Применение метода ААУ к прогнозированию временных рядов

А.Е. Антипов

Аннотация

Рассматривается возможность применения метода автономного адаптивного управления (ААУ) для прогнозирования временных рядов данных. Предлагается эволюционный способ оптимизации параметров «датчиков», наблюдающих временной процесс, с помощью метода автоматной оптимизаци. Рассмотрен пример прогнозирования биржевого курса ценных бумаг.

Общая идея применения метода ААУ к прогнозированию

Задача прогнозирования временного ряда заключается в следующем. Известен ряд $x_k(t_k)$ ... $x_{-2}(t_{-2})$ $x_{-1}(t_{-1})$ $x_0(t_0)$ - временная последовательность эмпирических данных, представляющих наблюдения величины x в моменты времени t_i. Интервалы между моментами x_i, x_{i+1} равны. Требуется построить последовательность $x_1(t_1)$, $x_2(t_2)$ ... $x_n(t_n)$, наилучшим образом приближающую последовательность $x_1(t_1)$, $x_2(t_2)$ ... $x_n(t_n)$ будущих значений ряда.

Основная аксиома в прогнозировании временных рядов – утверждение о том, что в эмпирических данных есть закономерность. Другими словами, во входных данных существуют похожие друг на друга последовательности значений. Распознав часть такой последовательности, можно с большой вероятностью угадать ее оставшуюся часть.

Структура метода *автономного адаптивного управления* (ААУ) [1] подходит к решению задачи прогнозирования. Теоретически, система ААУ может находить в потоке входных данных образы некоторых последовательностей, запоминать их, а во время прогноза -восстанавливать эти последовательности по их начальным частям. В этом состоит общая идея применения метода ААУ к прогнозированию.

Устройство и принцип действия модели

При построении прогнозирующей системы на базе метода ААУ общая схема системы управления [1] была изменена в соответствии с конкретной задачей. Перечислим основные компоненты системы и опишем их устройство и работу в рассматриваемом случае.

Блок датчиков и исходные данные

В качестве исходных данных выступает выборка $\{x_n\}$ - множество вещественных чисел, элементов временного ряда, соответствующих моментам времени t_n, момент t_1 указывает начало наблюдений.

В системе существует буфер, в котором хранятся последние K элементов ряда. Блок датчиков в каждый момент времени t_n воспринимает очередной элемент ряда x_n и помещает его значение в буфер, сдвигая при этом уже имеющуюся там информацию. Другими словами, блок датчиков передвигает по выборке временное окно шириной K, в каждый момент наблюдая только его содержимое - последовательность

$x_0, x_{-1}, x_{-2}, \ldots x_{-(K-1)}$, где x_0 - последний известный на текущий момент элемент ряда, x_{-1} – предпоследний и.т.д. В начале работы буфер пуст, и первые K тактов работы алгоритма уходят на его заполнение. Пока буфер не будет заполнен, никакие другие действия не предпринимаются.

Введем понятие верхней (x_{max}) и нижней (x_{min}) границ окна:

$$x_{max} = max\,(x_i) + w; \text{ и } x_{min} = min\,(x_i) - w,$$

где w – выбираемая некоторым образом константа, а $i = -(K-1)...0$.

Смысл верхней и нижней границ следующий: это значения, между которыми с некоторым запасом (величина запаса определяется константой w) помещается все текущее содержимое окна. Эти величины в каждый момент времени могут иметь разные значения.

Блок датчиков выбирает из содержимого буфера значимые элементы

$x_{i0}, x_{i1}, \ldots x_{i(k-1)}, i_j \in \{0...K-1\}$, где $k \in \{1...K\}$ – число значимых элементов. Предполагается, что только они повлияют на прогноз, прочие элементы будут проигнорированы. Номера i_j выбираются из некоторых соображений перед началом работы и затем остаются неизменными. Одним из этих элементов в силу изложенных ниже соображений непременно должен являться x_0 – последний по времени (и последний в буфере) элемент ряда.

