

УДК [62.83:621.94]:004.32.26

УДК 621.865.8+63-83

М. П. Белов, И. С. Носиров, Ч. Х. Фьонг
Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Исследование системы управления электроприводом подачи токарного станка с применением метода обобщенного нейроруправления с прогнозированием

Приведено математическое описание алгоритма обобщенного нейроруправления с предсказанием (Neural Generalized Predictive Control, NGPC) и оптимизационный алгоритм Ньютона-Рафсона. Алгоритм Ньютона-Рафсона позволяет минимизировать функционал качества, определяющий входной сигнал, необходимый для получения желаемого поведения объекта. Рассматривается процедура идентификации, позволяющая построить нейронную сеть, которая моделирует управляемый объект. Рассмотрен синтез нейросетевого управления для обеспечения высокой статической и динамической точности при воспроизведении контурно-позиционных перемещений. Разработана модель системы управления электроприводом подачи токарного станка с использованием нейроконтроллера с предсказанием (Neural Network Predictive Control, NNPC) в среде Matlab/Simulink. Выполнен синтез нескольких вариантов NNPC с разными значениями для установления его оптимальных параметров. Определены параметры нейроконтроллера с предсказанием, существенно влияющие на качество прогнозирования.

Электропривод подачи токарного станка, алгоритм обобщенного нейроруправления, нейроконтроллер с предсказанием, алгоритм Ньютона-Рафсона

Постановка проблемы. Повышение качества и точности обработки деталей – это одна из наиболее распространенных задач в машиностроении. Процесс заготовки деталей на металлорежущих станках непосредственно связан с его точностью, которая характеризуется степенью влияния различных погрешностей станков – геометрических, кинематических, упругих, температурных и динамических. В современных станках с числовым программным управлением (ЧПУ) приводы подачи линейных движений формообразующих узлов на базе шариковых винтовых передач (ШВП) играют важнейшую роль, в большинстве случаев определяя размерную, а в отдельных случаях и геометрическую точность.

Электропривод подачи токарных станков с ЧПУ должен обеспечивать скорость быстрых перемещений; широкий диапазон регулирования привода в современных станках с ЧПУ; требуемую тяговую силу для перемещения рабочего органа при различной нагрузке на всех скоростях вплоть до самых малых; иметь высокую жесткость, что во многом определяет точность пере-

мещения и установки рабочего органа, его динамическое качество; не иметь люфтов в направляющих и передачах, а также контактных деформаций; для точных расчетных позиционирований в станках с ЧПУ и в зуборезных станках при сложных движениях формообразования иметь высокую кинематическую точность. Изучение указанных погрешностей обработки и требований к электроприводам подачи современных станков с ЧПУ служит актуальной теоретической и практической задачей.

В этой статье предлагается применение метода обобщенного нейроруправления с предсказанием (NGPC) и синтез нейроконтроллера (NNPC) для прогнозирования скорости перемещения исполнительного органа (ИО) электропривода подачи токарного станка.

Обобщенное нейроруправление с предсказанием. Методика алгоритма обобщенного управления с предсказанием (NGPCA) основана на использовании эталонной модели для прогнозирования будущей траектории объекта [1]. Далее NGPC-алгоритм вычисляет будущие управляю-

шие воздействия для минимизации функционала качества на каждый образец k . Затем процедура повторяется для образца $k + 1$. Структурная схема обобщенного нейроуправления с предсказанием алгоритма NGPCA представлена на рис. 1.

Алгоритм NGPC состоит из следующих элементов: управляемого объекта, эталонной модели, которая описывает желаемое количество объекта, моделирующий объект нейронной сети и алгоритм минимизации функционала качества (Cost Function Minimization, CFM), который определяет входной сигнал, необходимый для получения желаемого поведения объекта. NGPC-алгоритм состоит из блока CFM и блока искусственной нейронной сети. NGPC-алгоритм работает следующим образом. На вход эталонной модели подается сигнал $r(n)$. Далее эталонная модель генерирует эталонный сигнал $y_m(n)$, который служит входом для CFM-блока, вычисляющего сигнал, который служит входом для объекта.

