначение каждой компоненты:

– R_{vel} – компонента скорости. Данная компонента необходима для стимулирования политики обеспечивать движение робота вдоль заданного вектора скорости. В нашем случае – направленного вдоль оси X мировой системы координат.

– R_{yaw} – компонента рыскания. Данная компонента необходима для стимулирования политики сохранять первоначальную угловую скорость тела робота. Достигается желаемое поведение за счет штрафа за отклонение угловой скорости по оси рыскания ω_z от целевого значения ω_z^*. Предотвращает неконтролируемые повороты робота.

– R_{post} – компонента ориентации. Данная компонента необходима для стимулирования политики сохранять ориентацию за счет штрафов отклонений углов крена ϕ и тангажа θ от нуля.

– R_{height} – компонента высоты. Данная компонента необходима для стимулирования политики сохранять целевую высоту корпуса за счет штрафа за отклонение высоты корпуса h от целевой h^*.

– R_{energy} – компонента энергоэффективности. Данная компонента необходима для нормализации действий, генерируемых политикой. Достигается это за счет штрафования агента за суммарный квадрат моментов τ_i на приводах.

– $R_{\Delta a}$ – компонента плавности действий. Данная компонента необходима для нормализации траекторий движителей робота, которые получаются вследствие исполнения последовательностей действий, генерируемых политикой. Достигается это за счет штрафования агента за суммарный квадрат разницы значений действий на соседних временных шагах политики.

– $R_{contact}$ – компонента контактов. Данная компонента необходима для стимулирования политики генерировать такие траектории дейтвий, которые будут сохранять частичный контакт робота с поверхностью, что не позволит политике выучивать прыжки вперед, и потребует генерировать стабильную походку.

– R_{stab} – компонента устойчивости. Данная компонента необходима для стимулирования политики сохранять устойчивость при ходьбе. Достигается это путем поощрения за высокий запас устойчивости. η – минимальное расстояние от проекции центра масс на опорную плоскость до ближайшей стороны опорного треугольника, образованного стопами в контакте с поверхностью.

– $R_{survive}$ – бонус выживания. Данный компонент необходим для того, чтобы политика старалась максимизировать итоговую функцию вознаграждения на протяжении всего времени моделирования, отведенного для одной итерации обучения.

Данный коэффициент позволяет избежать ситуаций, когда агент стремится максимизировать награду в долгосрочной перспективе обучения путем уменьшения времени до наступления терминального состояния, знаменующего начало новой итерации обучения.

– R_{fall} – штраф за переворот. Активируется при превышении критических углов крена или тангажа. Данный компонент необходим для того, чтобы политика не пыталась максимизировать функцию вознаграждения за счет увеличения линейной скорости при нежелательных ориентациях. Пороговые значения ϕ_{max}, θ_{max} вычисляются из габаритов корпуса по геометрии конкретного робота, в нашем случае Unitree Go1.

– $R_{progress}$ – компонента прогресса. Поощряет приращение координаты Δx по оси движения между соседними шагами. Непосредственно стимулирует перемещение робота вперед.

– Здесь v_x и v_y – линейные скорости, ω_z – угловая скорость по оси рыскания, ϕ и θ – углы крена и тангажа, h – высота корпуса, τ_i – моменты на приводах, $a_{t,i}$ – компоненты действия, c_i – контакты стоп, Δx – приращение координаты по оси движения. Пороговые значения ϕ_{max} и θ_{max} вычисляются из габаритов корпуса по геометрии URDF-модели.

Желательные и нежелательные состояния. Для формализации задачи тестирования введем понятия желательного и нежелательного состояний окружения.

Желательное состояние – перемещение центра масс робота вперед с заданной скоростью v_x^* на целевой высоте h^* с минимальными отклонениями ориентации и минимальными энергозатратами на протяжении заданного числа шагов управления. При достижении желательного состояния итоговая функция вознаграждения должна удовлетворять условию:

$$R_T \gg R_{survive}, R_t = const > 0. \quad (16)$$

Условия сформированы, исходя из принципа того, что нейросетевая политика в процессе обучения стремится решить первую задачу линейного программирования, аппроксимируя закон изменения функции вознаграждения в зависимости от входных функциональных элементов датчиков робота. Неотъемлемым компонентом любой функции вознаграждения является так называемый бонус выживания, в некоторых случаях он может быть равен нулю, однако, в свою очередь, всегда удовлетворяет условию $R_{survive} \geq 0$. Этот компонент необходим, чтобы агент в процессе обучения стремился избегать терминальных состояний, при которых максимизация функции вознаграждения невозможна. Такие состояния при обучении формируются внешними гиперпараметрами и ведут к старту новой итерации обучения. Множественное попадание в такие терминальные состояния делает выученные траектории сложно дифференцируемыми. Если этот коэффициент будет сопоставим с оставшейся взвешенной суммой компонент уравнения, то будет наблюдаться существенное замедление скорости сходимости обучения агента при нахождении весов политики в локальном минимуме, особенно если агент не является эпсилон-жадным, или же его коэффициент дисконтирования слишком высок. Нежелательные состояния классифицируются следующим образом:

1. Потеря устойчивости – уменьшение запаса устойчивости вплоть до падения;
2. Компенсация устойчивости снижением центра масс – движение с высотой корпуса $h \ll h^*$, при котором возрастает запас устойчивости, но нарушается целевая высота;

3. Переворот – превышение критических углов крена или тангажа ($\vee \phi \vee \phi_{max}$ или $\theta \vee \theta_t$).

