

**МОДИФИКАЦИЯ МЕТОДА ГЕНЕРАЦИИ УПРАВЛЯЮЩИХ
КОНЕЧНЫХ АВТОМАТОВ С НЕПРЕРЫВНЫМИ
ВОЗДЕЙСТВИЯМИ ПО ОБУЧАЮЩИМ ПРИМЕРАМ***

© 2015 г. И. П. Бужинский, С. В. Казаков, В. И. Ульянов, Ф. Н. Царев, А. А. Шальто

Санкт-Петербург, Университет ИТМО

e-mail: buzhinsky@rain.ifmo.ru

Поступила в редакцию 04.02.14 г., после доработки 26.01.15 г.

В связи с возможностью формальной верификации и наглядностью конечных управляющих автоматов их можно использовать при разработке надежных систем управления. В статье рассматривается решение задачи генерации автоматов управления объектами со сложным поведением на основе обучающих примеров. Входные и выходные воздействия автоматов задаются вещественными числами. Предлагается метод генерации автоматов, являющийся модификацией ранее предложенных подходов, основанных на генетическом и муравьином алгоритмах. Изменения включают в себя новый способ представления автоматов и улучшенные функции приспособленности. Метод позволяет генерировать автоматы, поведение которых больше соответствует обучающим примерам, чем поведение автоматов, сгенерированных известными подходами.

DOI: 10.7868/S0002338815050042

Введение. В настоящей работе рассматривается задача генерации конечных управляющих автоматов (далее — автоматов) по набору обучающих примеров, описывающих поведение объекта со сложным поведением — объекта, способного вырабатывать выходные воздействия, основываясь не только на текущих входных воздействиях, но и на предыстории последних. В качестве примера такого объекта выбрана модель беспилотного самолета.

Автоматы широко используются при разработке реактивных систем [1–3] и могут быть формально верифицированы методом *Model Checking*, что открывает возможность их применения как компонентов надежного программного обеспечения. Кроме того, автоматы могут служить моделями уже существующих программных систем. Например, в [4, 5] описаны методы получения таких автоматов по данным трассировок программы.

Построение автоматов вручную часто бывает весьма затруднительным или невозможным. Так, например, для задачи “Умный муравей” оптимальный автомат был найден только с помощью автоматизированного построения автоматов [6] с использованием генетических алгоритмов [7–10]. При применении таких алгоритмов, а также других методов поисковой оптимизации (например, эволюционных стратегий) необходимо определить критерий качества решения оптимизационной задачи — *функцию приспособленности* (ФП). Известны два подхода к заданию ФП автоматов. Первый основан на сравнении идеальной модели поведения объекта управления (например, заданного маршрута для модели самолета) с поведением, совершаемым под управлением автомата [11]. Второй подход использует заранее записанные примеры желаемого поведения автомата [12], такие, что проверка соответствия автомата этим примерам не требует симуляции объекта управления.

В [11] для решения задачи генерации автомата, управляющего моделью самолета, был применен генетический алгоритм. При этом ФП вычислялась на основе поведения модели в авиасимуляторе, что приводило к большим вычислительным затратам. В [13–15] для повышения производительности использовалась принципиально другая ФП, основанная на обучающих примерах. Существенным элементом предложенного в [13] подхода являлась автоматическая

* Результаты были получены в рамках выполнения государственного задания Минобрнауки России (задание № 2.1239.2014/К). Кроме того, работа выполнена при государственной финансовой поддержке ведущих университетов Российской Федерации (субсидия 074-U01), а также при финансовой поддержке РФФИ (проект № 14-07-31244 мол_а).

расстановка выходных воздействий на переходах автоматов. Этот подход применим не только при управлении моделью самолета, но и другими объектами — в том числе теми, модель поведения которых сложна или неизвестна. Именно такой подход и развивается в настоящей работе. В ней предлагается сохранить схему подхода, но изменить способ представления автоматов. Это должно привести к большему соответствию поведения генерируемых автоматов обучающим примерам при отсутствии проблемы “переобучения” (ситуации, в которой хорошее поведение автомата на обучающих примерах не обобщается на неизвестные во время оптимизации случаи), а также сделать автоматы более наглядными. Для генерации автоматов применяется муравьиный алгоритм, ранее использованный в [14, 15].

В разд. 1 настоящей работы формально описана решаемая задача. В разд. 2 детально представлены предлагаемые улучшения метода генерации автоматов. Их целесообразность продемонстрирована результатами экспериментов, приведенных в разд. 3.

1. Постановка задачи. Для начала приведем неформальное определение конечного управляющего автомата, используемое в настоящей работе, которое далее будет уточнено. Автомат представляет собой управляющую систему, работающую по тактам. В настоящей работе рассматриваются синхронные автоматы — автоматы, такты которых происходят через равные промежутки времени (0.1 с). У автомата есть конечное множество состояний S и начальное состояние $s_0 \in S$. В каждый момент времени автомат находится в одном из состояний. На каждом такте на основе входных воздействий автомат совершает переход (изменяет свое состояние) и вырабатывает выходное воздействие. Переходы задаются функцией переходов δ , выходные воздействия — функцией выходов λ .

Входные воздействия автомата в данной работе определяются состоянием объекта управления и являются наборами вещественных чисел, описывающих показания датчиков объекта. Выходные воздействия также представляются наборами вещественных чисел и являются изменениями управляющих параметров объекта. В следующем разделе механизм получения входных и выработки выходных воздействий будет описан подробно. Непрерывность входных и выходных воздействий (задание их вещественными числами) приводит к нетривиальным реализациям функций переходов и выходов по сравнению с классическими автоматами, оперирующими только дискретными воздействиями. Предлагаемая реализация этих функций будет изложена в разд. 2.1.

Пусть задан конечный набор обучающих примеров. Они описывают некоторые задания, которые должен выполнить объект управления. Для модели самолета в качестве таких заданий будем использовать определенные фигуры пилотажа. Каждый обучающий пример описывает значения входных параметров объекта управления (для самолета это параметры полета: скорость, высота, угол крена и т.д.) и его управляющих параметров в различные моменты времени. Управляющие параметры задают положения органов управления (для самолета это руль высоты, элероны и т.д.). Для записи обучающих примеров и запуска самолета, управляемого автоматом, применяется авиасимулятор *FlightGear* [1]. Рассматриваемая задача состоит в генерации автомата, поведение которого близко к описанному в обучающих примерах.

