

<https://doi.org/10.21869/2223-1560-2019-23-4-8-18>

Использование нейронных сетей для прогнозирования нормальных реакций шагающего робота

С.И. Савин¹, Л.Ю. Ворочаева² ✉

¹ Университет Иннополис, ул. Университетская, д.1, г. Иннополис, 420500, Российская Федерация

² ФГБОУ ВО «Юго-Западный государственный университет», ул. 50 лет Октября, 94, г. Курск, 305040, Российская Федерация

✉ e-mail: mila180888@yandex.ru

Резюме

Цель исследования. Данная работа посвящена решению одной из проблем, связанных с управлением шагающими роботами на основе их динамической математической модели, – наличию в ней явных механических связей, обусловленных реакциями связей с опорной поверхностью. Для решения указанной проблемы предлагается использовать полносвязную нейронную сеть для оценки сил нормальных реакций между поверхностью и стопами двуногого шагающего робота во время реализации им одного шага.

Методы. В работе рассмотрены две архитектуры нейронной сети, основанные на полносвязных слоях с ReLU активационными функциями. Архитектура нейронной сети включает в себя пять полносвязных слоев (входной, выходной и три скрытых), а альтернативная архитектура включает в себя слой прореживания после каждого полносвязного слоя. Входными данными для сети являются состояние робота и требуемые управляющие воздействия, а выходными – предсказанные силы реакции. Обучающая выборка генерируется с помощью моделирования полной динамической модели робота. Сеть построена и обучена с использованием библиотек машинного обучения Keras и TensorFlow.

Результаты. Описана генерация обучающей выборки для нейронной сети, проведено обучение двух архитектур нейронных сетей. На основании данных моделирования установлено, что обе обученные нейронные сети способны точно предсказывать значения нормальных реакций с использованием значений обобщенных координат и скоростей, а также управляющих воздействий в качестве входных данных, однако при этом наблюдается статическая ошибка предсказания.

Заключение. Полученные в рамках статьи результаты могут в дальнейшем использоваться для управления движением двуногих шагающих роботов по различным типам поверхностей.

Ключевые слова: двуногий шагающий робот; нейронная сеть; архитектура нейронной сети; слой прореживания; полносвязный слой; обучающая и верификационная выборки; нормальные реакции.

Благодарности: Работа выполнена при поддержке Гранта Президента МК-1537.2019.8, Савин С.И. работал при поддержке РФФИ, проект №18-38-00140\18.

Конфликт интересов: Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Для цитирования: Савин С.И., Ворочаева Л.Ю. Использование нейронных сетей для прогнозирования нормальных реакций шагающего робота // Известия Юго-Западного государственного университета. 2019; 23(4): 8-18. <https://doi.org/10.21869/2223-1560-2019-23-4-8-18>.

Статья поступила в редакцию 13.06.2019

Статья подписана в печать 22.07.2019

<https://doi.org/10.21869/2223-1560-2019-23-4-8-18>

Using Neural Networks to Predict the Normal Reactions of a Walking Robot

Sergey I. Savin ¹, Lyudmila Yu. Vorochaeva ² ✉

¹ Innopolis University, 1, Universitetskaya str., Innopolis, 420500, Russian Federation

² Southwest State University, 94, 50 Let Oktyabrya str., Kursk, 305040, Russian Federation

✉ e-mail: mila180888@yandex.ru

Abstract

Purpose of research. This work is devoted to solving one of the problems associated with the control of walking robots based on their dynamic mathematical model – the presence in it of obvious mechanical bonds due to reactions of bonds with the supporting surface. To solve this problem, it is proposed to use a fully connected neural network to evaluate the forces of normal reactions between the surface and the feet of a bipedal walking machine during its implementation of one step.

Methods. The paper considers two neural network architectures based on fully connected layers with ReLU activation functions. The architecture of the neural network includes five fully connected layers (input, output and three hidden), and an alternative architecture includes a thinning layer after each fully connected layer. The input data for the network are the state of the robot and the required control actions, and the output is the predicted reaction forces. The training sample is generated by modeling a complete dynamic model of the robot. The network is built and trained using machine learning libraries Keras and TensorFlow.