Затем для каждого из значимых элементов x_{ij} строится разбиение

$$\{T_0...T_{(dj-1)}\}\ ;\ T_k=[\ x_{min}+k(x_{max} - x_{min})/d_j\ ;\ x_{min}+(k+1)(x_{max} - x_{min})/d_j\ ],$$

где d_j – число интервалов для элемента x_{ij}. То есть, для каждого x_{ij} мы разбиваем окно по высоте (по ординате x) на d_j равных интервалов. Числа d_j также выбираются заранее и в процессе работы не изменяются. В общем случае интервалы могут быть разного размера.

Наконец, значение каждого элемента x_{ij} кодируется номером того интервала из разбиения, T_j, в который оно попало. В итоге блок датчиков формирует целочисленный вектор таких номеров $\{p_0...p_j...p_k\}$, в котором закодированы с некоторой погрешностью все значимые элементы буфера. Этот вектор является результатом работы блока датчиков и передается далее блоку формирования образов.

Смысл такого вектора следующий. Все его компоненты кроме p_0, соответствующего элементу ряда x_0, представляют собой «предысторию» p_0, т.е. в векторе содержится информация о том, что за последовательностью $\{p_1...p_j...p_k\}$ в этот раз следовала последовательность $\{p_0\}$. В дальнейшем мы будем называть такие последовательности *образами*.

Поясним работу блока датчиков с помощью рисунка 1.

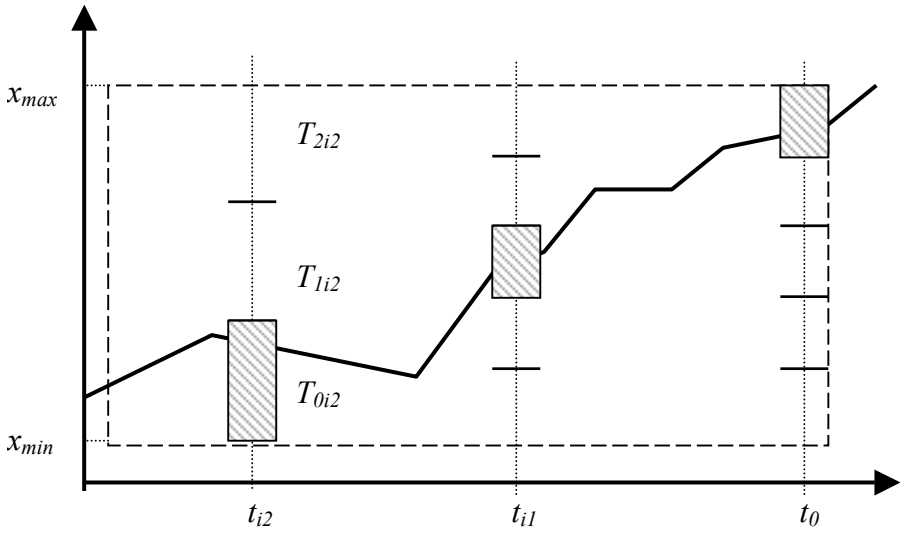

Рис. 1. Представление исходных данных, полученых с помощью «блока датчиков».

На рис. 1 t_0, t_{i1}, t_{i2}, - моменты времени, соответствующие значимым элементам, пунктиром обозначено окно, заштрихованными прямоугольниками - интервалы из разбиений T, в которые попадают значимые элементы. Все прямоугольники в совокупности образуют схематическое представление того, как блок датчиков «видит» поступающую к нему на вход информацию.

Блок формирования и распознавания образов

Блок формирования и распознавания образов в каждый момент времени получает на вход вектор $\{p_0...p_j...p_k\}$, сформированный блоком датчиков и представляющий собой закодированное представление пары образов «причина → следствие» (или «хвост буфера → голова буфера»). Так как природа исследуемого нами ряда, вообще говоря, неизвестна, либо недостоверна, и никаких априорных сведений о возможных встречающихся в нем образах мы не имеем, было принято решение в качестве образов использовать все возможные входные векторы. Таким образом, общее число элементов базы знаний будет равно произведению $d_0\ d_1\ ...\ d_k$, где d_j – число интервалов в разбиении T_j, соответствующем элементу x_{ij}.