При нахождении в нежелательном состоянии функция вознаграждения должна стремиться к $-\infty$:

$$R_T \to -\infty. \quad (17)$$

Принципиальным требованием является согласованность: нежелательное поведение, при котором улучшается только часть компонент (например, высокая скорость при потере устойчивости), не должно приводить к росту итоговой функции R_t [2, 6]. В противном случае агент 4 может выучить стратегию, оптимизирующую частные компоненты и ведущую к деградации реального поведения.

Методика тестирования

Предлагаемая методика представляет собой двухуровневую вложенную функцию вида:

$$f(\sum_i^n \omega_i g_i(s_i)), \quad (18)$$

где g_i – отдельный компонент; i, w_i – соответствующий весовой коэффициент; n – количество тестируемых компонент; f – итоговая функция вознаграждения. На первом этапе тестирования тестируются отдельные компоненты g_i: при введении робота в нежелательное состояние соответствующая компонента должна монотонно убывать. На втором этапе проверяется, что при тех же сценариях убывает и итоговая функция f.

Тест устойчивости (опорный треугольник). Метрика устойчивости определяется через опорный многоугольник, редуцируемый до треугольника, образованного тремя ногами в контакте с поверхностью. Пусть η – минимальное расстояние от проекции центра масс на опорную плоскость до ближайшей стороны опорного треугольника. Компонента устойчивости задается соотношением $R_{stab} = k_\eta \eta$. График изменения компоненты устойчивости во время теста устойчивости приведён на Рисунке 1. График изменения функции вознаграждения в ходе теста устойчивости приведён на Рисунке 2.

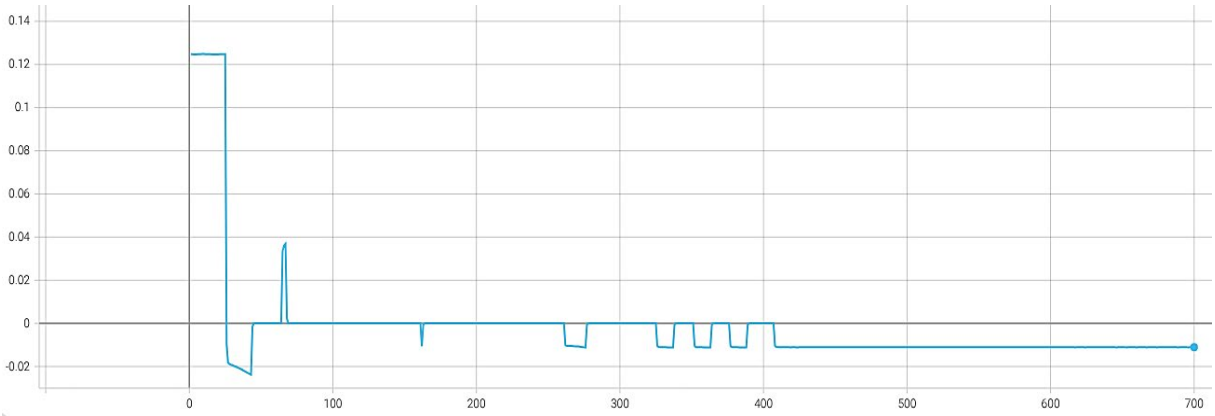

Рисунок 1 – График изменения компоненты устойчивости: по оси Y – значение компоненты R_{stab}; по оси X – значения шага моделирования

Figure 1 – Graph of the change in the stability component: the Y-axis shows the value of the R_{stab} component; the X-axis shows the values of the simulation step

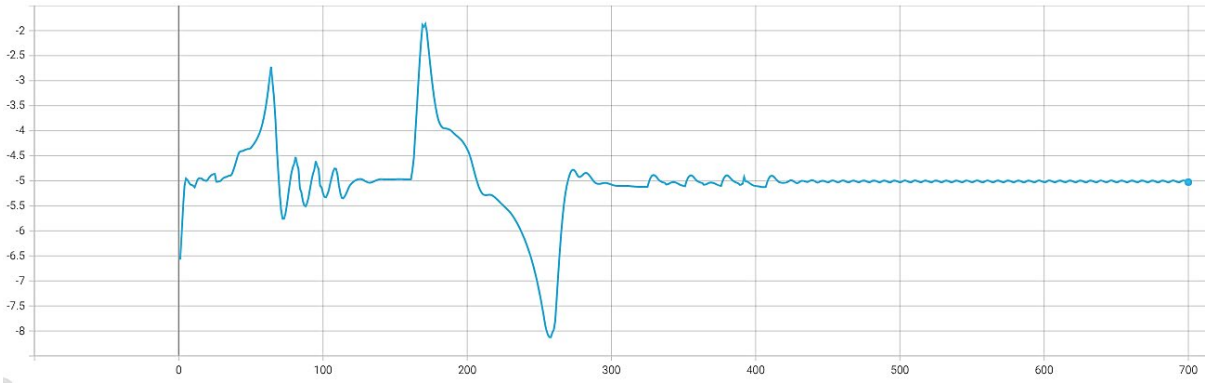

Рисунок 2 – График изменения итоговой функции вознаграждения: по оси Y – значение функции вознаграждения R_t; по оси X – значения шага моделирования
Figure 2 – Graph of the change in the final reward function: the Y-axis shows the value of the reward function R_t, the X-axis shows the values of the simulation step