Results. The generation of training sample for neural network is described here, and it is carried out the training of two architectures of neural networks. Based on the simulation data, it was established that both trained neural networks are able to accurately predict the values of normal reactions using the values of generalized coordinates and velocities, as well as control actions as input, however, a static prediction error is observed.

Conclusion. The results obtained within the framework of the article can be further used to control the movement of bipedal walking machines on various types of surfaces.

Keywords: bipedal walking machine; neural network; neural network architecture; thinning layer; fully connected layer; training and verification samples; normal reactions.

Acknowledgements: This work was supported by the Grant of the President MK-1537.2019.8, Savin S.I. worked with the support of the Russian Federal Property Fund, project No. 18-38-00140 \ 18.

Conflict of interest. The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

For citation: Savin S.I., Vorochaeva L.Yu. Using Neural Networks to Predict the Normal Reactions of a Walking Robot. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2019, 23(4): 8-18 (In Russ.). <https://doi.org/10.21869/2223-1560-2019-23-4-8-18>.

Received 13.06.2019

Accepted 22.07.2019

Введение

Шагающие роботы представляют собой системы с дефицитом управляющих воздействий, а их динамика зависит, в том числе, от сил взаимодействия с опорной поверхностью. Это играет важную роль в схемах управления, разработанных для таких роботов [1]. Регуляторы для управления шагающими роботами можно разделить на две категории по используемым в них методам управления. К первой категории относятся регуляторы, которые построены на методах, явно учитывающих силы реакции при расчете желаемых управляющих воздействий [2]. Эти методы позволяют использовать полную динамическую модель робота, что повышает их точность. Будем называть методы этой категории явными.

Вторая категория включает в себя методы управления, которые не учитывают силы реакции или учитывают их неявно [3]. Методы этой категории будем называть неявными, их преимуществами является относительная простота и независимость от сил реакции, которые на практике могут быть достаточно трудно измеряемыми, особенно это касается сил трения.

Силы реакций связей полностью определяются динамикой робота и управляющими воздействиями, для их прогнозирования можно использовать аппроксимирующие функции. Один из эффективных способов реализации аппроксимирующих функций основан на

использовании нейронных сетей, что обусловлено высокоэффективными программными и аппаратными решениями, предназначенными для обучения нейронных сетей со значительным числом слоев. Именно поэтому в данной работе рассмотрены вопросы прогнозирования сил реакции на основе данных о текущем состоянии робота (скоростях и положениях звеньев робота) и выбранных управляющих воздействиях с использованием нейронной сети.

Материалы и методы

Модель шагающего робота

Рассмотрим шагающего робота, движущегося в сагиттальной плоскости при выполнении одного шага (рис. 1).

На рис. 1 точки O_i – шарниры робота; N_i – нормальные силы реакции, F_{fr} – силы трения. Динамика робота может быть описана следующей системой уравнений:

$$\begin{cases} \mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{B}\mathbf{u} + \mathbf{F}^T\boldsymbol{\lambda}, \\ \mathbf{F}\ddot{\mathbf{q}} + \dot{\mathbf{F}}\dot{\mathbf{q}} = \mathbf{0}, \end{cases} \quad (1)$$

где \mathbf{q} – вектор обобщенных координат; \mathbf{H} – обобщенная матрица инерции; \mathbf{c} – вектор обобщенных сил, создаваемых гравитационными и инерционными силами, а также трением в шарнирах; \mathbf{u} – вектор управляющих воздействий; \mathbf{B} – матрица моментов приводов; $\boldsymbol{\lambda}$ – вектор коэффициентов Лагранжа, соответствующий силам реакции; \mathbf{F} – матрица Якоби уравнений связи.

