

УДК 621.873

## Возможности использования библиотеки проектирования нейронных сетей TensorFlow для кинематического исследования манипулятора

Андрей Бойченко<sup>1</sup>, Дмитрий Мищук<sup>2</sup>

Киевский национальный университет строительства и архитектуры,  
03037, Воздухофлотский проспект 31, Киев, Украина,

<sup>1</sup>[andriy@i.ua](mailto:andriy@i.ua), <https://orcid.org/0000-0000-0000-0000>,

<sup>2</sup>[mischuk.do@knuba.edu.ua](mailto:mischuk.do@knuba.edu.ua), <https://orcid.org/0000-0002-8263-9400>

Received: 29.04.2020; Accepted: 29.05.2020

<https://doi.org/10.32347/gbdmm2020.95.0201>

**Аннотация.** Исследование кинематических структур многозвенных манипуляторов является нетривиальной задачей. Сложности проявляются при исследовании кинематических схем у которых количество подвижных независимых частей превышает количество обобщенных независимых степеней подвижности, так как в таких случаях имеет место переизбыток подвижности системы и для одной позиции может существовать больше двух конфигураций манипулятора. Подобная неоднозначность сильно усложняет поиск зависимостей между обобщенными координатами системы и конечной точкой перемещения рабочего органа. Также необходимо учитывать, что в дальнейшем при составлении динамических уравнений движений для подобных систем, такая неоднозначность усложняет исследование задач динамики.

Решения подобных проблем осуществляется разными методами и в данной статье рассматривается возможность применения нейронных сетей для исследований подобных неоднозначных задач на примере кинематического исследования шарнирно-сочлененного манипулятора.

В представленной работе рассмотрены возможности использования библиотеки TensorFlow от компании Google при помощи которой создаются модели нейронных сетей и «нейронов», а также функции нахождения весовых коэффициентов настройки созданной модели сети. В ходе исследования была создана нейронная сеть для расчета функции регрессии зависимостей между геометрическими координатами точки перемещения захватного устройства манипулятора и обобщенными координатами его стреловой системы.

Целью данного исследования является демонстрация возможностей нейронных сетей при решении инженерных задач, которые требуют сложных математических преобразований.

Использование архитектур нейронных сетей в системах управления манипуляторами позволяет создавать универсальные механизмы для выполнения различных технологических процедур, при этом снижая затраты на разработку подобных решений.

В целом нейронные сети не являются универсальными средствами для решения всех инженерных задач, так как требуют большего набора тестовых выборок для настройки (обучения) параметров своих моделей, однако существует ряд проблем при решении которых нейронные сети имеют большие преимущества.

**Ключевые слова:** TensorFlow, манипулятор, система управления, нейронная сеть.

### ПОСТАНОВКА ПРОБЛЕМЫ

Способы эффективного определения величин перемещения обобщенных координат манипулятора в зависимости от построенной траектории движения его рабочего органа является одной из ключевых задач современной робототехники [1-4].

Наиболее распространёнными способами решения подобных задач являются классические и традиционные методы аналитического преобразования координат исходной точки рабочего органа манипулятора в величины перемещения органов исполнительных механизмов [5-8]. Однако

такие методы имеют существенный недостаток, который связан со значительными вычислительными преобразованиями. При этом увеличение количества подвижных независимых координат приводит к появлению переизбытка возможных конфигураций манипулятора. Таким образом при одних и тех же координатах положения рабочего органа манипулятора, звенья манипулятора будут занимать несколько разрешенных положений [9-12].

Очень часто в таких случаях применяют подход, когда поочередно фиксируют несколько координат, оставляя достаточное их количество для перемещения рабочего органа манипулятора [13-15]. Решение о том какие координаты нужно зафиксировать в таком случае принимается на основании результатов исследований за разными эвристическими методами [6-8, 16, 17]. Также для решения подобных неоднозначностей используют методы оптимизации, при помощи которых удается оптимально распределять подвижность обобщенных координат манипулятора [18-20]. Хотя такие подходы довольно точны, но они сложны в реализации, так как требуют задействования сложных математических преобразований.

Одним из современных подходов проектирования систем управления роботами, которые позволяют быстро рассчитывать траектории их движения является использование нейронных сетей [21-23].

Нейронные сети эффективно применяют для решения задач планирования траектории рабочего органа робота-манипулятора, который работает со сложной кинематической структурой [24]. Нейронные системы могут учиться как на моделях, так и во время функционирования робота.

Одним из подходов построения и программирования нейронных сетей является использование стандартных библиотек [25]. Наиболее популярной является библиотека TensorFlow от компании Google [26]. В основном при помощи этой библиотеки проектируют сети для распознавания изображений, звуков и т.п.

В данной статье приводятся результаты исследований применения TensorFlow для

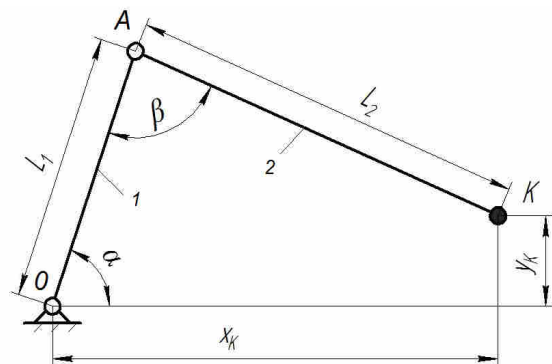
решения прикладной задачи механики на примере расчета регрессии применительно к обратной кинематической задаче.

## ЦЕЛЬ СТАТЬИ

Исследовать возможность применения библиотеки TensorFlow для решения задачи регрессии при кинематическом описании многозвенных манипуляционных систем.

