

О нейросетевой реализации алгоритмов управления летательными аппаратами

А. П. Соколов

Данная работа посвящена разработке метода синтеза нейронной сети, реализующей заданный наперед вычислительный алгоритм. Рассматриваются алгоритмы применяемые в системах управления летательными аппаратами. Синтезируемая нейросеть должна быть в некотором смысле эквивалентной исходному алгоритму. Это означает, что поведение нейросети для всей системы управления в целом не должно отличаться от поведения исходной вычислительной процедуры. Предполагается, что исходный алгоритм (вычислительная процедура) реализован на некотором языке программирования, например C++.

1. Постановка задачи

В качестве исходных данных задан некий алгоритм \mathcal{A} .

Алгоритм \mathcal{A} задан в виде набора вычислительных процедур $P = \{p_1, p_2, \dots, p_n\}$, объединенных в единую вычислительную процедуру. Фактически на множестве P задано отношение предшествования L . Полагаем $p_i L p_j$, где $p_i, p_j \in P$, если результат вычисления p_i является одним из исходных данных для вычисления p_j . Отношение L может быть задано в виде матрицы:

$$L_{n \times n} = \begin{pmatrix} l_{11} & \dots & l_{1n} \\ \dots & & \dots \\ l_{n1} & \dots & l_{nn} \end{pmatrix}, \text{ где } l_{ij} = \begin{cases} 1, & \text{если } p_i L p_j \\ 0, & \text{иначе.} \end{cases} \quad (1)$$

Процедуры из множества P реализованы на C++.

Ставится следующая задача — на основании описания алгоритма \mathfrak{A} на C++ построить матрицу $L_{n \times n}$ и нейронную сеть, реализующую алгоритм \mathfrak{A} .

Данная задача решалась на примере алгоритма, моделирующего работу системы стабилизации летательного аппарата (ЛА) по углу тангажа.

2. Моделирующий алгоритм

Алгоритм, моделирующий работу системы стабилизации ЛА, представляет собой алгоритм численного решения методом Эйлера следующей системы дифференциальных уравнений, описывающих поведение системы стабилизации во времени:

$$\begin{cases} \frac{d}{dt} \begin{pmatrix} \dot{\vartheta} \\ \Omega \end{pmatrix} = \begin{pmatrix} \Omega \\ \frac{K \cdot T_1 \cdot \dot{\delta}_в}{T^2} + \frac{K \cdot \delta_в}{T^2} - \frac{2\xi \cdot \Omega}{T} - \frac{\dot{\vartheta}}{T^2} \end{pmatrix}; \\ \frac{d}{dt}(j_n) = \frac{V \cdot \dot{\vartheta}}{T_1} - \frac{j_n}{T_1}; \\ \frac{d}{dt}(\dot{\delta}_{рп}) = \frac{K_{рп} \cdot U_y}{T_{рп}^2} - \frac{2\xi_{рп} \cdot \dot{\delta}_{рп}}{T_{рп}} - \frac{\delta_{рп}}{T_{рп}^2}. \end{cases} \quad (2)$$

Блок-схема моделирующего алгоритма представлена на рис. 1.

3. Построение отношения \mathbf{L}

Каждой элементарной операции алгоритма поставим в соответствие некий словарный оператор и условия вхождения слов в него. Так, например, выражению $K = \frac{a_{13}a_{42}}{a_{12}+a_{11}a_{42}}$ соответствует некий оператор $A = f(a_{11}; a_{12}; a_{13}; a_{42})$ и следующие условия вхождения слов в него: $a_{11} \& a_{12} \& a_{13} \& a_{42} \rightarrow A$. Данное выражение означает, что в результате одновременного введения в оператор A величин $a_{11}; a_{12}; a_{13}; a_{42}$ оператор вычисляет величину K , то есть значению K всегда предшествуют значения $a_{11}; a_{12}; a_{13}; a_{42}$.

Ниже приведена сводная таблица, содержащая систему вычислительных процедур (алфавитных операторов) P , а также соответствующие им условия вхождения слов (1).

