

НЕЙРОУПРАВЛЕНИЕ НЕЛИНЕЙНЫМ ДИНАМИЧЕСКИМ ОБЪЕКТОМ С ИСПОЛЬЗОВАНИЕМ МЕТОДА ОБОБЩЕННОГО УПРАВЛЕНИЯ С ПРЕДСКАЗАНИЕМ

Кузнецов Б.И., д.т.н., проф., Василец Т.Е., к.т.н., доц., Варфоломеев А.А.
Украинская инженерно – педагогическая академия
Украина, 61003, Харьков, ул. Университетская, 16, УИПА, кафедра СУТПиО
тел. (057) 733-79-59

У роботі запропоновано використовувати нейромережевий регулятор з прогнозом для забезпечення високих динамічних характеристик системи управління нелінійним динамічним об'єктом. При побудові регулятора використовується ефективна реалізація узагальненого управління з прогнозом із застосуванням багатослоєвої прямонаправленої нейронної мережі, як нелінійної моделі об'єкту управління. Розроблена схема системи управління, виконаний синтез регулятора і проведено моделювання системи.

В работе предложено использовать нейросетевой регулятор с предсказанием для обеспечения высоких динамических характеристик системы управления нелинейным динамическим объектом. При построении регулятора используется эффективная реализация обобщенного управления с предсказанием с применением многослойной прямонаправленной нейронной сети, как нелинейной модели объекта управления. Разработана схема системы управления, выполнен синтез регулятора и проведено моделирование системы.

Постановка проблемы. Наиболее эффективным направлением при модернизации систем управления огнём легкобронированных машин является совершенствование системы стабилизации основного и вспомогательного вооружения. В настоящее время широко используются двухканальные системы стабилизации вооружения. Боевой модуль таких систем устанавливается на корпусе машины с помощью погонной шестерни, относительно оси, которой он поворачивается в горизонтальной плоскости, а блок вооружения крепится на боевом модуле с помощью цапф, относительно оси, которых он поворачивается в вертикальной плоскости. Такое построение обеспечивает стабилизацию и стабилизированное наведение средств ведения огня в горизонтальной и вертикальной плоскости, создавая условия для ведения боевых действий с места и с ходу по наземным и воздушным целям. Однако используемые в настоящее время системы стабилизации вооружения не могут обеспечивать требуемых для эффективного ведения огня значений ошибки стабилизации, диапазона регулирования скорости, неплавности наведения и др. Поэтому разработка систем управления вооружением легкобронированных машин, имеющих высокие динамические характеристики, является важной и актуальной. Одним из наиболее перспективных направлений является построение нейросетевых систем управления.

Анализ последних достижений и публикаций. Количество структур систем нейрорегулирования, предложенных и опробованных на сегодняшний день, очень велико. Их обзор приводится, например, в [1]. Одни системы нейрорегулирования используют аналитический закон регулирования и нейромодель. Нейронная сеть восстанавливает при этом элементы матриц модели объекта. Это так называемые структурированные методы регулирования. Однако полностью использовать весь аппроксимационный потенциал нейронных сетей позволяют неструктурированные методы регулирования. Среди структур систем нейрорегулирования, актуальных на сегодняшний

день, можно отметить следующие. 1. Прямое инверсное регулирование [2], при котором сначала в режиме offline формируют модель инверсной динамики объекта регулирования, а затем полученная инверсная модель включается последовательно с объектом и используется в качестве регулятора. 2. Регулирование с Feedback-Error-Learning [3] аналогично прямому инверсному регулированию с классическим, например, ПИД-регулятором. 3. Регулирование методом компенсации задания [4], при котором компенсация имеющихся нелинейностей осуществляется на уровне сигналов задания, а не сигналов воздействия на объект. Преимущество данной техники – простота встраивания в готовый контур регулирования. Сложность метода заключается в выборе начальных значений параметров сети, т.к. иначе возможно формирование небезопасных сигналов задания. Существует ряд других методов, например: косвенное инверсное нейрорегулирование, регулирование с внутренней моделью, Instantaneous linearization и др. Наиболее эффективным для решения поставленной в настоящей работе задачи, является метод предиктивного регулирования на основе модели объекта [5].

Цель статьи. Целью работы является синтез нейрорегулятора, использующего принцип нелинейного предиктивного регулирования, для построения нейросетевой системы управления вооружением легкобронированной машины, имеющей заданные динамические характеристики.

Изложение материала исследования, полученных научных результатов. В работе используется эффективная реализация обобщенного управления с предсказанием (Generalized Predictive Control, GPC) с использованием многослойной прямонаправленной нейронной сети, как нелинейной модели объекта управления [6]. Благодаря использованию оптимизационного алгоритма Ньютона – Рафсона, число итераций, необходимых для сходимости, значительно меньше, чем при использовании других методов. Главные затраты алгоритма Ньютона-Рафсона в вы-

числении Гесса, но даже с этими дополнительными расчетами низкое число итераций делает алгоритм Ньютона-Рафсона более быстрым, чем другие методы, при этом он может использоваться для управления в режиме реального времени (real-time control).