## ОПИСАНИЕ ПРЕДМЕТА ИССЛЕДОВАНИЙ

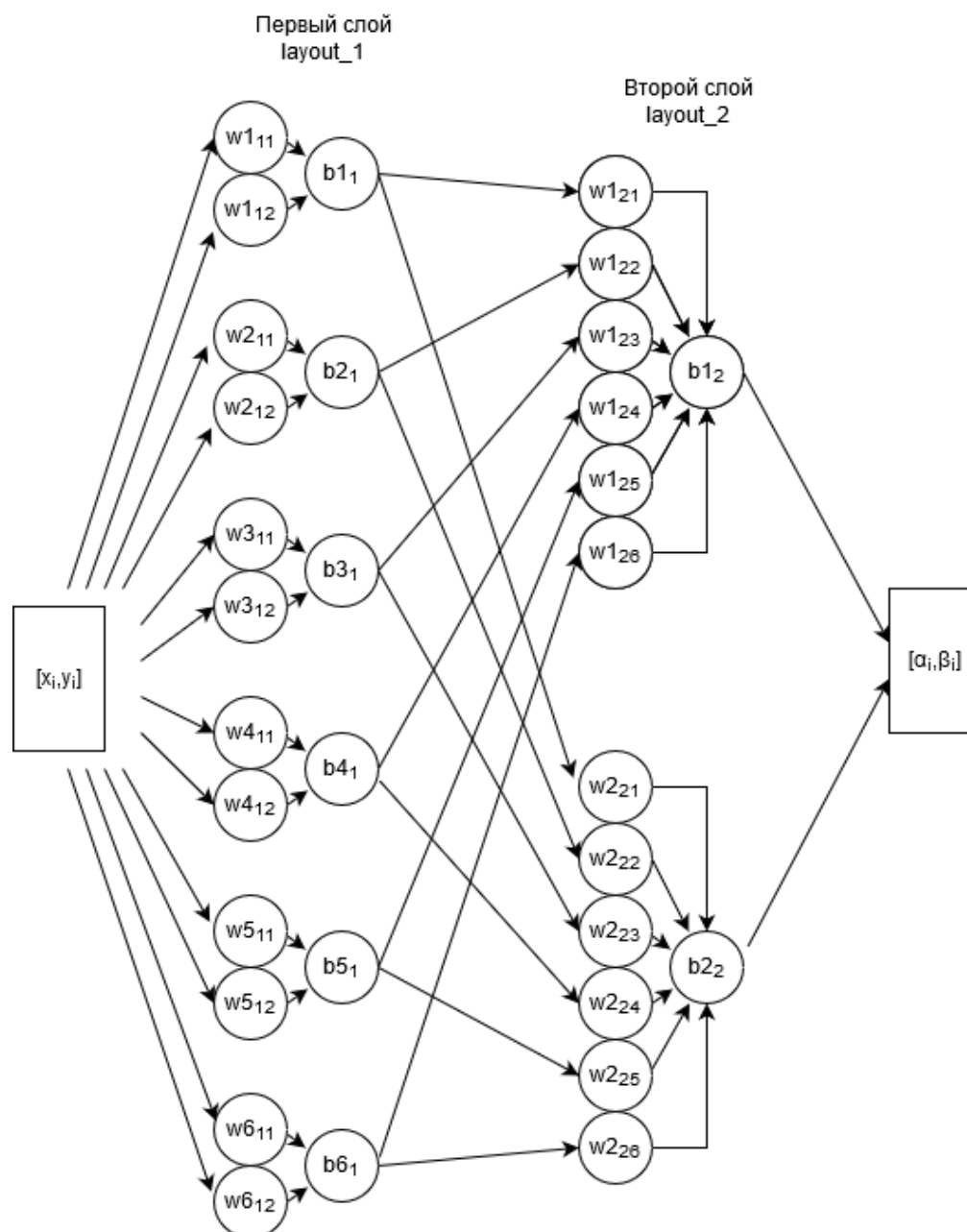
Ранее [21] была рассмотрена возможность применения нейронной сети для построения простой системы управления манипулятором шарнирно-сочлененная стреловая система которого представлена на Рис. 1.



**Рис. 1.** Кинематическая схема двухзвенного манипулятора: 1, 2 – звенья стрелы;  $\alpha$ ,  $\beta$  – углы поворота;  $x_K$ ,  $y_K$  – координаты захвата

**Fig. 1.** The kinematic scheme of the two-arm manipulator: 1, 2 – boom links;  $\alpha$ ,  $\beta$  – angles of rotation;  $x_K$ ,  $y_K$  – capture coordinates

Рассмотрим пример построения нейронной сети, которая состоит из шести нейронов первого слоя и двух нейронов второго слоя, при этом в TensorFlow для такой реализации используется полносвязный нейронный граф – то есть все нейроны первого слоя последовательно связаны со всеми входными параметрами, а все нейроны последующих слоев связаны со всеми выходами нейронов приводящего слоя. Архитектура такой модели сети подана на Рис. 2.



**Рис. 2.** Архитектура исследуемой нейронной сети в системе TensorFlow

**Fig. 2.** The architecture of the studied neural network into TensorFlow

В представленной модели принято, что  $w_{1_{1j}} \dots w_{6_{1j}}$  и  $w_{1_{2j}} \dots w_{2_{2j}}$  – веса нейронов первого и второго слоев,  $b_{1_1} \dots b_{6_1}$  и  $b_{1_2} \dots b_{2_2}$  – коэффициенты смещения нейронов соответственно первого и второго слоев. Реализацию представления рассмотренной модели выполнено с использованием инструмента Keras для TensorFlow [27], который упрощает взаимодействия с библиотекой за счет контейнеризации данных и яв-

ляется своеобразной прослойкой для взаимодействия с пользователем.