1.1. Управление объектом. *Кортежем входных воздействий* будем называть упорядоченный набор из $p \in \mathbb{N}$ вещественных чисел, соответствующих входным параметрам объекта управления. Будем полагать, что у объекта управления $c \in \mathbb{N}$ управляющих параметров, соответствующих его *органам управления*. Эти органы бывают двух типов: дискретные, имеющие конечное множество положений, и непрерывные, положение которых можно задать числом из некоторого отрезка вещественной оси. Примером дискретного органа управления является стартер (у него два положения: включенное и выключенное), а примером непрерывного — руль направления (числа от -1 до 1 соответствуют его положениям от крайнего левого до крайнего правого). Далее для простоты будем считать, что все органы управления являются непрерывными, однако описываемый метод допускает обобщение и на случай наличия органов управления обоих типов. *Кортеж управляющих параметров* — набор из c вещественных чисел — является “снимком” значений всех управляющих параметров в некоторый момент времени. Будем считать, что j -й ($j = \overline{1, c}$) элемент кортежа управляющих параметров всегда ограничен отрезком $[m_j, M_j]$, соответствующим возможным значениям j -го управляющего параметра.

Как упоминалось выше, автомат работает по тактам. В начале каждого такта он принимает кортеж входных воздействий от объекта управления. Каждый переход автомата помечен охраняемым условием — булевой формулой от *предикатов*, которые также являются булевыми величинами. Выполняется тот переход, для которого вычисление соответствующей формулы на текущих

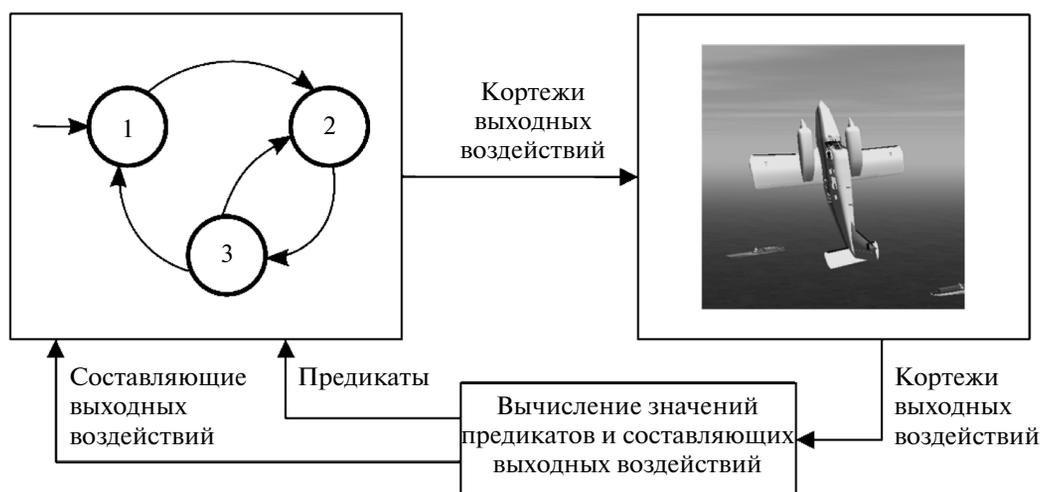


Рис. 1. Взаимодействие автомата и модели самолета

значениях предикатов дает истину (такой переход должен быть единственным). В общем случае значения предикатов зависят от всех кортежей входных воздействий, которые были переданы автомату к настоящему такту. Примеры предикатов – утверждения “высота уменьшается”, “угол тангажа положительный”.

Далее введем понятие *составляющих выходных воздействий*. Это вещественные величины, вычисляемые аналогично предикатам, т.е. на основе всех переданных автомату входных воздействий. Составляющими выходных воздействий могут быть входные воздействия, их степени и скорости их изменения, вычисляемые как разности двух последних значений соответствующего входного воздействия. Примеры составляющих – угол крена, отклонение угла тангажа от 5° , квадрат вертикальной скорости. Название этих величин связано с тем, что в настоящей работе они непосредственно определяют выходные воздействия, в то время как в [13–15] последние зависели от значений предикатов.

Схема процесса управления приведена на рис. 1. В левой части рисунка схематично (в виде графа переходов) представлен автомат. Вершины графа переходов соответствуют состояниям автомата, а цифры внутри вершин обозначают их номера. Дуги графа задают переходы автомата. Начальное состояние $s_0 = 1$ обозначено входящей в него дугой, не имеющей начала ни в одном состоянии.

1.2. **Обучающие примеры.** Формализуем понятие обучающего примера. Он образован двумя последовательностями кортежей входных воздействий и управляющих параметров, каждая из которых проиндексирована моментами времени. Так как автомат синхронный, то разность между соседними моментами времени равна промежутку между тактами. Длина этих последовательностей для i -го обучающего примера равна L_i ($i = \overline{1, N}$, где N – число обучающих примеров). Будем называть эти числа длинами обучающих примеров. Последовательность кортежей входных воздействий I_i состоит из чисел $I_{i,t,j}$, где $t = \overline{1, L_i}$ – номер кортежа в последовательности, а $j = \overline{1, p}$ – номер входного воздействия в кортеже. Ей соответствует последовательность кортежей управляющих параметров O_i , состоящая из чисел $O_{i,t,j}$, где $j = \overline{1, c}$ – номер управляющего параметра. Фрагмент обучающего примера для случая $p = 4, c = 3, L_i = 235$ приведен в табл. 1.

2. Предлагаемый метод. Настоящий раздел описывает предлагаемый подход как модификацию метода, предложенного в [13] и ускоренного в [14, 15]. Общая схема метода из [13] сохраняется: поиск автоматов производится с помощью алгоритма поисковой оптимизации, особями которого являются *каркасы* автоматов – автоматы, функции выходов которых не определены. Новыми являются способ представления автоматов, а также две модификации ФП, одна из которых предназначена для увеличения “наглядности” автоматов, а вторая позволяет улучшить качество выполнения фигур пилотажа, требующих от самолета поддержания определенной величины (например, угла крена) постоянной. В качестве алгоритма поисковой оптимизации используется алгоритм из семейства муравьиных [17], предложенный в [18], который показал

Таблица 1. Фрагмент обучающего примера

Значения	Описание	$t = 1$...	$t = 10$...	$t = 20$...	$t = 235$
$I_{i,t,1}$	Угол тангажа, град	3.078	...	3.544	...	4.112	...	2.412
$I_{i,t,2}$	Угол крена, град	-0.076	...	0.351	...	3.413	...	1.759
$I_{i,t,3}$	Угол курса, град	198.03	...	198.11	...	198.41	...	205.64
$I_{i,t,4}$	Скорость, узлы	251.42	...	252.29	...	253.20	...	289.40
$O_{i,t,1}$	Положение элеронов (число от -1 до 1)	0.000	...	0.032	...	0.073	...	-0.003
$O_{i,t,2}$	Положение руля направления (число от -1 до 1)	0.000	...	0.016	...	0.037	...	-0.001
$O_{i,t,3}$	Положение руля высоты (число от -1 до 1)	-0.035	...	-0.039	...	-0.037	...	-0.011

несколько лучшую, чем в [13], производительность [14, 15] на рассматриваемой задаче и не требует задания оператора кроссовера. В этом алгоритме пространство поиска представлено в виде ориентированного графа, вершинами которого являются автоматы (в настоящей работе – их каркасы), а дугами – их мутации (небольшие изменения). Далее рассмотрим детали реализации предлагаемого метода генерации автоматов.