Процедура тестирования: фиксация статической позы на конечном числе шагов, затем поочередное исключение опоры одной из ног (моделирование подъема конечности). Сравнивается среднее значение R_{stab} в этих режимах. Устойчивость не должна возрастать при потере опоры; 5 итоговая награда R_t не должна увеличиваться. Статическая поза сохраняется на протяжении 25 эпизодов. Затем каждая конечность робота поочередно поднимается. В силу инертности самого робота, а также погрешности измерений инерциальных датчиков, наблюдаются шумы при колебании тела робота. На графике видно, что устойчивость в эти периоды зашумления возрастает. Однако из-за компенсирующего воздействия других компонент функции вознаграждения, а также величины штрафующего весового коэффициента самой компоненты устойчивости, итоговая функция вознаграждения остается в отрицательной полуплоскости, несмотря на шумы измерения.

Тест высоты корпуса. Компонента высоты задается штрафом за отклонение: $R_{height} = -k_h(h - h^*)^2$. При статическом стоянии в базовой позе значение должно быть близким к нулю, при падении – существенно уменьшаться. Тест проводится в двух режимах: удержание позы и сценарий падения при отключении управляющих воздействий. График изменения компоненты высоты во время теста высоты корпуса приведён на Рисунке 3. График изменения функции вознаграждения в ходе теста высоты корпуса приведён на Рисунке 4.

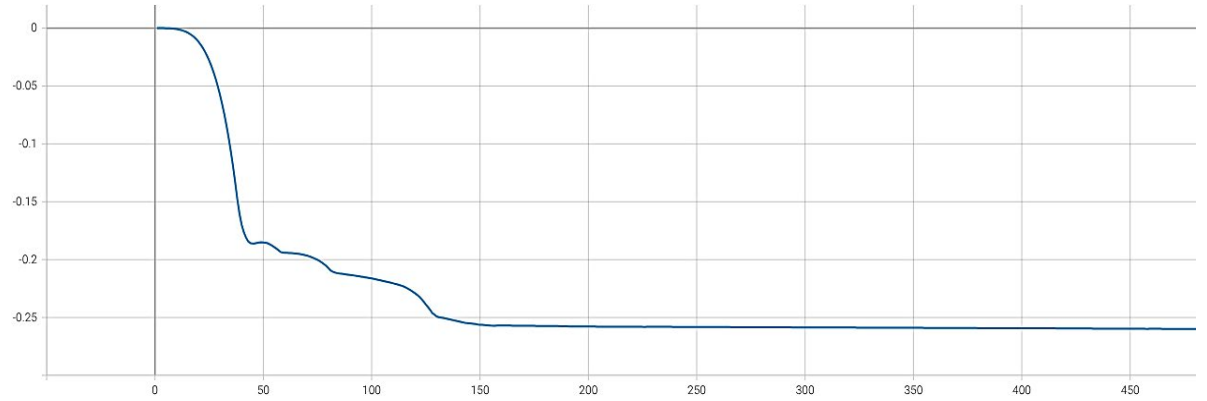

Рисунок 3 – График изменения компоненты высоты: по оси Y – значение компоненты R_{height}; по оси X – значения шага моделирования
Figure 3 – Graph of the change in the height component: the Y-axis shows the value of the R_{height} component; the X-axis shows the simulation step values

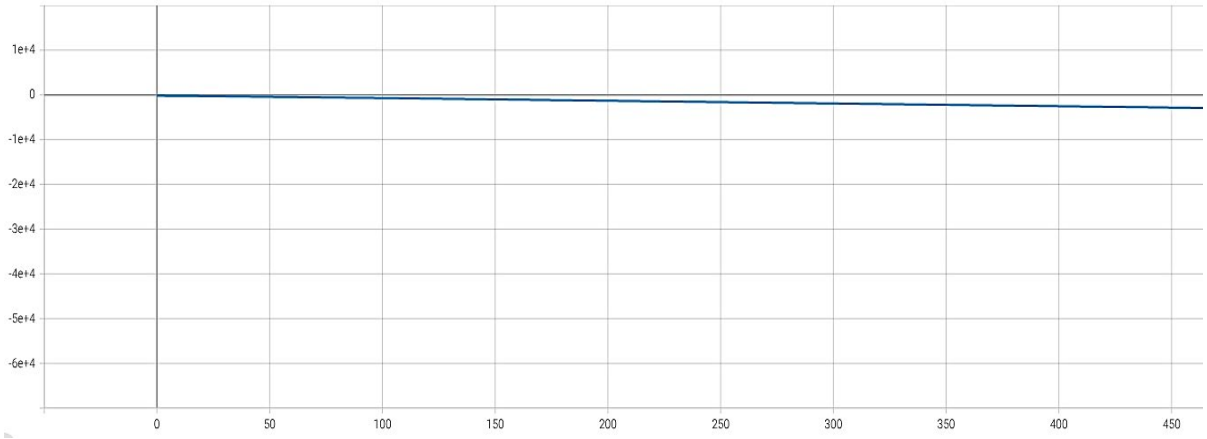

Рисунок 4 – График изменения итоговой функции вознаграждения: по оси Y – значение функции вознаграждения R_t; по оси X – значения шага моделирования
Figure 4 – Graph of the change in the final reward function: the Y-axis shows the value of the reward function R_t; the X-axis shows the values of the simulation step