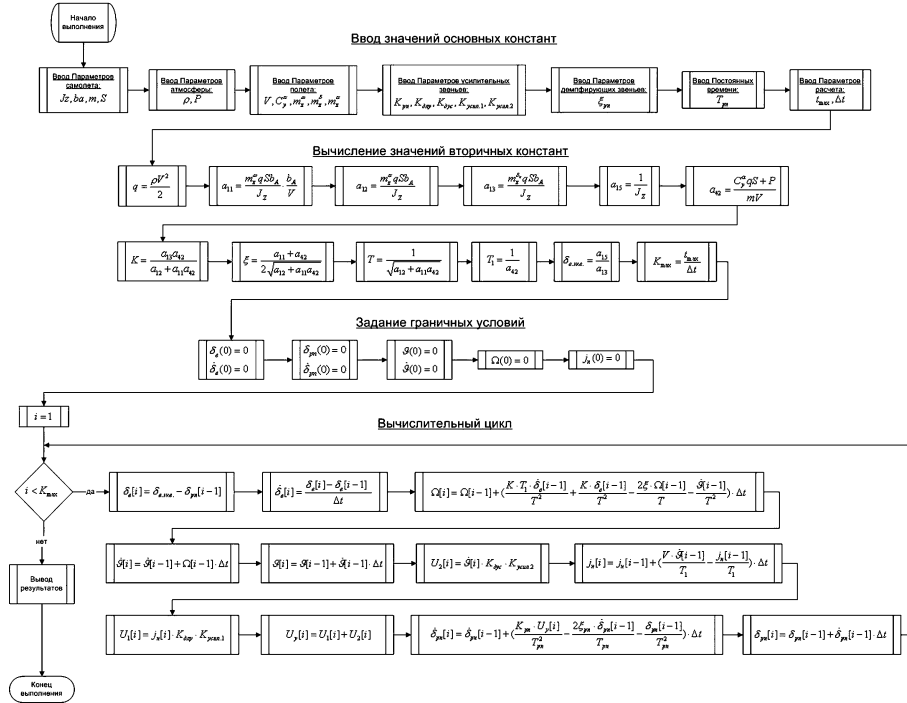


Рис. 1. Блок-схема моделирующего алгоритма.

Данная таблица фактически описывает отношение предшествования L , заданное на множестве P .

4. Нейросетевая реализация алгоритма

На основании полученного описания отношения L была построена сетевая модель алгоритма \mathcal{A} .

Сетевой моделью алгоритма \mathcal{A} , заданного множеством вычислительных процедур P , находящихся в отношении предшествования L , называется ориентированный граф, ребра которого соответствуют элементам из P . При этом конец произвольного ребра p_i является началом ребра p_j тогда и только тогда, когда p_i непосредственно предшествует p_j .

Обозначение оператора	Содержательный смысл оператора	Условия вхождения слов
X	Входной вектор общего алгоритма	-
a	$\delta_{e, \text{вх}}$	$X \rightarrow a$
b	V	$X \rightarrow b$
c	ξ	$X \rightarrow c$
d	T	$X \rightarrow d$
e	K	$X \rightarrow e$
f	T_1	$X \rightarrow f$
g	t_{max}	$X \rightarrow g$
h	Δt	$X \rightarrow h$
i	K_{pn}	$X \rightarrow i$
j	$K_{\text{дг}}$	$X \rightarrow j$
k	$K_{\text{дс}}$	$X \rightarrow k$
l	$K_{\text{учи},1}$	$X \rightarrow l$
m	$K_{\text{учи},2}$	$X \rightarrow m$
n	ξ_{pn}	$X \rightarrow n$
o	T_{pn}	$X \rightarrow o$
p	$\delta_i(i-1)$	$X \vee a' \rightarrow p$
q	$\delta_e(i-1)$	$X \vee a' \rightarrow q$
r	$\delta_{\text{pn}}(i-1)$	$X \vee a' \rightarrow r$
s	$\delta_{\text{дг}}(i-1)$	$X \vee a' \rightarrow s$
t	$\mathcal{A}(i-1)$	$X \vee a' \rightarrow t$
u	$\mathcal{B}(i-1)$	$X \vee a' \rightarrow u$
v	$\Omega(i-1)$	$X \vee a' \rightarrow v$
w	$J_n(i-1)$	$X \vee a' \rightarrow w$
z	i	$X \vee b' \rightarrow z$

Таблица 1. Система вычислительных процедур P (начало).