Для перевода входного вектора в номер, однозначно определяющий закодированную в нем пару и, соответственно, определяющий один и только один элемент базы знаний, применяется хэш-функция вида $p_0d_1d_2...d_k + p_1d_2...d_k + ... + p_k$. Легко убедиться, что эта функция действительно однозначно переводит любой вектор $\{p_0...p_j...p_k\}$ в целое число – фактически, она преобразует это число в привычный вид из записи в позиционной системе счисления с «переменным основанием», которой и является вектор $\{p_0...p_j...p_k\}$. Номер распознанного образа является результатом действия блока ФРО и поступает на вход блока формирования базы знаний.

Блок формирования базы знаний

База знаний в нашем случае устроена просто – это целочисленный массив размером в максимальное количество образов, которые способен распознать блок ФРО. Получая на вход номер распознанного образа, блок формирования базы знаний увеличивает значение соответствующей номеру ячейки массива. Если это значение превышает некоторый порог M, то отвечающая номеру пара «причина → следствие» признается

устойчивой, и в дальнейшем считается, что такая причина статистически связана с таким следствием. У каждого следствия может быть несколько разных причин, и наоборот.

Блок оценки состояния объекта управления

Этот блок отсутствует за ненадобностью. Однако имеется «эмоциональная оценка» входных образов: чем чаще образ встречается во входном потоке, тем он «лучше», и тем большее предпочтение будет отдано ему при выборе действия.

Блок определения времени принятия решения

Этот блок вырожденный – задержка между восприятием образа и принятием решения всегда равна 0.

Блок принятия решений

Перед блоком принятия решений стоит следующая задача: зная образ, воспринимаемый системой в данный момент, на основании хранящейся в базе знаний информации определить, какой образ с наибольшей вероятностью последует за ним в следующий момент времени. Это делается следующим способом: за образ «причины» принимается вектор $\{p_1 \ldots p_j \ldots p_k\}$. Затем в базе знаний находятся все устойчивые пары, соответствующие такой причине, среди них выбирается пара с наибольшим весом (т.е. с наибольшим значением в соответствующей ячейке базы знаний), и образ соответствующего этой паре «следствия» p_0 провозглашается искомым. Этот образ передается исполняющему органу.

Если блок принятия решений не может найти ни одной такой пары, он возвращает образ самого последнего известного элемента ряда (предполагая, что никаких изменений не предвидится).

Исполняющий орган

Задача исполняющего органа состоит в том, чтобы на основании переданного ему образа действия p_0 получить вещественное число – ожидаемый в следующий момент элемент ряда. В качестве такого числа берется середина интервала из разбиения T, соответствующего p_0.

Оптимизация. Эксперименты и результаты

Цель, постановка и решение задачи оптимизации

Первоначальные эксперименты производились на достаточно больших выборках (несколько тысяч значений), в качестве исходных данных брался курс доллара по отношению к рублю за последние несколько лет.

Проводился прогноз двух типов. Прогноз 1-го типа - кратковременный, когда на основании достоверных данных предсказывалась ситуация только на один шаг вперед. Прогноз 2-го типа – долговременный: вначале на основании достоверных данных $\{x_0 \ldots x_{-K}\}$ предсказывалось следующее значение x_1 , затем, на основании данных $\{ x_1 , x_0 \ldots x_{-(K-1)}\}$ - значение x_2 , и.т.д.

В ходе экспериментов выяснилось, что результаты прогноза очень сильно зависят от подбора количества интервалов d_{ij} и значимых элементов $x_{i0}, \; x_{i1}, \ldots \; x_{i(k-1)}$. В зависимости от выбранных значений качество прогноза колебалось от отклонения в 1.2 % при прогнозе на несколько сотен дней (на относительно ровных участках) до полного

вырождения прогноза в параллельную оси t прямую после первых нескольких итераций. Стало ясно, что метод необходимо оптимизировать для каждой конкретной выборки.

