

Возможности использования технологии детерминированного хаоса в системах автономного адаптивного управления¹

А. А. Жданов, А.Е. Устюжанин

Введение

Одной из принципиальных черт человеческой памяти является способность человека запоминать как единый объект образ, состоящий из множества упорядоченных деталей, особенно часто упорядоченных во времени (везде ниже будем иметь в виду упорядоченность именно во времени). Для того, чтобы распознать такой образ, вспомнить и воспроизвести его в памяти, человеку бывает достаточно увидеть лишь небольшой фрагмент, относящийся к произвольной части такого образа, остальные упорядоченные фрагменты образа вспоминаются уже по ассоциации. Примерами могут служить запоминание мелодии, стихотворения, длинной дороги. По-видимому, такими свойствами обладает память не только человека, но и память других естественных организмов. Употребленное здесь понятие ассоциативности заставляет сравнить такую память с так называемой ассоциативной памятью, используемой в технических системах обработки информации.

Ассоциативным подходом к использованию памяти в технических системах обработки информации называется подход, когда поиск информации происходит по части ее содержания. Ассоциативный подход используется, например, в ЭВМ для реализации сверхоперативной памяти и при работе с базами данных. Ассоциативными свойствами обладают также некоторые нейронные сети, например, нейронные сети Хопфилда и Хэмминга [1, 2]. При реализации ассоциативной памяти в нейронных сетях стремятся сделать так, чтобы при возбуждении некоторых нейронов возбуждался тот ансамбль нейронов, который включает первоначально возбужденные нейроны, а также и другие нейронные ансамбли, ассоциированные с первым, при этом предусматриваются средства для торможения процесса ассоциативного возбуждения в случае его лавинообразного расширения. В последнее время появился также новый подход, основанный на применении теории детерминированного хаоса [3-20], который подробнее будет рассмотрен ниже.

Однако использованное в указанных системах понимание ассоциативной памяти не вполне совпадает с тем феноменом естественной памяти, о котором мы говорим, поскольку в ассоциативной памяти упорядоченность запоминаемой информации во времени не является определяющим требованием. Соответственно, способ обращения к ассоциативной памяти не связан обязательно с упорядоченностью содержания данных. Например, в указанных реализациях ассоциативной памяти на нейронных сетях нейронные ансамбли возбуждаются в порядке, зависящем от ряда случайных причин. Напротив, в естественной памяти, как можно судить по собственным ощущениям, информация запоминается с сохранением порядка ее поступления (слова стихотворения, звуки мелодии), объединяясь при этом некоторым смысловым содержанием (технология некоторой операции, речь некоторого человека) и некоторым идентифицирующим признаком (название стихотворения, мелодии); наиболее хорошо запоминаются повторяющиеся последовательности данных (знакомая дорога), при этом в них запоминаются именно общие места; информация вспоминается в той последовательности, в которой она запоминалась (Вам с большим трудом удастся воспроизвести слова знакомого стихотворения в обратном порядке).

Однако наиболее важное отличие мы усматриваем в следующем. Память современного компьютера оптимизируется по различным критериям, отражающим возможные требования пользователей и исполнителя пользовательских и системных программ. Напротив, память всякого живого организма оптимизирована с точки зрения вполне определенных целевых функций, единых для всех нервных систем. Этими целевыми функциями, по нашему убеждению, являются выживание и накопление знаний.

Для того чтобы выжить, улучшить свое текущее состояние, нервная система имеет только одно средство – она должна воздействовать на окружающую ее среду, рассчитывая на определенные прогнозируемые нервной системой обратные реакции среды. Такой прогноз основывается на тех эмпирических и выводных знаниях, которые добыты данной нервной системой и хранятся в ее памяти. Таким образом, знания нервной системы должны отражать функциональные свойства среды, окружающей ее, т.е. приносить информацию об этих свойствах. Уточним, что нервную систему окружает тело организма и среда, в которую погружен организм, их совокупность и будем называть ниже просто средой. Информация о функциональных свойствах любого объекта, в данном случае среды, должна иметь вполне определенную структуру, именно, указывать, каким воздействием можно перевести данный объект из одного возможного его состояния в другое возможное его состояние. Другими словами, элементарное знание должно иметь вид последовательности импликаций, «начальное состояние объекта в момент t_1 » → «воздействие на объект в момент t_2 » → «возможные результирующие состояния объекта в момент t_3 ».

¹ Работа выполнена при поддержке РФФИ, грант № 0001-00372

Такая импликация отражает причинно-следственную связь явлений в нашей среде обитания, и самым тесным образом связана со временем, так как причина предшествует следствию именно во времени. Поэтому работа с данными, упорядоченными во времени, крайне важна для живых организмов. Анализ событий, последовательно происходящих во времени, является основой выявления причинно-следственных связей явлений в нашей среде обитания. Чтобы обеспечить выживание организмов в этом мире, их нервные системы должны иметь возможность запоминать временные последовательности данных о наблюдаемых явлениях и своих воздействиях, анализировать их, выявляя в них регулярности, и использовать выявленные регулярности для принятия решений на основании прогноза следствий альтернативных вариантов своих возможных действий.

Из сказанного следует, что известные способы реализации ассоциативной памяти не вполне подходят для моделирования указанных свойств естественной памяти при моделировании систем управления, имитирующих нервные системы. Особенно это касается таких автоматических адаптивных систем управления, которые имитируют не отдельные свойства нервных систем, но их свойства в целом. Это управляющие системы, которые должны автоматически приспособливаться к жизни в среде с заранее плохо известными им свойствами, это системы, в которых предусмотрены указанные целевые функции выживания и накопления знаний. Именно такого рода системы управления мы и будем рассматривать ниже.

Мы полагаем, что для воспроизведения памяти с описанными выше свойствами, в автоматических адаптивных системах управления должны автоматически решаться следующие задачи.

а) Задача автоматической классификации (формирования образов) пространственно-временных объектов, т.е. автоматическое обнаружение в пространстве признаков таких упорядоченных множеств пространственно-временных объектов, которые можно ассоциировать в классы - образы в соответствии с некоторыми заданными правилами;

б) Задача сохранения данных о сформированных классах - образах в некоторой "памяти образов";

в) Задача автоматического ассоциативного распознавания таких образов, т.е. выделение в памяти всех упорядоченных объектов, составляющих данный образ, при наблюдении только части принадлежащих ему объектов.

Настоящая работа представляет предварительные результаты исследования возможностей построения памяти с указанного рода ассоциативными свойствами на основе технологии детерминированного хаоса.

При исследованиях человеческого мозга была выдвинута гипотеза, что принципы работы человеческого мозга схожи с принципами работы технических систем, основанных на идеях детерминированного хаоса. Такие системы в

некоторых случаях заметно повышают эффективность обработки информации, и, одновременно, удобны для программной реализации. Тем самым, идеи детерминированного хаоса имеют, с одной стороны, глубокое бионическое обоснование, с другой стороны – могут дать важные практически полезные результаты.

Оба этих свойства детерминированного хаоса являются привлекательными для систем *автономного адаптивного управления* (ААУ), разрабатываемых авторами настоящей работы [22-37]. Идея систем ААУ также имеет бионическое основание, поскольку в системах ААУ мы моделируем нервные системы. В управляющей системе ААУ отдельными ее подсистемами взаимосогласованно решаются задачи автоматической классификации, распознавания образов, представления знаний, принятия решений и некоторые другие. Основным полезным свойством систем ААУ является их способность осуществлять адаптивное управление объектами, для которых трудно составить точную математическую модель, на основании которой можно было бы применить какой-либо из традиционных методов управления.

Проблема работы с образами протяженных упорядоченных во времени пространственно-временных объектов и организации ассоциативной памяти является актуальной для теории систем ААУ. Практически реализованные до настоящего момента системы ААУ позволяли решать задачу формирования и распознавания образов (ФРО) на данных по предыстории процесса управления длиной всего несколько тактов дискретного времени с прогнозированием всего на один-два такта вперед. При помощи технологий детерминированного хаоса удалось найти эффективный подход к расширению анализируемых и прогнозируемых временных диапазонов процесса управления. Были рассмотрены различные варианты построения памяти управляющей системы на основе детерминированного хаоса. Для исследования разработанной системы и проведения компьютерных экспериментов был построен программный комплекс, позволяющий собирать систему ААУ из модулей, реализующих ее функциональные подсистемы на базе различных подходов.

В результате проведенного математического моделирования была показана принципиальная возможность использования технологии детерминированного хаоса для организации таких подсистем системы ААУ, как аппарат формирования и распознавания образов, база знаний и аппарат принятия решений. Данная технология позволяет управляющей системе работать с образами очень протяженных пространственно-временных явлений. Свойства системы были продемонстрированы на модельном примере, представляющем собой управляющую систему транспортного робота, движущегося по лабиринту, имеющему набор повторяющихся фрагментов. Настоящая работа представляет первые результаты исследования возможности применения технологии детерминированного хаоса для построения некоторых подсистем системы ААУ.

1. Имитационный метод автономного адаптивного управления

Коротко представим основные положения разрабатываемой в ИСП РАН концепции адаптивных систем управления, называемых нами системами “автономного адаптивного управления” (ААУ) [22-37]. Системы ААУ моделируют некоторые черты естественных нервных систем, поэтому проводятся аналогии между определенными подсистемами системы ААУ и биологическими объектами.

Так под *объектом управления* (ОУ) понимается модель организма, под *управляющей системой* (УС) – модель нервной системы, под *системой* – совокупность УС, ОУ и среды.

При построении УС принимаются следующие четыре исходные условия, характерные для нервных систем.

“Условие автономности”, под которым понимается то обстоятельство, что УС является подсистемой ОУ, т.е. УС находится на борту ОУ и осуществляет управление на основе знаний, добываемых самостоятельно, взаимодействуя со своим окружением посредством *блока датчиков* (БД) и *исполнительного органа* (ИО), входящих в состав ОУ и не входящих в УС.

“Условие дискретности”, которое отражает дискретность структуры УС и принципа ее функционирования. Так, дискретными являются конечные множества нейронов, связей, датчиков, исполнителей, нервных импульсов, образов, моментов времени. Однако при этом возможен непрерывный характер изменения некоторых параметров УС, таких как размеры отдельных синапсов, частотные характеристики импульсных последовательностей, и некоторых других параметров. Однако элементарной дискретной единицей информации в УС полагается образ.

“Условие максимальной начальной приспособленности” отражает наличие приспособленности ОУ и УС к усредненным условиям жизни ОУ в данной среде в результате действия механизмов типа естественного эволюционного отбора. Такие механизмы отбора определяют типы датчиков и исполнителей, классы потенциально возможных в данной системе образов, оценки качества важнейших для ОУ образов, и т.п. При синтезе ОУ и УС процесс естественного отбора может быть заменен, например, оптимизацией системы на основе использования априорной информации.

“Условие минимума исходных знаний” отражает наличие информационных пространств, которые должны быть заполнены знаниями, найденными УС в процессе функционирования в реальной системе. Это условие соответствует наличию неопределенности свойств системы, максимальной для УС в момент начала ее функционирования. Адаптивностью УС мы называем способность УС накапливать знания, которые используются ею для управления. При этом качество управления возрастает, поскольку накопление знаний снижает

неопределенность свойств модели окружающей среды, которую в совокупности представляют собой накопленные знания.

Принимаются следующие целевые функции УС: а) выживание ОУ и б) накопление знаний. Эти две целевые функции взаимосвязаны в том отношении, что достижение одной из них повышает вероятность достижения другой.

Из сказанного вынужденно следует представление о системе (рис. 1), в котором можно видеть ОУ, погруженным в среду, и УС, погруженной в ОУ. Пусть каждый из этих макрообъектов оказывает воздействия на систему через свой выход и воспринимает воздействия системы через свой вход.

