

# **СБОРНИК НАУЧНЫХ ТРУДОВ НГТУ**

**№ 2 (92)  
2018**

**В НОМЕРЕ:**

- ♦ АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ И ИДЕНТИФИКАЦИЯ**
- ♦ СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

### **Учредитель**

Новосибирский государственный технический университет

### **Главный редактор**

**А.Г. Вострецов**, д-р техн. наук, проф., засл. деятель науки РФ, НГТУ, г. Новосибирск

### **Заместитель главного редактора**

**В.И. Денисов**, д-р техн. наук, проф., засл. деятель науки РФ, НГТУ, г. Новосибирск

### **Ответственный секретарь**

**А.А. Воевода**, д-р техн. наук, проф., НГТУ, г. Новосибирск

### **Редакционная коллегия:**

**Ю.Е. Воскобойников**, д-р физ.-мат. наук, проф., СибСТРИН, г. Новосибирск

**Ю.Д. Григорьев**, д-р техн. наук, проф., СПбГЭТУ (ЛЭТИ), г. С.-Петербург

**А.М. Малышенко**, д-р техн. наук, проф., ТПУ, г. Томск

**Ю.Ф. Мухомад**, д-р техн. наук, проф., засл. деятель науки РФ, ИрГУПС, г. Иркутск

**А.И. Рубан**, д-р техн. наук, проф., засл. деятель науки РФ,

СФУ, Ин-т косм. и информ. технологий, г. Красноярск

**В.И. Хабаров**, д-р техн. наук, проф., СГУПС, г. Новосибирск

### **Журнал зарегистрирован**

**в Министерстве РФ по делам печати, телерадиовещания**

**и средств массовых коммуникаций в 2002 г.**

**(свидетельство ПИ № 77-11739 от 08 февраля 2002 г.)**

Адрес редакции: 630073, г. Новосибирск, пр. К. Маркса, 20. E-mail: [ucit@ucit.ru](mailto:ucit@ucit.ru)

Web site: <http://journals.nstu.ru/sbornik/>

Editorial adress: 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation

---

**СБОРНИК  
НАУЧНЫХ ТРУДОВ  
НОВОСИБИРСКОГО ГОСУДАРСТВЕННОГО  
ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА**

---

ISSN 2307-6879

№ 2 (92)

2018

**СОДЕРЖАНИЕ**

*АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ И ИДЕНТИФИКАЦИЯ*

<b>Воевода А.А., Бобобеков К.М.</b> Преобразование полиномиального представления многоканального объекта к взаимно простому виду.....	7
<b>Филошов В.Ю.</b> Нелинейный объект: линеаризация обратными связями	36

*СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ*

<b>Алтухов В.Г.</b> Обзор технологии навигации мобильных роботов Omni-directional vSLAM.....	81
<b>Воевода А.А., Романников Д.О.</b> Обучение нейронной сети с предварительно рассчитанной структурой.....	93
<b>Воевода А.А., Романников Д.О.</b> Трансляция нейронной сети в сеть Петри.....	101
<b>Романников Д.О.</b> Пример решения минимаксной задачи преследования с использованием нейронных сетей.....	108
<b>Романников Д.О., Ибраева Д.М.</b> О проблемах синтеза нейронных сетей.....	117

Выпускающий редактор *И.П. Брованова*  
Корректор *Л.Н. Кинит*  
Компьютерная верстка *С.И. Ткачева*

---

Лицензия № ИД 04303 от 20.03.01. Подписано в печать 10.09.2018  
Формат 60×84/16. Бумага офсетная. Тираж 300 экз. Уч.-изд. л. 7,2  
Печ. л. 7,75. Изд. № 198. Заказ № 1199. Цена договорная

---

Отпечатано в типографии  
Новосибирского государственного технического университета  
630073, г. Новосибирск, пр. К. Маркса, 20

## **Editorial board**

Novosibirsk State Technical University

### ***Chief Editor***

***Vostretsov A.G.***, D. Sc. (Eng.), Novosibirsk State Technical University,  
Novosibirsk, RF

### ***Deputy Chief Editor***

***Denisov V.I.***, D. Sc. (Eng.), Novosibirsk State Technical University,  
Novosibirsk, RF

### ***Executive Secretary***

***Voevoda A.A.***, D. Sc. (Eng.), Novosibirsk State Technical University,  
Novosibirsk, RF

### ***The members of the editorial Board***

***Voskoboinikov Yu.E.***, D. Sc. (Eng.), Novosibirsk State University of Architecture  
and Civil Engineering, Novosibirsk, RF

***Grigor'ev Iu.D.***, D. Sc. (Eng.), Saint-Petersburg Electrotechnical University "LETI",  
Saint-Petersburg, RF

***Malysenko A.M.***, D. Sc. (Eng.), National Research Tomsk Polytechnic University,  
Tomsk, RF

***Mukhopad Iu.F.***, D. Sc. (Eng.), Irkutsk State University of Railway Engineering,  
Irkutsk, RF

***Ruban A.I.***, D. Sc. (Eng.), Siberian Federal University, Krasnoyarsk, RF

***Khabarov V.I.***, D. Sc. (Eng.), Siberian Transport University, Novosibirsk, RF

*The journal is registered  
in the RF Ministry for press, broadcasting  
and masscommunications in 2002  
(certificate PI № 77-11739 from February 08, 2002)*

Editorial address: 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation.

E-mail: [ucit@ucit.ru](mailto:ucit@ucit.ru)

Web Site: <http://journals.nstu.ru/sbornik/>

© Authors, 2018

© Novosibirsk State

Technical University, 2018

**CONTENTS**

*AUTOMATIC CONTROL AND IDENTIFICATION*

<b>Voevoda A.A., Bobobekov K.M.</b> Reduction of the matrix polynomial representation of the transfer function to a coprime form using the Sylvester matrix .....	7
<b>Filiushov V.Yu.</b> Examples of structural transformation of nonlinear object ..	36

*INFORMATION PROCESSING*

<b>Altukhov V.G.</b> A survey of mobile robot's navigation technology Omnidirectional vSLAM .....	81
<b>Voevoda A.A., Romannikov D.O.</b> Training of neural network with preliminary calculated structure .....	93
<b>Voevoda A.A., Romannikov D.O.</b> Translation of the neural network in the Petri network .....	101
<b>Romannikov D.O.</b> An example of solving a minimax pursuit problem using neural networks .....	108
<b>Romannikov D.O., Ibraeva D.M.</b> On the problems of synthesis of neural networks .....	117

Editor *L.N. Kinsht*  
Publishing Editor *I.P. Brovanova*  
Computer imposition *S.I. Tkacheva*

---

License № ID 04303 from 20.03.01. Signed in print 10.09.2018. Format 60 × 84/16  
Offset Paper. Circulation is 300 copies. Educational-ed. liter. 7,2. printed pages 7,75.  
Publishing number 198. Order number 1199.

---

It is printed in printing house of Novosibirsk State Technical University  
630073, Novosibirsk, 20 K. Marx prospekt

*АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ  
И ИДЕНТИФИКАЦИЯ*

УДК 681.513

DOI: 10.17212/2307-6879-2018-2-7-35

**ПРЕОБРАЗОВАНИЕ ПОЛИНОМИАЛЬНОГО  
ПРЕДСТАВЛЕНИЯ МНОГОКАНАЛЬНОГО ОБЪЕКТА  
К ВЗАИМНО ПРОСТОМУ ВИДУ\***

А.А. ВОЕВОДА<sup>1</sup>, К.М. БОБОБЕКОВ<sup>2</sup>

<sup>1</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, доктор технических наук, профессор кафедры автоматики. E-mail: ucit@ucit.ru

<sup>2</sup> 630087, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, аспирант кафедры автоматики. E-mail: kurbon\_111@mail.ru

Рассматривается приведение матричного полиномиального представления передаточной функции к взаимно простому виду. Нарушение требования взаимно простого представления приводит к появлению одинаковых корней в «числителе» и «знаменателе» как в одноканальных системах, так и в многоканальных. Это может привести к скрытой неустойчивости систем автоматического управления, неуправляемости или ненаблюдаемости; устранение подобных неприятностей иногда можно исключить введением дополнительных ограничений на задание корней желаемого характеристического полинома замкнутой системы. Кроме того, для существования решения полиномиального матричного уравнения при поиске регулятора требуется взаимно простое разложение передаточной функции объекта. Для проверки на взаимную простоту составляется система линейных однородных уравнений: другими словами, строится матрица Сильвестра и искомый матричный вектор из «элементов» взаимно простого представления. Далее при помощи команды  $[q, r] = qr(S)$  пакета Matlab, где  $S$  – матрица Сильвестра, проверяются линейно зависимые столбцы в направлении слева направо. Если диагональные элементы матрицы  $r$  ненулевые, то исходные полиномиальные матрицы взаимно простые, а если присутствуют нули на диагонали – ищем правые взаимно простые полиномиальные матрицы. Приведены алгоритм вычисления взаимно простого представления и три примера, иллюстрирующие предлагаемый алгоритм.

**Ключевые слова:** матричная передаточная функция, левое и правое матричное полиномиальное разложение, взаимно простое разложение, не взаимно простое разложение, полиномиальные матрицы, линейное уравнение, матрица Сильвестра, вырожденная матрица, алгоритм поиска взаимной простоты

---

\* Статья получена 10 января 2018 г.



## ВВЕДЕНИЕ

Рассмотрим вычисление полиномиального представления передаточной функции к взаимно простому виду. Невыполнение требования взаимной простоты может привести к проблемам, связанным с появлением скрытой неустойчивости систем автоматического управления, неуправляемости или ненаблюдаемости [1–7]. Подобные затруднения могут быть исключены преобразованием полиномиального разложения к взаимно простому виду. Доказано, что для существования решения полиномиального матричного уравнения, которое возникает при поиске регулятора, гарантирующего заданный характеристический полином – в одноканальном случае, или заданную характеристическую полиномиальную матрицу – в многоканальном случае, требуется взаимно простое разложение. Для проверки на взаимную простоту удобно перейти от полиномиальных матричных уравнений к матричным уравнениям с числовыми коэффициентами [1, 2, 15].

Взаимно простое разложение может быть получено на основе алгоритма, суть которого состоит в поиске линейно зависимых столбцов матрицы Сильвестра слева направо. Для этого можно использовать  $QR$ -разложение (декомпозицию): в пакете Matlab соответствующий оператор  $qr$ . Предполагаем, и это действительно так, что у матрицы Сильвестра строк больше или равно количеству столбцов. Суть  $QR$ -разложения состоит следующем: для матрицы  $S$  размером  $n \times m$  ( $n \geq m$ ) существует ортогональная матрица  $\bar{Q}$  размером  $n \times n$  такая, что

$$\bar{Q}S = R,$$

где  $R$  – верхнетреугольная матрица такого же размера, как  $S$  –  $n \times m$ . Так как  $\bar{Q}$  «преобразует» строчки матрицы  $S$ , линейная независимость столбцов  $S$  «сохраняется» в столбцах  $R$ . Другими словами, если какой-либо столбец  $R$  линейно зависим от столбцов слева от него, таким же будет и столбец в  $S$ . Так как  $R$  – это верхнетреугольная матрица, ее  $m$ -й столбец линейно *независим* от столбцов с левой стороны, если и только если его  $m$ -й элемент на диагонали *не нулевой*. Таким образом, анализируя диагональные элементы матрицы  $R$  слева направо, можем определить линейно зависимые/независимые столбцы матрицы  $S$  в зависимости от того, какие элементы стоят на диагонали, ноль или не ноль. Так как  $\bar{Q}$  ортогональная, имеем  $\bar{Q}^{-1} = \bar{Q}' = Q$  и  $\bar{Q}S = R$ , т. е.  $S = QR$ . Эта процедура называется  $QR$ -разложением (*decom-position*). В Matlab  $Q$  и  $R$  могут быть получены при помощи использования команды  $[q, r] = qr(s)$ .

Для скалярных передаточных функций можем использовать либо оператор *rank*, либо *QR*-разложение для определения количества  $\mu$  линейно независимых  $n$ -столбцов. В матричном случае использование оператора *rank* неудобно и предлагается использовать *QR*-разложение.

Приведены алгоритм вычисления взаимно простого представления и примеры, подтверждающие работоспособность предлагаемого алгоритма. Данная работа является продолжением исследований, начатых в [7, 8], продолженных в [10, 14–17, 23] и опирающихся на базовые результаты, полученные в работах Chen.

## 1. ФОРМАЛИЗАЦИЯ ЗАДАЧИ ПРИВЕДЕНИЯ К ВЗАИМНО ПРОСТОМУ ВИДУ

Сделаем предположение, что матричная передаточная функция, например, объекта представлена в виде левого, возможно, не в виде взаимно простого полиномиального разложения<sup>1</sup>

$$W_{ob}(s) = D^{-1}(s)N(s).$$

Поставим задачу поиска правого *взаимно простого разложения*:

$$W_{ob}(s) = \bar{N}(s)\bar{D}^{-1}(s),$$

т. е. имеет место равенство

$$W_{ob}(s) = D^{-1}(s)N(s) = \bar{N}(s)\bar{D}^{-1}(s).$$

---

<sup>1</sup> Можно сделать следующее пояснение о переходе от, например, правого не взаимно простого разложения  $\bar{N}(s)\bar{D}^{-1}(s)$  к правому взаимно простому разложению. Допустим, каким-либо образом нашли наибольший общий правый делитель  $R(s)$  матриц  $\bar{N}(s)$  и  $\bar{D}(s)$ , т. е.  $\bar{N}(s) = \hat{N}(s)R(s)$  и  $\bar{D}(s) = \hat{D}(s)R(s)$ . Тогда  $\bar{N}(s)\bar{D}^{-1}(s) = \hat{N}(s)R(s)R^{-1}(s)\hat{D}^{-1}(s) = \hat{N}(s)\hat{D}^{-1}(s)$ . Здесь  $\hat{N}(s)$  и  $\hat{D}(s)$  – правое взаимно простое разложение. Для дальнейших расчетов требуется, чтобы матрица  $\hat{D}(s)$  была столбцово-приведенной.

Очевидно, что справедливо

$$D(s)(-\bar{N}(s)) + N(s)\bar{D}(s) = 0. \quad (1)$$

Матрицы с полиномиальными элементами, входящие в (1), можем записать в виде полиномов с матричными коэффициентами, у которых элементы – вещественные числа:

$$\begin{aligned} D(s) &= D_0 + D_1s + \dots + D_{n-1}s^{n-1} + D_n s^n, \\ N(s) &= N_0 + N_1s + \dots + N_{m-1}s^{m-1} + N_m s^m, \\ \bar{D}(s) &= \bar{D}_0 + \bar{D}_1s + \dots + \bar{D}_{n-1}s^{n-1}, \\ \bar{N}(s) &= \bar{N}_0 + \bar{N}_1s + \dots + \bar{N}_{n-1}s^{n-1}. \end{aligned}$$

Здесь  $D_i$ ,  $N_i$ ,  $\bar{D}_i$  и  $\bar{N}_i$  – числовые матрицы размером  $p \times p$ ,  $p$  – число каналов и  $m \leq n$ . Матрицы  $D_i$  и  $N_i$  нам известны, и необходимо всего лишь определить  $\bar{D}_i$  и  $\bar{N}_i$ . Подставляя  $D_i$ ,  $N_i$ ,  $\bar{D}_i$  и  $\bar{N}_i$  в (1), выписываем систему линейных уравнений

$$\underbrace{\begin{bmatrix} D & N & D & N & \vdots & D & N \\ D_0 & N_0 & 0 & 0 & \vdots & 0 & 0 \\ D_1 & N_1 & D_0 & N_0 & \vdots & 0 & 0 \\ \dots & \dots & D_1 & N_1 & \vdots & D_0 & N_0 \\ D_n & N_n & \dots & \dots & \vdots & D_1 & N_1 \\ 0 & 0 & D_n & N_n & \vdots & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & D_n & N_n \end{bmatrix}}_S \cdot \underbrace{\begin{bmatrix} -\bar{N}_0 \\ \bar{D}_0 \\ -\bar{N}_1 \\ \bar{D}_1 \\ \dots \\ -\bar{N}_{n-1} \\ \bar{D}_{n-1} \end{bmatrix}}_x = 0, \quad (2)$$

где  $S$  – матрица Сильвестра<sup>2</sup> размером  $2n \times 2n$  блоков, столбцы которой обозначим через  $d_i$  и  $n_i$ :  $D = (d_1, d_2, \dots, d_p)$  и  $N = (n_1, n_2, \dots, n_p)$ . Раскроем матрицу  $S$  и блочный вектор  $x$ :

<sup>2</sup> Транспонированные матрицы  $S$  тоже назовем матрицами Сильвестра.

$$\begin{bmatrix}
 d_1 & d_2 & n_1 & n_2 & \vdots & d_1 & d_2 & n_2 & n_2 \\
 d_{11}^0 & d_{12}^0 & n_{11}^0 & n_{12}^0 & \vdots & 0 & 0 & 0 & 0 \\
 d_{21}^0 & d_{22}^0 & n_{21}^0 & n_{22}^0 & \vdots & 0 & 0 & 0 & 0 \\
 d_{11}^1 & d_{12}^1 & n_{11}^1 & n_{12}^1 & \vdots & d_{11}^0 & d_{12}^0 & n_{11}^0 & n_{12}^0 \\
 d_{21}^1 & d_{22}^1 & n_{21}^1 & n_{22}^1 & \vdots & d_{21}^0 & d_{22}^0 & n_{21}^0 & n_{22}^0 \\
 \dots & \dots & \dots & \dots & \vdots & d_{11}^1 & d_{12}^1 & n_{11}^1 & n_{12}^1 \\
 d_{11}^n & d_{12}^n & n_{11}^n & n_{12}^n & \vdots & d_{21}^1 & d_{22}^1 & n_{21}^1 & n_{22}^1 \\
 d_{21}^n & d_{22}^n & n_{21}^n & n_{22}^n & \vdots & \dots & \dots & \dots & \dots \\
 0 & 0 & 0 & 0 & \vdots & d_{11}^n & d_{12}^n & n_{11}^n & n_{12}^n \\
 0 & 0 & 0 & 0 & \vdots & d_{21}^n & d_{22}^n & n_{21}^n & n_{22}^n
 \end{bmatrix}
 \begin{bmatrix}
 x_1 & x_2 \\
 -\bar{n}_{11}^0 & -\bar{n}_{12}^0 \\
 -\bar{n}_{21}^0 & -\bar{n}_{22}^0 \\
 \hline
 \bar{d}_{11}^0 & \bar{d}_{12}^0 \\
 \bar{d}_{21}^0 & \bar{d}_{22}^0 \\
 \hline
 \dots & \dots \\
 \hline
 -\bar{n}_{11}^{n-1} & -\bar{n}_{12}^{n-1} \\
 -\bar{n}_{21}^1 & -\bar{n}_{22}^1 \\
 \hline
 \bar{d}_{11}^{n-1} & \bar{d}_{12}^{n-1} \\
 \bar{d}_{21}^{n-1} & \bar{d}_{22}^{n-1}
 \end{bmatrix} = 0.$$

Для краткости записи матрица выписана для двухканальной системы  $p = 2$ .

Обсудим некоторые общие свойства матрицы  $S$  в предположении, что необходимо найти линейно независимые столбцы  $S$  слева направо. Это приводит нас к тому, что каждый  $D$ -столбец в каждом  $D$ -блочном столбце линейно независим от  $D$ -столбцов слева (из-за смещения вниз). Ситуация для  $N$ -столбцов, однако, отличная. Напомним, что здесь имеется  $p$   $N$ -столбцов в каждом  $N$ -блочном столбце. Мы используем  $N_i$ -столбец для обозначения  $i$ -го  $N$ -столбца в каждом  $N$ -блочном столбце. Это приводит к тому, что если  $N_i$ -столбец в некотором  $N$ -блочном столбце линейно зависит от столбцов, входящих в  $N$ -блочные столбцы слева, тогда все последующие такие же  $N_i$ -столбцы ввиду повторяемости структуры  $S$  будут линейно зависимыми от столбцов слева. Пусть  $\mu_i, i = 1, 2, \dots, p$ , будет число линейно независимых  $N_i$ -столбцов в  $S$ . Назовем их *столбцовыми индексами (column indices)*  $W_{ob}(s)$ . Первый  $N_i$ -столбец, линейно зависимый от столбцов слева, называют первым зависимым  $N_i$ -столбцом (*primary dependent  $N_i$ -column*). Ясно, что  $(\mu_i + 1)$ -й  $N_i$ -столбец – это первый зависимый столбец.

Формируем матрицу  $S_i, i = 1, 2, \dots, p$ , которая включает в себя первый линейно зависимый столбец  $(\mu_i + 1)$ , и вычисляем нормированный вектор  $x_i$  из ядра,  $i$  принимает значения  $i = 1, 2, \dots, p$ .

Из таких нормированных векторов мы можем получить правое взаимно простое разложение. Результирующая матрица  $D(s)$  имеет наименьшую возможную столбцовую степень. Дополнительно  $\bar{D}(s)$  будет автоматически столбцово приведенной. Следующий пример иллюстрирует вышеприведенную процедуру.

## 2. ПРИМЕРЫ

**Пример 1<sup>3</sup>.** Найти взаимно простое правое полиномиальное матричное разложение передаточной функции

$$W_{ob}(s) = \begin{bmatrix} \frac{4s-10}{2s+1} & \frac{3}{s+2} \\ 1 & \frac{s+1}{(s+2)^2} \end{bmatrix}. \quad (3)$$

Сначала нужно найти левое разложение, не обязательно взаимно простое слева. Используя наименьший общий знаменатель каждой строки, можем легко получить

$$W_{ob}(s) = \begin{bmatrix} (2s+1)(s+2) & 0 \\ 0 & (2s+1)(s+2)^2 \end{bmatrix}^{-1} \times \\ \times \begin{bmatrix} (4s-10)(s+2) & 3(2s+1) \\ s+2 & (s+1)(2s+1) \end{bmatrix} =: D^{-1}(s)N(s).$$

Таким образом, имеем

$$D(s) = \begin{bmatrix} 2s^2 + 5s + 2 & 0 \\ 0 & 2s^3 + 9s^2 + 12s + 4 \end{bmatrix} = \\ = \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}}_{D_0} + \underbrace{\begin{bmatrix} 5 & 0 \\ 0 & 12 \end{bmatrix}}_{D_1} s + \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 9 \end{bmatrix}}_{D_2} s^2 + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}}_{D_3} s^3; \quad (4)$$

---

<sup>3</sup> Объект для анализа взят из Chen, который дан в виде матричной передаточной функции.

$$\begin{aligned}
 N(s) &= \begin{bmatrix} 4s^2 - 2s - 20 & 6s + 3 \\ s + 2 & 2s^2 + 3s + 1 \end{bmatrix} = \\
 &= \underbrace{\begin{bmatrix} -20 & 3 \\ 2 & 1 \end{bmatrix}}_{N_0} + \underbrace{\begin{bmatrix} -2 & 6 \\ 1 & 3 \end{bmatrix}}_{N_1} s + \underbrace{\begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}}_{N_2} s^2 + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{N_3} s^3. \tag{5}
 \end{aligned}$$

В нашем случае  $n=3$  и каналов  $p \times p = 2 \times 2$ . Сформируем матрицу Сильвестра (2) размером  $2n \times 2n = 6 \times 6$  блоков: 1-й и 2-й  $D$ - и  $N$ -блочные столбцы ( $N = (n_1 \ n_2 \ \dots \ n_p)$  и  $D = (d_1 \ d_2 \ \dots \ d_p)$ ), находящиеся рядом, дополняем снизу нулевыми матрицами для того, чтобы количество строк равнялось  $2n$  блокам. Следующие блочные столбцы, соответствующие 3-му и 4-му столбцам, получим из первого и второго блочных столбцов со смещением вниз на одну позицию и дополняем сверху нулевыми матрицами. Аналогичным образом продолжим процедуру до того, что количество  $D$ - и  $N$ -блочных столбцов равнялось  $2n$ . Также формируем блочный столбец  $x$  из матричных коэффициентов полиномов  $\bar{N}(s)$  и  $\bar{D}(s)$  размером  $2n \times 1$  блоков (2).

$$\begin{array}{cccccc}
 D & N & D & N & D & N \\
 \hline
 D_0 & N_0 & O & O & O & O \\
 D_1 & N_1 & D_0 & N_0 & O & O \\
 D_2 & N_2 & D_1 & N_1 & D_0 & N_0 \\
 D_3 & N_3 & D_2 & N_2 & D_1 & N_1 \\
 O & O & D_3 & N_3 & D_2 & N_2 \\
 O & O & 0 & 0 & D_3 & N_3
 \end{array}
 \underbrace{\hspace{10em}}_S
 \underbrace{\begin{bmatrix} -\bar{N}_0 \\ \bar{D}_0 \\ -\bar{N}_1 \\ \bar{D}_1 \\ -\bar{N}_2 \\ \bar{D}_2 \end{bmatrix}}_x = 0,$$

где  $O$  – нулевая матрица размером  $p \times p = 2 \times 2$ ,  $\bar{N}_i$  и  $\bar{D}_i$  – искомые взаимно простые полиномиальные матрицы. Если подставить значения элементов  $D_i$  и  $N_i$  из (4) и (5), то матрица Сильвестра принимает вид

$$S = \begin{array}{cccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
 \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \end{array} & \left[ \begin{array}{cccccccccccc}
 2 & 0 & -20 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 5 & 0 & -2 & 6 & 2 & 0 & -20 & 3 & 0 & 0 & 0 & 0 & 0 \\
 0 & 12 & 1 & 3 & 0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\
 2 & 0 & 4 & 0 & 5 & 0 & -2 & 6 & 2 & 0 & -20 & 3 & 0 \\
 0 & 9 & 0 & 2 & 0 & 12 & 1 & 3 & 0 & 4 & 2 & 1 & 0 \\
 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 5 & 0 & -2 & 6 & 0 \\
 0 & 2 & 0 & 0 & 0 & 9 & 0 & 2 & 0 & 12 & 1 & 3 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 9 & 0 & 2 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0
 \end{array} \right] \\
 & \begin{array}{cccccccccccc}
 d_1 & d_2 & n_1 & n_2 & d_1 & d_2 & n_1 & n_2 & d_1 & d_2 & n_1 & n_2
 \end{array}
 \end{array}$$

Матрицу  $S$  можно условно записать так:

$$S = \begin{array}{cccccccccccc}
 & d_1 & d_2 & n_1 & n_2 & d_1 & d_2 & n_1 & n_2 & d_1 & d_2 & n_1 & n_2 \\
 \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \end{array} & \left[ \begin{array}{cccccccccccc}
 | & | & | & | & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 | & | & | & | & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 | & | & | & | & | & | & | & | & 0 & 0 & 0 & 0 \\
 | & | & | & | & | & | & | & | & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & | & | & | & | & | & | & | & | \\
 0 & 0 & 0 & 0 & | & | & | & | & | & | & | & | \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & | & | & | & | \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & | & | & | & |
 \end{array} \right],
 \end{array}$$

где  $d_1, d_2, n_1$  и  $n_2$  – столбики размером восемь. Затем используем  $QR$ -разложение<sup>4</sup> для поиска линейно независимых столбцов в порядке слева направо в матрице  $S$ . Вычисления в пакете Matlab можно выполнить так:

```
>> d1=[2 0 5 0 2 0 0 0]; d2=[0 4 0 12 0 9 0 2]; % столбики размером восемь
n1=[-20 2 -2 1 4 0 0 0]; n2=[3 1 6 3 0 2 0 0];
S=[d1 0 0 0 0; d2 0 0 0 0; n1 0 0 0 0; n2 0 0 0 0;
0 0 d1 0 0; 0 0 d2 0 0; 0 0 n1 0 0; 0 0 n2 0 0;
0 0 0 0 d1; 0 0 0 0 d2; 0 0 0 0 n1; 0 0 0 0 n2]; % транспонирование!
[q, r]=qr(S); % Вычисление ортогональной матрицы q и матрицы
% r – верхнетреугольной таких, что  $q^{-1}S = r$ 
r % вывод матрицы r
```

Нам нужна только матрица  $r$ , а матрицу  $q$  не показываем. В результате получим матрицу  $r$ :

$$r = \begin{array}{c} \left[ \begin{array}{cccccc} d1 & d2 & n1 & n2 & d1 & d2 \\ -5,7 & 0 & 7,3 & -6,3 & -3,5 & 0 \\ 0 & -15,7 & -1,3 & -3,7 & 0 & -11,1 \\ 0 & 0 & -19,2 & 1,4 & -2,5 & 0,5 \\ 0 & 0 & 0 & -2,0 & 3,4 & 2,9 \\ 0 & 0 & 0 & 0 & -2,2 & 3,9 \\ 0 & 0 & 0 & 0 & 0 & -9,9 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \end{array}$$

1 2 3 4 5 6

<sup>4</sup> Для скалярных передаточных функций можем использовать либо оператор (команду) *rank*, либо команду  $QR$ -разложения для вычисления линейно зависимых столбцов (слева направо). В матричном случае использование команды *rank* неудобно и предлагается использование  $QR$ -разложение.



$$\begin{array}{cccccc}
 & n_1 & n_2 & d_1 & d_2 & n_1 & n_2 \\
 \left[ \begin{array}{cccccc}
 18,1 & -4,7 & -0,7 & 0 & 7 & -1 \\
 -2,1 & -2,8 & 0 & -3,8 & -1,3 & -1 \\
 5,3 & -2,6 & -0,7 & 0,3 & 6,9 & -1 \\
 6,6 & 4,5 & 1,7 & 3,2 & -16,4 & 3,3 \\
 -4,6 & -0,5 & -4,8 & 4,7 & 4,2 & -4,6 \\
 0,7 & -1,4 & -1,4 & -10,5 & -4,7 & -4,2 \\
 -0,3 & 0,6 & -0,5 & -8,5 & -0,2 & -2,5 \\
 0 & \boxed{0} & -0,5 & -3,2 & -4 & 0 \\
 0 & 0 & -2 & 0,2 & 0 & 0 \\
 0 & 0 & 0 & -2,5 & 0 & 0 \\
 0 & 0 & 0 & 0 & \boxed{0} & 0 \\
 0 & 0 & 0 & 0 & 0 & \boxed{0}
 \end{array} \right] \cdot \\
 & 7 & 8 & 9 & 10 & 11 & 12
 \end{array}$$

Проанализируем линейную зависимость столбцов матрицы  $S$  при помощи матрицы  $r$ : видим, что все  $D$ -столбцы линейно независимые от столбцов слева, так как их элементы, соответствующие диагонали, ненулевые. Что касается  $N$ -столбцов, то всего два линейно независимых  $n_1$ -столбца (3-й и 7-й,  $\mu_1 = 2$ ) и один линейно независимый  $n_2$ -столбец (4-й,  $\mu_2 = 1$ ). Ищем первый линейно зависимый  $N$ -столбец слева направо – это соответствует  $n_2$ -столбцу (8-й,  $\mu_2 + 1$ ), следующий линейно зависимый  $N$ -столбец слева направо – это  $n_1$ -столбец (11-й,  $\mu_1 + 1$ ) и последний линейно зависимый  $N$ -столбец – это  $n_2$ -столбец (12-й).