Постановка задачи оптимизации выглядит следующим образом: для заданного количества значимых элементов k требуется подобрать их номера i_j и количество интервалов d_{ij} так, чтобы минимизировать среднее отклонение прогноза от реальных значений $\Delta P(i_0 \ldots i_j \ldots i_k\ ,\ d_0 \ldots d_j \ldots d_k\ ,\ \{x_n\})$, где $\{x_n\},\ n \in [0,N-1]$ – выборка исходных значений. Очевидно, данная функция является сеточной. Введены ограничения на максимальное количество интервалов d_{max} и на максимальную ширину окна K. Эти ограничения вызваны требованиями реализации: количество памяти, нужной алгоритму, не должно превышать разумных пределов.

Для решения задачи был выбран метод автоматной оптимизации *[2]*, предложенный А.Н. Рапопортом. Выбор обусловлен тем, что данный метод не требует от функции явного вида, и потому удобен для минимизации сеточных функций.

При оптимизации использовался коллектив стохастических автоматов с глубиной памяти от 2 до 6 и вероятностью принятия противоположного решения 0.1. В случае попытки выйти за границы штраф становился положительным, независимо от изменения минимизируемой функции. Все значения подобраны экспериментальным путем.

Метод оптимизировался на последовательности значений курса доллара к рублю по данным ЦБ РФ за 1995 - 1998 годы. В качестве отклонения ΔP была выбрана функция $\sum_{k \in [K,N-1-K]} ((\sum_{n \in [i,I+K-1]} |(x_n - p_{n,K}) / x_n|)/(K))/(N-1-K)$ – среднее относительное отклонение от реальности всевозможных прогнозов длины K на выборке. Число K в случае реального использования вытекает из требуемого горизонта прогноза, в нашем случае оно колебалось от 30 до100.

Вначале на оптимизируемые параметры были наложены следующие ограничения:

1) $2 \le d_j \le d_{max}$ - число интервалов в разбиении не должно превышать заданного и не должно быть меньше двух. Первое нужно для избежания неограниченного роста базы знаний, а при разбиении из одного интервала данный узел не даст нам никакой полезной информации.

2) $0 \le i_j \le i_{max}$ – значимые узлы не должны попадать в будущее (так как там еще ничего не известно) и не должны уходить в прошлое более чем на i_{max} периодов, чтобы избежать неограниченного увеличения буфера.

Оптимизация с такими параметрами привела к неожиданным результатам: все значимые узлы собрались в точке 0. Из этого следовал полностью вырожденный прогноз – прямая, параллельная оси t. Введение следующего ограничения : $| i_j - i_k | < \Delta i_{max}\ ;\ i \neq j$ (расстояние между значимыми узлами не должно быть меньше заданного) существенно не повлияло на качество прогноза. Узлы все равно распределялись так, чтобы прогноз оказывался вырожденным, т.е. чтобы встречающиеся во входном потоке образы как можно реже распознавались.

Причиной такого поведения системы, видимо, является то, что тактика выжидания при «угадывании» следующего значения – не самая худшая из возможных. В точке, соответствующей этой тактике, метод автоматной оптимизации находит локальный минимум и зацикливается на нем.

Поэтому было введено еще одно ограничение. Всякий раз, когда в процессе оптимизации система давала вырожденный прогноз (т.е. не могла распознать текущий образ как устойчивый), коллектив оптимизирующих автоматов наказывался. Это

положительно повлияло на результат оптимизации и в ряде случаев привело к достаточно высокому качеству прогноза.

Эксперименты

Проведены эксперименты на реальных входных данных: значениях курса доллара к рублю за 1993 – 2002 г, значениях индекса S&P500, значениях процентной ставки ЦБ РФ. Приведем несколько примеров, иллюстрирующих работу модели на разных выборках.