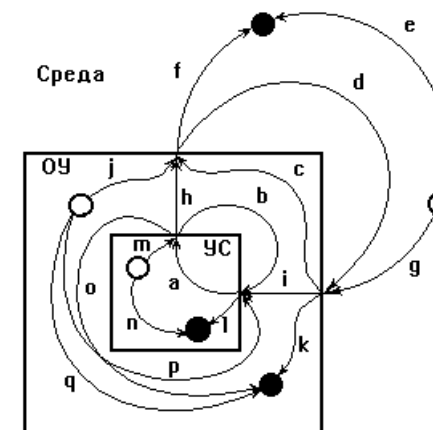


Рис. 1.

Кроме того, в системе важно наличие источников случайных воздействий (белые кружки на рисунке - “истоки”) и мест поглощения воздействий (черные кружки на рисунке - “стоки”). Такая схема позволяет увидеть все возможные маршруты распространения воздействий в системе (помеченные буквами стрелки на рисунке) [22, 36].

Очевидно, что для достижения своих целевых функций, УС должна найти те из воздействий h , которые образуют цикл “управляемого взаимодействия (УВ)” $h \rightarrow d \rightarrow i \rightarrow a \rightarrow h \rightarrow \dots$, зафиксировать информационное отображение цикла УВ в своей памяти – *базе знаний* (БЗ), оценить полезность тех или иных элементов знаний и использовать эти знания для выживания, одновременно стремясь к получению и накоплению новых знаний. Здесь a это информационный процесс в УС, h – процесс преобразования информационных команд в физические воздействия, d это различные процессы в окружающей среде, i – процесс преобразования части входных воздействий в информационные входные сигналы для УС. Наличие в системе истоков и стоков вносит в УВ многочисленные случайные компоненты и приводит к потерям информации.

Накапливая знания, УС уменьшает неопределенность своей БЗ, отображающей УВ.

Подобное макроописание позволяет понять задачу, которую должна решать УС. В общем виде можно видеть следующую ее постановку. На заданное множество входных полюсов УС $v_1, v_2, \dots, v_i, \dots, v_n$ (например, это n бинарных выходов БД) поступает входной поток информации. Пусть, например, это будет эквидистантная во времени последовательность двоичных векторов $V_a(t_1), V_b(t_2), \dots, V_c(t_k), \dots, V_d(t)$, где t – текущий момент времени. Если последовательность не эквидистантная, то должны быть средства синхронизации потоков данных. Согласно рис. 1, семантически каждый вектор $V_c(t_k)$ может содержать информацию, пришедшую ко входу УС по маршрутам i, b и/или p . Здесь i это информация, поступившая с датчиков, b – информация, отражающая действия, совершенные УС, p это помехи от истоков в ОУ (информация от i и b также содержит случайные компоненты). Информация, поступающая по маршрутам i, b и p , может отображаться на определенные подмножества компонент вектора $V_c(t_k)$.

Первая задача, которую должна и может решать УС, состоит в нахождении неслучайных регулярных пространственно-временных комбинаций компонент в потоке входных векторов $V_a(t_1), V_b(t_2), \dots, V_c(t_k), \dots, V_d(t)$. Произвольные примеры трех таких образов показаны на рис. 2.

Если УС обнаруживает наличие такой регулярности, то УС должна:

- запомнить информацию о ней как самостоятельный объект – образ;
- уметь распознавать прообраз этого образа во входной информации при его последующих появлениях, где под прообразом будем понимать регулярность, которая привела к формированию образа;
- при накоплении достаточной статистической надежности – уметь распознавать образ протяженного во времени прообраза раньше, чем закончится его наблюдение на входных полюсах и при наличии помех;
- оценить соответствие этого образа целевой функции выживания ОУ.

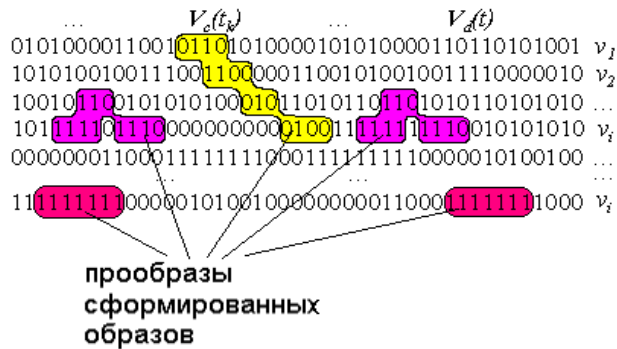


Рис. 2.

Пути к решению последней задачи мы видим в организации в УС специальной подсистемы, которая эквивалентна *аппарату эмоций* в живом организме. Это многофункциональная подсистема [25], основная функция которой – соотносить отвлеченные информационные сигналы, например, сигналы о распознавании образов, отображающих текущие состояния ОУ, с объективной полезностью или опасностью их прообразов для выживания ОУ. В результате сформированные образы наделяются своими качественными оценками. Оценки базовых образов (изначально заложенные в УС образы наиболее важных для выживания состояний) могут быть найдены на основе анализа эмпирических данных и априорной информации. Оценки для образов, формируемых в процессе функционирования ОУ, могут быть найдены УС опытным путем посредством корреляционного анализа фактов распознавания этих образов и значений текущей интегральной оценки состояния ОУ, которую обозначим S^t . В свою очередь, текущее значение оценки S^t зависит от оценок всех распознанных в текущий момент образов. Здесь обратим внимание на то, что целевая функция выживания реализуется через цель максимизации оценок S^t . Это обеспечивает внутреннюю активность УС, тем самым, работая и на вторую целевую функцию – накопление знаний. В процессе управления текущее значение оценки S^t определяет также целесообразную глубину просмотра базы знаний при принятии решений в текущей ситуации. Пусть УС сформировала некоторый образ, запомнила его в памяти сформированных образов в виде объекта, который обозначим O_k , и в некоторый момент t распознает его, что обозначим как $O_k^t = 1$ (если сформированный образ O_k не распознан в момент t , то полагаем $O_k^t = 0$). Пусть, например, прообразом данного образа является некоторый пространственно-временной процесс длительностью T тактов, но УС распознала образ уже через T' тактов после его начала t_{ks} , и $T' < T$. Если семантически прообраз данного образа в векторах $V_a(t_{ks}), V_b(t_{ks+1}), \dots, V_c(t_{ks+T'}), \dots, V_d(t_{ks+T})$ представлял собой отображение информации, пришедшей на вход УС по маршруту i (см. рис. 1), то это сравнительно простой случай, интересный, в основном, тем, что УС имеет прогноз, в частности, изменения оценки своего состояния S^t к моменту, когда завершится наблюдение всего прообраза. Одновременно может быть распознано некоторое множество образов. Однако ситуация становится более интересной, если в прообразе распознанного образа O_k имел место хотя бы один вектор $V(t)$, отражающий информацию, пришедшую к УС по маршруту b , т.е. в прообраз входило некоторое действие, совершенное УС (например, в нервной системе это сигнал от рецепторов о том, что была сокращена такая-то мышца в результате соответствующей команды). Обозначим такое действие символом Y_i . Тем самым прообраз образа O_k (рис. 3а) можно разбить на три последовательных во времени составляющих: 1) информация о ситуации X_a , 2) информация о действии Y_i , 3) информация о ситуации X_b . Поскольку, по определению, прообраз сформированного образа – явление не случайное, то данную тройку (X_a, Y_i, X_b)

можно интерпретировать как импликацию "условие → действие → следствие". Очевидно, что в данной ситуации могло быть распознано два или более образов O_k (рис. 3b), имеющих одинаковое условие (ситуацию X_a), разные возможные альтернативные действия (действие Y_i) и, возможно, разные следствия (ситуацию X_b).

Итак, если в некоторый момент а) УС одновременно распознает несколько временных образов, при этом б) распознавание наступило раньше окончания наблюдения составляющей X_a прообраза, в) в составе прообразов ожидаются альтернативные действия Y_1, Y_2, \dots , ведущие к альтернативным следствиям X_b, X_c, \dots , г) распознанные образы имеют свои оценки, то появляется возможность принятия решения, на основании сравнения ожидаемых приращений интегральной оценки качества состояния S^t для альтернативных действий [23, 25].

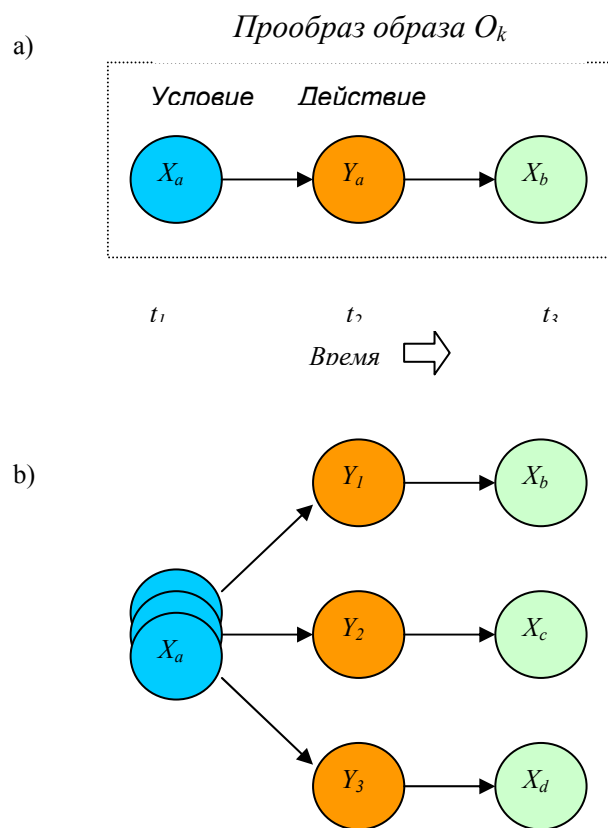


Рис. 3.

Если при распознавании образа O_k вида "условие → действие → следствие", его «следствие» соответствует «условиям» других образов такого же вида "условие → действие → следствие", записанным в БЗ, то прогнозируемая причинно-следственная цепочка образов и действий продолжается на несколько шагов в будущее, позволяя УС глубоко планировать свои действия. Для этого, конечно, УС должна обладать специальными языковыми средствами манипулирования с такими образами и действиями, а у нас появляются основания говорить об интеллекте этой УС [27].

Поскольку постоянно идет формирование новых образов, то возможна ситуация, когда альтернативные действия, приносящие ранее одинаковый ожидаемый конечный результат - распознавание одинаковых образов, начнут различаться в отношении новых сформированных образов, т.е. – более тонких последствий выбираемого действия. Учет этих различий делает действия УС со временем все более точными, а их следствия – все более точно предсказуемыми. Это одна из возможностей саморазвития УС.

Рассмотренная интеллектуальная управляющая система обладает следующими основными свойствами.

1. УС обладает внутренней активностью, направленной на цели выживания, улучшения текущего состояния и накопления знаний.
2. Накопление знаний в УС повышает вероятность ее выживания.
3. УС обладает свойствами адаптивности и саморазвития. Именно, формируются и оцениваются новые образы, формируются и уточняются новые действия, увеличивается глубина прогнозирования следствий альтернативных вариантов действий и др.
4. Знания УС сильно зависят от ее индивидуального опыта, в том числе от случайных событий, т.е. УС обладает своей индивидуальностью.

Система с подобными свойствами не соответствует общепринятому пониманию и назначению систем искусственного интеллекта (ИИ), к которым сегодня относят экспертные системы, распознающие системы, нейросети, лингвистические процессоры и робототехнику. Поэтому предлагается выделить системы с подобными свойствами в отдельное направление "системы автономного искусственного интеллекта" (АИИ) [27]. Если основное назначение систем ИИ - быть исполнителями заданий, поступивших извне от пользователя, то назначение систем АИИ – выжить и накопить знания, подчиняясь относительно независимому от внешних управляющих воздействий внутреннему целеполаганию. Помимо теоретического интереса, прагматическое значение систем АИИ, имитирующих естественные организмы, может состоять в их способности к выживанию, работе и накоплению знаний в плохоформализуемых условиях, требующих адаптации и саморазвития.

Как видно из сделанного качественного описания рассматриваемых систем, их реализация сопряжена с необходимостью решения следующих непростых задач:

1) задачи формирования и распознавания образов (ФРО) в потоке входной многоканальной информации, 2) задачи построения базы знаний и аппарата принятия решений, 3) задачи построения аппарата эмоций.