## ОСНОВНЫЕ РЕЗУЛЬТАТЫ

На первом этапе исследований был рассмотрен процесс моделирования нейросети для управления двухзвенным манипулятором на примере решения геометрической задачи о положениях. В работе [21] приведено вариант поиска решения подобной

ISSN(online) 2709-6149. Mining, constructional, road and melioration machines, 95, 2020, 14-24

задачи при помощи нейронной сети с тестовым набор входных данных в виде координат перемещения груза:

$$\begin{aligned}
 x_1 &= 1,030 \text{ м}; y_1 = -0,347 \text{ м}; \\
 x_2 &= 1,109 \text{ м}; y_2 = -0,423 \text{ м}; \\
 x_3 &= 1,222 \text{ м}; y_3 = -0,479 \text{ м}; \\
 x_4 &= 1,366 \text{ м}; y_4 = -0,509 \text{ м}; \\
 x_5 &= 1,534 \text{ м}; y_5 = -0,506 \text{ м}; \\
 x_6 &= 1,719 \text{ м}; y_6 = -0,464 \text{ м}; \\
 x_7 &= 1,914 \text{ м}; y_7 = -0,379 \text{ м}; \\
 x_8 &= 2,110 \text{ м}; y_8 = -0,249 \text{ м}; \\
 x_9 &= 2,298 \text{ м}; y_9 = -0,072 \text{ м}; \\
 x_{10} &= 2,469 \text{ м}; y_{10} = 0,152 \text{ м}; \\
 x_{11} &= 2,612 \text{ м}; y_{11} = 0,419 \text{ м}; \\
 x_{12} &= 2,721 \text{ м}; y_{12} = 0,725 \text{ м}; \\
 x_{13} &= 2,786 \text{ м}; y_{13} = 1,066 \text{ м}; \\
 x_{14} &= 2,800 \text{ м}; y_{14} = 1,433 \text{ м}; \\
 x_{15} &= 2,758 \text{ м}; y_{15} = 1,819 \text{ м}; \\
 x_{16} &= 2,656 \text{ м}; y_{16} = 2,214 \text{ м}; \\
 x_{17} &= 2,491 \text{ м}; y_{17} = 2,607 \text{ м}; \\
 x_{18} &= 2,261 \text{ м}; y_{18} = 2,989 \text{ м}.
 \end{aligned}
 \tag{1}$$

В качестве выходных параметров модели приняты координаты углов поворота  $\alpha$  и  $\beta$  (в градусах) звеньев шарнирно-сочлененной стрелы манипулятора, которые также являются ее обобщенными координатами. Каждая пара приведенных угловых координат является отображением соответствующих входных параметров (1):

$$\begin{aligned}
 \alpha_1 &= 0^\circ; \beta_1 = 10^\circ; \\
 \alpha_2 &= 5^\circ; \beta_2 = 15^\circ; \\
 \alpha_3 &= 10^\circ; \beta_3 = 20^\circ; \\
 \alpha_4 &= 15^\circ; \beta_4 = 25^\circ; \\
 \alpha_5 &= 20^\circ; \beta_5 = 30^\circ; \\
 \alpha_6 &= 25^\circ; \beta_6 = 35^\circ; \\
 \alpha_7 &= 30^\circ; \beta_7 = 40^\circ; \\
 \alpha_8 &= 35^\circ; \beta_8 = 45^\circ; \\
 \alpha_9 &= 40^\circ; \beta_9 = 50^\circ; \\
 \alpha_{10} &= 45^\circ; \beta_{10} = 55^\circ; \\
 \alpha_{11} &= 50^\circ; \beta_{11} = 60^\circ; \\
 \alpha_{12} &= 55^\circ; \beta_{12} = 65^\circ; \\
 \alpha_{13} &= 60^\circ; \beta_{13} = 70^\circ; \\
 \alpha_{14} &= 65^\circ; \beta_{14} = 75^\circ; \\
 \alpha_{15} &= 70^\circ; \beta_{15} = 80^\circ; \\
 \alpha_{16} &= 75^\circ; \beta_{16} = 85^\circ; \\
 \alpha_{17} &= 80^\circ; \beta_{17} = 90^\circ; \\
 \alpha_{18} &= 85^\circ; \beta_{18} = 95^\circ.
 \end{aligned}
 \tag{2}$$

Представим входной и выходной наборы параметров в виде матриц:

$$[x, y] = \begin{bmatrix} [x_1, y_1] \\ [x_2, y_2] \\ \dots \\ [x_i, y_i] \\ \dots \\ [x_n, y_n] \end{bmatrix};$$

$$[\alpha, \beta] = \begin{bmatrix} [\alpha_1, \beta_1] \\ [\alpha_2, \beta_2] \\ \dots \\ [\alpha_i, \beta_i] \\ \dots \\ [\alpha_n, \beta_n] \end{bmatrix},$$

где  $n$  – количество точек тестовой выборки.

В библиотеке TensorFlow основной единицей с которой приходится работать является «тензор» который по сути представляет из себя матрицу. Для создания тензора в данном примере дополнительно использована библиотека Numpy, при помощи которой создаются матрицы:

```

from __future__ import absolute_import,
division, print_function, unicode_literals
import tensorflow as tf
import numpy as np

coordinatu_gruza = np.array
([ [x1, y1], [x2, y2], ..., [x18, y18]],
dtype=float);

uglu_povorota = np.array
([ [alpha1, beta1], [alpha2, beta2], ..., [alpha18, beta18]],
dtype=float),

```

где *coordinatu\_gruza* – тензор отвечающий за входной набор координат груза; *uglu\_povorota* – тензор отвечающий за выходной набор координат углов поворота звеньев манипулятора.

Для построения модели сети в TensorFlow необходимо задать количество нейронов в каждом из проектированных слоев, а также количество таких слоев. Входной

слой также необходимо пометить, чтобы в дальнейшем программа системы расчета понимала, где нужно искать вход:

```
layout_1 = tf.keras.layers.Dense
(units=6, input_shape=[2]);
layout_2 = tf.keras.layers.Dense
(units=2);
```

(4)

```
model = tf.keras.Sequential
([layout_1, layout_2]),
```

где *layout\_1* и *layout\_2* – переменные, которые соответствуют описанию слоев нейронов; *model* – переменная, в которую записывается соответствующее описание модели создаваемой нейронной сети; *units* – количество нейронов в каждом из слоев; *input\_shape* – пометка входного слоя, которая в данном случае обозначает привязку к квадратной матрице размерности 2.