Двухполюсный двухпозиционный переключатель SA может переключаться в положения 1 и 2. Переключатель устанавливается в положение 1, т. е. к объекту, используя алгоритм CFM для определения оптимального входного сигнала $u(n)$, минимизирует выбранный критерий качества управления. Между выборками переключатель устанавливает модели объекта, где алгоритм CFM использует эту модель для вычисления следующего управляющего входа, $u(n + 1)$, для предсказания реакции сигнала, полученного от объекта.

После минимизации функционала качества этот входной сигнал подается на вход объекта. Качество вычисления в основном зависит от выбора минимизирующего алгоритма CFM-блока. Выбор алгоритма минимизации основан на нескольких критериях – количестве итераций к ре-

шению, вычислительных затратах и точности расчета. В общем случае, эти методы требуют большого числа итераций, что затрудняет управление в режиме реального времени.

Точность предсказания тесно связана с качеством модели объекта. Существует довольно много методов для моделирования линейного объекта, а если объект нелинейный, то задача усложняется. В настоящее время существует два метода для моделирования нелинейных объектов: один из них заключается в линеаризации объекта на множества рабочих точек, а второй – в разработке нелинейной модели, которая зависит от предположений о динамике объекта. Если эти допущения неверны, то точность метода уменьшается. Таким образом, модели с использованием нейронной сети способны управлять нелинейными динамическими объектами.

Формулировка функционала качества NGPC. Как уже упоминалось, NGPC-алгоритм основан на минимизации функционала качества. Функционал качества имеет следующий вид:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} [y_m(t+j) - y_n(t+j)]^2 - \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2, \quad (1)$$

где N_1, N_2 – нижний и верхний пределы предсказания; N_u – диапазон управления; $y_m(t+j)$ – прогнозируемый результат от нейронной сети; $y_n(t+j)$ – эталонная траектория; $\Delta u(t+j-1)$ – управляемый вход; λ – весовой множитель. Функционал качества минимизирует не только среднеквадратичную ошибку эталонного сигнала, но также и взвешенную среднеквадратичную ско-

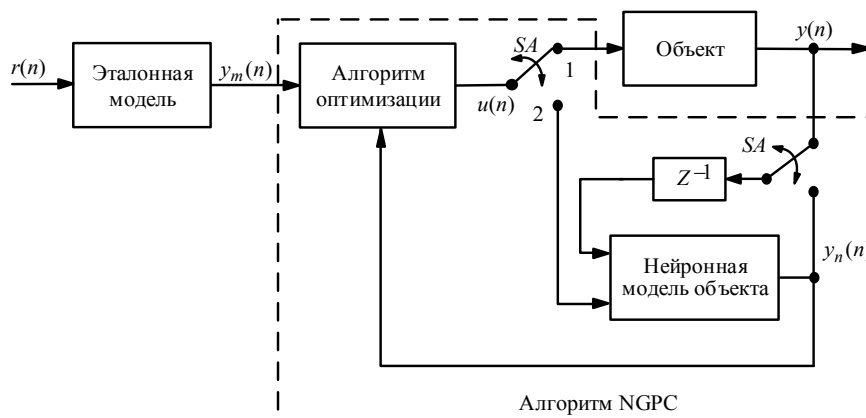


Рис. 1

рость управляющего сигнала. Управляющий входной сигнал генерируется, если функционал качества минимизирован. Функционал качества позволяет объекту наблюдать заданную траекторию в пределах некоторой точности и имеет четыре настраиваемых параметра: N_1, N_2, N_u и λ . Значение N_u является пределом диапазона управления. Единственное ограничение для значений N_1 и N_u – то, что эти пределы должны быть меньше или равны N_2 . Весовой множитель λ введен для балансирования между первым и вторым суммированиями.