Алгоритм нейроуправления с предсказанием

Рассмотрим кратко основные положения обобщенного алгоритма нейроуправления с предсказанием (Neural Generalized Predictive Control algorithm) с применением оптимизационного алгоритма Ньютона-Рафсона. Обобщенное управление с предсказанием, принадлежит к классу методов цифрового управления, называемых модель-ориентированное управление с предсказанием (Model-Based Predictive Control, MBPC) [6]. MBPC техники были успешно проанализированы и внедрены в процесс управления, т.к. они системно учитывают ограничения реальных объектов в режиме реального времени. GPC применяются в управлении не-минимально фазовыми объектами, неустойчивыми объектами с открытым циклом и объектами, с переменным или неизвестным временем задержки. Обобщенное управление с предсказанием также является робастным по отношению к ошибкам

моделирования, переопределению и недоопределению значений параметров и шумам датчиков. GPC было первоначально разработано для моделей линейных объектов, что приводит к уравнениям, которые могут быть решены аналитически. При использовании нелинейных моделей необходимы нелинейные оптимизационные алгоритмы. Это влияет на качество и эффективность вычислений, определяющих управляющие входные сигналы. Для нелинейных объектов, способность GPC делать точные предсказания может быть усовершенствована, если для изучения динамики объекта вместо стандартных методов нелинейного моделирования применять нейронные сети. Выбор минимизационного алгоритма влияет на вычислительную эффективность алгоритма. Как было указано, при использовании для оптимизации алгоритма Ньютона-Рафсона, число итерации, необходимых для сходимости, существенно меньше, чем при использовании других методов.

Схема системы обобщенного нейроуправления с предсказанием (Neural Generalized Predictive Control, NGPC) показана на рис. 1.

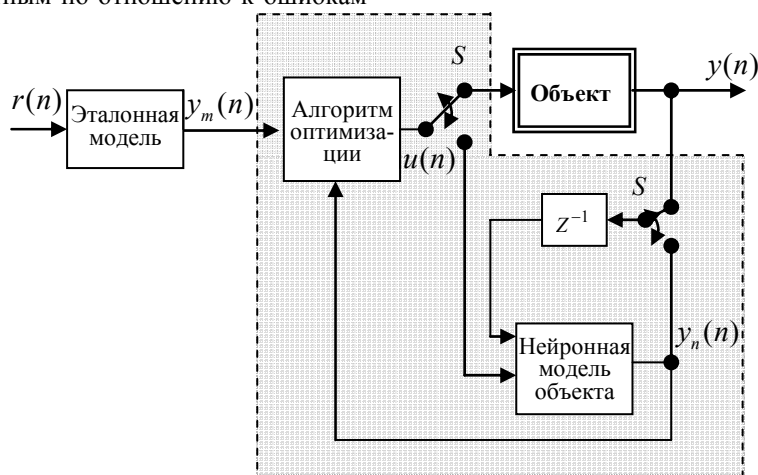


Рис. 1. Схема системы обобщенного нейроуправления с предсказанием

Она состоит из четырех компонент: управляемого объекта, эталонной модели, которая описывает желаемое качество объекта, моделирующей объект нейронной сети, и алгоритма минимизации функционала качества (Cost Function Minimization, CFM), определяющего входной сигнал, необходимый для достижения желаемого поведения объекта. NGPC алгоритм состоит из блока CFM и блока нейронной сети.

Принцип работы системы состоит в следующем. Входной сигнал $r(n)$ подается эталонной модели. Эта модель выдает эталонный сигнал $y_m(n)$, который служит входом для CFM блока. CFM алгоритм рассчитывает сигнал, который служит входом для объекта или модели объекта. Двухполюсной двухпозиционный переключатель S устанавливается в положение к объекту, когда CFM алгоритм используется для определения оптимального входного сигнала $u(n)$, который минимизирует выбранный критерий качества управления. Между тактами переключатель установлен в положение к модели объекта, где CFM алгоритм использует эту модель для расчета следующего

управляющего входа $u(n+1)$, путем предсказания ответного сигнала, полученного от модели объекта. Как только функционал качества минимизирован, этот входной сигнал подается на объект.

Обобщенный алгоритм нейроуправления с предсказанием состоит из следующих основных шагов:

- 1) генерируется задающая траектория; если бу- дущая траектория $y_m(n)$ неизвестна, полагается $y_m(n) = const$;
- 2) используя предварительно рассчитанный управляющий входной вектор и нейронную модель объекта управления, выполняется предсказание пове- дения объекта;
- 3) рассчитывается новый управляющий входной сигнал, минимизирующий функционал качества;
- 4) повторяются шаги 2 и 3, пока не будет дос- тигнута требуемая минимизация;
- 5) посылается первый управляющий входной сигнал на объект;
- 6) повторяется весь процесс для каждого вре- менного шага.

Качество вычисления при реализации обобщенного управления с предсказанием в основном зависит от выбора минимизирующего алгоритма для CFM блока. Существуют несколько минимизационных алгоритмов, которые реализуются в CFM, такие как Non-gradient, Simplex и Successive Quadratic Programming. Выбор минимизационного метода может быть основан на следующих критериях: число итераций, вычислительные затраты и точность решения. В общем случае, эти алгоритмы требуют большого числа итераций, что делает управление в режиме реального времени затруднительным. Очень небольшое число публикаций посвящены реализациям управления в режиме реального времени для объектов, имеющих большие постоянные времени. Для возможности использования в системах с объектами, имеющими малые постоянные времени, требуется более быстрый оптимизационный алгоритм. Алгоритм Ньютона-Рафсона обладает квадратичной сходимостью, в то время как другие алгоритмы имеют более низкую скорость сходимости. Более высокая скорость сходимости алгоритма Ньютона-Рафсона требует высоких вычислительных затрат, но оправдывается показателем скорости сходимости.