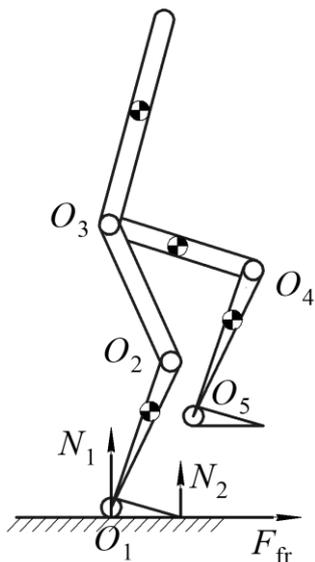


Рис. 1. Расчетная схема шагающего робота

Fig. 1. Settlement scheme for a walking machine

Уравнения (1) используются для моделирования динамики робота. Вывод уравнений (1), эффективные алгоритмы вычисления упомянутых матриц и векторов, а также другие детали, связанные с моделированием механических систем с явными связями, можно найти в работе [4]. Траектории робота, используемые в данной работе, не включают ударные взаимодействия, а также приобретение контакта с опорной поверхностью или отрыв звеньев робота от нее.

Будем использовать уравнения динамики, полученные из уравнений Лагранжа, и будем интегрировать их для заданного регулятора и входного управляющего сигнала. Также возможно использование программного обеспечения для моделирования абсолютно твердого тела для получения аналогичных результатов. В этом случае явная модель динамики робота не требуется.

Сбор данных для обучения нейронной сети

Нейронная сеть должна предсказывать значение вектора λ на основе текущих значений обобщенных координат \mathbf{q} , обобщенных скоростей $\dot{\mathbf{q}}$ и управляющих воздействий \mathbf{u} . Можно записать входной вектор \mathbf{y} для нейронной сети следующим образом:

$$\mathbf{y} = [\mathbf{q}^T \quad \dot{\mathbf{q}}^T \quad \mathbf{u}^T]. \quad (2)$$

Для хорошего обобщения информации нейронные сети должны получать различные данные, описывающие все аспекты и особенности поведения системы. Для сбора обучающей выборки производится моделирование выполнения роботом ряда различных движений, задаваемых желаемыми траекториями $\mathbf{q}_k^*(t)$, где k является индексом траектории. Робот использует CLQR регулятор для генерации управляющих воздействий \mathbf{u} . Алгоритм работы CLQR регулятора предложен в [5] и более детально изучен в [6].

$$\mathbf{u} = -\mathbf{K}_N \mathbf{N}^T (\mathbf{x} - \mathbf{x}^*) + \mathbf{u}^*, \quad (3)$$

где \mathbf{K}_N – матрица коэффициентов регулятора, получаемая путем решения уравнения Риккати для линеаризованной модели динамики робота; \mathbf{N} – матрица проекции в ядро линейного оператора матрицы связей; \mathbf{x} – вектор состояния, $\mathbf{x} = [\mathbf{q}^T \quad \dot{\mathbf{q}}^T]^T$; \mathbf{u}^* – вектор желаемых управляющих воздействий, полученный путем решения обратной задачи динамики.

Для заданных значений \mathbf{q} , $\dot{\mathbf{q}}$ и \mathbf{u} можно найти $\ddot{\mathbf{q}}$ и $\boldsymbol{\lambda}$, используя динамическую модель робота (1):

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{H} & -\mathbf{F}^T \\ \mathbf{F} & \mathbf{0} \end{bmatrix}^+ \begin{bmatrix} \mathbf{B}\mathbf{u} - \mathbf{c} \\ -\dot{\mathbf{F}}\dot{\mathbf{q}} \end{bmatrix}, \quad (4)$$

где $(\cdot)^+$ – псевдообратная матрица Мура-Пенроуза.

В целях предотвращения переобучения и повышения устойчивости предиктора к малым ошибкам входных данных можно использовать нелинейные модели датчика для генерации обучающей выборки. В данном исследовании рассмотрим следующую модель датчика:

$$\bar{\mathbf{q}} = s \cdot \left\lfloor \frac{\mathbf{q} + \boldsymbol{\eta}}{s} \right\rfloor, \quad (5)$$

где $\bar{\mathbf{q}}$ – измеренное значение \mathbf{q} ; s – разрешение датчика; $\boldsymbol{\eta}$ – случайный шум датчика, $\lfloor \cdot \rfloor$ – функция округления к меньшему целому числу.