Метод построения сетевой модели на основании описания отношения L описан в [1].

На основе полученной сетевой модели алгоритма \mathcal{A} была реализован метод построения нейронной сети, реализующей данный алгоритм. Описание данного метода также приводится в [1]. Фактически, данный метод заключается в последовательной реализации каждого из элементов множества P в виде элементарной нейросети. После этого ребра сетевой модели заменяются соответствующими нейросетями, образуя тем самым единую нейронную сеть, реализующую алгоритм \mathcal{A} .

После построения нейронной сети, реализующей исходный алгоритм, ее функционирование было смоделировано в среде MATLAB.

<i>A</i>	$\delta_e[i] = f_1(\delta_{e, \max}; \delta_{pn}[i-1]; i)$	$a \& r \& z \rightarrow A$
<i>B</i>	$\delta_e[i] = f_2(\delta_e[i]; \delta_e[i-1]; \Delta t; i)$	$h \& p \& z \& A \rightarrow B$
<i>C</i>	$\Omega[i] = f_3(\Omega[i-1]; K; T; \delta_e[i-1]; \delta_e[i-1]; \xi; \theta[i-1]; \Delta t; i)$	$c \& d \& e \& h \& p \& q \& u \& v \& z \rightarrow C$
<i>D</i>	$\theta[i] = f_4(\theta[i-1]; \Omega[i-1]; \Delta t; i)$	$h \& u \& v \& z \rightarrow D$
<i>E</i>	$\theta[i] = f_5(\theta[i-1]; \theta[i-1]; \Delta t; i)$	$h \& t \& u \& z \rightarrow E$
<i>F</i>	$U_2[i] = f_6(\theta[i]; K_{\text{оп}}; K_{\text{учн.2}}; i)$	$k \& m \& z \& D \rightarrow F$
<i>G</i>	$j_n[i] = f_7(j_n[i-1]; V; \theta[i-1]; T; \Delta t; i)$	$b \& f \& h \& u \& w \& z \rightarrow G$
<i>H</i>	$U_1[i] = f_8(j_n[i]; K_{\text{оп}}; K_{\text{учн.1}}; i)$	$j \& l \& z \& G \rightarrow H$
<i>I</i>	$U_y[i] = f_9(U_1[i]; U_2[i]; i)$	$z \& F \& H \rightarrow I$
<i>J</i>	$\delta_{pn}[i] = f_{10}(\delta_{pn}[i-1]; K_{pn}; U_y[i]; T_{pn}; \xi_{pn}; \delta_{pn}[i-1]; \Delta t; i)$	$h \& i \& n \& o \& r \& s \& z \& I \rightarrow J$
<i>K</i>	$\delta_{pn}[i] = f_{11}(\delta_{pn}[i-1]; \delta_{pn}[i-1]; \Delta t; i)$	$h \& r \& s \& z \rightarrow K$
<i>L</i>	$K_{\max} = f_{12}(t_{\max}; \Delta t)$	$g \& h \rightarrow L$
<i>a'</i>	$\delta_e[i]; \delta_e[i]; \Omega[i]; \theta[i]; \theta[i]; U_2[i]; j_n[i]; U_1[i]; U_y[i]; \delta_{pn}[i]; \delta_{pn}[i]; i; K_{\max}$	$z \& A \& B \& C \& D \& E \& F \& G \& H \& I \& J \& K \& L \rightarrow a'$
<i>b'</i>	$i = f_{13}(i)$	$a' \rightarrow b'$
<i>Y</i>	Печать результатов	$a' \rightarrow Y$

Таблица 2. Система вычислительных процедур *P* (окончание).

Было показано, что поведение выхода нейронной сети практически идентично поведению выхода исходного вычислительного алгоритма.

Автор выражает признательность В.Б. Кудрявцеву за научное руководство.

Список литературы

[1] Ермилов Л. И. Синтез сетевых моделей сложных процессов и систем. М.: Министерство обороны СССР, 1970.

