

# Возможности оптимизации системы автономного адаптивного управления с помощью генетических алгоритмов

*Земских Л.В.*

**Аннотация.** В данной работе приводятся основные результаты разработки методики применения генетических алгоритмов для синтеза и оптимизации прикладных систем, создаваемых на основе методологии Автономного Адаптивного Управления.

## 1. Введение

Современную деятельность человека трудно представить без использования различных автоматических и автоматизированных устройств. Такие устройства получают все большее распространение во всех сферах человеческой деятельности – от бытовой техники до сложных систем управления производственными процессами.

Как известно, автоматические и автоматизированные устройства состоят из аппаратной части и системы управления. Традиционно системы управления конструировались на основе математических моделей объектов управления и их поведения в среде функционирования. В последние годы всё более широкое распространение получают системы управления, чьи принципы функционирования относятся к области работы со знаниями – к искусственному интеллекту (ИИ). К системам такого рода относятся экспертные системы, искусственные нейронные сети, системы с подкрепляющим обучением, системы на основе нечеткой логики, и т.п. [1-7] К развиваемым в этой области подходам также относится оригинальный подход, называемый методом Автономного Адаптивного Управления (ААУ) [8-10], развиваемый в отделе имитационных систем Института Системного Программирования РАН.

Методология ААУ позволяет создавать адаптивные системы управления для приложений, в которых создание точных математических моделей объекта управления существенно затруднено или невозможно. Используя входящие в состав системы ААУ подсистемы (блоки датчиков, исполнительных органов, систему формирования и распознавания образов, базу знаний, систему моделирования эмоций, блок принятия решений), система ААУ накапливает эмпирически найденный знаний о свойствах объекта управления и находит

способ управлять им. Подсистемы организованы в виде структуры, общей для всех систем ААУ. Подсистемы могут быть реализованы на основе различных подходов, в том числе и на основе сетей нейроноподобных элементов [9]. В настоящее время проблемы, связанные с последним вариантом реализации систем ААУ, являются актуальными для дальнейшего развития метода. В частности, при проектировании прикладных систем ААУ на основе сетей нейроноподобных элементов возникают задачи подбора параметров нейроноподобных элементов, проектирования топологий их сетей и задача оптимизации полученных конструкций.

Настоящая работа посвящена разработке методов синтеза систем Автономного Адаптивного Управления (ААУ) на основе сетей нейроноподобных элементов с помощью применения генетических алгоритмов (ГА).

Предложенные в работе методы синтеза могут быть использованы при построении систем ААУ для широкого класса практических приложений.

Одной из основных проблем, связанных с методологией ААУ, являлось отсутствие четких рекомендаций по синтезу и оптимизации параметров систем управления, разрабатываемых для конкретных прикладных задач. Наличие методики синтеза и оптимизации систем ААУ позволит получать более эффективные системы управления.

Системы ААУ являются новым видом нейроноподобных систем управления. Поскольку такого рода системы имитируют работу биологических нервных систем, то исследования в этой области опираются не только на знания теории управления, распознавания, статистики и других необходимых разделов математических наук, но и на знания основ нейрофизиологии, генетики, биологии развития. Концепция систем ААУ опирается на представление о структуре и функциях нервной системы, которые логически вынужденно следуют из тех условий, в которых существуют нервные системы. Согласно концепции ААУ, такая система управления должна одновременно решать несколько сложных задач, а именно: задачу автоматической классификации, распознавания, моделирования эмоций, получения, вывода и представления знаний, принятия решений и некоторые другие. К этим свойствам, которые характерны и для системы ААУ, относятся: адаптивность, обучение и управление в одном процессе, многокритериальность управления, независимость от математических моделей объектов управления, универсальность, пригодность для адаптивного управления разнообразными объектами – от технических и технологических процессов до социальных объектов.

Система ААУ состоит из нескольких функциональных подсистем, ответственных за выполнение каждой из перечисленных выше задач. Именно, это следующие подсистемы: блок датчиков, система формирования и распознавания образов, база знаний и система ее формирования, система принятия решения, блок исполнительных органов (актуаторов).

Все подсистемы могут быть реализованы на основе сетей специальных нейроноподобных элементов.

На сегодняшний день одной из основных нерешенных проблем в рамках методологии ААУ является отсутствие формализованной методики синтеза систем управления для прикладных задач. В виду этого обстоятельства, при синтезе прикладных систем ААУ, используются эвристические соображения и простые математические модели.

Синтез системы ААУ представляет собой задачу большой размерности, решить которую аналитически на данном этапе не удастся. Для ее решения требуется знать точную модель взаимосвязи параметров, определяющих систему, а также критерии, характеризующие успешность ее реализации. Такие характеристики можно получить только в результате наблюдений серии экспериментов по моделированию поведения системы. В виду отсутствия полной математической модели, связывающей параметры, определяющие систему ААУ с критериями качества ее реализации, задачу синтеза необходимо решать методом подбора параметров. Одним из самых перспективных направлений, реализующим метод подбора параметров, являются генетические алгоритмы.

Задачу синтеза систем ААУ можно рассматривать как задачу, состоящую из двух подзадач:

- задачи подбора оптимальных параметров блока датчиков и блока исполнительных органов (привода или актуаторов) в комплексе, и
- задачи подбора оптимальных параметров подсистем, входящих в состав непосредственно системы управления (подсистем формирования и распознавания образов, базы знаний, системы принятия решений).

## **2. Задача оптимизации блока датчиков и блока актуаторов системы Автономного Адаптивного Управления**

В настоящей части рассмотрена постановка в общем виде задачи подбора оптимальных параметров блока датчиков и блока актуаторов, предложена схема ее решения на основе ГА. Разработанные положения были применены к решению прикладной задачи оптимизации параметров блока датчиков и блока актуаторов автономного мобильного робота.

Система из блока датчиков и блока исполнителей рассматривается как интерфейсная система, осуществляющая информационное и физическое взаимодействие среды, объекта управления и системы управления. Блок датчиков вырабатывает данные о текущем состоянии таких объектов как «внешняя среда», «объект управления», «внутренняя среда объекта управления», осуществляет их предварительную обработку, преобразует в вид двоичного вектора и передает такой вектор системе управления. Система управления после обработки полученных входных данных вырабатывает

команду, поступающую на блок актуаторов, которые выполняют те или иные воздействия на среду.

В прикладных задачах, для которых осуществляется построение систем ААУ, должны быть известны типы и количество как датчиков, так и исполнительных органов, а также перечень параметров, которыми определяется их функционирование. В диссертационной работе рассмотрена задача подбора оптимальных значений параметров, определяющих функционирование системы датчиков и исполнительных органов, с помощью такого метода эволюционной оптимизации как генетические алгоритмы.

Система датчиков и исполнительных органов должна обеспечивать системе ААУ наблюдаемость и управляемость. В случае конкретной прикладной задачи эти требования можно сформулировать либо математически, либо в виде некоторых качественных эвристических соображений, которые организованы в логическую схему, определяющую выбор параметров датчиков и исполнителей.

Приведем основные определения первой главы, задающие описание датчиков, актуаторов и связанных с ними понятий в рамках метода ААУ. Предложенное описание позволило поставить в общем виде задачу оптимизации параметров датчиков и актуаторов системы ААУ.

Пусть имеется некоторая система, состоящая из одного или нескольких объектов. Каждый из объектов в системе, равно как и вся система в целом, характеризуется несколькими параметрами. Пусть всего имеется  $N_i$  таких параметров, каждый из которых принимает значения из некоторого множества  $\tilde{X}_i$ .

**Определение 1.** Множеством  $\tilde{X}$  состояний системы будем называть Декартово произведение  $\tilde{X} = \tilde{X}_1 \times \tilde{X}_2 \times \dots \times \tilde{X}_{N_i}$  множеств  $\tilde{X}_i$ , каждое из которых соответствует одному и только одному параметру системы.

Пусть множество  $X_i$  есть множество возможных значений некоторого наблюдаемого параметра или свойств объекта, для измерения которых можно использовать некоторый датчик.

**Определение 2.** Бинарным датчиком будем называть определенную на множестве  $X_i$  функцию  $F_d^i(x)$ , принимающую значения 0 или 1, и зависящую от  $k$  параметров  $a_i, i = \overline{1, k}$ .

Измеряемый параметр или объект, к которому применяется датчик, имеет свойства, принимающие значения из множества  $X_i$ . Пусть в системе имеется  $N$  датчиков.

**Определение 3.** Множеством наблюдения  $X$  блока из  $N$  датчиков будем называть Декартово произведение  $X = X_1 \times X_2 \times \dots \times X_N$  множеств  $X_i$ , каждое из которых соответствует одному и только одному датчику.

Блок датчиков имеет определенные правила его организации. Эти правила можно выразить математически через систему равенств и неравенств, которым должны удовлетворять параметры всех  $N$  датчиков системы:

$$\begin{cases} g_i^s(\bar{a}) = 0, i = \overline{1, l} \\ h_j^s(\bar{a}) > 0, j = \overline{1, k} \end{cases} \quad (1)$$

В системе (1)  $\bar{a}$  - совокупное множество параметров  $N$  датчиков, а  $g^s$  и  $h^s$  - функции, определяющие правила построения блока датчиков. Пусть число свободных параметров в (1) равно  $p$ . Обозначим их как  $y_i^s, i = \overline{1, p}$ .

Исполнительным органом (актуатором) в реальной физической системе является устройство, осуществляющее воздействие объекта управления на среду, или изменяющее способ воздействия объекта управления на среду. Такому устройству было дано следующее формальное определение.

**Определение 4.** Блоком актуаторов будем называть определенную на множестве команд  $\Theta = \{\theta_1, \theta_2, \dots, \theta_m\}$  функцию  $U(\theta)$ , реализующую различные варианты управлений (стратегий) и характеризуемую свободными параметрами  $y_i^a, i = \overline{1, q}$ , допустимые значения которых задаются системой неравенств (2):

$$\begin{cases} g_i^a(\bar{y}^a) = 0, i = \overline{1, l_2} \\ h_j^a(\bar{y}^a) > 0, j = \overline{1, k_2} \end{cases} \quad (2)$$

где  $\bar{y}^a$  -  $q$ -компонентный вектор параметров, а  $g^a$  и  $h^a$  - функции, описывающие допустимые сочетания параметров.

Пусть  $p+q=n$ . Сформируем  $n$ -ку

$$\|y_1, y_2, \dots, y_n\|, y_i = y_i^s, i = \overline{1, q}, y_i = y_{i-q}^a, i = \overline{q+1, n} \quad (3)$$

**Определение 5.** Множеством поиска будем называть множество  $Y$   $n$ -ок, определенных по правилу (3), компоненты которых удовлетворяют системам (1) и (2).