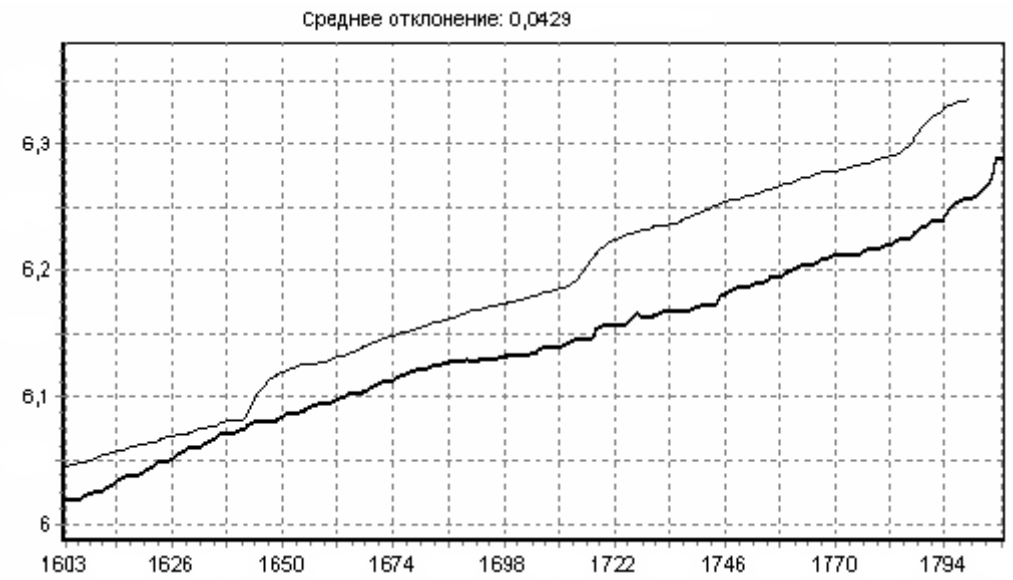

Рис. 2. Прогноз (тонкая линия) курса доллара (жирная линия) на 300 периодов (дней) вперед. Отклонение от реальных значений – 0.04 (примерно 0.6%). Заметна периодичность (“горбы” на графике через равные промежутки времени), выявленная системой ААУ во входных данных.