Тест штрафа за наклон. Штраф за переворот принимает значение, отличное от нуля, при превышении пороговых углов крена и тангажа. Тест проводится при контролируемом изменении ориентации корпуса в условиях отсутствия гравитации: при превышении порогов R_{fall} принимает отрицательное значение, а итоговая награда R_t заметно снижается. Проверяется отсутствие компенсации за счет других компонент. График изменения компоненты штрафа за переворот во время теста штрафа за наклон приведён на Рисунке 5. График изменения функции вознаграждения в ходе теста штрафа за наклон приведён на Рисунке 6.

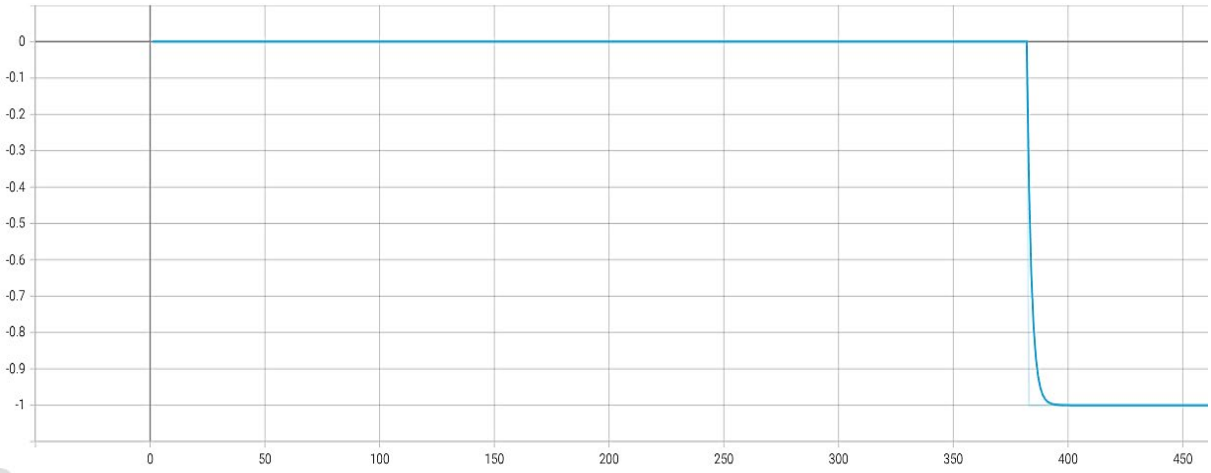

Рисунок 5 – График изменения компоненты штрафа за переворот: по оси Y – значение компоненты R_{fall}; по оси X – значения шага моделирования
Figure 5 – Graph of the change in the rollover penalty component: the Y-axis shows the value of the R_{fall} component; the X-axis shows the simulation step values

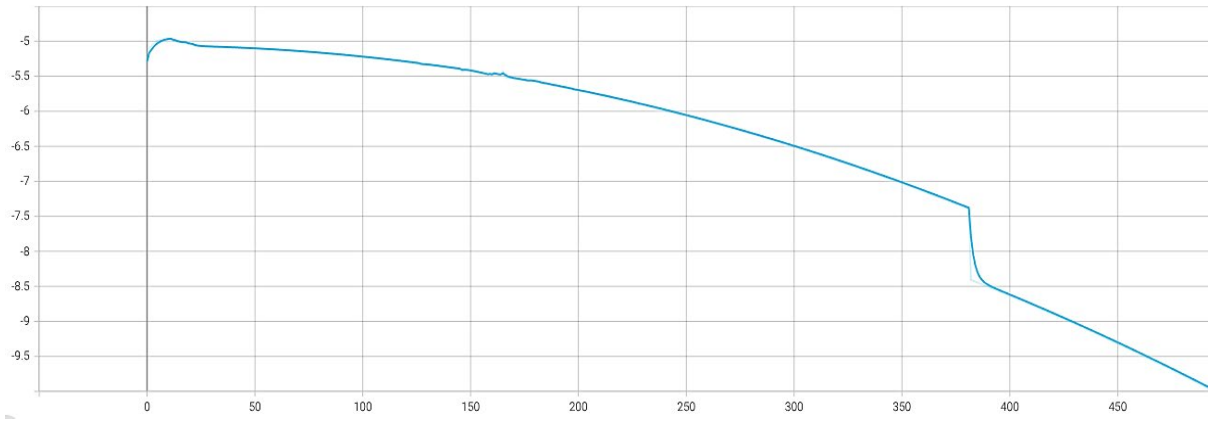

Рисунок 6 – График изменения итоговой функции вознаграждения: по оси Y – значение функции вознаграждения R_t; по оси X – значения шага моделирования
Figure 6 – Graph of the change in the final reward function: the Y-axis shows the value of the reward function R_t; the X-axis shows the values of the simulation step

Тест скорости. Компонента скорости (5) поощряет движение в направлении целевой скорости с квадратичным штрафом за отклонение. Тест проводится при постоянном внешнем воздействии по оси X с силой F, что ведет к постоянному набору скорости. Из-за этого робот сначала достигает целевой скорости v_x^*, а затем превосходит ее. Сила трения скольжения в ходе теста устанавливается ≈ 0. Прохождение теста валидируется наблюдением за изменением динамики R_{vel} и по согласованности с итоговой функцией вознаграждения R_t. График изменения компоненты скорости во время теста скорости приведён на Рисунке 7. График изменения функции вознаграждения в ходе теста скорости приведён на Рисунке 8.