Для настройки модели использован стандартный алгоритм среднеквадратической ошибки и функция оптимизации «Адам» (Adam), которая предоставляется библиотекой TensorFlow:

```
model.compile(loss =
'mean_squared_error', optimizer=tf.k
eras.optimizers.Adam(0.1)),
```

(5)

где *model.compile* – команда TensorFlow для сборки функции настройки нейронной сети; *'mean\_squared\_error'* (функция среднеквадратической ошибки) – название применяемой стандартной функции для поиска ошибки настройки сети нейронов; *optimizer* – переменная отвечающая за создание функции оптимизации (в данном примере использована стандартная функция градиентного спуска *tf.keras.optimizers.Adam(0.1)* библиотеки TensorFlow).

Для обучения созданной модели сети на ее вход необходимо подать тестовый набор тренировочных данных, а выход подключить к ожидаемому параметру системы:

```
model.fit(coordinatu_gruza,
uglu_povorota, epochs=500,
verbose=False),
```

(6)

где *epochs* – количество итераций.

Выполнив команду *model.fit* созданная нейронная сеть автоматически рассчитывает параметры своих нейронов, после чего можно проверить нейронную сеть на точность, задав функцию *model.predict* которой в качестве входных данных нужно передать пару заданных координат точки подвеса груза. Результатом выполнения будет матрица с координатами, которые отвечают углам поворота звеньев манипулятора, например:

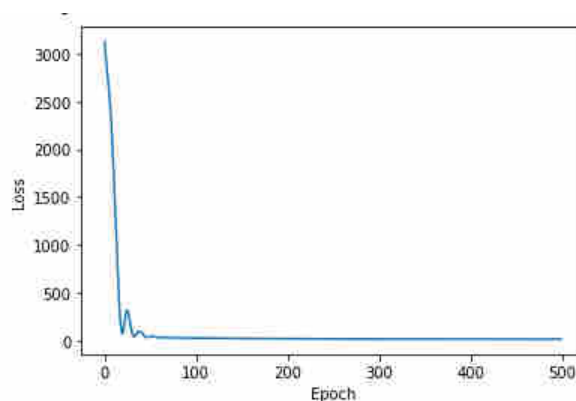
```

      функция печати      функция про-
      ↓                  ↙
print(model.predict(
[[1.030, -0.347], [2.469, 0.152]])),
```

(7)

↑  
матрица входных данных в  
виде:  $\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$ .

На Рис. 3 приведен график демонстрирующий обучение нейронной сети при количестве итераций 500. График показывает зависимость уменьшения ошибки, которая определяется как разность между заданными и прогнозируемыми координатами выхода. Из представленного графика видно, что для данного случая достаточно не более 100-200 итераций.



**Рис. 3.** График обучения нейронной сети: Epoch – количество итераций при обучении; Loss – значение ошибки

**Fig. 3.** Schedule of neural network training: Epoch – the number of iterations during training; Loss – error value

Метод `get_weights()`, который применяется для конкретного слоя нейронов позволяет получить выборку коэффициентов настройки нейронной сети. Для примера при прогнозировании параметров (2) и входных данных (1) коэффициенты настройки для каждого слоя будут принимать следующие значения:

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \\ w_{51} & w_{52} \\ w_{61} & w_{62} \end{bmatrix}^T = \begin{bmatrix} 1,9617 & -1,6782 \\ 3,4183 & -1,2148 \\ 2,7383 & 0,0268 \\ -2,732 & -0,5495 \\ -1,586 & -0,6424 \\ -2,140 & -0,3908 \end{bmatrix}^T ;$$

$$\begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \\ b_{51} \\ b_{61} \end{bmatrix}^T = \begin{bmatrix} -4,9281 \\ -0,5325 \\ -0,17725 \\ 0,66964 \\ 0,27901 \\ 3,24579 \end{bmatrix}^T ; \quad (8)$$

$$\begin{bmatrix} w_{121} & w_{221} \\ w_{122} & w_{222} \\ w_{123} & w_{223} \\ w_{124} & w_{224} \\ w_{125} & w_{225} \\ w_{126} & w_{226} \end{bmatrix} = \begin{bmatrix} 4,1111 & 2,7545 \\ 1,6010 & 2,5459 \\ 1,5047 & 2,5431 \\ -2,2437 & -1,823 \\ -0,8949 & -0,767 \\ -3,2490 & -2,1937 \end{bmatrix} ;$$

$$[b_{12} \quad b_{22}] = [0,5203 \quad 1,2381].$$

## АНАЛИЗ РЕЗУЛЬТАТОВ

Для качественной оценки полученных результатов, которые получено из нейронной сети при прогнозировании углов поворота звеньев двухзвенного манипулятора, было проведено ряд экспериментов в ходе, которых ставилась задача оценки стойкости системы на потерю части входных данных. За базовый вариант было принято данные (1) и смоделировано случаи 70%, 50% и 30% заполнения входных данных при их

равномерном распределении (см. Таблицу). Для каждого из вариантов рассчитано среднеквадратическую ошибку прогнозируемых данных от реальных, которая соответственно для полной 100% выборки составила  $3,61^\circ$  – по  $\alpha$  по  $\beta$ ; для 70% выборки:  $3,62^\circ$  – по  $\alpha$  и  $\beta$ ; для 50% выборки  $3,69^\circ$  – по  $\alpha$  и  $4,1^\circ$  – по  $\beta$ ; для 30% выборки  $4,02^\circ$  – по  $\alpha$  по  $\beta$ . Таким образом при потере до 70% данных и равномерном их распределении рассмотренная нейронная сеть теряет в точности до 11%. Также из полученных данных наблюдаются значительные отклонения в начале и конце выборки.

Проведенные многократные эксперименты также показали, что данная модель при одних и тех входных и выходных данных может по-разному рассчитывать коэффициенты настройки нейронов и тем самым практически не возможно построить точную систему прогноза, что скорее всего связано с несовершенством функции оптимизации и функции активации нейронов.

На Рис. 4 и Рис. 5 построены графики сравнения для разных случаев заполнения модели выборками обучения.

## ВЫВОДЫ

В ходе проведения исследований библиотеки TensorFlow удалось построить нейронную сеть, состоящую из двух слоев с шестью нейронами в первом слое и двумя – во втором. TensorFlow позволяет легко и быстро произвести настройку полученной модели сети что дает возможности проводить экспресс анализ манипуляционной системы за кинематическими параметрами, однако для точной работы необходимо более детальное изучение данной системы.