2.1. Способ представления автоматов. Базовый подход [13] к решению рассматриваемой задачи позволяет генерировать автоматы, выходные воздействия которых зависят только от значений предикатов. Это накладывает некоторые нежелательные ограничения на выходные воздействия. Так, при использовании только предикатов автомат не может вырабатывать произвольно малые выходные воздействия (изменения значений управляющих параметров), что могло бы быть полезно, если требуется стабилизировать некоторый параметр объекта управления, например угол тангажа самолета. В одной из фигур пилотажа, рассматриваемых в настоящей работе (при развороте самолета на 180° в горизонтальной плоскости), как раз это и требуется. Кроме того, у автоматов из [13] за такт происходит несколько переходов, что делает функцию переходов сложной для анализа.

Для устранения указанных недостатков предлагается другой способ представления автоматов, позволяющий задействовать как предикаты, так и составляющие выходных воздействий. Рассмотрим набор предикатов x_1, \dots, x_m . В каждом состоянии автомата используется только их подмножество – так называемые *значимые* в этом состоянии предикаты. Для каждого состояния $s \in S$ и для каждого набора значений значимых в s предикатов определен переход из этого состояния s и при этих значениях предикатов. В программном представлении подобного автомата в каждом его состоянии s хранятся маска значимости предикатов и таблица переходов, элементы которой являются конечными состояниями исходящих из s переходов. Этот подход был назван в [11] методом сокращенных таблиц. Он позволяет сократить размер описания автомата при больших n по сравнению с более простым методом полных таблиц [3] (применение последнего приводит к тому, что в каждом состоянии автомата хранится информация обо всех 2^m переходах). Отметим, что число состояний автомата $|S|$ и число значимых предикатов считаются фиксированными во время оптимизации. Число значимых предикатов одинаково для всех состояний.

Составляющие выходных воздействий используются для выработки выходных воздействий автомата. Для каждого управляющего параметра j существует набор составляющих выходных воздействий $v_{j,1}, \dots, v_{j,n_j}$. Причина использования нескольких (возможно, пересекающихся) наборов составляющих состоит в том, что для каждого управляющего параметра важны свои собственные составляющие. Например, для управления элеронами важны угол крена и скорость его изменения, а для руля высоты – составляющие, связанные с углом тангажа.

На каждом такте выходное воздействие автомата формируется как линейная комбинация значений составляющих выходных воздействий с коэффициентами, определяемыми только состоянием автомата – в этом смысле автомат является автоматом Мура [3]. Более формально значение j -го управляющего параметра u_j на каждом такте изменяется на величину

$$\Delta u_j = \sum_{i=1}^{n_j} r_{s,i,j} v_{j,i}, \quad (2.1)$$



Рис. 2. Пример программного представления состояния автомата

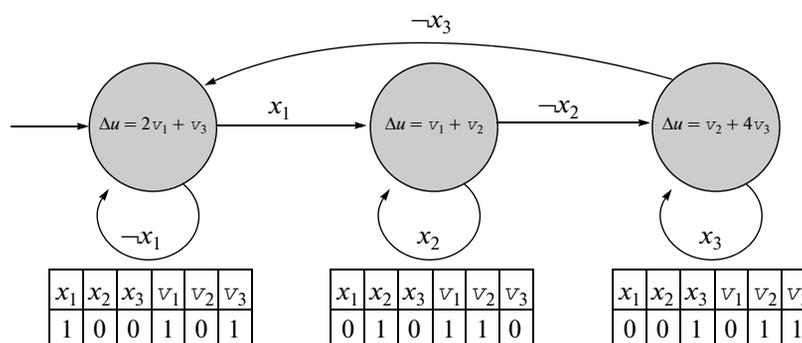


Рис. 3. Пример автомата с тремя предикатами и тремя составляющими выходных воздействий

где $r_{s,i,j}$ – коэффициент, соответствующий состоянию s автомата до совершения перехода и составляющей $v_{j,i}$. Линейность (2.1) далее позволит автоматически определять коэффициенты так, чтобы ФП автомата достигала максимума на заданном каркасе.

Как и предикаты, не все составляющие выходных воздействий являются значимыми, и для каждого управляющего параметра в каждом состоянии автомата хранится дополнительная маска значимости составляющих выходных воздействий. Будем считать, что числа $r_{s,i,j}$, соответствующие незначимым составляющим, заведомо равны нулю. Остальные числа будут определяться автоматически. Использование масок значимости составляющих выходных воздействий, во-первых, упрощает генерацию автоматов при больших n_1, \dots, n_c , так как время работы алгоритма расстановки выходных воздействий, который будет описан в разд. 2.3, пропорционально кубу от числа определяемых им чисел. Появление заведомо нулевых чисел $r_{s,i,j}$ сокращает это время. Во-вторых, маски значимости составляющих выходных воздействий можно использовать для избавления от “переобучения”, которое может проявиться, если число оптимизируемых параметров велико.

Пример предлагаемого представления состояния автомата приведен на рис. 2. В левом верхнем углу рисунка находится маска значимости предикатов (их четыре). Значимые предикаты помечены единицей, незначимые – нулем. Правее показана таблица переходов из текущего состояния. Ниже для двух органов управления приведены маски значимости составляющих выходных воздействий, числа $r_{s,i,j}$ и выходные воздействия вида (2.1), задаваемые этими числами. Несмотря

на достаточно сложное представление состояний автоматов, графы переходов последних выглядят сравнительно просто. Пример такого графа приведен на рис. 3. На нем для простоты рассмотрен случай одного органа управления, при этом маски значимости и предикатов, и составляющих выходных воздействий помещены под состояния, а внутри состояний находятся задаваемые ими выходные воздействия вида (2.1).

При использовании муравьиного алгоритма [18] для скелета автомата необходимо задать оператор мутации. В настоящей работе рассматриваются следующие мутации: изменение начального состояния автомата на случайное, изменение конечного состояния некоторого перехода на случайное, обмен двух неравных элементов некоторой маски значимости предикатов или составляющих выходных воздействий.