Датчики скоростей $\dot{\mathbf{q}}$ и реакций $\boldsymbol{\lambda}$ имеют аналогичные модели. Схожие модели датчиков были рассмотрены в работах [7]. Здесь предполагается, что разрешение датчика равно $s = 10^{-2}$, а шум $\boldsymbol{\eta}$ равномерно распределяется в диапазоне $[-10^{-2} \quad 10^{-2}]$ с соответствующими единицами измерения.

Для данного робота векторы \mathbf{q} и $\dot{\mathbf{q}}$ содержат по 9 элементов каждый, вектор \mathbf{u} имеет 6 элементов и вектор $\boldsymbol{\lambda}$ состоит из 3 элементов. Таким образом, обучающая выборка может быть представлена в виде таблицы с 27 столбцами

и N строками. Будем использовать 100 численных экспериментов с различными желаемыми траекториями $\mathbf{q}_k^*(t)$, количество собранных данных составляет $N = 2 \cdot 10^5$.

Архитектура нейронной сети

Используемые здесь нейронные сети можно рассматривать как функцию:

$$\tilde{\boldsymbol{\lambda}} = \Phi(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}), \quad (6)$$

где $\tilde{\boldsymbol{\lambda}}$ – предсказанное значение $\boldsymbol{\lambda}$.

Для реализации предиктора используется архитектура из 5 полносвязных слоев, в том числе входного, выходного и трех скрытых слоев. Предлагаемая структура нейронной сети показана на рис. 2. На рис. 2 FC означает «полносвязный» и ReLU означает использование ReLU активационной функции [8]; количество нейронов в каждом слое, а также функция активации указаны в скобках. Такая архитектура позволяет сравнительно быстро производить обучение сети и получать необходимые прогнозы, что особенно важно ввиду того, что эти прогнозы учитываются регулятором робота как часть контура управления в режиме реального времени. Будем называть такую архитектуру нейронной сети главной.

Альтернативная архитектура, рассматриваемая в данном исследовании, включает в себя слой прореживания (dropout layer) после каждого полносвязного слоя. Использование слоев прореживания позволяет предотвратить переобучение [9].

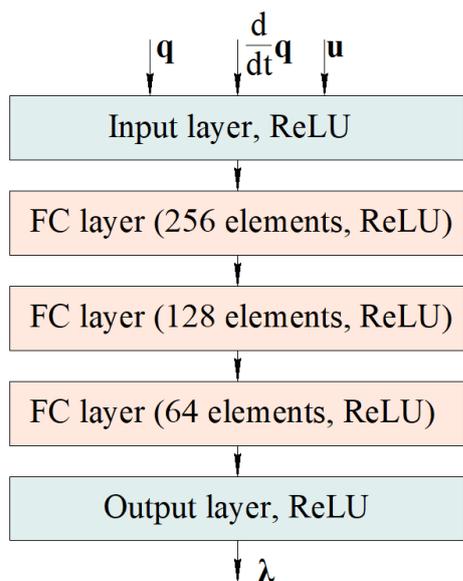


Рис. 2. Структура нейронной сети

Fig. 2. Neural network structure

Каждый из слоев прореживания имеет пропорцию прореживания 15%.