1.1. Задача формирования и распознавания образов

Задача ФРО описана в работах [24-26,32]. В общем виде она имеет следующую постановку. На многоканальный вход УС поступает двоичная информация о непрерывных воздействиях различной физической природы, испытываемых ОУ. Аппарат ФРО должен в процессе самообучения сформировать множество дискретных образов, соответствующих классам пространственно-временных регулярностей входной информации и далее распознавать такие образы. Распознаваемые образы должны порождать формирование образов над образами, т.е. образов более высокого порядка. Переход от континуальных величин к дискретным должен происходить уже в БД. В нетривиальных случаях при отсутствии априорного алфавита классов такого рода задача является трудной. В некоторых условиях для построения БД и/или начальных трактов аппарата ФРО возможно применение перцептронов и генетического подхода, либо использование аппарата эмоций в качестве учителя. Напомним, что задача распознавания образов даже таких простых регулярностей, как периодические компоненты во временных рядах, требует применения сложного математического аппарата анализа временных рядов. Мы полагаем, что поиск неслучайных составляющих во входном многоканальном потоке входных данных должен основываться на аппарате корреляционного анализа, тогда решение о формировании нового образа может приниматься при накоплении достаточных статистических оснований.

1.2. Построение базы знаний и аппарата принятия решений

Задача построения базы знаний сводится к накоплению информации о среде, позволяющей выбирать действия, опираясь на некоторую предысторию, на текущее состояние управляемой системы и на прогноз следствий альтернативных действий. Наиболее органичным подходом является сохранение в БЗ троек вида "образ условия – образ действия – образ следствия". В элементарном случае БЗ может быть представлена трехмерной матрицей, измерения и размерности которой соответствуют образам указанной тройки множеств. Принятие решения осуществляется выбором элемента-действия из данной матрицы, которому соответствует наилучший образ-следствие.

1.3. Построение аппарата эмоций

Идея определения оценок качества образов основана на корреляционном анализе сигналов от распознанных образов и сигналов оценки текущего состояния ОУ. Этот подход подробно описан в работе [25].

В целом, управляющую систему, построенную по методу ААУ, можно изобразить следующей схемой (рис. 4).

Заметим, что предложенная схема допускает весьма простые решения. Так, в указанной тройке множеств «множество образов условия – множество образов действия – множество образов результатов», каждое из распознаваемых множеств может быть представлено, например, всего одним образом. Даже такое простое решение позволяет строить практически полезные приложения [30,36-38]. Однако в общем случае каждый из образов указанной тройки может кодировать очень длинную упорядоченную во времени цепочку подобразов. Реализация такой возможности существенно расширила бы возможности практического использования метода, чему и посвящена настоящая работа.

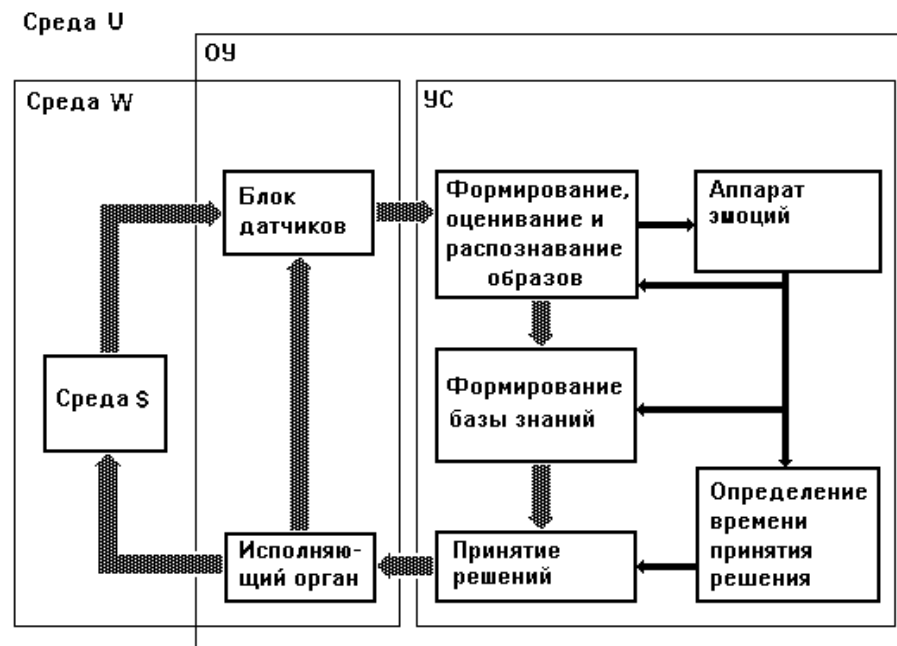


Рис. 4.

2. Детерминированный хаос. Представление о процессах обработки информации на основе сложной динамики

Технология использования детерминированного хаоса для организации хранилищ информации, способных запоминать длинные последовательности символов, взята нами из работ А.С. Дмитриева [3, 33-36]. В настоящей главе мы приводим краткое изложение этой технологии по указанным публикациям.

Исследования человеческого мозга за последнее время показали, что обработка информации животными и людьми протекает по совершенно иным алгоритмам, чем алгоритмы, используемые сейчас в электронной технике. Например, на основе анализа электроэнцефалограмм было показано существование детерминированных хаотических режимов в мозге для ряда состояний активности человека [4-10]. Электроэнцефалограмма в том или ином виде отражает информационные процессы в мозге. Поэтому указанные результаты позволяют предположить, что мозг, в отличие от компьютеров, работает также с хаотическими в некотором смысле сигналами и ведет обработку информации, используя сложную динамику такой нелинейной системы, как нейронная. Если сложная динамическая система, поведение которой в общем случае считается хаотическим или случайным, при некоторых условиях меняет свои состояния по некоторым детерминированным законам, то говорят, что данная система может находиться в режимах с детерминированным хаосом, или что она обладает таковыми.

2.1. Информация и динамические системы

Исследования систем с детерминированным хаосом также свидетельствуют о тесной связи между теорией динамических систем и информационными процессами. Ряд основополагающих результатов динамической теории формулируется применительно к объектам, так или иначе связанным с информацией [10-12].

В [13] анализируется вопрос об информационных потоках в одномерных отображениях. Автор ссылается на исследования [14], в которых аргументируется тот факт, что информация является фундаментальной концепцией в теории динамических систем и хаоса. В частности, чувствительность к начальным условиям строго относится к созданию информации. Далее автор рассматривает динамическую систему, описываемую отображением f интервала в себя, и исследует, как итерация f индуцирует специальный процесс, который автор и называет информационным потоком.

Еще одна причина, связанная с необходимостью понимания динамического хаоса с информационной точки зрения — наличие естественных объектов с детерминированной хаотической динамикой [15] или со смешанной динамикой, содержащей в себе как детерминированный хаос, так и случайный процесс.

2.2. Аттракторы и бифуркационные явления в одномерных отображениях

Имеется множество динамических систем, которые могут производить хаос. Однако мы будем говорить только о преобразованиях одного типа — одномерных отображениях, которые могут быть записаны в виде

$$X_{n+1} = f(X_n, a),$$

где a — вектор параметров. Данное отображение может обладать чрезвычайно широким набором явлений сложной динамики, присущим хаотическим системам общего вида, и, как станет ясно из дальнейшего, может служить полноценной основой для моделирования различных операций обработки информации с использованием хаоса.

В отображениях с хаосом вследствие неустойчивости возникает неопределенность в положении фазовой траектории. Степень этой неопределенности ограничивается динамическими уравнениями, связывающими предыдущее и последующее состояния системы. Информация о начальном состоянии системы (заданная всегда с конечной точностью) теряется при переходе от предыдущего состояния к последующему. Поэтому можно говорить о производстве информации в таких системах. Если же отображение демонстрирует регулярное поведение (например, периодические колебания), то его будущее полностью определено и производство информации равно нулю. Стартуя из любых начальных условий, образующих бассейн притяжения регулярного аттрактора, траектория отображения притягивается к этому аттрактору (неподвижной точке, предельному циклу), и информация о начальных условиях теряется, поскольку траектории, выходящие из разных начальных точек, становятся через некоторое время неразличимыми. Информация "исчезает", и поэтому регулярные аттракторы можно рассматривать как "стоки" информации.

Таким образом, нелинейные одномерные отображения способны как производить, так и "уничтожать" информацию. Каждую траекторию динамической системы можно рассматривать как некоторый информационный сигнал. Тем самым, совокупность траекторий отображения представляет собой своеобразное "хранилище" информации в виде множества траекторий системы. Это "хранилище" обладает рядом полезных свойств, проявление которых зависит от того, имеются ли аттракторы в динамической системе и какого они типа.

Рассмотрим некоторые из этих свойств.

Если в момент времени $t = t_0$ задать начальные условия, то формально траектория отображения известна и однозначно определена для всех $t > t_0$ (аналог теоремы Коши для отображений). Если траектории вдоль ее длины сопоставлен информационный сигнал, он может быть воспроизведен путем итерирования отображения с заданными начальными условиями, т.е. информация извлекается из памяти путем решения эволюционного уравнения.

"Хранилище" информации как динамическая память обладает естественными ассоциативными свойствами. Действительно, задавая любой фрагмент сигнала, можно воспроизвести все его последующие значения, а в случае циклического сигнала — и весь сигнал целиком.

Если в отображении имеется аттрактор — устойчивый предельный цикл, то все траектории со временем стягиваются к нему. Периодическую траекторию,

соответствующую предельному циклу, можно рассматривать как периодически воспроизводимый информационный сигнал. Объем информации в таком сигнале определяется длиной цикла и степенью его изрезанности (информационной насыщенностью).

За последние несколько лет были разработаны методы синтеза специальных отображений с записанной информацией, соответствующие алгоритмы и программные комплексы, практически не имеющие ограничений на объем информации, записываемой на предельных циклах отображений. Кроме того, итерации отображения вдоль цикла нейтральны в смысле сжатия информации: они не приводят ни к исчезновению старой информации, ни к порождению новой и просто воспроизводят периодически одну и ту же информацию. т.е. идеально сохраняют ее.

По перечисленным причинам будем рассматривать предельные циклы отображений в качестве основных объектов для записи и хранения информации в отображениях с хаосом.

Рассмотрим случай, когда отображение имеет два или более регулярных аттрактора. В зависимости от начальных условий траектория притягивается к тому или иному аттрактору. Отображение тем самым "различает" начальные условия и "классифицирует" их на несколько типов по числу аттракторов. Оно сжимает множество всех начальных условий в несколько классов, т.е. способно выполнять роль распознавателя (классификатора). В таком отображении информация как бы обобщается, "производства" информации нет. Результат функционирования система переносит на новый уровень иерархии: вместо исходного множества траекторий имеется несколько специальных траекторий — аттракторов, каждый из которых характеризует класс информационных объектов.

В качестве среды для записи и хранения информации в [16] было предложено использовать динамическую систему с дискретным временем — одномерное отображение отрезка в себя. В этой простой системе была продемонстрирована возможность записи на предельных циклах, показана возможность реализации ассоциативной памяти, распознавания образов и ряда других базовых функций обработки информации с использованием сложной динамики [16-18].

Рассмотрим процедуру записи и восстановления информации на основе предельных циклов одномерных динамических систем. Записываемая информация представляется в виде информационных блоков — конечных последовательностей вида

$$a_1, a_2, a_3, \dots, a_n$$

Каждый элемент a_i информационного блока принадлежит конечному упорядоченному множеству $A = \{a_j, j = 1 \dots N\}$, которое далее будем называть алфавитом. Примерами таких алфавитов являются кириллица, нотная азбука, набор ASCII-кодов и т.д.

Следует заметить, что при записи на замкнутую траекторию информационный блок мысленно "сшивается" в кольцо, так что после последнего элемента a_n следует первый a_1 , поэтому фрагментом информационного блока будем называть односвязный "отрезок" этого кольца.