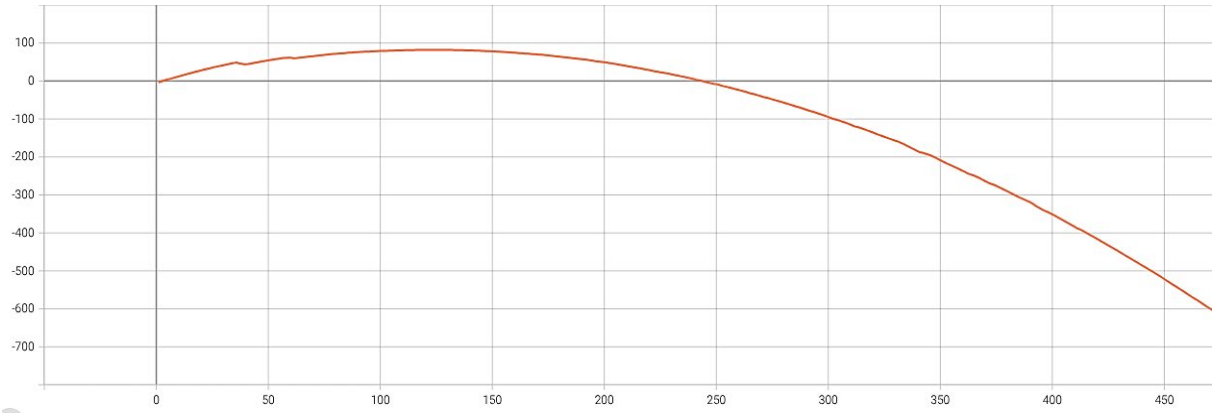

Рисунок 7 – График изменения компоненты скорости: по оси Y – значение компоненты R_{vel}; по оси X – значения шага моделирования
Figure 7 – Graph of the change in the velocity component. The Y-axis shows the value of the R_{vel} component; the X-axis shows the simulation step values

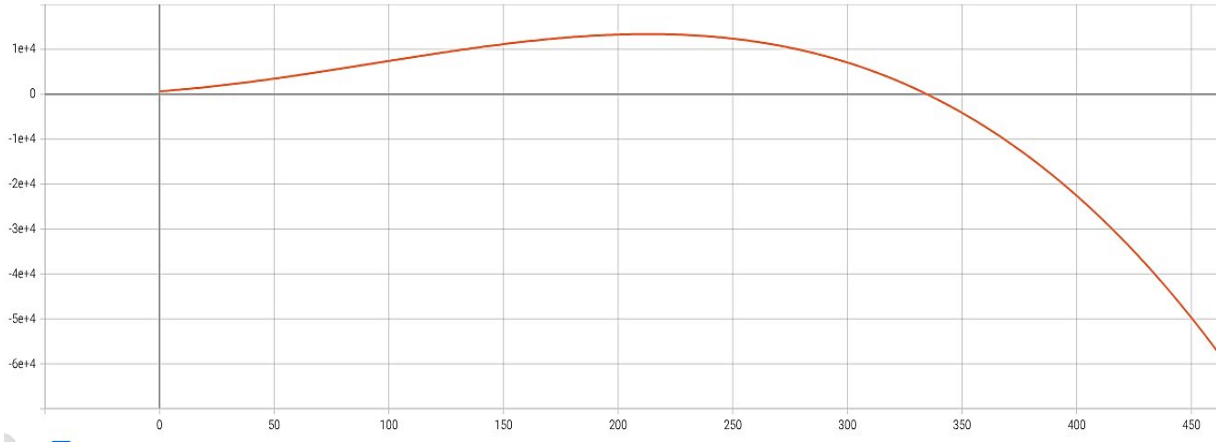

Рисунок 8 – График изменения итоговой функции вознаграждения: по оси Y – значение функции вознаграждения R_t; по оси X – значения шага моделирования
Figure 8 – Graph of the change in the final reward function: the Y-axis shows the value of the reward function R_t; the X-axis shows the values of the simulation step

По итогам тестирования функции вознаграждения в выбранных нежелательных состояниях мы наблюдаем согласованность всех функциональных компонент функции. Согласованность выражается в соблюдении взаимного убывания самой функции и отвечающей нежелательному состоянию компоненты функции.

Тест на соответствие желательному состоянию (идеальное прямолинейное движение). Для проверки того, что итоговая функция вознаграждения принимает устойчивое положительное значение при желательном поведении, моделируется сценарий идеального прямолинейного движения. Условия сценария:

– коэффициент трения с поверхностью $\mu \approx 0$;
– робот перемещается вдоль оси X с постоянной целевой скоростью v_x^* на высоте h^*;
– ориентация корпуса фиксирована (нулевые углы крена, тангажа, рыскания);
– сочленения удерживаются в номинальной позе позиционным управлением.