Результаты и их обсуждение

Реализация нейронной сети и её обучение

Предлагаемая нейронная сеть реализована с использованием TensorFlow – машинного обучения, которое позволяет эффективно использовать GPU (graphic processor units) для вычислений [10]. Система построена с применением библиотеки высокого уровня Keras. Обучение проводилось с помощью алгоритма оптимизации Adam, скорость обучения составила $5 \cdot 10^{-3}$ на обучающей выборке типа minibatch из 64 образцов. 85% данных в выборке использовались в процессе обучения, 15% были зарезервированы для проверки. Было проведено несколько сеансов обучения,

каждый из которых начинался с весовыми коэффициентами, инициализированными с помощью Glorot инициализатора [11]. Сеансы обучения были проведены отдельно для набора данных, генерируемых с использованием нелинейной модели датчика (5), и отдельно для набора данных, полученного в результате моделирования с идеальными сенсорами. В каждом случае точность нейронной сети составила более 98%, а значение целевой функции – меньше $1 \cdot 10^{-4}$ в течение первых 10 эпох. Это свидетельствует о том, что архитектура сети и доступные наборы данных позволяют производить обучение быстро, эффективно и надежно.

Результаты одного из сеансов обучения показаны на рис. 3, а. Обучение длилось в течение 150 эпох, скорость обучения была снижена до $2,5 \cdot 10^{-3}$ после первых 75 эпох, обучающая выборка содержит нелинейные модели датчиков.

По приведенным графикам видно, что нейронная сеть демонстрирует устойчивую сходимость на обучающей выборке (сплошная синяя линия) и умеренно колебательное поведение на верификационной выборке (черная пунктирная линия). В проводимых экспериментах сеть со слоями прореживания не имеет значительных преимуществ, что подтверждается графиками рис. 3, б. Это может быть обусловлено значительным числом данных в обучающей выборке и относительно короткими сеансами обучения.

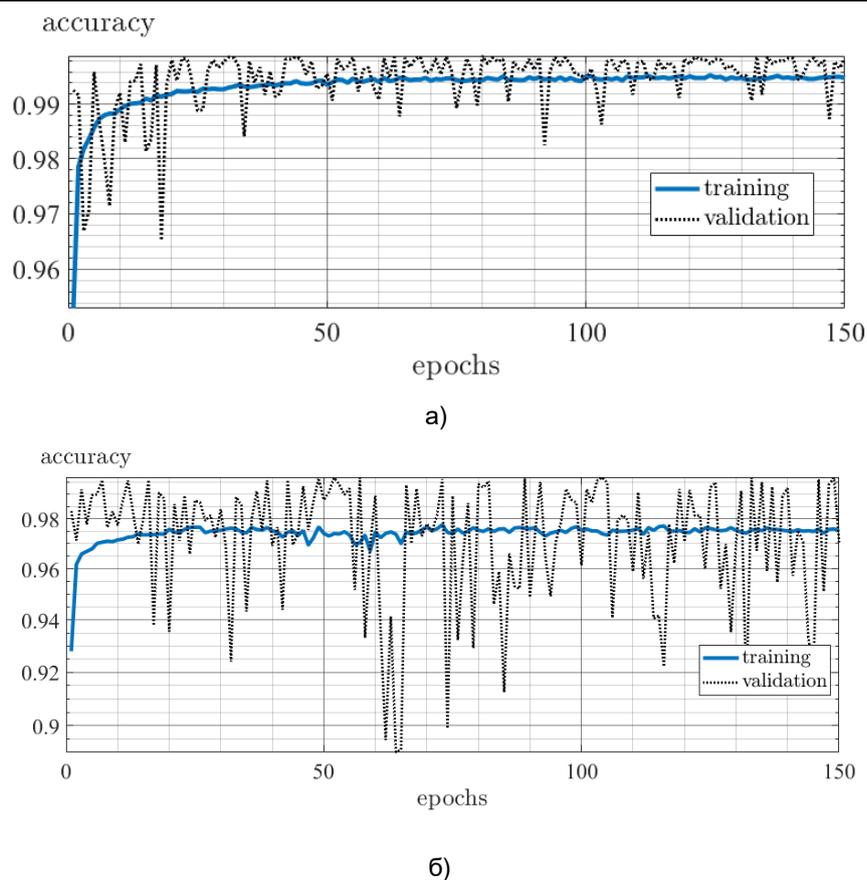


Рис. 3. Функция точности: а – для главной архитектуры сети; б – для сети с архитектурой прореживания

Fig. 3. Function of accuracy: a - main architecture network; b - network architecture thinning

Возможно, регуляризующий эффект слоев прореживания будет проявляться при более длительных сеансах обучения и разной пропорции прореживания [9]. Однако поскольку обучающая выборка генерируется в симуляторе, существует риск того, что при длительных сеансах обучения сеть будет адаптироваться к особенностям конкретного симулятора или модели. Вопросы достижения оптимальной производительности предиктора включают в себя такие моменты, как оптимальное генерирование и первичная обработка обучающей выборки.