Качество модели объекта влияет на точность предсказания. Для реализации обобщенного управления с предсказанием требуется адекватная модель объекта. Существуют методы, обеспечивающие высокую точность моделирование в случае линейного объекта. Но, в случае нелинейного объекта, задача является более сложной. На сегодняшний день есть два подхода для моделирования нелинейных объектов. Первый заключается в линеаризации объекта на множестве рабочих точек. Если объект существенно нелинейный, то множество рабочих точек может быть очень большим. Второй подход заключается в разработке нелинейной модели, которая зависит от допущений относительно динамики нелинейного объекта. Если эти допущения не верны, то снижается точность модели. Модели с использованием нейросетей дают возможность справиться с нелинейной динамикой. Для нелинейного объекта способность GPS делать точные предсказания может быть значительно улучшена, если для изучения динамики объекта использовать нейронную сеть, вместо стандартных подходов моделирования.

Обобщенный алгоритм нейроуправления с предсказанием основывается на минимизации функционала качества на конечном диапазоне предсказаний. Функционала качества используемый в настоящей работе, имеет следующий вид

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

где N_1 – нижний предел предсказания; N_2 - верхний предел предсказания; N_u – диапазон управления; y_m – желаемая траектория; y_n – предсказанный выход нейронной сети; λ – весовой множитель; $\Delta u(n+j)$ – изменение u , определяемое следующим образом: $u(n+j) - u(n+j-1)$.

Этот функционал минимизирует не только сред-

неквадратическую ошибку между эталонным сигналом и сигналом, выдаваемым моделью объекта, но также взвешенную среднеквадратическую скорость изменения управляющего сигнала.

Когда функционал качества минимизирован, генерируется управляющий входной сигнал, что позволяет объекту отслеживать заданную траекторию с некоторой точностью. Функционал качества содержит четыре настраиваемых параметра, N_1 , N_2 , N_u , λ . Параметры N_1 и N_2 , задают пределы, внутри которых вычисляется ошибка слежения. N_u является границей диапазона управления. Единственное ограничение, накладываемое на значения N_u и N_1 следующее: они должны быть меньше или равны N_2 . Вторая сумма содержит весовой множитель λ , который введен для управления балансом между первой и второй суммой.

Для реализации оптимального управления, функционал качества минимизируется с помощью алгоритма Ньютона-Рафсона. Цель алгоритма – минимизировать функционал J (1) по вектору $[u(n+1), u(n+2), \dots, u(n+N_u)]^T$. Обозначим данный вектор как \mathbf{U} . Используя в качестве алгоритма CFM алгоритм Ньютона-Рафсона, итерационно минимизируется функционал J для определения оптимального \mathbf{U} . Итерационный процесс определяет значение J на каждой итерации, которое обозначим как $J(k)$. Для каждой итерации рассчитывается также значение управляющего вектора:

$$\mathbf{U}(k) = [u(n+1); u(n+2); \dots; u(n+N_u)]^T,$$

где $k = 1, \dots$ - номер итерации, T - знак транспонирования.

Правило обновления $\mathbf{U}(k+1)$ в алгоритме Ньютона-Рафсона выглядит следующим образом

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

где якобиан обозначен

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

а гессиан

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

Решение уравнения (2) требует явного обращения матрицы Гессе. Этот процесс может потребовать больших вычислительных затрат. Одним из подходов, используемых, чтобы избежать обращения матрицы, служит LU декомпозиция [8] для получения входного вектора $\mathbf{U}(k+1)$. Это достигается путем записи уравнения (2) в виде системы линейных уравнений $\mathbf{Ax} = \mathbf{b}$, что приводит к выражению

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

где $\frac{\partial^2 \mathbf{J}}{\partial \mathbf{U}^2}(k) = \mathbf{A}$; $-\frac{\partial \mathbf{J}}{\partial \mathbf{U}}(k) = \mathbf{b}$; $\mathbf{U}(k+1) - \mathbf{U}(k) = \mathbf{x}$.

В данной форме уравнение (3) может быть решено с помощью двух процедур, содержащихся в [6], процедура LU декомпозиции, ludcmp, и процедура решения системы линейных уравнений lubksb. После того, как вычислен вектор \mathbf{x} , $\mathbf{U}(k+1)$ получается из соотношения $\mathbf{U}(k+1) = \mathbf{U}(k) + \mathbf{x}$. Процедура повторяется до тех пор, пока изменение каждой компоненты $\mathbf{U}(k+1)$ не становится меньшим некоторого ε . При решении системы относительно \mathbf{x} , для каждой итерации алгоритма Ньютона-Рафсона необходимо вычисление каждого элемента якобиана и гессиана. Элемент якобиана с индексом h имеет следующий вид

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

Если расписать выражение $\frac{\partial \Delta u(n+j)}{\partial u(n+h)}$ и воспользоваться символом Кронекера дельта функции, которая определяется как $\delta(h, j) = 1$ если $h = j$ и $\delta(h, j) = 0$ если $h \neq j$, получим $\frac{\partial u(n+j)}{\partial u(n+h)} - \frac{\partial u(n+j-1)}{\partial u(n+h)} = \delta(h, j) - \delta(h, j-1)$.

Элемент с индексами m, h гессиана определяется следующим образом:

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

$$h = 1, \dots, N_u; \quad m = 1, \dots, N_u.$$

Можно воспользоваться также определением дельта-функции для выражения

$$\begin{aligned} \frac{\partial \Delta u(n+j)}{\partial u(n+m)} \frac{\partial \Delta u(n+j)}{\partial u(n+h)} &= \\ &= (\delta(h, j) - \delta(h, j-1)) (\delta(m, j) - \delta(m, j-1)) \end{aligned}$$

Выражение $\frac{\partial^2 \Delta u(n+j)}{\partial u(n+m) \partial u(n+h)}$ всегда равняется нулю.