Множество поиска  $Y$  есть множество возможных сочетаний параметров датчиков и актуаторов.

На основе *Определения 1* введем понятие показаний блока датчиков.

**Определение 6.** Вектором показаний блока датчиков будем называть вектор  $\eta(x) = \|F_d^1(x_1), F_d^2(x_2), \dots, F_d^N(x_N)\|$ , где  $x = \|x_1, x_2, \dots, x_N\| \in X$

Так как  $F_d^i(x_i) \in \{0, 1\}$ , то всего возможно  $2^N$  векторов показаний датчиков, каждый из которых вызывается некоторым множеством точек в пространстве

наблюдений  $X$ . Для одной и той же точки  $x$ , значение  $\eta(x)$  будет зависеть от параметров  $y_i^s$ .

В прикладных задачах система датчиков может быть построена таким образом, что  $\eta(x)$  сможет принимать только  $\tilde{N} \leq 2^N$  значений.

В процессе работы системы, наблюдая некоторый вектор  $\eta$ , управляющая система может выдать одну из команд из множества  $\Theta$ , по совершении которой система будет наблюдать некоторый другой (результатирующий для совершенного действия) вектор  $\eta_r$ . Эксперт должен классифицировать возможные результаты совершения действий и приписать им оценки относительно каждого возможного действия. Классы результатов и оценки определяются спецификой прикладной задачи.

Так как множество возможных векторов  $\eta$  конечно, то его можно перенумеровать.

**Определение 7.** Событием  $\omega_i$  будем называть тот факт, что после совершения произвольного действия из  $\Theta$  (при некотором исходном векторе  $\eta$ ) вектор результата есть  $\eta_r = \eta_i$ . Множество  $\Omega$  событий  $\omega_i$  будем называть множеством различных событий.

По построению  $|\Omega| = \tilde{N}$ .

**Определение 8.** Классом событий будем называть подмножество  $r$  множества различных событий  $\Omega$ .

**Определение 9.** Системой классов событий, размера  $k \leq \tilde{N}$ , будем называть множество  $R_k = \left\{ r_i \mid \forall i: 1 \leq i \leq k, r_i \subseteq \Omega \text{ и } \bigcup_{i=1}^k r_i = \Omega, \text{ а } \bigcap_{i=1}^k r_i = \Lambda \right\}$  (разбиение множества  $\Omega$  на  $k$  непересекающихся подмножеств).

Для каждого вектора  $\eta$  и множества действий  $\Theta$  введем функцию штрафов  $f_{\eta_i}$ , выражающую соответствие результата совершения действия ожиданиям эксперта. Функция  $f_{\eta_i}$  задается с использованием таблицы (Табл.1.), и вычисляется по следующему алгоритму. Зафиксируем точку  $x_0$  из множества  $X$ , вызвавшую вектор  $\eta_i(x_0)$ . Для одних и тех же исходных  $x_0$  и  $\eta_i(x_0)$ , объект управления совершает поочередно все возможные действия из множества  $\Theta$ . После совершения действия, определяется вектор показаний датчиков и класс произошедшего события. Затем, по таблице штрафов определяется текущий штраф  $\varphi_i$ , находящийся на пересечении строки, соответствующей совершенному действию, и столбца, соответствующего классу произошедшего

события. Значение штрафной функции вычисляется как  $f_{\eta_i} = \sum_{k=1}^m \varphi_k$ , где  $\varphi_k$  - штраф, полученный при совершении действия  $\theta_k$ .

**Табл.1.** Таблица штрафов, используемая для вычисления значения штрафной функции. Если было совершено действие  $\theta_l$  и его результат  $\omega_i \in r_j$ , то  $\varphi_l = s_{lj}$ .

$\eta_i$	$r_1$	$r_2$	...	$r_k$
$\theta_1$	$s_{11}$	$s_{12}$	...	$s_{1k}$
$\theta_2$	$s_{21}$	$s_{22}$	...	$s_{2k}$
...	...	...	...	...
$\theta_m$	$s_{m1}$	$s_{m2}$	...	$s_{mk}$

Штрафы  $s_{ij}$  назначаются экспертом с соблюдением принципа – при совершении какого-либо действия, наиболее благоприятный класс событий получает наибольшую оценку.

Очевидно, что значения  $f_{\eta_i}$  зависят от  $x_0$  и от набора параметров  $y$ . Определим общую штрафную функцию  $F_s(x)$  как  $F_s(x) = f_{\eta_i}$ , при  $\eta(x) = \eta_i$ . При такой формализации, можно говорить о постановке задачи оптимизации в форме:

$$\int_X F_s(x) dx \rightarrow \max_{y \in Y} \quad (4)$$

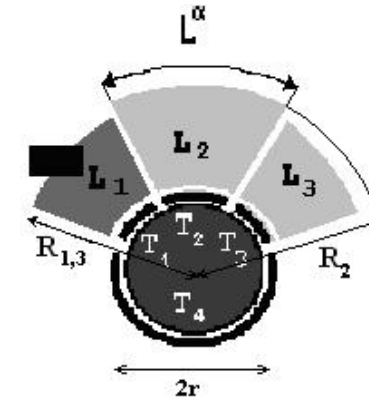
Необходимо отметить, что результат совершения системой того или иного действия зависит от точки из множества состояний  $\tilde{X}$  системы, по которой, затем, вычисляется значение точки  $x \in X$ . Таким образом, задачу оптимизации в полной форме необходимо рассматривать в виде

$$\int_{\tilde{X}} F_s(\tilde{F}(\tilde{x})) d\tilde{x} \rightarrow \max_{y \in Y} \quad (4a)$$

Постановка задачи (4) и (4a) подразумевает замену интеграла на суммы в случае, когда некоторые компоненты  $x$  дискретны. Зачастую вычисление точного значения такого интеграла трудно провести аналитически. В работе предлагается процедура оптимизации на основе генетического алгоритма (ГА). В предложенном ГА индивидом является точка в пространстве поиска  $Y$ , а значением приспособленности является значение интеграла в (4).

Предложенная постановка задачи в виде (4), применена к примеру постановки и решения задачи оптимизации блока датчиков и блока актуаторов программной модели автономного мобильного робота, имеющего целевую функцию – выработка стереотипов поведения при обходе стандартных случайно расставленных препятствий.

Моделируемый автономный мобильный робот [13] представляет собой круглый в плане объект, снабженный реверсивным двигателем и рулевым управлением, тремя визуальными и четырьмя тактильными датчиками. Робот движется с постоянной скоростью  $v$  по прямой, либо по окружности радиуса  $\rho$ , центр которой находится на прямой, проходящей через центр робота и перпендикулярной прямолинейному направлению движения. Система работает в потактовом режиме, с постоянной длительностью такта  $\Delta t$ . Визуальные датчики локаторного типа характеризуются углом раскрытия и длиной зоны видимости. Тактильные датчики расположены на корпусе робота и регистрируют факты соприкосновения корпуса робота с препятствиями. Визуальные датчики расположены таким образом, что центральный датчик направлен по направлению движения робота вперед, а боковые датчики направлены влево и вправо от него таким образом, что зоны обзора касаются, но не пересекаются. Общий угол раскрытия визуальных датчиков не превышает  $\pi$ . Три тактильных датчика по своему положению совпадают с пересечением зонами обзора визуальных датчиков корпуса робота. Четвертый датчик занимает оставшуюся часть корпуса.



Автономный мобильный робот имеет  $n=7$  датчиков и может совершать  $m=6$  действий. Радиус корпуса  $r=10$  фиксирован.

Параметрами, определяющими блок датчиков и исполнителей, являются:

угол раскрытия центрального визуального датчика ( $L_2^\alpha$ ).

скорость перемещения ( $v$ )

радиус поворота ( $\rho$ )

Перечисленные параметры являются свободными, а остальные являются связанными и вычисляются по формулам (5):

- Радиус видимости центрального датчика (от центра робота)  $R_2 = r + v\Delta t$ ; (5.1)

- Радиусы видимости боковых датчиков (от центра робота)  $R_1 = R_3 = r + 2\rho \sin\left(\frac{v\Delta t}{2\rho}\right)$ ; (5.2)

- Угол раскрытия боковых датчиков  $L_1^\alpha = L_3^\alpha = \frac{\pi}{2} - \frac{L_2^\alpha}{2} - \varphi$ , где  $\varphi = \arctg\left(\frac{x_0 y_0 - r\sqrt{x_0^2 + y_0^2 - r^2}}{x_0^2 - r^2}\right)$ ,  $x_0 = \rho\left(1 - \cos\left(\frac{v\Delta t}{\rho}\right)\right)$ , (5.3)

если  $x_0^2 + y_0^2 - r^2 > 0$  и  $\varphi=0$  иначе.

- Угол раскрытия заднего тактильного датчика  $T_4^\alpha = 2\pi - L_1^\alpha - L_2^\alpha - L_3^\alpha$  (5.4)

- Углы раскрытия передних тактильных датчиков и их ориентация равны соответствующим параметрам визуальных датчиков. (5.5)

- Положение правых границ датчиков:  $L_2^\beta = \frac{\pi}{2} - L_2^\alpha$ ,  $L_1^\beta = L_2^\beta + L_2^\alpha$ ,  $L_3^\beta = L_2^\beta - L_2^\alpha$  (5.6)

Свободные параметры удовлетворяют следующим ограничениям:

$$\begin{cases} \rho \geq r \\ 0 \leq v \leq \min\left\{v_0, \frac{\pi}{2} \frac{\rho}{\Delta t}\right\} \\ 0 \leq \frac{L_2^\alpha}{2} \leq \frac{\pi}{2} - \varphi \end{cases} \quad (6)$$

где  $v_0 > 0$  и  $\varphi$  вычисляется как указано в (5.3) формуле

Построена следующая схема испытаний.

Датчики поставляют управляющей системе сигналы в форме  $\eta = \|L_1, L_2, L_3, T_1, T_2, T_3, T_4\|$ ,  $L_i \in \{0,1\}$ ,  $T_i \in \{0,1\}$ . В задаче навигации робота существенным является то обстоятельство, что робот должен избегать столкновений с препятствиями. В процессе навигации для принятия решений о совершении того или иного маневра важно наличие препятствия в зоне видимости. Роль системы оповещения о близости препятствия выполняют только визуальные датчики. Тактильные датчики выполняют только роль регистраторов столкновений и их параметры являются зависимыми от параметров визуальных датчиков.

Построены таблицы штрафов только для тех входных векторов  $\eta$ , в которых  $T_i = 0$ ,  $i = \overline{1,4}$ . Для иных векторов построение таблиц не имеет смысла с точки зрения рассматриваемой задачи.