Результаты моделирования работы сенсорной системы робота с обученной нейронной сетью

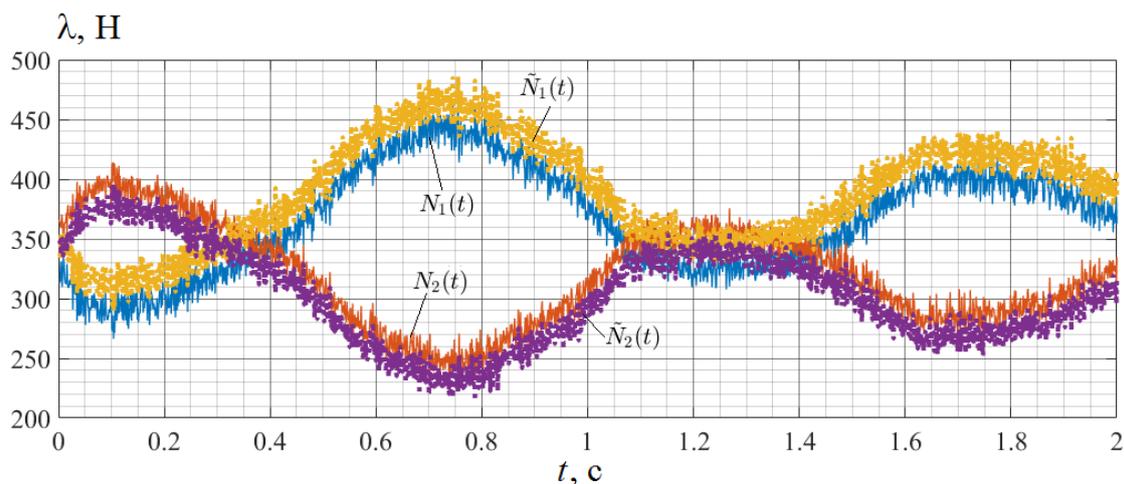
Для оценки качества обученной сети проведем численный эксперимент, в котором сеть будет использоваться совместно с моделью робота. Моделирование проводится с постоянным шагом по времени 1 мс. В этом эксперименте использована главная архитектура обучения в течение 150 эпох. На рис. 4 представлены графики фактических (N_1 , N_2) и предсказанных (\tilde{N}_1 , \tilde{N}_2) значений нормальных реакций, действующих на робота. На рис. 4, а пока-

зана вся продолжительность моделирования (2 с), а на рис. 4, б – только первые 0,08 с.

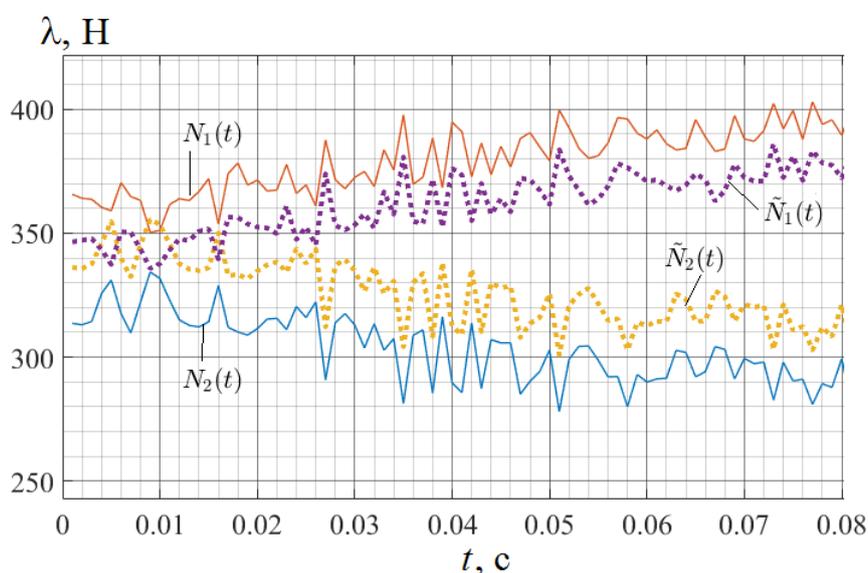
По графикам видно, что между фактическими и прогнозируемыми значениями нормальных реакций существует статическая ошибка. В противном случае при прогнозировании уде-