Последняя компонента, требуемая для получения $\mathbf{U}(k+1)$, это вычисление выхода объекта $y_n(n+j)$ и его производных. Для этого используется нейронная сеть.

Структуры нейронной сети для моделирования объекта. Моделью объекта в алгоритме NGPC служит нейронная сеть. Начальная тренировка нейронной сети обычно выполняется в автономном режиме (offline), до применения управления. Блочная схема для тренировки нейронной сети, моделирующей объект, приведена на рис. 2

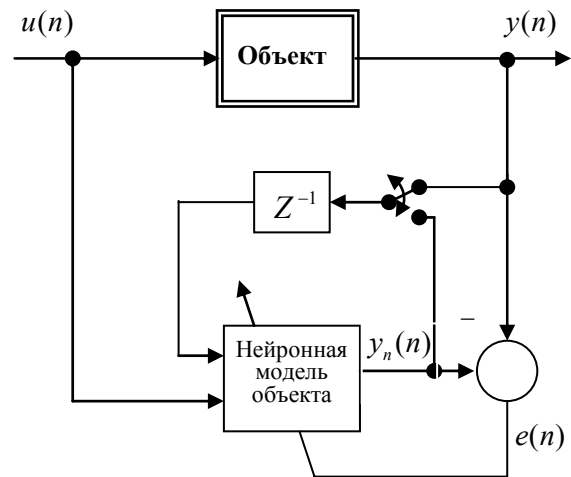


Рис. 2. Схема тренировки сети в режиме offline

Нейронная сеть и объект получают одно и то же входное значение $u(n)$. Нейронная сеть также имеет дополнительный вход, на который подается либо выходное значением объекта, $y(n)$, либо нейронной сети, $y_n(n)$. Выбор значения зависит от шага предсказания. В процессе тренировки сети ее веса настраиваются таким образом, что множество входных значений продуцирует желаемые выходные значения. Ошибка формируется между ответами нейронной сети, $y_n(n)$, и объекта, $y(n)$. В дальнейшем эта ошибка используется для обновления весов с помощью градиентного спуска. Этот процесс повторяется до тех пор, пока ошибка не достигнет допустимого уровня.

Так как для моделирования объекта будет использована нейронная сеть, должна быть рассмотрена конфигурация архитектуры нейронной сети. Нейронная сеть может быть настроена либо для моделирования входа/выхода, либо для моделирования пространства состояний. В данной реализации алгоритм NGPC используются модели, использующие только входные/выходные координаты, т.к. моделирование пространства состояний требует измерения состояний объекта, что не всегда возможно. Сети типа многослойный перцептрон (МП) отличаются универсальностью при моделировании различных объектов регулирования. В связи с этим дальнейшие исследования ведутся только с этим типом сети.

МП в настоящее время – наиболее полно теоретически исследованная сеть. Существующие теоремы утверждают, что МП при условии достаточного количества нейронов могут отражать практически любые взаимосвязи. Поэтому они часто используются для

построения нейросетевых моделей нелинейных объектов. Однако первое, с чем сталкиваются исследователи, это вопрос о том, какую структуру должна иметь сеть. Нахождение оптимальной структуры сети - проблема, которая до сих пор полностью не решена. В этом случае на помощь приходит целенаправленное моделирование и практический опыт.

Определение структуры нейронной сети состоит, во-первых, из определения входов сети и, во-вторых, из определения внутренней топологии сети (количество слоев и нейронов). На рис. 3, приведена многослойная нейронная сеть прямого распространения со структурой, имеющей узлы задержек по времени. В данном примере входы сети состоят из внешних входов, $u(n)$ и $y(n-1)$ и соответствующих им узлов задержек $u(n-1), \dots, u(n-n_d)$ и $y(n-2), \dots, y(n-d_d)$.

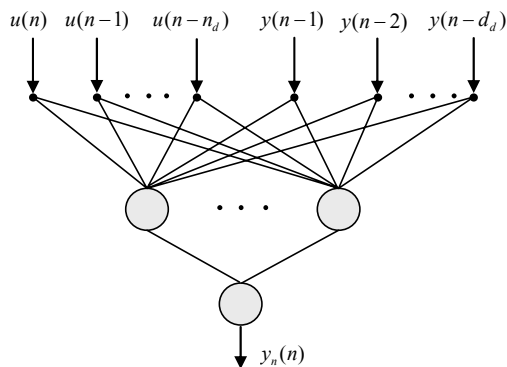


Рис. 3. Многослойная нейронная сеть прямого распространения с элементами задержек по времени

Параметры n_d , d_d отражают число узлов задержки, относящихся к соответствующим узлам входа. Второй вход сети может иметь внешний вход $y_n(n)$ и соответствующие задержанные значения. Сеть имеет один скрытый слой, состоящий из нескольких скрытых нейронов, которые используют общую функцию вывода (активации) $f_j(\cdot)$. Выходной нейрон использует линейную функцию вывода с единичным наклоном для масштабирования выходных значений. Уравнения сети с данной архитектурой выглядят следующим образом

$$y_n(n) = \sum_{j=1}^{S^{L-1}} w_j f_j(\text{net}_j(n)) + b, \quad (5)$$

$$n_j(l) = \sum_{i=0}^{n_d} w_{j,i} u(n-i) + \sum_{i=1}^{d_d} w_{j,n_d+i+1} y(l-i) + b_j, \quad (6)$$

где $y_n(n)$ – выход нейронной сети; $f_j(\cdot)$ – функция активации для j -го нейрона скрытого слоя; $\text{net}_j(n)$ – аргумент функции активизации j -го нейрона; S^{L-1} – число нейронов в скрытом слое; n_d – число узлов входа, ассоциированных с $u(\cdot)$ без учета $u(n)$; d_d – число узлов входа, ассоциированных с $y(\cdot)$; w_j – вес, соединяющий j -й скрытый нейрон с нейроном выхода; $w_{j,i}$ вес, соединяющий i -й узел входа с j -м скрытым нейроном; $y(n-i)$ – задержанный выход объек-

та, используемый как вход для сети; $y(n-i)$ – вход сети и его задержки; b_j – смещение j -го скрытого нейрона; b – смещение нейрона выхода.