Таким образом, из анализа вида матрицы  $r$  следует, что необходимо в матрице  $S$  оставить столбцы с 1-го по 8-й, и 8-й столбец – это  $\mu_2 + 1$  – первый линейно зависимый столбец, и ввиду того, что две последние строки нулевые, их отбрасываем и обозначим через  $S_1$ . В результате получили матрицу

$$S_1 = \begin{array}{cccccccc|cccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & & & & \\ \left[ \begin{array}{cccccccc|cccc} 2 & 0 & -20 & 3 & 0 & 0 & 0 & 0 \\ 0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \\ 5 & 0 & -2 & 6 & 2 & 0 & -20 & 3 \\ 0 & 12 & 1 & 3 & 0 & 4 & 2 & 1 \\ 2 & 0 & 4 & 0 & 5 & 0 & -2 & 6 \\ 0 & 9 & 0 & 2 & 0 & 12 & 1 & 3 \\ 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 \\ 0 & 2 & 0 & 0 & 0 & 9 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \end{array} \right] & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ d1 & d2 & n1 & n2 & d1 & d2 & n1 & n2 \end{array},$$

используя которую найдем вектор из ядра с помощью команды Matlab:

```
>> S1=[d1 0 0; d2 0 0; n1 0 0; n2 0 0; 0 0 d1; 0 0 d2; 0 0 n1; 0 0 n2]';
x2=null(S1); % вектор из ядра
x2b= x2/ x2(8) % Нормирование вектора – 8-й элемент равен единице
```

В результате получим

$$x_{2b} = [7 \quad -1 \quad 1 \quad 2 \quad -4 \quad 0 \quad 2 \quad 1]^T.$$

Первый слева линейно зависимый 8-й столбец использовали. Следующий линейно зависимый столбец – 11-й, который соответствует  $\mu_1 + 1$  – первый линейно зависимый  $n_1$ -столбец: матрицу  $S_2$  формируем из столбцов слева направо по 11-й включительно, кроме 8-го:

$$S_2 = \begin{array}{ccccccccccc|c}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 9 & 10 & 11 & \\
\hline
2 & 0 & -20 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
5 & 0 & -2 & 6 & 2 & 0 & -20 & 0 & 0 & 0 & 0 & 3 \\
0 & 12 & 1 & 3 & 0 & 4 & 2 & 0 & 0 & 0 & 0 & 4 \\
2 & 0 & 4 & 0 & 5 & 0 & -2 & 2 & 0 & -20 & 0 & 5 \\
0 & 9 & 0 & 2 & 0 & 12 & 1 & 0 & 4 & 2 & 0 & 6 \\
0 & 0 & 0 & 0 & 2 & 0 & 4 & 5 & 0 & -2 & 0 & 7 \\
0 & 2 & 0 & 0 & 0 & 9 & 0 & 0 & 12 & 1 & 0 & 8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 9 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 9 & 0 & 0 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 12 \\
\hline
d1 & d2 & n1 & n2 & d1 & d2 & n1 & d1 & d2 & n1 & 
\end{array}$$

Получили матрицу размером  $12 \times 10$ , у которой линейно зависимый 11-й столбец выделен другим цветом. Для определения вектора из ядра используем, как и выше, команды

```
>> d1=[2 0 5 0 2 0 0 0]; d2=[0 4 0 12 0 9 0 2];
    n1=[-20 2 -2 1 4 0 0 0]; n2=[3 1 6 3 0 2 0 0];
    S2=[d1 0 0 0 0; d2 0 0 0 0; n1 0 0 0 0;
        n2 0 0 0 0; 0 0 d1 0 0; 0 0 d2 0 0;
        0 0 n1 0 0; 0 0 0 0 d1; 0 0 0 0 d2;
        0 0 0 0 n1]';
    x1=null(S2);
    x1b= x1/ x1(10)
```

В результате получим

$$x_{1b} = [10 \quad -0,5 \quad 1 \quad 0 \quad 1 \quad 0 \quad 2,5 \quad -2 \quad 0 \quad 1]'$$

Таким образом, мы имеем

$$\begin{bmatrix} \frac{-\bar{N}_0}{\bar{D}_0} \\ \frac{-\bar{N}_1}{\bar{D}_1} \\ \frac{-\bar{N}_2}{\bar{D}_2} \end{bmatrix} = \begin{bmatrix} \frac{-\bar{n}_{11}^0}{\bar{d}_{11}^0} & \frac{-\bar{n}_{12}^0}{\bar{d}_{12}^0} \\ \frac{-\bar{n}_{21}^0}{\bar{d}_{21}^0} & \frac{-\bar{n}_{22}^0}{\bar{d}_{22}^0} \\ \frac{-\bar{n}_{11}^1}{\bar{d}_{11}^1} & \frac{-\bar{n}_{12}^1}{\bar{d}_{12}^1} \\ \frac{-\bar{n}_{21}^1}{\bar{d}_{21}^1} & \frac{-\bar{n}_{22}^1}{\bar{d}_{22}^1} \\ \frac{-\bar{n}_{11}^2}{\bar{d}_{11}^2} & \frac{-\bar{n}_{12}^2}{\bar{d}_{12}^2} \\ \frac{-\bar{n}_{21}^2}{\bar{d}_{21}^2} & \frac{-\bar{n}_{22}^2}{\bar{d}_{22}^2} \end{bmatrix} = \begin{matrix} x_{1b} & x_{2b} \\ \begin{bmatrix} 10 & 7 \\ -0,5 & -1 \\ 1 & 1 \\ 0 & 2 \\ 1 & 4 \\ 0 & 0 \\ 2,5 & 2 \\ 0 & 1 \\ -2 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \end{matrix}.$$

Можем выписать взаимно простые полиномиальные матрицы  $\bar{D}(s)$  и  $\bar{N}(s)$ :

$$\bar{D}(s) = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}}_{\bar{D}_0} + \underbrace{\begin{bmatrix} 2,5 & 2 \\ 0 & 1 \end{bmatrix}}_{\bar{D}_1} s + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}}_{\bar{D}_2} s^2 = \begin{bmatrix} s^2 + 2,5s + 1 & 2s + 1 \\ 0 & s + 2 \end{bmatrix},$$

$$\bar{N}(s) = \underbrace{\begin{bmatrix} -10 & -7 \\ 0,5 & 1 \end{bmatrix}}_{\bar{N}_0} + \underbrace{\begin{bmatrix} -1 & 4 \\ 0 & 0 \end{bmatrix}}_{\bar{N}_1} s + \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}}_{\bar{N}_2} s^2 = \begin{bmatrix} 2s^2 - s - 10 & 4s - 7 \\ 0,5 & 1 \end{bmatrix}.$$

Таким образом, выпишем правое взаимно простое полиномиальное представление матрицы передаточной функции  $\bar{D}(s)$  и  $\bar{N}(s)$ :

$$W_{ob}(s) = \bar{N}(s)\bar{D}^{-1}(s) = \begin{bmatrix} (2s-5)(s+2) & 4s-7 \\ 0,5 & 1 \end{bmatrix} \times \begin{bmatrix} (s+2)(s+0,5) & 2s+1 \\ 0 & s+2 \end{bmatrix}^{-1}.$$

**Пример 2.** Для двухканального динамического объекта в виде двух масс [10, 11], координаты которых  $y_1$  и  $y_2$  отсчитываются от состояния равновесия, и подвешенных последовательно на двух пружинах жесткости  $k_1$  и  $k_2$  решим задачу приведения описания к взаимно простому полиномиальному виду (для упрощения предполагаем отсутствие демпфирования). На каждую массу  $m_1$  и  $m_2$  действует управляемая внешняя сила  $u_1$  и  $u_2$ . Найдем взаимно простые полиномиальные матрицы передаточной функции, которая дана в виде левого полиномиального разложения

$$W_{ob}(s) = D^{-1}(s)N(s) = \begin{bmatrix} 6s^2 + 3 & -2 \\ -2 & 2s^2 + 2 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

где

$$D(s) = \underbrace{\begin{bmatrix} 3 & -2 \\ -2 & 2 \end{bmatrix}}_{D_0} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{D_1} s + \underbrace{\begin{bmatrix} 6 & 0 \\ 0 & 2 \end{bmatrix}}_{D_2} s^2, \quad (6)$$

$$N(s) = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{N_0} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{N_1} s + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{N_2} s^2. \quad (7)$$

Сформируем матрицу Сильвестра (2)

$$\underbrace{\begin{pmatrix} D_0 & N_0 & O & O \\ D_1 & N_1 & D_0 & N_0 \\ D_2 & N_2 & D_1 & N_1 \\ O & O & D_2 & N_2 \end{pmatrix}}_S \begin{pmatrix} -\bar{N}_0 \\ -\bar{D}_0 \\ -\bar{N}_1 \\ -\bar{D}_1 \end{pmatrix} = 0, \quad (8)$$

где  $O$  – нулевая матрица размером  $p \times p = 2 \times 2$ ,  $\bar{N}_i$  и  $\bar{D}_i$  – искомые взаимно простые полиномиальные матрицы. Если подставить значения элементов  $D_i$  и  $N_i$  из (6) и (7) в (8) и раскрыть их, то получим

$$\underbrace{\begin{pmatrix} d1 & d2 & n1 & n2 & d1 & d2 & n1 & n2 \\ 3 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ -2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & -2 & 2 & 0 & 1 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}}_S \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{pmatrix} \begin{pmatrix} -\bar{n}_{11}^0 & -\bar{n}_{12}^0 \\ -\bar{n}_{21}^0 & -\bar{n}_{22}^0 \\ \bar{d}_{11}^0 & \bar{d}_{12}^0 \\ \bar{d}_{21}^0 & \bar{d}_{22}^0 \\ -\bar{n}_{11}^1 & -\bar{n}_{12}^1 \\ -\bar{n}_{21}^1 & -\bar{n}_{22}^1 \\ \bar{d}_{11}^1 & \bar{d}_{12}^1 \\ \bar{d}_{21}^1 & \bar{d}_{22}^1 \end{pmatrix} = 0,$$

где  $d_1, d_2, n_1$  и  $n_2$  – столбцы размером шесть. Процедура формулирования матрицы  $S$  и  $x$  делается аналогично, как в первом примере. Также для определения линейно независимых столбцов в порядке слева направо в матрице  $S$  используем  $QR$ -разложение:

>> d1=[3 -2 0 0 6 0]; d2=[-2 2 0 0 0 2]; % столбики размером шесть

n1=[1 0 0 0 0 0]; n2=[0 1 0 0 0 0];

S=[d1 0 0; d2 0 0; n1 0 0; n2 0 0];

0 0 d1; 0 0 d2; 0 0 n1; 0 0 n2]; % транспонирование!

[q, r]=qr(S); % Вычисление ортогональной матрицы q и матрицы

% r – верхнетреугольной таких, что  $q^{-1}S = r$

r % вывод матрицы r

В результате получим матрицу  $r$ :

$$r = \begin{array}{cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ & d1 & d2 & n1 & n2 & d1 & d2 & n1 & n2 \\ \left[ \begin{array}{cccccccc} -7 & * & * & * & * & * & * & * & * \\ & -3,2 & * & * & * & * & * & * & * \\ & & -0,8 & * & * & * & * & * & * \\ & & & -0,7 & * & * & * & * & * \\ & & & & -7 & * & * & * & * \\ & & & & & & -0,3 & * & * \\ & & & & & & & 0,8 & * \\ & & & & & & & & -0,7 \end{array} \right]. \end{array}$$

Все элементы  $d_i$  и  $n_i$  на диагонали – ненулевые, и это подтверждает, что все столбцы линейно не зависимые и полиномиальные матрицы  $D(s)$  и  $N(s)$  *взаимно простые*. Задача проверки взаимной простоты выполнена.

**Пример 3.** В качестве объекта возьмем камеру полимерной покраски, используемой в технологическом процессе покраски узлов и деталей на лифто-строительном производстве. Камера покраски, представляющая собой большой контейнер с движущимися подвешенными деталями, состоит из четырех соединенных между собой секций. Передаточная функция четырехканального объекта приведена в виде левого полиномиального разложения [8, с. 142]

$$W_{ob}(s) = D^{-1}(s)N(s) =$$

$$= \begin{bmatrix} s+0,1 & -0,015 & 0 & 0 \\ 0,01 & s+0,1 & -0,015 & 0 \\ 0 & 0,01 & s+0,1 & 0,01 \\ 0 & 0 & -0,015 & s+0,1 \end{bmatrix}^{-1} \begin{bmatrix} 0,1 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0 \\ 0 & 0 & 0,1 & 0 \\ 0 & 0 & 0 & 0,1 \end{bmatrix}.$$

Выпишем матрицы  $D(s)$  и  $N(s)$  в виде полиномов:

$$D(s) = \underbrace{\begin{bmatrix} 0,1 & -0,015 & 0 & 0 \\ 0,01 & 0,1 & -0,015 & 0 \\ 0 & 0,01 & 0,1 & 0,01 \\ 0 & 0 & -0,015 & 0,1 \end{bmatrix}}_{D_0} + \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{D_1} s; \quad (9)$$

$$N(s) = \underbrace{\begin{bmatrix} 0,1 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0 \\ 0 & 0 & 0,1 & 0 \\ 0 & 0 & 0 & 0,1 \end{bmatrix}}_{N_0} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{N_1} s \quad (10)$$

и проверим их на взаимную простоту, одновременно при этом вычисляется правое взаимно простое разложение. Но в данном случае этого можно не делать ввиду диагонального вида матрицы  $N(s)$ . Отметим, что полиномиальные матрицы  $N_i$  и  $D_i$  имеют размеры  $p \times p = 4 \times 4$ . Запишем систему линейных уравнений (2)

$$\underbrace{\begin{pmatrix} D_0 & N_0 \\ D_1 & N_1 \end{pmatrix}}_S \underbrace{\begin{pmatrix} -\bar{N}_0 \\ \bar{D}_0 \end{pmatrix}}_x = 0, \quad (11)$$

где  $\bar{N}_i$  и  $\bar{D}_i$  размером  $p \times p = 4 \times 4$  – искомые взаимно простые полиномиальные матрицы. Если подставить значения элементов  $D_i$  и  $N_i$  из (9) и (10) в (11) и раскрыть, то (11) принимает следующий вид:



$$\begin{pmatrix}
 d1 & d2 & d3 & d4 & n1 & n2 & n3 & n4 \\
 0,1 & -0,015 & 0 & 0 & 0,1 & 0 & 0 & 0 \\
 0,01 & 0,1 & -0,015 & 0 & 0 & 0,1 & 0 & 0 \\
 0 & 0,01 & 0,1 & 0,01 & 0 & 0 & 0,1 & 0 \\
 0 & 0 & -0,015 & 0,1 & 0 & 0 & 0 & 0,1 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
 \end{pmatrix}
 \begin{bmatrix}
 -\bar{n}_{11}^0 & -\bar{n}_{12}^0 & -\bar{n}_{13}^0 & -\bar{n}_{14}^0 \\
 -\bar{n}_{21}^0 & -\bar{n}_{22}^0 & -\bar{n}_{23}^0 & -\bar{n}_{24}^0 \\
 -\bar{n}_{31}^0 & \bar{d}_{32}^0 & -\bar{n}_{33}^0 & \bar{d}_{34}^0 \\
 -\bar{n}_{42}^0 & -\bar{n}_{42}^0 & -\bar{n}_{43}^0 & -\bar{n}_{44}^0 \\
 \bar{d}_{11}^0 & \bar{d}_{12}^0 & \bar{d}_{13}^0 & \bar{d}_{14}^0 \\
 \bar{d}_{21}^0 & \bar{d}_{22}^0 & \bar{d}_{23}^0 & \bar{d}_{24}^0 \\
 \bar{d}_{31}^0 & \bar{d}_{32}^0 & \bar{d}_{33}^0 & \bar{d}_{34}^0 \\
 \bar{d}_{41}^0 & \bar{d}_{42}^0 & \bar{d}_{43}^0 & \bar{d}_{44}^0
 \end{bmatrix} = 0.$$

Здесь  $d_i, i=1, \dots, 4$ , и  $n_i, i=1, \dots, 4$ , – столбцы размером восемь. Как и в предыдущих примерах, ищем линейно независимые столбцы в матрице  $S$  в порядке слева направо:

```

>> d1=[0.1 0.01 0 0 1 0 0 0]; d2=[-0.015 0.1 0.01 0 0 1 0 0];
    d3=[0 -0.015 0.1 -0.015 0 0 1 0]; d4=[0 0 0.01 0.1 0 0 0 1]; % столбцы разме-
                                % ром восемь
n1=[0.1 0 0 0 0 0 0 0]; n2=[0 0.1 0 0 0 0 0 0];
n3=[0 0 0.1 0 0 0 0 0]; n4=[0 0 0 0.1 0 0 0 0];
S=[d1; d2; d3; d4; n1; n2; n3; n4]'; % транспонирование!
[q, r]=qr(S); %
r % вывод матрицы r
    
```

В результате получим матрицу  $r$ :

$$r = \begin{bmatrix}
 -1,005 & * & * & * & * & * & * & * \\
 & -1,005 & * & * & * & * & * & * \\
 & & -1,005 & * & * & * & * & * \\
 & & & -1,005 & * & * & * & * \\
 & & & & 0,1 & * & * & * \\
 & & & & & 0,1 & * & * \\
 & & & & & & 0,1 & * \\
 & & & & & & & -0,1
 \end{bmatrix}$$

Из вида матрицы  $r$  следует, что все элементы  $d_i$  и  $n_i$  на диагонали – ненулевые, что подтверждает линейную независимость столбцов. Делаем вывод, что полиномиальные матрицы  $D(s)$  и  $N(s)$  *взаимно простые*. Задача проверки взаимной простоты выполнена.

Приведем теорему, на основании которой разработаем алгоритм вычисления взаимно простого разложения матричной передаточной функции. Это необходимо сделать ввиду того, что *алгоритм* вычисления в [2] *не приведен*.

**Теорема 1** [2, р. 219]. Пусть  $W_{ob}(s) = D^{-1}(s)N(s)$  – левое матричное полиномиальное разложение, не обязательно взаимно простое. Используем коэффициентные матрицы  $D(s)$  и  $N(s)$  для формирования матрицы Сильвестра  $S$  и исследуем линейную независимость столбцов слева направо. Пусть  $\mu_i$ ,  $i = 1, 2, \dots, p$ , будет число линейно независимых  $N_i$ -столбцов. Тогда

$$\deg W_{ob}(s) = \mu_1 + \mu_2 + \dots + \mu_p,$$

и правое взаимно простое разложение  $\bar{N}(s)\bar{D}^{-1}(s)$  может быть получено вычислением  $p$  «штук» нормированных векторов из ядра  $p$  «штук» матриц, сформированных из каждого первого зависимого  $N_i$ -столбца и всех столбцов, линейно независимых слева от него.

### 3. АЛГОРИТМ ВЫЧИСЛЕНИЯ ВЗАИМНО ПРОСТОГО РАЗЛОЖЕНИЯ ПЕРЕДАТОЧНОЙ ФУНКЦИИ

Предполагаем, что задано левое матричное полиномиальное разложение (представление) передаточной функции

$$W_{ob}(s) = D^{-1}(s)N(s),$$

где

$$D(s) = \sum_{i=0}^n D_i s^i, \quad N(s) = \sum_{i=0}^m N_i s^i$$

такие, что  $\deg N(s) \leq \deg D(s)$ ,  $m \leq n$ ,  $\dim D_i = \dim N_j = p \times p$ , где  $p$  – число каналов. Отметим, что  $n \geq 1$ . В дальнейшем используем понятия « $N$ -блочные столбцы» и « $D$ -блочные столбцы»:

$$N = \left( N_0^t, N_1^t, \dots, N_n^t \right)^t = (n_1 \ n_2 \ \dots \ n_p), \quad D = \left( D_0^t, D_1^t, \dots, D_n^t \right)^t = (d_1 \ d_2 \ \dots \ d_p).$$

Два блочных столбца  $D$  и  $N$ , стоящие рядом, дополняются нулевыми матрицами ( $n \geq 2!$ ) сверху и снизу так, чтобы число (блочных) элементов в столбцах равнялось  $2n$  блокам. В алгоритме осуществляется исследование уравнения (2), включающее в себя матрицу Сильвестра  $S$  и вектор матричных коэффициентов из взаимно простого представления матричной передаточной функции

$$x = (-\bar{N}_0^t \quad \bar{D}_0^t \mid -\bar{N}_1^t \quad \bar{D}_1^t \quad \vdots \quad -\bar{N}_{n-1}^t \quad \bar{D}_{n-1}^t)^t = (x_1 \ x_2 \ \dots \ x_p).$$

Предполагается использование пакета Matlab.

**Алгоритм** состоит из следующих действий.

1. Составление матрицы Сильвестра  $S$  размером  $2n \times 2n$  блоков: первый и второй  $D$ - и  $N$ -блочные столбцы, стоящие рядом, дополняем снизу нулевыми матрицами для того, чтобы количество строк равнялось  $2n$  блокам. Следующие блочные столбцы, соответствующие 3-му и 4-му столбцам, получим из 1-го и 2-го блочных столбцов смещением вниз на одну позицию и дополнением сверху нулевыми матрицами. Аналогичным образом строим  $2n$  «штук»  $D$ - и  $N$ -блочных столбцов. Матрица Сильвестра сформирована. Формируем блочный столбец  $x$  из матричных коэффициентов полиномов  $\bar{N}(s)$  и  $\bar{D}(s)$  размером  $2n \times 1$  блоков.

2. Вычисление линейно зависимых столбцов матрицы  $S$  в направлении слева направо, для чего используем команду  $[q, r] = qr(S)$ , которая выдает две матрицы –  $q$  и  $r$ ; для дальнейших вычислений матрица  $q$  не требуется. Наличие линейно зависимых столбцов соответствует нулевым элементам на диагонали в верхнетреугольной матрице  $r$ .

3. Если все диагональные элементы матрицы  $r$  ненулевые, то полиномиальные матрицы  $N(s)$  и  $D(s)$  взаимно простые и, следовательно, **выход из алгоритма**; если имеются нулевые элементы – ищем взаимно простые полиномиальные матрицы  $\bar{N}(s)$  и  $\bar{D}(s)$  – переход на следующий шаг.

4. Формирование матрицы  $S_1$  :

– так как все  $n$  «штук»  $D$ -блочных столбцов матрицы  $S$  линейно независимы от всех столбцов слева, определяем количества  $\mu_1, \mu_2, \dots, \mu_p$  линейно независимых  $n_1, n_2, \dots, n_p$ -столбцов слева направо в матрице  $S$  ;

– матрица  $S_1$  состоит из линейно независимых столбцов матрицы  $S$ , взятых слева направо, включая первый линейно зависимый  $n_1$  ( $\mu_1 + 1$ -столбец),  $n_2$  ( $\mu_2 + 1$ -столбец), ...,  $n_p$  ( $\mu_p + 1$ -столбец) – остается первый линейно зависимый столбец;

– если в матрице  $S_1$  имеется нулевая/нулевые строка/строки, то ее/их вычеркиваем – матрица  $S_1$  сформирована.

5. Формируем вектор  $x_1$ , или  $x_2$ , или ...  $x_p$  : это зависит от того, какой столбец  $n_1$ , или  $n_2$ , или ...,  $n_p$  линейно зависимый оставили в матрице  $S_1$  : если  $n_1$ , то найдем  $x_1$ , а если  $n_2$ , то  $x_2$  и т. д. У блочного вектора  $x_i$  вычеркиваем столько последних элементов, сколько столбцов вычеркнули при формировании матрицы  $S_1$  .

6. Для определения ненулевого решения  $x$  (состоит из матричных коэффициентов взаимно простых полиномов числителя и знаменателя матричной передаточной функции уравнения  $S_1 x_1 = 0$ , или  $S_1 x_2 = 0$ , или ...  $S_1 x_p = 0$ ) воспользуемся командой Matlab **null**( $S_1$ ) – вектор  $x_i$  принадлежит ядру  $S_1$  ;

– найденный вектор нормируем – элементы вектора  $x_i$  делим на последний элемент его же и обозначим через  $x_{ib}$  (можно назвать нормированным вектором из ядра – *monic null vector*).

7. Формирование матрицы  $S_2$ , состоящей из линейно независимых столбцов матрицы  $S$  (предыдущий линейно зависимый столбец удаляется), взятых слева направо, включая первый линейно зависимый  $n_i$ -столбец (если в матрице  $S_1$   $n_2$ -столбец был линейно зависимый, то в матрице  $S_2$  первый линейно зависимый столбец –  $n_1$  или наоборот).

8. Действия шагов 4–7 выполняем для всех каналов, т. е. проводим вычисления для матриц  $S_1, S_2, \dots, S_p$ .

9. Выпишем взаимно простые полиномиальные матрицы передаточной функции  $\bar{N}(s)$  и  $\bar{D}(s)$ . Те столбцы, которые удалили из матриц

$S_1, S_2, \dots, S_p$ , соответствующие элементам векторов  $x_1, x_2, \dots, x_p$ , заполняем нулями.

Конец алгоритма.

## ЗАКЛЮЧЕНИЕ

В теории автоматического управления используются различные виды описания объектов, например: в виде матричных передаточных функций  $W(s)$ , в пространстве состояний  $(A, B, C)$  и в виде матричных полиномиальных разложений  $W(s) = D_l^{-1}(s)N_l(s) = N_r(s)D_r^{-1}(s)$  – левого и правого разложения [10, 14, 22, 23].

В работе предложен формализованный алгоритм поиска взаимно простого правого полиномиального матричного разложения  $N(s)$  и  $D(s)$  при использовании матрицы *Сильвестра* и команды в Matlab, базирующийся на теореме [Theorem 7.M4, p.219, 2]. Методики поиска взаимно простого разложения, рассмотренные в различных работах (например, в [1, 2], а также в диссертационных работах [7, 8]), приведены в недостаточно формализованном и четком виде, что вызывает определенные затруднения.

Задача взаимной простоты играет важную роль при решении задачи синтеза как в одноканальных системах [9, 16, 17, 25], так и в многоканальных системах [13, 18–20, 24]: взаимная простота полиномиальных матриц позволяет использовать так называемую матрицу Сильвестра или ее незначительные модификации при синтезе многоканальных регуляторов. Кроме того, если полиномы числителя и знаменателя передаточной функции объекта в одноканальных системах имеют одинаковые корни или полиномиальные матрицы не взаимно простые (имеют общий матричный полиномиальный не унимодальный множитель), это может привести к скрытой неустойчивости САУ, или неуправляемости / ненаблюдаемости системы. Если матрица Сильвестра не вырожденная, а это соответствует взаимной простоте полиномиального матричного разложения, то существует решение задачи синтеза многоканального регулятора, сводящееся к решению линейного алгебраического уравнения. С другими словами находят векторы из ядра матрицы Сильвестра. Для этого используется *QR*-разложение с целью поиска линейно зависимых столбцов.

На основе анализа нескольких примеров поиска взаимно простых правых полиномиальных матричных разложений  $\bar{N}(s)$  и  $\bar{D}(s)$  сформулирован *алгоритм вычисления взаимно простого разложения*.

**СПИСОК ЛИТЕРАТУРЫ**

1. *Kailath T.* Linear systems. – Englewood Cliffs, NJ: Prentice Hall, 1980. – 350 p.
2. *Chen C.T.* Linear system theory and design. – 3<sup>rd</sup> ed. – New York: Oxford University Press, 1999. – 334 p.
3. *Гайдук А.Р., Беляев В.Е., Пьявченко Т.А.* Теория автоматического управления в примерах и задачах с решениями в MATLAB: учебное пособие. – 2-е изд., испр. – СПб.: Лань, 2011. – 464 с.
4. *Ким Д.П.* Теория автоматического управления. Т. 2. Многомерные, нелинейные, оптимальные и адаптивные системы. – М.: Физматлит, 2004. – 464 с.
5. *Гайдук А.Р.* Теория и методы аналитического синтеза систем автоматического управления (полиномиальный подход). – М.: Физматлит, 2012. – 360 с.
6. *Гайдук А.Р.* Теория автоматического управления: учебник. – М.: Высшая школа, 2010. – 415 с.
7. *Вороной В.В.* Полиномиальный метод расчета многоканальных регуляторов пониженного порядка: дис. ... канд. техн. наук: 05.13.01. – Новосибирск, 2013. – 173 с.
8. *Шоба Е.В.* Модальный метод синтеза многоканальных динамических систем с использованием полиномиального разложения: дис. ... канд. техн. наук: 05.13.01. – Новосибирск, 2013. – 192 с.
9. *Воевода А.А., Ижицкая Е.А.* Стабилизация двухмассовой системы: модальный метод синтеза // Сборник научных трудов НГТУ. – 2009. – № 2 (56). – С. 3–10.
10. *Воевода А.А., Бобобеков К.М.* Решение переопределенной линейной системы уравнений при полиномиальном синтезе регуляторов // Современные технологии. Системный анализ. Моделирование. – 2017. – № 4 (56). – С. 84–99.
11. *Бобобеков К.М., Воевода А.А.* Синтез двухканальной системы полиномиальным методом: обеспечение астатизма // Сборник научных трудов НГТУ. – 2016. – № 1 (83). – С. 7–19.
12. *Воевода А.А.* Стабилизация двухмассовой системы: полиномиальный метод синтеза двухканальной системы // Сборник научных трудов НГТУ. – 2009. – № 4 (58). – С. 121–124.
13. *Воевода А.А., Вороной В.В.* Полиномиальный метод расчета многоканальных регуляторов заданной структуры // Научный вестник НГТУ. – 2013. – № 2 (51). – С. 214–218.

14. *Воевода А.А., Бобобеков К.М.* Синтез линейных многоканальных регуляторов с использованием структурных преобразований // Вестник Астраханского государственного технического университета. – 2017. – № 3. – С. 7–20.
15. *Бобобеков К.М., Тауров Э.Ш.* Вычисление взаимно простого разложения для одноканальных передаточных функций с использованием матрицы Сильвестра // Сборник научных трудов НГТУ. – 2018. – № 1 (91). – С. 7–30.
16. *Бобобеков К.М.* Полиномиальный метод синтеза одноканальной двухмассовой системы // Сборник научных трудов НГТУ. – 2016. – № 4 (86). – С. 25–36.
17. *Бобобеков К.М.* О нормировании полиномов знаменателей объекта и регулятора при полиномиальном методе синтеза // Сборник научных трудов НГТУ. – 2016. – № 4 (86). – С. 7–24.
18. *Воевода А.А.* Стабилизация двухмассовой системы: модальный метод синтеза с использованием полиномиального разложения // Научный вестник НГТУ. – 2010. – № 1 (38). – С. 195–198.
19. *Воевода А.А., Вороной В.В., Шоба Е.В.* Модальный синтез многоканального регулятора пониженного порядка с использованием «обратной» производной на примере трехмассовой системы // Научный вестник НГТУ. – 2012. – № 1 (46). – С. 15–22.
20. *Воевода А.А., Вороной В.В.* Модальный синтез регуляторов пониженного порядка методом дифференцирования характеристического полинома // Сборник научных трудов НГТУ. – 2011. – № 1 (63). – С. 3–12.
21. The modeling tests of the new PID-regulators structures / A.A. Voevoda, V.A. Zhmud, R.Y. Ishimtsev, V.M. Semibalamut // Proceedings of the 18th IASTED International Conference on Applied Simulation and Modelling, ASM 2009, 7–9 September, 2009, Palma de Mallorca, Spain. – [S. l.], 2009. – P. 165–168.
22. *Воевода А.А.* Матричные передаточные функции: (основные понятия): конспект лекций по курсу «Проектирование систем управления» для 4–5 курсов АВТФ (специальность 2101) / Новосибирский государственный технический университет. – Новосибирск: Изд-во НГТУ, 1994. – 94 с.
23. *Бобобеков К.М.* О структурных преобразованиях многоканальных линейных систем в матричном полиномиальном представлении // Научный вестник НГТУ. – 2017. – № 2 (67). – С. 7–25.
24. *Воевода А.А., Чехонадских А.В., Шоба Е.В.* Модальный метод синтеза с использованием полиномиального разложения: разделение движений при стабилизации трехмассовой системы // Научный вестник Новосибирского государственного технического университета. – 2011. – № 2 (43). – С. 39–46.
25. *Воевода А.А., Бобобеков К.М.* О необходимом условии существования решения при полиномиальном методе синтеза одноканальных систем // Сборник научных трудов НГТУ. – 2017. – № 4 (90). – С. 7–21.