Алгоритм минимизации функционала качества. Оптимальное управление зависит от алгоритма минимизации функционала качества CFM. В данной статье алгоритм Ньютона–Рафсона используется как метод оптимизации для функционала качества. Целью данного алгоритма служит минимизация функционала качества J (1) по вектору $\mathbf{U} = [u(n+1), u(n+2), u(n+N_u)]^T$. Итерационный процесс вычисляет значение \mathbf{J} , обозначенное как $J(k)$. Для каждой итерации $J(k)$ также генерируется промежуточный управляющий входной вектор и представляется в виде уравнения

$$\mathbf{U}(k) = \begin{bmatrix} u(t+1) \\ u(t+2) \\ \vdots \\ \vdots \\ u(t+N_u) \end{bmatrix}, \quad k = 1, \dots, N_u. \quad (2)$$

С использованием алгоритма Ньютона–Рафсона $\mathbf{U}(k+1)$ определяется следующим образом:

$$\mathbf{U}(k+1) = \mathbf{U}(k) - \left(\frac{\partial^2 \mathbf{J}}{\partial \mathbf{U}^2}(k) \right)^{-1} \frac{\partial \mathbf{J}}{\partial \mathbf{U}}(k). \quad (3)$$

Якобиан описывается уравнением

$$\frac{\partial \mathbf{J}}{\partial \mathbf{U}}(k) = \begin{bmatrix} \frac{\partial J}{\partial U(t+1)} \\ \vdots \\ \vdots \\ \frac{\partial J}{\partial U(t+N_u)} \end{bmatrix}, \quad (4)$$

гессиан – уравнением

$$\frac{\partial^2 \mathbf{J}}{\partial \mathbf{U}^2}(k) = \begin{bmatrix} \frac{\partial^2 J}{\partial u(t+1)^2} & \cdots & \frac{\partial^2 J}{\partial u(t+1)\partial u_k(t+N_u)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial u(t+N_u)\partial u_k(t+1)} & \cdots & \frac{\partial^2 J}{\partial u(t+N_u)^2} \end{bmatrix}. \quad (5)$$

Каждый элемент якобиана вычисляется с помощью частичного дифференцирования (5) относительно вектора \mathbf{U} :

$$\frac{\partial J}{u(t+h)} = 2 \sum_{j=N_1}^{N_2} [y_m(t+j) - y_n(t+j)] \frac{\partial y_n(t+j)}{\partial u(t+h)} + 2 \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j)] \frac{\partial \Delta u(t+j)}{\partial u(t+h)}, \quad (6)$$

где $h = 1, \dots, N_u$.

Чтобы получить каждый элемент Гессена, необходимо (6) дифференцировать относительно вектора \mathbf{U} :

$$\begin{aligned} \frac{\partial^2 J}{\partial u(t+m)\partial u(t+h)} = & 2 \sum_{j=N_1}^{N_2} \left\{ \frac{\partial^2 y_n(t+j)}{\partial u(t+m)\partial u(t+h)} [y_m(t+j) - y_n(t+j)] - \right. \\ & \left. - \frac{\partial y_m(t+j)}{\partial u(t+m)} \frac{\partial y_m(t+j)}{\partial u(t+h)} \right\} + \\ & + 2 \sum_{j=1}^{N_u} \lambda(j) \left\{ \frac{\partial \Delta n(t+j)}{\partial u(t+m)} \frac{\partial \Delta n(t+j)}{\partial u(t+h)} + \right. \\ & \left. + \frac{\partial^2 \Delta n(t+j)}{\partial u(t+m)\partial u(t+h)} \right\}. \quad (7) \end{aligned}$$

Элементы m^{th}, h^{th} матрицы Гессена в уравнении (7) получаются такими: $h = 1, \dots, N_u, m = 1, \dots, N_u$.

Архитектура нейронной сети. В качестве модели объекта в алгоритме NGPC используется многослойная прямая нейронная сеть [2]. Выход обученной нейронной сети используется в качестве прогнозируемого выхода объекта. Этот прогнозируемый выходной сигнал используется в алгоритме минимизации функционала качества.

Первоначальное обучение нейронной сети обычно выполняется в автономном режиме (offline) перед попыткой управления. В качестве прогнозируемого объекта используется выход обученной нейронной сети. На рис. 2 приведена блочная схема для обучения нейронной сети моделирующего управляемого объекта.