Было принято, что все препятствия выпуклы. Так как визуальный датчик не различает положение препятствия в зоне обзора, а в случае столкновения в контакте участвует только одна точка из границы препятствия, то положение препятствия в зоне обзора можно моделировать точечным препятствием. Случаи, когда препятствие попадает сразу в несколько зон обзора датчиков, можно рассматривать как суперпозицию отдельных срабатываний датчиков на точечные препятствия в зоне их видимости.

Таким образом, для построения схемы испытаний требовалось построить только четыре таблицы для векторов  $\|0,0,0,0,0,0,0\|$ ,  $\|1,0,0,0,0,0,0\|$ ,  $\|0,10,0,0,0,0,0\|$ ,  $\|0,0,1,0,0,0,0\|$ . Нулевой вектор необходим, так как возможны ситуации, в которых робот может столкнуться с препятствием при движении вперед (а визуальные датчики обязаны предупредить систему управления о наличии препятствия).

Определим классы событий, являющихся результатами совершения действий. Будем рассматривать только два класса:  $r_1$  - «столкновение» и  $r_2$  - «нет столкновения». Событием  $\omega_i$  будем считать тот факт, что после совершения произвольного маневра, вектор показаний  $\eta_r$  принимает конкретное значение  $\eta_i$  ( $\omega_i = \{\eta_r = \eta_i\}$ ) из множества возможных исходов  $\Psi$ , где

$$\Psi = \{\eta_i\} = \{\|0,0,0,0,0,0,0\|, \|1,0,0,0,0,0,0\|, \|0,1,0,0,0,0,0\|, \|0,0,1,0,0,0,0\|, \|1,0,0,1,0,0,0\|, \|0,1,0,0,1,0,0\|, \|0,0,1,0,0,1,0\|, \|0,0,0,0,0,0,1\|\}$$

Будем говорить, что  $\omega_i \in r_1$ , если для  $\eta_i = \|L_1, L_2, L_3, T_1, T_2, T_3, T_4\| \exists j : T_j = 1$ .  $\omega_i \in r_2$  во всех остальных случаях.

Для схемы испытаний были определены следующие маневры. В виду того, что система оповещения рассчитана на предупреждение о препятствиях,

находящихся спереди, все движения назад опускаются, так как они заведомо не приводят к столкновениям с препятствиями, находящимися спереди.

Были предложены следующие экспертные штрафные таблицы

$\eta = \parallel 0,0,0,0,0,0,0 \parallel$	$r_1$	$r_2$
FL	-2	0
FF	-2	0
FR	-2	0

$\eta = \parallel 1,0,0,0,0,0,0 \parallel$	$r_1$	$r_2$
FL	+3	-3
FF	-1	0
FR	-1	0

$\eta = \parallel 0,1,0,0,0,0,0 \parallel$	$r_1$	$r_2$
FL	-1	0
FF	+3	-3
FR	-1	0

$\eta = \parallel 0,0,1,0,0,0,0 \parallel$	$r_1$	$r_2$
FL	-1	0
FF	-1	0
FR	+3	-3

Множество  $X$  представляет собой зоны обзора визуальных датчиков и ту часть следа от движения робота (передних поворотов), которая в эту зону не попадает. Для того, чтобы какой-либо из датчиков отреагировал на объект, последний должен находиться в зоне  $\tilde{X}$ , границы которой в полярных координатах выражаются следующим способом:

$$\begin{cases} r \leq \rho \leq \rho_0 \cos \phi - \sqrt{\rho_0^2 \cos^2 \phi - 2r\rho_0 + r^2}, & 0 \leq \phi < \phi_0 \\ r \leq \rho \leq R_1, & \phi_0 \leq \phi < \phi_0 + L_3^\alpha \\ r \leq \rho \leq \min\left\{R_2, r + \frac{\pi}{2} \rho_0\right\}, & \phi_0 + L_3^\alpha \leq \phi < \phi_0 + L_2^\alpha + L_3^\alpha \\ r \leq \rho \leq R_3, & \phi_0 + L_2^\alpha + L_3^\alpha \leq \phi < \phi_0 + L_1^\alpha + L_2^\alpha + L_3^\alpha \\ r \leq \rho \leq \rho_0 \cos(\pi - \phi) - \sqrt{\rho_0^2 \cos^2(\pi - \phi) - 2r\rho_0 + r^2}, & \phi_0 + L_1^\alpha + L_2^\alpha + L_3^\alpha \leq \phi \leq \pi \end{cases} \quad (7)$$

где

$$\phi_0 = \arctg\left(\frac{x_0 y_0 - r \sqrt{x_0^2 + y_0^2 - r^2}}{x_0^2 - r^2}\right), \quad x_0 = \rho_0 \left(1 - \cos\left(\frac{v\Delta t}{\rho_0}\right)\right), \quad y_0 = \rho_0 \sin\left(\frac{v\Delta t}{\rho_0}\right),$$

$\rho_0, L_2^\alpha$  и  $v$  удовлетворяют ограничениям (6), а  $R_1, R_2, R_3, L_1^\alpha$  и  $L_3^\alpha$  определяются соотношениями (5).

Так как положение объекта в зоне  $\tilde{X}$  однозначно определяет показания датчиков, то интегрирование по  $X$  в (4) можно заменить интегрированием по множеству  $\tilde{X}$ , учитывая тот факт, что само значение интеграла в контексте решения задачи оптимизации несущественно. Интегрирование по множеству

$\tilde{X}$  осуществлялось по методу Монете-Карло с использованием подслучайных сеток Соболя в двумерном случае.

Предложенный алгоритм вычислений интеграла для некоторых параметров  $y \in Y$  состоит в следующем:

1. Рассчитать площадь  $S_X$  области интегрирования (области  $\tilde{X}$ )
2. Повторить  $N$  раз
  - 2.1. Получить точку  $x$  из псевдослучайной последовательности точек сетки Соболя
  - 2.2. Определить показания датчиков  $\eta(x)$
  - 2.3. Повторить для каждого  $\theta_i$  из  $\Theta$ 
    - 2.3.1. Совершить маневр  $\theta_i$
    - 2.3.2. Определить показания датчиков  $\eta_r$  и класс  $r_k$  события  $\omega_i$
    - 2.3.3. Определить штраф  $\varphi_i$
    - 2.3.4. Прирастить значение штрафной функции  $f_{\eta_i}$  на сумму штрафа  $\varphi_i$
    - 2.3.5. Занять исходную позицию.
  - 2.4. Прирастить интегральную сумму  $S_i$  на значение штрафной функции  $f_{\eta_i}$ .
  - 2.5. Прирастить квадратичную сумму  $S_i^2$  на квадрат значения штрафной функции  $f_{\eta_i}$ .
  - 2.6. Обнулить  $f_{\eta_i}$
3. Вычислить значение интеграла по формуле  $I = \frac{S_i}{N} S_X$ ,
4. Вычислить дисперсию по формуле  $Var = S_X \sqrt{\frac{\langle S_i^2 \rangle - \langle S_i \rangle^2}{N}}$ , где  $\langle \cdot \rangle$  - знак усреднения.
5. Выдать результат вычислений.

Для решения поставленной задачи оптимизации применялся генетический алгоритм, построенный следующим образом.

Фенотипом является объект, реализующий конструкцию и функционал движения робота. Для кодирования хромосомы использовалось машинное представление числа с плавающей точкой. Генотип состоял из трех хромосом, соответствовавших закодированному представлению свободных параметров  $\rho$ ,  $v$  и  $L_2^\alpha$ .

Точки в пространстве  $X$  были представлены как трехкомпонентные векторы. Оператор скрещивания (кроссинговер) определен как двуместный оператор  $C(x1,x2)$ :  $\tilde{x} = C(x_1, x_2) = x_1 + (x_2 - x_1) \cdot \left(\frac{3}{2}c - \frac{1}{4}\right)$ , где  $c$  – случайная переменная, равномерно распределенная на отрезке  $[0,1]$ . Оператор  $C(x1,x2)$  применялся к одним и тем же значениям  $x_1$  и  $x_2$  до тех пор, пока не удовлетворялось условие попадания точки  $\tilde{x}$  в множество  $X$ .

Оператор мутации реализован как одноместный оператор  $M(x)$ :  $\tilde{x} = M(x) = \begin{cases} \|\tilde{x}_1, \tilde{x}_2, \tilde{x}_3\|, & \text{если } c \leq P_{мут} \\ x, & \text{если } c > P_{мут} \end{cases}$ , где  $c$  – случайная переменная,

равномерно распределенная на отрезке  $[0,1]$ ,  $P_{мут}$  – вероятность мутации, а  $\tilde{x}_i \in N(x_i, \sigma_i)$ , где  $\sigma_i$  вычисляются как:

$$\begin{aligned} \sigma_1 &= \frac{y_1 - r}{10} \\ \sigma_2 &= \frac{1}{10} \min\left\{v_0, \frac{\pi}{2} \rho\right\} \\ \sigma_3 &= \frac{1}{10} \left(\frac{\pi}{2} - \varphi\right) \end{aligned} \quad (8)$$

Отбор индивидов для воспроизводства производился по следующей схеме: индивид с наибольшим значением функции приспособленности сразу же переходит в новую популяцию (элитизм). Затем для всех индивидов в популяции рассчитывается вероятность отбора для воспроизводства

$$p_i = \frac{f_i}{\sum_{i=1}^N f_i}$$

принципу «рулетки» (выбор индивида номер  $i$  с вероятностью  $p_i$ ). При этом не допускается повторное вхождение одного и того же индивида в одну пару (самовоспроизводство). После отбора проводится операция скрещивания, и к вновь полученным особям применяется оператор мутации.

Для решения задачи оптимизации использовалась популяция из 100 индивидов. Значение функции приспособленности каждого индивида вычислялось по 50000 точек интегрирования. Последний параметр генетического алгоритма – вероятность мутации  $P_{мут}$  – был определен как  $P_{мут}=0,05$ . Критерием останова алгоритма являлось число итераций  $N=500$ . График изменения лучшего и среднего значения приспособленности представлен на Рис.1. На рисунке можно видеть, что значение приспособленности лучшего индивида в популяции возрастает с увеличением числа поколений, что характеризует способность

алгоритма находить квазиоптимальное решение. В результате применения разработанной методики оптимизации были получены оптимальные значения параметров датчиков и актуаторов для случая с ограничением скорости  $v_0=35$  и радиусом корпуса робота  $r=10$  (в единицах масштаба. Рис. 2).

Такое ограничение скорости соответствует требованию оптимизации датчиков и привода для навигации робота в лабиринтах, в которых среднее расстояние между препятствиями не меньше  $v_0$ .