Предсказание с использованием нейронной сети. NGPC алгоритм использует выход модели объекта для предсказания динамики объекта для произвольного входа от текущего момента времени n до некоторого будущего момента $n+k$. Это достигается сдвигом уравнений (4) на k , имеем:

$$y_n(n+k) = \sum_{j=1}^{S^{L-1}} \{w_j f_j(\text{net}_j(n+k))\} + b, \quad (5)$$

$$n_j(n+k) = \sum_{i=0}^{n_d} w_{j,i+1} \begin{cases} u(n+k-i), & k-N_u < i \\ u(n+N_u), & k-N_u \geq i \end{cases} + \sum_{i=1}^{\min(k,d_d)} (w_{j,n_d+i+1} y_n(n+k-i)) + \sum_{i=k+1}^{d_d} w_{j,n_d+i+1} y_n(n+k-i) + b_j. \quad (6)$$

Особенности уравнения (6) связаны с рекурсивным характером предсказания. Первая сумма в (6) определяется в соответствии с приведенными условиями. Условие $k-N_u < i$ управляет предварительными будущими значениями u до $u(n+N_u-1)$. Условие $k-N_u \geq i$ устанавливает входы от $u(n+N_u)$ до $u(n+k)$ равными $u(n+N_u)$. Второе условие возникает только, если $N_2 > N_u$. Следующее суммирование в (6) управляют рекурсивной частью предсказания. Суммирование соответствует обратной связи по выходу нейронной сети y_n , и выполняется k либо d_d раз, в зависимости от того, какое значение меньше. Последняя сумма в (6) содержит предыдущие значения y . В следующем подразделе выводятся производные от уравнений (5) и (6) для входа $u(n+h)$.

Производные уравнений нейронной сети. Для минимизации функционала качества в режиме реального времени с помощью итерационного решения, основанного на градиенте, необходимо использовать формулы для эффективного вычисления градиента нейронной сети.

Для вычисления якобиана и гессиана необходимы первая и вторая производная сети по входному вектору. Элементы якобиана получаются дифференцированием $y_n(n+k)$ из уравнения (6) по $u(n+h)$

$$\frac{\partial y_n(n+k)}{\partial u(n+h)} = \sum_{j=1}^{S^{L-1}} w_j \frac{\partial f_j(\text{net}_j(n+k))}{\partial \text{net}_j(n+k)}. \quad (7)$$

Применяя правило дифференцирования сложной функции для $\frac{\partial f_j(\text{net}_j(n+k))}{\partial u(n+h)}$ имеем

$$\frac{\partial f_j(\text{net}_j(n+k))}{\partial u(n+h)} = \frac{\partial f_j(\text{net}_j(n+k))}{\partial \text{net}_j(n+k)} \frac{\partial \text{net}_j(n+k)}{\partial u(n+h)}, \quad (8)$$

где $\frac{\partial f_j(\text{net}_j(n+k))}{\partial \text{net}_j(n+k)}$ – производная функции выхода и

$$\frac{\partial net_j(n+k)}{\partial u(n+h)} = \sum_{i=0}^{n_d} w_{j,i+1} \begin{cases} \delta(k-i, h), & k - N_u < i \\ \delta(N_u, h), & k - N_u \geq i \end{cases} + \\ + \sum_{i=1}^{\min(k, d_d)} \left(w_{j, n_d+i+1} \frac{y_n(n+k-i)}{\partial u(n+h)} \delta_1(k-i-1) \right). \quad (9)$$

Заметим, что в (9) в последней сумме введена ступенчатая функция δ_1 . Это было сделано, чтобы подчеркнуть, что сумма равна нулю для $k-i < 1$, следовательно слагаемые, удовлетворяющие этому условию, не требуют вычисления. Чтобы получить элементы гессиана, необходимо продифференцировать уравнения (7), (8) и (9) по $u(n+m)$, в результате имеем

$$\frac{\partial^2 y_n(n+k)}{\partial u(n+h) \partial u(n+m)} = \sum_{j=1}^{L-1} w_j \frac{\partial^2 f_j(net_j(n+k))}{\partial u(n+h) \partial u(n+m)}, \quad (10)$$

$$\frac{\partial^2 f_j(net_j(n+k))}{\partial u(n+h) \partial u(n+m)} = \frac{\partial f_j(net_j(n+k))}{\partial net_j(n+k)} \frac{\partial^2 f_j(net_j(n+k))}{\partial u(n+h) \partial u(n+m)} + \\ + \frac{\partial^2 f_j(net_j(n+k))}{\partial net_j(n+k)^2} \frac{\partial net_j(n+k)}{\partial u(n+h)} \frac{\partial net_j(n+k)}{\partial u(n+m)}, \quad (11)$$

$$\frac{\partial^2 net_j(n+k)}{\partial u(n+h) \partial u(n+m)} = \\ = \sum_{i=1}^{\min(k, d_d)} w_{j, n_d+i+1} \frac{\partial^2 y_n(n+k-i)}{\partial u(n+h) \partial u(n+m)} \delta_1(k-i-1)$$

Уравнение (11) является результатом применения правила дифференцирования сложной функции дважды.