На каждом шаге симуляции положение и скорость базы корпуса принудительно устанавливаются:

$$p_t = (v_x^* \cdot t \cdot \Delta t, 0, h^*), \tag{19}$$

$$\dot{p_t} = (v_x^*, 0,0),\ \omega_t = 0. \tag{20}$$

После каждого шага вычисления обновленных физических параметров системы вычисляется награда R_t. Критерий прохождения теста описан в формуле (16).

Если условие (16) нарушается, это свидетельствует о том, что штрафующие компоненты функции вознаграждения доминируют над поощряющими даже при идеальных условиях, и весовые коэффициенты нуждаются в пересмотре.

Верификация весовых коэффициентов. После прохождения покомпонентных тестов проводится верификация весовых коэффициентов, при этом определяются два набора коэффициентов:

– «Корректные» коэффициенты – баланс поощряющих и штрафующих компонент обеспечивает выполнение условия (16) при желательном состоянии и условия (17) – при нежелательных;

– «Некорректные» коэффициенты – завышенные штрафующие и заниженные поощряющие веса; итоговая функция не проходит тест идеального движения (16).

В частности, для исследуемой функции вознаграждения робота Go1 при «корректных» коэффициентах k_v = 2,0, а при «некорректных» k_v = 0,01. Уменьшение поощряющего коэффициента скорости приводит к тому, что штрафующие компоненты (энергия, плавность действий) подавляют положительный вклад скоростной составляющей даже при идеальном движении; функция вознаграждения принимает отрицательные значения, что нарушает условие (16).

Таким образом, предложенный набор тестов позволяет отсеивать некорректные конфигурации весовых коэффициентов до запуска длительной процедуры обучения агента.

На Рисунке 9 изображены графики функций вознаграждения при:

– удовлетворительных коэффициентах – пунктирная линия;
– неудовлетворительных коэффициентах – сплошная линия.

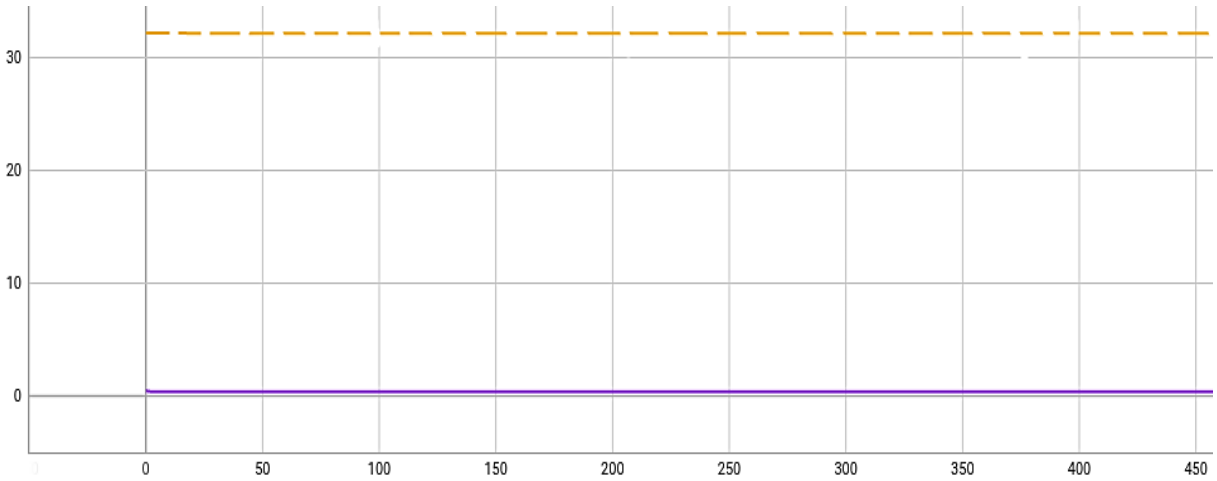

Рисунок 9 – Функции вознаграждения при удовлетворительных и неудовлетворительных коэффициентах поощряющих компонент: по оси Y – значение функции вознаграждения R_t; по оси X – значения шага моделирования

Figure 9 – Reward functions for satisfactory and unsatisfactory coefficients of incentive components: the Y-axis shows the value of the reward function R_t; the X-axis shows the values of the simulation step

При изменении знака основных поощряющих коэффициентов функции наблюдается ее отражение в отрицательную полуплоскость, что подтверждает корректность моделирования самого симуляционного окружения. На Рисунке 10 голубая линия отражает функцию вознаграждения при удовлетворительных и положительных коэффициентах функции вознаграждения. Розовая линия представляет собой значение функции вознаграждения при некорректных отрицательных поощряющих компонентах.