2.2. Функции приспособленности. Пусть \tilde{O}_i – последовательность кортежей управляющих параметров, выработанная фиксированным автоматом в ответ на I_i . Будем считать, что $\tilde{O}_{i,1,j} = O_{i,1,j}$ (начальные выходные воздействия автомата равны начальным выходным воздействиям, записанным в обучающем примере), а результатом такта автомата с номером t является выработка воздействий $\tilde{O}_{i,t+1,j}$. Для измерения близости поведения автомата на конкретном обучающем примере к записанному в этом же обучающем примере поведению используется следующее расстояние-штраф между выходными последовательностями:

$$\rho(\tilde{O}_i, O_i) = \sqrt{\frac{1}{L_i c} \sum_{t=2}^{L_i} \sum_{j=1}^c \left(\frac{\tilde{O}_{i,t,j} - O_{i,t,j}}{M_j - m_j} \right)^2}. \quad (2.2)$$

На основе этого штрафа в [14, 15] применялась следующая ФП:

$$\tilde{f} = 1 - \sqrt{\frac{1}{N} \sum_{i=1}^N \rho^2(\tilde{O}_i, O_i)}.$$

Введем первую модификацию ФП. Заметим, что автоматы, которые часто изменяют свое состояние, менее наглядны и более сложны для анализа. Поэтому имеет смысл минимизировать величины $\max(0, \tau_i - |S| + 1)$, где τ_i – число изменений состояния автомата при прохождении автоматом i -го обучающего примера. Будем учитывать эти величины как штрафы при вычислении ФП:

$$P_\tau = K \sqrt{\frac{1}{N} \sum_{i=1}^N (\max(0, \tau_i - |S| + 1))^2};$$

$$f = 1 - \sqrt{\frac{1}{N} \sum_{i=1}^N \rho^2(\tilde{O}_i, O_i) - P_\tau}, \quad (2.3)$$

где K – небольшое число (в настоящей работе $K = 0.00015$). В предыдущих работах [13–15] слагаемое P_τ не учитывалось. Поскольку в предыдущем подходе автомат совершал несколько переходов за такт, минимизации P_τ не было бы достаточно для установления назначения отдельных состояний автомата.

Рассмотрим вторую модификацию ФП. Аналогично функции f определим функцию f_Δ :

$$f_\Delta = 1 - \sqrt{\frac{1}{N} \sum_{i=1}^N \rho^2(\Delta\tilde{O}_i, \Delta O_i) - P_\tau},$$

где $\Delta\tilde{O}_i$ и ΔO_i являются последовательностями разностей соседних элементов соответствующих им последовательностей \tilde{O}_i и O_i . Более формально, $\Delta\tilde{O}_{i,1,j} = 0$, $\Delta\tilde{O}_{i,t,j} = \tilde{O}_{i,t,j} - \tilde{O}_{i,t-1,j}$ для $t > 1$. Аналогичное утверждение верно для ΔO_i .

Потребность в использовании f_Δ возникает тогда, когда обучающие примеры описывают “пилообразные” изменения управляющих параметров, цель которых – поддержание некоторого параметра объекта управления. Авторами экспериментально установлено, что в этом случае оптимизация f приводит к чрезмерно “гладким” выходным воздействиям. Такие выходные воздействия реагируют на изменения поддерживаемого параметра слишком медленно, что ухудшает

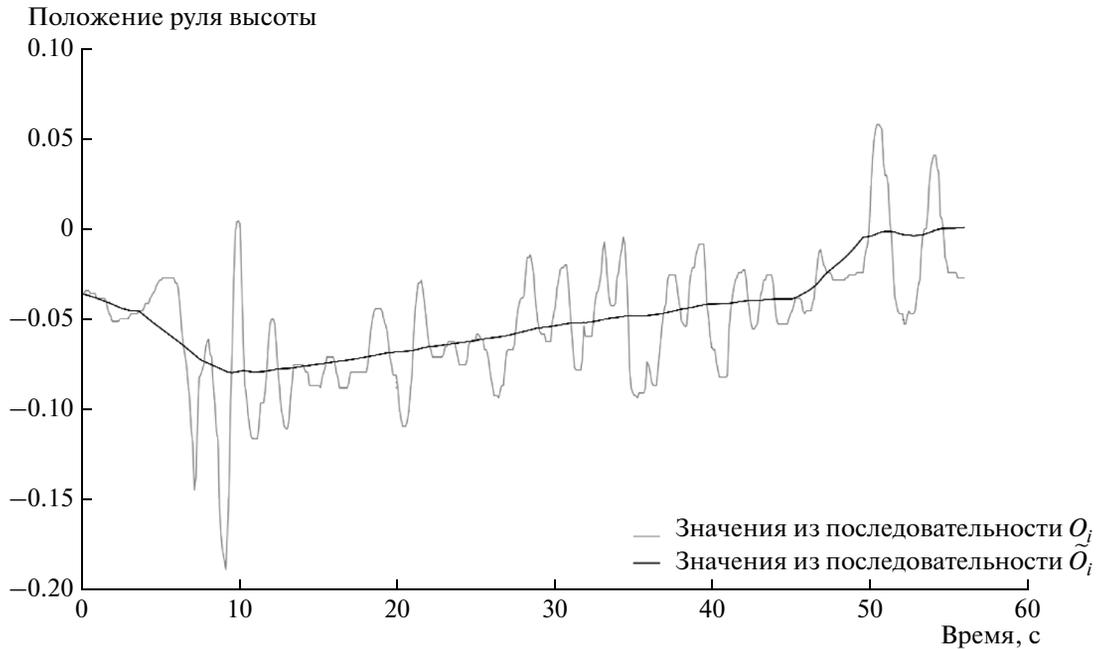


Рис. 4. Пример “пилообразных” выходных воздействий в обучающем примере и выходных воздействий автомата на этом же обучающем примере

качество управления этим параметром. Пример описываемой ситуации приведен на рис. 4, где показаны кривая, задающая положение руля высоты для одного из обучающих примеров, и кривая, показывающая поведение одного из сгенерированных автоматов на этом обучающем примере. Первая кривая описывает значения из последовательности O_i в различные моменты времени, вторая — значения из последовательности \tilde{O}_i . Данных проблем не наблюдается при оптимизации f_{Δ} .

2.3. Определение выходных воздействий. В [13–15] выходные воздействия представлялись на переходах автоматов автоматически. В настоящей работе также применяется подобный подход, однако процедура их расстановки имеет специфику в силу используемого способа представления автоматов. Для определения выходных воздействий, согласно (2.1), достаточно задать числа $r_{s,i,j}$. Напомним, что числа $r_{s,i,j}$, соответствующие незначимым составляющим выходных воздействий, заведомо равны нулю. Остальные числа можно определить так, чтобы на заданном каркасе автомата ФП (f или f_{Δ}) достигла максимума. Для этого для каждого управляющего параметра необходимо решить систему линейных уравнений. Построение этой системы описано в Приложении. Также в конце Приложения приведена оптимизация, позволяющая совмещать вычисления систем для различных управляющих параметров. Определение выходных воздействий производится для каждого сгенерированного в процессе поисковой оптимизации каркаса автомата и является наиболее вычислительно трудоемким этапом метода.