Следует заметить, что нелинейные оптимизации являются процессами, требующими больших вычислительных затрат. Алгоритм Ньютона-Рафсона используется для создания эффективного вычислительного процесса. При написании кода, реализующего данный алгоритм, все значения, посчитанные в одной процедуре, передаются другим процедурам для избежания повторения расчета. Наиболее важная часть кода – вычисление гессиана. Переменные $\frac{\partial net_j(n+k)}{\partial u(n+h)}$ и $\frac{y_n(n+k)}{\partial u(n+h)}$ вычисляемые для якобиана, также используются для вычисления гессиана, поэтому они передаются в процедуру вычисления гессиана. Т.к. гессиан симметричен, только верхнетреугольная часть матрицы подлежит определению. Использование данных двух замечаний, позволяет значительно снизить время загрузки процессора.

Таким образом, обобщенное нейруправление с предсказанием с использованием оптимизационного алгоритма Ньютона-Рафсона для минимизации функционала качества позволяет получить высокое качество управления в случае сложных нелинейностей в объекте и даёт возможность применения алгоритма в режиме реального времени.

Контроллер с предсказанием (NN Predictive Controller). В пакете прикладных программ Neural Network Toolbox системы MATLAB реализован контроллер с предсказанием NN Predictive Controller, при построении которого использован принцип нелинейного предиктивного управления, описанный выше. При синтезе нейроконтроллера NN Prediction Controller ис-

пользуются следующие файлы, размещенные в каталоге toolbox/nnet/nncontrol системы SIMULINK.

Функции одномерной оптимизации: Csrchbac – поиск с обратной прогонкой; Csrchbre – метод Брента, объединяющий методы золотого сечения и квадратичной интерполяции; Csrchcha – метод кубической интерполяции Чараламбуca; Csrchgol – метод золотого сечения; Csrchhyb – гибридный метод бисекции и кубической интерполяции.

Функции для синтеза управления с предсказанием: Calcjddj – вычисление функционала качества и его градиента; Predopt – оптимизация регулятора с предсказанием; Dyduvar – вычисление частных производных выхода по входу.

Модели SIMULINK. Predcstr – GUI – приложение для контроллера с предсказанием; Prest3sim2 – нейросетевая модель объекта управления используемая M – функцией deport для предсказания процесса в будущем.

Вспомогательные функции: Sfunxy2 – функция для вывода графиков; Nncontrolutil – поддержка, обеспечивающая возможность обращения к частным функциям из системы SIMULINK; Nnident.m – основная функция, используемая при идентификации объекта, находящаяся в каталоге private. Обеспечивает GUI пользователя, генерацию обучающей выборки, создание и тренировку сети.

В качестве примера выполним синтез нейросетевой системы наведения и стабилизации вооружения легкобронированных машин с использованием нейроконтроллера NN Predictive Controller. Исполнительное устройство стабилизатора вооружения состоит из усилителя (преобразователя) мощности, приводного двигателя и кинематического устройства сопряжения. Реальная кинематическая схема системы наведения и стабилизации содержит упругие элементы (элементы конечной жесткости). Наличие упругих элементов усложняет расчетную схему механической части системы, превращая её в многомассовую. Исследования показали, что с достаточной для практических расчётов точностью механическую часть системы может быть представлена в виде двухмассовой системы.

В результате исследований установлено, что эффективной является система с нейронной компенсацией нелинейностей исполнительного устройства стабилизатора вооружения в контуре скорости, при этом в контуре положения может быть использован простой П – регулятор. Структурная схема нейросетевой системы управления, разработанная в Simulink системы MATLAB показана на рис. 4. Схема включает блок контроллера NN Prediction Controller, блоки генерации эталонного ступенчатого сигнала со случайной амплитудой Random Reference, блоки построения графиков и блоки, относящиеся к объекту управления. Схема модели объекта управления приведена на рис. 5. В схеме не учитывается возмущающий момент, обусловленный колебаниями корпуса носителя при движении машины по пересечённой местности и люфт между зубьями ведущей и ведомой шестерни. Схема объекта управления составлена с учётом момента сухого трения в подшипниках ПД и момента сухого трения кинематического устройства сопряжения. Для их задания использованы два блока MATLAB Fcn, предназначенные для написания выражений на языке MATLAB.