К синтезируемой функции $f(x)$ одномерного отображения $x_{n+1} = f(x_n)$ с записанной информацией предъявляются следующие требования: предельный цикл, несущий информацию об информационном блоке длиной n , последовательно проходит через n точек; между точками цикла и элементами соответствующего информационного блока существует взаимно однозначное соответствие; контролируется характер устойчивости предельного цикла.

Синтез одномерного отображения начинается с того, что в одномерном фазовом пространстве — отрезке $[0, 1]$, для каждого из записываемых информационных блоков $a_1, a_2 \dots a_n$, строится замкнутая траектория - цикл $\gamma = \{x_1 \dots x_m\}$, каждая точка которого взаимно однозначно связана с соответствующим фрагментом информационного блока. После того, как в фазовом пространстве построен скелет замкнутых траекторий-циклов, отвечающих записанным информационным блокам, конструируется собственно функция $f(x)$.

Для этого на плоскости (X_m, X_{m+1}) откладываются пары последовательных точек всех циклов (x_i, x_{i+1}) . Любая однозначная кривая, проходящая через эти точки, определяет функцию $f(x)$ одномерной динамической системы, удовлетворяющую первым двум требованиям, предъявляемым к функции отображения. Для управления устойчивостью циклов через точки на плоскости (X_m, X_{m+1}) проводятся короткие отрезки с фиксированным наклоном s (далее называемые информативными участками). Как известно, устойчивость предельного цикла определяется его мультипликатором μ . В случае одномерного отображения для цикла $\gamma_n = \{x_1, \dots, x_n\}$ он равен $\mu = f'(x_1) \cdot \dots \cdot f'(x_n)$, а в данном случае $\mu = s^n$. При $\mu < 1$ ($s < 1$) цикл устойчив, при $\mu > 1$ ($s > 1$) — неустойчив.

Синтез функции $f(x)$ искомого одномерного отображения завершается последовательным соединением концов информативных участков между собой и концами отрезка $[0,1]$ прямыми линиями.

Пример 1. Продемонстрируем принцип записи информации на предельных циклах одномерных динамических систем на примере записи информационного блока — слова "бег". В качестве алфавита возьмем подмножество кириллицы: $A = \{a, б, в, г, д, е, ж, з, и, к\}$, длина алфавита $N = 10$. Разделим фазовое пространство — единичный отрезок $I = [0, 1]$ — на N регионов — отрезков длиной 0,1 и каждому из них поставим в соответствие букву — элемент алфавита (рис. 5). Попадание фазовой траектории на тот или иной отрезок (тот или иной регион фазового пространства) будем интерпретировать как появление на "выходе" динамической системы соответствующего элемента алфавита.

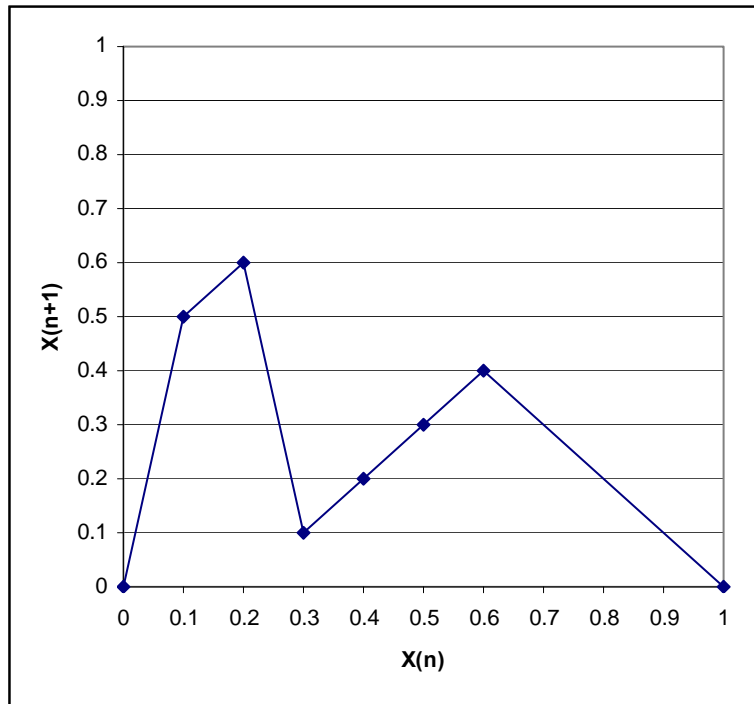


Рис. 5. Отображение с одним циклом, несущим слово "бег".

Теперь для информационного блока "бег" построим цикл $\gamma_3 = \{x_1, x_2, x_3\} = \{0,15; 0,55; 0,35\}$. Каждая точка этого цикла однозначно связана с одним из элементов (фрагментом единичной длины) информационного блока, и представляет собой центр соответствующего региона $x_j = (m_j - 0,5)/N$, где m_j — порядковый номер элемента блока a_i в алфавите.

После того как в одномерном фазовом пространстве построен цикл, отвечающий записанному информационному блоку, на плоскости (X_n, X_{n+1}) отложим точки вида (x_j, x_{j+1}) : $(0,15; 0,55)$, $(0,55; 0,35)$, $(0,35; 0,15)$. Через каждую из них проведем отрезок с наклоном $s = 0,5$, доходящий до границ региона. Эти информативные отрезки обеспечивают устойчивость цикла. Соединим концы информативных участков между собой и с концами отрезка $[0, 1]$ прямыми линиями. Синтез функции $f(x)$ одномерного отображения $x_{n+1} = f(x_n)$ закончен.

2.3. Уровень записи

Пример 2. Попробуем записать на одном отображении два слова "бег" и "вид" на двух предельных циклах. Будем использовать алфавит из примера 1. В этом

случае описанную выше процедуру придется немного усложнить, введя понятие уровня записи, иначе отображение получится неоднозначным. Проведем запись этих слов на втором уровне.

Как и в предыдущем случае, для каждого информационного блока строим свой цикл $\gamma_n = \{x_1, \dots, x_n\}$, каждая точка x , которого будет определяться не одним элементом, информационного блока, а парой соседних элементов (фрагментом длины 2): $x_j = (m_j - 1)/N + (m_{j+1} - 0,5)/N^2$. Чтобы вычислить эту точку, единичный отрезок I делим на N регионов первого уровня и ставим им в соответствие свои элементы алфавита. Каждый из этих регионов, в свою очередь, делим на N регионов второго уровня, и с каждым из них также сопоставляем свой элемент алфавита.

Для информационного блока "бег" три точки цикла γ_3^1 связаны с фрагментами "бе", "ег", "зб". Чтобы найти точку цикла, соответствующую фрагменту "бе" в регионе первого уровня "б" возьмем регион второго уровня "е", центр этого региона поставим в соответствие фрагменту "бе". Аналогично поступим с другими фрагментами слов "бег" и "вид". В результате получим два цикла $\gamma_n^{1,2}$ однозначно связанные с этими информационными блоками.

Для построения функции $f(x)$ отображения с этими циклами отложим на плоскости (X_n, X_{n+1}) точки с координатами (x_j, x_{j+1}) , где x_j, x_{j+1} — последовательные точки циклов γ_3^1 и γ_3^2 . Для управления устойчивостью циклов через каждую такую точку проведем информативный отрезок с фиксированным наклоном s , как в примере 1, но на этот раз проведем его до границ участков уровня 2.

Синтез функции $f(x)$ завершается последовательным соединением концов информативных отрезков между собой и с концами единичного отрезка прямыми линиями (рис. 6).

Аналогично может быть произведена запись информации на произвольном уровне q . Каждая точка соответствующего цикла определяется фрагментом длиной q соответствующего информационного блока, а длина информативных интервалов (проекций информативных участков) составляет N^q , это длина региона уровня q .

Выбор уровня записи диктуется, во-первых, требованием однозначности функции отображения $f(x)$ (ограничение снизу), во-вторых, точностью вычислений (ограничение сверху), поскольку с ростом уровня записи уменьшается длина информативных интервалов.

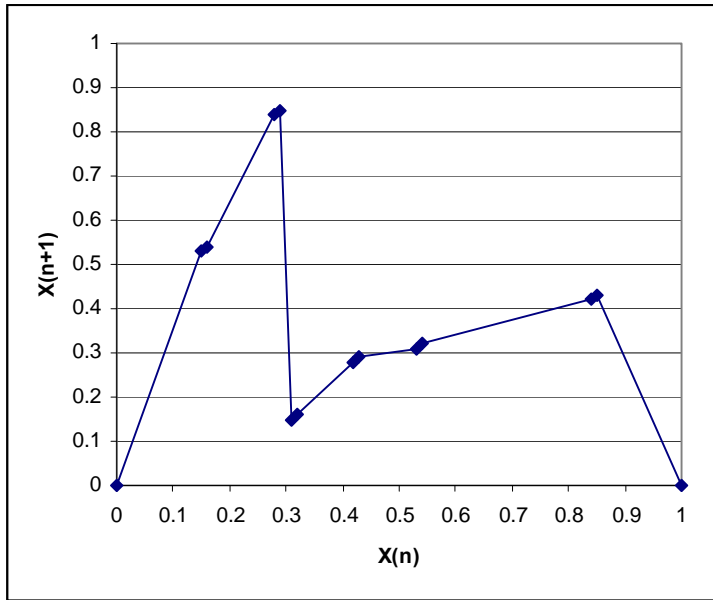


Рис. 6. Функция отображения: слова "бег" и "вид" записаны на двух циклах на втором уровне.

2.4. Ассоциативное извлечение информации

Восстановление (извлечение) информации, записанной на предельных циклах одномерных отображений, наиболее просто осуществляется при записи информации на устойчивых циклах (при $|s| < 1$). При этом проекции информативных участков на ось X_m информативные интервалы) составляют области гарантированной (непосредственной) сходимости фазовой траектории к предельным циклам. При задании начальных условий x_0 на одном из информативных интервалов фазовая траектория динамической системы быстро сходится к соответствующему предельному циклу, а на "выходе" отображения наблюдается воспроизведение записанного информационного блока, поскольку фазовая траектория системы при этом последовательно обходит соответствующие регионы первого уровня.

При задании начальных условий x_0 на неинформативном участке отображения выходу траектории на предельный цикл предшествует переходный хаос, когда фазовая траектория системы блуждает по неинформативным участкам отображения. Рано или поздно она попадает на информативный участок отображения и сходится к тому или иному предельному циклу с записанной информацией.

При задании произвольных начальных условий попадание фазовой траектории системы на тот или иной информационный предельный цикл для стороннего наблюдателя представляется случайным. Однако рассматриваемый принцип записи информации дает возможность извлечения конкретного информационного блока, если известна какая-либо его малая часть (цитата, фрагмент).

Если известен точный фрагмент информационного блока длиной не меньшей, чем номер уровня записи, то, используя описанную выше процедуру построения точек цикла, по предъявленному фрагменту информационного блока можно восстановить, по крайней мере, одну точку соответствующего предельного цикла и, проитерировав отображение, восстановить весь информационный блок.

Свойство ассоциативного доступа к записанной информации по известному фрагменту (ассоциативная память) напрямую вытекает из динамических свойств отображения и аттракторов - носителей информации.

2.5. Кодирование информационных блоков

Анализ информационных последовательностей (изображений, текстов и т.д.) показывает, что наибольшие трудности с их записью связаны с длинными однородными участками, хотя здравый смысл подсказывает, что они содержат немного информации. Эти соображения приводят к мысли о записи предварительно сжатой информации, но сжатой особым образом, чтобы обеспечить запись на уровне q . Процесс сжатия информации (устранения избыточности) назовем "ортогонализацией", чтобы подчеркнуть, что ее целью является обеспечение записываемости образов [19].

Один из возможных алгоритмов заключается в кодировании информационных последовательностей с построением алфавита повторяющихся фрагментов. Идея такова: если в информационных блоках обнаруживается повторяющийся фрагмент длиной q , то алфавит расширяется на один символ, представляющий собой этот фрагмент, а в последовательностях он кодируется этим новым символом. Получаются новые информационные блоки меньшей длины, выраженные в алфавите длиной $(N + 1)$. Процедура повторяется до тех пор, пока они не станут "ортогональными", т.е. не содержащими одинаковых фрагментов длиной q .