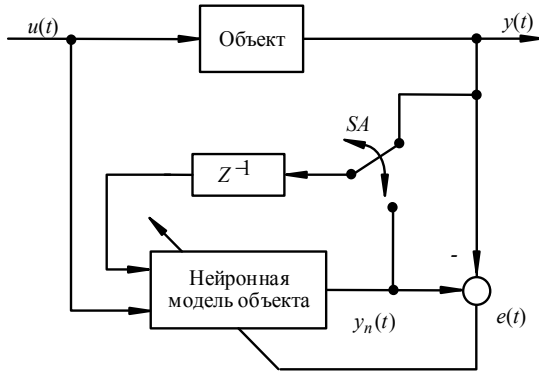


Рис. 2

Управляемый объект и нейронная сеть получают одно и то же входное значение $u(t)$. Нейронная сеть имеет дополнительный вход, сигнал с которого может поступать от выхода объекта $y(t)$ или от самого выхода нейронной модели объекта $y_n(t)$ через двухпозиционный переключатель. Тренировка нейронной сети и ее веса настраиваются таким образом, что множество входов создают желаемый набор выходов. Ошибка $e(t)$ образуется между выходом нейронной сети $y_n(t)$ и управляемого объекта $y(t)$, т. е. $e(t) = y_n(t) - y(t)$. Далее эта ошибка используется для обновления весов нейронной сети посредством обучения градиентного спуска. В качестве алгоритма обучения градиентного спуска используется метод Левенберга–Марквардата. На рис. 3 изображена многослойная нейронная сеть прямого распространения с элементами, имеющими узлы задержек по времени. Данная многослойная нейронная сеть прямого распространения состоит из внешних входов, $u(t)$ и $y(t-1)$ с соответствующими элементами задержки $u(t-1)$, $y(t-1)$, $y(t-2)$. Сеть имеет один скрытый слой, содержащий пять нейронов, и использует двухполярные активационные функции.

Уравнения нейронной сети данной архитектуры можно приставить следующим образом:

$$y_n(t) = \sum_{j=1}^{hid} w_j f_j [\text{net}(t)],$$

$$\text{net}_j(t) = \sum_{i=1}^{n_d} w_{j,i+1} u(t-i) + \sum_{i=1}^{d_d} w_{j,i+1} y(t-i),$$

(8)

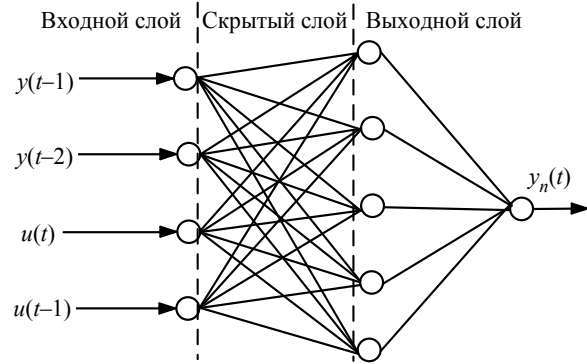


Рис. 3

где $y_n(t)$ – выход нейронной сети; hid – число нейронов в скрытом слое; w_j – вес, соединяющий j -й скрытый нейрон с нейроном выхода; $f_j(\cdot)$ – функция активации для j -го нейрона скрытого слоя; $u(t-i)$ – вход сети и его задержки; $\text{net}_j(t)$ – аргумент функции активации j -го нейрона; n_d – число узлов входа, ассоциированных с $u(\cdot)$, без учета $u(t)$; d_d – число узлов входа, ассоциированных с $y(\cdot)$; $w_{j,i}$ – вес, соединяющий i -й узел входа с j -м скрытым нейроном; $y(t-i)$ задержанный выход объекта, используемый как вход для сети; b_j – смещение j -го скрытого нейрона; b – смещение нейрона выхода.

Результаты моделирования. Нейроконтроллер с предсказанием (Neural Network Predictive Control, NNPC), реализованный в пакете прикладных программ (Neural Network Toolbox, NNT) в среде Matlab/Simulink [3], использует нелинейный управляемый объект в виде многослойной прямой нейронной сети, чтобы спрогнозировать будущее поведение объекта. Кроме того, нейроконтроллер с предсказанием NNPC способен вычислять сигнал управления, который оптимизирует поведение объекта на заданном интервале времени.