ся точно отслеживать фактические значения, полностью копируя форму исходного сигнала.

Графики рис. 4 иллюстрируют, что потеря приобретенных в процессе обучения данных происходит в основном из-за указанной выше статической ошибки предсказания.



а



б

Рис. 4. Графики временных зависимостей нормальных реакций при времени моделирования: а – 2 с; б – 0,08 с

Fig. 4. Time dependence graphs of normal reactions at simulation time: а – 2 s; б – 0,08 s

Эксперименты с другими сетями (с архитектурой прореживания и с главной архитектурой, но с более короткими сеансами обучения) показали, что точность обучения может варьироваться, но статическая ошибка остается доминирующим типом ошибки.

Выводы

В данной работе предложена полносвязная нейронная сеть для оценки сил нормальных реакций между поверхностью и стопами двуногого шагающего робота во время реализации им одного шага. Входными данными для сети являются состояние робота и требуемые управляющие воздействия, а выходными – предсказанные силы ре-

акции. Обучающая выборка данных генерируется с помощью моделирования полной динамической модели робота. Сеть построена и обучена с использованием библиотеки Keras и машинного обучения TensorFlow.

Проведено тестирование работы двух архитектур нейронной сети: со слоями прореживания и без них. Обе показали хорошие результаты работы, демонстрируя схождение результатов в процессе обучения. С помощью численного эксперимента установлено, что обученная сеть может точно прогнозировать силы реакции, однако при этом наблюдается статическая ошибка предсказания.

Список литературы

1. Werner A., Henze B., Rodriguez D.A., Gabaret J., Porges O., Roa M.A. Multi-contact planning and control for a torque-controlled humanoid robot // *Intelligent Robots and Systems (IROS): Proc. IEEE/RSJ Intern. Conf., Daejeon, South Korea. 2016. P. 5708-5715.*
2. Posa M., Cantu C., Tedrake R. A direct method for trajectory optimization of rigid bodies through contact // *Intern. J. of Robotics Research. 2014. Vol. 33(1). P. 69-81.*
3. Jatsun S., Savin S., Yatsun A. Parameter optimization for exoskeleton control system using sobol sequences // *Symposium on Robot Design, Dynamics and Control. Springer, Cham. 2016. P. 361-368.*
4. Featherstone R. *Rigid Body Dynamics Algorithms.* Boston, MA: Springer US, 2014. 271 p.
5. Mason S., Righetti L., Schaal S. Full dynamics LQR control of a humanoid robot: An experimental study on balancing and squatting // *Humanoid Robots: Proc. IEEE-RAS Intern. Conf., Madrid, Spain, 2014. P. 374-379.*
6. Savin S., Jatsun S., Vorocheva L. Modification of constrained LQR for control of walking in-pipe robots // *Dynamics of Systems, Mechanisms and Machines (Dynamics): Proc. IEEE Intern. Conf., Omsk, Russia, 2017. P. 1-6.*
7. Savin S., Jatsun S., Vorocheva L. State observer design for a walking in-pipe robot // *MATEC Web of Conferences: EDP Sciences. 2018. Vol. 161. P. 03012.*