В результате применения такого алгоритма получаются ортогональные, более короткие закодированные информационные последовательности и расширенный алфавит, состоящий из исходного алфавита длиной N и дополнительного, элементы которого состоят из некоторого фрагмента длиной q и символа, представляющего (кодирующего) этот фрагмент в последовательностях.

Таким образом, ортогонализация состоит в устранении избыточной информации в виде повторяющихся фрагментов за счет кодирования их более короткими

символами. В отличие от рассмотренного выше подхода, где запись обеспечивалась увеличением ее уровня, в данном случае фиксируется уровень и наращивается алфавит. Это дает принципиальную возможность записи любой совокупности информационных блоков на любом уровне, начиная со второго.

Данный метод кодирования информационных блоков является обратимым, потеря информации не происходит. Для декодирования информации символы дополнительного алфавита в закодированной последовательности заменяются фрагментами длиной q из алфавита. Эта процедура может повторяться несколько раз, либо следует предварительно выразить все фрагменты дополнительного алфавита через элементы начального алфавита.

Естественно, процедура кодирования неоднозначна. Окончательный вид закодированных последовательностей, расширенного алфавита, их длина определяются порядком пополнения алфавита совпадающими фрагментами, т.е. процедурой их поиска.

2.6. Сжатие информации

Как отмечалось выше, метод кодирования (ортогонализации) информационных последовательностей разработан для записи любого набора информационных блоков на одномерном отображении. "Побочным" результатом ортогонализации является обратимое (без потерь) сжатие информации — уменьшение общего объема данных, необходимых для полного и точного описания информационных объектов, например текстов или изображений.

Если взглянуть на описанный метод ортогонализации как на метод сжатия информации, то оказывается, что коэффициент сжатия, достигаемый применением этого метода, примерно совпадают с тем, что дают другие известные методы сжатия информации без потерь — арифметический, Лемпеля-Зива, Хаффмана.

2.7. Ассоциативное извлечение информации при использовании кодирования

Для реализации ассоциативной памяти, т.е. восстановления записанного образа по его произвольному фрагменту, необходимо задать начальные условия, лежащие на соответствующем аттракторе. Для этого нужно знать фрагмент $(a_j, a_{j+1}, \dots, a_{j+1+q-l})$ длиной q информационного блока и вычислить начальную точку x_0

$$x_0 = \sum_{k=1}^q \frac{a_j + k - 1}{N^k} \quad (1)$$

При записи на уровне q сначала проводится ортогонализация информационных блоков, а затем закодированные блоки записываются на предельные циклы отображения. Если известен фрагмент закодированного информационного блока

длиной q , то по нему, согласно (1), может быть построена начальная точка x_0 , восстановлен закодированный информационный блок, декодирован исходный информационный блок. Однако гораздо больший интерес представляет восстановление последовательности не по фрагменту закодированного блока, а по фрагменту исходного паттерна в исходном алфавите.

Для реализации ассоциативного доступа в рассматриваемом случае необходимо сначала получить из предъявленного фрагмента паттерна в исходном алфавите соответствующий фрагмент закодированной информационной последовательности. Для этого нужно сначала закодировать его, пользуясь имеющимся в наличии дополнительным алфавитом — таблицей фрагментов, полученным при ортогонализации информационных блоков.

Проблема заключается в том, что фрагмент, предъявленный для распознавания, может начинаться с произвольного элемента исходного информационного блока, включая и такие, которые не существуют в закодированном блоке, потому что вошли в состав фрагментов, замененных одним символом. Пусть, например, блок $(abcdefghijk)$ после кодирования на третьем уровне принял вид $(xdyhz)$, где $x = (abc)$, $y = (efg)$, $z = (ijk)$, а системе ассоциативной памяти предъявлен фрагмент исходного блока $(cdefghi)$. При преобразовании предъявляемого фрагмента из исходного в новый алфавит получается некая новая последовательность $(cdyhi)$, содержащая "правильный" кусок закодированного блока (dyh) в середине и "неправильные" элементы по краям.

Чтобы "отсеять" их, используем следующую процедуру. По первым q элементам закодированного последовательного фрагмента строим начальную точку, согласно (1). Затем, используя знание отображения, определяем, попадает ли она на информативный интервал. Если x_0 не попадает на информативный интервал, значит, по крайней мере, первый элемент закодированного последовательного фрагмента "неправильный". В таком случае переходим к следующему элементу, и так до тех пор, пока не окажется, что x_0 попадает на информативный интервал. Некоторый текущий фрагмент $(b_1 b_2 \dots b_q)$, из которого получена начальная точка x_0 , является кандидатом на начало "правильной" части закодированного фрагмента. Чтобы убедиться в этом, проитерируем отображение несколько раз. Начальная точка x_0 порождает последовательность итераций $x_0, x_1=f(x_0), x_2=f(f(x_0)), \dots$, а также соответствующий информационный поток $c_0 = \text{int}(Nx_0), c_1, c_2, \dots$. Если $(b_1 b_2 \dots b_q)$ действительно является "правильным" участком, и "неправильные" элементы спереди уже отфильтрованы, то $c_0=b_1, c_1=b_2$ и т.д. Если это не так, значит, попадание на информативный интервал было случайным, и элемент последовательности b_1 тоже "неправильный". Если в закодированном последовательном фрагменте имеется "правильный" кусок, по крайней мере, длиной q , то он будет найден таким образом. Если длина его не меньше q , его можно использовать для получения начальных условий на

аттракторе. После того как получена начальная точка x_0 , легко восстановить всю информационную последовательность и, соответственно, весь паттерн.

Если весь закодированный фрагмент просмотрен, а найти начальную точку на аттракторе не удалось, значит, длина фрагмента была недостаточна — после кодирования в нем не нашлось "правильного" фрагмента длиной q .

Таким образом, система ассоциативной памяти, построенная на описанных принципах, в ответ на предъявленный информационный блок или его часть практически мгновенно формирует один из двух возможных откликов: либо возвращает начальное условие x_0 на соответствующем аттракторе — предельном цикле, по которому может быть полностью восстановлен блок; либо дает ответ, что предъявленной информации недостаточно для однозначного формирования начального условия на аттракторе, это же происходит и тогда, когда предъявленный блок не записан на отображении. При предъявлении некоторого фрагмента блока система ассоциативной памяти либо безошибочно восстанавливает информационный блок, либо сообщает о неспособности сделать это. Ошибка невозможна в силу однозначности процедуры кодирования предъявляемого фрагмента, "ортогональности" записанных информационных блоков, однозначности самого отображения, а также благодаря использованию при формировании начального условия описанной выше процедуры сравнения закодированного фрагмента с порождаемым информационным потоком.

Следует заметить, что формирование начального условия на аттракторе и соответственно восстановление исходного паттерна по его фрагменту происходит без сравнения предъявляемого фрагмента с каждым из записанных образов. После кодирования предъявляемого фрагмента для вычисления начальной точки x_0 может понадобиться выполнение лишь нескольких итераций для проверки попадания на аттрактор отображения, а время, затраченное на каждую итерацию отображения, пропорционально логарифму объема записанной информации.

Таким образом, описанная схема ассоциативной памяти для изображений обладает свойствами очень быстрого коррелятора.

3. Использование технологии детерминированного хаоса в системах автономного адаптивного управления

Основная идея настоящей работы состояла в том, чтобы использовать технологию детерминированного хаоса, как способа запоминания длинных последовательностей упорядоченных данных и ассоциативного обращения к ним, в управляющей системе ААУ в качестве памяти для хранения длинных упорядоченных во времени последовательностей событий, отражающих историю

деятельности управляющей системы и состоящих из идентификаторов распознанных образов и совершенных действий.

Требовалось найти способы встраивания хаотических процессоров в состав управляющей системы ААУ, исследовать эти варианты, сравнив их с альтернативными подходами, и проверить работу системы на модельных примерах.

3.1. Возможные варианты использования хаотических процессоров в системах автономного адаптивного управления

Для математического моделирования рассматриваемых систем и их сравнения с альтернативными подходами, была разработана программная реализация модуля памяти управляющей системы ААУ с интерфейсом, который позволял проводить анализ результатов моделирования, именно, запоминать необходимые объемы информации и оценивать необходимые для этого ресурсы, оценивать производительность метода, и т.д. При этом оказалось неэффективным использование таких структур, как хэш-массивы и деревья, потому что эти структуры предполагают сортировку хранимой информации. Приемлемыми свойствами, которые могут рассматриваться как альтернатива хаотическому процессору, обладают только такие структуры, как списки.

Исходя из приведенного описания структуры управляющей системы, построенной по методу ААУ, и функций ее подсистем, можно видеть, что каждая из подсистем управляющей системы может обладать адаптивными свойствами. В самом общем плане это следует уже из того, что естественная нервная система построена на «элементной базе» нервных клеток, а каждая нервная клетка обладает адаптивными свойствами – она способна накапливать статистическую информацию в своих синапсах и изменять свое функционирование соответствующим образом. Разрабатывая управляющую систему, мы также стремимся каждую ее подсистему сделать адаптивной. Тем не менее, можно указать основные подсистемы, в которых в первую очередь требуется запоминание последовательных данных по предьстории процесса управления и анализ этих данных для их автоматической классификации (кластеризации, формирования образов) и распознавания. Это подсистемы ФРО и БЗ.

Рассмотрим оба варианта подробнее.

3.2. Система формирования и распознавания образов

В подсистеме ФРО должны автоматически выделяться повторяющиеся пространственно-временные фрагменты во входной информации, и, при их обнаружении, формироваться в виде отдельных объектов – образов, записываемых в память образов. Сформированные образы должны этой же подсистемой ФРО распознаваться в текущей входной информации ассоциативным (в указанном смысле) способом. Чем более протяженными могут

быть прообразы формируемых и распознаваемых образов, тем интереснее свойства управляющей системы. Поэтому подсистема ФРО является местом, где применение хаотического процессора целесообразно.

В случае построения модуля ФРО на базе хаотического процессора управляющая система получает возможность формировать образы более длинных закономерных цепочек сигналов, повторяющихся в потоке сенсорной информации. Изначально умея распознавать элементарные образы, система ФРО, обрабатывая последовательности неоднократно встречающихся последовательностей элементарных образов, может формировать образы более высокого порядка.

3.3. База знаний

Подсистема БЗ фактически является частью подсистемы ФРО, ее можно рассматривать как отдельную подсистему только с той точки зрения, что особо следует выделять образы, прообраз которых состоит из трех частей, упорядоченных во времени: «образ начального состояния; образ действия; образ результирующего состояния». Такие составные образы являются, по сути, элементами знания, поскольку позволяют осуществлять управление. Такой образ является составным, так как состоит из трех частей, однако каждая из его частей может представлять собой образ как угодно протяженной последовательности событий. Отсюда следует актуальность придания базе знаний свойств, присущих хаотическому процессору.

В целях исследования эффективности применения технологии динамического хаоса, была реализована управляющая система, в которой поток образов поступал в подсистему БЗ через модуль хаотического процессора, обеспечивающий работу подсистемы ФРО.

3.4. Модель окружающей среды

Очевидно, что универсальную систему управления построить невозможно, и для каждого конкретного приложения существуют свои наиболее эффективные методы обработки информации. Эти соображения относятся как к системе управления в целом, так и к каждой ее подсистеме в отдельности.