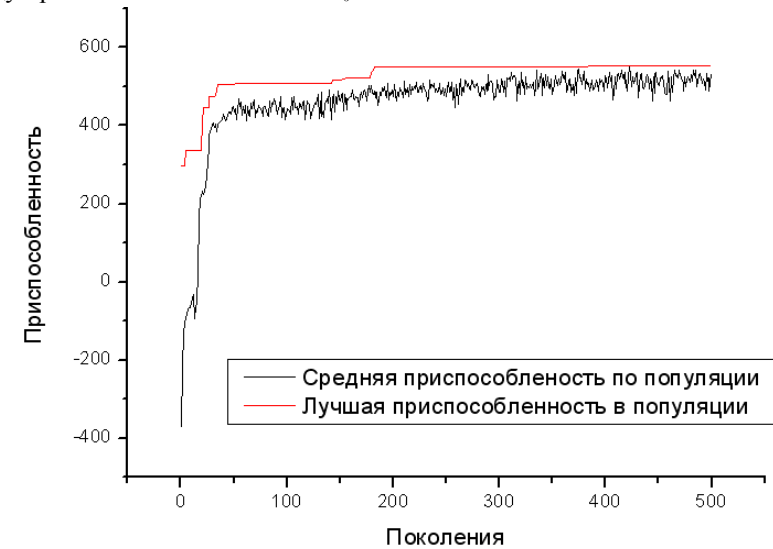
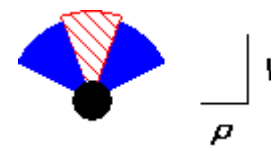


Рис. 1. График изменения лучшей и средней приспособленности в популяции в зависимости от номера поколения.



- $V=34,90$
- $\rho = 22,57$
- $L_2^\alpha = 0,7052$
- $L_1^\alpha = L_3^\alpha = 0,7434$
- $R_2= 44,90$
- $R_1=R_3= 41,52$

Рис. 2. Оптимальные параметры датчиков и актуаторов, найденные в результате решения задачи оптимизации датчиков и актуаторов.

### 3. Процедура динамического формирования сети формальных нейроноподобных элементов для аппарата Формирования и Распознавания образов и Базы Знаний систем Автономного Адаптивного Управления

В настоящей части представлен формализм для описания топологии сетей нейроноподобных элементов подсистемы формирования и распознавания образов системы ААУ, а также представлена процедура динамического формирования такой сети в рамках введенного формализма. Некоторые обоснования необходимости введения процедуры эволюционной оптимизации предложены в [14].

Пусть у управляющей системы имеется  $N_0$  входных датчиков, каждый из которых выдает значение из множества  $\{0,1\}$ . Совокупность  $N_0$  сигналов со всех датчиков представим в виде  $N_0$  – компонентного вектора  $\eta_i$ , который является входным вектором для управляющей системы. Аналогично соображениям, изложенным в 1-й главе, система может воспринять  $\tilde{N} \leq 2^{N_0}$  входных векторов. Пусть множество  $X$  есть множество всех входных векторов  $\eta_i$  ( $|X| = \tilde{N}$ ). Снабдим каждый датчик двумя выходными полюсами, один из которых выдает единичный сигнал в случае срабатывания датчика (прямой сигнал), а другой выдает единичный сигнал в противном случае (инверсный сигнал). В управляющей системе, подсистемой, ответственной за обработку входных векторов, является система формирования и распознавания образов, а выходные полюса датчиков являются ее входными полюсами. Следовательно, у системы формирования и распознавания образов имеется  $2N_0$  входных полюсов.

Приведем определения и понятия формального описания топологии сети нейроноподобных элементов подсистемы формирования и распознавания образов. В работе использовались нейроноподобные элементы, описанные в [9]. Эти нейроноподобные элементы в главных чертах отличаются от «формальных нейронов», используемых в современных искусственных нейросетях тем, что а) являются самообучаемыми распознающими элементами, статистическим способом обнаруживающими коррелирующие входные сигналы (образы), запоминающими и распознающими их при последующих наблюдениях, б) имеют временную задержку срабатывания, в) автоматически снижают активационный порог при накоплении статистики.

В данной работе подключение нейроноподобных элементов к датчикам было организовано следующим способом. Образовывалось  $\tilde{N} \leq 2^{N_0}$  нейроноподобных элементов, входная арность каждого из которых равна  $N_0$ , и каждому из них был поставлен в соответствие один и только один вектор входных сигналов  $\eta_i$ . Чтобы вектор входных сигналов  $\eta_i$  вызывал

срабатывание соответствующего ему нейроноподобного элемента, он соединяется с полюсами датчиков следующим способом. Если  $j$ -я компонента вектора  $\eta_i$  равна 1, то организуется соединение от прямого выходного полюса  $j$ -го датчика к нейроноподобному элементу. В противном случае организовывалось соединение с инверсным выходным полюсом.

**Определение 8.** Структурой связей  $i$  нейроноподобного элемента к датчикам будем называть такую организацию связей от полюсов датчиков к нейроноподобному элементу, при которой, при наличии на датчиках выходного вектора  $\eta_i$  все связи берут начало на полюсах, выдающих единичный сигнал, и оканчиваются на входе нейроноподобного элемента.

**Определение 9.** Если на датчиках наблюдается выходной вектор  $\eta_i$ , то будем говорить, что состояние входов системы формирования и распознавания образов есть  $\eta_i$ .

Для соединения нейроноподобных элементов с полюсами датчиков в сети был использован способ, приведенный в Определении 8.

**Определение 10.** Будем говорить, что нейроноподобный элемент находится в слое  $K$  ( $K > 1$ ) и имеет тип  $i$ , если его структура связей с датчиками имеет номер  $i$ , и кроме того нейроноподобный элемент имеет дополнительную (и только одну) входящую связь от нейроноподобного элемента, находящегося в слое  $K-1$ . Нейроноподобный элемент слоя 1 типа  $i$  имеет только  $N_0$  входящих связей от полюсов датчиков, структура которых имеет номер  $i$ .

Рассмотрим сеть произвольной (в рамках определения 10) топологии. На каждом такте работы, система формирования и распознавания образов осуществляет:

Процедуру распространения сигнала в сети (от датчиков к элементам сети и между соседними слоями).

Обработку сигналов нейроноподобными элементами.

Пусть все нейроноподобные элементы в сети обучены. Срабатывание элемента типа  $i$  в момент времени  $t$  означает, что в этот момент времени на датчиках находится сигнал  $\eta_i$ . Если при этом нейроноподобный элемент находится в слое  $K$  ( $K > 1$ ), то в момент времени  $t-1$  сработал элемент в слое  $K-1$ , от которого рассматриваемый элемент имеет входящую связь. Если такой элемент в слое  $K-1$  имеет тип  $j$ , то в момент времени  $t-1$  на датчиках находился сигнал  $\eta_j$ . Для  $K=2$ , факт срабатывания элемента в этом слое означает распознавание системой формирования и распознавания образов последовательности из двух сигналов  $(\eta_i, \eta_j)$ . Распространяя эти рассуждения на  $K > 2$ , получаем, что срабатывание элемента в слое  $K$  означает распознавание последовательности из  $K$  входных векторов. Конкретный вид последовательности определяется типами элементов, вызвавших распознавание.



В соответствии с методологией ААУ, факт обучения нейроноподобного элемента сети отождествляется с фактом формирования образа, а срабатывание элемента - с фактом его распознавания.

На основе указанного свойства распознавания последовательностей входных сигналов и принципа использования для управления адекватной информации, был сформулирован и доказан ряд утверждений.

**Утверждение 1.** Описанный в *Определении 10* способ соединения нейроноподобных элементов достаточен для построения системы распознавания, способной распознать любой наперед заданный конечный набор конечных последовательностей, существующих в потоке входных векторов.

**Утверждение 2.** Для любой сети нейроноподобных элементов, построенной в рамках методологии ААУ, не содержащей циклов, можно построить аналогичную по функциональности сеть с топологией, организуемой согласно *Определению 10*.

Пусть оператор  $L$  определен на множестве конечных последовательностей входных векторов длины больше чем 1. Оператор  $L$  ставит в соответствие последовательности  $\tilde{\eta}$  последовательность  $\tilde{\eta}'$ , получаемую из  $\tilde{\eta}$  удалением ее последнего члена. Если длина последовательности  $\tilde{\eta}$  равна  $l$ , то с помощью оператора  $L$  из  $\tilde{\eta}$  можно получить  $l-1$  дополнительную последовательность путем многократного применения оператора  $L$ . Введем в рассмотрение расширение над множеством  $\Delta$ , сформулировав следующее определение.

**Определение 11.** Расширением над множеством  $\Delta$  несовпадающих последовательностей входных векторов будем называть множество  $\Delta'$ , содержащее в себе все элементы из  $\Delta$ , а также для каждой последовательности  $\tilde{\eta} \in \Delta$  длиной больше 1, все последовательности, получаемые из  $\tilde{\eta}$  последовательным применением оператора  $L$ :  $L\tilde{\eta}, L^2\tilde{\eta}, \dots, L^{l-1}\tilde{\eta}$ , где  $l$  – длина последовательности  $\tilde{\eta}$ , при этом, если  $\exists i, j, k, l: L^k\tilde{\eta}_i = L^l\tilde{\eta}_j$ , то  $L^k\tilde{\eta}_i$  и  $L^l\tilde{\eta}_j$  соответствуют одному и тому же элементу в  $\Delta'$ .

На основе предложенной схемы описания строения сети нейроноподобных элементов была предложена процедура динамического формирования таких сетей в процессе функционирования системы ААУ.

**Определение 12.** Графом переходов состояний входов системы будем называть ориентированный граф  $H_s$ , узлами которого являются состояния входов системы, а ребрами – разрешенные переходы между состояниями, допускающие переходы состояний в себя (петли).

Граф  $H_s$  задается матрицей смежности  $G$ , элементами которой являются 0 и 1. Наличие 1 в строке  $i$  и столбце  $j$  означает, что допускается переход из состояния  $\eta_i$  в состояние  $\eta_j$ , а наличие 0 запрещает такой переход. Граф  $H_s$

фиксируется на этапе синтеза системы и не изменяется в процессе ее функционирования. Он отражает представления исследователя о допустимых сменах одних входных векторов другими.

Динамическое формирование сети нейроноподобных элементов происходит по следующему алгоритму.

В начале функционирования системы ААУ, сеть нейроноподобных элементов системы формирования и распознавания образов состоит только из элементов первого слоя. В последствии в сеть добавляются элементы в слои с более высокими номерами. Входной сигнал  $\eta_i$  обрабатывается сетью, после чего выявляется множество сработавших нейроноподобных элементов. По построению, все сработавшие нейроноподобные элементы будут иметь тип  $i$ . Для каждого из сработавших элементов определяется наличие исходящих связей. Для всех элементов, не имеющих исходящих связей, формируются новые, расположенные в следующем за ними слое. Пусть сработавший элемент без исходящих связей находится в слое  $K$  и имеет тип  $i$ . В слое  $K+1$  размещается столько новых нейроноподобных элементов, сколько единиц находится в  $i$ -й строке матрицы смежности  $G$ . Тип подключения к датчикам для каждого нового элемента определяется номером столбца соответствующей ему единицы в  $i$ -й строке матрицы  $G$ . Дополнительная входящая связь организуется от рассматриваемого элемента в слое  $K$ .