**Воевода Александр Александрович**, доктор технических наук, профессор кафедры автоматике Новосибирского государственного технического университета. Основное направление научных исследований – управление многоканальными объектами. Имеет более 200 публикаций. E-mail: ucit@ucit.ru

**Бобобеков Курбонмурод Мулломиракович**, специалист по технологиям машиностроения, 2008–2013 гг. – кафедра «Технология машиностроения, металлорежущие станки и инструменты» механико-технологического факультета Таджикского технического университета (ТТУ) им. акад. М.С. Осими. С 2013 по 2015 г. ассистент Таджикского технического университета. С 2015 г. аспирант кафедры автоматике Новосибирского государственного технического университета. В настоящее время специализируется в области синтеза систем управления техническими системами. Имеет более 25 публикаций. E-mail: kurbon\_111@mail.ru

DOI: 10.17212/2307-6879-2018-2-7-35

## **Transformation of a polynomial representation of a multichannel object to a coprime form\***

**A.A. Voevoda<sup>1</sup>, K.M. Bobobekov<sup>2</sup>**

<sup>1</sup> *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, D. Sc. (Eng.), professor. E-mail: ucit@ucit.ru*

<sup>2</sup> *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, the post-graduate student of Department Automatics. E-mail: kurbon\_111@mail.ru*

Is considered bringing of the matrix polynomial representation of the transfer function of the object to a coprime form. Disarrangement requirements coprime representation leads to the appearance in the same roots "numerator" and "denominator" as in single-channel systems and in multi-channel systems. This can lead to latent instability of automatic control systems, unmanageability or non-observability; to eliminate such troubles, it can sometimes be ruled out by introducing additional restrictions on specifying the roots of the desired characteristic polynomial of the closed system. In addition, for the existence of a solution of a polynomial matrix equation in the search for a regulator, a mutually simple expansion of the transfer function of the object is required. To check for mutual simplicity, a system of linear homogeneous equations is compiled: in other words, the Sylvester matrix and sought-for matrix vector of the "elements" of the coprime representation are constructed. Then, using the  $[q, r] = qr(S)$  command of the Matlab package, where  $S$  is the Sylvester matrix, the linearly dependent columns are checked in the direction from left to right. If the diagonal elements of  $r$  are non-zero, then the original polynomial matrices are coprime, and if there are zeros on the diagonal, we look for the right mutually simple polynomial matrices. An algorithm for calculating a coprime representation is presented and three examples illustrating the proposed algorithm are given.

---

\* Received 10 January 2018.



**Keywords:** matrix transfer function, left and right matrix polynomial decomposition, coprime decomposition, the not coprime decomposition, polynomial matrices, linear equation, Sylvester matrix, the singular matrix, search algorithm coprime

## REFERENCES

1. Kailath T. *Linear systems*. Englewood Cliffs, NJ, Prentice Hall, 1980. 350 p.
2. Chen C.T. *Linear system theory and design*. 3<sup>rd</sup> ed. New York, Oxford University Press, 1999. 334 p.
3. Gaiduk A.R., Belyaev V.E., P'yavchenko T.A. *Teoriya avtomaticheskogo upravleniya v primerakh i zadachakh s resheniyami v MATLAB* [Theory of automatic control in examples and problems with solutions in MATLAB]. 2nd ed., rev. St. Petersburg, Lan' Publ., 2011. 464 p.
4. Kim D.P. *Teoriya avtomaticheskogo upravleniya. T. 2. Mnogomernye, nelineinye, optimal'nye i adaptivnye sistemy* [The theory of automatic control. Vol. 2. Multidimensional, nonlinear, optimal and adaptive systems]. Moscow, Fizmatlit Publ., 2004. 464 p.
5. Gaiduk A.R. *Teoriya i metody analiticheskogo sinteza sistem avtomaticheskogo upravleniya (polinomial'nyi podkhod)* [Theory and methods of analytical synthesis of automatic control systems (polynomial approach)]. Moscow, Fizmatlit Publ., 2012. 360 p.
6. Gaiduk A.R. *Teoriya avtomaticheskogo upravleniya* [The theory of automatic control]. Moscow, Vysshaya shkola Publ., 2010. 415 p.
7. Voronoi V.V. *Polinomial'nyi metod rascheta mnogokanal'nykh regulyatorov ponizhennogo poryadka*. Diss. kand. tekhn. nauk [A polynomial method for calculating the multi-channel controllers low order. PhD eng. sci. diss.]. Novosibirsk, 2013. 173 p.
8. Shoba E.V. *Modal'nyi metod sinteza mnogokanal'nykh dinamicheskikh sistem s ispol'zovaniem polinomial'nogo razlozheniya*. Diss. kand. tekhn. nauk [The modal method for the synthesis of multi-channel dynamic systems using a polynomial expansion. PhD eng. sci. diss.]. Novosibirsk, 2013. 192 p.
9. Voevoda A.A., Izhitskaya E.A. *Stabilizatsiya dvukhmassovoi sistemy: modal'nyi metod sinteza* [Stabilization of two-mass systems: modal synthesis method]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2009, no. 2 (56), pp. 3–10.
10. Voevoda A.A., Bobobekov K.M. *Reshenie pereopredelennoi lineinoi sistemy uravnenii pri polinomial'nom sinteze regulyatorov* [Solution of an overdetermined linear system of equations for polynomial synthesis of regulators]. *Sov-*

*remennye tekhnologii. Sistemyi analiz. Modelirovanie – Modern Technologies. System analysis. Modeling*, 2017, no. 4 (56), pp. 84–99.

11. Bobobekov K.M., Voevoda A.A. Sintez dvukhkanal'noi sistemy polinomial'nym metodom: obespechenie astatizma [Synthesis of two-channel system polynomial method: ensuring astatic]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 1 (83), pp. 7–19.

12. Voevoda A.A. Stabilizatsiya dvukhmassovoi sistemy: polinomial'nyi metod sinteza dvukhkanal'noi sistemy [Two-mass system stabilization: polynomial method of two-channel system synthesis]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2009, no. 4 (58), pp. 121–124.

13. Voevoda A.A., Voronoi V.V. Polinomial'nyi metod rascheta mnogokanal'nykh regulyatorov zadannoi struktury [Polynomial method for calculating multi-channel controllers of a given structure]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2013, no. 2 (51), pp. 214–218.

14. Voevoda A.A., Bobobekov K.M. Sintez lineinykh mnogokanal'nykh regulyatorov s ispol'zovaniem strukturnykh preobrazovaniy [Synthesis of linear multi-channel regulators using structural transformations]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta – Vestnik of Astrakhan State Technical University*, 2017, no. 3, pp. 7–20.

15. Bobobekov K.M., Taurov E.Sh. Vychislenie vzaimno prostogo razlozheniya dlya odnokanal'nykh peredatochnykh funktsii s ispol'zovaniem matritsy Sil'vestra [Calculation of a mutually simple expansion for single-channel transfer functions using the Sylvester matrix]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 1 (91), pp. 7–30.

16. Bobobekov K.M. Polinomial'nyi metod sinteza odnokanal'noi dvukhmassovoi sistemy [A polynomial method for the synthesis of single-channel two-mass system]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 4 (86), pp. 25–36.

17. Bobobekov K.M. O normirovaniy polinomov znamenatelyi ob"ekta i regulyatora pri polinomial'nom metode sinteza [About rationing polynomials denominator object and regulator during polynomial method of synthesis]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 4 (86), pp. 7–24.

18. Voevoda A.A. Stabilizatsiya dvukhmassovoi sistemy: modal'nyi metod sinteza s ispol'zovaniem polinomial'nogo razlozheniya [Stabilisation of two-mass system by a modal method of synthesis with polynomial factorization]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2010, no. 1 (38), pp. 195–198.

19. Voevoda A.A., Voronoy V.V., Shoba E.B. Modal'nyi sintez mnogokanal'nogo regulatora ponizhennogo poryadka s ispol'zovaniem "obratnoi" proizvodnoi na primere trekhmassovoi sistemy [Modal synthesis of multi-channel low-order controller using the "reverse" derivative principle for three-mass system]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2012, no. 1 (46), pp. 15–22.

20. Voevoda A.A., Voronoy V.V. Modal'nyi sintez regulátorov ponizhennogo poryadka metodom differentsirovaniya kharakteristicheskogo polinoma [Modal design of reduced order controllers by method of differentiation of the characteristic polynomial]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2011, no. 1 (63), pp. 3–12.

21. Voevoda A.A., Zhmud V.A., Ishimtsev R.Y., Semibalamut V.M. The modeling tests of the new PID-regulators structures. *Proceedings of the IASTED International Conference on Applied Simulation and Modelling, ASM 2009*, Palma de Mallorca, Spain, 7–9 September, 2009, pp. 165–168.

22. Voevoda A.A. *Matrichnye peredatochnye funktsii: (osnovnye ponyatiya)* [Matrix transfer functions (basic concepts)]. Novosibirsk State Technical University. Novosibirsk, NSTU Publ., 1994. 94 p.

23. Bobobekov K.M. O strukturnykh preobrazovaniyakh mnogokanal'nykh lineinykh sistem v matrichnom polinomial'nom predstavlenii [About structural transformations of multichannel linear systems in the matrix polynomial representation]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2017, no. 2 (67), pp. 7–25.

24. Voevoda A.A., Chekhonadskikh A.V., Shoba E.V. Modal'nyi metod sinteza s ispol'zovaniem polinomial'nogo razlozheniya: razdelenie dvizhenii pri stabilizatsii trekhmassovoi sistemy [Modal synthesis method using a polynomial decomposition: the separation of motions in the stabilization of the three-mass plant]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2011, no. 2 (43), pp. 39–46.

25. Voevoda A.A., Bobobekov K.M. O neobkhodimom uslovii sushchestvovaniya resheniya pri polinomial'nom metode sinteza odnokanal'nykh sistem [About the necessary conditions of existence of the solution in polynomial method of syn-

thesis of single-channel systems]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2017, no. 4 (90), pp. 7–21.

Для цитирования:

Воевода А.А., Бобобеков К.М. Приведение матричного полиномиального представления передаточной функции к взаимно простому виду с использованием матрицы Сильвестра // Сборник научных трудов НГТУ. – 2018. – № 2 (92). – С. 7–35. – doi: 10.17212/2307-6879-2018-2-7-35.

For citation:

Voevoda A.A., Bobobekov K.M. Privedenie matrichnogo polinomial'nogo predstavleniya peredatochnoi funktsii k vzaimno prostomu vidu s ispol'zovaniem matritsy Sil'vestra [Transformation of a polynomial representation of a multichannel object to a coprime form]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 2 (92), pp. 7–35. doi: 10.17212/2307-6879-2018-2-7-35.

## НЕЛИНЕЙНЫЙ ОБЪЕКТ: ЛИНЕАРИЗАЦИЯ ОБРАТНЫМИ СВЯЗЯМИ\*

В.Ю. ФИЛЮШОВ

*630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, аспирант кафедры автоматики. E-mail: filiushov.vladislav@gmail.com*

В настоящее время набирают популярность нелинейные методы синтеза, которые используют исходную нелинейную модель. Такой подход дает возможность формировать управление не только в окрестности точки линеаризации, а в большей области изменения переменных состояния, что позволяет в законе управления учитывать поведение объекта наиболее полно. Для моделей, линеаризованных в окрестности некоторой точки, применяются линейные методы синтеза, такие как модальный метод синтеза, методы синтеза с использованием полиномиального разложения, алгоритмы оптимизации, построение ПИД-регуляторов и другие. Но такое представление не описывает всех свойств работы исходного объекта. Для того чтобы применять линейные методы синтеза для нелинейных объектов, целесообразней использовать линеаризацию обратной связью, которая приводит поведение исходного объекта к линейному виду. Суть метода заключается в поиске такого компенсирующего управления, при котором поведение замкнутой системы будет соответствовать поведению системы, описанной линейными дифференциальными уравнениями. Тогда для синтезированной системы появляется возможность рассчитать регулятор линейными методами синтеза. В настоящей работе осуществлен обзор предыдущих работ по данной теме. Рассмотрены различные одноканальные нелинейные объекты; объект, содержащий тригонометрическую функцию; показана возможность получения различного решения задачи линеаризации нелинейными обратными связями для одного и того же объекта. На примере подсистемы «маятник» модели объекта перевернутый маятник на тележке произведен синтез системы управления с модальным регулятором и наблюдателем состояния полного порядка, где в одном случае использовалась старшая производная выходной величины, а в другом – нет; показана возможность применения данного метода для многоканальных объектов (двух- и трехканальных), нелинейных по выходу, а также показан формализованный метод поиска линеаризующего управления на основе алгебры и скобок Ли.

**Ключевые слова:** нелинейное управление, линеаризация обратной связью, регулятор, наблюдатель состояния, синтез, многоканальный, алгебра Ли

---

\* Статья получена 19 февраля 2018 г.

## ВВЕДЕНИЕ

К достаточно новым методам синтеза линейных многоканальных объектов можно отнести стабилизацию с использованием *полиномиального разложения*. Полиномиальный метод расчета многоканальных регуляторов описан в работе [1], в которой приводится алгоритм построения управления. В [2] приведены примеры модального метода синтеза с использованием полиномиального разложения для двухмассовой системы. В [3] рассмотрена стабилизация двухканальной системы методом разделения движений с использованием полиномиального разложения, что сводится к решению диофантова уравнения. Также к многоканальным объектам относится перевернутый маятник на тележке. В работе [4] приведен пример синтеза ПИД-регуляторов для стабилизации вертикального положения маятника и перемещения тележки в заданное положение. В работе [5] на примере объекта второго порядка с ограничением на входе и с запаздыванием сопоставляются критерии устойчивости процедуры оптимизации. В работе [6] проведено моделирование новой структуры ПИД-регулятора.

Синтез нелинейных систем, в отличие от линейных, не так однозначен. Так, существуют системы, в которых нелинейная часть выражена существенными нелинейностями (реле, реле с гистерезисом и так далее) и с нелинейностями, выраженными гладкими функциями. В данной работе рассмотрим объекты с нелинейностями, выраженными гладкими функциями, такими как умножение, деление и взятие корня, причем управление аддитивно (так называемые *аффинные системы*). Эти функции могут связывать как переменные состояния, так и выходные величины.

Для синтеза систем такого вида применима *линеаризация обратной связью* – ЛОС (Linearization by output injection) [7–9], суть которой заключается в том, что нелинейная система, замкнутая искомым управлением, будет вести себя эквивалентно линейной. Линеаризация обратной связью применима для объектов, которые заданы нелинейными дифференциальными уравнениями с аддитивным входным воздействием, или для объектов, выходные величины которых нелинейно связаны с переменными состояния. В первом случае такая линеаризация называется *по состоянию*, а во втором случае – *по выходу*.

Компенсирующие воздействия могут быть в некоторых случаях найдены по структурной схеме модели. Преобразование структурных схем распространено в линейных системах, где они называются эквивалентными структурными преобразованиями. В нелинейных системах также можно преобразовать структурную схему путем переноса нелинейности через интегратор, но новая схема не будет полностью эквивалентна в связи с зависимостью пере-

ходных процессов нелинейных систем от начальных условий. Тем не менее такой подход возможен и рассмотрен в работах.

Рассмотрены модели объектов, имеющих многоканальную структуру с выходными переменными, нелинейно связанными друг с другом. Для такого типа объектов, к которому относится машина переменного тока, осуществлена линеаризация путем дифференцирования выхода и поиска взаимосвязи между управлением и нелинейностью. Данная взаимосвязь найдена, что позволило развязать выходные величины друг от друга и дало возможность применять различные линейные методы синтеза. Ограничения на переменные вызваны структурой самого объекта.

В заключительной части работы рассмотрим нахождение такого управления, которое позволяет линеаризовать исходную нелинейную модель, применяя дифференциально-геометрический подход. Данный подход заключается в использовании производных Ли (скобки Ли, алгебра Ли, группы Ли). Использование этого инструмента позволяет проводить анализ и синтез нелинейных систем. В частности, матрицы наблюдаемости и управляемости похожи на те, что получены от линеаризованного объекта. Также используется понятие о каноническом виде, который аналогичен каноническому виду в линейных системах. Производные Ли особенно удобно использовать для непрерывных дифференцируемых функций, что соответствует нашему классу объектов.

## ПОСТАНОВКА ЗАДАЧИ

Рассмотрим способы нахождения такого управления, которое позволяет линеаризовать исходную нелинейную модель. Эта задача будет выполнена различными способами – структурными преобразованиями и дифференциально-геометрическим подходом. Данный подход заключается в использовании производных Ли (скобки Ли, алгебра Ли, группы Ли) [8]. Для нелинейных объектов, линейных по управлению (*аффинных*) вида

$$\dot{x} = A(x) + B(x)u ,$$

возможно выразить старшую производную через управления (минимально фазовые), что и является критерием применимости этих методов, где  $u$  – гладкая функция в некоторой окрестности  $\Omega(0)$  начала координат. Начало координат является положением равновесия  $f(0, 0) = 0$ . Здесь  $x$  – вектор состояния,  $u$  – управление.

## 1. СТРУКТУРНЫЕ ПРЕОБРАЗОВАНИЯ

В линейных системах для удобства применения методов управления используют структурные преобразования. Так, например, регулятор удобнее рассчитывать в управляемой канонической форме, а наблюдатель состояния – в наблюдаемой. В нелинейных системах также есть возможность при определенных ограничениях производить преобразования по структурной схеме модели. Рассмотрим несколько динамически нелинейных объектов, в которых нелинейная часть выражена умножением и делением [10]

$$\dot{x} = ax^3 + u. \quad (1)$$

Структурная схема объекта (1) представлена на рис. 1. При доступности всего вектора состояния для заданных объектов необходимо найти управление, удовлетворяющее критерию асимптотической устойчивости.

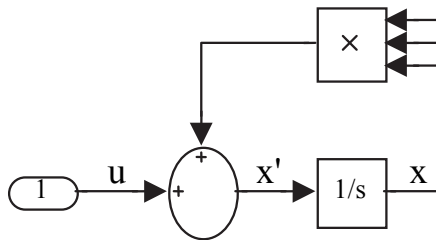


Рис. 1. Структурная схема объекта (1)

На примере объекта (1) ЛОС аналогична методу обратной задачи динамики. Выберем регулятор таким образом, чтобы замкнутая система имела вид

$$\dot{x} = v. \quad (2)$$

Для этого приравняем правые части уравнений (1) и (2):

$$v = ax^3 + u \quad (3)$$

и находим  $u$ :

$$u = v - ax^3. \quad (4)$$

Найденное управление (3) линеаризует объект (1) и приводит его к виду (2). Полученная система показана на рис. 2.



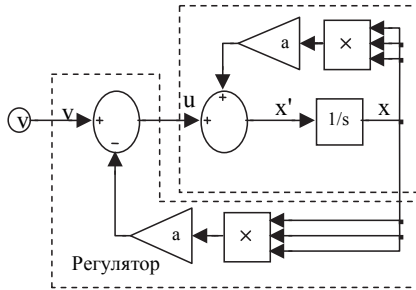


Рис. 2. Система объект–регулятор

Объект, замкнутый управлением (3), имеет линейное поведение, что позволяет для замкнутой системы вида (2) провести синтез управления, основанного на линейных методах.

Рассмотрим **систему второго порядка**:

$$\dot{x}_1 = cx_1^3 + x_2, \quad \dot{x}_2 = u. \tag{5}$$

Путем структурных преобразований представим уравнение (5) в виде линейной системы с нелинейной старшей производной (рис. 3).

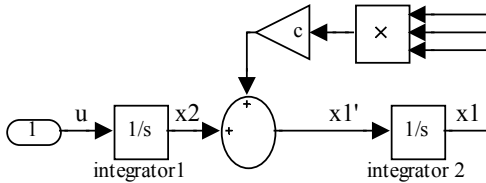


Рис. 3. Структурная схема объекта (3)

Перенесем нелинейные звенья объекта в управление для его последующей компенсации. В данном случае необходимо перенести  $c(x_1)^3$  «через» integrator1 (рис. 4).

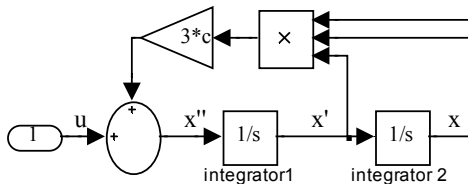


Рис. 4. Структурная схема после преобразования

Уравнение объекта после преобразования

$$\ddot{x} = 3cx^2\dot{x} + u, \quad (6)$$

в новых переменных

$$x = x_1 = z_1,$$

$$\dot{x} = \dot{x}_1 = z_2 = cx_1^3 + x_2,$$

$$\ddot{x} = \dot{z}_2 = 3cx^2\dot{x} + u = 3cx_1^2(cx_1^3 + x_2) + u,$$

далее для компенсации нелинейности выберем управление вида

$$u = -3cx_1^2(cx_1^3 + x_2) + v. \quad (7)$$

Система, замкнутая обратной связью, имеет вид

$$\ddot{x} = \ddot{x}_1 = v, \quad (8)$$

а в новых переменных

$$\dot{z}_1 = z_2, \quad \dot{z}_2 = v.$$

Подставим полученное управление (7) в исходное уравнение объекта (6)

$$\dot{x}_1 = cx_1^3 + x_2, \quad \dot{x}_2 = -3cx_1^2(cx_1^3 + x_2) + v. \quad (9)$$

Структурная схема системы «объект + регулятор» представлена на рис. 5.

Замкнутая система (9), имея несколько умножений, тем не менее эквивалентна системе «двойной интегратор» (8), где задание  $v$  после двойного интегрирования подается на выход объекта (рис. 6).

Рассмотрим следующую **систему третьего порядка**:

$$\dot{x}_1 = x_2,$$

$$\dot{x}_2 = x_3^3 + u, \quad (10)$$

$$\dot{x}_3 = x_1 + cx_3^3.$$

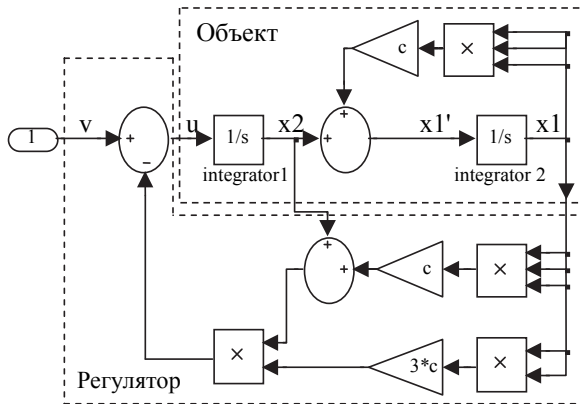


Рис. 5. Структурная схема системы «объект + регулятор»

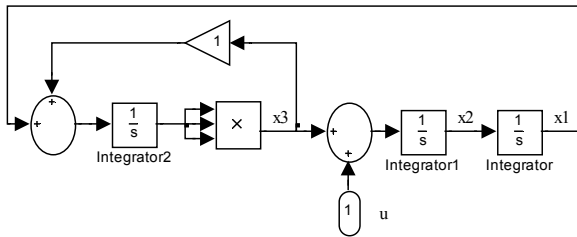


Рис. 6. Структурная схема объекта (8)

Найдем преобразование координат эвристически. Для этого нам необходимо через старшую производную выразить все нелинейности объекта, выразим  $x_1$  из уравнения (8):

$$x_1 = \frac{x_3^3 + u}{p^2},$$

подставим найденный  $x_1$  в уравнение производной  $x_3$ :

$$\dot{x}_3 = \frac{x_3^3 + u}{p^2} + cx_3 \Rightarrow \ddot{x}_3 = x_3^3 + 6cx_3\dot{x}_3 + 3cx_3^2\ddot{x}_3 + u.$$

Соответственно управление  $u$ , необходимое для компенсации нелинейностей, будет выглядеть так:

$$u = -\left(x_3^3 + 6cx_3(x_1 + cx_3)^2 + 3cx_3^2(x_2 + 3cx_3^2(x_1 + cx_3))\right) + v.$$

Преобразование координат для перехода к линейной модели:

$$z_1 = x_3, \quad z_2 = \dot{x}_3 = x_1 + cx_3^3, \quad z_3 = \ddot{x}_3 = x_2 + 3cx_3^2(x_1 + cx_3).$$

В новых переменных модель объекта примет вид

$$\dot{z}_1 = z_2, \quad \dot{z}_2 = z_3, \quad \dot{z}_3 = v.$$

В качестве следующего примера возьмем модель объекта, содержащую **тригонометрическую функцию**:

$$\ddot{y} + (\sin y)' = u. \tag{11}$$

Уравнение (11) было использовано в работе [12], но в данной работе будет рассмотрена та же линеаризация, но на основе структурных преобразований. Также уравнение (11) можно представить в следующем виде

$$\ddot{y} + \sin y = \int u dt.$$

Ниже представлены графики переходных процессов при различных входных воздействиях (рис. 7). Структурная схема объекта представлена на рис. 8.

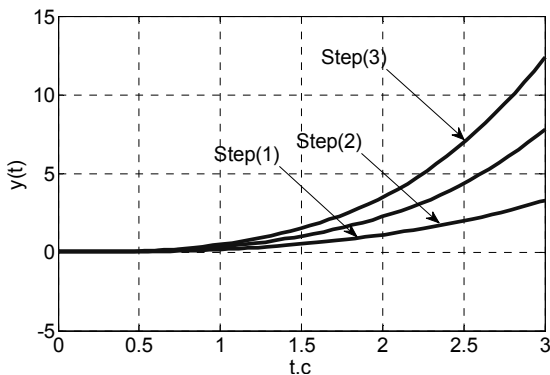


Рис. 7. Графики переходных процессов объекта (11)

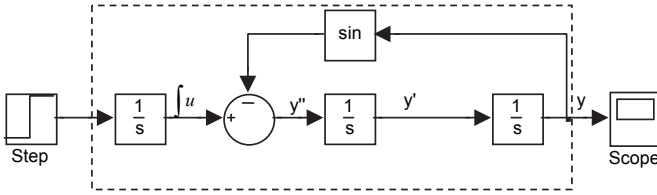


Рис. 8. Структурная схема объекта (11)

Компенсацию нелинейности будем производить через обратную связь. Для этого необходимо найти такое управляющее воздействие, которое в замкнутой системе приведет ее к необходимому нам линейному виду (в нашем случае тройной интегратор). Для этого ниже приведена структурная схема, эквивалентная структурной схеме объекта (рис. 9).

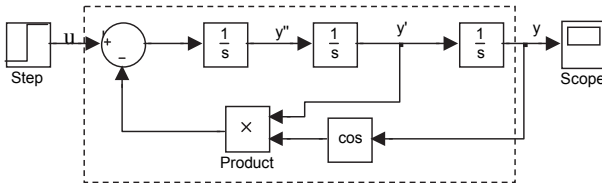


Рис. 9. Эквивалентная схема объекта (11)

Данную структурную схему описывает следующее дифференциальное уравнение:

$$\ddot{y} + \cos y \cdot \dot{y} = u .$$

По приведенной структурной схеме наглядно видно, что, прибавив косинус и производную выхода, мы будем иметь линейную систему вида 3 интегратора. На рис. 9, 10 и 11 приведены структурные схемы результирующей системы и эквивалентные ей:

$$\begin{aligned} \ddot{y} + (\sin y)' &= u, \\ u &= \cos y \cdot \dot{y}. \end{aligned} \tag{12}$$

На рис. 11 имеются тригонометрические функции, которые, как известно периодические. Поэтому интересны случаи попадания на их период. Это может возникнуть при выборе начальных условий или при выходе  $y$  за какие-то пределы.

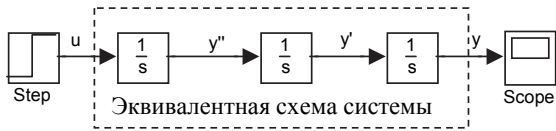


Рис. 10. Эквивалентная схема системы (12)

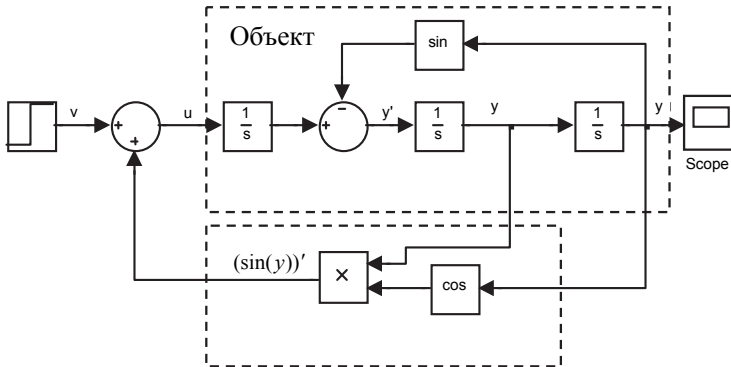


Рис. 11. Структурная схема результирующей системы (12)

Рассмотрим **различные представления** одного объекта. Уравнение объекта [11] описывается уравнением

$$\dot{y}(1+y) = u. \tag{13}$$

Структурная схема этого объекта представлена на рис. 12.

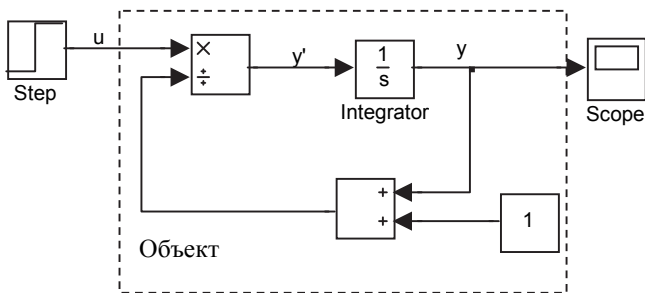


Рис. 12. Структурная схема объекта (13)

Переходные процессы на выходе объекта при подаче различных ступенчатых воздействий на вход показаны на рис. 13.

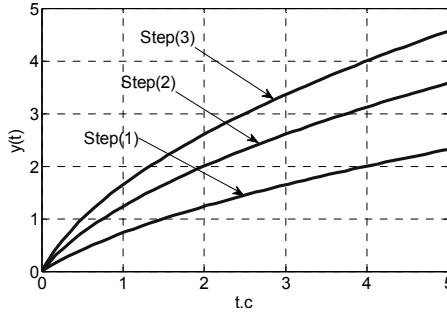


Рис. 13. Переходные процессы объекта (13)

Как видно из рис. 12, нелинейная часть типа деления  $\frac{1}{y+1}$  имеет явную зависимость от управления  $u$ . Это позволяет нам без применения каких-либо преобразований компенсировать ее за счет нелинейной обратной связи. Очевидно, что компенсация деления будет производиться умножением. Тогда, выбрав управление вида

$$u = (y+1)v, \quad (14)$$

получим замкнутую систему вида «интегратор»:

$$\dot{y} = v.$$

Структурная схема полученной системы представлена на рис. 14, а графики переходных процессов при различных входных ступенчатых воздействиях – на рис. 13.