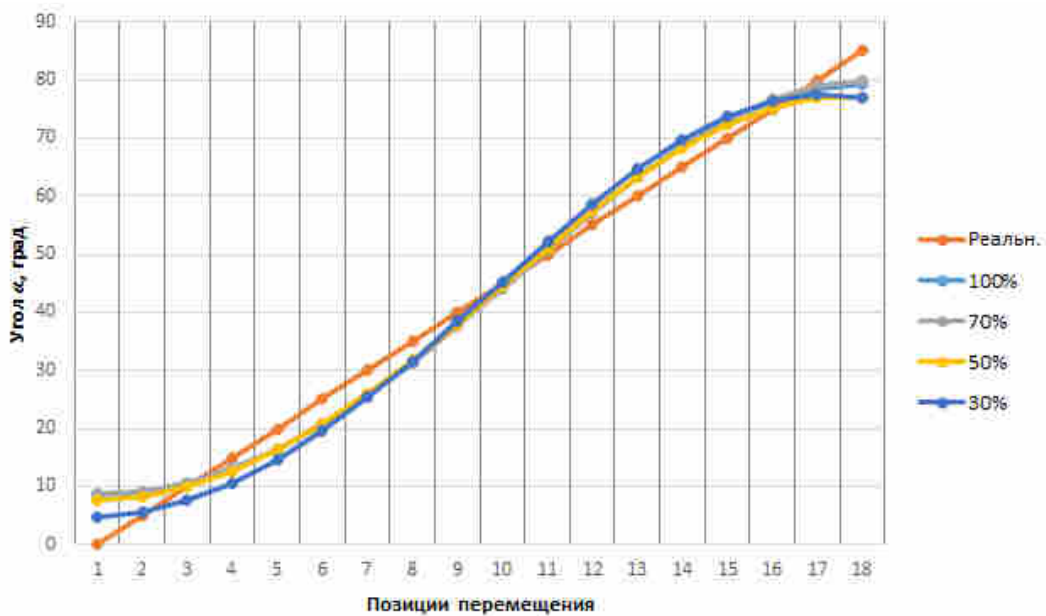
При проведении экспериментов в TensorFlow с принятой моделью сети при повышении точности расчета и количества итераций значительного эффекта со стандартными функциями оптимизации и активации нейронов не наблюдалось.

Подобный подход моделирования применим также в системах диагностики [28] и исследования [29] различных механических систем.

**Таблица.** Значение прогнозируемых параметров  $\alpha$  и  $\beta$  в градусах

**Table.** The value of the predicted parameters  $\alpha$  and  $\beta$  in degrees

№ п.п.	Прогноз для полной 100% выборки		Прогноз для 70% выборки		Прогноз для 50% выборки		Прогноз для 30% выборки	
	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$
1	8,54	18,18	8,72	18,05	7,64	15,46	4,69	14,68
2	9,08	18,77	9,21	18,63	8,33	16,46	5,59	15,58
3	10,58	20,33	10,65	20,18	10,00	18,52	7,55	17,54
4	13,06	22,89	13,07	22,72	12,65	21,62	10,58	20,58
5	16,48	26,38	16,42	26,21	16,25	25,68	14,61	24,61
6	20,77	30,75	20,66	30,58	20,71	30,61	19,55	29,55
7	25,86	35,92	25,70	35,75	25,94	36,29	25,29	35,29
8	31,59	41,71	31,38	41,56	31,78	42,51	31,63	41,63
9	37,80	47,98	37,57	47,85	38,05	49,10	38,38	48,39
10	44,31	54,54	44,09	54,44	44,57	55,83	45,34	55,35
11	50,86	61,10	50,65	61,05	51,06	62,39	52,19	62,20
12	57,26	67,49	57,10	67,49	57,33	68,59	58,74	68,75
13	63,25	73,46	63,16	73,51	63,13	74,15	64,70	74,70
14	68,58	78,73	68,59	78,85	68,18	78,79	69,77	79,78
15	73,05	83,10	73,18	83,29	72,29	82,31	73,77	83,77
16	76,44	86,37	76,71	86,63	75,25	84,50	76,45	86,45
17	78,53	88,31	78,98	88,65	76,85	85,15	77,62	87,61
18	79,16	88,76	79,81	89,18	76,94	84,11	77,09	87,08