Матрица  $G$  определяет правила динамического формирования сети. В случае, когда правила переходов между состояниями, определяемые матрицей, совпадают с действительно существующими в системе, структура сети нейроноподобных элементов будет способна распознавать наибольшее число последовательностей входных сигналов. При этом, если в прикладной задаче имеются ограничения на количество нейроноподобных элементов, то оптимальная матрица  $G$  обеспечивает построение такой топологии сети, в которой будет задействовано наибольшее число нейроноподобных элементов.

Пусть оператор  $\tilde{L}$ , определен на всех последовательностях  $\tilde{\eta} \in \Delta$ , длина которых больше 1 и ставит в соответствие последовательности  $\tilde{\eta} \in \Delta$  последовательность, получаемую из  $\tilde{\eta}$  удалением первого ее элемента.

**Определение 13.** Для последовательности  $\tilde{\eta} \in \Delta$  длины  $l$  *предысторией* длины  $l$  конечного элемента последовательности будем называть саму последовательность  $\tilde{\eta}$ . Если  $l > 1$ , то *предысторией* длины  $l-1$  того же элемента той же последовательности входных векторов будем называть последовательность  $\tilde{L}\tilde{\eta}$ .

**Определение 14.** Расширением с *предысторией* множества  $\Delta$  будем называть множество  $\Delta''$ , получаемое из  $\Delta$  по следующим правилам. Каждой последовательности  $\tilde{\eta} \in \Delta$  поставим в соответствие набор

последовательностей, получаемый последовательным применением оператора  $\tilde{L}$ , а именно  $\tilde{L}\tilde{\eta}, \tilde{L}^2\tilde{\eta}, \dots, \tilde{L}^{l-1}\tilde{\eta}$ . Сформируем множество  $\Delta_0$  присоединением к  $\Delta$  всех таких наборов. Множество  $\Delta''$  есть расширение множества  $\Delta_0$ , т.е.  $\Delta'' = \Delta'_0$ .

Алгоритм динамического формирования сети приводит к тому, что получаемая в результате его работы сеть обладает свойством, сформулированным как следующее утверждение.

**Утверждение 3.** Для некоторого множества  $\Delta$  последовательностей входных векторов, сеть нейроноподобных элементов, создаваемая в результате работы алгоритма динамического формирования, имеющего в качестве правил формирования сети матрицу  $G$ , получаемую по  $\Delta$ , способна распознавать все элементы множества расширения с предысторией  $\Delta''$ .

В диссертационной работе механизм формирования базы знаний реализован на основе вероятностного принятия решений. Параметры такого процесса определяются параметрами системы формирования и распознавания образов. Способ реализации системы принятия решений и базы знаний, основан на результатах более ранних исследований [14].

Пусть система формирования и распознавания образов способна распознавать  $n$  образов и совершать  $m$  действий. Подсистема базы знаний и подсистема принятия решений были представлены в виде единого блока, реализованного в виде матрицы  $\Gamma$  размерности  $n \times m$ , построенной следующим способом. Каждый столбец матрицы соответствует единственному из  $n$  образов, а каждая строка - единственному из  $m$  действий. Элементами  $p_{ij}$  матрицы  $\Gamma$  являются числа из

интервала  $[0, 1]$ , удовлетворяющие условию:  $\forall j \sum_{i=1}^n p_{ij} = 1$ . Элементы  $p_{ij}$

матрицы  $\Gamma$  являются вероятностями выбора действия номер  $i$  при распознавании системой образа номер  $j$ .

**Определение 15.** Совокупность значений элементов матрицы  $\Gamma$  будем называть *недетерминированным законом управления*.

**Определение 16.** *Опорным столбцом матрицы  $\Gamma$*  будем называть столбец, который будет использоваться для принятия решения на текущем такте работы системы.

Правила принятия решения для недетерминированного закона управления были определены следующим способом. При распознавании единственного образа, имеющего номер  $j$ , выбор  $i$ -го действия будет совершен с вероятностью  $p_{ij}$ . Если системой распознано  $q$  образов, то вероятности выбора действий рассчитываются по следующему алгоритму. Столбец номер  $j$  матрицы  $\Gamma$  будем представлять как вектор  $\vec{P}_j$ . Пусть последовательность номеров распознанных

образов есть  $\{j_k\}_{k=1}^q$ . Тогда вероятности выбора действий представляются

$$\text{вектором } \vec{P}_{\text{выб}} = \frac{\sum_{k=1}^q \vec{P}_{j_k}}{\left| \sum_{k=1}^q \vec{P}_{j_k} \right|}.$$

Процесс обучения для недетерминированного закона управления реализован по следующим правилам. Пусть в момент времени  $t-1$  было распознано  $q$  образов, и оценка состояния системы была  $S(t-1)$ , было принято решение совершить действие  $\theta_{i_0}$ , и в момент времени  $t$  система имеет оценку состояния  $S(t)$ . Для коррекции вероятностей выбора действий для каждого из  $q$  распознанных образов рассчитывается параметр изменения

$$\Delta p_{i_0} = a \frac{S(t) - S(t-1)}{|S_{\max} - S_{\min}|}, \text{ где } S_{\max} - \text{максимально возможная оценка состояния}$$

системы и  $a$  - коэффициент изменения вероятности ( $0 < a < 1$ ). Для каждого из  $q$  распознанных на предыдущем такте образов новые значения вероятностей

определяются по формуле  $\vec{P}_{i_k}^{\text{нов}} = \frac{\vec{P}_{i_k} + \Delta \vec{P}_{i_k}}{|\vec{P}_{i_k} + \Delta \vec{P}_{i_k}|}, k = \overline{1, q}$ , где  $\Delta \vec{P}_{i_k}$  есть вектор

размерности  $m$ , все компоненты которого равны 0, за исключением  $i_0$ -й, которая равна  $\Delta \vec{p}_{i_0} = \Delta p_{i_0} \vec{P}_{i_0}^{i_0}$ .

Предложенный здесь формализм описания архитектуры сети нейроноподобных элементов системы формирования и распознавания образов, построенная на его основе процедура динамического формирования таких сетей, а также механизм формирования базы знаний и принятия решений позволяют поставить в общем виде задачу оптимизации параметров подсистем управляющей системы ААУ.

#### 4. Генетический алгоритм подбора значений параметров процедуры динамического формирования сети нейроноподобных элементов системы ФРО

В настоящей части в общем виде представлена схема постановки задачи оптимизации для подбора правил формирования сети нейроноподобных элементов системы формирования и распознавания образов, а также постановка и решение такой задачи на примере автономного мобильного робота [11].

Для используемой при оптимизации оценки эффективности функционирования систем ААУ был предложен следующий критерий.

В каждый момент времени блок датчиков поставляет системе управления входной вектор  $\eta_i$ , который интерпретируется как состояние системы. Если все множество входных векторов есть  $\Psi$  ( $|\Psi| = \bar{N} \leq 2^{N_0}$ ), то его можно разбить на непересекающиеся подмножества  $\psi_i, i = \overline{1, k}, k \leq N_0$ , соответствующие некоторым классам состояний системы ААУ, характерных для прикладной задачи, для которой осуществляется оптимизация. Каждому подмножеству  $\psi_i$  приписывается оценка  $S_i$ , характеризующая благоприятность такого состояния системы с точки зрения некоторого заданного критерия качества. Одним из показателей удачности реализации системы служит средняя оценка состояния системы за весь период ее функционирования. Чем удачнее реализация, тем выше средняя оценка.

**Определение 17.** Средней оценкой за последние  $M_{\text{средн}}$  тактов будем называть число  $\xi_M(t) = \frac{1}{M_{\text{средн}}} \sum_{i=0}^{M_{\text{средн}}-1} s(t-i)$ , где  $s(t)$  - оценка состояния системы в момент времени  $t$ .

Для того чтобы сравнить различные системы ААУ использовался период времени от начала функционирования систем (в тактах), в течение которого, полагалось, что происходит основной этап формирования сетей нейроноподобных элементов подсистемы ФРО и накопление знаний в подсистеме БЗ. Такой период времени был назван *периодом обучения*  $T_{\text{обуч}}$ . Эффективность системы ААУ измерялась в течение некоторого периода времени  $T_{\text{стат}}$ , следующего непосредственно за  $T_{\text{обуч}}$ .

**Определение 18.** Оценкой эффективности по одному эксперименту системы ААУ будем называть число  $\xi_x(G) = \frac{1}{T_{\text{стат}}} \sum_{i=0}^{T_{\text{стат}}-1} \xi_M(T_{\text{обуч}} + i)$ , при условии, что  $T_{\text{обуч}} \geq M_{\text{средн}}$ , и система ААУ обладает матрицей переходов  $G$ .

**Определение 19.** Оценкой  $\xi(G)$  эффективности системы ААУ в среднем будем называть математическое ожидание оценки  $\xi_x(G)$  эффективности по одному эксперименту:  $\xi(G) = E(\xi_x(G))$ .

Предложенная в *Определении 18* оценка эффективности системы ААУ, обладающей матрицей  $G$ , зависит от начальных условий эксперимента. При изменении начальных условий, прочие параметры эксперимента -  $M_{\text{средн}}$ ,  $T_{\text{стат}}$ ,  $T_{\text{обуч}}$  - не изменяются. При этом, для  $\xi_x(G)$  характерна дисперсия  $\sigma^2(G) = D\xi_x(G)$ .

При проведении серии из  $N$  экспериментов выборочными оценками для  $\xi(G)$  и  $\sigma^2(G)$ , будут являться соответственно

$$\begin{aligned} \tilde{\xi}(G) &= \frac{1}{N} \sum_{i=1}^N \xi_{x,i}(G) \text{ и} \\ \tilde{\sigma}^2(G) &= \frac{1}{N-1} \sum_{i=1}^N (\xi_{x,i}(G) - \tilde{\xi}(G))^2 \end{aligned} \quad (9)$$

Для сравнения эффективности реализации системы ААУ был введен следующий критерий.

**Определение 20.** Для двух систем ААУ, начальные конфигурации которых совпадают во всем, кроме матриц смежности  $G_i$  и  $G_j$ , будем говорить, что система ААУ с матрицей  $G_i$  лучше, чем система ААУ с матрицей  $G_j$ , если  $\tilde{\xi}(G_i) > \tilde{\xi}(G_j)$ , и что системы эквивалентны в случае если  $\tilde{\xi}(G_i) = \tilde{\xi}(G_j)$ .