Заметим, что в структурной схеме объекта имеется деление, и при значении выхода  $y = -1$  будет происходить деление на ноль, что говорит о неработоспособности модели в этой точке. Для решения этой проблемы существуют различные методы.

На следующем примере [13] покажем, что модель объекта можно представить в различных вариантах, в зависимости от которых будут различаться линеаризующие обратные связи, что придает конечной системе различные свойства:

$$\dot{y} + y\dot{y} = u, \quad (15)$$

где  $u$  является входным воздействием, а  $y$  – выходной величиной.

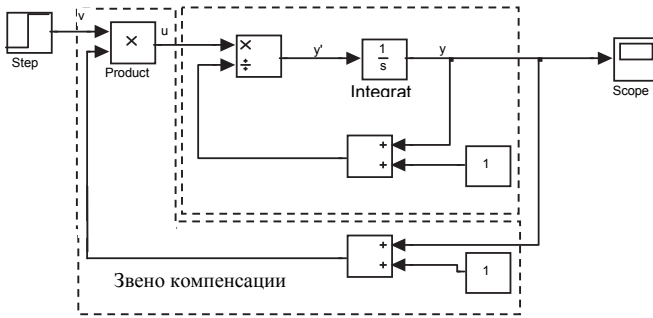


Рис. 14. Структурная схема объекта (13) с управлением (15)

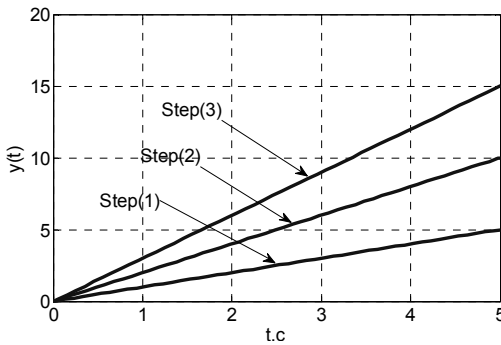


Рис. 15. Переходные процессы результирующей системы

Первый вариант представления модели объекта (12) – выразим производную

$$\dot{y} = \frac{u}{y+1} \tag{16}$$

Структурная схема показана на рис. 16. Эта схема включает блок деления. Если выбрать управление в виде

$$u = (y+1)v,$$

где  $v$  – задающее воздействие, результирующая система будет следующая:

$$\dot{y} = v \tag{17}$$



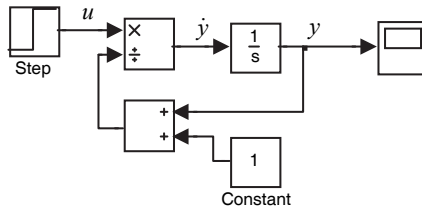


Рис. 16. Структурная схема модели (16)

Система «объект – устройство управления» представлена на рис. 17.

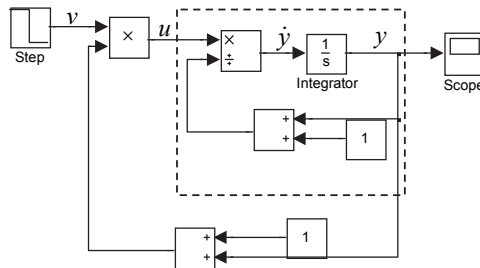


Рис. 17. Структурная схема замкнутой системы

Как видно из (16), при  $y = -1$  происходит деление на ноль, и этот случай требует отдельного исследования.

Второй вариант записи объекта (13) следующий:

$$\dot{y} = \frac{u - \dot{y}}{y}. \quad (18)$$

Структурная схема уравнения (18) изображена на рис. 18.

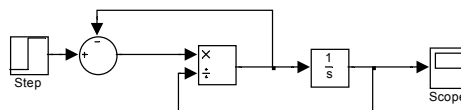


Рис. 18. Структурная схема уравнения (18)

Для компенсации нелинейной части модели (18) введем следующее управление:

$$u = \nu y + \dot{y}. \quad (19)$$

Замкнутая управлением (19) модель (18) будет эквивалентна простому интегратору аналогично (17). Структурная схема полученной системы приведена на рис. 19.

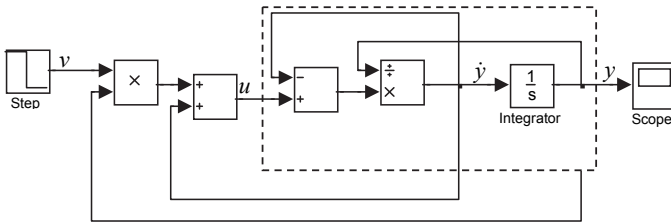


Рис. 19. Структурная схема замкнутой системы

Как видно из (18), при  $y = 0$  также происходит деление на ноль, что означает возможные проблемы при моделировании из нулевых начальных условий.

Третий вариант представления модели. Запишем уравнение объекта (13) в следующем виде:

$$\dot{y} = u - y\dot{y}. \quad (20)$$

Компенсирующее управление, приводящее модель (20) к эквивалентной замкнутой системе  $\dot{y} = v$ :

$$u = y\dot{y} + v. \quad (21)$$

Схема замкнутой системы представлена на рис. 20.

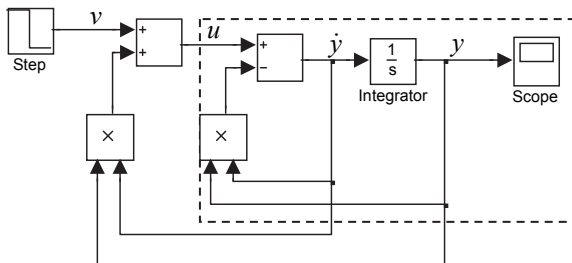


Рис. 20. Структурная схема модели (20), замкнутой управлением (21)

Сформированное управление, как и в предыдущих вариантах, компенсирует нелинейные элементы путем деления, умножения или вычитания.

*Четвертый вариант.* Если уравнение (13) представить как

$$\frac{d}{dt} \left( y + \frac{1}{2} y^2 \right) = u$$

и провести замену переменной

$$y + \frac{1}{2} y^2 = z, \quad (22)$$

то получим

$$\dot{z} = u. \quad (23)$$

Объект (13) в такой записи линейный с нелинейным выходом:

$$y_{1,2} = -1 \pm \sqrt{1 + 2z}. \quad (24)$$

Получено два выхода, это объясняется тем, что график зависимости  $z(y)$  схож с параболой, и в зависимости от начальных условий мы будем идти по какой-либо ветви. Из (24) следует, что на  $z$ , а следовательно, и на  $y$  наложены ограничения:

$$1 + 2z > 0 \Rightarrow z > -0.5 \Rightarrow y_1 > -1, y_2 < -1.$$

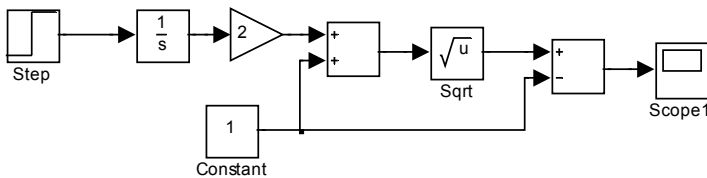


Рис. 21. Структурная схема модели (23)

Для поиска управления, линеаризующего модель (18), найдем производную выхода (22) для случая  $y_1$ :

$$\dot{y} = \frac{\dot{z}}{\sqrt{1 + 2z}} = \frac{u}{\sqrt{1 + 2z}}.$$

Управление, приводящее (20) к виду (17), будет следующее:

$$u = v\sqrt{1+2z} = v\sqrt{1+2\left(y + \frac{1}{2}y^2\right)}. \quad (25)$$

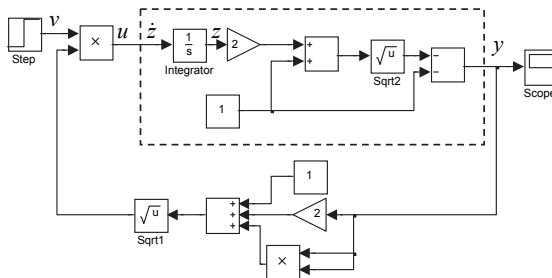


Рис. 22. Структурная схема замкнутой системы

На рис. 22 показана схема модели (20), замкнутой управлением (25). Разная запись исходного нелинейного уравнения (13) позволяет находить различное управление по обратной связи, компенсирующее нелинейности. Также в различных вариантах записи имеются свои ограничения на переменные.

## 2. СИНТЕЗ СИСТЕМЫ УПРАВЛЕНИЯ ПЕРЕВЕРНУТОГО МАЯТНИКА

Исходный объект описывается двумя нелинейными дифференциальными уравнениями [14]:

$$\left(1 - \frac{ml}{M_t L} \cos^2 \theta\right) \ddot{\theta} - \frac{g}{l} \sin \theta + \frac{ml}{M_t L} \cos \theta \sin \theta (\dot{\theta})^2 = \frac{1}{M_t L} \cos \theta u, \quad (26)$$

$$\left(1 - \frac{ml}{M_t L} \cos^2 \theta\right) \ddot{s} - \frac{ml}{M_t} \sin \theta (\dot{\theta})^2 + \frac{m l g}{M_t L} \cos \theta \sin \theta = \frac{1}{M_t} u,$$

где  $m$  – масса маятника;  $mg$  – вес маятника;  $M$  – масса тележки;  $M_t = m + M$  – отклонение маятника от вертикали;  $l$  – длина звена маятника,  $L = (I + ml^2) / ml$ ;  $I$  – момент инерции;  $s$  – перемещение тележки.

Под  $u$  понимается задание на положение маятника по углу  $\theta$ . Рассматривается частный случай, когда  $M \neq 0, I = 0$ , тогда (26) будет иметь вид

$$\ddot{\theta} = \frac{(m+M)g \sin \theta + ml \cos \theta \sin \theta (\dot{\theta})^2 + \cos \theta u}{(M+m \sin^2(\theta))l},$$

$$\ddot{s} = \frac{mg \cos \theta \sin \theta - ml \sin \theta (\dot{\theta})^2 + u}{M+m \sin^2(\theta)}.$$
(25)

Также имеет место другая общепринятая модель перевернутого маятника:

$$(I + ml^2)\ddot{\theta} - ml\dot{s} \cos \theta - mgl \sin \theta = 0,$$

$$(M+m)\ddot{s} - ml\ddot{\theta} \cos \theta - ml(\dot{\theta})^2 \sin \theta = u.$$

В отличие от (26), в таком представлении модели мы имеем одно управление, но в каждом уравнении мы имеем обе старшие производные.

Для управления перевернутым маятником зачастую используют его линейное представление в нулевой окрестности угла отклонения  $\theta = 0$ . В этом случае  $\dot{\theta}^2 = 0$ ,  $\sin \theta = \theta$ ,  $\cos \theta = 1$ . Мы будем рассматривать задачу стабилизации только угла отклонения маятника [15, 16], тогда исходное нелинейное уравнение будет иметь вид

$$(M+m \sin^2(\theta))l \cdot \ddot{\theta} - (m+M)g \sin \theta - ml \cos \theta \sin \theta (\dot{\theta})^2 = \cos \theta u, \quad (28)$$

а в окрестности нуля

$$Ml\ddot{\theta} - (m+M)g\theta = u. \quad (29)$$

Структурная схема модели подсистемы «маятник» представлена на рис. 23.

**Линеаризация.** Покажем на примере модели (28) отличия между разными линеаризующими добавками. Далее приведем исходное уравнение объекта (27) к линеаризованному виду (29) путем обратных связей. Для этого будем использовать линеаризующее звено вида

$$\frac{a_1\ddot{\theta} + a_2\dot{\theta} + a_3\theta + u_r}{\cos \theta} = u. \quad (30)$$

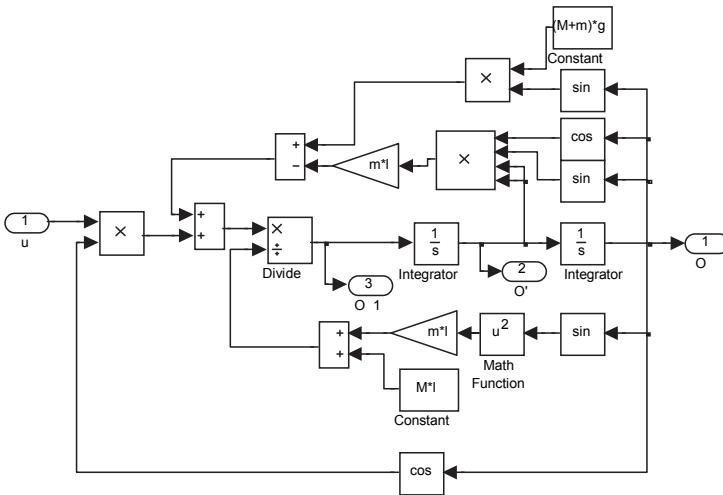


Рис. 23. Структурная схема подсистемы «маятник»

Приравнивая левые части уравнений (28) и (30), получим

$$\begin{aligned} & ((M + m \sin^2(\theta))l - a_1)\ddot{\theta} - (ml \cos \theta \sin \theta \dot{\theta} - a_2)\dot{\theta} - \\ & - \left( \frac{(m + M)g \sin \theta}{\theta} + a_3 \right) \theta = u_r. \end{aligned} \tag{31}$$

Поиск коэффициентов будем производить путем приравнивания коэффициентов уравнения (29) и линеаризованного (31), тогда

$$\begin{aligned} (M + m \sin^2(\theta))l - a_1 = Ml &\Rightarrow a_1 = m \sin^2(\theta)l, \\ ml \sin \theta \cos \theta \cdot \dot{\theta} - a_2 = 0 &\Rightarrow a_2 = ml \sin \theta \cos \theta \cdot \dot{\theta}, \end{aligned} \tag{32}$$

$$-\left( \frac{(m + M)g \sin \theta}{\theta} + a_3 \right) = -(M + m)g \Rightarrow a_3 = \frac{(m + M)g(\theta - \sin \theta)}{\theta}.$$

Таким образом, линеаризующее звено (30) с коэффициентами (32) преобразует исходное нелинейное уравнение (27) к линейному виду (29).

Рассмотрим другой способ линеаризации уравнения (28). Для этого запишем его в преобразованном виде

$$a_1(\theta) = (M + m)g \sin \theta - ml \cos \theta \sin \theta \cdot \dot{\theta}^2,$$

$$a_2(\theta) = (M + m \sin^2 \theta)l,$$

тогда

$$\ddot{\theta} = \frac{a_1(\theta)}{a_2(\theta)} + \frac{\cos \theta}{a_2(\theta)} u. \quad (33)$$

Далее будем слева направо составлять управление такое, чтобы скомпенсировать нелинейности и привести исходную модель к виду

$$\ddot{\theta} = u_r, \quad (34)$$

или в виде структурной схемы (рис. 24).

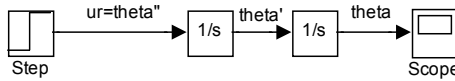


Рис. 24. Желаемая структурная схема после преобразования

Такое преобразование получается исходя из следующих соображений. Необходимо получить уравнение вида (34) из уравнения (33). Тогда, приравняв их правые части, получим

$$u_r = \frac{a_1(\theta)}{a_2(\theta)} + \frac{\cos \theta}{a_2(\theta)} u,$$

или

$$u = \frac{u_r a_2(\theta) - a_1(\theta)}{\cos \theta}. \quad (35)$$

Как видим, такое управление не имеет решения при угле  $\pi/2$ , так как происходит деление на ноль и управляющий сигнал принимает бесконечные значения. Также заметим, что движение маятника при углах, приближающихся к его горизонтальному положению, требует все больших усилий для придания

ему вертикального положения. В нашем случае мы не будем рассматривать стабилизацию угла из горизонтального положения или близкого к нему. Ниже представлена структурная схема полученной системы «объект–линеаризирующая добавка».

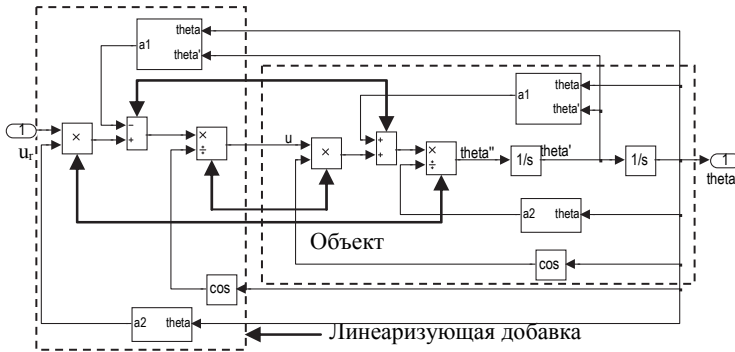


Рис. 25. Структурная схема объекта (33) с управлением (35)

На рис. 25 линиями с двойными стрелками показаны компоненты, компенсирующие друг друга, что в итоге приводит исходное уравнение объекта (33) к эквивалентному виду (34).

В отличие от представления (30), в (33) поиск управляющего воздействия ведется без использования второй производной.

**Синтез регулятора** для обоих случаев будем производить аналогично модальным методом синтеза с корнями характеристического полинома  $s_{1,2} = -4$ .

Наблюдатель полного порядка описывается уравнением

$$\dot{\tilde{x}} = A\tilde{x} + Bu + L(y - C\tilde{x}) .$$

Матрица коэффициентов  $L$  подбирается из условий необходимого быстрого действия наблюдателя. Ошибка оценивания  $\tilde{x} = x - \hat{x}$  вектора состояния  $x$  описывается уравнением  $\dot{\tilde{x}} = (A - LC)x$ . Как известно, для корректной работы наблюдателя переходные процессы в нем должны проходить в 5–10 раз быстрее процессов в системе «объект–регулятор». Входом наблюдателя является управление и выход объекта. Управление – это сила, приложенная к каретке, а выход – это угол отклонения маятника от вертикальной оси.



Управление сформируем в виде

$$u = -K\tilde{x} + u_r.$$

Вычисление матрицы  $L$  удобнее проводить, представляя объект в наблюдаемой канонической форме, а матрицу  $K$  – в управляемой.

**Расчет матрицы обратной связи** будем производить для линеаризованного обратной связью перевернутого маятника. Так как для обоих представлений (30) и (35) синтез регулятора не отличается, рассмотрим его на примере линеаризованной (замкнутой линеаризующей (32) добавкой) системы.

Подставим в уравнение (29) численные значения:

$$M = 30, \quad m = 70, \quad l = 1, \quad g = 10,$$

тогда

$$\ddot{\theta} = 33.3\theta + 0.033u$$

или в переменных состояния

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= 33.3x_1 + 0.033u, \\ y &= \theta = x_1. \end{aligned} \tag{36}$$

Уравнение (36) в матричном представлении  $\dot{x} = Ax + Bu$  имеет вид

$$A = \begin{pmatrix} 0 & 1 \\ 33.3 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0.033 \end{pmatrix}, \quad C = (1 \quad 0).$$

Для выбора коэффициентов регулятора смоделируем переходные процессы объекта (рис. 26).

Как видно из графиков, сильное отклонение угла от вертикали «падение маятника» происходит примерно после 1.2...1.6 с. Исходя из этого зададим собственные свойства системы «объект–регулятор» желаемыми полюсами:

$$s_{1,2} = -4; \quad (s + 4)^2 = s^2 + 8s + 16.$$

Выбор таких полюсов в теории должен дать время переходного процесса системы «объект–регулятор»  $t_{п.пер} = 1,5$  с.

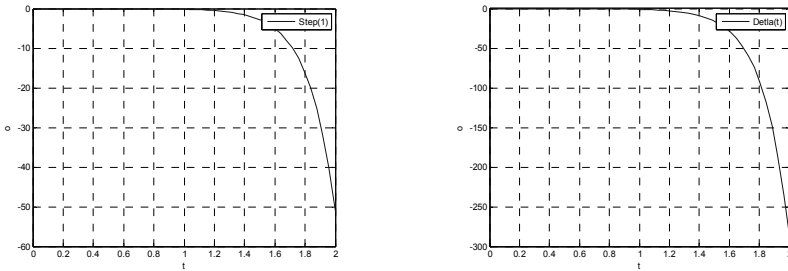


Рис 26. Переходные процессы объекта при различных входных воздействиях

Таким образом, желаемая матрица объекта с регулятором будет выглядеть так:

$$A_d = \begin{pmatrix} 0 & 1 \\ -16 & -8 \end{pmatrix}.$$

Добиваться желаемых полюсов будем введением в обратную связь матрицы  $K = (k_1 \quad k_2)$ :

$$A_d = A - BK = \begin{pmatrix} 0 & 1 \\ -16 & -8 \end{pmatrix}.$$

Использование модального метода синтеза подразумевает доступность всего вектора состояния объекта. Будем вычислять компоненты вектора  $x$  при помощи наблюдателя состояния полного порядка.

**Расчет наблюдателя полного порядка** будем производить для объекта, представленного в наблюдаемой форме. Для того чтобы исходное уравнение объекта представить в наблюдаемой форме, возьмем за выход объекта переменную  $x_2$ , тогда

$$\dot{x}_1 = 33.3x_2 + 0.033u,$$

$$\dot{x}_2 = x_1,$$

$$y = \theta = x_2.$$

Запишем в матричном виде:

$$A_o = \begin{pmatrix} 0 & 33.3 \\ 1 & 0 \end{pmatrix}, \quad B_o = \begin{pmatrix} 0.033 \\ 0 \end{pmatrix}, \quad C_o = (0 \quad 1).$$

Исходя из условий быстродействия наблюдателя выберем собственные числа:  $s_{1,2} = -20$ ;  $(s+2)^2 = s^2 + 40s + 400$ . Таким образом, желаемая матрица будет выглядеть так:

$$A_{od} = \begin{pmatrix} 0 & -400 \\ 1 & -40 \end{pmatrix}.$$

Наблюдатель с выбранными полюсами будет в пять раз быстрее системы «объект–регулятор». Добиваться этого будем введением в обратную связь  $L_o = (l_1 \ l_2)^T$ . Динамические свойства наблюдателя описываются уравнением  $\dot{\hat{x}} = (A - LC)x$  Таким образом:

$$A_{od} = A_o - L_o C_o = \begin{pmatrix} 0 & -400 \\ 1 & -40 \end{pmatrix},$$

$$L_o = (433.3 \ 40)^T.$$

Таким образом получены два варианта решения задачи линеаризации нелинейной модели перевернутого маятника с последующим синтезом наблюдателя состояния полного порядка и регулятора, рассчитанного модальным методом синтеза. На рис. 27 и 28 представлены структурные схемы.

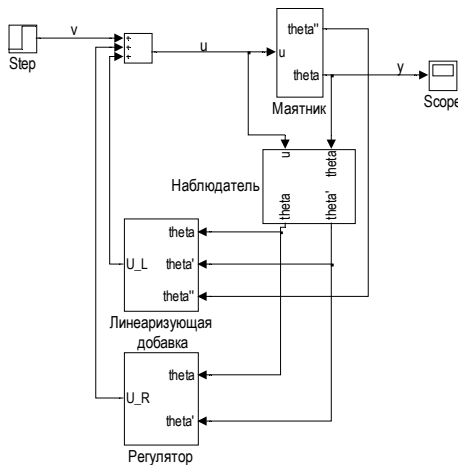


Рис. 27. Структурная схема на основе (30)

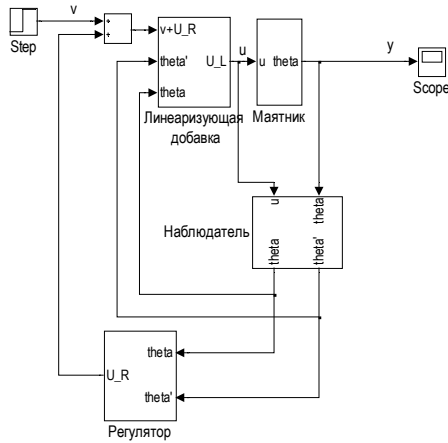


Рис. 28. Структурная схема на основе (35)

По структурным схемам видно, что рис. 28 отличается от рис. 27 тем, что линеаризующая добавка на рис. 28 входит не аддитивно, а на рис. 27 – аддитивно. Также в уравнении (35) не используется старшая производная в отличие от уравнения (30). Сравним качество переходных процессов полученных систем. Вначале сравним систему, построенную на основе уравнения (30), с системой с тем же управлением, но без линеаризующей добавки.

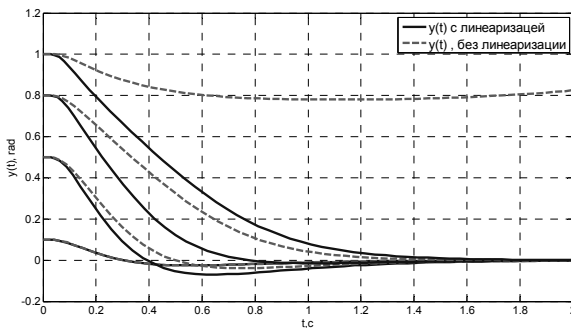


Рис. 29. Графики переходных процессов при различных начальных условиях

На рис. 29 представлены графики переходных процессов системы «объект–линеаризующая добавка–наблюдатель–регулятор» при различных начальных условиях объекта и нулевых начальных условиях наблюдателя состояния.

Перерегулирование у переходных процессов объясняется появившимися нулями в передаточной функции линеаризованной системы. Время переходного процесса по сравнению с системами без наблюдателя существенно не изменилось, это объясняется правильным выбором скорости убывания ошибки оценивания. Но в системах с наблюдателем возрастает сигнал управления, поэтому в системах с сильно ограниченным управляющим сигналом применение наблюдателя состояния ограничивает возможности регулирования.

Далее сравним переходные процессы системы (30) с системой (35)

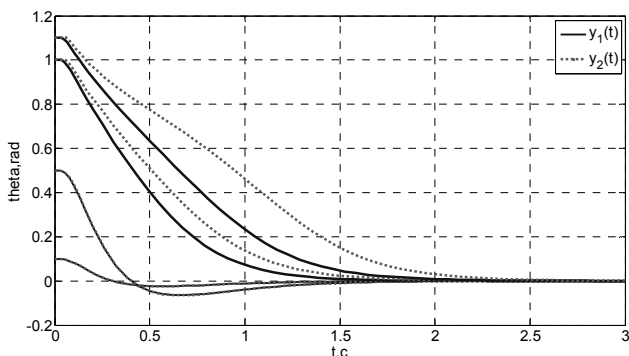


Рис. 30. Графики переходных процессов синтезированной системы (35)  $y_1(t)$  и (30)  $y_2(t)$

Как видно из рис. 30, при сравнительно небольших углах отклонения графики переходных процессов совпадают, но при увеличении начального угла система (35) работает лучше.

### 3. ЛИНЕАРИЗАЦИЯ МНОГОКАНАЛЬНЫХ ОБЪЕКТОВ ПО ВЫХОДУ

Предлагается решить задачу линеаризации нелинейной многоканальной системы [17,18]. Эта система описывает большой класс многоканальных объектов, в том числе и электрическую машину переменного тока [19, 20]. Выходные величины рассматриваемой системы нелинейно связаны с переменными состояниями. На рис. 31 показана структурная схема объекта электрической машины.

Пусть по аналогии с машиной переменного тока свойства объекта (тепловые потери, реактивная мощность, величина напряжения) зависят от соотно-

шений регулируемых переменных. Эти свойства могут выступать в качестве выходных величин в виде вектора  $z(t)$ , компоненты  $z_i(t)$  которого нелинейно связаны с регулируемыми переменными.

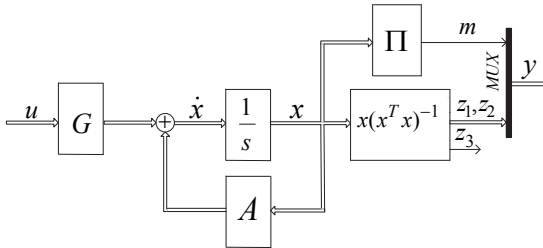


Рис. 31. Структурная схема электрической машины переменного тока

Представим описание объекта моделирования в переменных состояния, в котором  $u \in R^3$  – вектор входных воздействий объекта;  $x \in R^3$  – вектор состояния;  $y \in R^3$  – вектор выходных величин;  $A \in R^{3 \times 3}$  – матрица размером  $3 \times 3$ , состоящая из вещественных чисел; матрица  $G \in R^{3 \times 3}$  – единичная матрица;  $MUX$  – мультиплексор,  $T$  – знак транспонирования,  $\Pi$  – знак умножения компонент вектора. Математическое описание объекта приведено ниже:

$$\dot{x} = Ax + Gu,$$

$$z = (z_1 \quad z_2 \quad z_3)^T,$$

$$m = x_1 x_2 x_3,$$

где

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad G = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$u = (u_1 \quad u_2 \quad u_3)^T, \quad x = (x_1 \quad x_2 \quad x_3)^T,$$

$$Z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} \frac{x_1}{\sqrt{x_1^2 + x_2^2 + x_3^2}} & \frac{x_2}{\sqrt{x_1^2 + x_2^2 + x_3^2}} & \frac{x_3}{\sqrt{x_1^2 + x_2^2 + x_3^2}} \end{pmatrix}^T.$$

Компоненты вектора  $z$  связаны между собой следующим уравнением:

$$z_1^2 + z_2^2 + z_3^2 = 1. \quad (37)$$

По аналогии с машиной переменного тока свойства объекта (тепловые потери, реактивная мощность, величина напряжения) зависят от соотношений регулируемых переменных. Эти свойства могут выступать в качестве выходных величин в виде вектора  $z$ , компоненты  $z_i$  которого нелинейно связаны с регулируемыми переменными. Далее рассмотрим двухканальный вариант, структурная схема которого представлена ниже.