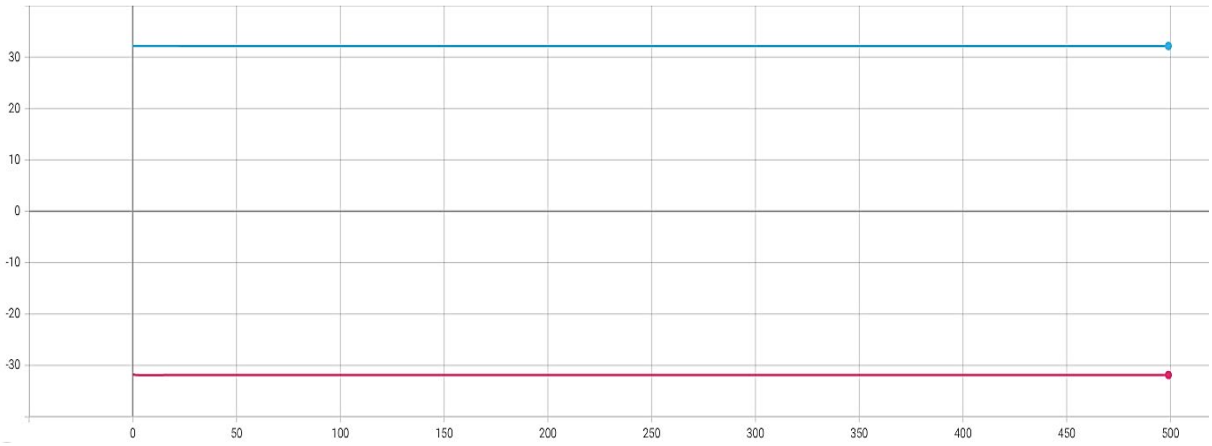

Рисунок 10 – Функции вознаграждения при положительных и отрицательных коэффициентах поощряющих компонент: по оси Y – значение функции вознаграждения R_t; по оси X – значения шага моделирования

Figure 10 – Reward functions for positive and negative coefficients of the incentive components: the Y-axis shows the value of the reward function R_t; the X-axis shows the values of the simulation step

Дизайн экспериментов

В работе используется симуляционная модель четвероногого робота Unitree Go1 в среде физического моделирования PyBullet [7]. Робот имеет 12 приводов по три на каждую конечность.

Эксперименты проводились на двух наборах весовых коэффициентов. Значения весовых коэффициентов в этих наборах представлены в Таблице 1.

Таблица 1 – Весовые коэффициенты функции вознаграждения Go1
Table 1 – Weight coefficients of the reward function Go1

Коэффициент	«Корректные»	«Некорректные»
k_v (скорость)	2,0	0,01
k_ω (рыскание)	0,5	0,5
k_p (ориентация)	5,0	5,0
k_h (высота)	10,0	10,0
k_τ (энергия)	0,001	0,001
$k_{\Delta a}$ (плавность)	0,01	0,01
k_c (контакт)	0,1	0,1
k_η (устойчивость)	1,0	1,0
k_s (выживание)	0,05	0,05

Для каждого набора коэффициентов выполнялся тест идеального прямолинейного движения с параметрами: $v_x^* = 1{,}6\,\text{м/с}$, $h^* = 0{,}25\,\text{м}$, длительность – 500 шагов симуляции.

Дополнительно выполнялись покомпонентные тесты нежелательных сценариев:

1. Потеря устойчивости – поочередный подъем каждой из конечностей при фиксированной позе (500 шагов на каждый сценарий);
2. Падение – отключение управляющих воздействий, робот падает под действием гравитации (500 шагов);
3. Переворот – контролируемое увеличение угла крена по 0,6 рад. в течение 500 шагов.

Результаты

Тесты компонент. Тесты функциональных компонент подтвердили их корректность:

– При устойчивой статической позе R_{stab} максимальна, при подъеме любой конечности – монотонно убывает;

– При стоянии $R_{height} \approx 0$; при падении – значительно уменьшается;

– При нулевом крене и тангаже $R_{fall} = 0$;

– При превышении пороговых значений $R_{fall} = -1$;

– Компонента скорости R_{vel} максимальна при достижении целевой скорости и убывает с увеличением отклонения от целевого значения.

Во всех нежелательных сценариях итоговая функция R_t также убывала, что подтверждает согласованность компонент

Тест идеального состояния при прямолинейном движении. При «корректных» коэффициентах ($k_v = 2{,}0$) функция вознаграждения на каждом из 500 шагов принимала строго положительные значения, образуя устойчивое плато $R_t \approx const > 0$. Условие (16) выполнялось для всех шагов. При «некорректных» коэффициентах ($k_v = 0{,}01$) функция вознаграждения принимала отрицательные значения уже на первых шагах симуляции. Заниженный поощряющий коэффициент скорости не компенсирует штрафующие компоненты даже при идеальных условиях, что подтверждает чувствительность теста к балансу весов.

Обсуждение

Предложенная методика позволяет на раннем этапе, до запуска длительной процедуры обучения, выявлять две категории ошибок:

1. Ошибки в функциональных компонентах – некорректная реализация отдельных составляющих награды (например, неверное вычисление высоты корпуса или запаса устойчивости);

2. Ошибки в весовых коэффициентах – дисбаланс, при котором штрафующие компоненты подавляют поощряющие даже при идеальном поведении, или наоборот – нежелательное поведение компенсируется ростом отдельных компонент.

Ключевым свойством методики является универсальность: тесты сформулированы в терминах общей структуры (3) и не зависят от конкретного вида компонент R_i. Это позволяет применять те же сценарии для верификации других функций вознаграждения – например, функции из работы [8], построенной по аналогичному принципу взвешенной суммы. Отдельно отметим возможность использования предложенных тестов в контексте sim-to-sim моделирования. При переносе политики между симуляторами функция вознаграждения и окружение целевого симулятора также нуждаются в верификации [4, 8]. Предложенные сценарии могут служить средством проверки корректности нового окружения перед дообучением, что снижает риск накопления ошибок при каскадном переносе sim-to-sim-to-real [3] и согласуется с современными подходами к анализу и верификации функций вознаграждения в задачах локомоции [9].