**Рис. 4.** Графики сравнения данных для угла поворота  $\alpha$

**Fig. 4.** Graphs of data comparison for the angle of rotation  $\alpha$

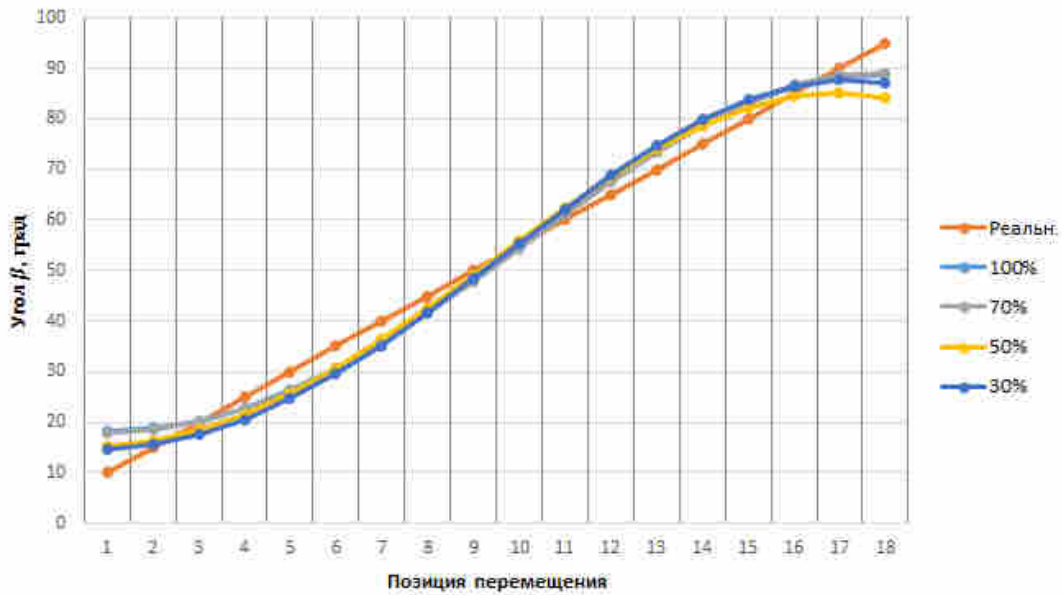


Рис. 5. Графики сравнения данных для угла поворота  $\beta$

Fig. 5. Graphs of data comparison for the angle of rotation  $\beta$

#### ЛІТЕРАТУРА

- Щербаков В. С., Реброва И. А. Планирование траектории рабочего органа строительного манипулятора в автоматическом режиме // В. С. Щербаков, И. А. Реброва / Омский научный вестник. – 2006. – №7. – С. 107-109.
- Пелевін Л. Є., Почка К. І. та ін. Синтез робототехнічних систем в машинобудуванні: Підручник // Л. Є. Пелевін, К. І. Почка, Д. О. Міщук, О. М. Гаркавенко, І. В. Русан. – К.: Київ, – 2016, – ТОВ "НВП "Інтерсервіс" – 258 с.
- Wu, Q. C., Wang, X. S., Chen, B., & Wu, H. T. Development of an RBFN-based neural-fuzzy adaptive control strategy for an upper limb rehabilitation exoskeleton. *Mechatronics*, - 2018. – 53. – P. 85-94. <https://doi.org/10.1016/20j.mechatronics.2018.05.014>.
- Міщук Д. О. Розробка системи керування будівельного робота із застосуванням технологій «Microsoft robotic studio». II Міжнародна науково-технічна конференція «Енергоощадні машини і технології», Київ, КНУБА, – 29 вересня - 1 жовтня. – 2015. – С. 47.
- Міщук Д. О. Кінематичний аналіз маніпуляційної системи будівельного робота // Д. О. Міщук / Вестник Харьковского национального автомобильно-дорожного университета. – 2014. – 65-66. – С. 90-96.
- Зенкевич С. Л. Управление роботами. Основы управления манипуляционными роботами: Учеб. Для вузов / С. Л. Зенкевич, А. С. Ющенко. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2000. – 400 с.
- Karlik B., Aydin S. An improved approach to the solution of inverse kinematics problems for robot manipulators // *Engineering applications of artificial intelligence*. – 2000. – Vol. 13. – No. 2. – P. 159-164.
- Спыну Г. А. Промышленные роботы. Конструирование и применение: Учеб. Пособие. – 2е изд., перераб. и доп. / Г. А. Спыну. – К.: Выща шк., 1991. – 331 с.
- Feng Y., Yaonan W., Yimin Y. Inverse kinematics solution for robot manipulator based on neural network under joint subspace // *International Journal of Computers Communications & Control*. – 2014. – Vol. 7. – No. 3. – P. 459-472.
- Ловейкін В. С. Оптимізація режимів зміни вильоту маніпулятора з гідроприводом. Монографія / В. С. Ловейкін, Д. О. Міщук. – К.: ЦП «Компринт», – 2013. – 206 с.
- Ловейкін, В. С., Міщук, Д. О. Математичне моделювання зміни вильоту вантажу маніпулятором з гідроприводом // В.С. Ловейкін, Д.О. Міщук / Гірничі, будівельні, дорожні і меліоративні машини, – 2012, – 79. – С. 9-15.
- Міщук Д., Горбатюк Є., Воляннюк В. Визначення геометричних параметрів маніпу-



- лятора за характеристиками робочого середовища / Д. Міщук, Є. Горбатюк, В. Воляннюк // Гірничі, будівельні, дорожні і меліоративні машини, – 2019, – 94. – С. 25-34.
13. **Momani S., Abo-Hammour Z. S., Alsmadi O. M. K.** Solution of inverse kinematics problem using genetic algorithms // *Applied Mathematics & Information Sciences*. – 2016. – Т. 10. – No. 1. – P. 225.
  14. **LaValle S. M.** Planning algorithms. – Cambridge university press / S. M. LaValle – 2006. – P. 1007.
  15. **Stilman M. et al.** Manipulation planning among movable obstacles / M. Stilman // *Robotics and Automation 2007 IEEE International Conference on*. – IEEE, 2007. – P. 3327-3332.
  16. **Hirano Y., Kitahama K., Yoshizawa S.** Image-based object recognition and dexterous hand/arm motion planning using rrt\* for grasping in cluttered scene // *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. – IEEE, 2005. – P. 2041-2046.
  17. **Graca Marcos M., Machado J. A. T., Azevedo-Perdicoulis T. P.** Trajectory planning of redundant manipulators using genetic algorithms // *Communications in nonlinear science and numerical simulation*. – 2009. – Vol. 14. – No. 7. – P. 2858-2869.
  18. **Mischuk D. O.** Increasing efficiency manipulator by optimal control // Д. О. Міщук / Гірничі, будівельні, дорожні і меліоративні машини, – 2015. – No. 85, 43-50. – (in Ukrainian).
  19. **Ловейкін В.С., Міщук Д.А.** Синтез оптимального динамічного режиму руху стрели манипулятора, встановленого на упругому основанні. Наука і техніка. – 2019, – 18(1). – С. 55-61. <https://doi.org/10.21122/2227-1031-2019-18-1-55-61>.
  20. **Ловейкін В. С.** Визначення оптимальних режимів руху манипулятора за процес пуску (гальмування) під час роботи за однією з узагальнених координат / В. С. Ловейкін, Д. О. Міщук // Гірничі, будівельні, дорожні та меліоративні машини. – 2009. – № 73. – С. 35-42.
  21. **Міщук, Д., Бойченко А.** Розробка концепції системи керування роботом для шпукатурних робіт на основі нейронної мережі / Д. Міщук, А. Бойченко // Гірничі, будівельні, дорожні та меліоративні машини. – 2019. – № 93. – С. 46-60. <https://doi.org/10.32347/gbdmm2019.93.0501>.
  22. **Bingul Z., Ertunc H. M., Oysu C.** Applying neural network to inverse kinematic problem for 6R robot manipulator with offset wrist // *Adaptive and Natural Computing Algorithms*. – Springer, Vienna, – 2005. – P. 112-115.
  23. **Köker R., Çakar T., Sari Y.** A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators // *Engineering with Computers*. – 2014. – Vol. 30. – No. 4. – P. 641-649.
  24. **Міщук Д.** Модульний принцип побудови будівельного робота / Д. Міщук, В. Воляннюк, Є. Горбатюк. // Гірничі, будівельні, дорожні та меліоративні машини. – 2017. – №89. – С. 90-97. <http://gbdmm.knuba.edu.ua/article/view/114398>.
  25. **Дорогов А. Ю.** Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей / А. Ю. Дорогов. – 2014. – Политехника. – 328 с.
  26. **Жерон О.** Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем / О. Жерон. - Диалектика-Вильямс. – 2020. – 688 с.
  27. **TensorFlow Core for beginners. Keras basics.** <https://www.tensorflow.org/tutorials?hl=en>.
  28. **Комоцька С., Бігун М.** Випробування пневмоциліндрів з використанням реле часу / С. Комоцька, М. Бігун // Гірничі, будівельні, дорожні та меліоративні машини. – 2019. – № 94. – С. 51-62. <https://doi.org/10.32347/gbdmm2019.94.0501>
  29. **Сукач М. К.** Лабораторные исследования скобовидных пластинчатых рессор // М. К. Сукач / Гірничі, будівельні, дорожні та меліоративні машини, – 2019, – Вип.94. – С. 16-24. <https://doi.org/10.32347/gbdmm2019.94.0102>.