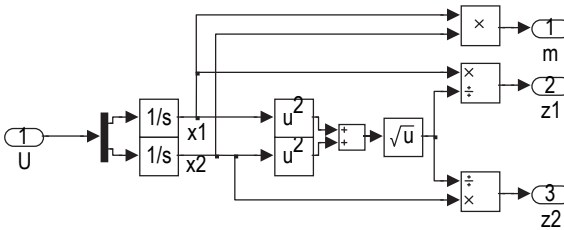


Рис. 32. Структурная схема двухканального объекта

В переменных состояния модель на рис. 32 имеет вид

$$\dot{x} = U,$$

$$m = x_1 x_2, \quad (38)$$

$$Z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \frac{x_1}{\sqrt{x_1^2 + x_2^2}} \\ \frac{x_2}{\sqrt{x_1^2 + x_2^2}} \end{pmatrix},$$

где

$$\begin{aligned} z_1^2 + z_2^2 &= 1, \\ U &= (u_1 \quad u_2)^T. \end{aligned} \quad (39)$$

Выходными переменными аналогично (38) и исходя из (39) возьмем

$$Y = \begin{pmatrix} m \\ z_1 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

Для нахождения зависимости между входными и выходными величинами возьмем производную выхода  $Y$ :

$$\dot{Y} = \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} \frac{x_2^2 \dot{x}_1 - x_1 x_2 \dot{x}_2}{\sqrt{(x_1^2 + x_2^2)^3}} \\ x_2 \dot{x}_1 + x_1 \dot{x}_2 \end{pmatrix}. \quad (40)$$

Так как линейная часть описывается параллельными интеграторами, то исходя из (38) уравнение (40) запишем в виде

$$\dot{Y} = \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} \frac{x_2^2 u_1 - x_1 x_2 u_2}{\sqrt{(x_1^2 + x_2^2)^3}} \\ x_2 u_1 + x_1 u_2 \end{pmatrix}; \quad (41)$$

$$\dot{Y} = F(x_1, x_2)U = \begin{pmatrix} \frac{x_2^2}{\sqrt{(x_1^2 + x_2^2)^3}} & \frac{x_1 x_2}{\sqrt{(x_1^2 + x_2^2)^3}} \\ x_2 & x_1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}. \quad (42)$$

Для нахождения такого управления, чтобы полученное линеаризованное уравнение системы имело вид  $\dot{Y} = V = (v_1 \quad v_2)^T$ , запишем (42) в следующем виде:

$$V = F(x_1, x_2)U,$$



тогда искомое управление имеет вид

$$U = F(x_1, x_2)^{-1}V = \begin{pmatrix} \frac{\sqrt{(x_1^2 + x_2^2)^3}}{2x_2^2} & \frac{1}{2x_2} \\ \frac{\sqrt{(x_1^2 + x_2^2)^3}}{2x_1x_2} & \frac{1}{2x_1} \end{pmatrix} V = \begin{pmatrix} F_1(x) & F_2(x) \\ F_3(x) & F_4(x) \end{pmatrix} V. \quad (43)$$

Полученное управление (43) будет компенсировать нелинейную зависимость выходных величин и переменных состояния, приводя ее к виду двух независимых друг от друга интеграторов следующего вида:

$$V \frac{1}{p} = Y, \quad (44)$$

тогда решением уравнения (44) будет

$$Y = Vt + C. \quad (45)$$

Из уравнения (45) необходимо найти постоянный вектор  $C$ , который определяет начальные условия выходных величин. Выходами объекта являются  $z_1$  и  $m$ , а начальные условия, задаваемые на интеграторах, определяют начальное положение вектора  $x$ . Для нахождения такого  $x(0)$ , что будут получены необходимые начальные условия  $Y(0)$ , выразим вектор переменных состояния  $x$  через вектор выхода  $Y$ :

$$Y = \begin{pmatrix} m \\ z_1 \end{pmatrix} = \begin{pmatrix} x_1x_2 \\ \frac{x_1}{\sqrt{x_1^2 + x_2^2}} \end{pmatrix}.$$

Отсюда находим, что

$$x(0) = \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} \frac{m(0)}{\sqrt{\frac{m^2(0)}{z_1^2(0)}(1-z_1^2(0))}} \\ \sqrt{\frac{m^2(0)}{z_1^2(0)}(1-z_1^2(0))} \end{pmatrix}. \quad (46)$$

Используя уравнение (46), можем задавать необходимые начальные условия для  $m$  и  $z_1$  и вычислять начальные условия для  $x$ . Как видно из (46),  $z_1^2(0) < 1$ , что соответствует условию (40).

Далее рассмотрим трехканальную модель, в которой динамическая часть описывается тремя параллельными интеграторами без матрицы  $A$ . Структурная схема представлена на рис. 33.

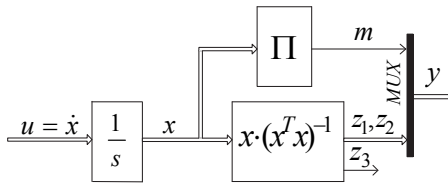


Рис. 33. Структурная схема объекта моделирования

Данный объект имеет векторный вход  $u = (u_1 \ u_2 \ u_3)^T$  и четыре выходные переменные, это вектор  $z = (z_1 \ z_2 \ z_3)^T$  и скалярная величина  $m$ . Элементы вектора  $z$  связаны соотношением (37), что позволяет при заданных  $z_1$  и  $z_2$  вычислить  $z_3$  из соотношения

$$z_3 = \sqrt{1 - z_1^2 - z_2^2}. \tag{47}$$

Таким образом, составим вектор выходных переменных

$$y = (m \ z_1 \ z_2)^T. \tag{48}$$

С таким выходом модель объекта представляется как система «3 входа – 3 выхода». Очевидно, что (47) вводит ограничение по модулю на изменение элементов вектора  $z$ . Поиск управляющих воздействий выполним, считая, что вектор состояния  $x$  доступен; при недоступности вектора состояния применяются наблюдатели состояния.

Выходные величины (48) нелинейно связаны с переменными состояния. Для нахождения их линейной зависимости от задания  $v = (v_1 \ v_2 \ v_3)^T$  найдем такое управление  $u$  по обратной связи, чтобы замкнутая система опи-

сывалась системой линейных дифференциальных уравнений. Для поиска такого управления необходимо найти его прямую зависимость от вектора выходных величин. Возьмем производную выхода, тогда

$$\dot{y} = \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{m} \end{pmatrix} = \begin{pmatrix} \frac{x_2^2 + x_3^2}{\|x\|^3} \dot{x}_1 - \frac{x_1 x_2}{\|x\|^3} \dot{x}_2 - \frac{x_1 x_3}{\|x\|^3} \dot{x}_3 \\ -\frac{x_1 x_2}{\|x\|^3} \dot{x}_1 + \frac{x_1^2 + x_3^2}{\|x\|^3} \dot{x}_2 - \frac{x_2 x_3}{\|x\|^3} \dot{x}_3 \\ x_2 x_3 \dot{x}_1 + x_1 x_3 \dot{x}_2 + x_1 x_2 \dot{x}_3 \end{pmatrix}, \quad (49)$$

где  $\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$  является евклидовой нормой.

Так как линейная часть описывается параллельными интеграторами, то уравнение (49) принимает вид

$$\dot{y} = \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{m} \end{pmatrix} = \begin{pmatrix} \frac{x_2^2 + x_3^2}{\|x\|^3} & -\frac{x_1 x_2}{\|x\|^3} & -\frac{x_1 x_3}{\|x\|^3} \\ -\frac{x_1 x_2}{\|x\|^3} & \frac{x_1^2 + x_3^2}{\|x\|^3} & -\frac{x_2 x_3}{\|x\|^3} \\ x_2 x_3 & x_1 x_3 & x_1 x_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}. \quad (50)$$

В уравнении (48) обозначим множитель перед  $u$  через  $F(x_1, x_2, x_3)$ , тогда

$$\dot{y} = F(x_1, x_2, x_3)u. \quad (51)$$

Так как целью работы является линейная зависимость вектора выходных величин от задания, то, приравняв производную выхода к заданию, получим  $\dot{y} = v = (v_1 \ v_2 \ v_3)^T$ , тогда уравнение (51) примет вид

$$v = F(x_1, x_2, x_3)u,$$

а искомое управление будет соответствовать

$$u = F(x_1, x_2, x_3)^{-1} v, \tag{52}$$

где

$$F(x_1, x_2, x_3)^{-1} = \begin{pmatrix} \frac{x_1^2 + 2x_2^2}{3x_3^2} \|x\| & \frac{x_1(x_2^2 - x_3^2)}{3x_2x_3^2} \|x\| & \frac{1}{3x_2x_3} \\ \frac{x_1(x_1^2 - x_3^2)}{3x_1x_3^2} \|x\| & \frac{x_2^2 + 2x_3^2}{3x_3^2} \|x\| & \frac{1}{3x_1x_3} \\ \frac{2x_1^2 + x_3^2}{3x_1x_3^2} \|x\| & \frac{2x_2^2 + x_3^2}{3x_2x_3^2} \|x\| & \frac{1}{3x_1x_2} \end{pmatrix}.$$

На рис. 34 представлена структурная схема полученной системы, которая состоит из модели объекта, представленной на рис. 33, замкнутой компенсирующим сигналом (52).

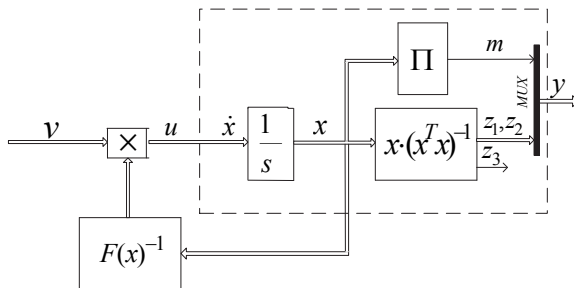


Рис. 34. Структурная схема системы «объект – компенсирующий сигнал»

Задание начальных условий возможно только на интеграторах, т. е. для переменных состояния  $x(0)$ . Расчет начальных условий вектора выходных величин  $y$  необходимо производить заданием определенных значений вектору  $x$ .

На рис. 35 представлены графики переходных процессов при  $m(0) = z_1(0) = z_2(0) = (0.01, 0.2)$  и  $v = (0.5 \ 0.5 \ 0.5)^T$ . Как видим, графики

имеют вид прямых, что соответствует интегрированию постоянного входного воздействия  $v$ .

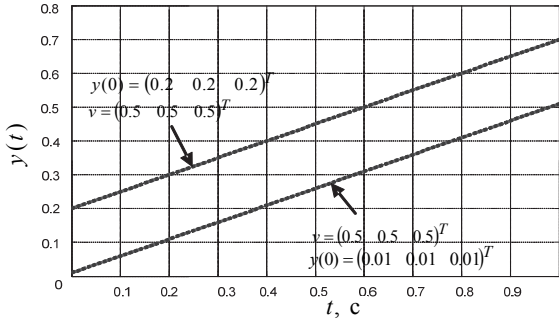


Рис. 35. Переходные процессы выходного вектора  $y$  при одинаковых начальных условиях на всех каналах

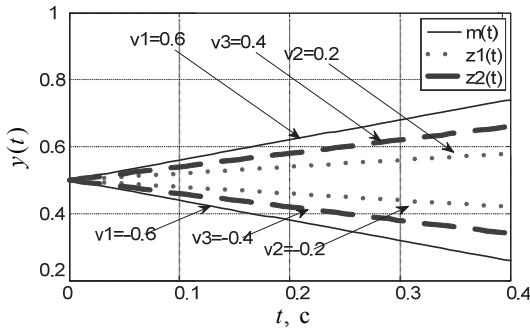


Рис. 36. Переходные процессы выхода при  $y(0) = (0.5 \ 0.5 \ 0.5)^T$

Исходя из графиков, представленных на рис. 36, можно сделать вывод, что при различных задающих воздействиях на каждый канал выходные величины изменяются независимо друг от друга и соответствуют интегрированию своего задающего воздействия.

Как видно из рис. 37, все три выходные величины изменяются независимо друг от друга. При  $v_2 = v_3 = 0$  и  $v_1 = 2$  изменения во времени  $z_1$  и  $z_2$  не происходит, значения остаются равными своим начальным условиям, а компонента  $m$  изменяется в соответствии со своим заданием.

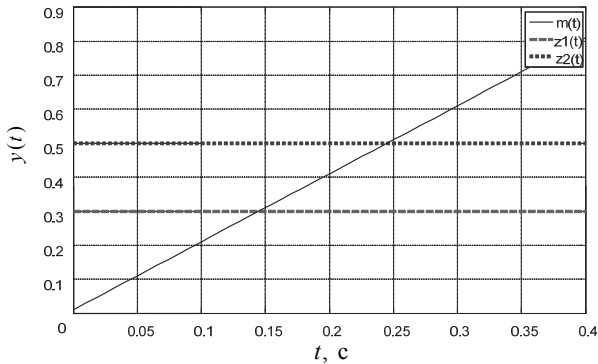


Рис. 37. Переходные процессы выхода при  $y(0) = (0.01 \ 0.3 \ 0.5)^T$  и входном воздействии  $v(0) = (2 \ 0 \ 0)^T$

Решена задача линеаризации для трехканального объекта с нелинейным выходным вектором. Модель такого объекта описывает электрическую машину переменного тока. Решение заключается в нахождении нелинейной обратной связи такой, что исходный нелинейный объект, представленный на рис. 32, замкнутый компенсирующим сигналом (52), принял вид трехканального интегрирующего звена (рис. 34). На рис. 35 и 36 показано, что выходные переменные в синтезированной системе не связаны друг с другом и могут изменяться отдельно в соответствии со своим задающим воздействием, несмотря на то что в самом объекте выходные величины нелинейно связаны как между собой, так и с переменными состояния. Данный факт позволяет нам формировать управление линейными методами синтеза на каждый канал отдельно в соответствии с желаемой динамикой. Работоспособность синтезированной системы ограничена некоторой областью изменения выходных величин. Для синтеза такого компенсирующего управления необходима известность всего вектора состояния.

#### 4. ПРЕОБРАЗОВАНИЕ КООРДИНАТ НА ОСНОВЕ АЛГЕБРЫ ЛИ

Рассмотрим способ нахождения такого управления, которое позволяет линеаризовать исходную нелинейную модель, применяя дифференциально-геометрический подход. Данный подход [21] заключается в использовании

производных Ли (скобки Ли, алгебра Ли, группы Ли). Он применяется для нелинейных объектов, линейных по управлению (*аффинных*):

$$\dot{x} = A(x) + B(x)u,$$

где  $u$  – гладкая функция в некоторой окрестности  $\Omega(0)$  начала координат. Начало координат является положением равновесия  $f(0, 0) = 0$ . Здесь  $x$  – вектор состояния,  $u$  – управление.

Использование этого инструмента позволяет проводить анализ и синтез нелинейных систем. В частности, матрицы наблюдаемости и управляемости похожи на те, что получены от линеаризованного объекта. Также используется понятие о каноническом виде, который, в общем, аналогичен каноническому виду в линейных системах. Производные Ли особенно удобно использовать для непрерывных дифференцируемых функций, что соответствует нашему классу объектов. Для рассмотрения линеаризации обратной связью нам понадобятся некоторые сведения из дифференциальной геометрии, такие как производная Ли, скобки Ли, матрица управляемости, полученная для нелинейной системы.

**Производная по направлению векторного поля** является обобщенным понятием производной по направлению. Такая производная от скалярной функции задается формулой

$$\sum_{i=1}^n a_i(x) \frac{\partial f}{\partial x_i}.$$

Такую производную функции по векторному полю называют **производной Ли** ( $L_A f(x)$  – «производная функции  $f(x)$  по направлению векторного поля  $\mathbf{A}$ »).

Если брать производную не от скалярной функции, а от другого векторного поля, то уравнение производной запишется в виде

$$[A, B] = \frac{\partial B(x)}{\partial x} A(x) - \frac{\partial A(x)}{\partial x} B(x). \quad (53)$$

Такую производную векторного поля по векторному полю называют **скобкой Ли (коммутатором)**  $[A, B] \equiv ad_A B$  – «производная векторного поля  $\mathbf{B}$  по направлению векторного поля  $\mathbf{A}$ ». Описание свойств этих производных осуществляется при помощи *алгебры Ли*. Для работы с такими производными существует свой математический аппарат.

Использование производных Ли позволяет приводить аналогии в описании линейных и нелинейных систем. Так, используя производную Ли, удастся

проверить наблюдаемость и управляемость системы, которые проверяются аналогично с линейными системами по матрице наблюдаемости и управляемости, у них также проверяется ранг.

**Управляемость** нелинейного объекта, как и линейного объекта, определяется при помощи матрицы управляемости, которая в нелинейном случае состоит из коммутаторов векторных полей **A** и **B**:

$$U = \left( B(x), [A, B], \dots, \left[ A, ad_A^{k-1} B \right] \right). \quad (54)$$

**Канонические формы** в нелинейных системах, как и в линейных, широко используются для анализа и синтеза систем управления. Для нелинейных систем, линейных по управлению (аффинных), представленных в канонической форме, разработаны методы решения задачи стабилизации. Системы, не представленные в канонической форме, возможно преобразовать в каноническую форму. Для этого с использованием алгебры Ли, существует достаточно формализованный подход нахождения преобразования координат.

Система, записанная в виде

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= x_3, \\ &\dots \\ \dot{x}_{n-1} &= x_n, \\ \dot{x}_n &= f(x) + b(x)u, \\ y &= x_1, \end{aligned} \quad (55)$$

называется системой, представленной в *канонической форме*.

**Синтез алгоритма управления** состоит из пяти этапов:

- 1) найти матрицу управляемости (54);
- 2) найти определитель матрицы управляемости и проверить инвалютивность всех столбцов матрицы управляемости, кроме последнего;
- 3) если определитель матрицы управляемости не равен нулю и проверка на инвалютивность прошла успешно, то определить функцию  $T_1$  из преобразования  $z = T(x)$  из условий:

$$\frac{\partial T_1}{\partial x} ad_A^i B = 0, \quad \frac{\partial T_1}{\partial x} ad_A^{n-1} B \neq 0, \quad i = 0, 1, \dots, n-2;$$



4) найти преобразование  $z = T(x) = [T_1(x), \dots, T_n(x)]^T$ , где  $T_n(x) = L_A^{n-1}T_1$ ;

5) управление выберем в виде

$$u = \frac{-L_A^n T_1(x) - \sum_{k=0}^{n-1} c_k L_A^k T_1(x)}{L_B L_A^{n-1} T_1(x)}, \text{ которое в переменных } z \text{ запишется в виде}$$

$$u = \frac{-f(z) - \sum_{k=0}^{n-1} c_k z_k}{g(z)}.$$

**Пример** нахождения стабилизирующего управления для аффинной управляемой динамической системы при условии доступности всего вектора состояния будет рассмотрен ниже. Пусть АУДС задано уравнениями

$$\dot{x}_1 = x_2 + cx_1^3, \quad \dot{x}_2 = u, \quad (56)$$

$$A(x) = \begin{pmatrix} x_2 + cx_1^3 & 0 \end{pmatrix}^T, \quad B(x) = \begin{pmatrix} 0 & 1 \end{pmatrix}^T.$$

Найдем матрицу управляемости (54), для чего вычислим ее компоненты (53). Вектор  $B(x)$  постоянный, и его производная равна нулю

$$\frac{\partial B(x)}{\partial x} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

а производная от  $A(x)$  равна

$$\frac{\partial A(x)}{\partial x} = \begin{pmatrix} 3cx_1 & 1 \\ 0 & 0 \end{pmatrix}.$$

Подставив найденные значения в (54), получим

$$[A, B] = ad_A B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_2 + cx_1^3 \\ 0 \end{pmatrix} - \begin{pmatrix} 3cx_1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}. \quad (57)$$

$$U = [B \quad ad_A B] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Определитель матрицы (57) отличен от нуля, и ее столбцы образуют инволютивное множество как постоянные векторы.

Найдем преобразование  $z = T(x)$  :

$$\frac{dT_1}{dx} B = \begin{pmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 0,$$

$$\frac{dT_1}{dx} ad_A B = \begin{pmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} \end{pmatrix} \begin{pmatrix} -1 \\ 0 \end{pmatrix} \neq 0.$$

Соответственно  $T_1$  зависит только от  $x_1$ . Пусть  $T_1 = x_1 = z_1$ .

Найдем  $T_2$ :

$$T_2 = L_A T_1 = z_2 = x_2 + cx_1^3.$$

Найдем  $L_A T_2$ :

$$L_A T_2 = \dot{z}_2 = \begin{pmatrix} 3cx_1^2 & 1 \end{pmatrix} \begin{pmatrix} x_2 + cx_1^3 \\ 0 \end{pmatrix} = 3cx_1^2 (x_2 + cx_1^3).$$

Управление, приводящее систему (4) к виду  $\dot{x}_1 = x_2$ ,  $\dot{x}_2 = v$ , имеет вид

$$u = -3cx_1^2 (x_2 + cx_1^3) + v,$$

а управление, стабилизирующее систему (56), имеет вид

$$u = -3cx_1^2 (x_2 + cx_1^3) - c_2 (x_2 + cx_1^3) - c_1 x_1. \quad (58)$$

В переменных  $z$  система уравнений (56) с управлением (58) примет вид

$$\dot{z}_1 = z_2, \quad \dot{z}_2 = -c_1 z_1 - c_2 z_2. \quad (59)$$

Для сравнения запишем уравнение (56), линеаризованное в окрестности  $x_1 = 0$  :

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = u. \quad (60)$$

Выберем управление в виде  $u = -c_1x_1 - c_2x_2$ , тогда система уравнений (60) с выбранным управлением запишется в виде

$$\dot{x}_1 = x_2 \quad \dot{x}_2 = -c_1x_1 - c_2x_2. \quad (61)$$

Система уравнений (59) и система уравнений (61) имеют одинаковый вид, но система (59) записана в переменных  $z$ , которые получаются путем нелинейного преобразования  $z = T(x)$  переменных  $x$ . На рис. 37 представлены графики переходных процессов систем (59) и (61) при различных начальных условиях  $x_1(0) = 0,4$  и  $x_1(0) = 0,83$ .

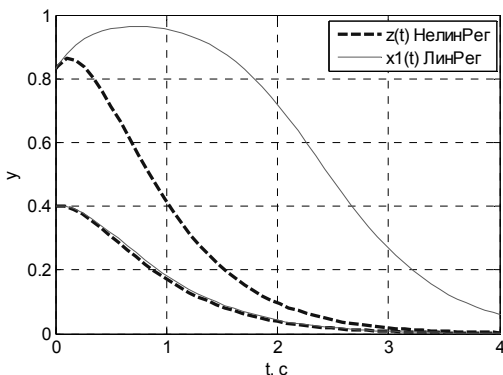


Рис 38. Графики переходных процессов систем (59) и (61)

Моделирование проводилось при  $y(0) = 0,4$  и  $y(0) = 0,83$ ,  $c_1 = c_2 = 1$ . Как видно из рис. 38, при возрастании начальных условий (отдаление от малой окрестности точки линеаризации) качество переходного процесса системы (61) ухудшается, а у системы (59) остается прежним.

## ЗАКЛЮЧЕНИЕ

Применение линеаризации по обратной связи придает исходной нелинейной модели линейное поведение. Такой подход дает возможность применять линейные методы синтеза. В работе была показана эквивалентная линеаризация на различных примерах. Рассмотрены модели с нелинейно связанными переменными состояния, модели с нелинейным выходом. Рассмотрен синтез системы управления для подсистемы «маятник» в объекте

«перевернутый маятник на тележке». Также для многоканальных объектов, нелинейных по выходу, показаны эквивалентная линеаризация и применение алгебры и скобок Ли для поиска нелинейных обратных связей. В зависимости от исходного представления модели можно найти разное линеаризующее управление с отличающимися свойствами. Этими свойствами могут быть различные области определения функции, а также использование различных компонент вектора состояния. На примере перевернутого маятника проведен синтез системы управления с модальным регулятором и наблюдателем полного порядка. В одном случае была использована старшая производная, а в другом – нет. Для многоканального объекта применение следующего алгоритма дает определенные преимущества. В модели объекта, нелинейного по выходу, выходные величины были связаны друг с другом, и их удалось их развязать. Замкнутая система получилась эквивалентна трем одноканальным системам вида интегратор с ограничениями на переменные состояния, вызванные структурой самого объекта. Таким образом, в работе выявлена возможность применения линеаризации обратными связями вместо обычной линеаризации, что усложняет алгоритм, но улучшает качество синтезированной системы.

## СПИСОК ЛИТЕРАТУРЫ

1. Воевода А.А., Вороной В.В. Полиномиальный метод расчета многоканальных регуляторов заданной структуры // Научный вестник НГТУ. – 2013. – № 2 (51). – С. 214–218.
2. Воевода А.А. Стабилизация двухмассовой системы: модальный метод синтеза с использованием полиномиального разложения // Научный вестник НГТУ. – 2010. – № 1 (38). – С. 195–198.
3. Воевода А.А., Чехонадских А.В., Шоба Е.В. Модальный метод синтеза с использованием полиномиального разложения: разделение движений при стабилизации трехмассовой системы // Научный вестник Новосибирского государственного технического университета. – 2011. – № 2 (43). – С. 39–46.
4. Воевода А.А., Шоба Е.В. Управление перевернутым маятником // Сборник научных трудов НГТУ. – 2012. – № 2 (68). – С. 3–14.
5. Воевода А.А., Жмудь В.А. Сходимость алгоритмов оптимизации регулятора для объекта с ограничением и с запаздыванием // Научный вестник НГТУ. – 2007. – № 4. – С. 179–184.
6. The modeling tests of the new PID-regulators structures / A.A. Voevoda, V.A. Zhmud, R.Y. Ishimtsev, V.M. Semibalamut // Proceedings of the 18th IASTED International Conference on Applied Simulation and Modelling,

ASM 2009, 7–9 September, 2009, Palma de Mallorca, Spain. – [S. 1.], 2009. – P. 165–168.

7. *Ким Д.П.* Теория автоматического управления. Т. 2. Многомерные, нелинейные, оптимальные и адаптивные системы. – М.: Физматлит, 2004. – 464 с.

8. *Slotine J.J.E., Li W.* Applied nonlinear control. – Englewood Cliffs, NJ: Prentice Hall, 1991.

9. *Marino R., Yomei P.* Nonlinear control design: geometric, adaptive, and robust. – London; New York: Prentice Hall, 1995. – 396 p.

10. *Филюшов В.Ю.* Линеаризация обратной связью: эвристический подход // Сборник научных трудов НГТУ. – 2016. – № 1 (83). – С. 37–46.

11. *Филюшов В.Ю.* Примеры использования нелинейных обратных связей для нелинейных объектов // Сборник научных трудов НГТУ. – 2016. – № 3 (85). – С. 61–70.

12. *Воевода А.А., Иванов А.Е.* Пример модального синтеза для нелинейного объекта с использованием нелинейных обратных связей // Сборник научных трудов НГТУ. – 2013. – № 2 (72). – С. 3–9.

13. *Филюшов В.Ю.* Примеры структурных преобразований нелинейного объекта // Сборник научных трудов НГТУ. – 2016. – № 4 (86). – С. 72–82.

14. *Воевода А.А., Шоба Е.В.* О модели перевернутого маятника // Сборник научных трудов НГТУ. – 2012. – № 1 (67). – С. 3–14.

15. *Воевода А.А., Филюшов В.Ю.* Линеаризация обратной связью: перевернутый маятник // Сборник научных трудов НГТУ. – 2016. – № 3 (85). – С. 49–60.

16. *Воевода А.А., Филюшов В.Ю.* Линеаризация обратной связью: перевернутый маятник // Сборник научных трудов НГТУ. – 2016. – № 4 (86). – С. 62–71.

17. *Воевода А.А., Филюшов В.Ю.* Линеаризация двухканальной системы с нелинейным выходом при помощи обратной связи // Сборник научных трудов НГТУ. – 2016. – № 4 (86). – С. 49–61.

18. *Филюшов В.Ю.* Линеаризация нелинейного трехканального динамического объекта обратной связью // Научный вестник НГТУ. – 2017. – № 1 (66). – С. 74–85.

19. *Онищенко Г.Б.* Электрический привод: учебник для вузов. – М.: Академия, 2006. – 288 с.

20. *Иванов-Смоленский А.В.* Электрические машины. – М.: Энергия, 1980. – 928 с.

21. *Воевода А.А., Филюшов В.Ю.* Линеаризация обратной связью // Сборник научных трудов НГТУ. – 2016. – № 2 (84). – С. 68–76.

**Филишов Владислав Юрьевич**, аспирант кафедры автоматики Новосибирского государственного технического университета. Основное направление научных исследований – синтез многоканальных нелинейных объектов. Имеет более десяти публикаций. E-mail: [filiushov.vladislav@gmail.com](mailto:filiushov.vladislav@gmail.com)

DOI: 10.17212/2307-6879-2018-2-36-80

## **Examples of structural transformation of nonlinear object\***

**V.Yu. Filiushov**

*Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, undergraduate of department automatics. E-mail: [filiushov.vladislav@gmail.com](mailto:filiushov.vladislav@gmail.com)*

At present, nonlinear synthesis methods that use the original nonlinear model are gaining popularity. Such an approach makes it possible to form a control not only in the vicinity of the linearization point, but in a larger domain of change in the state variables, which makes it possible to take into account the behavior of the object most fully in the control law. Linear synthesis methods, such as the modal synthesis method, synthesis methods using polynomial decomposition, optimization algorithms, construction of PID regulators, and others are used for models linearized in the vicinity of a certain point. But this representation does not describe all the properties of the operation of the source object. In order to apply linear synthesis methods for nonlinear objects, it is more appropriate to use feedback linearization, which leads the behavior of the original object to a linear form. The essence of the method lies in the search for such compensating control, in which the behavior of a closed system will correspond to the behavior of the system described by linear differential equations. Then, for the synthesized system, it becomes possible to calculate the regulator by linear synthesis methods. In the present work a review of previous works on this topic was carried out. Various single-channel nonlinear objects are considered; an object containing a trigonometric function; the possibility of obtaining a different solution of the problem of linearization by nonlinear feedbacks for the same object is shown. Using the example of the pendulum subsystem of the object model, an inverted pendulum on the trolley produces a synthesis of the control system with a modal regulator and a full-order observer, where in one case the higher derivative of the output quantity was used, and in the other there is not; the possibility of using this method for multichannel objects (two and three channel), nonlinear on the output, is shown; and also shows a formalized method for finding linearizing controls based on algebra and Lie brackets.

**Keywords:** nonlinear control, feedback linearization, regulator, state observer, synthesis, multichannel, Lie algebra

---

\* Received 19 February 2018.

## REFERENCES

1. Voevoda A.A., Voronoi V.V. Polinomial'nyi metod rascheta mnogokanal'nykh regulyatorov zadannoi struktury [Polynomial method for calculating multi-channel controllers of a given structure]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2013, no. 2 (51), pp. 214–218.
2. Voevoda A.A. Stabilizatsiya dvukhmassovoi sistemy: modal'nyi metod sinteza s ispol'zovaniem polinomial'nogo razlozheniya [Stabilisation of two-mass system by a modal method of synthesis with polynomial factorization]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2010, no. 1 (38), pp. 195–198.
3. Voevoda A.A., Chekhonadskikh A.V., Shoba E.V. Modal'nyi metod sinteza s ispol'zovaniem polinomial'nogo razlozheniya: razdelenie dvizhenii pri stabilizatsii trekhmassovoi sistemy [Modal synthesis method using a polynomial decomposition: the separation of motions in the stabilization of the three-mass plant]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2011, no. 2 (43), pp. 39–46.
4. Voevoda A.A., Shoba E.V. Upravlenie perevernutym mayatnikom [Management of the inverted pendulum]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2012, no. 2 (68), pp. 3–14.
5. Voevoda A.A., Zhmud V.A. Skhodimost' algoritmov optimizatsii regulyatora dlya ob"ekta sogra-nicheniem i s zapazdyvaniem [The convergence of the algorithms for the optimization of regulator for an object with restriction and delay]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2007, no. 4, pp. 179–184.
6. Voevoda A.A., Zhmud V.A., Ishimtsev R.Y., Semibalamut V.M. The modeling tests of the new PID-regulators structures. *Proceedings of the IASTED International Conference on Applied Simulation and Modelling, ASM 2009*, Palma de Mallorca, Spain, 7–9 September, 2009, pp. 165–168.
7. Kim D.P. *Teoriya avtomaticheskogo upravleniya. T. 2. Mnogomernye, nelineinye, optimal'nye i adaptivnye sistemy* [The theory of automatic control. Vol. 2. Multidimensional, nonlinear, optimal and adaptive systems]. Moscow, Fizmatlit Publ., 2004. 464 p.
8. Slotine J.J.E., Li W. *Applied nonlinear control*. Englewood Cliffs, NJ, Prentice Hall, 1991.
9. Marino R., Yomei P. *Nonlinear control design: geometric, adaptive, and robust*. London, New York, Prentice Hall, 1995. 396 p.