Заключение

В работе предложена методика верификации функции вознаграждения для задач обучения локомоции четвероногого робота, основанная на моделировании желательных и нежелательных сценариев поведения. Методика реализована как набор

автоматизированных тестов и экспериментально проверена на модели Unitree Go1 в среде PyBullet. Показано, что предложенные тесты чувствительны к дисбалансу весовых коэффициентов и позволяют выявлять некорректные конфигурации до начала обучения агента. Направления дальнейших исследований:

– распространение методики на другие классы роботов (двуногие, колесные, манипуляторы) [10];

– интеграция тестов в конвейер непрерывной интеграции (CI/CD) для автоматической проверки при изменении функции вознаграждения или параметров окружения;

– применение тестов как критерия остановки при автоматическом подборе весовых коэффициентов;

– использование в процедуре sim-to-sim для верификации целевого окружения перед дообучением политики.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O. *Proximal policy optimization algorithms.* arXiv. URL: https://arxiv.org/abs/1707.06347 [Accessed 5th February 2026].
2. Tobin J., Fong R., Ray A., et al. Domain randomization for transferring deep neural networks from simulation to the real world. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 24–28 September 2017, Vancouver, BC, Canada.* IEEE; 2017. P. 23–30. https://doi.org/10.1109/IROS.2017.8202133
3. Muratore F., Gienger M., Peters J. Assessing transferability from simulation to reality for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2021;43(4):1172–1183. https://doi.org/10.1109/TPAMI.2019.2952353
4. Ma Y.J., Liang W., Wang H.-J., et al. DrEureka: Language Model Guided Sim-To-Real Transfer. In: *Robotics: Science and Systems 2024, 15–19 July 2024, Delft, The Netherlands.* 2024. https://doi.org/10.15607/RSS.2024.XX.094
5. Kim M.-S., Kim J.-S., Park J.-H. Automated Hyperparameter Tuning in Reinforcement Learning for Quadrupedal Robot Locomotion. *Electronics.* 2024;13(1). https://doi.org/10.3390/electronics13010116
6. Hwangbo J., Lee J., Dosovitskiy A., et al. Learning agile and dynamic motor skills for legged robots. *Science Robotics*. 2019;4(26). https://doi.org/10.1126/scirobotics.aau5872
7. Bellegarda G., Chen Y., Liu Zh., Nguyen Q. *Robust High-speed Running for Quadruped Robots via Deep Reinforcement Learning.* arXiv. URL: https://arxiv.org/abs/2103.06484 [Accessed 12th February 2026].
8. Zhao Y., Wu T., Zhu Y., et al. *ZSL-RPPO: Zero-Shot Learning for Quadrupedal Locomotion in Challenging Terrains using Recurrent Proximal Policy Optimization.* arXiv. URL: https://arxiv.org/abs/2403.01928 [Accessed 5th February 2026].
9. Van Marum B., Shrestha A., Duan H., et al. *Revisiting Reward Design and Evaluation for Robust Humanoid Standing and Walking.* arXiv. URL: https://arxiv.org/abs/2404.19173 [Accessed 10th February 2026].
10. Soni R., Harnack D., Isermann H., et al. End-to-End Reinforcement Learning for Torque Based Variable Height Hopping. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 01–05 October 2023, Detroit, MI, USA.* IEEE; 2023. P. 7531–7538. https://doi.org/10.1109/IROS55552.2023.10342187

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Героев Александр Сергеевич, аспирант, младший научный сотрудник, Институт проблем управления имени В.А. Трапезникова РАН, Москва, Российская Федерация.
e-mail: geroev_sasha@mail.ru
ORCID: 0009-0000-1280-4709

Alexander S. Geroyev, Postgraduate, Research Assistant, V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, the Russian Federation.

Гергет Ольга Михайловна, доктор технических наук, доцент, ведущий научный сотрудник, Институт проблем управления имени В.А. Трапезникова РАН, Москва, Российская Федерация.
e-mail: olgagerget@mail.ru
ORCID: 0000-0002-6242-9502

Olga M. Gerget, Doctor of Engineering Sciences, Docent, Leading Researcher, V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, the Russian Federation.

Башкирова Анастасия Вячеславовна, магистр, инженер, институт проблем управления имени В.А. Трапезникова РАН, Москва, Российская Федерация.
e-mail: basana235@yandex.ru

Anastasiia V. Bashkirova, Master Degree, Engineer, V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, the Russian Federation.

Фильченков Александр Александрович, аспирант, ассистент, Московский политехнический университет, Москва, Российская Федерация.
e-mail: al.filchenkov@gmail.com

Alexander A. Filchenkov, Postgraduate, Assistant, Moscow Polytechnic University, Moscow, the Russian Federation.

Статья поступила в редакцию 06.03.2026; одобрена после рецензирования 27.04.2026; принята к публикации 11.05.2026.

The article was submitted 06.03.2026; approved after reviewing 27.04.2026; accepted for publication 11.05.2026.