8. Schmidhuber J. Deep learning in neural networks: An overview // *Neural Networks*. 2015. Vol. (61). P. 85–117.
9. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting // *J. of Machine Learning Research*. 2014. Vol. 15(1). P. 1929-1958.
10. Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., ... Kudlur M. Tensorflow: A system for large-scale machine learning // *Operating Systems Design and Implementation: Proc. 12th Symposium, Savannah, GA, USA, 2016*. P. 265-283.
11. Glorot X., Bengio Y. Understanding the difficulty of training deep feedforward neural networks // *Artificial Intelligence and Statistics: Proc. of the 13-th Intern. Conf., Scottsdale, AZ, USA, 2010*. P. 249-256.

References

1. Werner A., Henze B., Rodriguez D.A., Gabaret J., Porges O., Roa M.A. Multi-contact planning and control for a torque-controlled humanoid robot. *Intelligent Robots and Systems (IROS): Proc. IEEE/RSJ Intern. Conf., Daejeon, South Korea, 2016*, pp. 5708-5715.
2. Posa M., Cantu C., Tedrake R. A direct method for trajectory optimization of rigid bodies through contact. *Intern. J. of Robotics Research*, 2014, vol. 33(1), pp. 69-81.
3. Jatsun S., Savin S., Yatsun A. Parameter optimization for exoskeleton control system using sobol sequences. *Symposium on Robot Design, Dynamics and Control*. Springer, Cham, 2016, pp. 361-368.
4. Featherstone R. *Rigid Body Dynamics Algorithms*. Boston, MA: Springer US, 2014, 271 p.
5. Mason S., Righetti L., Schaal S. Full dynamics LQR control of a humanoid robot: An experimental study on balancing and squatting. *Humanoid Robots: Proc. IEEE-RAS Intern. Conf., Madrid, Spain, 2014*, pp. 374-379.
6. Savin S., Jatsun S., Vorochaeva L. Modification of constrained LQR for control of walking in-pipe robots. *Dynamics of Systems, Mechanisms and Machines (Dynamics): Proc. IEEE Intern. Conf., Omsk, Russia. 2017*, pp. 1-6.
7. Savin S., Jatsun S., Vorochaeva L. State observer design for a walking in-pipe robot. *MATEC Web of Conferences: EDP Sciences*. 2018, vol. 161, pp. 03012.
8. Schmidhuber J. Deep learning in neural networks: An overview. *Neural Networks*, 2015, vol. (61), pp. 85–117.
9. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J. of Machine Learning Research*, 2014, vol. 15(1), pp. 1929-1958.

10. Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., ... Kudlur M. Tensorflow: A system for large-scale machine learning. Operating Systems Design and Implementation: Proc. 12th Symposium, Savannah, GA, USA, 2016, pp. 265-283.

11. Glorot X., Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Artificial Intelligence and Statistics: Proc. of the 13-th Intern. Conf., Scottsdale, AZ, USA, 2010, pp. 249-256.

Информация об авторах / Information about the Authors

Савин Сергей Игоревич, кандидат технических наук, старший научный сотрудник лаборатории мехатроники, управления и прототипирования, Университет Иннополис, г. Иннополис, Российская Федерация, e-mail: s.savin@innopolis.ru

Sergey I. Savin, Candidate of Engineering Science, Higher Senior Officer, Laboratory of Mechatronics, Management and Prototyping, Innopolis University, Innopolis, Russian Federation
e-mail: s.savin@innopolis.ru

Ворочаева Людмила Юрьевна, кандидат технических наук, доцент кафедры механики, мехатроники и робототехники, ФГБОУ ВО «Юго-Западный государственный университет», г. Курск, Российская Федерация, e-mail: mila180888@yandex.ru

Lyudmila Yu. Vorochaeva, Candidate of Engineering Science, Associate Professor, Department of Mechanics, Mechatronics and Robotics, Southwest State University, Kursk, Russian Federation
e-mail: mila180888@yandex.ru