10. Filiushov V.Yu. Linearizatsiya obratnoi svyaz'yu: evristicheskii podkhod [Feedback linearization: heuristic approach]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 1 (83), pp. 37–46.
11. Filiushov V.Yu. Primery ispol'zovaniya nelineinykh obratnykh svyazei dlya nelineinykh ob"ektov [Examples of applying of nonlinear output injection for nonlinear objects]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 3 (85), pp. 61–70.
12. Voevoda A.A., Ivanov A.E. Primer modal'nogo sinteza dlya nelineinogo ob"ekta s ispol'zovaniem nelineinykh obratnykh svyazei [Modal synthesis example for nonlinear object using nonlinear feed-backs]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2013, no. 2 (72), pp. 3–9.
13. Filiushov V.Yu. Primery strukturnykh preobrazovaniy nelineinogo ob"ekta [Examples of structural transformation of nonlinear object]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 4 (86), pp. 72–82.
14. Voevoda A.A., Shoba E.V. O modeli perevernutogo mayatnika [About model inverted pendulum]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2012, no. 1 (67), pp. 3–14.
15. Voevoda A.A., Filiushov V.Yu. Linearizatsiya obratnoi svyaz'yu: perevernutyi mayatnik [Feedback linearization: inverted pendulum]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 3 (85), pp. 49–60.
16. Voevoda A.A., Filiushov V.Yu. Linearizatsiya obratnoi svyaz'yu: perevernutyi mayatnik [Feedback linearization: inverted pendulum]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 4 (86), pp. 62–71.
17. Voevoda A.A., Filyushov V.Yu. Linearizatsiya dvukhkanal'noi sistemy s nelineinym vykhodom pri pomoshchi obratnoi svyazi [Linearization by feedback of two channel object]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 4 (86), pp. 49–61.
18. Filyushov V.Yu. Linearizatsiya nelineinogo trekhkanal'nogo dinamicheskogo ob"ekta obratnoi svyaz'yu [Linearization of multichannel object by output



injectio]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2017, no. 1 (66), pp. 74–85.

19. Onishchenko G.B. *Elektricheskii privod* [Electric drive]. Moscow, Academia Publ., 2006. 288 p.

20. Ivanov-Smolenskii A.V. *Elektricheskie mashiny* [Electric machines]. Moscow, Energiya Publ., 1980. 928 p.

21. Voevoda A.A., Filyushov V.Yu. Linearizatsiya obratnoi svyaz'yu [Feedback linearization]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 2 (84), pp. 68–76.

Для цитирования:

Филишов В.Ю. Нелинейный объект: линеаризация обратными связями // Сборник научных трудов НГТУ. – 2018. – № 2 (92). – С. 36–80. – doi: 10.17212/2307-6879-2018-2-36-80.

For citation:

Filiushov V.Yu. Nelineinyi ob"ekt: linearizatsiya obratnymi svyazyami [Examples of structural transformation of nonlinear object]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 2 (92), pp. 36–80. doi: 10.17212/2307-6879-2018-2-36-80.

СОВРЕМЕННЫЕ  
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 629.05

DOI: 10.17212/2307-6879-2018-2-81-92

**ОБЗОР ТЕХНОЛОГИИ НАВИГАЦИИ МОБИЛЬНЫХ  
РОБОТОВ OMNI-DIRECTIONAL VSLAM\***

В.Г. АЛТУХОВ

*630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, аспирант кафедры автоматки. E-mail: altukhov@ngs.ru*

Технология SLAM (от англ. Simultaneous Localization And Mapping – одновременное построение карты и определение местоположения) на сегодняшний день является передовой в вопросе навигации мобильных роботов [1]. Она необходима для успешного перемещения робота по различного типа местности и заключается в построении карты окружающей среды и отслеживании его (робота) местоположения в этой среде, что осуществляется его бортовым компьютером. Как правило, информацию об окружающей среде бортовая система робота получает посредством датчиков, таких как лазерные и ультразвуковые дальномеры, одна, две или три видеокамеры. Ультразвуковые и лазерные датчики являются весьма дорогостоящими и, что самое главное, массивными устройствами, а использование ограниченного количества видеокамер хотя и дает намного больше информации (цвет, освещенность и т. д.), но обеспечивает далеко не полный обзор окружающего робот пространства. В связи с этим автор статьи ознакомился с публикациями, связанными с вопросами получения мобильным роботом информации об окружающей среде с большим полем обзора. Было обнаружено, что существует так называемая технология Omni-directional SLAM (от англ. Omni-directional – многонаправленный), которая для получения информации об окружающей среде использует следующие датчики: катадиоптрическая камера, массив из камер или камера со сверхширокоугольным объективом. В данном обзоре представляется информация о таких устройствах и разработках алгоритмов SLAM, работающих на их основе.

**Ключевые слова:** навигация мобильных роботов, SLAM, vSLAM, построение карты, определение местоположения, многонаправленные камеры, катадиоптрические камеры, “lady-bug” камеры, камеры со сверхширокоугольным объективом

---

\* Статья получена 24 января 2018 г.

## ВВЕДЕНИЕ

За последние несколько десятилетий проблема автономного функционирования мобильных роботов широко обсуждается в различных областях науки. Помимо робототехники, это и область обработки изображений, и машинное обучение, и многое другое. Большинство таких областей задействовано в вопросах навигации, которая является неотъемлемой частью автономных роботов. На сегодняшний день самая передовая технология, реализующая навигацию роботов, – это SLAM. В свою очередь, самые перспективные алгоритмы SLAM – те, которые основываются на использовании камер, так называемые алгоритмы visual SLAM (vSLAM). Это связано с тем, что, в отличие от алгоритмов, основанных на использовании лазерных или ультразвуковых датчиков в качестве основных источников информации об окружающей среде, алгоритмы vSLAM могут с помощью камер получать более детальную информацию об объектах окружающей среды, об их цвете. Кроме того, камеры – более легкие устройства, что позволяет использовать vSLAM на мобильных роботах с ограниченной грузоподъемностью. Вместе с тем визуальную информацию намного сложнее обрабатывать, и ввиду этого многие проблемы vSLAM еще далеки от своих оптимальных решений.

Обычно vSLAM, действующий в качестве системы навигации мобильного робота, анализирует изображения окружающей среды, полученные посредством одной, двух или трех видеокамер. Использование такого количества датчиков создает ограничение поля обзора мобильного робота, что делает его уязвимым как при построении карты, где ввиду резких поворотов при появлении преград или в плохоструктурированной местности алгоритм может потерять ориентиры, используемые при ее построении, так и при идентификации динамических объектов, которые могут внезапно преградить роботу путь или в худшем случае стать причиной его поломки. В связи с этим автор изучил работы по исследованию так называемых многонаправленных датчиков и основанных на них алгоритмов vSLAM. Многонаправленные датчики (в случае vSLAM) – это основанные на камерах датчики, способные получать большую визуальную информацию об окружающей среде за счет значительно большего угла обзора – от примерно 150 градусов до полного обзора вокруг своей оси и полного «сферического» обзора. Такой тип сенсоров позволяет снизить влияние вышеперечисленных проблем, но, в свою очередь, усложняет алгоритмы vSLAM ввиду своих особенностей.

В разделе 1 дано краткое описание основных многонаправленных датчиков; в разделе 2 рассказано о существующих алгоритмах vSLAM, базирующихся на таких датчиках; в разделе 3 описана методологическая база, которая применяется в исследованиях алгоритмов многонаправленного SLAM.

## 1. МНОГОНАПРАВЛЕННЫЕ ДАТЧИКИ

На сегодняшний день существует три основных типа многонаправленных датчиков: катадиоптрические датчики, камеры со сверхширокоугольным объективом и массивы камер с пересечением области видимости и без ее пересечения. Далее последует описание каждой из этих разновидностей.

### 1.1. Катадиоптрические сенсоры

Катадиоптрическая камера представляет собой установку, состоящую из обычной видеокамеры и изогнутого зеркала (в форме полусферы, конуса и т. п.), расположенного напротив (рис. 1).

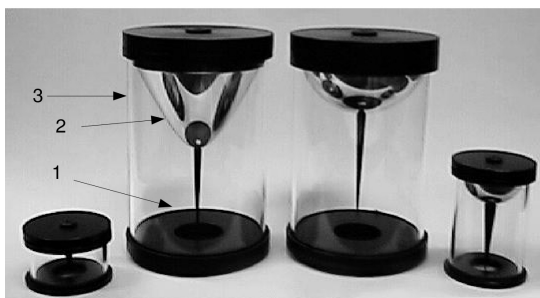


Рис. 1. Внешний вид катадиоптрического сенсора:

1 – отверстие для видеокамеры, 2 – изогнутое зеркало,  
3 – стеклянный корпус



Рис. 2. Изображение на выходе катадиоптрической камеры

На рис. 2 изображен кадр, полученный на выходе катадиоптрической камеры, из которого можно понять, что робот с помощью нее может наблюдать за окружающей средой на 360 градусов вокруг своей вертикальной оси. При этом картинка достаточно искажена, а в центре находится ненаблюдаемый участок – отражение камеры в зеркале.

## 1.2. Массив камер

Массив камер, или lady-bug камера (от англ. *lady-bug* – «божья коровка»), представляет собой набор направленных в разные стороны камер с пересекающимся (стерео) или непересекающимся полем обзора, которые могут предоставить роботу наиболее полный обзор окружающей среды (рис. 3).



Рис. 3. Примеры внешнего вида камеры «божья коровка»

Изображение, захваченное с подобного девайса, показано на рис. 4. Данное изображение получено с датчика с малым количеством камер (рис. 3, камеры слева) и без пересечения полей зрения. Также мы можем заметить, что картинка весьма искажена – это опять же зависит от количества камер и параметров их оптической системы.



Рис. 4. Выходное изображения камеры «божья коровка»

Вышеперечисленные типы многонаправленных датчиков, кроме прочего, были исследованы в работе [2].

### 1.3. Камера со сверхширокоугольным объективом

Другое название такого типа сенсора – камера «рыбий глаз» (от англ. *fishyecamera*) – это обычная камера с широкоугольными линзами в оптическом тракте (рис. 5).



Рис. 5. Примеры внешнего вида камеры «рыбий глаз»

Изображение, полученное посредством камеры *fisheye*, показано на рис. 6. На нем мы можем наблюдать средний уровень искажения и наименьший угол обзора в сравнении с остальными сенсорами. Угол обзора в таких камерах может достигать порядка 220 градусов, причем с его увеличением будет расти и искажение.

Геометрические модели всех вышеперечисленных разновидностей сенсоров кратко рассматриваются в работе [3].

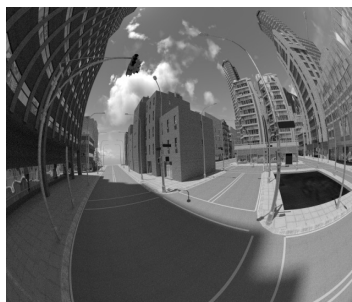


Рис. 6. Выходное изображение камеры «рыбий глаз»

## 2. АЛГОРИТМЫ МНОГОНАПРАВЛЕННОГО SLAM

По сути алгоритмы многонаправленного *vSLAM* являются обычными *vSLAM* (ORB-SLAM, LSD-SLAM и др.), адаптированными под использование многонаправленных датчиков. По большей части их адаптация заключается в замене модели камеры для корректного представления точек окружающей среды на карте, строящейся алгоритмом *vSLAM*.

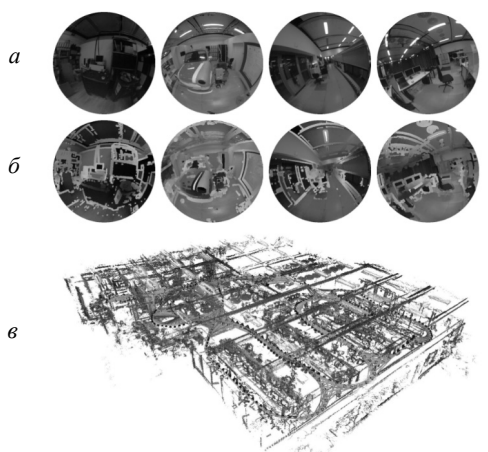
В статье [4] авторы описывают то, как они интегрировали модель сферической камеры (МСК) для катадиоптрической системы в алгоритм vSLAM. МСК – это проекционная модель, которая объединяет катадиоптрическую систему и обычную видеокамеру. Авторы внедрили данную систему в vSLAM, основанную на расширенном фильтре Калмана (РФК); провели эксперимент, сравнив многонаправленную систему с алгоритмом, использующим стандартную перспективную камеру. В результате эксперимента выяснилось, что использование многонаправленной камеры дает намного лучшую точность оценки траектории видеокамеры.

В публикации [5] авторы также использовали сферическую модель камеры в алгоритме vSLAM, основанном на РФК, но с камерой типа «божья коровка». Был проведен эксперимент, который доказал эффективность такого подхода при использовании его на неизвестной местности.

Исследователи, авторы статьи [3], предложили реализацию монокулярного SLAM (алгоритм, использующий одну видеокамеру для построения карты и локализации робота) для массива камер и камеры типа «рыбий глаз». Алгоритмы отслеживания местоположения и построения карты были адаптированы под общую многонаправленную модель, которая позволяет работать с камерами, угол обзора которых превышает 150 градусов. Такой подход, в сравнении с ранее предложенными обычными методами, позволил не только получать сравнительно большую часть информации об окружающей среде, но и сделать систему навигации более надежной в случае с вырожденным (в частности, с вращательным) движением. На рис. 7 показан результат работы алгоритма при построении карты окружающей среды.

Авторы статьи [6] с помощью 3D-моделирования сгенерировали набор изображений, которые соответствуют захваченным кадрам с обычной (перспективной) камеры, камеры типа «рыбий глаз» и катадиоптрического датчика. Данные наборы кадров представляют собой изображения, полученные как внутри помещения, так и в городской среде. Модели камер, используемых при генерировании данных изображений, были внедрены в систему навигации визуальной одометрии (от англ. *Visual Odometry*). После этого был проведен опыт, сравнивающий эффективность различных типов датчиков при использовании их как внутри помещения, так и снаружи. В результате эксперимента выяснилось, что катадиоптрические сенсоры при одинаковом разрешении захваченного кадра эффективнее для использования внутри помещения, тогда как перспективные камеры лучше подходят для городской среды.

Авторы [6] также предоставили общий доступ к этим сгенерированным наборам изображений (датасетам), о целесообразности использования которых будет рассказано в разделе 3.



*Рис. 7.* Этапы реконструкции карты местности:  
*а* – изображения, захваченные с помощью камеры «рыбий глаз»; *б* – выделение особенностей на полученных изображениях; *в* – соединение выделенных особенностей, результатом чего является построенная карта местности

Исследования в области алгоритмов vSLAM, основанных на многонаправленных датчиках, проводились также в работах [7–11].

### 3. МЕТОДОЛОГИЧЕСКАЯ БАЗА ИССЛЕДОВАНИЯ

Для выполнения исследования, результатом которого будет являться эксперимент над разработанной системой визуальной навигации, использующей в качестве органа зрения один из многонаправленных датчиков, необходимо предпринять следующие шаги:

а) осуществить обоснованный выбор исследуемых алгоритма навигации и датчика;

б) реализовать данную систему навигации;

в) исследовать ее экспериментально.

Далее вкратце будет описан каждый шаг.

Выбор алгоритма навигации и датчика должен производиться на основании опыта предыдущих исследований. Так, изучив публикации по современным алгоритмам, можно сделать предположения об их плюсах и минусах в связке с многонаправленными датчиками (связанные с ними работы также необходимо



изучить). В первую очередь следует обособленно изучить работы по алгоритмам и датчикам, которые ранее не исследовались вместе, ввиду того что это может сформировать такой пункт исследования, как научная новизна. При этом не стоит забывать о том, что такая комбинация алгоритма и датчика может дать результаты, которые будут хуже уже существующих. Это, конечно, тоже будет являться результатом, но не обоснованным ничем, кроме новизны.

Так как на сегодняшний день алгоритмы навигации в своем большинстве разработаны для стереокамерных или монокулярных датчиков, в основе которых лежат обычные цифровые камеры, реализация системы навигации заключается в адаптации выбранного алгоритма визуальной навигации для использования с многонаправленным датчиком, будь то катадиоптрическая камера, массив из камер или камера с широкоугольной линзой. Адаптация заключается в изменении, в алгоритме, математической модели используемого датчика и оптимизации других функций для работы с ним.

Весьма проблематично, даже в наши дни, иметь в своем распоряжении многонаправленные камеры. Для того чтобы это не становилось причиной «проста» исследования, исследователи, которые обладают подобными сенсорами, выкладывают в общее пользование так называемые «датасеты» – последовательность изображений, сделанную каким-либо датчиком. Датасеты полезны не только для восполнения пробела, связанного с отсутствием нужного датчика, но и для сравнения результатов экспериментов различных исследователей. Экспериментальные исследования двух разных алгоритмов, производящиеся на базе одного датасета, дают данные для более детального сравнительного анализа. Например, в случае с многонаправленными датчиками (а именно катадиоптрическим сенсором и камерой с широкоугольной линзой) имеется общедоступный датасет, исследованный в статье [6], который может быть использован при выполнении экспериментальной части исследования.

## **ЗАКЛЮЧЕНИЕ**

В данной статье проведен краткий обзор технологии многонаправленного vSLAM. В частности, рассмотрены типы датчиков, применяемых в рамках данной технологии для получения информации об окружающей среде, и описаны уже реализованные алгоритмы vSLAM, основанные на датчиках такого типа. Также для прояснения вопроса проведения исследований таких систем рассказывается о методологической базе. В целом на основании изученной литературы можно считать, что навигация мобильных роботов посредством технологии Omni-directional vSLAM является весьма перспективной темой исследований.

**СПИСОК ЛИТЕРАТУРЫ**

1. RoboCraft (технология SLAM) [Электронный ресурс]. – URL: <http://robocraft.ru/blog/technology/724.html> (дата обращения: 22.06.2018).
2. Omnidirectional sensors for mobile robots / S. Blumenthal, D. Droschel, D. Holz, T. Linder, P. Molitor, H. Surmann // Workshop Proceedings of SIMPAR 2008, Venice, Italy, November, 3–7, 2008. – Venice, 2008. – P. 414–425.
3. *Caruso D., Engel J., Cremers D.* Large-scale direct slam for omnidirectional cameras // 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 28 – October 2, 2015, Hamburg, Germany. – Hamburg, 2015. – P. 141–148.
4. *Rituerto A., Puig L., Guerrero J.* Visual SLAM with an omnidirectional camera // 2010 20<sup>th</sup> International Conference on Pattern Recognition CPR 2010, 23–26 August 2010, Istanbul, Turkey: Proceedings. – Istanbul, 2010. – P. 348–351.
5. An omni-directional vSLAM based on spherical camera model and 3D modeling / Guofeng Tong, Zizhang Wu, NingLong Weng, Wenbo Hou // Proceedings of the 10th World Congress on Intelligent Control and Automation. – Beijing, China, 2012. – P. 4551–4556.
6. Benefit of large field-of-view cameras for visual odometry / Z. Zhang, H. Rebecq, C. Forster, D. Scaramuzza // 2016 IEEE International Conference on Robotics and Automation (ICRA). – Stockholm, Sweden, 2016. – P. 801–808.
7. *Kaess M., Dellaert F.* Probabilistic structure matching for visual SLAM with a multi-camera rig // Computer Vision and Image Understanding. – 2010. – Vol. 114. – P. 286–296.
8. *Carrera G., Angeli A., Davison A.J.* SLAM-based automatic extrinsic calibration of a multi-camera rig // 2011 IEEE International Conference on Robotics and Automation. – Shanghai, China, 2011. – P. 2652–2659.
9. *Davison A., González Y., Kita N.* Real-time 3D SLAM with wide-angle vision // IFAC Proceedings Volumes. – 2004. – Vol. 37 (8). – P. 868–873.
10. *Scaramuzza D., Siegwart R.* Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles // IEEE Transactions on Robotics. – 2008. – Vol. 24 (5). – P. 1015–1026.
11. *Kim J., Chung M.* SLAM with Omni-directional stereo vision sensor // Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. – Las Vegas, NV, 2003. – Vol. 1. – P. 442–447.

*Алтухов Виктор Григорьевич*, аспирант кафедры автоматике Новосибирского государственного технического университета. Основное направление научных исследований – системы автоматического управления, робототехника. E-mail: altukhov@ngs.ru

DOI: 10.17212/2307-6879-2018-2-81-92

## **A survey of mobile robot's navigation technology Omni-directional vSLAM\***

**V.G. Altukhov**

*Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, the post-graduate student of Department Automatics. E-mail: altukhov@ngs.ru*

Today "Simultaneous Localization and Mapping" technology is most advanced in the issue of mobile robots navigation [1]. It is necessary to successfully moving the robot at a different type of location and is to building an environmental map and tracking its (robot) location in this environment, which is carried out by its on-board computer. Typically, the robot's on-board system receives information about the environment through sensors such as laser and ultrasonic rangefinders, one, two or three cameras. Ultrasonic and laser sensors are very expensive and massive devices, and the use of a limited number of video cameras, although they give much more information (color, illumination, etc.), but provide a quite short view of the robot's surrounding space. In this regard, the author of the article decided to get acquainted with publications related to the issues of obtaining mobile information about the environment with a large field of view. It was discovered that there is a so-called "Omni-directional SLAM" technology, which uses the following sensors to obtain information about the environment: a catadioptric cameras, an arrays of cameras or a cameras with a wide-angle lens. In this review, information on such devices and development of SLAM algorithms working on their basis is presented.

**Keywords:** navigation of mobile robots, SLAM, vSLAM, mapping, localization, omni-directional cameras, catadioptric cameras, "lady-bug" cameras, fisheye cameras

---

\* Received 24 January 2018.

## REFERENCES

1. *RoboCraft (tekhnologiya SLAM)* [Robocraft (SLAM technology)]. Available at: <http://robocraft.ru/blog/technology/724.html> (accessed 22.06.2018).
2. Blumenthal S., Droschel D., Holz D., Linder T., Molitor P., Surmann H. Omnidirectional sensors for mobile robots. *Workshop Proceedings of SIMPAR 2008*, Venice, Italy, November, 3–7, 2008, pp. 414–425.
3. Caruso D., Engel J., Cremers D. Large-scale direct slam for omnidirectional cameras. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 28 – October 2, 2015, Hamburg, Germany, pp. 141–148.
4. Rituerto A., Puig L., Guerrero J. Visual SLAM with an omnidirectional camera. *2010 20<sup>th</sup> International Conference on Pattern Recognition CPR 2010*, 23–26 August 2010, Istanbul, Turkey: proceedings, pp. 348–351.
5. Guofeng Tong, Zizhang Wu, NingLong Weng, Wenbo Hou. An Omni-directional vSLAM based on spherical camera model and 3D modeling. *Proceedings of the 10th World Congress on Intelligent Control and Automation*, Beijing, China, 2012, pp. 4551–4556.
6. Zhang Z., Rebecq H., Forster C., Scaramuzza D. Benefit of large field-of-view cameras for visual odometry. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016, pp. 801–808.
7. Kaess M., Dellaert F. Probabilistic structure matching for visual SLAM with a multi-camera rig. *Computer Vision and Image Understanding*, 2010, vol. 114, pp. 286–296.
8. Carrera G., Angeli A., Davison A.J. SLAM-based automatic extrinsic calibration of a multi-camera rig. *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 2652–2659.
9. Davison A., González Y., Kita N. Real-time 3D SLAM with wide-angle vision. *IFAC Proceedings Volumes*, 2004, vol. 37 (8), pp. 868–873.
10. Scaramuzza D., Siegwart R. Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, 2008, vol. 24 (5), pp. 1015–1026.
11. Kim J., Chung M. SLAM with omni-directional stereo vision sensor. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2003, vol. 1, pp. 442–447.

Для цитирования:

Алтухов В.Г. Обзор технологии навигации мобильных роботов Omni-directional vSLAM // Сборник научных трудов НГТУ. – 2018. – № 2 (92). – С. 81–92. – doi: 10.17212/2307-6879-2018-2-81-92.

For citation:

Altukhov V.G. Obzor tekhnologii navigatsii mobil'nykh robotov Omni-directional vSLAM [A survey of mobile robot's navigation technology Omni-directional vSLAM]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 2 (93), pp. 81–92. doi: 10.17212/2307-6879-2018-2-81-92.

## ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ С ПРЕДВАРИТЕЛЬНО РАССЧИТАННОЙ СТРУКТУРОЙ\*

А.А. ВОЕВОДА<sup>1</sup>, Д.О. РОМАННИКОВ<sup>2</sup>

<sup>1</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, доктор технических наук, профессор кафедры автоматики. E-mail: ucit@ucit.ru

<sup>2</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, кандидат технических наук, доцент кафедры автоматики. E-mail: dmitry.romannikov@gmail.com

При разработке моделей, основанных на искусственных нейронных сетях, выбор строится на эмпирических знаниях исследователя, что крайне негативно сказывается на пороге вхождения в технологию, а также на обоснованности и корректности работы модели, времени обучения и других ее аспектах. В статье рассматривается подход к формированию структуры нейронной сети на основе предварительного формирования структурной схемы системы и расчета или подбора ее коэффициентов. Целью является демонстрация того, что синтез структуры нейронной сети может быть выполнен не только интуитивным подбором слоев и связей. Для компактности предлагаемый подход показан на примере выбора максимального значения из двух чисел (используемые в примере числа «-1» и «1»). В результате составлена структурная схема нейронной сети и подобраны значения ее коэффициентов. Экспериментами показано, что скорость обучения нейронной сети при предварительно рассчитанных коэффициентах значительно выше, чем при случайных коэффициентах, взятых из нормального распределения в диапазоне  $[-1, 1]$ . Также проведены другие эксперименты по обучению нейронной сети с максимальными / минимальными начальными значениями коэффициентов из диапазона  $[4, 20]$ . В результате эксперимента при всех используемых начальных условиях получены различные значения коэффициентов, что приводит к выводу о том, что существует неединственное решение поставленной задачи, что несомненно является отдельной темой исследования. Также стоит отметить, что скорость обучения нейронной сети с синтезированной структурой выше, чем с у сетей прямого распространения с несколькими слоями (отдельно стоит сказать, что выбор количества слоев и нейронов является неоднозначным).

**Ключевые слова:** нейронные сети, сети Петри, искусственный интеллект, преобразование, функция активации, keras, регуляризация, обучение

---

\* Статья получена 24 января 2018 г.

## ВВЕДЕНИЕ

В настоящее время актуальной задачей является создание систем, способных к обучению или на основании обучения, вместо разработки алгоритмов [1–6]. Для решения данных задач применяют различные инструменты и способы, начиная от случайных лесов (*randomtrees*) и заканчивая опорной машиной векторов (*supportvectormachine*). В настоящее время методы, основанные на использовании нейронных сетей [1–4], имеют наибольшую популярность в силу того, что с их помощью можно создавать более сложные модели. Так, в работах [1–4, 7] используются нейронные сети с возможностью обучения срока хранения информации, в [8] приводятся нейронные сети с памятью. В работе [9] исследуется задача создания нейронной машины Тьюринга, или частный случай использования нейронных сетей в алгоритмических задачах – задача сортировки [10].

Однако в настоящее время в описании моделей нейронных сетей [11–16] существует ряд нерешенных вопросов, к которым относятся выбор и разработка алгоритмов быстрого обучения (в частности, обоснование выбора структуры нейронной сети).

Данная работа посвящена исследованию выбора структуры нейронной сети на основании предварительно рассчитанной модели.

## 1. ПОСТАНОВКА ЗАДАЧИ И ЕЕ РЕШЕНИЕ

В настоящее время задача выбора структуры ИНС является нерешенной задачей. На практике выбор вида ИНС осуществляется эмпирически на основе опыта исследователя и множества попыток обучения. Таким образом, задача обоснования выбора структуры ИНС и /или ее предварительный расчет на основе других моделей является актуальной задачей.

В работе предлагается подход к формированию ИНС на основе построенной модели. В качестве исследуемого объекта будем использовать функцию выбора максимального числа из двух поступивших на вход системы (в работе ограничимся набором из двух чисел «-1» и «1»), несмотря на то что в классическом варианте используются числа «0» и «1»).

## 2. ИССЛЕДОВАНИЕ

Рассмотрим задачу определения наибольшего из двух чисел. Будем решать данную задачу путем расчета (или подбора) параметров нейронной сети, а саму структуру будем представлять в виде рисунка (см. ниже). На данном рисунке представлена структура нейронной сети, в которой входные данные поступают на входы «x1» и «x2». Далее умноженные на коэффициенты  $W_1$  и

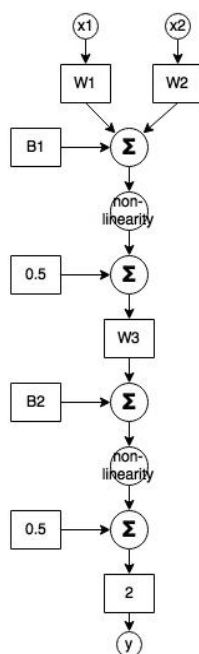
W2 поступают на сумматор, в котором дополнительно к ним добавляется коэффициент B1. Выход сумматора соединен с нелинейностью (в работе используется функция сигмоида). От выхода первого нейрона дополнительно отнимается 0.5 для смещения диапазона: выход сигмоида ограничен диапазоном значений  $0 \dots 1$ , и для смещения его в отрицательную часть необходимо отнять половину диапазона и масштабировать на коэффициент W3. Второй нейрон состоит из одного входа (соединенного с первым нейроном) и приведенного к диапазону  $[-1, 1]$  выхода.

Эмпирически можно подобрать коэффициенты данной схемы:  $W1 = 4$ ,  $W2 = 4$ ,  $B1 = 6$ ,  $W3 = 20$ ,  $B2 = 0$ . Данные коэффициенты дают на выходе следующие значения (далее идут значения в формате «x1/x2 входы: выход»): «1/1 : 0.9866», «-1/1 : 0.9862», «1/-1 : 0.9862», «-1/-1 : -0.9565».

При обучении методом обратного распространения ошибки после 20 000 шагов при случайных начальных условиях в диапазоне от 0 до 1 и шаге обучения 0.01. Получим следующие коэффициенты:  $W1 = 2.6$ ,  $W2 = 2.47$ ,  $B1 = 2.35$ ,  $W3 = 4.46$ ,  $B2 = 0$ , при которых получим следующие результаты: «1/1 : 0.80», «-1/1 : 0.71», «1/-1 : 0.71», «-1/-1 : -0.75». При шаге 0.1 и случайных начальных условиях в диапазоне от 0 до 1 после 3000 итераций: «1/1 : 0.974», «-1/1 : 0.972», «1/-1 : 0.972», «-1/-1 : -0.971» при коэффициентах  $W1 = 4.6$ ,  $W2 = 4.37$ ,  $B1 = 4.63$ ,  $W3 = 8.72$ ,  $B2 = 0$ .

Стоит отметить, что изменение начальных условий на 10 %, т. е. подача вместо «1» значения «0.9», практически не влияет на конечный результат: «0.9/0.9 : 0.9866», «-0.9/0.9 : 0.9862», «0.9/-0.9 : 0.9862», «-0.9/-0.9 : -0.872».

Рассмотрим влияние начальных условий на коэффициенты, полученные при обучении. По результатам предыдущих экспериментов видно, что значения коэффициентов находятся в диапазоне  $4 \dots 20$ . Проведем эксперимент, в котором будем использовать начальные условия из первой колонки таблицы. Результаты посчитанных коэффициентов после 10 000 итераций обучения показаны в колонке 2.