3. Экспериментальное исследование. Эксперименты включали в себя поиск автоматов с двумя способами представления (с использованным в [13–15] и с предлагаемым в настоящей работе), а также определение качества сгенерированных автоматов в авиасимуляторе *FlightGear*. Два упомянутых способа представления автоматов далее будем называть *предыдущим* и *предлагаемым* со-

Таблица 2. Свойства наборов обучающих примеров

Набор обучающих примеров	Модель самолета	Число обучающих примеров	Средняя длина обучающего примера в наборе, с
“Мертвая петля”	PA-34 Seneca II	33	35.4
“Бочка”	Gloster Meteor	28	27.2
Разворот	Gloster Meteor	19	56.3

Таблица 3. Используемые параметры муравьиного алгоритма

Способ представления автомата	N_{ants}	N_{mut}	n_{stag}	p_{new}
Предлагаемый	5	13	55	0.1428
Предыдущий	2	29	8	0.2471

Таблица 4. Медианные значения ФП, достигнутые на различных экземплярах задачи

$ S $	Способ представления автомата	“Мертвая петля”, f	“Бочка”, f	Разворот, f	Разворот, f_{Δ}
3	Предлагаемый	0.9856	0.9854	0.9892	0.99748
	Предыдущий	0.9812	0.9832	0.9894	0.99731
4	Предлагаемый	0.9866	0.9863	0.9898	0.99749
	Предыдущий	0.9836	0.9856	0.9901	0.99734
5	Предлагаемый	0.9873	0.9868	0.9901	0.99751
	Предыдущий	0.9842	0.9858	0.9902	0.99734

ответственно. Предварительно в авиасимуляторе были вручную записаны три набора обучающих примеров, описывающих выполнение “мертвой петли”, “бочки” и разворота самолета в горизонтальной плоскости на 180° под углом 60° . Некоторые свойства наборов обучающих примеров приведены в табл. 2. При генерации автоматов для выполнения “мертвой петли” и “бочки” в качестве ФП применялась f , но при работе с автоматами с предыдущим способом представления штрафы P_{τ} не использовались ($K=0$): при сложной структуре такта автомата цель введения штрафов не достигается. На третьем наборе обучающих примеров, описывающем указанный разворот, сравнивается эффективность двух ФП: f и f_{Δ} . Функция f_{Δ} была опробована и на первых двух обучающих наборах, но эксперименты показали, что ее применение на них не оправдано.

Для каждого набора обучающих примеров был вручную подобран набор предикатов для генерации автоматов с предыдущим способом представления и два набора – предикатов и составляющих выходных воздействий – для генерации автоматов с предлагаемым способом представления. Процесс подбора этих величин является итеративным: изначально интуитивно подобранный набор корректируется до тех пор, пока поведение автоматически генерируемых автоматов не становится приемлемым. Однако хороший набор предикатов для выполнения разворота с помощью автоматов с предыдущим способом представления так и не был подобран, что сказалось на результатах моделирования поведения автоматов, которые будут приведены в разд. 4.2. Заметим, что ручной подбор предикатов является недостатком базового метода, который в настоящей работе сохраняется.

3.1. Вычислительные эксперименты. Для каждой из четырех рассматриваемых комбинаций набора обучающих примеров и ФП (совокупность этих двух параметров будем называть *экземпляром задачи*), для числа состояний $|S| = 3, 4, 5$ и для двух способов представления автоматов были проведены по 50 запусков муравьиного алгоритма [18]. Алгоритм, предложенный в [18], предполагает использование феромона при выборе муравьем следующей дуги своего пути. Однако в настоящей работе была выбрана более простая стратегия, которая не оказалась менее эффективной, – равновероятный выбор следующей дуги.

Число муравьев N_{ants} , а также другие не связанные с феромоном параметры алгоритма N_{mut} , n_{stag} и p_{new} (описание параметров приведено в [18]) были настроены при помощи программного средства *irace* [19]. Применение специального инструмента для настройки параметров алгоритма более предпочтительно, чем ручная их настройка, поскольку последняя подвержена влиянию человеческого фактора. В табл. 3 приведены значения параметров, найденные при помощи средства *irace*. При настройке параметров учитывалось поведение различных наборов их значений на трех из четырех рассматриваемых экземплярах задач. При этом был исключен экземпляр с f_{Δ} в качестве ФП, поскольку он оказался наиболее прост для поисковой оптимизации и использова-

Таблица 5. Медианные числа вычислений ФП на различных экземплярах задачи

$ S $	Способ представления автомата	“Мертвая петля”, f	“Бочка”, f	Разворот, f	Разворот, f_{Δ}
3	Предлагаемый	8556	8472	8339	7556
	Предыдущий	9979	10071	10195	11400
4	Предлагаемый	9234	8934	9119	9261
	Предыдущий	12565	14883	15256	15456
5	Предлагаемый	9008	10048	8982	10068
	Предыдущий	17850	18431	17436	15441

Таблица 6. Медианное отклонение угла крена, град, для автоматов, сгенерированных для различных экземпляров задачи

$ S $	Способ представления автомата	“Мертвая петля”, f	“Бочка”, f	Разворот, f	Разворот, f_{Δ}
3	Предлагаемый	1.71	16.52	4.80	2.07
	Предыдущий	6.37	18.56	50.29	21.60
4	Предлагаемый	2.41	15.35	4.10	1.99
	Предыдущий	6.32	21.86	57.04	22.88
5	Предлагаемый	3.21	14.74	4.07	1.99
	Предыдущий	9.54	22.99	45.83	24.32

ние его для настройки параметров алгоритмов, скорее всего, снизило бы эффективность полученных параметров на сложных экземплярах.

Критерием останова алгоритма являлась стагнация в течение 5000 вычислений ФП. В табл. 4 приведены медианные значения ФП, достигнутые муравьиным алгоритмом на различных экземплярах рассматриваемой задачи. Приведенные значения ФП несравнимы между различными экземплярами задачи, но сравнимы между различными представлениями автоматов. Отметим, что штраф за смену состояний автомата, присутствующий в ФП автоматов с предлагаемым способом представления, в результате оптимизации достигал малых значений и не оказывал существенного влияния на значение ФП, что сделало такое сравнение возможным. Из табл. 4 видно, что использование предлагаемого способа представления автоматов увеличивает значение ФП на “мертвой петле” и “бочке”.

Кроме того, имеет смысл учитывать время, требуемое для поиска автоматов с тем или иным способом представления. В табл. 5 приведены медианные числа вычислений ФП для различных экземпляров задачи. С одной стороны, для генерации автоматов с предыдущим способом представления, как правило, требуется больше вычислений ФП. С другой стороны, расстановка выходных воздействий для автоматов с предлагаемым способом представления требовала до 60% больше процессорного времени. В любом случае время генерации автоматов обычно не превосходило 10 мин на персональном компьютере с четырехъядерным процессором *Intel Core i7–2670QM* при параллельном вычислении ФП различных автоматов.