**Утверждение 4.** Для оценок  $\xi(G)$  существует верхний предел.

Создание системы ААУ на основе предложенного во второй главе формализма заключается в подборе матрицы переходов  $G$ . Для постановки задачи оптимизации и построения генетического алгоритма по подбору оптимальной матрицы  $G$  приведем несколько определений.

**Определение 21.** Пространством поиска  $Y^G$  будем называть множество матриц смежности  $G$  графа  $H_i$  переходов состояний входов системы.

Для подбора матрицы  $G$  при создании системы ААУ, на основе *Определений 19-21* была сформулирована следующая постановка задачи оптимизации:

$$\xi(G) \rightarrow \max_{G \in Y^G} \quad (10)$$

При проектировании системы ААУ исследователь, опираясь на свойства прикладной задачи, иногда может заранее указать некоторые ограничения относительно смены одних входных векторов другими, накладывая, тем самым, ограничения на возможный вид матриц  $G$ . Такие ограничения позволяют выделить из множества  $Y^G$  некоторое подмножество  $Y_0^G \subseteq Y^G$  матриц, имеющих смысл для рассматриваемой задачи. В связи с этим задачу (10) можно уточнить и представить в виде:

$$\xi(G) \rightarrow \max_{G \in Y_0^G \subseteq Y^G} \quad (11)$$

*Определение 20* позволяет сравнить любые две системы, определяемые матрицами  $G \in Y^G$ , и, следовательно, любые две, определяемые матрицами  $G \in Y_0^G \subseteq Y^G$ . Решением задач (10) и (11) будем считать множество  $Y^*$  эквивалентных систем, оценка которых максимальна на множествах  $Y^G$  и  $Y_0^G$  соответственно. Очевидно, что  $Y^* \subseteq Y_0^G \subseteq Y^G$ .

Приведенная задача оптимизации решается методом подбора. В качестве такого метода в работе были выбраны генетические алгоритмы, как эффективный метод эволюционной оптимизации нулевого порядка.

Приведенный в работе генетический алгоритм не зависит от особенностей оптимизируемой прикладной задачи. Индивидом является объект управления, оснащенный системой ААУ, имеющей в качестве правил динамического формирования сети нейроноподобных элементов матрицу  $G$  из множества  $Y_0^G$ . Алгоритм реализован по общей схеме, приведенной во 2-й главе. Способ формирования родительских пар остается неизменным. В качестве генома каждый индивид имеет матрицу  $G$ . Фенотипом является объект, реализующий саму систему ААУ. Генетические операторы имеют следующие особенности.

Оператор скрещивания (кроссинговер) - это двухместных оператор, определенный на множестве матриц  $Y_0^G$  для двух различных матриц  $G_1$  и  $G_2$ .

$$\tilde{G} = C(G_1, G_2),$$

$$\tilde{g}_{ij} = \begin{cases} g_{ij}^1, & \text{если } g_{ij}^1 = g_{ij}^2 \\ g_{ij}^1, & \text{если } g_{ij}^1 \neq g_{ij}^2 \text{ и } c \geq \frac{1}{2} \\ g_{ij}^2, & \text{если } g_{ij}^1 \neq g_{ij}^2 \text{ и } c < \frac{1}{2} \end{cases} \quad (12)$$

где  $\tilde{g}_{ij}$ ,  $g_{ij}^1$ ,  $g_{ij}^2$  - элементы матриц  $\tilde{G}$ ,  $G_1$  и  $G_2$  соответственно, а  $c$  - равномерно распределенная на отрезке  $[0, 1]$  случайная переменная, значение которой определяется для каждого сочетания индексов  $i$  и  $j$ .

Оператор мутации является одноместным оператором, определенным на множестве матриц  $Y_0^G$ . Формально он определяется следующим способом:

$M(G): Y_0^G \rightarrow Y_0^G$ , при этом

$$M(G) = \begin{cases} G, & \text{если } c_1 > P_{мут} \\ \tilde{G}, & \text{если } c_1 < P_{мут} \end{cases} \quad (13)$$

где  $c_1$  - случайная переменная, равномерно распределенная на отрезке  $[0, 1]$ .

Так же как и для оператора кроссинговера, конкретная реализация оператора мутации может зависеть от специфики прикладной задачи. В качестве простейшего оператора рассмотрим следующий. В случае если в (13)  $c_1 < P_{мут}$ , то элементы матрицы  $\tilde{G}$  определяются следующим способом:

$$\tilde{g}_{ij} = \begin{cases} g_{ij}, & \text{если } c_2 > \tilde{P}_{мут} \\ 1, & \text{если } c_2 < \tilde{P}_{мут} \text{ и } g_{ij} = 0 \\ 0, & \text{если } c_2 < \tilde{P}_{мут} \text{ и } g_{ij} = 1 \end{cases} \quad (14)$$

где  $c_2$  - случайная переменная, равномерно распределенная на отрезке  $[0, 1]$ ,  $g_{ij}$  - элементы матрицы  $G$ , а  $\tilde{P}_{мут} \ll 1$  - вероятность изменения элементов матрицы. Изменение значения элемента допускается, если новое значение матрицы не выводит ее из множества  $Y_0^G$ .

Для того, чтобы определить критерий останова генетического алгоритма, необходимо ввести следующие определения.

**Определение 22.** Расстоянием между матрицами  $G_1$  и  $G_2$  будем считать расстояние по Хеммингу  $\rho_G(G_1, G_2)$ , равное количеству несовпадающих по значениям элементов матриц  $G_1$  и  $G_2$ .

**Определение 23.** Если размер популяции равен  $N_{инд}$ , а матрица, имеющая наибольшее значение функции приспособленности, имеет номер  $i_0$ , то алгоритм считается завершенным, если  $\forall i \leq N_{инд}$  и  $i \neq i_0$   $\rho_G(G_i, G_{i_0}) < A$ .

В диссертационной работе рассмотрен пример постановки и решения задачи оптимизации параметров процедуры динамического формирования сети нейроноподобных элементов управляющей системы автономного мобильного робота. Автономный мобильный робот имеет устройство, описанное в 1-й главе, и конфигурацию блока датчиков и актуаторов, полученную в результате решения поставленной в той же главе задачи оптимизации.

Так как в данном примере датчиков всего 7, то система распознавания способна распознавать всего  $2^7=128$  входных сигналов. Таким образом, для рассмотренного примера матрица смежности  $G$  графа  $H_i$  перехода состояний входов системы имела размерность  $128 \times 128$ .

В рассмотренном примере справедливы следующие эвристические соображения относительно возможных переходов. Для состояний  $\eta = \|L_1, L_2, L_3, T_1, T_2, T_3, T_4\|$ , у которых  $\exists i: T_i = 1$ , возможны переходы либо в себя, либо в состояния, в которых  $\forall i T_i = 0$ . Также, для состояний  $\|1, 1, *, 1, 1, 0, 0\|$ ,  $\|*, 1, 1, 0, 1, 1, 0\|$ ,  $\|1, *, *, 1, 0, 0, 1\|$ ,  $\|*, *, 1, 0, 0, 1, 1\|$ ,  $\|1, *, 1, 1, 0, 1, 1\|$ ,  $\|*, 1, 1, 0, 1, 1, 1\|$ ,  $\|0, 0, *, 1, 1, 0, 1\|$ ,  $\|1, 1, 1, 1, 1, 1, 0\|$ ,  $\|1, *, *, 1, 0, 0, 0\|$ ,  $\|*, 1, *, 0, 1, 0, 0\|$ ,  $\|*, *, 1, 0, 0, 1, 0\|$  и обратно. При этом, запрещаются состояния, в которых для одного и того же  $i$  выполняются равенства  $T_i = 1$  и  $L_i = 0$ . Предложенные соображения позволили определить подмножество поиска  $Y_0^G$ . Матрица, имеющая значения элементов равное 1 на всех разрешенных местах будем называть матрицей-шаблоном  $G^*$ .

Для решения задачи оптимизации, для данного примера на основе предложенной схемы был разработан проблемно-ориентированный генетический алгоритм. Для измерения значения приспособленности, соответствующего индивиду с генотипом, соответствующим матрице  $G$  проводилась серия экспериментов, в которой использовался прямоугольный лабиринт, размером  $640 \times 505$  (единиц масштаба), содержащий 12 препятствий размером  $35 \times 35$  (единиц масштаба). Препятствия располагались регулярно на расстоянии не менее 100 (единиц масштаба) друг от друга, образуя ряды. Для снятия характеристики  $\xi_x(G)$  были установлены следующие параметры единичного эксперимента:  $T_{обуч}$  = 10000 тактов,  $T_{стат}$  = 2000 тактов, размер окна усреднения  $M_{средн}$  = 600 тактов. В системе ФРО разрешалось разместить 1000 нейроноподобных элементов, а для каждого элемента был установлен статистический порог  $M=10$ . Характеристика  $\tilde{\xi}(G)$  вычислялась по серии из  $N=10$  экспериментов. Перед началом каждого эксперимента все параметры системы ААУ робота приводятся в исходное состояние, а сам робот случайным образом размещается в лабиринте. Генотип представлял собой матрицу смежности  $G'$ , с модифицированным алфавитом символов. Для элемента (правила)  $g'_{ij}$  матрицы  $G'$ , символы алфавита имели следующий смысл. '0' – запрещает переход из состояния  $i$  в состояние  $j$ , '1' – разрешает переход из состояния  $i$  в состояние  $j$  и означает, что такой переход был разрешен, но не использован в сети, '2' – разрешает переход из состояния  $i$  в состояние  $j$  и обозначает, что такой переход был использован и нейроноподобный элемент типа  $j$  обучился, '3' – запрещает переход из состояния  $i$  в состояние  $j$  и означает, что такое правило было опробовано, однако нейроноподобный элемент типа  $j$  не обучился. Для функционирования робота использовалась матрица  $G$ , элементами которой являлись '0' и '1', получаемая из  $G'$ , заменой символов '2' и '3' на '1' и '0' соответственно. После каждого эксперимента сформированная сеть нейроноподобных элементов анализировалась на предмет использования правил перехода. Соответствующая информация вносилась в генотип, путем замены соответствующих символов в генотипе по правилам кодирования, представленным в таб. 4.

Состояние окончного элемента связи	Текущее значение $g'_{ij}$	Новое значение $g'_{ij}$
Обучен	1	2
	2	2
Не обучен	1	3
	2	1
	3	3

Таб.4. Схема кодирования генотипа индивида

Пример сформированных сетей нейроноподобных элементов для некоторого индивида представлен на Рис.3.