Схематичное представление структуры нейронной сети



Таблица с начальными условиями и обученными коэффициентами

Начальные условия (W1, W2, B1, W3)				Коэффициенты после обучения (W1, W2, B1, W3)			
1				2			
4	4	4	4	7.619	7.619	7.619	12.104
4	4	4	20	6.986	6.986	6.986	27.694
4	4	20	4	9.763	9.763	14.236	11.157
4	4	20	20	8.830	8.830	15.169	21.452
4	20	4	4	9.763	14.236	9.763	11.157
4	20	4	20	8.830	15.169	8.830	21.453
4	20	20	4	7.610	20.000	20.000	12.108
4	20	20	20	6.931	20.003	19.996	27.748
20	4	4	4	14.236	9.763	9.763	11.158
20	4	4	20	15.169	8.830	8.832	1.453
20	4	20	4	20.000	7.610	20.000	12.108
20	4	20	20	20.003	6.931	19.996	27.748
20	20	4	4	20.000	20.000	7.611	12.108
20	20	4	20	20.000	20.000	6.933	27.748
20	20	20	4	21.121	21.121	18.878	12.116
20	20	20	20	20.012	20.012	19.987	27.882

Из результатов, представленных в таблице, можно сделать вывод о том, что при различных начальных условиях получаются различные коэффициенты итоговой нейронной сети.

## ЗАКЛЮЧЕНИЕ

В данной статье проанализирован пример решения задачи выбора максимального числа из двух с ограниченным диапазоном значений «-1» и «1». При решении данной задачи была предварительно разработана структура нейронной сети и рассчитаны коэффициенты. Итоговая структура нейронной сети содержат два нейрона. Выполнен ряд экспериментов по обучению полученной нейронной сети. В частности, за 20 000 итераций обучения при начальных значениях, выбранных случайным равномерным образом из диапазона [0, 1], получены результаты (в формате «x1/x2 входы: выход»): «1/1 : 0.80», «-1/1 : 0.71», «1/-1 : 0.71», «-1/-1 : -0.75». При использовании рассчитанных начальных условий за 3000 итераций получены более точные результаты, а именно: «1/1 : 0.974», «-1/1 : 0.972», «1/-1 : 0.972», «-1/-1 : -0.971».

По результатам эксперимента по обучению из возможных начальных условий были получены различные значения коэффициентов нейронной сети. Из этого можно сделать вывод о существовании множества решений в нейронной сети (даже состоящей только из двух нейронов).

## СПИСОК ЛИТЕРАТУРЫ

1. *Bishop C.* Pattern recognition and machine learning. – New York: Springer, 2007. – 738 p. – (Information science and statistics).
2. *Richert W., Coelho L.* Building machine learning systems with Python. – Birmingham: Packt Publ., 2013. – 290 p.
3. *Hastie T., Tibshirani R., Friedman J.* The elements of statistical learning: data mining, inference, and prediction. – 2<sup>nd</sup> ed. – New York: Springer, 2013. – 745 p. – (Springer series in statistics).
4. *Lantz B.* Machine learning with R. – Birmingham: Packt Publ., 2013. – 396 p.
5. *Mohri M., Rostamizadeh A., Talwalkar A.* Foundations of machine learning. – Cambridge, MA: MIT Press, 2012. – 432 p. – (Adaptive computation and machine learning series).
6. *Conway D., White J.M.* Machine learning for hackers. – Sebastopol, CA: O'Reilly, 2012. – 324 p.
7. *Haykin S.* Neural networks: a comprehensive foundation. – New York: MacMillan Publ., 1994. – 1104 p.
8. End-to-end memory networks / S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus // Advances in Neural Information Processing Systems 28. – Montreal, Canada, 2015. – P. 2440–2448.
9. *Graves A. Wayne G., Danihelka I.* Neural Turing machines // ArXiv.org. – 2014. – arXiv:1410.5401. – P. 1–26.
10. *Воевода А.А., Полубинский В.Л., Романников Д.О.* Сортировка массива целых чисел с использованием нейронной сети // Научный вестник НГТУ. – 2016. – № 2 (63). – С. 151–157.
11. *Hastie T., Tibshirani R., Friedman J.* The elements of statistical learning: data mining, inference, and prediction. – New York: Springer. 2001. – 533 p.
12. *Hagan M., Demuth H., Jesús O.* A neural network predictive control system for paper mill wastewater treatment // Engineering Applications of Artificial Intelligence. – 2003. – Vol. 16 (2). – P. 121–129.
13. *Touretzky D.S., Pomerleau D.A.* What is hidden in the hidden layers? // Byte. – 1989. – Vol. 14. – P. 227–233.
14. *LeCun Y., Bengio Y., Hinton G.* Deep learning // Nature. – 2015. – Vol. 521, N 7553. – P. 436–444.

15. Dropout: a simple way to prevent neural networks from overfitting / N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov // Journal of Machine Learning Research. – 2014. – Vol. 15. – P. 1929–1958.

16. Early stopping without a validation set / M. Mahsereci, L. Balles, C. Lassner, P. Hennig // ArXiv.org. – 2017. – arXiv:1703.09580.

**Воевода Александр Александрович**, доктор технических наук, профессор кафедры автоматике Новосибирского государственного технического университета. Основное направление научных исследований – управление многоканальными объектами. Имеет более 200 публикаций. E-mail: ucit@ucit.ru

**Романников Дмитрий Олегович**, кандидат технических наук, доцент кафедры автоматике Новосибирского государственного технического университета. Основное направление научных исследований – нейронные сети, сети Петри. Имеет более 50 публикаций. E-mail: dmitry.romannikov@gmail.com

DOI: 10.17212/2307-6879-2018-2-93-100

## **Training of neural network with preliminary calculated structure\***

**А.А. Voevoda<sup>1</sup>, D.O. Romannikov<sup>2</sup>**

<sup>1</sup> *Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, doctor of Technical Sciences, professor of the automation department. E-mail: ucit@ucit.ru*

<sup>2</sup> *Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, candidate of Technical Sciences, associate professor of the automation department. E-mail: dmitry.romannikov@gmail.com*

In the development of models based on artificial neural networks, the choice of the neural network model is based on the empirical knowledge of the researcher, which has a very negative effect on the threshold of entry into the technology, the validity and correctness of the model's work, training time and other aspects. The article deals with the approach to the formation of the neural network structure on the basis of the preliminary formation of the structural scheme of the system and the calculation or selection of its coefficients. The goal is to demonstrate that the synthesis of the neural network structure can be performed not only by intuitively selecting layers and links. For compactness, the proposed approach is illustrated by the example of selecting the maximum value from two numbers (used in the example numbers - "-1", "1"). As a result, a structural diagram of the neural network is made up and the values of its coefficients are chosen. Experiments show that the speed of training a neural network

---

\*Received 24 January 2018.

with previously calculated coefficients is much higher than for random coefficients taken from the normal distribution in the range  $[-1, 1]$ . Also, other experiments were conducted on training the neural network with the maximum / minimum initial values of the coefficients from the range  $[4, 20]$ . As a result of the experiment, for all the initial conditions used, different values of the coefficients were obtained, which leads to the conclusion that there is a non-unique solution to the problem posed, which is undoubtedly a separate topic of the study. It should also be noted that the learning speed of a neural network with a synthesized structure is higher than that of networks with direct propagation with several layers (separately it is necessary to say that the choice of the number of layers and neurons is ambiguous).

**Keywords:** neural networks, Petri nets, artificial intelligence, transformation, activation function, keras, regularization, training

## REFERENCES

1. Bishop C. *Pattern recognition and machine learning. Information science and statistics*. New York, Springer, 2007. 738 p.
2. Richert W., Coelho L. *Building machine learning systems with Python*. Birmingham, Packt Publ., 2013. 290 p.
3. Hastie T., Tibshirani R., Friedman J. *The elements of statistical learning: data mining, inference, and prediction*. 2<sup>nd</sup> ed. *Springer series in statistics*. New York, Springer, 2013. 745 p.
4. Lantz B. *Machine learning with R*. Birmingham, Packt Publ., 2013. 396 p.
5. Mohri M., Rostamizadeh A., Talwalkar A. *Foundations of machine learning. Adaptive computation and machine learning series*. Cambridge, MA, MIT Press, 2012. 432 p.
6. Conway D., White J.M. *Machine learning for hackers*. Sebastopol, CA, O'Reilly, 2012. 324 p.
7. Haykin S. *Neural networks: a comprehensive foundation*. New York, MacMillan Publ., 1994. 1104 p.
8. Sukhbaatar S., Szlam A., Weston J., Fergus R. End-to-end memory networks. *Advances in Neural Information Processing Systems 28*, Montreal, Canada, 2015, pp. 2440–2448.
9. Graves A. Wayne G., Danihelka I. Neural Turing machines. *The Computing Research Repository* 1410.5401, 2014, pp. 1–24.
10. Voevoda A.A., Polubinsky V.L., Romannikov D.O. Sortirovka massiva tselnykh chisel s ispol'zovaniem neironnoi seti [Sorting the array of integers using a neural network]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2016, no. № 2 (63), pp. 151–157.
11. Hastie T., Tibshirani R., Friedman J. *The elements of statistical learning: data mining, inference, and prediction*. New York, Springer, 2001, 533 p.

12. Hagan M., Demuth H., Jesús O. A neural network predictive control system for paper mill wastewater treatment. *Engineering Applications of Artificial Intelligence*, 2003, vol. 16 (2), pp. 121–129.

13. Touretzky D.S., Pomerleau D.A. What is hidden in the hidden layers? *Byte*, 1989, vol. 14, pp. 227–233.

14. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*, 2015, vol. 521, no. 7553, pp. 436–444.

15. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014, vol. 15, pp. 1929–1958.

16. Mahsereci M., Balles L., Lassner C., Hennig P. Early stopping without a validation set. *ArXiv.org*, 2017, arXiv:1703.09580.

Для цитирования:

Воевода А.А., Романников Д.О. Обучение нейронной сети с предварительно рассчитанной структурой // Сборник научных трудов НГТУ. – 2018. – № 2 (92). – С. 93–100. – doi: 10.17212/2307-6879-2018-2-93-100.

For citation:

Voevoda A.A. Romannikov D.O. Obuchenie neironnoi seti s predvaritel'no rasschitannoi strukturoi [Training of neural network with preliminary calculated structure]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 2 (92), pp. 93–100. doi: 10.17212/2307-6879-2018-2-93-100.

## ТРАНСЛЯЦИЯ НЕЙРОННОЙ СЕТИ В СЕТЬ ПЕТРИ\*

А.А. ВОЕВОДА<sup>1</sup>, Д.О. РОМАННИКОВ<sup>2</sup>

<sup>1</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, доктор технических наук, профессор кафедры автоматизи. E-mail: ucit@ucit.ru

<sup>2</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, кандидат технических наук, доцент кафедры автоматизи. E-mail: dmitry.romannikov@gmail.com

В статье приводится пример реализации нейронной сети при помощи сети Петри, что позволяет добавить асинхронности в нейронную сеть – это больше соответствует работе человеческого мозга и, следовательно, может иметь перспективу применения для решения задач искусственного интеллекта. Математический аппарат сетей Петри, с одной стороны, позволяет моделировать поведение асинхронных систем, а с другой – по своей структуре в достаточной степени похож на структуру нейронов, чем и обосновывается выбор сетей Петри. Приведен пример реализации нейронной сети для решения задачи XOR как стандартной задачи в области машинного обучения. В рамках решения этой задачи функции суммирования в нейронной сети представлены в виде переходов, в которых при передаче метки в место происходит суммирование значений входных меток. Функции активации также представлены в виде переходов, в которых для выходных дуг выполняется вызов функций активации (сигмоидальной функции в данном случае). Стоит отметить, что при моделировании реализованная нейронная сеть является асинхронной лишь частично (в частности, для срабатывания переходов требуется наличие меток во всех входящих в нее местах). Данный вариант нейронной сети при помощи сетей Петри является не единственным. В частности, можно также использовать отдельные места и метки для моделирования весовых коэффициентов, что позволит выполнять обучение нейронной сети без модификации ее структуры. Дальнейшим направлением исследования является реализация процедуры обучения нейронной сети с асинхронностью.

**Ключевые слова:** нейронные сети, сети Петри, искусственный интеллект, преобразование, функция активации, keras, регуляризация, обучение

---

\* Статья получена 13 декабря 2017 г.

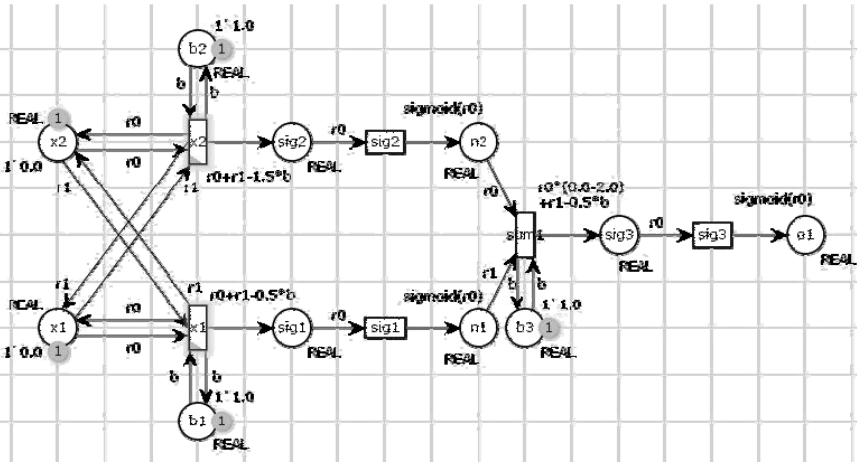
## ВВЕДЕНИЕ

На сегодняшний момент все используемые на практике модели нейронных сетей, к которым относятся модели распознавания изображений [1], речи [2], генерации названий изображений [3] и другие [4–17], реализованы на логике их синхронной работы. При этом работа некоторых частей человеческого мозга является асинхронной, что является мотиватором для реализации модели нейронной сети, в которой присутствуют асинхронные операции между нейронами. В статье предлагается модель нейронной сети для решения классической задачи машинного обучения – задачи исключающего или (*XOR*) реализованной при помощи сети Петри.

## РЕАЛИЗАЦИЯ НЕЙРОННОЙ СЕТИ

Для примера реализации была выбрана классическая задача машинного обучения – реализация модели для решения задачи исключающего или (*XOR*), которая неоднократно решалась в работе [4].

Один из возможных вариантов решения данной задачи приведен в исследованиях [4, 5]. В нем предлагается нейронная сеть, состоящая из двух входов и трех нейронов (один нейрон в скрытом слое), а также предлагаются пред-рассчитанные значения обучаемых параметров. На основании приведенной нейронной сети можно построить сеть Петри (см. рисунок).



Нейронная сеть для решения задачи *XOR*, реализованная при помощи сетей Петри

Данная сеть содержит два входа –  $x_1$  и  $x_2$ . Первый нейрон состоит из  $x_1$  и  $x_2$  входов,  $b_1$  – смещения,  $sig1$  – буферного места для суммирования и  $n_1$  – выходного значения нейрона. Второй входной нейрон и нейрон в выходном слое устроены таким же образом. В нейронах применяется сигмоидальная функция активации.

При моделировании данная нейронная сеть является асинхронной лишь частично. В частности, для срабатывания перехода  $sum1$  требуется наличие меток в местах  $n_1$  и  $n_2$ .

## ЗАКЛЮЧЕНИЕ

В статье предлагается реализация нейронной сети для решения задачи XOR при помощи сетей Петри. Использование сетей Петри для реализации нейронной сети придает им асинхронность срабатывания переходов, что больше соответствует работе человеческого мозга. Стоит отметить, что данный вариант нейронной сети при помощи сетей Петри является не единственным. В частности, можно также использовать отдельные места и метки для моделирования весовых коэффициентов, что позволит выполнять обучение нейронной сети без модификации ее структуры.

## СПИСОК ЛИТЕРАТУРЫ

1. *Krizhevsky A., Sutskever I., Hinton G.E.* ImageNet classification with deep convolutional neural networks // *Advances in Neural Information Processing Systems* 25 (NIPS 2012). – Lake Tahoe, Nevada, 2012. – P. 1090–1098.
2. *Graves A., Mohamed A., Hinton G.* Speech recognition with deep recurrent neural networks // *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2013.* – Vancouver, Canada, 2013. – P. 6645–6649.
3. *Karpathy A., Fei-Fei L.* Deep visual-semantic alignments for generating image descriptions // *IEEE Transactions on Pattern Analysis and Machine Intelligence.* – 2016. – Vol. 39. – P. 664–676.
4. *Haykin S.* *Neural networks and learning machines.* – 3<sup>rd</sup> ed. – New York: Prentice Hall/Pearson, 2009. – 938 p.
5. *Goodfellow I., Bengio Y., Courville A.* *Deep learning.* – Cambridge: MIT Press, 2016. – 800 p.
6. *Воевода А.А., Романников Д.О.* Синтез нейронной сети для решения логико-арифметических задач // *Труды СПИИРАН.* – 2017. – Вып. 54. – С. 205–223.



7. *Haykin S., Deng C.* Classification of radar clutter using neural networks // IEEE Transactions on Neural Networks. – 1991. – Vol. 2. – P. 589–600.
8. *Hastie T., Tibshirani R., Friedman J.* The elements of statistical learning: data mining, inference, and prediction. – New York: Springer, 2001.
9. *Hagan M., Demuth H., Jesús O.* A neural network predictive control system for paper mill wastewater treatment // Engineering Applications of Artificial Intelligence. – 2003. – Vol. 16 (2). – P. 121–129.
10. *Touretzky D.S., Pomerleau D.A.* What is hidden in the hidden layers? // Byte. – 1989. – Vol. 14. – P. 227–233.
11. *LeCun Y., Bengio Y., Hinton G.* Deep learning // Nature. – 2015. – Vol. 521, N 7553. – P. 436–444.
12. Dropout: a simple way to prevent neural networks from overfitting / N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov // Journal of Machine Learning Research. – 2014. – Vol. 15. – P. 1929–1958.
13. Early stopping without a validation set / M. Mahsereci, L. Balles, C. Lassner, P. Hennig // ArXiv.org. – 2017. – arXiv:1703.09580.
14. *Prechelt L.* Early Stopping – but when? // Neural networks: tricks of the trade. – 2<sup>nd</sup> ed. – Berlin; Heidelberg: Springer, 2012. – P. 53–67.
15. *Воевода А.А., Романников Д.О.* Асинхронный алгоритм сортировки массива чисел с использованием ингибиторных сетей Петри // Труды СПИИРАН. – 2016. – Вып. 48. – С. 198–213.
16. *Воевода А.А., Полубинский В.Л., Романников Д.О.* Сортировка массива целых чисел с использованием нейронной сети // Научный Вестник НГТУ. – 2016. – № 2 (63). – С. 151–157.
17. *Voevoda A.A., Romannikov D.O.* A binary array asynchronous sorting algorithm with using Petri nets // Journal of Physics: Conference Series. – 2017. – Vol. 803, N 1. – P. 012178.

**Воевода Александр Александрович**, доктор технических наук, профессор кафедры автоматки Новосибирского государственного технического университета. Основное направление научных исследований – управление многоканальными объектами. Имеет более 200 публикаций. E-mail: ucit@ucit.ru

**Романников Дмитрий Олегович**, кандидат технических наук, доцент кафедры автоматки Новосибирского государственного технического университета. Основное направление научных исследований – нейронные сети, сети Петри. Имеет более 50 публикаций. E-mail: dmitry.romannikov@gmail.com

DOI: 10.17212/2307-6879-2018-2-101-107

## Translation of the neural network in the Petri network\*

A.A. Voevoda<sup>1</sup>, D.O. Romannikov<sup>2</sup>

<sup>1</sup> Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, doctor of Technical Sciences, professor of the automation department. E-mail: ucit@ucit.ru

<sup>2</sup> Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, candidate of Technical Sciences, associate professor of the automation department. E-mail: dmitry.romannikov@gmail.com

The article gives an example of the implementation of a neural network with the help of a Petri net, which allows adding asynchrony to the neural network, which is more in line with the work of the human brain and, therefore, can have a perspective for using to solve problems of artificial intelligence. The mathematical apparatus of Petri networks, on the one hand, allows one to simulate the behavior of asynchronous systems, and with another in its structure is quite similar to the structure of neurons, which justifies the choice of Petri nets. An example of the implementation of a neural network for solving the XOR problem is given as a standard problem in the field of machine learning. In the framework of solving this problem, summation functions in the neural network are represented in the transitions where the value of the input labels occurs when the label is transferred to the place. Activation functions are also represented in the form of transitions in which the output arcs call the activation functions (the sigmoid function in this case). It is worth noting that in the simulation, the implemented neural network is only partially asynchronous, in particular, the presence of labels in all the places entering it is required to trigger transitions. This variant of the neural network with the help of Petri nets is not the only one, and in particular, it is also possible to use separate places and marks for modeling the weight coefficients, which will make it possible to train the neural network without modifying its structure. The further direction of the research is the implementation of the procedure for training a neural network with asynchrony.

**Keywords:** neural networks, Petri nets, artificial intelligence, transformation, activation function, keras, regularization, training

## REFERENCES

1. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, Lake Tahoe, Nevada, 2012, pp. 1097–1105.
2. Graves A., Mohamed A., Hinton G. Speech recognition with deep recurrent neural networks. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2013*, Vancouver, Canada, 2013, pp. 6645–6649.

---

\* Received 13 December 2017.

3. Karpathy A., Fei-Fei L. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, vol. 39, pp. 664–676.
4. Haykin S. *Neural networks and learning machines*. 3<sup>rd</sup> ed. New York, Prentice Hall/Pearson, 2009. 938 p.
5. Goodfellow I., Bengio Y., Courville A. *Deep learning*. Cambridge, MIT Press, 2016. 800 p.
6. Voevoda A.A., Romannikov D.O. Sintez neironnoi seti dlya resheniya logiko-arifmeticheskikh zadach [Synthesis of neural network for solving logical-arithmetic problems]. *Trudy SPIIRAN – SPIIRAS proceedings*, 2017, vol. 54, no. 5, pp. 205–223.
7. Haykin S., Deng C. Classification of radar clutter using neural networks. *IEEE Transactions on Neural Networks*, 1991, vol. 2, pp. 589–600.
8. Hastie T., Tibshirani R., Friedman J. *The elements of statistical learning: data mining, inference, and prediction*. New York, Springer, 2001.
9. Hagan M., Demuth H., Jesús O. A neural network predictive control system for paper mill wastewater treatment. *Engineering Applications of Artificial Intelligence*, 2003, vol. 16 (2), pp. 121–129.
10. Touretzky D.S., Pomerleau D.A. What is hidden in the hidden layers?. *Byte*, 1989, vol. 14, pp. 227–233.
11. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*, 2015, vol. 521, no. 7553, pp. 436–444.
12. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014, vol. 15, pp. 1929–1958.
13. Mahsereci M., Balles L., Lassner C., Hennig P. Early stopping without a validation set. *ArXiv.org*, 2017, arXiv:1703.09580.
14. Prechelt L. Early Stopping – but when? *Neural networks: tricks of the trade*. 2<sup>nd</sup> ed. Berlin, Heidelberg, Springer, 2012, pp. 53–67.
15. Voevoda A.A., Romannikov D.O. Asinkhronnyi algoritm sortirovki massiva chisel s ispol'zovaniem ingibitornykh setei Petri [Asynchronous sorting algorithm for array of numbers with the use of inhibitory Petri nets]. *Trudy SPIIRAN – SPIIRAS Proceedings*, 2014, iss. 3 (34), pp. 218–232.
16. Voevoda A.A., Polubinsky V.L., Romannikov D.O. Sortirovka massiva tsel'nykh chisel s ispol'zovaniem neironnoi seti [Sorting the array of integers using a neural network]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2016, no. 2 (63), pp. 151–157.

17. Voevoda A.A., Romannikov D.O. A binary array asynchronous sorting algorithm with using Petri nets. *Journal of Physics: Conference Series*, 2017, vol. 803, no. 1, p. 012178.

Для цитирования:

Воевода А.А., Романников Д.О. Трансляция нейронной сети в сеть Петри // Сборник научных трудов НГТУ. – 2018. – № 2 (92). – С. 101–107. – doi: 10.17212/2307-6879-2018-2-101-107.

For citation:

Voevoda A.A. Romannikov D.O. Translyatsiya neironnoi seti v set' Petri [Translation of the neural network in the Petry network]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 2 (92), pp. 101–107. doi: 10.17212/2307-6879-2018-2-101-107.

## ПРИМЕР РЕШЕНИЯ МИНИМАКСНОЙ ЗАДАЧИ ПРЕСЛЕДОВАНИЯ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ\*

Д.О. РОМАННИКОВ

*630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, кандидат технических наук, доцент кафедры автоматики.  
E-mail: dmitry.romannikov@gmail.com*

При разработке систем, основанных на использовании нейронных сетей, часто возникают проблемы с выбором конкретной модели нейронной сети. В статье приводится реализация системы для задачи максимизации времени «убегания» управляемого объекта от «догоняющего» объекта в ограниченном поле с препятствиями. Используется метод, заключающийся в построении конечного автомата на основании декомпозиции исходной задачи, в котором каждому состоянию соответствует своя нейронная сеть, а смена состояний выполняется на основании определения событий для их смены. Согласно вышеприведенному методу выделены три состояния: 1) начальное состояние, в котором на основании информации с поля координат определяются «безопасные» координаты на поле (т. е. координаты такой позиции, куда убегающий объект может переместиться быстрее догоняющего объекта и в дальнейшем перемещаться вокруг какого-либо препятствия в поле) для последующего перемещения; 2) состояние, в котором выполняется перемещение от начальных координат в заданную на поле позицию; 3) состояние, в котором выполняется перемещение вокруг препятствия в поле. В статье реализованы три нейронные сети, соответствующие каждому из вышеприведенных состояний. В силу того, что каждая из декомпозированных задач сама по себе является достаточно простой, то для их решения были использованы сети прямого распространения с одним скрытым слоем. В итоге нейронные сети состояли из 100 нейронов во входном слое, 70 нейронов в скрытом, двух – в выходном для определения безопасных координат и четырех – в выходном слое для формирования управляющих сигналов. Обучение выполнялось при помощи метода Adam с коэффициентом обучения 0.001 на сгенерированных для каждого состояния обучающих примерах.

**Ключевые слова:** нейронные сети, конечный автомат, искусственный интеллект, преобразования, функция активации, keras, регуляризация, обучение

---

\* Статья получена 21 декабря 2017 г.

## **ВВЕДЕНИЕ**

В настоящее время нейронные сети [1–8] приобрели высокую популярность из-за возможности их обучения для решения задач, решение которых путем построения алгоритмов является затруднительным. К таким задачам можно отнести рассматриваемую в работе [9] задачу «убегания» объекта от «охотника», которую предлагается решать при помощи импульсных нейронных сетей, основанных на модели «ключ–порог». В статье предлагается решение вышеприведенной задачи при помощи метода построения нейронной сети, основанного на декомпозиции исходной задачи и построении более простых в обучении нейронных сетей для решения каждой из подзадач [10–19].

## **ПОСТАНОВКА И РЕШЕНИЕ ЗАДАЧИ**

В решаемой задаче в поле, которое представляется матрицей  $10 \times 10$  элементов, присутствует объект, задача которого догнать убегающий от него второй объект. Первый будем называть «охотником». В статье рассматривается построение нейронной сети для управления убегающим объектом, для которого доступны четыре команды перемещения: влево/вверх/вправо/вниз. На каждом такте оба объекта перемещаются синхронно. Объект охотника будет двигаться оптимально, т. е. по пути наименьшего расстояния до убегающего объекта в графе поля. Фактом того, что объект охотника догнал убегающий объект, будем считать совпадение координат их после совершения перемещений обоими объектами.

Согласно [10], где предлагается использовать способ разбивания исходной задачи на группу подзадач с последующей реализацией нейронной сети для каждой из подзадач, представим поставленную задачу в виде конечного автомата со следующими состояниями: 1) исходное состояние, в котором убегающий объект определяет позицию на поле, в которую он считает необходимым переместиться для предотвращения встречи с объектом «охотник»; 2) состояние перемещения, в котором убегающий объект передвигается в определенную в состоянии 1 позицию; 3) состояние, в котором убегающий объект перемещается вокруг объекта препятствия.

Рассмотрим построение нейронных сетей для вышеприведенных состояний. Результатом исполнения нейронной сети является позиция, рядом с которой убегающий объект может убежать от объекта «охотник». Архитектура этой сети представлена на рис. 1. Она является сетью прямого распространения с одним скрытым слоем. Для обучения данной нейронной сети использовались сгенерированные примеры, в которых выбирались координаты таких позиций, рядом с которыми находились укрытия и можно было

скрыться за ними от объекта «охотник». Обучение выполнялось на 1000 примерах по 128 перемешанных примеров в одной выборке для обучения (*batch*) в течение 10 000 эпох. Валидация выполнялась на 50 примерах после каждой эпохи.

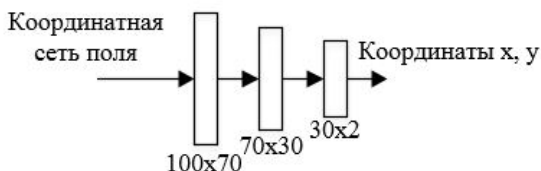


Рис. 1. Архитектура нейронной сети для определения координат для последующего перемещения

Структура нейронной сети для выполнения перемещения из начальной позиции в позицию, полученную в результате исполнения нейронной сети на предыдущем этапе, представлена на рис. 2. Данная нейронная сеть также представляет собой сеть прямого распространения с одним скрытым слоем. Обучение выполнялось на 5000 сгенерированных примерах, основанных на поиске перемещения по кратчайшему пути в графе, в течение 100 эпох.

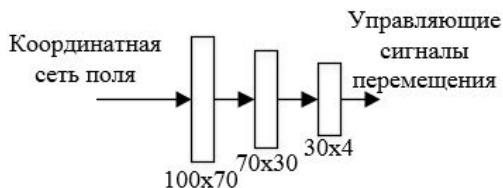


Рис. 2. Архитектура нейронной сети для определения управляющих сигналов для перемещения в состояниях 2 и 3

Структура нейронной сети для реализации третьего состояния совпадает со структурой нейронной сети на рис. 2. Данная нейронная сеть, как и предыдущие две сети, представляет собой сеть прямого распространения с одним скрытым слоем. Ее обучение выполнялось на 5000 сгенерированных примерах в течение 100 эпох.

В качестве метода обучения был выбран *Adam* с коэффициентом обучения 0.001. Также для определения позиции для перемещения (нейронная сеть, соответствующая состоянию 1) использовалась целевая функция (*loss function*)

ошибки среднего квадратического отклонения, для остальных нейронных сетей – функция перекрестной энтропии (*cross entropy*).

## ЗАКЛЮЧЕНИЕ

В статье рассматривается пример реализации системы, цель которой максимизировать время убегания управляемого объекта от догоняющего объекта в ограниченном поле. Используется предложенный в работе [10] метод, заключающийся в реализации нейронной сети для каждой части декомпозированной задачи. Согласно вышеприведенному методу выделены три состояния: 1) начальное состояние, в котором на основании информации с поля координат определяются «безопасные» координаты на поле (т. е. координаты такой позиции, куда убегающий объект может переместиться быстрее догоняющего объекта и в дальнейшем перемещаться вокруг какого-либо препятствия в поле) для последующего перемещения; 2) состояние, в котором выполняется перемещение от начальных координат в заданную на поле позицию; 3) состояние, в котором выполняется перемещение вокруг препятствия в поле.