Нейроконтроллер с предсказанием работает по описанному ранее алгоритму. В [4] приведена процедура синтеза нейроконтроллера с предсказанием для управления нелинейными объектами. В качестве примера выполним синтез нейроконтроллера с предсказанием NNPC для прогнозирования скорости линейного перемещения электропривода подачи токарного станка. Кинематика приводов подачи металлорежущих станков имеет сложные конструкции и содержит упругие элементы. Наличие упругих элементов усложняет расчетную механическую систему, превращая ее многомассовую. Для упроще-

ния в данной статье рассматривается двухмассовая электромеханическая система (ЭМС) электропривода подачи токарного станка. Структурная схема нейроруправления, разработанная в среде Simulink, показана на рис. 4, а параметры структурной схемы электропривода подачи – в табл. 1. Упрощенная схема модели объекта управления представлена на рис. 5.

Таблица 1

| Параметр | Обозначение | Значение |
|----------------------------------------------------------------------------|--------------|----------|
| Момент инерции двигателя, кг · м ² | $J_{дв}$ | 0.0131 |
| Момент инерции механизма подачи, кг · м ² | $J_{мех}$ | 0.0186 |
| Коэффициент демпфирования механической части привода подачи, Н · м · с/рад | β_{12} | 5.6 |
| Коэффициент жесткости, Н · м/рад | C_{12} | 100 |
| Коэффициент механизма привода подачи | k_M | 1.592 |
| Максимальное продольное перемещение суппорта, мм | S_{max} | 9000 |

Рассмотрим несколько вариантов синтеза нейроконтроллера с предсказанием для установления его оптимальных параметров. Прежде чем установить параметры нейроконтроллера с предсказанием, необходимо построить модель управляемого объекта. Для выполнения данной процедуры следует идентифицировать управляемый объект, т. е. построить его нейросетевую модель, задавая основные параметры контроллера: размер

скрытого слоя определяется количеством используемых нейронов N_n ; тактика дискретности в секундах определяет интервал между двумя последовательностями момента съема данных Δt и длиной обучающей выборки N_B . После этого устанавливаются следующие параметры оптимизации: верхний предел суммирования в показателе качества N_2 , верхний предел суммирования при оценки мощности управления N_u , коэффициент веса для составляющей мощности управления ρ и параметр одномерного поиска α . Выбрана процедура `csrchbas`.

В результате многократных исследований было доказано, что величины N_u , ρ и α незначительно влияют на результаты синтеза нейроконтроллера, а значение верхнего предела суммирования N_2 существенно влияет на работу нейроконтроллера с предсказанием. Число итераций на 1 такт дискретности, который в данной статье принят $\gamma = 2$, слабо влияет на работу контроллера. Для данной решаемой задачи оптимальные значения находятся в пределах $N_2 = 4...12$.

Как показали исследования, идентификация объекта управления зависит от выбора количества нейронов скрытого слоя N_n и архитектуры нейронной сети. Выбор количества нейронов зависит от сложности динамики объекта управления, т. е. если выбрать количество нейронов больше