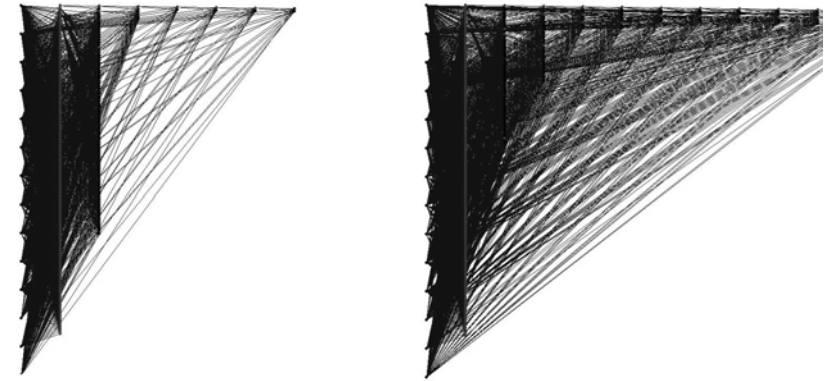


Рис.3. Пример полученных сетей нейроноподобных элементов подсистемы ФРО. Крайние левые элементы – выходные полюса датчиков.

Из представленных на рис. 3 примеров сформированных сетей нейроноподобных элементов видно, что во время своего функционирования система управления некоторого индивида оперировала образами, соответствующими последовательностям достаточно большой длины (в данном случае более 7-ми).

Генетические операторы были построены следующим способом. Для двух матриц  $G'_1$  и  $G'_2$  оператор кроссинговера  $C(G'_1, G'_2)$  определяется следующим способом. Примем за основу схему построения оператора кроссинговера (12), но с некоторыми изменениями. Зафиксируем значение  $c_0$  равномерно распределенной на отрезке  $[0,1]$  случайной переменной  $c$ . Тогда элементы  $\tilde{g}_{ij}$  матрицы потомка  $\tilde{G}$  будут определяться согласно следующим правилам. Если  $g_{ij}^1 = g_{ij}^2$ , то  $\tilde{g}_{ij} = g_{ij}^1$ . В противном случае рассматривается значение равномерно распределенной на отрезке  $[0,1]$  случайной переменной  $c$ , в соответствии с которым значения  $\tilde{g}_{ij}$  определяются по Таб. 5.

$\tilde{g}_{ij}$	$g_{ij}^1$	$g_{ij}^2$	$c$
0	0	1 или 2	$c < c_0$
	1 или 2	0	$c > c_0$
1	1	0	$c < c_0$

$\tilde{g}_{ij}$	$g'_{ij}{}^1$	$g'_{ij}{}^2$	$c$
	0	1	$c > c_0$
	2	3	любое
	3	2	
2	2	0	$c < c_0$
	0	2	$c > c_0$
	2	1	любое
	1	2	
3	0 или 1	3	любое
	3	0 или 1	

Таб.5. Правила преобразования генов при кроссинговере.

Оператор мутации для матрицы индивида с геномом  $G'$  построим следующим способом. Примем за основу определение оператора мутации выражаемое соотношениями (13) и (14). В схему замены элементов (14) внесем изменения, представленные в виде правил, указанных в Таб. 6.

$\tilde{g}_{ij}$	$g'_{ij}$	$c_2$
$g'_{ij}$	Любое	$c_2 > \tilde{P}_{мут}$
1	0 или 3	$c_2 < \tilde{P}_{мут}$
0	1 или 2	

Таб.6. Правила изменения генов в операторе мутации.

То обстоятельство, что время измерения приспособленности одного индивида занимает значительное время, накладывает ограничения на численность популяции. При производительности примерно 150 тактов работы системы ААУ в секунду (на РС Pentium IV 2800 MHz), измерение характеристики  $\xi_x(G)$ , при определенных выше параметрах эксперимента, занимает примерно 1 мин. 25 сек., а  $\tilde{\xi}(G)$  при  $N=10$  – примерно 14 мин. При таких временных параметрах процесса вычислений, приспособленность индивидов в популяции из 100 индивидов будет вычисляться в течение суток (примерно 23 часа 30 мин.), что неприемлемо. Для уменьшения времени вычислений использовались

популяции небольшого размера, а также стратегия генерации начальной популяции, способствующая быстрой сходимости алгоритма.

Для создания стартовой популяции была использована следующая стратегия. Пусть задано некоторое число  $0 < P_{ген} < 1$ . Тогда матрицу  $G'$  индивида будем создавать следующим способом. Для каждого элемента  $g'_{ij}$ , для которого значение соответствующего элемента  $g_{ij}^*$  матрицы-шаблона  $G^*$  равно 0, установим значение 0. В противном случае установим следующие правила присвоения значений. Если для значения равномерно распределенной на отрезке  $[0,1]$  случайной переменной  $c$  выполняется неравенство  $c < P_{ген}$ , то  $g'_{ij} = 1$ , и  $g'_{ij} = 0$  иначе. Если  $K$  есть количество единичных элементов в матрице-шаблоне  $G^*$ , то для предложенного способа формирования матриц  $G'$ ,  $K \cdot P_{ген}$  есть математическое ожидание количества единичных элементов в такой матрице. В самом деле, количество бинарных матриц, состоящих из  $K$  элементов,  $N$  из которых единицы, есть  $C_K^N$ . Из таких матриц, количество тех, у которых значение элемента в определенной позиции равно единице, есть  $C_{K-1}^{N-1}$ . Тогда, вероятность того, что для выбранной случайным образом из всего множества матриц с  $N$  единицами и  $K$  элементами, значение элемента, расположенного в определенной позиции, равно 1, есть  $P_1 = \frac{C_{K-1}^{N-1}}{C_K^N} = \frac{N}{K}$ .

Нетрудно видеть, что если для элемента матрицы вероятность равенства единице есть  $P_{ген} = P_1$ , то математическое ожидание количества единичных элементов матрицы равно  $N$ .

Для рассматриваемого примера максимально возможное количество элементов равных 1 есть  $K = 1357$  (разрешены не все переходы между допустимыми состояниями). Стартовую популяцию индивидов создавалась со значением вероятности  $P_{ген} = \frac{650}{1357}$ . Предложенная стратегия позволила получить начальную популяцию индивидов, для которых математическое ожидание количества единичных элементов в матрице  $G'$  равно 650.

Для вычислений на персональном компьютере была использована популяция, состоящая из  $N_{инд} = 20$  индивидов, приспособленность которых вычисляется по серии из  $N = 10$  экспериментов. Критерием сходимости алгоритма была принята локализация множества матриц  $G$  индивидов в шаре радиуса  $A=5$  в смысле расстояния по Хеммингу (Определения 22 и 23).

Работа генетического алгоритма представлена на рис. 4 и 5. По результатам применения алгоритма сделан вывод о том, что предложенная методика оптимизации и проектирования генетических алгоритмов работоспособна и приводит к созданию более удачных (согласно введенным критериям) систем.

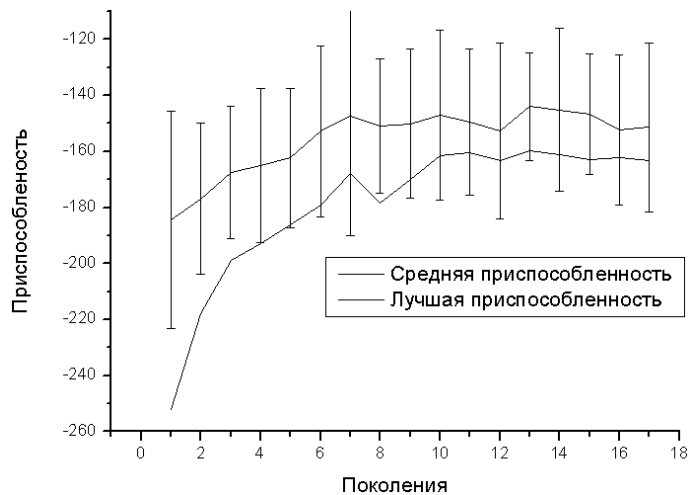


Рис.4. График изменения лучшей и средней приспособленности в популяции в зависимости от номера поколения

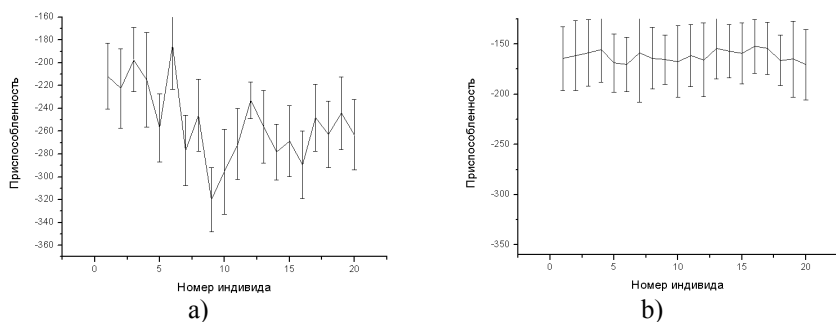


Рис.5. Распределение значений приспособленности по популяции: а) первое поколение, б) 17-е поколение.

Из графика видно, что предложенный генетический алгоритм обладает сходимостью и способностью локализовывать популяцию в районе экстремума. Графики распределения значения приспособленности по популяции представлены на рис. 5. На рис. 5 (а) и (б) представлены распределения значения приспособленности в 1-й и 17-й популяция соответственно. Из этих рисунков видно, что в 1-м поколении распределение значения приспособленности имеет значительный разброс, в то время как для 17-го поколения такой разброс незначителен. Это обстоятельство также

подчеркивает способность предложенного генетического алгоритма локализовывать индивидов популяции в некоторой окрестности экстремума.

## 5. Альтернативные и дополнительные возможности применения генетических алгоритмов в системах автономного адаптивного управления

В настоящей части представлены результаты исследований возможности применения генетических алгоритмов для оптимизации подсистемы «База Знаний» (БЗ) систем ААУ. Рассмотрены как методы оптимизации структуры базы знаний, так и возможности по применению генетических алгоритмов в процессе управления. Предложенные методы были опробованы на примерах системы ААУ «Пилот» предназначенной для адаптивного управления угловым движением комического аппарата [13], и системы ААУ автономного мобильного робота.

В системе ААУ «Пилот» блок базы знаний представляет собой двумерную таблицу, столбцы которой соответствуют заранее зафиксированным образам, распознаваемым системой формирования и распознавания образов (ФРО), а строки соответствуют действиям, которые могут выполнить актуаторы. В клетках таблицы фиксируются статистически достоверные сведения о том, какую оценку состояния получает система при условии распознавания конкретного образа и совершения конкретного действия. В процессе управления используются только те команды, которые приводят к лучшей из возможных оценок. Большая часть эмпирически найденной таблицы оказывается неиспользуемой. Требовалось построить процедуру распределения ресурсов базы знаний таким способом, чтобы база знаний реализовывалась как можно меньшим числом элементов. Для решения этой проблемы были предложены два подхода, основанные на применении генетических алгоритмов.