Нами для исследования системы был выбран модельный пример, представляющий собой некоторое пространство - лабиринт со случайно расположенными в нем объектами. В расположении объектов встречаются неоднократно повторяющиеся ситуации. По лабиринту движется робот, наблюдающий объекты и получающий при этом определенные вознаграждения или наказания. Лабиринт был представлен сетевой структурой, в которой содержание узлов (объектов, или символов) и соединительных связей (отрезков дороги) не были заданы жестко. Содержание узла представляет собой выборку M из N элементарных объектов, которые априори умеет распознавать система ФРО. Каждый из объектов имеет определенную ценность (вес), которые определены для каждого из объектов. Управляющая система может считывать содержание

узла, и выбирать одну из возможных связей этого узла для перехода к следующему узлу. Связи между узлами на языке системы управления выбираются (активизируются) одним из действий (“вверх”, “вперед”, “вниз”, “влево”, и т.д.). Поток информации от модели внешней среды в режиме обучения имеет следующий вид упорядоченных во времени событий:

$$O_{i1}, O_{i2}, \dots, O_{iM}, A_i O_{(i+1)1}, O_{(i+1)2} \dots O_{(i+1)M} A_{(i+1)}, \dots,$$

где O_{ij} – распознавание объекта, находящегося в текущем узле, а A_i – действие-переход к следующему узлу (в режиме управления действия A_i генерируются управляющей системой, точнее ее системой принятия решений, опирающуюся на базу знаний). Как уже отмечалось, в принятой нами модели среды отсутствует жесткость в определении связей и наполнении узлов лабиринта, т.е. выбор очередного объекта O_i может опираться либо на генератор случайных чисел, либо на правила, заданные априори. При описании модели среды задается R_n правил, которые определяются *информационной строчкой* образов/действий A_j^k ($k = 1..R_n, j = 1..L_k$) и *числом* C^k ($k = 1..R_n$) - минимальным числом совпадений, которые должны произойти, чтобы правило вступило в силу. Строчка A_j^k ($k = 1..C^k$) должна быть уникальна для каждого правила. Алгоритм выглядит следующим образом:

1 Если есть активные правила, то возвращается следующий элемент информационной строчки этого правила. Если выбранный элемент оказывается последним в строчке, то правило становится пассивным.

2 Если активных правил нет, то просматривается k -ое ($k = 1..R_n$) правило и сравнивается предыстория (последние C^k элементов, возвращенные окружающей средой) с информационной строчкой A_j^k ($j = 1..C^k$).

3 Если совпадений нет, то следующий образ/действие выбирается случайно.

4 Если найдено совпадение ровно с одним (k -ым) правилом, то следующим символом будет $A_{C^k+1}^k$ -ый элемент строчки правила, которое становится активным.

Использование технологий детерминированного хаоса в выбранном контексте управления при определенных условиях на M, N, C_{\min}^k (минимальная длина префикса правил), R_n может оказаться более или менее эффективным. Первый вывод, который можно сделать, состоит в том, что чем больше объектов характеризует каждый узел и чем более уникальны эти узлы, тем более оправданной может оказаться рассматриваемая технология.

Пример модели окружающей среды:

Объекты: APPLE (A), BAT(B), CANDLE(C), DIAMNOD(D), EMERALD(E)

Оценки:

APPLE

1

BAT	-1
CANDLE	0
DIAMOND	5
EMERALD	4

Возможные действия (связи):

RIGHT, LEFT, FORTH

Правила:

APPLE BAT LEFT DIAMOND EMERALD, $C_k = 3$

APPLE BAT RIGHT CANDLE APPLE, $C_k = 3$

4. Результаты математического моделирования

Компьютерное моделирование системы показало, что управляющая система способна обнаруживать и запоминать повторяющиеся фрагменты лабиринта, затем распознавать их при встрече, причем распознавать до того, как они заканчиваются, и когда еще можно принять решение о том или другом варианте продолжения пути, и выбирать то продолжение пути, которое способно дать большее вознаграждение.

Построенные программные структуры позволили провести сравнение систем управления, использующих память на технологиях детерминированного хаоса, и память в виде списков. Учитывались следующие динамические характеристики:

- $V_{mem}(N_{saved})$ - объем памяти, необходимый для хранения записанной информации,
- $t_s(N_{saved})$ - зависимость времени запоминания от объема накопленной информации,
- $t_r(N_{saved}, L_{substr})$ - зависимость времени воспроизведения от объема накопленной информации и от длины подстроки запроса,
- $p_r(N_{saved}, L_{substr})$ - вероятность воспроизведения полного образа по его части, где N_{saved} – число сохраненных паттернов,
- L_{substr} – длина подстроки запроса на воспроизведение паттерна.

Для сравнения были рассмотрены случаи с различными значениями параметров N - размера входного алфавита, и M - максимальной длины паттерна.

Разработанные системы исследовались на описанном выше модельном примере с лабиринтом. Вычисления проводились на PC Celeron 350, 192MB ОЗУ.

Рассмотрим полученные зависимости более подробно.

4.1. Время записи

Во всех четырех рассматриваемых случаях время, необходимое на запись новых паттернов хаотическим процессором, во много раз превышало время добавлении информации в несортированный список. Общая тенденция роста времени от объема записанной информации выглядит следующим образом (рис 7.):

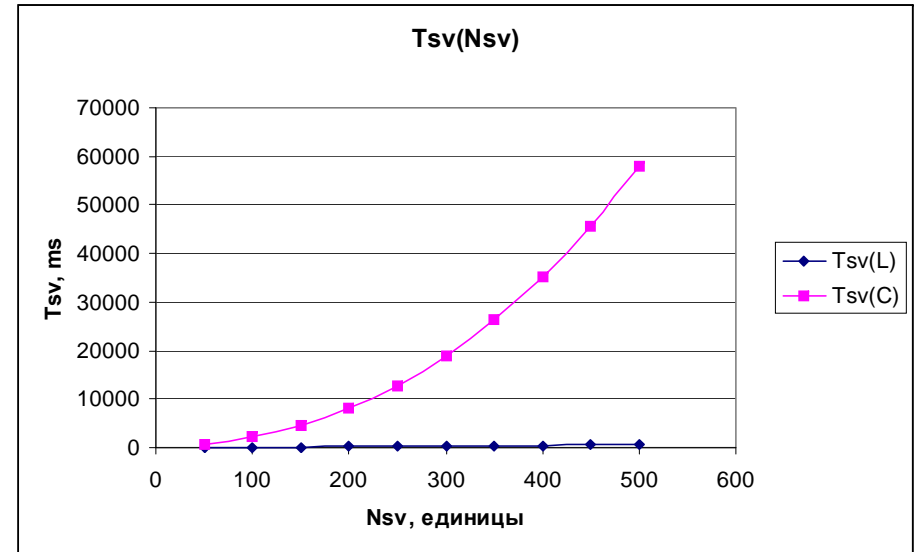


Рис. 7. Зависимость времени записи от числа записанных паттернов. Здесь $T_{sv}(C)$ – время записи на хаотический процессор, $T_{sv}(L)$ – время записи в память-список, N_{sv} – количество сохраненных паттернов.

Экспоненциальный рост времени записи хаотическим процессором объясняется тем, что информация должна быть определенным образом структурирована – переведена из машинного формата (линейная последовательность байт) в формат одномерного отображения, предварительно пройдя процедуру ортогонализации, которая является весьма трудоемкой. Отсюда следует, что время записи во многом определяется тем, как именно представляется одномерное отображение в памяти компьютера. Поиск более оптимальных форм этого представления является предметом дальнейших исследований.

4.2. Объем памяти

Графики зависимости объема памяти от количества записанных паттернов (рис. 8) в рассматриваемых случаях тоже были принципиально похожи друг на друга:

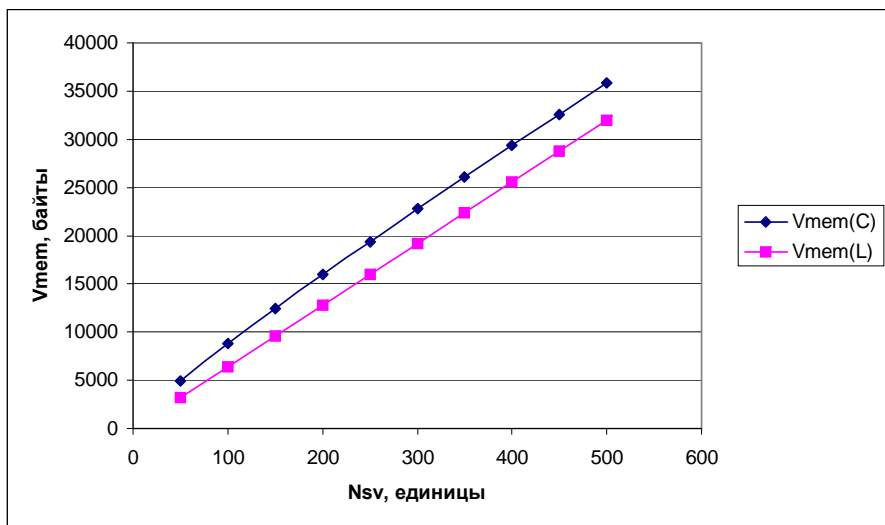


Рис. 8. Зависимость объема памяти от числа записанных паттернов. Здесь $V_{\text{mem}}(C)$ – это объем в случае записи на хаотический процессор, $V_{\text{mem}}(L)$ – это объем памяти при записи на память-список

Объем памяти, необходимый для сохранения K паттернов в списке линейно зависит от K и средней длины паттерна, то есть на графике он выглядит как прямая линия. В случае хаотического процессора рост необходимой памяти в начале (при небольшом количестве записанной информации) идет более круто, чем линейная зависимость. Но поскольку в процессе преобразования информации к виду одномерного отображения происходит ее кодирование (повторяющиеся части паттернов заменяются на новые элементы алфавита - на один символ), то вскоре рост объема памяти становится медленнее линейной зависимости. Оговоримся, что здесь под объемом памяти мы имеем ввиду те ресурсы памяти, которые необходимы для воспроизведения информации. Для записи новой информации используется большее количество ресурсов, минимальный объем которых примерно равен всему объему записанной ранее информации. Такие затраты связаны с необходимостью кодирования информации с добавлением почти каждого слова (фактически перекодирование происходит каждый раз, когда увеличивается размер алфавита).

4.3. Время воспроизведения

Важным условием, благодаря которому мы отказались от использования упорядоченных структур хранения информации, в нашем сравнении является ассоциативность памяти: то есть мы должны уметь вспомнить паттерн целиком, располагая лишь его частью, и эта часть не обязательно должна быть его началом. Рассмотрим отдельно графики зависимости времени воспроизведения паттерна хаотическим процессором и хранилищем, основанном на списках (рис 9, 10).

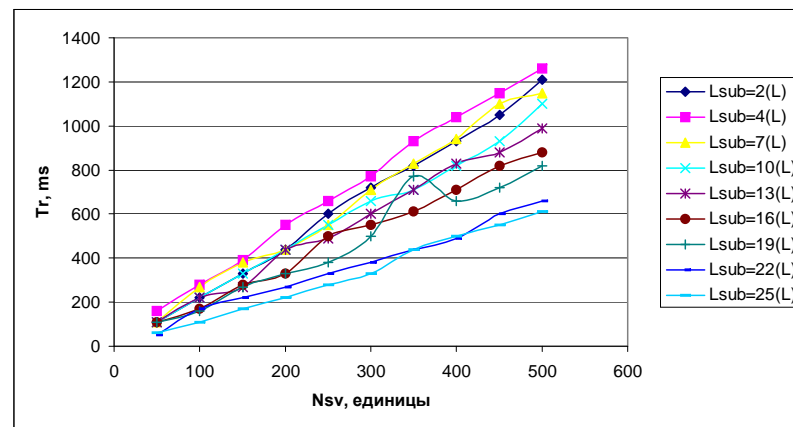


Рис. 9. Списки. Зависимость времени восстановления паттерна по его части. T_r – время воспроизведения паттерна.

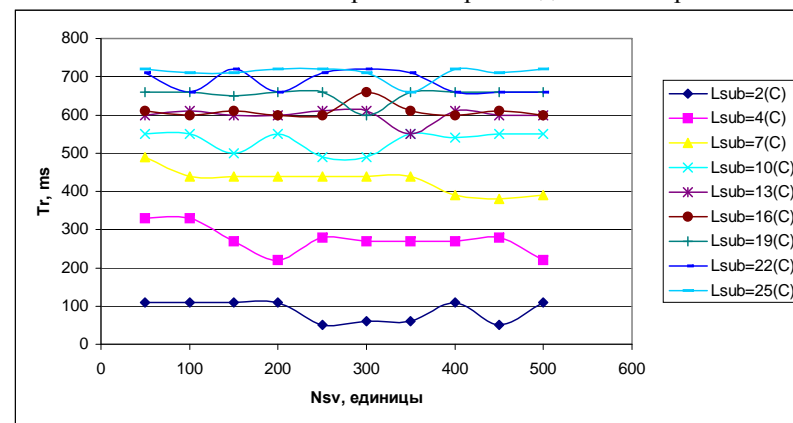


Рис. 10. Хаотический процессор. Зависимость времени восстановления паттерна по его части.