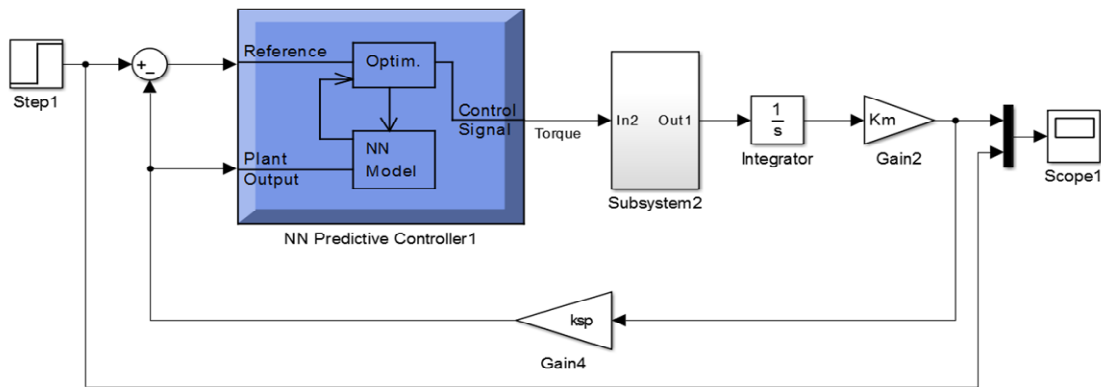


Рис. 4

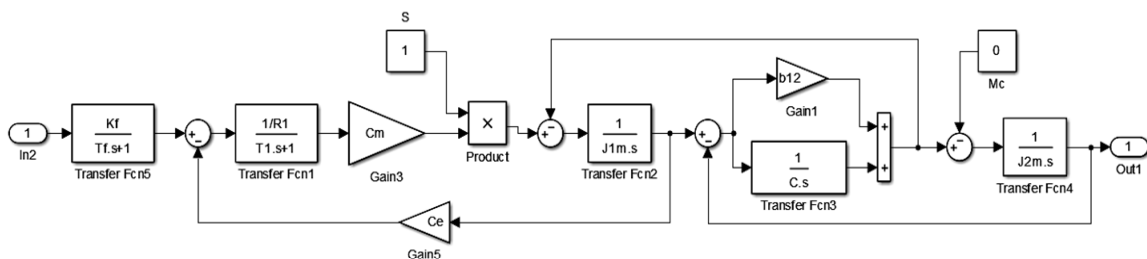


Рис. 5

Таблица 2

| Параметр нейроконтроллера (NN Predictive Control) | Обозначение | Вариант | | | | | |
|------------------------------------------------------|---------------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | <i>a</i> | <i>б</i> | <i>в</i> | <i>г</i> | <i>д</i> | <i>e</i> |
| Size of Hidden Layer | N_n | 3 | 5 | 6 | 8 | 10 | 12 |
| Sampling Interval | Δt | 0.15 | 0.3 | 0.2 | 0.4 | 0.19 | 0.17 |
| No. Delayed Plant Inputs | N_i | 2 | 2 | 2 | 2 | 2 | 2 |
| No. Delayed Plant Outputs | N_j | 2 | 2 | 2 | 2 | 2 | 2 |
| Training Epochs | – | 200 | 200 | 200 | 200 | 200 | 200 |
| Training samples | N_B | 700 | 800 | 1000 | 1200 | 1500 | 700 |
| Maximum Plant Input | Φ_{\max} | 1 | 1 | 1 | 1 | 1 | 1 |
| Minimum Plant Input | Φ_{\min} | -1 | -1 | -1 | -1 | -1 | -1 |
| Maximum Interval Value | t_{\max} | 1 | 1 | 1 | 1 | 1 | 1 |
| Minimum Interval Value | t_{\min} | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Cost Horizon | N_2 | 12 | 6 | 8 | 10 | 4 | 12 |
| Control Horizon | N_u | 2 | 2 | 2 | 2 | 2 | 2 |
| Control Weighting Factor | ρ | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| Search parameter | α | 10^{-3} | 10^{-3} | 10^{-3} | 10^{-3} | 10^{-3} | 10^{-3} |
| Iterations Per Sample Time | t | 2 | 2 | 2 | 2 | 2 | 2 |

Таблица 3

| Показатель качества | Обозначение | Вариант | | | | | |
|------------------------|-------------|----------|----------|----------|----------|----------|----------|
| | | <i>a</i> | <i>б</i> | <i>в</i> | <i>г</i> | <i>д</i> | <i>e</i> |
| Время регулирования, с | t_p | 4 | 10 | 6 | 9 | 2.3 | 2.1 |
| Время нарастания, с | t_n | 0.8 | 0.5 | 0.7 | 0.5 | 2.5 | 2.3 |
| Перерегулирование, % | σ | 20 | 38.46 | 31.42 | 66.66 | 0 | 0 |
| СКО, % | δ | 0.114 | 0.026 | 0.023 | 0.027 | 0.025 | 0.019 |