3.2. Эксперимент с использованием компьютерной симуляции. Сгенерированные муравьиным алгоритмом автоматы были протестированы в авиасимуляторе. Тестирование каждого автомата заключалось в его десятикратном запуске в условиях, аналогичным условиям записи обучающих примеров. Для измерения качества автоматов использовались два критерия, показывающие средние отклонения углов тангажа и крена при тестировании от соответствующих углов, записанных в обучающих примерах. Более точно, в качестве критерия качества для угла тангажа применялось среднее арифметическое значений $|\alpha_{i,t}^{\text{test}} - \alpha_{j,t}^{\text{run}}|$ ($i = \overline{1, N}, t = \overline{1, L_i}, j = \overline{1, 10}$), где $\alpha_{i,t}^{\text{test}}$ – угол тангажа в момент времени t обучающего примера i , а $\alpha_{j,t}^{\text{run}}$ – угол тангажа в момент времени t запуска j автомата в авиасимуляторе. Аналогичный критерий был использован для угла крена.

Отбор автоматов для тестирования в моделировании проводился следующим образом. Для каждой пары из экземпляра задачи и числа состояний были рассмотрены лучшие (по максималь-

Таблица 7. Медианное отклонение угла тангажа, град, для автоматов, сгенерированных для различных экземпляров задачи

$ S $	Способ представления автомата	“Мертвая петля”, f	“Бочка”, f	Разворот, f	Разворот, f_{Δ}
3	Предлагаемый	17.21	3.20	1.95	0.50
	Предыдущий	20.54	4.44	7.58	3.85
4	Предлагаемый	23.04	2.51	1.42	0.48
	Предыдущий	22.11	4.08	6.79	4.61
5	Предлагаемый	25.27	2.43	1.36	0.47
	Предыдущий	24.44	4.68	7.83	5.31

ному достигнутому значению ФП) автоматы каждого из 50 проведенных запусков муравьиного алгоритма. Эти автоматы и тестировались в авиасимуляторе. Таблицы 6, 7 содержат медианные значения критериев качества для описанных групп из 50 автоматов.

Различия значений критериев качества для одинаковых фигур пилотажа и чисел состояний, но для разных способов представления автомата были проверены при помощи U -критерия Манна–Уитни (всего для каждого из двух критериев качества проведено 12 статистических тестов, в каждом из которых сравнивались две группы из 50 значений). Статистически значимыми ($p < 0.001$, p -значения откорректированы по методу Холма) оказались различия для всех пар выборок, кроме двух – значений для угла тангажа при выполнении мертвой петли автоматами с четырьмя и пятью состояниями. На основе таблиц можно сделать следующие выводы:

значения критериев качества в целом лучше (меньше) для автоматов с предлагаемым способом представления;

использование f_{Δ} улучшает качество выполнения разворота для автоматов независимо от их способа представления.

Отметим, что различия значений критериев качества в действительности соответствуют качественным различиям поведения автоматов в моделировании. Автоматы с предложенным способом представления лучше выравнивают положения самолета в конце “мертвой петли” и “бочки”, а также выполняют “бочку” без рывков, которые иногда свойственны автоматам с предыдущим способом представления. Если говорить о развороте, то медианные значения критериев качества для этой фигуры при использовании автоматов с предыдущим способом представления достаточно велики. Причина этого состоит в том, что ни один из этих автоматов не смог справиться с выполнением разворота от начала и до конца. В то же время большинство рассмотренных автоматов с предлагаемым способом представления выполняли разворот и отличались лишь качеством поддержания его угла. Как оказалось, постоянные выходные воздействия на переходах автоматов с предыдущим способом представления не оказались достаточными для поддержания угла крена вблизи требуемого. Для предложенного способа представления автоматов значения критериев качества существенно меньше, а рассмотренные автоматы оказались способны выполнить разворот, при этом автоматы, построенные с использованием f_{Δ} , лучше поддерживали угол крена в 60° .

Итак, проведенное экспериментальное исследование свидетельствует об успешности предлагаемого способа представления автоматов на различных наборах обучающих примеров и функции приспособленности f_{Δ} на одном из наборов обучающих примеров, а также о достаточно высокой производительности метода.

Снимки экрана с истребителем *Gloster Meteor*, выполняющим “бочку” под управлением одного из автоматов, сгенерированных с использованием предложенного в работе подхода, приведены на рис. 5, *a–e*. Снимки показывают 2-ю, 5-ю, 9-ю, 14-ю, 19-ю и 23-ю секунды выполнения фигуры пилотажа. На каждый снимок экрана справа наложен автомат, текущее состояние которого выделено полужирной окружностью. Кроме того, видеозаписи выполнений различных фигур пилотажа автоматами с предложенным способом представления доступны по следующим адресам:

“мертвая петля”: http://www.youtube.com/watch?v=vnJ_-gMLjx8;

“бочка”: <http://www.youtube.com/watch?v=2GxzcPx0XNw>;

разворот: <http://www.youtube.com/watch?v=n9q5FmCYs6M>.