Система ААУ «Пилот» распознает  $N=121$  образ и может совершать 41 действие. Первый подход ставит в соответствие каждому столбцу базы знаний один генетический алгоритм, каждая операция которого совершается при распознавании системой ФРО соответствующего образа. Каждый генетический алгоритм реализован следующим способом. Рассматривается отрезок  $[0,41]$  числовой прямой и  $n=5$  размещенных на нем точек, каждая из которых считается индивидом. При попадании точки в интервал  $(i-1, i]$ ,  $i=1,41$ , считается, что она соответствует команде номер  $i$ . Функцией приспособленности индивида считается значение оценки состояния при совершении системой действия, соответствующего оцениваемой точке. Операторы кроссинговера и мутации определены аналогично способу, описанному в главе 1, но применительно к одномерному случаю. Все 121 генетический алгоритм выполняются условно параллельно. Каждый такт работы системы один шаг выполнения совершает только тот алгоритм,

который соответствует распознанному на данном такте образу. Так как генетические алгоритмы эволюционируют в сторону наилучших оценок состояния системы, а таким оценкам соответствует определенная группа образов, то система большую часть времени будет проводить в состояниях с высокой оценкой. Это приводит к тому, что алгоритмы, которые таким образом соответствуют, эволюционируют гораздо быстрее, чем те, которые соответствуют состояниям с низкой оценкой. При достаточно долгом периоде обучения ( $10^6$  тактов) популяции каждого генетического алгоритма сходятся к тем командам, которые вызывают высокие оценки состояния. Получившаяся структура базы знаний затем фиксируется и используется в типичном для ААУ процессе управления. Оптимизированная структура базы знаний позволяет сразу использовать только те команды, которые приводят к достаточно высокому качеству управления, т.е. позволяет получить такую базу знаний, которая содержит только такие нейроноподобные элементы, которые с большой вероятностью будут использоваться управляющей системой для управления, в том числе, и при изменении свойств объекта управления. Другими словами, из базы знаний удаляются элементы, которые заведомо не потребуются управляющей системе (рис. 6).

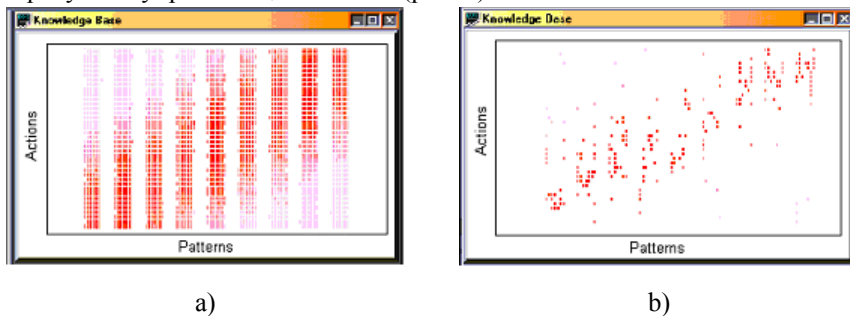


Рис. 6. Выделение закона управления из заполненной БЗ: а) заполненная БЗ, б) выделенный закон управления.

Предложенный подход может использоваться и как метод, реализующий работу базы знаний. Отсутствие критерия останова и свойство генетического алгоритма локализовать решения в некоторой небольшой окрестности оптимума делают базу знаний, построенную на основе рассмотренного подхода, способной к обучению и адаптации.

Рассмотренная задача была также решена на основе другого метода применения генетических алгоритмов. В качестве индивида рассматривалась система ААУ «Пилот», имеющая в базе знаний лишь один закон управления – по одному действию на один образ. Такой закон управления являлся генотипом индивида. Каждая система управления получала оценку приспособленности в результате наблюдения ее работы на основе единственного закона управления, которая

равнялась средней оценке состояния за фиксированный промежуток времени. Были определены следующие генетические операторы. Кроссинговер – классический оператор одноточечного кроссинговера, определенный на хромосомах длины 121 и имеющих алфавит аллелей из 41 элемента. Оператор мутации в мутирующем индивиде с низкой вероятностью изменяет значение каждого гена на одно из алфавита аллелей. Критерием останова является количество итераций алгоритма. Решением задачи является объединение всех законов управления, локализованных в шаре заданного радиуса в смысле определенной для законов управления метрики.

Рассмотренный метод был также опробован на примере задачи оптимизации базы знаний системы ААУ автономного мобильного робота. От приведенного выше описания модели автономного мобильного робота рассматриваемая отличалась тем, что в ней были зафиксированы образы, распознаваемые системой ФРО, а база знаний была организована по принципу, использованному в системе ААУ «Пилот». В обоих рассмотренных примерах генетический алгоритм демонстрирует сходимость. Особенно интересным результатом является то, что итоговые конфигурации базы знаний, полученные разными методами, основанными на генетических алгоритмах, близки.

На примере системы ААУ автономного мобильного робота с фиксированной топологией сети нейроноподобных элементов был рассмотрен еще один алгоритм оптимизации, построенный на основе рассмотренного выше. Основным отличием является использование недетерминированного закона управления, состоящего в том, что при распознавании образа вводятся вероятности совершения того или иного действия. Для одного образа все такие вероятности нормированы на 1. Индивидом в таком генетическом алгоритме является система ААУ, имеющая вероятностный принцип выбора действия при распознавании образов. Генотипом является собственно конфигурация базы знаний. Операторы кроссинговера и мутации определены следующим образом. Кроссинговер для каждого образа в БЗ реализует следующую процедуру. Значения вероятностей для каждого действия рассматриваются как компоненты вектора, размерность которого совпадает с количеством действий (для робота 6). Распределение вероятностей для соответствующего образа в

базе знаний потомка вычисляется по формуле 
$$\vec{P} = \frac{(\vec{P}_1 + \vec{P}_2)}{|\vec{P}_1 + \vec{P}_2|}$$
. Оператор мутации

реализован следующим способом. В базе знаний мутирующего индивида, для каждого образа с очень низкой вероятностью производится процедура случайного изменения вероятностей выбора действий с последующим их нормированием на 1. Критерием останова является фиксированное количество итераций алгоритма.

Во всех рассмотренных примерах, в генетических алгоритмах процедура отбора в пары для воспроизводства была реализована по 2-х туровому алгоритму отбора с запрещением саморепродукции. Результаты приведенных в



4-й главе исследований были использованы в диссертационной работе. Результаты, касающиеся оптимизации структуры базы знаний могут использоваться как дополнение к результатам оптимизации, изложенным в главах 2 и 3.

## 6. Основные результаты

В целом в работе представлен найденный, обоснованный и разработанный подход к построению и оптимизации прикладных систем автономного адаптивного управления с помощью генетических алгоритмов, что решает проблему синтеза таких систем в условиях недостаточной априорной информации. Основные результаты:

1. введен формализм описания блока датчиков и актуаторов систем ААУ, предложена процедура количественного оценивания согласованности их параметров; сформулирована в общем виде задача оптимизации таких параметров, предложены рекомендации по построению генетических алгоритмов для ее решения;
2. на основании введенного формализма и постановки задачи оптимизации был разработан генетический алгоритм подбора параметров блока датчиков и актуаторов на примере практического приложения – системы автономного адаптивного управления мобильного робота;
3. введен формализм описания строения сети нейроноподобных элементов подсистемы формирования и распознавания образов;
4. на основе предложенного формализма разработана процедура динамического формирования сети нейроноподобных элементов подсистемы формирования и распознавания образов, а также сформулирована в общем виде задача оптимизации ее параметров;
5. был разработан генетический алгоритм решения задачи оптимизации параметров процедуры динамического формирования сети нейроноподобных элементов подсистемы формирования и распознавания образов, общая схема которого не зависит от специфики прикладной задачи. Практическое решение такой задачи было рассмотрено на практическом примере.

## Литература

- [1] Чечкин А.В. Математическая информатика. М.: «Наука» Главная редакция физико-математической литературы, 1991 – 416с.
- [2] Bale T. A., Modular Connectionist Architectures and the Learning of Quantification Skill, 1998

- [3] Головкин В.А. Нейронные сети: обучение, организация и применение. Кн.4: Учеб. Пособие для вузов / Общая ред. А.И. Галушкина. – М.: ИПРЖР, 2001. – 256 с.: ил. (Нейрокомпьютеры и их применение)
- [4] Reinforcement Learning: An Introduction Richard S. Sutton and Andrew G. Barto MIT Press, Cambridge, MA, 1998
- [5] Artificial Life An Overview Christopher G. Langton (Ed.) MIT Press, 1995
- [6] From Animal to Animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior / Eds J.-A. Meyer, S.W. Wilson. Cambridge at al: MIT Press, 1990
- [7] Albus, J.S., Meystel, A.M. "A Reference Model Architecture for Design and Implementation of Intelligent Control in Large and Complex Systems," International Journal of Intelligent Control and Systems, Vol. 1, No. 1, pgs. 15-30, 1996.
- [8] Жданов А.А. Об одном имитационном подходе к адаптивному управлению. // сб. Вопросы Кибернетики. Выпуск 2 М., 1996 с.171-206.
- [9] Жданов А.А. Формальная модель нейрона и нейросети в методологии автономного адаптивного управления. // сб. Вопросы Кибернетики. Выпуск 3 М., 1997. С.258-274
- [10] 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 (англ. Том. Zhdanov A.A., Arsenjev S.V., Polovnikov V.A., On autonomous adaptive control methodology. // Proceedings of the Russian Academy of Sciences Institute for System Programming. N 1, 1999, pp.55-70)
- [11] А.А. Жданов, М.В. Крыжановский, Н.Б. Преображенский. Бионическая интеллектуальная автономная адаптивная система управления мобильным роботом // Мехатроника, 2004. №1, с. 21-30 (часть 1) и №2 (часть 2), с. 17-22.
- [12] Жданов А.А. Земских Л.В. Беляев Б.Б. Система стабилизации углового движения космического аппарата на основе нейроноподобной системы автономного адаптивного управления. Космические Исследования, М. 2004, т. 42, №3, с.1-15. (принята редакцией в 2002).
- [13] Жданов А.А. Земских Л.В., Беляев Б.Б. Применение генетических алгоритмов для оптимизации нейросетевой базы знаний адаптивной системы стабилизации углового движения космического аппарата. Сб. тезисов докладов XXV академических чтений по космонавтике, Москва, 24-26 января 2001 г. Сс. 128-129, «Война и мир», Москва.
- [14] Zhdanov A.A., L.V. Zemskikh, The Evolutionary Growth of Neural Networks for the Autonomous Adaptive Control System. // The 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and the 7th International Conference on Information Systems Analysis and Synthesis (ISAS 2001), Orlando, USA, July 22-25, 2001. Pp. 404-409, 2001.