В статье приведены реализации нейронных сетей для каждого из состояний. В силу того, что каждая из декомпозированных задач сама по себе является достаточно простой, то для их решения достаточно использования нейронной сети прямого распространения с одним скрытым слоем.

## СПИСОК ЛИТЕРАТУРЫ

1. *Bishop C.* Pattern recognition and machine learning. – New York: Springer, 2007. – 738 p. – (Information science and statistics).
2. *Richert W., Coelho L.* Building machine learning systems with Python. – Birmingham: Packt Publ., 2013. – 290 p.
3. *Hastie T., Tibshirani R., Friedman J.* The elements of statistical learning: data mining, inference, and prediction. – 2<sup>nd</sup> ed. – New York: Springer, 2013. – 745 p. – (Springer series in statistics).
4. *Lantz B.* Machine learning with R. – Birmingham: Packt Publ., 2013. – 396 p.
5. *Mohri M., Rostamizadeh A., Talwalkar A.* Foundations of machine learning. – Cambridge, MA: MIT Press, 2012. – 432 p. – (Adaptive computation and machine learning series).
6. *Conway D., White J.M.* Machine learning for hackers. – Sebastopol, CA: O'Reilly, 2012. – 324 p.



7. Welcome to the Deep Learning tutorial [Electronic resource]. – URL: <http://deeplearning.stanford.edu/tutorial/> (accessed: 26.06.2018).
8. *Haykin S.* Neural networks: a comprehensive foundation. – New York: MacMillan Publ., 1994. – 1104 p.
9. *Малявко А.А., Гаерилов А.В.* Импульсная нейронная сеть на основе модели «ключ–порог» // Интеллектуальный анализ сигналов, данных и знаний: методы и средства: сборник статей Всероссийской научно-практической конференции с международным участием, Новосибирск, 14–17 октября 2017 г. – Новосибирск: Изд-во НГТУ, 2017. – С. 325–330.
10. *Воевода А.А., Романников Д.О.* Нейронные сети в системах с последовательно выполняемыми действиями // Интеллектуальный анализ сигналов, данных и знаний: методы и средства: сборник статей Всероссийской научно-практической конференции с международным участием, Новосибирск, 14–17 октября 2017 г. – Новосибирск: Изд-во НГТУ, 2017. – С. 306–310.
11. *Трошина Г.В., Воевода А.А.* Параметрическая идентификация многоканального объекта на основе итерационного метода наименьших квадратов // 21 Международная конференция по мягким вычислениям и измерениям (SCM-2018): сборник докладов, Санкт-Петербург, 23–25 мая 2018 г.: в 2 т. – СПб.: СПбГЭТУ ЛЭТИ, 2018. – Т. 1. – С. 595–598.
12. *Troshina G.V., Voevoda A.A.* The iterative procedure modeling for the dynamic parameters estimation at the active identification task // Siberian symposium on data science and engineering (2017 SSDSE): proceedings, Akademgorodok, Novosibirsk, 12–13 April 2017. – Novosibirsk: IEEE, 2017. – P. 80–83.
13. *Воевода А.А., Романников Д.О.* Синтез нейронной сети на основе сети Петри для задач движения и стабилизации строя группы беспилотных летательных аппаратов // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2018. – № 2. – С. 26–33.
14. *Goodfellow I., Bengio Y., Courville A.* Deep learning. – Cambridge, MA: MIT Press, 2016.
15. *Воевода А.А., Романников Д.О.* Асинхронный алгоритм сортировки массива чисел с использованием ингибиторных сетей Петри // Труды СПИИРАН. – 2016. – Вып. 48. – С. 198–213.
16. *Воевода А.А., Шоба Е.В.* О разрешимости задачи автономизации многоканальной системы // Сборник научных трудов НГТУ. – 2010. – № 2 (60). – С. 9–16.
17. *Воевода А.А., Марков А.В., Романников Д.О.* Разработка программного обеспечения: проектирование с использованием UML диаграмм и сетей Петри на примере АСУ ТП водонапорной станции // Труды СПИИРАН. – 2014. – Вып. 3 (34). – С. 218–231.

18. Марков А.В., Воевода А.А. Развитие системы «Перемещение манипулятора в пространстве с препятствиями» при помощи рекурсивных функций // Автоматика и программная инженерия. – 2013. – № 2 (4). – С. 35–41.

19. Воевода А.А., Шоба Е.В. Управление перевернутым маятником // Сборник научных трудов НГТУ. – 2012. – № 2 (68). – С. 3–14.

**Романников Дмитрий Олегович**, кандидат технических наук, доцент кафедры автоматки Новосибирского государственного технического университета. Основное направление научных исследований – нейронные сети, сети Петри. Имеет более 50 публикаций. E-mail: dmitry.romannikov@gmail.com

DOI: 10.17212/2307-6879-2018-2-108-116

## **An example of solving a minimax pursuit problem using neural networks\***

**D.O. Romannikov**

*Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, candidate of Technical Sciences, associate professor of the automation department. E-mail: dmitry.romannikov@gmail.com*

In the development of systems based on the use of neural networks, there are often problems with the choice of a particular neural network model. The article presents the implementation of the system for the task of maximizing the "runaway" time of a controlled object from a "catch-up" object in a restricted field with obstacles. We use a method consisting in constructing a finite automaton on the basis of decomposition of the initial problem, in which each state corresponds to its own neural network, and state changes are performed based on the definition of events for their replacement. According to the above method, three states are distinguished: 1) the initial state in which, based on the information from the coordinate field, the "safe" coordinates on the field are defined (ie, the coordinates of the position where the evader can move faster than the catching object and then move around any obstacle in the field) for subsequent movement; 2) the state in which the displacement from the initial coordinates to the position set on the field is performed; 3) the state in which the movement around the obstacle in the field is performed. Three neural networks corresponding to each of the above states are realized in the article. Due to the fact that each of the decomposed tasks in itself is quite simple, then for the solution they used direct propagation networks with one hidden lay-

---

\* Received 21 December 2017.

er. As a result, neural networks consisted of 100 neurons in the input layer, 70 neurons in the concealed and 2 in the output to determine the safe coordinates, and 4 in the output to generate the control signals. The training was carried out using the Adam method with a training coefficient of 0.001 on the training examples generated for each state.

**Keywords:** neural networks, finite state machine, artificial intelligence, transformation, activation function, keras, regularization, training

## REFERENCES

1. Bishop C. *Pattern recognition and machine learning. Information science and statistics*. New York, Springer, 2007. 738 p.
2. Richert W., Coelho L. *Building machine learning systems with Python*. Birmingham, Packt Publ., 2013. 290 p.
3. Hastie T., Tibshirani R., Friedman J. *The elements of statistical learning: data mining, inference, and prediction*. 2<sup>nd</sup> ed. *Springer series in statistics*. New York, Springer, 2013. 745 p.
4. Lantz B. *Machine learning with R*. Birmingham, Packt Publ., 2013. 396 p.
5. Mohri M., Rostamizadeh A., Talwalkar A. *Foundations of machine learning. Adaptive computation and machine learning series*. Cambridge, MA, MIT Press, 2012. 432 p.
6. Conway D., White J.M. *Machine learning for hackers*. Sebastopol, CA, O'Reilly, 2012. 324 p.
7. *Welcome to the Deep Learning tutorial*. Available at: <http://deeplearning.stanford.edu/tutorial/> (accessed 26.06.2018).
8. Haykin S. *Neural networks: a comprehensive foundation*. New York, MacMillan Publ., 1994. 1104 p.
9. Malyavko A.A., Gavrilov A.V. [Spiking neural network based on “key-threshold” model]. *Intellektual'nyi analiz signalov, dannykh i znanii: metody i sredstva* [Intellectual analysis of signals, data and knowledge: methods and means], Novosibirsk, 14–17 October 2017, pp. 325–330. (In Russian).
10. Voevoda A.A., Romannikov D.O. [Neural networks in systems with sequential actions]. *Intellektual'nyi analiz signalov, dannykh i znanii: metody i sredstva* [Intellectual analysis of signals, data and knowledge: methods and means], Novosibirsk, 14–17 October 2017, pp. 306–310. (In Russian).
11. Troshina G.V., Voevoda A.A. [Parametric identification of a multi-channel object based on the iterative least-squares method]. *21 Mezhdunarodnaya konferentsiya po myagkim vychisleniyam i izmereniyam (SCM-2018)* [21 International

Conference on Soft Computing and Measurements (SCM-2018)], St. Petersburg, 23–25 May 2018, pp. 595–598. (In Russian).

12. Troshina G.V., Voevoda A.A. The iterative procedure modeling for the dynamic parameters estimation at the active identification task. *Siberian symposium on data science and engineering (2017 SSDSE): proceedings*, Akademgorodok, Novosibirsk, 12–13 April 2017, pp. 80–83.

13. Voevoda A.A., Romannikov D.O. Sintez neironnoi seti na osnove seti Petri dlya zadach dvizheniya i stabilizatsii stroya gruppy bespilotnykh letatel'nykh apparatov [Synthesis of a neural network based on the petri network for the tasks of movement and stabilization of the formation of a group of unmanned aerial vehicles]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika – Vestnik of Astrakhan State Technical University. Series: Management, Computer Science and Informatics*, 2018, no. 2, pp. 26–33.

14. Goodfellow I., Bengio Y., Courville A. *Deep learning*. Cambridge, MA, MIT Press, 2016.

15. Voevoda A.A., Romannikov D.O. Asinkhronnyi algoritm sortirovki massiva chisel s ispol'zovaniem ingibitornykh setei Petri [Asynchronous sorting algorithm for array of numbers with the use of inhibitory Petri nets]. *Trudy SPIIRAN – SPIIRAS Proceedings*, 2014, iss. 3 (34), pp. 218–232.

16. Voevoda A.A., Shoba E.V. O razreshimosti zadachi avtonomizatsii mnogokanal'noi sistemy [About diagonally decoupling for multi-input multi-output systems]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2010, no. 2 (60), pp. 9–16.

17. Voevoda A.A., Markov A.V., Romannikov D.O. Razrabotka programmno-go obespecheniya: proektirovanie s ispol'zovaniem UML diagramm i setei Petri na primere ASU TP vodonapornoj stantsii [Software development: software design using UML diagrams and Petri nets for example automated process control system of pumping station]. *Trudy SPIIRAN – SPIIRAS proceedings*, 2014, iss. 3 (34), pp. 218–231.

18. Markov A.V., Voevoda A.A. Razvitie sistemy "peremeshchenie manipulyatora v prostranstve s prepyatstviyami" pri pomoshchi rekursivnykh funktsii [Development of the system "moving the manipulator in the obstacle space" with the help of recursive functions]. *Avtomatika i programmaya inzheneriya – Automatics & Software Enginery*, 2013, no. 2 (4), pp. 35–41.

19. Voevoda A.A., Shoba E.V. Upravlenie perevernutym mayatnikom [Management of the inverted pendulum]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2012, no. 2 (68), pp. 3–14.

Для цитирования:

Романников Д.О. Пример решения минимаксной задачи преследования с использованием нейронных сетей // Сборник научных трудов НГТУ. – 2018. – № 2 (92). – С. 108–116. – doi: 10.17212/2307-6879-2018-2-108-116.

For citation:

Romannikov D.O. Primer resheniya minimaksnoi zadachi presledovaniya s ispol'zovaniem neuronnykh setei [An example of solving a minimax pursuit problem using neural networks]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 2 (92), pp. 108–116. doi: 10.17212/2307-6879-2018-2-108-116.

## О ПРОБЛЕМАХ СИНТЕЗА НЕЙРОННЫХ СЕТЕЙ\*

Д.О. РОМАННИКОВ<sup>1</sup>, Д.М. ИБРАЕВА<sup>2</sup>

<sup>1</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, кандидат технических наук, доцент кафедры автоматизации. E-mail: dmitry.romannikov@gmail.com

<sup>2</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, магистрант кафедры систем сбора и обработки данных. E-mail: ibraeva-dishik@yandex.ru

В статье приводятся рассуждения о тех проблемах, которые возникают у исследователей при разработке систем с использованием нейронных сетей. Наибольшее внимание уделено проблемам преобразования известных алгоритмов из классического описания в нейронные сети. К таким проблемам можно отнести следующие: 1) необходимость разработки инструмента для выполнения таких математических операций, как сложение/вычитание, умножение/деление и другие, а также на необходимости разработки инструмента для выполнения манипуляции над данными, к которым можно отнести такие, как перемещение, копирование и другие; 2) необходимость разработки конструкций для управления при вычислении (т. е. отсутствие операторов *if*, *for*, *while* и других); 3) отсутствие привычных структур данных, таких как массив, стек, очередь и другие. Такие проблемы могут быть продемонстрированы на примере поиска наименьшего пути в графе. Необходимо релаксировать ребра графа, для этого используются операции сравнения и сложения. Однако для выполнения данных операций в нейронных сетях их нужно предварительно реализовать. Стоит отметить, что решение задач такого вида не характерно для нейронных сетей, но может являться частью задачи реализации создания инструмента для преобразования классических алгоритмов в нейронную сеть с целью как их совместного использования, так и объяснения работы нейронной сети. Также предложен подход к решению задачи реализации нейронной системы с помощью разделения общей задачи на множество более мелких решений, которые существенно проще. Приведен пример такого разбиения исходной задачи.

**Ключевые слова:** нейронные сети, сети Петри, искусственный интеллект, преобразование, функция активации, keras, регуляризация, обучение

---

\* Статья получена 30 января 2018 г.

## ВВЕДЕНИЕ

Решение множества задач при помощи нейронных сетей достаточно сильно повысило популярность этого инструмента в последнее время [1–8]. К таким задачам можно отнести в первую очередь распознавание образов [1–6] и речи [1–4], а также использование техники *reinforcement learning*, позволяющей автоматически обучать нейронную сеть. Также одной из причин их распространения является то, что с их помощью можно решать задачи, алгоритмическое решение которых на данный момент времени не известно или решения с использованием нейронных сетей дают более лучшие результаты. Другой существенной причиной является то, что нейронные сети по своей сути используют распределенную обработку входной информации, что выгодно их отличает от классических подходов к разработке и от обработки информации. И третья причина – это потенциальная возможность перехода между классическим алгоритмическим представлением решения задачи к нейронному и обратно. Решение последней задачи может привести к тому, что появится возможность получать объяснение поведения нейронной сети, а также переводить известные алгоритмы к нейронному виду.

Далее в статье рассматриваются некоторые проблемы синтеза нейронных систем, с которыми сталкиваются исследователи и пути их решения.

## ПРОБЛЕМЫ СИНТЕЗА

Задачу синтеза нейронной сети будем рассматривать на примере создания системы управления для игры «Змейка» (игра, в которой «змейка» передвигается по ограниченной площадке с целью сбора требуемых артефактов (далее целей) и уклонения от нежелательных объектов, а также границ площадки (далее препятствий)). На сегодняшний день распространенным методом решения данной задачи является использование техники *reinforcement learning* [1, 8] (техника, основанная на обучении нейронной сети на основе автоматического подбора данных для обучения). С другой стороны, данную задачу легко решить при помощи классических алгоритмов, тогда становится потенциально возможным решение, в котором классический алгоритм будет реализован при помощи нейронной сети. В статье будет рассмотрен именно этот подход к синтезу нейронной сети.

Решение задачи перемещения змейки к цели может быть легко выполнено при помощи представления площадки как графа и поиска кратчайшего пути при помощи алгоритма Дейкстры, или поиска в ширину (из-за того, что в данном случае все веса в графе одинаковы).

Реализация функции *XOR* [1–4] является классической задачей машинного обучения. Кроме того, другие логические функции, такие как *AND* и *OR*, также могут быть реализованы [5–19]. Реализация более сложных алгоритмов при помощи нейронных сетей может быть проблематичной по следующим причинам:

1) необходимость разработки инструмента для выполнения таких математических операций, как сложение/вычитание, умножение/деление и другие, а также необходимость разработки инструмента для выполнения манипуляции над данными, к которым можно отнести такие, как перемещение, копирование и другие;

2) необходимость разработки конструкций для управления при вычислении (другими словами – отсутствие операторов *if*, *for*, *while* и других);

3) отсутствие привычных структур данных, таких как массив, стек, очередь и другие.

Рассмотрим эти недостатки более подробно. Первая и вторая погрешности – принципиальные, так как они являются основой при построении алгоритмов. Они могут быть продемонстрированы на примере того, что для поиска наименьшего пути в графе необходимо релаксировать ребра графа, для чего используются операции сравнения и сложения. Однако для выполнения данных операций в нейронных сетях их нужно предварительно реализовать. Третий недостаток является скорее следствием вышеприведенных. Сама структура нейронных сетей не предполагает, что в ней сигнал может быть отправлен в какую-то часть сети, однако этого можно избежать путем приведения части сигнала к нулевому. Циклы в нейронных сетях отсутствуют и «заменены» связями между нейронами.

В некотором смысле нейронные сети можно сравнивать с сетями Петри (и с программами в обычном своем смысле) по своему строению. В программах для передачи информации между различными участками программы используются различные преобразования над данными (в частности, в сетях Петри используются операторы на переходах между местами); с другой стороны, нейронные сети можно рассматривать как последовательность слоев, в которых каждый слой – оператор преобразования входных данных в выходные, где в последнем слое данные представляются в понятном человеку виде, а в промежуточных слоях данные представлены в каком-то виде, полученном при обучении сети. При этом количество слоев можно трактовать как количество операций по преобразованию входных данных.

С другой стороны, такая трактовка показывает, что одну задачу, которую нейронная сеть не в состоянии обучать (к такой задаче, например, можно отнести задачу поиска цели змейкой и передвижение по периметру площадки), можно разбить на множество мелких подзадач, для каждой из них можно



составить нейронную сеть, обучить ее и далее состыковать с целью получения решения всей исходно поставленной задачи.

В частности, вышеприведенную задачу можно разбить на следующие подзадачи: 1) определение наличия цели на площадке; 2) движение к цели и 3) передвижение по периметру площадки. Каждая из полученных задач может менять общее состояние системы, которое составляется из выходных значений сетей. В целом можно провести аналогию с построением программной системы из микросервисов.

## ЗАКЛЮЧЕНИЕ

В статье приведен список проблем, которые на данный момент не позволяют выполнять переход от классического представления алгоритмов к их реализации в нейронных сетях. К таким проблемам можно отнести необходимость разработки средств для выполнения вычислений (сложение/вычитание, умножение/деление), классических структур данных; необходимость разработки конструкций для управления при вычислении (*if, for, while* и др.).

Также предложен подход к решению задачи реализации нейронной системы с помощью разделения общей задачи на множество более мелких. Приведен пример такого разбиения исходной задачи.

## СПИСОК ЛИТЕРАТУРЫ

1. Воевода А.А., Романников Д.О. Тезисы о нейронных сетях // Сборник научных трудов НГТУ. – 2017. – № 1 (87). – С. 98–108.
2. Bishop C. Pattern recognition and machine learning. – New York: Springer, 2007. – 738 p. – (Information science and statistics).
3. Richert W., Coelho L. Building machine learning systems with Python. – Birmingham: Packt Publ., 2013. – 290 p.
4. Hastie T., Tibshirani R., Friedman J. The elements of statistical learning: data mining, inference, and prediction. – 2<sup>nd</sup> ed. – New York: Springer, 2013. – 745 p. – (Springer series in statistics).
5. Lantz B. Machine learning with R. – Birmingham: Packt Publ., 2013. – 396 p.
6. Mohri M., Rostamizadeh A., Talwalkar A. Foundations of machine learning. – Cambridge, MA: MIT Press, 2012. – 432 p. – (Adaptive computation and machine learning series).
7. Conway D., White J.M. Machine learning for hackers. – Sebastopol, CA: O'Reilly, 2012. – 324 p.

8. Welcome to the Deep Learning tutorial [Electronic resource]. – URL: <http://deeplearning.stanford.edu/tutorial/> (accessed: 26.06.2018).

9. *Haykin S.* Neural networks: a comprehensive foundation. – New York: MacMillan Publ., 1994. – 1104 p.

10. *Воевода А.А., Романников Д.О.* Нейронные сети в системах с последовательно выполняемыми действиями // Интеллектуальный анализ сигналов, данных и знаний: методы и средства: сборник статей Всероссийской научно-практической конференции с международным участием, Новосибирск, 14–17 октября 2017 г. – Новосибирск: Изд-во НГТУ, 2017. – С. 306–310.

11. *Трошина Г.В., Воевода А.А.* Параметрическая идентификация многоканального объекта на основе итерационного метода наименьших квадратов // 21 Международная конференция по мягким вычислениям и измерениям (SCM-2018): сборник докладов, Санкт-Петербург, 23–25 мая 2018 г.: в 2 т. – СПб.: СПбГЭТУ ЛЭТИ, 2018. – Т. 1. – С. 595–598.

12. *Troshina G.V., Voevoda A.A.* The iterative procedure modeling for the dynamic parameters estimation at the active identification task // Siberian symposium on data science and engineering (2017 SSDSE): proceedings, Akademgorodok, Novosibirsk, 12–13 April 2017. – Novosibirsk: IEEE, 2017. – P. 80–83.

13. *Воевода А.А., Романников Д.О.* Синтез нейронной сети на основе сети Петри для задач движения и стабилизации строя группы беспилотных летательных аппаратов // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2018. – № 2. – С. 26–33.

14. *Goodfellow I., Bengio Y., Courville A.* Deep learning. – Cambridge, MA: MIT Press, 2016.

15. *Воевода А.А., Романников Д.О.* Асинхронный алгоритм сортировки массива чисел с использованием ингибиторных сетей Петри // Труды СПИИРАН. – 2016. – Вып. 48. – С. 198–213.

16. *Воевода А.А., Шоба Е.В.* О разрешимости задачи автономизации многоканальной системы // Сборник научных трудов НГТУ. – 2010. – № 2 (60). – С. 9–16.

17. *Воевода А.А., Марков А.В., Романников Д.О.* Разработка программного обеспечения: проектирование с использованием UML диаграмм и сетей Петри на примере АСУ ТП водонапорной станции // Труды СПИИРАН. – 2014. – Вып. 3 (34). – С. 218–231.

18. *Марков А.В., Воевода А.А.* Развитие системы «Перемещение манипулятора в пространстве с препятствиями» при помощи рекурсивных функций // Автоматика и программная инженерия. – 2013. – № 2 (4). – С. 35–41.

19. *Воевода А.А., Шоба Е.В.* Управление перевернутым маятником // Сборник научных трудов НГТУ. – 2012. – № 2 (68). – С. 3–14.

**Романников Дмитрий Олегович**, кандидат технических наук, доцент кафедры автоматки Новосибирского государственного технического университета. Основное направление научных исследований – нейронные сети, сети Петри. Имеет более 50 публикаций. E-mail: dmitry.romannikov@gmail.com

**Ибраева Динара Миннахатовна**, магистрант кафедры систем сбора и обработки данных Новосибирского государственного технического университета. E-mail: ibraeva-dishik@yandex.ru

DOI: 10.17212/2307-6879-2018-2-117-124

### **On the problems of synthesis of neural networks \***

**D.O. Romannikov<sup>1</sup>, D.M. Ibraeva<sup>2</sup>**

<sup>1</sup>*Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, candidate of Technical Sciences, associate professor of the automation department. E-mail: dmitry.romannikov@gmail.com*

<sup>2</sup>*Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, undergraduate of the data collection and processing systems department. E-mail: ibraeva-dishik@yandex.ru*

The article gives reasoning about the problems that arise in the analysis and use of technologies using neural networks. The greatest attention is paid to the problem of converting known algorithms from the classical description into neural networks. Such problems can be: 1) the need to develop a tool for performing such mathematical operations as addition/subtraction, multiplication/division and others, and also the development of a tool for manipulating data, such as moving, copying, and others, is required; 2) the need to develop structures for management in the calculation (ie, the absence of if, for, while, and others); 3) lack of familiar data structures, such as array, stack, queue and others. Such problems can be demonstrated using the example of finding the smallest path in a graph, it is necessary to relax the edges of a graph, using comparisons and constraints. However, to perform these operations in neural networks, they must first be implemented. It should be noted that the solution of problems of this kind is not typical for neural networks, but it can be part of the realization part of the implementation of the intro-tool for the transformation of classical algorithms into a neural network for the purpose of their joint use, and the explanation of the neural network operation. Also, the proposed approach to solving the problems of implementing a neural system by means of divided functions into several smaller ones, which can be accessed with the help of neural networks. An example of such a partition of the original problem is given.

**Keywords:** neural networks, Petri nets, artificial intelligence, transformation, activation function, keras, regularization, training

---

\* Received 30 January 2018.

## REFERENCES

1. Voevoda A.A., Romannikov D.O. Tezisy o neironnykh setyakh [Theses about neural networks]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2017, no. 1 (87), pp. 98–108.
2. Bishop C. *Pattern recognition and machine learning. Information science and statistics*. New York, Springer, 2007. 738 p.
3. Richert W., Coelho L. *Building machine learning systems with Python*. Birmingham, Packt Publ., 2013. 290 p.
4. Hastie T., Tibshirani R., Friedman J. *The elements of statistical learning: data mining, inference, and prediction*. 2<sup>nd</sup> ed. *Springer series in statistics*. New York, Springer, 2013. 745 p.
5. Lantz B. *Machine learning with R*. Birmingham, Packt Publ., 2013. 396 p.
6. Mohri M., Rostamizadeh A., Talwalkar A. *Foundations of machine learning. Adaptive computation and machine learning series*. Cambridge, MA, MIT Press, 2012. 432 p.
7. Conway D., White J.M. *Machine learning for hackers*. Sebastopol, CA, O'Reilly, 2012. 324 p.
8. *Welcome to the Deep Learning tutorial*. Available at: <http://deeplearning.stanford.edu/tutorial/> (accessed 26.06.2018).
9. Haykin S. *Neural networks: a comprehensive foundation*. New York, MacMillan Publ., 1994. 1104 p.
10. Voevoda A.A., Romannikov D.O. [Neural networks in systems with sequential actions]. *Intellectual'nyi analiz signalov, dannykh i znanii: metody i sredstva* [Intellectual analysis of signals, data and knowledge: methods and means], Novosibirsk, 14–17 October 2017, pp. 306–310. (In Russian).
11. Troshina G.V., Voevoda A.A. [Parametric identification of a multi-channel object based on the iterative least-squares method]. *21 Mezhdunarodnaya konferentsiya po myagkim vychisleniyam i izmereniyam (SCM-2018)* [21 International Conference on Soft Computing and Measurements (SCM-2018)], St. Petersburg, 23–25 May 2018, pp. 595–598. (In Russian).
12. Troshina G.V., Voevoda A.A. The iterative procedure modeling for the dynamic parameters estimation at the active identification task. *Siberian symposium on data science and engineering (2017 SSDSE): proceedings*, Akademgorodok, Novosibirsk, 12–13 April 2017, pp. 80–83.
13. Voevoda A.A., Romannikov D.O. Sintez neironnoi seti na osnove seti Petri dlya zadach dvizheniya i stabilizatsii stroya gruppy bespilotnykh letatel'nykh apparatov [Synthesis of a neural network based on the petri network for the tasks of movement and stabilization of the formation of a group of unmanned aerial vehicles]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika – Vestnik of Astrakhan State*

*Technical University. Series: Management, Computer Science and Informatics*, 2018, no. 2, pp. 26–33.

14. Goodfellow I., Bengio Y., Courville A. *Deep learning*. Cambridge, MA, MIT Press, 2016.

15. Voevoda A.A., Romannikov D.O. Asinkhronnyi algoritm sortirovki massiva chisel s ispol'zovaniem ingibitornykh setei Petri [Asynchronous sorting algorithm for array of numbers with the use of inhibitory Petri nets]. *Trudy SPIIRAN – SPIIRAS Proceedings*, 2014, iss. 3 (34), pp. 218–232.

16. Voevoda A.A., Shoba E.V. O razreshimosti zadachi avtonomizatsii mnogokanal'noi sistemy [About diagonally decoupling for multi-input multi-output systems]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2010, no. 2 (60), pp. 9–16.

17. Voevoda A.A., Markov A.V., Romannikov D.O. Razrabotka programmno obespecheniya: proektirovanie s ispol'zovaniem UML diagramm i setei Petri na primere ASU TP vodonapornoi stantsii [Software development: software design using UML diagrams and Petri nets for example automated process control system of pumping station]. *Trudy SPIIRAN – SPIIRAS proceedings*, 2014, iss. 3 (34), pp. 218–231.

18. Markov A.V., Voevoda A.A. Razvitiye sistemy "peremeshchenie manipulyatora v prostranstve s prepyatstviyami" pri pomoshchi rekursivnykh funktsii [Development of the system "moving the manipulator in the obstacle space" with the help of recursive functions]. *Avtomatika i programmnaya inzheneriya – Automatics & Software Enginery*, 2013, no. 2 (4), pp. 35–41.

19. Voevoda A.A., Shoba E.V. Upravlenie perevernutym mayatnikom [Management of the inverted pendulum]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2012, no. 2 (68), pp. 3–14.

Для цитирования:

Романников Д.О., Ибраева Д.М. Рассуждения о проблемах синтеза нейронных сетей // Сборник научных трудов НГТУ. – 2018. – № 2 (92). – С. 117–124. – doi: 10.17212/2307-6879-2018-2-117-124.

For citation:

Romannikov D.O., Ibraeva D.M. Rassuzhdeniya o problemakh sinteza neironnykh setei [Discussions on the problems of synthesis of neural networks]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 2 (92), pp. 117–124. doi: 10.17212/2307-6879-2018-2-117-124.

## **ПРАВИЛА ДЛЯ АВТОРОВ**

### **УСЛОВИЯ ПРИЕМА СТАТЕЙ**

В редакцию **Сборника научных трудов НГТУ** представляются следующие материалы.

1. Статья объемом 8–12 страниц от аспирантов и 16–22 страницы от докторантов, печатная версия – два экземпляра, подписанных авторами (требования к оформлению см. на сайте: <http://sbornik.infoterra.ru>).

2. Контактная информация (телефоны, адреса электронной почты, место работы, должность, ученая степень, ученое звание автора) – печатная версия, два экземпляра.

3. Описание статьи для базы данных «Российский индекс научного цитирования (РИНЦ)», печатная версия – один экземпляр (<http://sbornik.infoterra.ru>).

4. Электронная версия статьи, контактной информации, описание статьи для базы данных РИНЦ, реферат на англ. языке (в отдельных файлах на CD).

5. Рецензия объемом одна-две страницы в двух экземплярах.

6. Экспертное заключение о возможности опубликования.

Информацию о сборнике, дополнительную информацию, документацию по оформлению см. на сайте: <http://sbornik.infoterra.ru>  
[journals.nstu.ru/sbornik](http://journals.nstu.ru/sbornik)

### **РЕКОМЕНДУЕМЫЕ РАЗДЕЛЫ СБОРНИКА**

Автоматическое управление (и идентификация)

Моделирование процессов и устройств

Обработка информации

Современные информационные технологии

Электроэнергетика, электромеханика и электротехнологии

Механика (газодинамика)

Физика (и математика)

Материаловедение

Сообщения и дискуссии

Педагогика

**Авторы, не являющиеся сотрудниками НГТУ, представляют сопроводительное письмо на имя проректора по научной работе НГТУ.**

Все рукописи рецензируются, по результатам рецензирования редколлегия принимает решение о целесообразности опубликования материалов.

Плата за публикацию рукописей не взимается.