Названия графиков $L_{\text{sub}}=XX(Y)$ говорят о длине подстрок, по которым воспроизводились паттерны (XX = длина подстроки), и используемом

хранилище информации ($Y \equiv C$ – хаотический процессор, $Y \equiv L$ – список). Зависимости времени извлечения информации из списка линейно зависят от объема записи информации (что хорошо согласуется с теорией: время поиска прямо пропорционально времени поиска подстроки в одной строке и количеству строк в хранилище), а время извлечения информации из хаотического процессора заметно зависит лишь от длины подстроки, по которой мы хотим найти целый паттерн.

4.4. Достоверность

Время воспроизведения информации – это лишь одна составляющая эффективности использования того или иного хранилища данных. Другой составляющей является достоверность воспроизводимой информации (см рис 11, 12). Достоверность определялась следующим образом: выбирался один из сохраненных паттернов. На вход памяти подавалась подстрока этого паттерна произвольной длины и паттерн, получаемый на выходе, сравнивался с исходным. В случае их совпадения испытание считалось успешным. Доля, которую составляют успешные испытания от проводимых испытаний, мы и называем *достоверностью*. В данном случае число проводимых испытаний составляло 1000, и длины подстрок выбирались равными 2, 4, 7, 10, 13, 16, 19, 22 и 25. Для памяти в виде списка достоверность строго равна единице в том случае, если длина подстроки на входе оказывается больше максимальной длины подстроки, встречающейся одновременно хотя бы в паре записанных паттернов (L_{submax}). Для хаотического процессора меньшая достоверность оказывается своеобразной платой за высокую скорость воспроизведения. Чтобы результат оказался достоверным на 100%, подстрока на входе должна быть более длинной, чем в случае памяти-списка, поскольку информация в памяти хранится в сжатом виде. Математически строго обозначить ограничения для подстроки-запроса довольно сложно: она зависит от длины алфавита, длины запоминаемых паттернов, количества сохраненных паттернов и «похожести» запоминаемых паттернов. В общем, этот вопрос может являться темой для дальнейших исследований. Эмпирически полученные результаты показывают, что начиная с длины подстроки $L_{substr} = 16$, достоверность не опускается ниже 98%, а для $L_{substr} \geq 22$ строго равна 1. То есть начиная с $Nsv \approx 350$, память на детерминированном хаосе оказывается более эффективной, нежели память с линейной организацией данных.

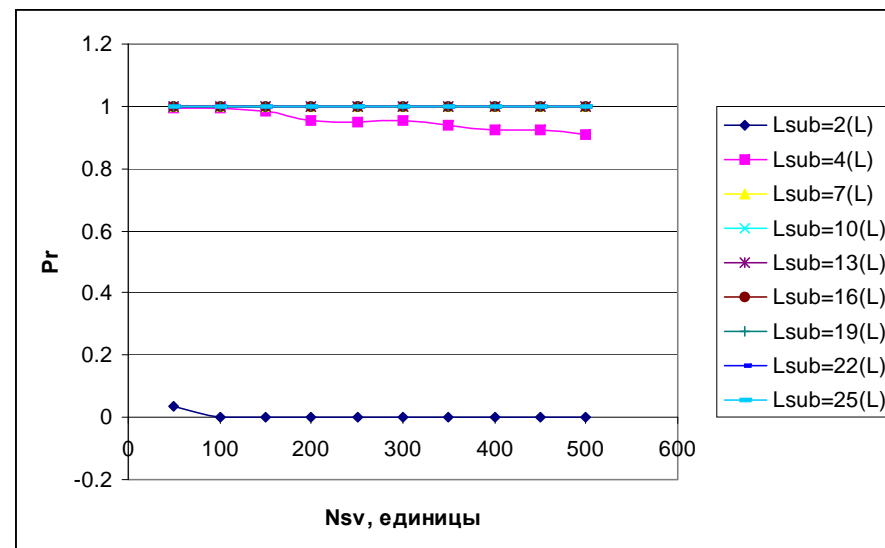


Рис. 11. Списки. Достоверность восстановления паттерна по его части.

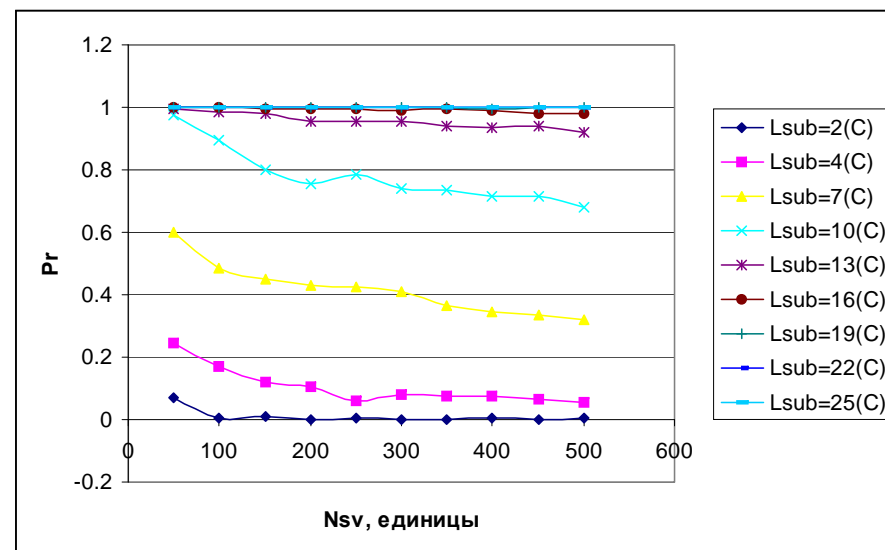


Рис. 12. Хаотический процессор. Достоверность восстановления паттерна по его части.

4.5. Модель системы ААУ

С использованием памяти на хаотическом процессоре была построена работоспособная модель системы ААУ. Модельный пример объекта управления и окружающей среды был описан в предыдущей главе – лабиринт, наполнение которого определялось заданными априори правилами. Система ААУ может работать в двух режимах – режим обучения и режим управления. В первом из этих режимов модуль ААУ получает от имитатора окружающей среды информацию о встречающихся объектах и о выполняемых управляющей системой действиях. В режиме управления на вход поступает лишь информация об объектах, а действия генерируются самой управляющей системой. В режиме обучения цикл управления выглядел следующим образом:

1. Данные от сенсора о текущем наблюдаемом символе поступают к датчикам.
2. Датчики воспринимают символ и передают его системе ФРО.
3. В системе ФРО накапливается информация до получения символа-разделителя, после которого по накопленным данным происходит распознавание воспринятого объекта (будь то действие или объект среды).
4. Если распознанный образ – это действие, то он просто передается базе знаний, если был распознан образ объекта, то ему сопоставляется оценка, и пара {образ, оценка} передается базе знаний.
5. База знаний накапливает в себе информацию, поступающую от ФРО, в виде последовательности троек $O_n - A_k - O_{n+1}$, где O_n – сложный образ, составленный из последовательности элементарных образов, пришедших на вход БЗ между действиями.
6. В БЗ встроен модуль памяти, который осуществляет распознавание и хранение цепочек образов O_i , частот их повторения, вероятности их появления и весов.

В режиме управления цикл работы системы выглядит следующим образом.

1. Данные от сенсора о текущем символе поступают к датчикам.
2. Датчики воспринимают символ и передают его системе ФРО.
3. В системе ФРО накапливается информация до получения символа-разделителя, после которого по накопленным данным происходит распознавание воспринятого объекта.
4. Распознанный образ передается базе знаний.
5. База знаний накапливает в себе информацию, поступающую от ФРО, и с поступлением каждого нового образа пытается распознать (спрогнозировать) образ целиком, опираясь на информацию модуля памяти. Для того чтобы убедиться в достоверности прогноза, обращения к модулю памяти происходят каждый раз, когда появляется новый образ, и если несколько раз

подряд память ответила одинаково утвердительно, сложный O_i образ считается распознанным.

6. Среди сохраненных цепочек $O_n - A_k - O_{n+1}$ ищутся цепочки, начинающиеся на O_i , и среди возможных действий выбирается то, которое приводит к образу с наибольшей вероятностью появления и наибольшим весом. Если среди сохраненных цепочек не оказалось начинающихся на O_i , то действие выбирается случайно.

Созданная модель управляющей системы продемонстрировала свою жизнеспособность, показав умение выделять из входящего потока протяженные регулярности, заложенные в модель среды. Графики накопления веса (суммирование весов, встретившихся на пути УС образов) обученной и необученной систем выглядят следующим образом:

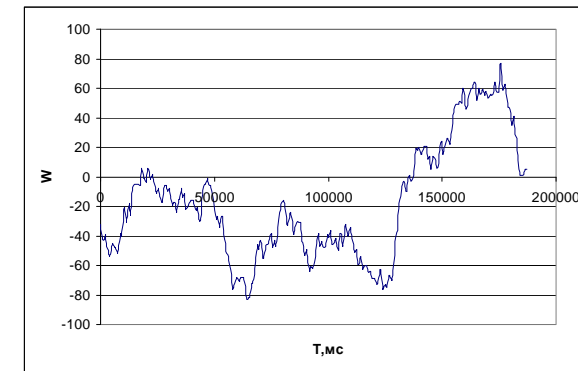


Рис 13. Зависимость накопленного веса от времени. Режим Обучения.

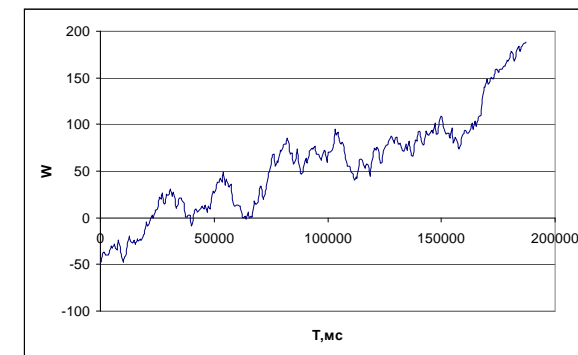


Рис 14. Зависимость накопленного веса от времени. Режим Управления.

Если считать, что на распознавание одного образа системой ФРО тратится время порядка 10 мс, то зависимости накопления веса от времени для различных модулей памяти будут следующими:

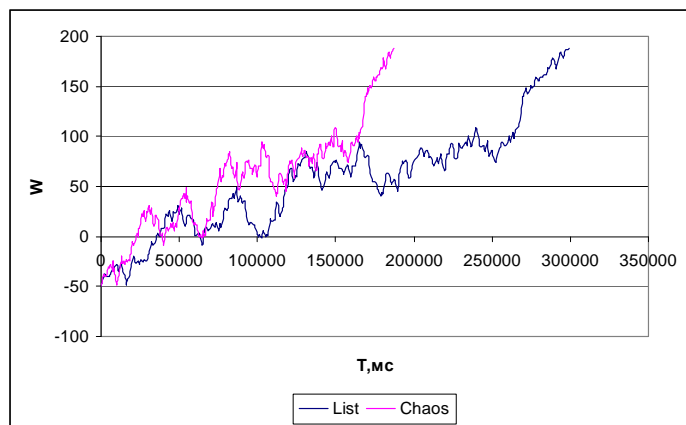


Рис 15. Зависимость накопления веса от времени в случае использования различных модулей памяти.

5. Заключение

В заключение коротко перечислим представленные в настоящей работе результаты. Основной результат состоит в том, что впервые было осуществлено слияние двух независимых технологий, обогащающее каждую из них, и расширяющее возможности их практического применения. Первая из технологий – это разрабатываемый авторами метод построения систем автономного адаптивного управления (ААУ) [22-37]. Вторая технология – способ использования детерминированного хаоса для запоминания больших массивов упорядоченной информации с ассоциативным способом обращения к ней, взятая нами из работ [3, 16-19].