ожидаемого, то будет наблюдаться явление переобучения, и процесс обучения может занять длительное время, а при малом количестве нейронов сеть не может выполнить поставленную задачу. Было установлено, что для решения данной задачи количество нейронов находится в пределах $N_n = 10...12$. Обучение сети в значительной степени зависит от длины обучающей выборки N_B , и тактика дискретности Δt (с) определяет интервал между двумя последовательностями момента съема данных. Результат исследования показал, что если дискретность $\Delta t > 1$, то снижется точность вычисления и сокращается время обучения. Если дискретность $\Delta t < 1$, то происходит обратное. Таким образом, можно сделать вывод о том, что дискретность Δt оказывает большое влияние на среднеквадратичные ошибки, а также на обучение нейронной сети. Оптимальные значения в решаемой задаче находятся в пределах: $N_B = 700...1500$, $\Delta t = 0.4...0.17$. Была выбрана обучающая функция trainlm .

В результате проведенных исследований были установлены оптимальные параметры нейроконтроллера, которые соответствуют вариантам *a–e*, приведенным в табл. 2.

В табл. 3 представлены варианты прямых показателей качества полученных переходных процессов при синтезе нейроконтроллера с предсказанием. Из представленных в табл. 3 данных можно заключить, что вариант *e* показателей качества синтезируемого контроллера обеспечивает лучшее качество переходного процесса.

На рис. 6 представлены графики переходных процессов скорости линейного перемещения исполнительного органа привода подачи токарного станка (S , мм/с), которые соответствуют вариантам *a–e* табл. 2. Как видно из графиков, вариант *e* имеет высокую динамическую характеристику.

В результате проведенных исследований было доказано, что для обеспечения хороших динамических характеристик и повышения точности перемещения исполнительного органа (суппорта) привода подачи токарного станка с заданной тра-