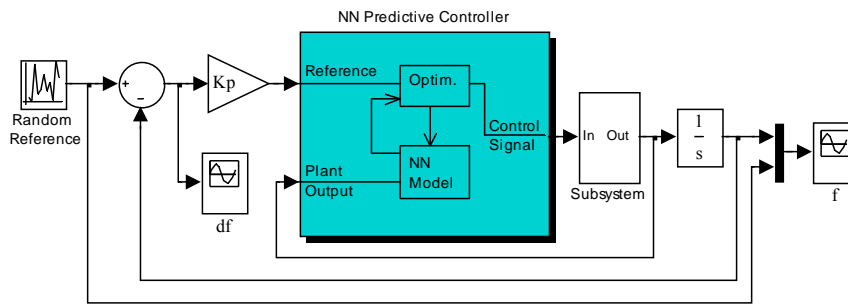


Рис. 4. Схема нейросетевой системы управления

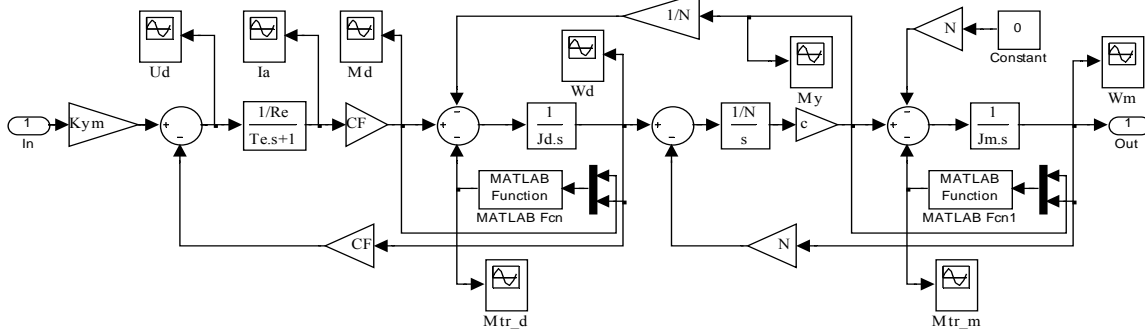


Рис. 5. Схема модели объекта управления

Процесс синтеза нейроконтроллера NN Prediction Controller начинается с построения нейросетевой модели объекта управления. Процедура идентификации позволяет построить нейронную сеть, которая будет моделировать динамику объекта. Поскольку модель должна использоваться при настройке контроллера, то её следует создать прежде, чем начнется расчет контроллера. Вначале задаются параметры и архитектура сети, а затем выполняется процедура её обучения. Для этого используется предварительно сгенерированная обучающая последовательность, которая формируется путем воздействия ряда случайных ступенчатых сигналов на модель SIMULINK управляемого объекта (рис. 5). После завершения обучения сети параметры нейросетевой модели управляемого объекта вводятся в блок NN Predictive Controller системы Simulink. В системе Simulink формируется так же схема ptest3sim2, показанная на рис. 6.

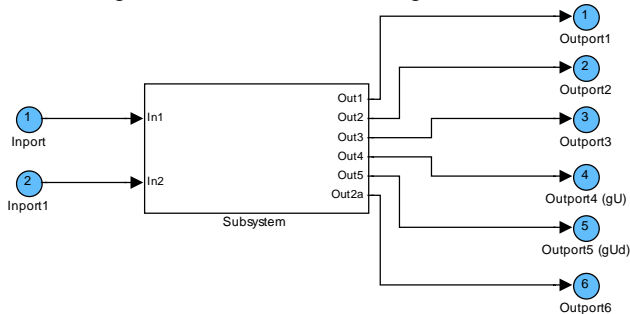


Рис. 6. Схема ptest3sim2, используемая М-функцией predopt для предсказания процесса в будущем

При активизации блока Subsystem открывается окно со схемой рис. 7. Данная схема так же является нейросетевой моделью объекта управления, имеющей дополнительные выходы, и используется М-функцией predopt для предсказания процесса в будущем.

После завершения построения нейросетевой модели управляемого объекта задаются параметры оптимизации. Затем они вводятся в блок NN Predictive controller системы Simulink.

На рис. 8 приведены графики переходных процессов переменных состояния синтезированной системы. Все обозначения переменных соответствуют рис. 4 и рис. 5. Переходные процессы приведены для значений скачков задающего воздействия, находящихся в пределах $\varphi_3 = -0,1 \div 0,1$ рад.

Как видно из графиков, переходные процессы имеют высокие динамические характеристики. Отработка угла происходит за минимально возможное время, определяемое постоянными времени электропривода, при номинальном напряжении двигателя U_d . Так, например, угол $\varphi_3 = 0,05$ рад отрабатывается за время $t = 0,12$ с. Ограничение напряжения двигателя достигается путём задания соответствующих значений параметров обучающей последовательности Maximum Plant Input и Minimum Plant Input при выполнении процедуры идентификации объекта управления. Параметры устанавливались равными +1 и -1 соответственно.

Выводы. В данной работе для обеспечения высоких динамических характеристик системы стабилизации основного и вспомогательного вооружения легкобронированной машины предложено использовать нейросетевой регулятор, при построении которого используется метод обобщённого управления с предсказанием. Рассмотрена эффективная реализация обобщенного управления с предсказанием с использованием многослойной прямонаправленной нейронной сети, как нелинейной модели объекта управления.