## REFERENCES

1. **Sherbakov V. S., Rebrova I. A.** (2006). Plannirovanie traektorii rabocheho organa stroitel'nogo manipulyatora v avtomaticheskom rezhime [Planning the trajectory of the working body of the construction manipulator in automatic mode]. *Omskij nauchnyj vestnik*, Nr.7, 107-109. – (in Russian).
2. **Pelevin L. Ye., Pochka K. I., Mischuk D. O., Garkavenko O. M., Rusan I. V.** (2016). Sintez robototekhnichnih sistem v mashinobuduvanni: Pidruchnik [Synthesis of robotic systems in mechanical engineering: Textbook]. Kyiv, "NVP "Interservis"" Publ., 258. – (in Ukrainian).

3. **Wu, Q. C., Wang, X. S., Chen, B., & Wu, H. T.** (2018). Development of an RBFN-based neural-fuzzy adaptive control strategy for an upper limb rehabilitation exoskeleton. *Mechatronics*, 53, 85-94. <https://doi.org/10.1016/j.mechatronics.2018.05.014>.
4. **Mischuk D. O.** Development of control system for construction robot using technologies "Microsoft robotic studio". II International scientific and technical conference "Energy-saving machines and technologies", September 29 - October 1, (2015). Kyiv, KNUCA, 47. – (in Ukrainian).
5. **Mischuk D. O.** (2014). Kinematichnij analiz manipulyacijnoyi sistemi budivelnogo robota [Kinematic analysis of the manipulation system of a construction robot]. *Vestnik Harkovskogo nacionalnogo avtomobilno-dorozhnogo universiteta*, Vol. 65-66, 90-96. – (in Ukrainian).
6. **Zenkevich S. L., Yushenko A. S.** (2000). *Upravlenie robotami. Osnovy upravleniya manipulyacionnymi robotami: Ucheb. Dlya vuzov* [Robot control. Fundamentals of manipulation robots control: Textbook]. Moscow, MGТУ Publ., 400.
7. **Karlik B., Aydin S.** (2000). An improved approach to the solution of inverse kinematics problems for robot manipulators. *Engineering applications of artificial intelligence*, Vol.13, Nr. 2, 159-164.
8. **Spiny G. A.** (1991). *Promyshlenye roboty. Konstruirovaniye i primeneniye* [Industrial robots. Design and Application]. Kyiv, Vysha shk. Publ., 331. – (in Russian).
9. **Feng Y., Yaonan W., Yimin Y.** (2014). Inverse kinematics solution for robot manipulator based on neural network under joint subspace. *International Journal of Computers Communications & Control*, Vol.7, Nr. 3, 459-472.
10. **Lovejkin V. S., Mischuk D. O.** (2013). Optimizaciya rezhimiv zmini vilotu manipulyatora z gidroprivodom. Monografiya. [The optimization modes of change departure the manipulator with hydraulic drive. Monograph]. Kyiv, CP Komprint Publ., 206. – (in Ukrainian).
11. **Lovejkin V. S., Mischuk D. A.** (2012). Matematichne modeljuvannja zmini vil'otu vantazhu manipulyatorom z gidroprivodom [Mathematical modeling of changes in cargo flight with hydraulic manipulator]. *Girnichy, budivel'ni, dorozhni ta meliorativni mashini* [Mining, construction, road and reclamation machines], Nr. 79, 9-15. – (in Ukrainian).
12. **Mischuk D., Gorbatyuk Ye., Volyanyuk V.** (2019). Vznachennya geometrichnih parametrov manipulyatora za harakteristikami robochogo seredovisha [Determination of geometric parameters of a manipulator using by characteristics the working environment]. *Girnichy, budivel'ni, dorozhni i meliorativni mashini*, Nr. 94, 25-34. – (in Ukrainian). <https://doi.org/10.32347/gbdmm2019.94.0201>
13. **Momani S., Abo-Hammour Z. S., Alsmadi O. M. K.** (2016). Solution of inverse kinematics problem using genetic algorithms. *Applied Mathematics & Information Sciences*, Vol. 10, No.1, 225.
14. **LaValle S. M.** (2006). *Planning algorithms*. Cambridge university press, 1007.
15. **Stilman M. et al.** Manipulation planning among movable obstacles. *Robotics and Automation*. (2007) IEEE International Conference on., IEEE, 3327-3332.
16. **Hirano Y., Kitahama K., Yoshizawa S.** Image-based object recognition and dexterous hand/arm motion planning using rrts for grasping in cluttered scene. *Intelligent Robots and Systems, (IROS 2005)*. (2005) IEEE/RSJ International Conference on., IEEE, 2041-2046.
17. **Graca Marcos M., Machado J.A.T., Azevedo-Perdicoulis T. P.** (2009). Trajectory planning of redundant manipulators using genetic algorithms. *Communications in non-linear science and numerical simulation*, Vol.14, Nr.7, 2858-2869.
18. **Mischuk D. O.** (2015). Increasing efficiency manipulator by optimal control. *Girnichy, budivel'ni, dorozhni i meliorativni mashini*, Nr. 85, 43-50. – (in Ukrainian).
19. **Lovejkin V. S., Mischuk D. A.** (2019). Synthesis of Optimal Dynamic Mode of Manipulator Boom Movement Mounted on Elastic Base. *Science & Technique*, Nr. 18(1), 55-61. – (in Russian) <https://doi.org/10.21122/2227-1031-2019-18-1-55-61>.
20. **Lovejkin V. S., Mishuk D. O.** (2009). Vznachennya optimalnih rezhimiv ruhu manipulyatora za proces pusku (galmuvannya) pid chas roboti za odniyeyu z uzagalnenih koordinat [Determining the optimal modes of movement of the manipulator for the process of starting (braking) while working on one of the generalized coordinates]. *Girnichy, budivel'ni, dorozhni ta meliorativni mashini*, Nr.73, 35-42. – (in Ukrainian).
21. **Mischuk, D., Bojchenko A.** (2019). Rozrobka koncepciyi sistemi keruvannya robotom dlya shtukaturnih robit na osnovi nejronnoyi merezhi [Development of the concept of control system work for plaster works based on neural net-