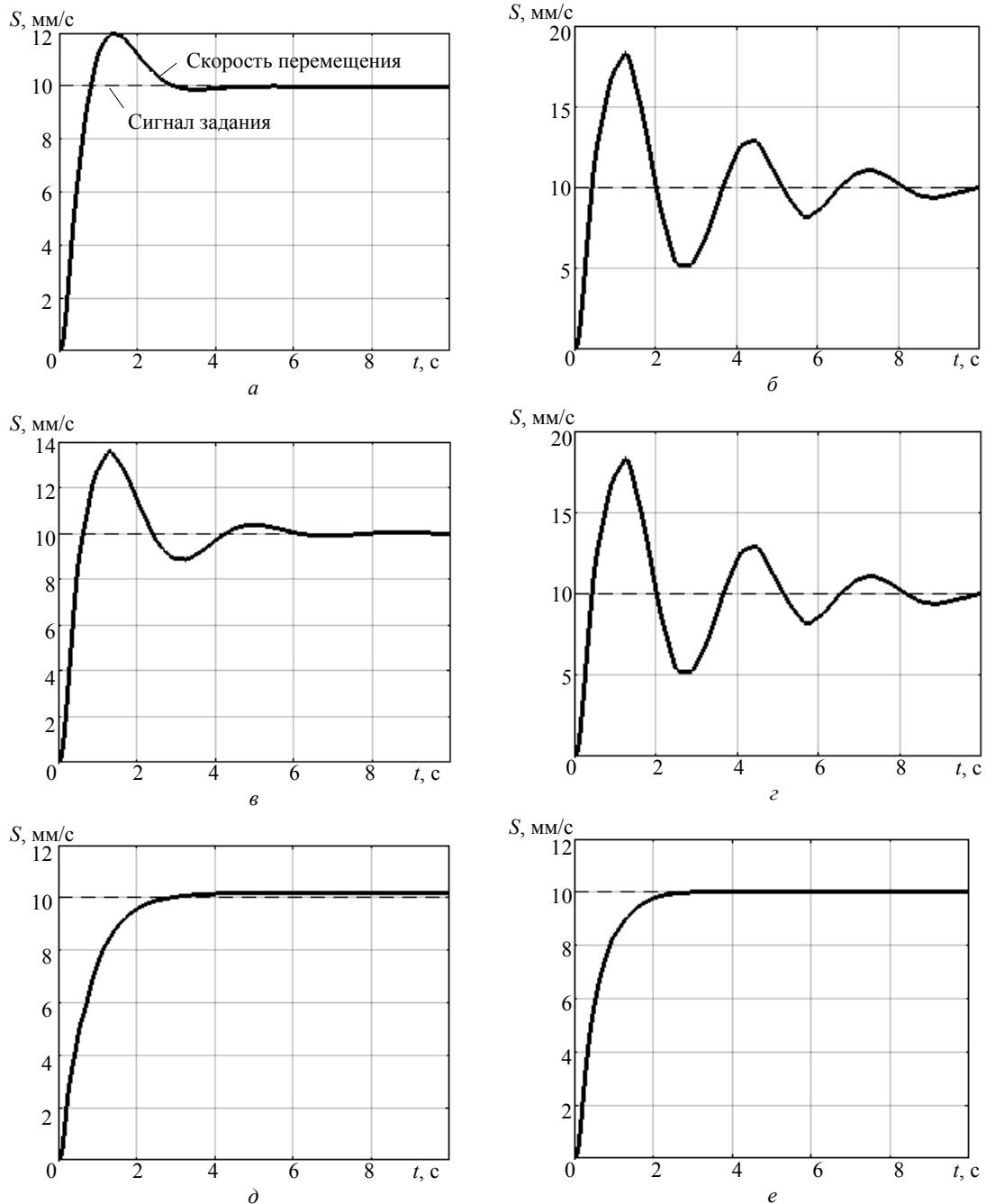


Рис. 6

екторией в качестве прогнозирующей модели (регулятор) можно использовать нейроконтроллер с предсказанием NNPC.

На основании проведенных исследований можно сделать следующие выводы:

1. Подтвердилась возможность применения алгоритма обобщенного управления с предсказанием и использованием многослойной нейронной сети для эффективного управления нелинейными объектами.

2. Для поиска оптимальных параметров нейроконтроллера с предсказанием было проведено

несколько вариантов синтеза контроллера с разными значениями.

3. Разработана многослойная прямая нейросетевая модель объекта в среде Matlab/Simulink, обеспечивающая высокие динамические характеристики системы управления приводами подачи токарного станка.

4. Недостаток нейроконтроллера с предсказанием заключается в том, что процесс обучения может занимать длительное время.

СПИСОК ЛИТЕРАТУРЫ

1. Merabet A., Ouhrouche M., Bui R.-T. Neural Generalized Predictive Controller for Induction Motor // Intern. J. of Theoretical and Appl. Comp. Sci. 2006. Vol. 1, № 1. P. 83–100.
2. Hunt K. J., Sbarbaro D. Neural Networks for Non-linear Internal Model Control // IEE Proc.-D. 1991. Vol. 138, № 5. P. 431–438.
3. Медведев В. С., Потемкин В. Г. Нейронные сети. MATLAB 6 / под общ. ред. канд. техн. наук В. Г. Потемкина. М.: ДИАЛОГ-МИФИ, 2002. 496 с.
4. Кузнецов Б. И., Василец Т. Е., Варфоломеев А. А. Синтез нейроконтроллера с предсказанием для двухмассовой электромеханической системы // Электротехника і електромеханіка. 2008. № 3. С. 27–32.

M. P. Belov, I. S. Nosirov, T. H. Phuong
Saint Petersburg Electrotechnical University «LETI»

ELECTRIC SERVO DRIVE OF TURNING LATHE CONTROL SYSTEM USING NEURAL GENERALIZED PREDICTIVE CONTROL

This paper deals with mathematical description of the algorithm for neural generalized Predictive Control (NGPC) with prediction and the Newton-Raphson optimization algorithm. The Newton-Raphson algorithm allows minimize the Cost Function Minimization (CFM) which determines the input signal necessary to obtain the desired behavior quality control. Identification procedure is considered that allows build neural network plant. The synthesis of neural network control is considered to ensure high static and dynamic accuracy when reproducing contour-positional displacements. A model control system for electric drive of a lathe with using Neural Network Predictive Control (NNPC) is developed in Matlab/Simulink environment. Several variants of the NNPC were synthesized with different values to establish its optimal parameters. The parameters of NNPC regulator prediction are determined that they significantly effect to the quality servo control system.

Electric drive of turning of the lathe, neural generalized Predictive Control, synthesis of the neuron controller, Newton-Raphson algorithm