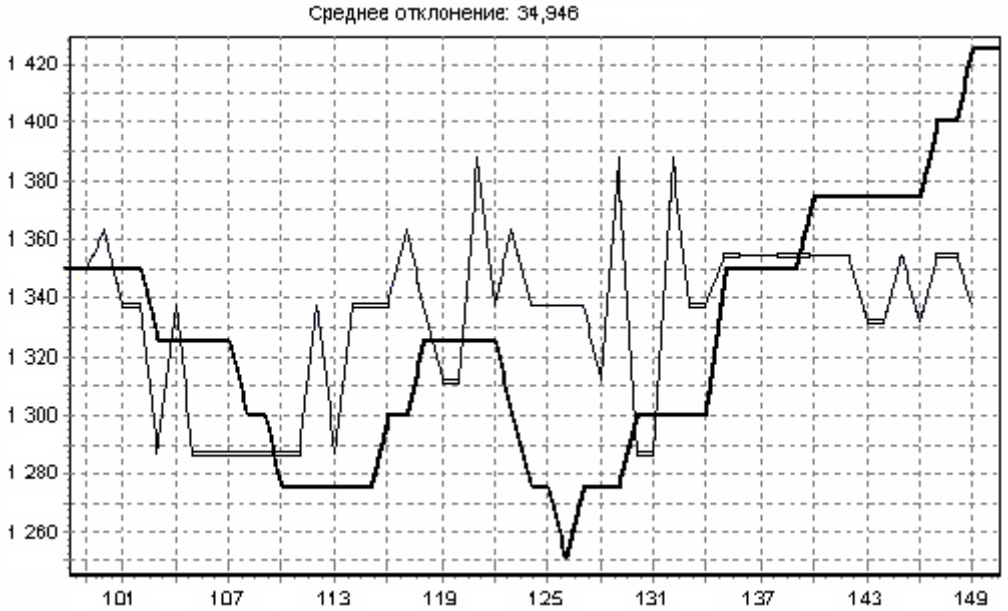

Рис.3. Прогноз (тонкая линия) курса S&P500 (жирная линия) на 50 периодов (дней) вперед. Отклонение от реальных значений – 35 (примерно 3%). Качество прогноза хуже, чем в первом случае, так как входные данные более хаотичны. Система ААУ работает в условиях нехватки информации. Участки с двойной линией на графике прогноза – промежутки, на которых имеющихся в базе знаний сведений недостаточно для принятия решения.

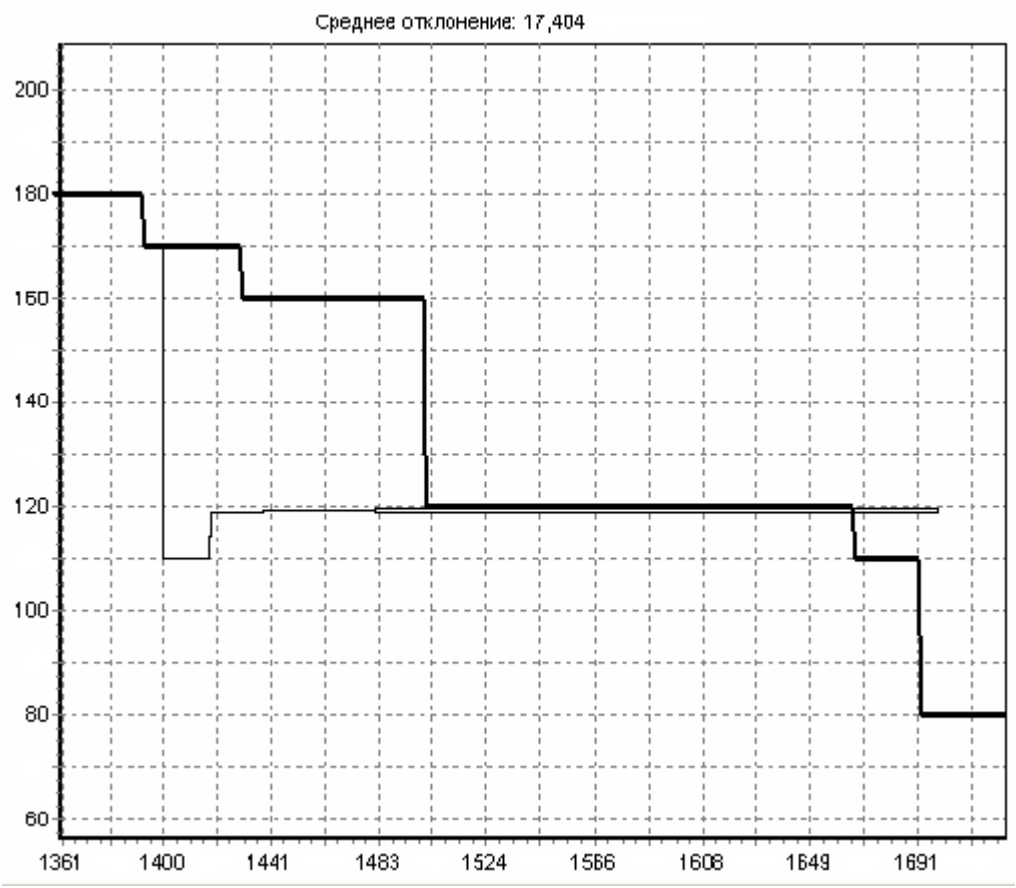

Рис.3. Прогноз (тонкая линия) процентной ставки ЦБ РФ (толстая линия) на 300 периодов (дней) вперед. Отклонение от реальных значений – 17 (примерно 14%). Система ААУ не может найти достаточного количества устойчивых образов во входном потоке, и поэтому на протяжении последних двух третей прогнозируемого промежутка не способна принять решение (двойная линия).