work]. Girnichi, budivelni, dorozhni ta meliorativni mashini [Mining, construction, road and reclamation machines], Nr. 93, 46-60. – (in Ukrainian). <https://doi.org/10.32347/gbdmm2019.93.0501>.

22. **Bingul Z., Ertunc H. M., Oysu C.** Applying neural network to inverse kinematic problem for 6R robot manipulator with offset wrist. Adaptive and Natural Computing Algorithms. (2005). Springer, Vienna, 112-115.
23. **Koker R., Cakar T., Sari Y.** (2014). A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators. Engineering with Computers, Vol.30, Nr. 4, 641-649.
24. **Mischuk D., Voliyanuk V., Gorbatyk Eu.** (2017). The module design of construction robot. Girnichi, budivelni, dorozhni ta meliorativni mashini [Mining, construction, road and reclamation machines], Nr. 89, 90-97. – (in Ukrainian). <http://gbdmm.knuba.edu.ua/article/view/114398>.
25. **Dorogov A. Yu.** (2014). Teoriya i proektirovanie bystrykh perestraivaemykh preobrazovaniy i slabosvyazannykh nejronnykh setej [Theory and design of fast reconstructed transformations and loosely coupled neural networks]. Politehnika Publ, 328. – (in Russian).
26. **Zheron O.** (2020). Prikladnoe mashinnoe obuchenie s pomoshyu Scikit-Learn i TensorFlow: koncepcii, instrumenty i tehniki dlya sozdaniya intellektualnykh sistem [Hands-On Machine Learning with Scikit-Learn and TensorFlow]. Dialektika-Vilyams, 688.
27. **TensorFlow Core for beginners.** Keras basics. <https://www.tensorflow.org/tutorials?hl=en>.
28. **Komotska S., Bigun M.** (2019). Viprobuвання pnevmocilindriv z vikoristannyam rele chasu [Testing of pneumatic cylinders using time relays]. Girnichi, budivelni, dorozhni ta meliorativni mashini [Mining, construction, road and reclamation machines], Nr. 94, 51-62. – (in Ukrainian). <https://doi.org/10.32347/gbdmm2019.94.0501>.
29. **Sukach M. K.,** (2019). Laboratory research of staple leaf springs. Mining, construction, road and melioration machines, Nr. 94, 16-24. – (in Russian). <https://doi.org/10.32347/gbdmm2019.94.01.02>.

**Possibilities of using the packet TensorFlow neural network design library for the kinematic study of a manipulator**

*Andriy Boychenko<sup>1</sup>, Dmitry Mishchuk<sup>2</sup>*

*Kyiv National University of  
Construction and Architecture*

**Abstract.** The study of the kinematic structures of multi-link manipulators is not a trivial task. Difficulties appear in the study of kinematic schemes in which the number of moving independent parts exceeds the number of generalized independent degrees of mobility, since in such cases there is an overabundance of mobility of the system and more than two manipulator configurations can exist for one position. Such ambiguity greatly complicates the search for dependencies between the generalized coordinates of the system and the endpoint of movement of the working body. It is also necessary to take into account that in the future, when compiling dynamic equations of motion for such systems, such ambiguity complicates the study of problems of dynamics. The solutions to such problems are carried out by different methods, and this article discusses the possibility of using neural networks to study such ambiguous problems using the example of a kinematic study of an articulated manipulator. In the presented work, the possibilities of using the Tensor-Flow library from Google are considered with the help of which models of neural networks and "neurons" are created, as well as functions for finding the weight coefficients of setting the created network model. In the course of the study, a neural network was created to calculate the regression function of the dependencies between the geometric coordinates of the point of movement of the manipulator gripper and the generalized coordinates of its boom system.

The purpose of this study is to demonstrate the capabilities of neural networks in solving engineering problems that require complex mathematical transformations.

The use of neural network architectures in manipulator control systems makes it possible to create universal mechanisms for performing various technological procedures, while reducing the cost of developing such solutions.

In general, neural networks are not universal means for solving all engineering problems, since they require a larger set of test samples to adjust (train) the parameters of their models however, there are a number of problems in solving which neural networks have great advantages.

**Keywords:** TensorFlow, manipulator, control system, neural network.