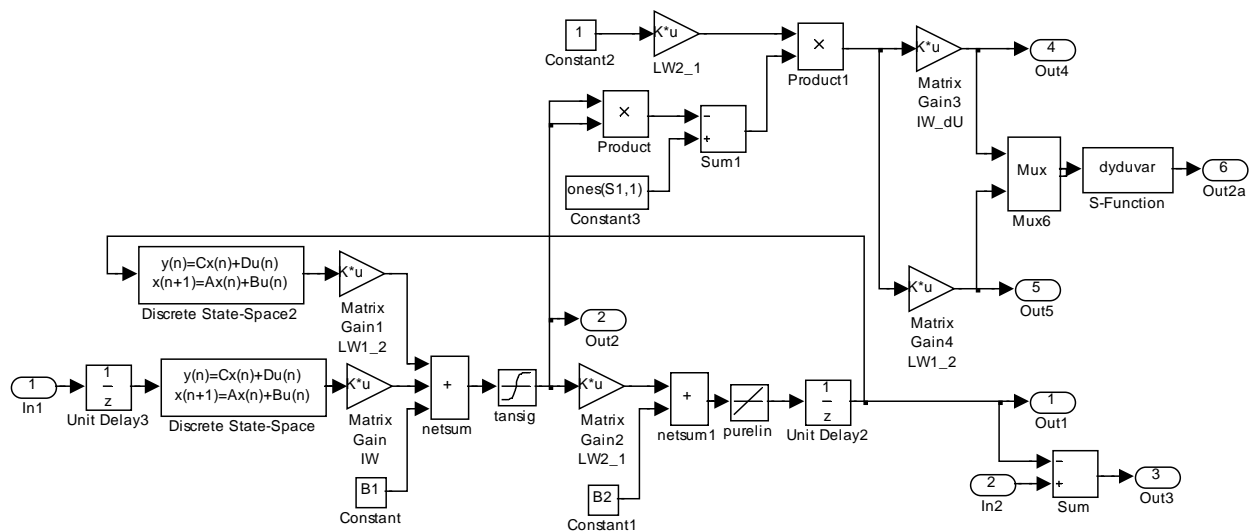


Рис. 7. Схема блока Subsystem системы ptest3sim2

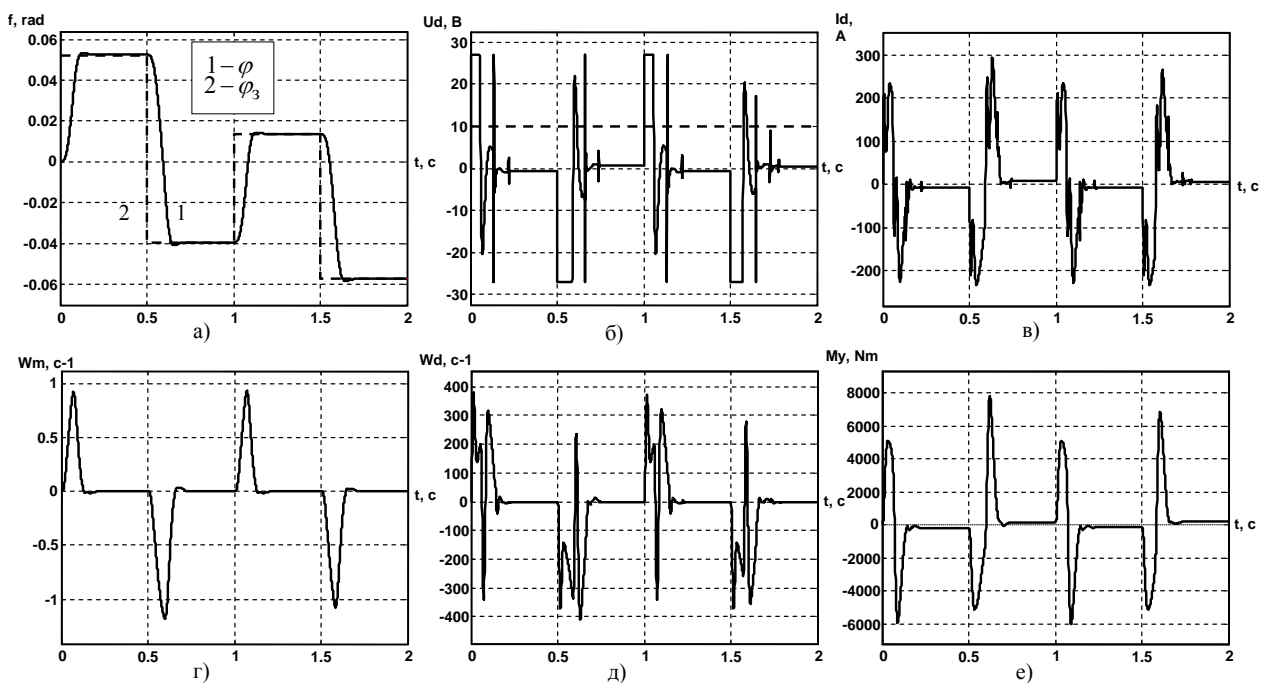


Рис. 8. Графики переходные процессы переменных состояния системы:

- а) – отработки заданного угла φ ; б) – напряжения двигателя U_d ; в) – ток двигателя I_d ;
 г) – скорости механизма ω_M ; д) – скорости двигателя ω_d ; е) – момента упругости M_y

В работе выполнен синтез нейроконтроллера с предсказанием NN Prediction Controller, который реализован в пакете прикладных программ Neural Network Toolbox системы MATLAB. В режиме Simulink разработана система управления с нейроконтроллером и проведено моделирование системы. Анализ переходных процессов основных переменных состояния системы показал, что синтезированный нейроконтроллер обеспечивает высокие динамические характеристики проектируемой системы.

ЛИТЕРАТУРА

[1] Sen P., Hearn G.E., Zhang Y. Adaptive Neural Controller // In Neural Network Systems Techniques and Applications. Hrsg. Leondes C.T. 1998.- Vol.4.- P. 274 - 343.
 [2] Werbos P.J. Backpropagation and neurocontrol: A review

and prospects // Proc. of International Joint Conf. On Neural Networks.- Washington, DC. - 1989. -Vol.1.-P.209-216.
 [3] Park Y.-M., Choi M.S.; Lee Y. An optimal tracking neurocontroller for nonlinear dynamic systems // IEEE Trans. Neural Networks.- 1996.- №7.- P. 1099-1110.
 [4] Jung S.; Hsia T.C. A new neural network control technique for robot manipulators // Robotica.- 1995.- Vol 13, P. 477-484.
 [5] Garcia C.E., Prett D.M., Morari M. Model predictive control: theory and practice-a survey // Automatica.- 1989.- Vol.25.- P.335-348.
 [6] Clarke D. W. Advances in model-based predictive control.- In Advances in Model-Based Predictive Control / D. W. Clarke: Oxford University Press.- 1994.-274 p.

Поступила 09.11.2007