Идея слияния состояла в следующем. Управляющая система ААУ в своей работе осуществляет запись в память последовательных во времени событий вида («образы, описывающие состояние объекта управления в предыстории текущего момента» → «последнее действие, которое совершила управляющая система» → «образы результирующего состояния объекта управления» → «оценка качества этого результирующего состояния»). Среди этих последовательностей отыскиваются неслучайные последовательности, которые переписываются в базу знаний управляющей системы. При принятии решения управляющая система в случае распознавания начала такой цепочки, анализирует возможные варианты ее продолжения, связанные с альтернативными действиями, и, используя оценки качества прогнозируемых состояний, выбирает действие, наиболее

оптимальное в текущих условиях. Очевидно, что свойства такой системы управления становятся тем более интересными и для теории и для практики, чем длиннее предыстория, характеризующая текущее состояние объекта, которую может проанализировать управляющая система, и чем дальше в будущее простирается прогноз, который может сделать управляющая система, принимая решение. Однако организация памяти, способной запоминать длинные цепочки событий, является проблемой. Так ранее системы ААУ, которые разрабатывались авторами, могли работать с предысторией всего в несколько тактов дискретного времени, и прогнозировать всего на один-два такта вперед. При этом память системы была построена в виде массивов, либо в виде специальной многослойной нейроноподобной сети. Несмотря на то, что возможности нейросетевого подхода далеко не исчерпаны, мы искали способ радикального продвижения в этой области. В технологии детерминированного хаоса авторы увидели возможность организации памяти для хранения упорядоченных последовательностей данных, длина которых может измеряться величиной в несколько порядков.

Авторам удалось применить технологию детерминированного хаоса для построения памяти управляющей системы ААУ, играющей роль базы знаний. Это позволило управляющей системе автоматически выявлять протяженные во времени закономерности событий, составляющие как предысторию совершаемого действия, так и его последствия, в случае, если таковые действительно имеются в той среде, в которой действует объект управления. Это и составляет основной результат настоящей работы.

Для проверки свойств такой управляющей системы был разработан модельный пример, представляющий собой робота, движущегося в пространстве - лабиринте, состоящем из последовательностей наблюдаемых символов, в котором встречаются закономерные длинные повторяющиеся фрагменты. Показано, что робот может адаптироваться к таким фрагментам, т.е. при встрече с ними узнавать их и выбирать наиболее выгодные варианты продолжения своего пути на основе обоснованного прогнозирования.

Среди частных полученных результатов, назовем следующие. Показано, что динамические характеристики модуля памяти, построенного на технологии детерминированного хаоса, являются приемлемыми для практического использования. Высокая скорость воспроизведения информации и ее кодирование, состоящее в извлечении из входного потока повторяющихся паттернов, позволяют эффективно использовать его в системах ААУ, где он может выполнять как функции системы ФРО, так и функции базы знаний. Недостатки, связанные с относительно большим временем запоминания новой информации, выглядят приемлемыми, поскольку они проявляются лишь на этапе обучения, менее критичном ко времени, чем режим управления. Тем не менее, проблема поиска более эффективного метода кодирования информации с целью

уменьшения времени записи новой информации, является актуальной и может стать темой дальнейших исследований.

Возможности построения систем ААУ с использованием подсистем памяти и распознавания на основе технологий детерминированного хаоса, расширяют сферу практического использования таких систем. Существует обширное количество задач, связанных с анализом длинных цепочек входной информации. В частности, применение таких систем возможно в химии для анализа спектральных паттернов, в анализе временных рядов, эргономике и психологии для анализа паттернов поведения человека, при управлении различными техническими системами и технологическими процессами.

С другой стороны, на наш взгляд, полученные в настоящей работе результаты могут расширить направления развития теории систем детерминированного хаоса, поскольку показывают возможности построения на его основе систем искусственного интеллекта, именно, бионических систем управления, имитирующих естественные нервные системы.

6. Направления дальнейших исследований

Как отмечалось выше, настоящая работа представляет лишь начальные результаты исследования в данном направлении. В дальнейших исследованиях планируется рассмотреть следующие аспекты.

1. Исследовать возможности реализации данной технологии в системах ААУ в других контекстах.
2. Исследовать возможности реализации более оптимальных форм хранения информации для хаотических процессоров с целью улучшения их динамических характеристик
3. Исследовать теоретические ограничения применимости хаотических процессоров, принимая во внимания такие параметры, как длина подстроки-запроса, длина алфавита, длина запоминаемых паттернов, количество сохраненных паттернов и степень «похожести» запоминаемых паттернов и др.

Литература

1. Artificial Neural Networks: Concepts and Theory, IEEE Computer Society Press, 1992.
2. Ф. Уоссермен, Нейрокомпьютерная техника, М., Мир, 1992.
3. Ю.В. Андреев, А.С. Дмитриев, Д.А. Куминов – Хаотические процессоры – Москва. Успехи современной радиоэлектроники, N10, 1997
4. Осовец С.М., Гинзбург Д.А. Груфинкель В.С, Зенков Л.Р., Латаш Л.П, Малкин В.Б., Мельничук П.В., Пастернак Е.Б. Электрическая активность мозга: механизмы и интерпретация. - УФН, 1983, т. 141, № 1, с.103-150
5. Rapp P.E., Zimmerman I.D., Albano A.M., Deguzman G.C., Greenbaum N.N. Dynamics of spontaneous neural activity in the simian cortex: the dimension of chaos neurons. - Phys Lett. A, 1985, vol. 110, no.6, p335-338

6. Bablyoantz A., Salazar J.M., and Nicolis G. Evidence of chaotic dynamics of brain activity during the sleep cycle. - Phys. Lett. A. 1985, vol 132, pp 101-106
7. Bablyoantz A., Destexhe A. Low-dimensional chaos in an instance of epilepsy. - Proc. Natl. Acad. Sci. USA 1986, vol. 83, pp. 3515-3517
8. Destexhe A., Sepulchre J.A., and Babyloyantz A. A comparative study of the experimental quantification of deterministic chaos. - Phys. Lett. 1988, vol. 132, pp 101-106
9. Bablyoantz A., Destexhe A. - in: Temporal Disorder in Human Oscillatory Systems. Springer Series in Synergetics, no. 36, Eds. Rensing L., Van der Heiden U., and Mackey M.C. Berlin. Springer, 1987, p 48
10. Шарковский А.Н., Майстренко Ю.Л., Романенко Е.Ю. Разностные уравнения и их приложения. - Киев: Наука.думка, 1986
11. Шарковский А.Н., Коляда С.Ф., Сивак А.Г., Федоренко В.В. Динамика одномерных отображений. - Киев: Наукова думка, 1989
12. Procaccia I. Universalities in Condensed Matter. - Proc Phys., vol. 32, Ed. Jullien R., Springer, 1988, p 213
13. Weigrinch W. and Tennekes H. On the information flow for one-dimensional maps. – Phys. Lett. A, 1990, vol. 144, no 3
14. Шустер Г. Детерминированный хаос. Введение. – М: Мир, 1988
15. Atmanspacher H., Scheingraber H. Global scaling properties of the chaotic attractor reconstructed from the experimental data. – Phys. Rev. A, 1988, vol. 37, pp 1314-1322
16. Дмитриев В.А. Запись и восстановление информации в одномерных динамических системах. – Радиотехника и электроника. 1991, т. 36, N1, с101-108
17. Dmitriev A.S., Panas A.L., and Strakov S.O. Storing and recognition information based on stable cycles of one-dimensional maps. – Phys. Lett. A., 1991, vol.155, pp 494-499
18. Andreev Yu.V., Dmitriev A.S. and Starkov S.O. Information processing in 1-D systems with chaos. – IEEE Transaction on circuit and systems, 1997, vol. 44, pp. 21-28
19. Андреев Ю.В., Бельский Ю.Л., Дмитриев А.С. Запись и восстановление информации с использованием устойчивых циклов двумерных и многомерных отображений. – Радиотехника и электроника, 1994, т.39, с.114-123
20. Т. Кормен, Ч. Лейзерсон, Р. Ривест – Алгоритмы. Построение и анализ – МЦНМО, Москва, 1999.
21. Д.А. Страбыкин, А.А. Жданов. Адаптация диалога в цифровых измерительных приборах со встроенными микроЭВМ. Депонир. ЦНИИТЭИ приборостроения N4108-ИР88 от 23.02.88. 7 стр.
22. Жданов А.А. О подходе к моделированию управляемых объектов. Препринт ВЦ РАН СССР, М., 1991.
23. Zhdanov A.A.. Application of Pattern Recognition Procedure to the Acquisition and Use of Data in Control.// Pattern Recognition and Image Analysis vol.2, N2,1992. (ISSN: 1054-6618).

24. Zhdanov A.A.. A principle of Pattern Formation and Recognition.// Pattern Recognition and Image Analysis vol.2, N3,1992. (ISSN: 1054-6618).
25. Жданов А.А. Об одном имитационном подходе к адаптивному управлению. Сб. "Вопросы кибернетики". Научный совет по комплексной проблеме "Кибернетика" РАН. Вып. 2. М., 1996.
26. Жданов А. А. Формальная модель нейрона и нейросети в методологии автономного адаптивного управления. Сб. "Вопросы кибернетики". Научный совет по комплексной проблеме "Кибернетика" РАН. Вып. 3. М., 1997.
27. Жданов А.А. О понятии автономного искусственного интеллекта // Сб. научн. тр. Искусственный интеллект в технических системах. - М.: Гос.ИФТП. 1997.
28. Zhdanov A. A. About an Autonomous Adaptive Control Methodology. ISIC/CIRA/(ISAS'98), NIST, Gaithersburg, Maryland. September 14-17, 1998.
29. Zhdanov A. A. The Mathematical Models of Neuron and Neural Network in Autonomous Adaptive Control Methodology. WCCI'98(IJCNN'98), IEEE World Congress on Computational Intelligence, Anchorage, Alaska, May 4-9, 1998.
30. Жданов А. А., Норкин Н. А., Гуриев М. А. Некоторые практические приложения метода автономного адаптивного управления // Сб. научн. тр. Искусственный интеллект в технических системах. Вып. 19.- М.: Гос.ИФТП. 1998.
31. Жданов А. А., Винокуров А. Н. Система проектирования и исследования нейронных сетей "СПИНС" в методологии автономного адаптивного управления // Сб. научн. тр. Всероссийской научн.-техн. конференции "Нейроинформатика-99", ч. 1. - М.: МИФИ. 1999.
32. Рябовиков А. В., Жданов А. А., О некоторых формальных моделях нейронов. // Сб. научн. тр. Всероссийской научн.-техн. конференции "Нейроинформатика-99", ч. 1. - М.: МИФИ. 1999.
33. Жданов А. А., Арсеньев С.В. О некоторых приложениях принципа автономного адаптивного управления // Сб. научн. тр. Всероссийской научн.-техн. конференции "Нейроинформатика-99", ч. 3. - М.: МИФИ. 1999.
34. Жданов А. А., Винокуров А. Н. О нейросетевом методе "Автономного адаптивного управления", Сб. докл. V Всероссийской конференции "Нейрокомпьютеры и их применение (НКП-99)". М., 17-19 февраля 1999.
35. <http://www.ispras.ru/~zhdanov/>
36. Жданов А. А., Метод автономного адаптивного управления // Известия Академии Наук. Теория и системы управления, 1999, № 5, с. 127-134
37. Жданов А.А., Арсеньев С.В., Половников В.А. Об одной методологии автономного адаптивного управления. Труды института системного программирования РАН. 1999. Том 1. М.: Биоинформсервис, 2000.- С. 66-83.
38. Жданов А.А., Беляев Б.Б., Мамаев В.В. Использование принципа автономного адаптивного управления в системе угловой стабилизации космического аппарата «Спектр РГ» // Сб. научн. тр. Информационная бионика и моделирование. - М.: ГосИФТП, 1955. – С. 87 - 114.