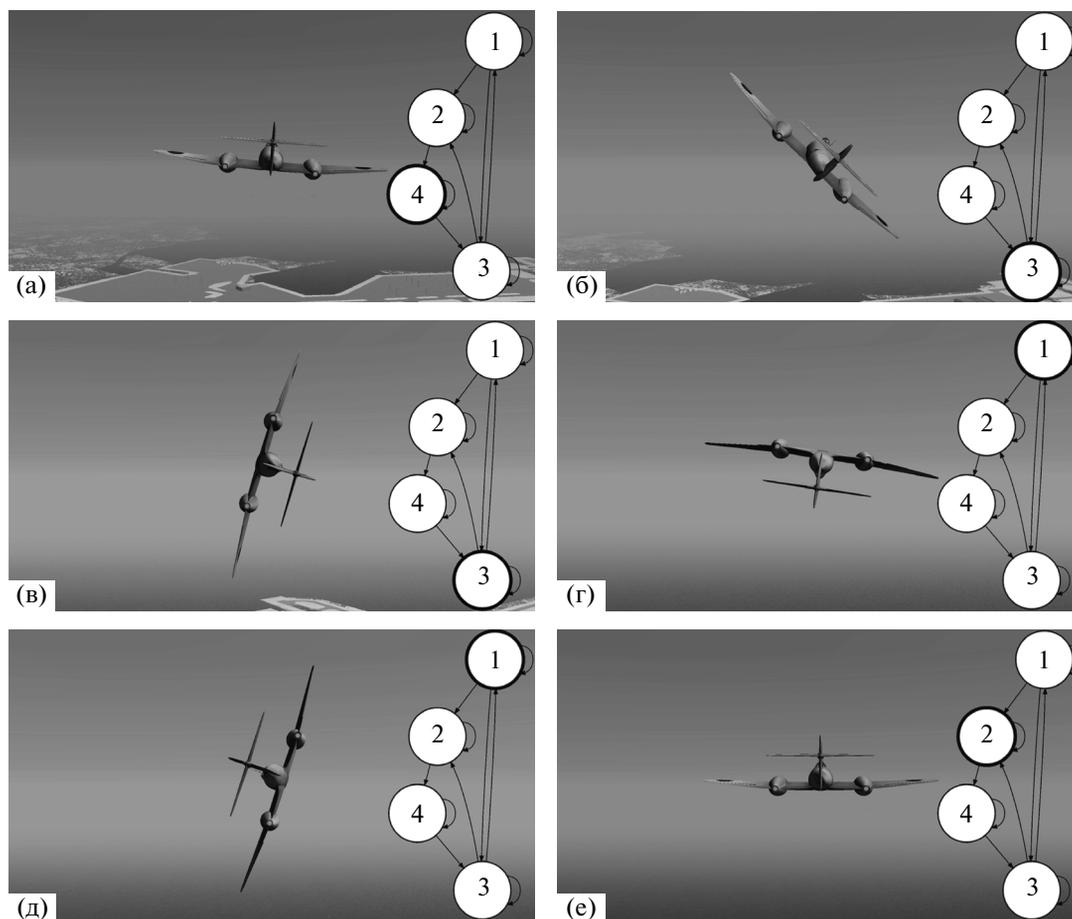


Рис. 5. Выполнение разворота под управлением сгенерированного автомата (снимки экрана)

Различные состояния автоматов соответствуют различным отрезкам выполнения фигур пилотажа. Например, автомат, показанный на видеозаписи “бочки”, в процессе выполнения фигуры последовательно сменяет состояния 4 (начало поворота), 1 (первая половина поворота), 2 (вторая половина поворота), 3 (выравнивание положения самолета).

Заключение. В работе предложена модификация метода генерации конечных управляющих автоматов с непрерывными воздействиями. На примере задачи управления моделью самолета продемонстрированы преимущества предложенных подходов по сравнению с методами, описанными в [13–15]. Новый подход к представлению автоматов улучшил качество выполнения рассмотренных в работе фигур пилотажа. Кроме того, реализация автоматов в виде автоматов Мура делает их более легкими для интерпретации и анализа. Предложенные подходы также позволили построить автомат, выполняющий разворот в горизонтальной плоскости, что оказалось невозможным при использовании ранее разработанных методов.

Полученные результаты могут быть применены к разработке систем управления для беспилотных летательных аппаратов (БПЛА). Так, разработкой БПЛА и систем управления для них занимается компания Транзас [20], при этом модели БПЛА испытываются в симуляторе *FlightGear*. Кроме того, один из возможных путей развития настоящего исследования — это применение подхода к управлению мобильными роботами, для которых также существуют симуляторы. Этот путь привлекателен простотой применения построенных автоматных систем управления на реальных моделях.

ПРИЛОЖЕНИЕ

Сведение задачи расстановки выходных воздействий к решению набора систем линейных алгебраических уравнений. Построим систему

уравнений, решение которой требуется для определения выходных воздействий автомата, максимизирующих ФП на заданном каркасе. Рассмотрим случай использования f (случай с f_Δ аналогичен). Напомним, что оптимизируемыми величинами являются числа $r_{s,i,j} \in \mathbb{R}$, для которых составляющая выходных воздействий $v_{j,i}$ (i -я составляющая для j -го управляющего параметра) является значимой в состоянии s (остальные числа заведомо равны нулю). Переиндексируем оптимизируемые величины и обозначим их как $\hat{r}_{l,j}$ ($j = \overline{1, c}$, $l = \overline{1, M}$, где $M = k_j |S|$, а k_j – число значимых составляющих для управляющего параметра j). Когда индекс управляющего параметра будет зафиксирован, будем его опускать и обозначать оптимизируемые величины как $\hat{r}_1, \dots, \hat{r}_M$. Каждому числу \hat{r}_l , $l = \overline{1, M}$, будут соответствовать некоторое состояние \hat{s}_l и некоторая составляющая выходных воздействий \hat{v}_l . Значение \hat{v}_l , вычисленное по $I_{i,t,j}$ ($j = \overline{1, p}$), обозначим как $\hat{v}_{i,t,l}$.

Теперь поясним возможность решения оптимизационной задачи для каждого управляющего параметра в отдельности. Для фиксированного каркаса штраф P_τ не зависит от выходных воздействий, поэтому задача максимизации (2.3) сводится к следующей:

$$1 - \sqrt{\frac{1}{N} \sum_{i=1}^N \rho^2(\tilde{O}_i, O_i)} \rightarrow \max_{\hat{r}_{j,l=1, \dots, M, j=1, \dots, c}}.$$

Раскрывая (2.2) и делая элементарные преобразования, получим

$$\sum_{i=1}^N \frac{1}{L_i} \sum_{t=2}^{L_i} \sum_{j=1}^c \left(\frac{\tilde{O}_{i,t,j} - O_{i,t,j}}{M_j - m_j} \right)^2 \rightarrow \min_{\hat{r}_{j,l=1, \dots, M, j=1, \dots, c}}.$$

Отсюда видно, что сумму можно минимизировать для каждого управляющего параметра j отдельно. Для фиксированного управляющего параметра $j = j_0$ задача приобретает следующий вид:

$$g_{j_0} = \sum_{i=1}^N \frac{1}{L_i} \sum_{t=2}^{L_i} (\tilde{O}_{i,t,j_0} - O_{i,t,j_0})^2 \rightarrow \min_{\hat{r}_1, \dots, \hat{r}_M}. \quad (\text{П.1})$$

Выразим \tilde{O}_{i,t,j_0} через $\hat{r}_1, \dots, \hat{r}_M$. Для этого введем дополнительное обозначение: $\beta_{i,t,l} = \hat{v}_{i,t,l}$, если \hat{s}_l – текущее состояние автомата при прохождении им i -го обучающего примера перед тактом t , иначе $\beta_{i,t,l} = 0$. Переписывая (2.1) в новых обозначениях, получим

$$\tilde{O}_{i,t,j} = \tilde{O}_{i,t-1,j} + \sum_{l=1}^M \beta_{i,t-1,l} \hat{r}_l.$$

Используя равенство $\tilde{O}_{i,1,j} = O_{i,1,j}$, формулу можно замкнуть:

$$\tilde{O}_{i,t,j} = O_{i,1,j} + \sum_{l=1}^M \sum_{t'=1}^{t-1} \beta_{i,t',l} \hat{r}_l.$$

С применением обозначения

$$\alpha_{i,t,l} = \sum_{t'=1}^{t-1} \beta_{i,t',l}$$

она переписывается следующим образом:

$$\tilde{O}_{i,t,j} = O_{i,1,j} + \sum_{l=1}^M \alpha_{i,t,l} \hat{r}_l. \quad (\text{П.2})$$