Из приведенных примеров видно, что качество работы модели очень сильно зависит от исходной выборки. Чем глаже прогнозируемая функция, тем выше точность прогноза. Дать оценку погрешности метода в общем случае, на основании которой можно было бы сопоставить данный метод с другими и сделать выводы об их сравнительном качестве и области применимости, авторы пока не в состоянии. Такая оценка должна представлять собой зависимость средней ошибки прогнозирования от некоторых характеристик исходного ряда. На данном этапе непонятно, какие именно характеристики следует учитывать.

Выводы

К преимуществам прогнозирования с помощью метода ААУ можно отнести:

- Отсутствие необходимости описывать математическую модель исследуемого процесса.
- Возможность обрабатывать информацию поточно, без переучивания после появления каждого нового значения. Другими словами, в системе ААУ реализуется принцип самообучения в процессе управления.
- Гибкость, способность отбрасывать некогда найденные, но уже долго не встречавшиеся закономерности.
- Достаточно высокое качество прогноза (при горизонте от 50 до 300 периодов на тестовых выборках достигнуто отклонение 0.6 – 3%).

Недостатками являются:

- Устойчивая работа только на больших (сотни, тысячи значений) выборках.
- Большой объем необходимых вычислений. По вычислительной сложности метод сравним с нейросетевыми алгоритмами.
- Невозможность представления алгоритма в виде явно заданной функции, что затрудняет классические методы его исследования.

Варианты развития.

Возможны несколько путей развития описанного способа предсказания временных рядов:

1) *Введение эффекта «забывания».* В случае изменения свойств прогнозируемого ряда образы «старых» закономерностей, сохранившиеся в базе знаний, будут мешать прогнозу. Возможно, следовало бы в соответствии с некоторым законом уменьшать вес пары образов, долгое время не наблюдавшейся во входном потоке.
2) *Усовершенствование исполнительного органа.* Зная интервал, в котором ожидается появление следующего значения, можно не брать это значение из середины интервала, а вычислять более точно на основании сведений о поведении ряда за последние несколько тактов.
3) *Усовершенствование блока датчиков.*
   a) Разбиение T для каждого из значимых узлов вовсе не обязательно должно содержать интервалы равной длины. Может оказаться, что введение разбиений с неравными интервалами, выбираемыми из неизвестных пока соображений, положительно скажется на качестве прогноза.
   b) Количество значимых узлов в данной работе подбиралось экспериментально. Необходимо более детально исследовать эту проблему.
   c) Высота окна также подбиралась экспериментально, требуется теоретически обосновать ее выбор и научиться определять оптимальную высоту.
   d) Образы могут кодироваться отличным от описанного способом. Например, можно представлять их в виде набора коэффициентов полинома заданной

степени, приближающего соответствующую образу кривую. Не исключено, что такое кодирование окажется более эффективным.

e) Блок датчиков может воспринимать элементы ряда, полученного из исходного каким-либо способом. Например, взятием знака от первой производной. Или, в общем случае, вычислением тренда (тенденции ряда). Во многих экономических задачах важно качественное, а не количественное предсказание, поэтому такая замена уместна. К тому же она позволяет сократить размерность базы знаний, так как диапазон наблюдаемых значений сокращается. В этой области проведено несколько экспериментов, однако об однозначных результатах говорить еще рано.

Список использованной литературы

1) Жданов А. А. Накопление и использование информации при управлении в условиях неопределенности. Сб. н. тр. ИФТП РАН "Информационная технология и численные методы анализа распределенных систем." М.1992, 112-133.

2) Жданов А. А. Метод автономного адаптивного управления // Известия Академии Наук. Теория и системы управления, 1999, № 5, с. 127-134.

3) Рапопорт А.Н. Автоматные модели поисковой оптимизации и управления. Монография. Киров, 1999.