Теперь подставим (П.2) в (П.1) и приравняем нулю частные производные g_{j_0} :

$$\frac{\partial g_{j_0}}{\partial \hat{r}_{l_0}} = 2 \sum_{i=1}^N \frac{1}{L_i} \sum_{t=2}^{L_i} \alpha_{i,t,l_0} \left(O_{i,1,j_0} + \sum_{l=1}^M \alpha_{i,t,l} \hat{r}_l - O_{i,t,j_0} \right) = 0, \quad l_0 = \overline{1, M}.$$

Это выражение можно представить в более удобной форме:

$$\sum_{l=1}^M \left(\sum_{i=1}^N \frac{1}{L_i} \sum_{t=2}^{L_i} \alpha_{i,t,l_0} \alpha_{i,t,l} \right) \hat{r}_l = \sum_{i=1}^N \frac{1}{L_i} \sum_{t=1}^{L_i} \alpha_{i,t,l_0} (O_{i,t,j_0} - O_{i,1,j_0}), \quad l_0 = \overline{1, M}. \quad (\text{П.3})$$

Из изложенного следует, что оптимальные выходные воздействия для заданного управляющего параметра j_0 могут быть найдены путем решения набора уравнений (П.3), представляющих собой систему из M линейных алгебраических уравнений. Для нахождения ее матрицы потребуется $\mathcal{O}(M^2(L_1 + \dots + L_N))$ времени, а для ее решения методом Гаусса — $\mathcal{O}(M^3)$. Система должна быть решена для каждого управляющего параметра, однако существует следующая возможность сократить объем вычислений.

Если наборы составляющих выходных воздействий для некоторых управляющих параметров совпадают и для этих параметров используются одни и те же маски значимости составляющих выходных воздействий, то левые части систем вида (П.3) для этих управляющих параметров также совпадают. Это позволяет не вычислять их повторно. В настоящем исследовании было решено применить такую оптимизацию к элеронам и рулю направления, поскольку для них оказалось целесообразно использовать одинаковые наборы составляющих выходных воздействий: оба органа управления применяются в основном для управления углом крена, а соответствующие составляющие зависят от этого угла. Для оставшегося органа управления — руля высоты — использование таких же составляющих неразумно, поскольку он служит для управления углом тангажа. Применение же совмещенного набора составляющих выходных воздействий, основанных и на угле крена и на угле тангажа, для всех трех органов управления привело бы к тому, что составляющие, основанные на одном угле, использовались бы сразу для всех органов управления, что нежелательно. Это произошло бы из-за совмещения масок значимости, которые также необходимы для применения оптимизации.

СПИСОК ЛИТЕРАТУРЫ

1. Harel D., Pnueli A. On the Development of Reactive Systems // “Logic and Models of Concurrent Systems”. NATO Advanced Study Institute on Logic and Models for Verification and Specification of Concurrent Systems. Springer-Verlag, 1985. P. 477–498.
2. Harel D., Politi M. Modeling Reactive Systems with Statechart. The Statemate Approach. N.Y.: McGraw-Hill, 1998.
3. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. СПб: Питер, 2011. 176 с. http://is.ifmo.ru/books/_book.pdf.
4. Walkinshaw N., Bogdanov K., Holcombe M. et al. Reverse Engineering State Machines by Interactive Grammar Inference // Proc. 14th Working Conf. on Reverse Engineering (WCRE '07). Vancouver: IEEE Computer Society Press, 2007. P. 209–218.
5. Walkinshaw N., Taylor R., Derrick J. Inferring Extended Finite State Machine Models from Software Executions // Proc. 20th Working Conf. on Reverse Engineering (WCRE '13). Koblenz: IEEE Computer Society Press, 2013. P. 301–310.
6. Царев Ф.Н., Шалыто А.А. Применение генетического программирования для генерации автоматов в задаче об “умном муравье” // Тр. IV Междунар. научно-практической конф. “Интегрированные модели и мягкие вычисления в искусственном интеллекте”. Т. 2. М.: Физматлит, 2007. С. 590–597. http://is.ifmo.ru/genalg/_ant_ga.pdf.
7. Koza J.R. Genetic Programming: on the Programming of Computers by Means of Natural Selection. Cambridge: MIT Press, 1992.
8. Курейчик В.М. Генетические алгоритмы. Состояние. Проблемы. Перспективы // Изв. РАН. ТиСУ. 1999. № 1. С. 144–160.
9. Курейчик В.М., Родзин С.И. Эволюционные алгоритмы: генетическое программирование // Изв. РАН. ТиСУ. 2002. № 1. С. 127–137.
10. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. М.: Физматлит, 2006. 320 с.
11. Поликарпова Н.И., Точилин В.Н., Шалыто А.А. Метод сокращенных таблиц для генерации автоматов с большим числом входных переменных на основе генетического программирования // Изв. РАН. ТиСУ. 2010. № 2. С. 100–117.

12. Царев Ф.Н. Метод построения управляющих конечных автоматов на основе тестовых примеров с помощью генетического программирования. // Информационно-управляющие системы. 2010. № 5. С. 31–36.
13. Александров А.В., Казаков С.В., Сергушичев А.А. и др. Применение эволюционного программирования на основе обучающих примеров для генерации конечных автоматов, управляющих объектами со сложным поведением // Изв. РАН. ТИСУ. 2013. № 3. С. 85–100.
14. Buzhinsky I., Ulyantsev V., Shalyto A. Test-Based Induction of Finite-State Machines with Continuous Output Actions // Proc. 7th IFAC Conf. on Manufacturing Modelling, Management, and Control (MIM'13). St. Petersburg: IFAC, 2013. P. 1049–1054.
15. Бужинский И.П., Ульянцев В.И., Чивилихин Д.С. и др. Генерация управляющих автоматов по обучающим примерам на основе муравьиного алгоритма // Изв. РАН. ТИСУ. 2014. № 2. С. 111–121.
16. FlightGear [Электронный ресурс]. Режим доступа <http://www.flightgear.org/> свободный. Яз. англ. (дата обращения 29.09.14).
17. Dorigo M., Stützle T. Ant Colony Optimization. MIT Press, US, 2004.
18. Chivilikhin D., Ulyantsev V. Learning Finite-State Machines with Ant Colony Optimization // Lecture Notes in Computer Science. 2012. V. 7461. P. 268–275.
19. López-Ibáñez M., Dubois-Lacoste J., Stützle T. et al. The irace Package, Iterated Race for Automatic Algorithm Configuration // Technical Report TR/IRIDIA/2011-004, IRIDIA, Université libre de Bruxelles, Belgium, 2011.
20. Transas [Электронный ресурс]. Режим доступа <http://www.transas.ru/> свободный. Яз. рус. (дата обращения 29.09.14).