



Российская Академия Наук

А Т АВТОМАТИКА И ТЕЛЕМЕХАНИКА

Журнал основан в 1936 году

Выходит 12 раз в год

6

И Ю Н Ъ

Москва

2021

Учредители журнала:

Отделение энергетики, машиностроения, механики и процессов управления РАН,
Институт проблем управления им. В.А. Трапезникова РАН (ИПУ РАН),
Институт проблем передачи информации им. А.А. Харкевича РАН (ИППИ РАН)

Главный редактор:

Галяев А.А.

Заместители главного редактора:

Соболевский А.Н., Рубинович Е.Я., Хлебников М.В.

Ответственный секретарь:

Родионов И.В.

Редакционный совет:

Васильев С.Н., Желтов С.Ю., Каляев И.А., Кулешов А.П., Куржанский А.Б.,
Мартынюк А.А. (Украина), Пешехонов В.Г., Поляк Б.Т., Попков Ю.С.,
Рутковский В.Ю., Федосов Е.А., Черноусько Ф.Л.

Редакционная коллегия:

Алескерев Ф.Т., Бахтадзе Н.Н., Бобцов А.А., Виноградов Д.В., Вишневский В.М.,
Воронцов К.В., Глумов В.М., Граничин О.Н., Губко М.В., Каравай М.Ф.,
Кибзун А.И., Краснова С.А., Красносельский А.М., Крищенко А.П.,
Кузнецов Н.В., Кузнецов О.П., Кушнер А.Г., Лазарев А.А., Ляхов А.И.,
Маликов А.И., Матасов А.И., Меерков С.М. (США), Миллер Б.М.,
Михальский А.И., Мунасыпов Р.А., Назин А.В., Немировский А.С. (США),
Новиков Д.А., Олейников А.Я., Пакшин П.В., Пальчунов Д.Е.,
Поляков А.Е. (Франция), Рапопорт Л.Б., Рублев И.В., Степанов О.А.,
Уткин В.И. (США), Фрадков А.Л., Хрусталеv М.М., Цыбаков А.Б. (Франция),
Чеботарев П.Ю., Щербаков П.С.

Адрес редакции: 117997, Москва, Профсоюзная ул., 65

Тел./факс: (495) 334-87-70

Электронная почта: redacsia@ipu.ru

Зав. редакцией *Е.А. Мартехина*

Москва

ООО «Объединённая редакция»

© 2021 г. И.А. ХОДАШИНСКИЙ, д-р техн. наук (hodashn@rambler.ru)
(Томский государственный университет систем управления и радиоэлектроники)

МЕТОДЫ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ РОЕВЫХ АЛГОРИТМОВ ОПТИМИЗАЦИИ¹

Роевые алгоритмы относятся к классу популяционных метаэвристических методов оптимизации. Несмотря на использование различных метафор, большинство роевых алгоритмов имеют схожую структуру, в них можно выделить такие общие компоненты, как инициализация популяции решений, диверсификация и интенсификация решений. На основании концепции общности был проведен анализ ключевых подходов, методов и способов повышения эффективности роевых алгоритмов оптимизации. В обзоре роевые алгоритмы оптимизации рассматриваются как совокупность операторов, без детального обсуждения каждого алгоритма, основное внимание сосредоточено на анализе ключевых компонентов алгоритмов. Основная идея повышения эффективности заключается в соблюдении баланса между диверсификацией и интенсификацией. В этом контексте рассмотрены механизмы поддержки популяционного разнообразия, методы настройки и регулировки параметров роевых алгоритмов, подходы к гибридизации алгоритмов, обозначено несколько открытых проблем, связанных с темой обзора.

Ключевые слова: оптимизация, метаэвристики, роевые алгоритмы, диверсификация, интенсификация.

DOI: 10.31857/S0005231021060015

1. Введение

Многие сложные проблемы в области науки, техники, экономики и бизнеса могут быть сформулированы как задачи оптимизации. Традиционные методы оптимизации, основанные на производных, доказали свою эффективность в решении различных типов задач оптимизации. Однако применение этих методов связано с рядом ограничений, таких как “застывание” в локальных оптимумах, сильная зависимость от начальных условий, большая вычислительная сложность и неприменимость к конкретным классам целевых функций. Это привело к необходимости разработки альтернативных методов, которые могли бы преодолеть эти ограничения [1].

Поиск оптимальных решений NP-трудных задач часто выполняется с помощью методов, получивших название “метаэвристики”. Прежде чем перейти

¹ Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект № 19-17-50050).

к рассмотрению метаэвристик, дадим рабочее определение понятию эвристика — это способ нахождения “достаточно хороших” решений без доказательства оптимальности найденных решений. Качество или эффективность решения могут быть выражены через точность, стабильность, экономию памяти или времени. Термин “мета” указывает на обобщенность и “методологию верхнего уровня”. Как правило, эвристики предназначены для решения конкретных задач, в то время как метаэвристики независимы от решаемых проблем. По мнению Шона Люка, автора известной книги “Основы метаэвристик” [2], метаэвристика — общее, но неудачное название стохастического алгоритма оптимизации, который применяется в качестве последней надежды на пути к решению задачи оптимизации. Стохастическая оптимизация — большой класс алгоритмов и методов, использующих случайность для поиска оптимального (субоптимального) решения сложных задач. Применяются метаэвристики в том случае, когда 1) *неизвестен* способ поиска оптимального решения; 2) *недостаточно* имеющейся априорной информации; 3) *невозможен* полный перебор; 4) *можно* проверить качество решения, т.е. неизвестна аналитическая форма целевой функции $f(\mathbf{x})$, но ее значение может быть вычислено для конкретного решения \mathbf{x} .

Метаэвристики не используют вычисление производных целевой функции, однако, в отличие от детерминированных методов оптимизации, метаэвристики позволяют находить за приемлемое время удовлетворительные решения задач большой размерности. Широкому применению метаэвристик способствуют увеличение вычислительной мощности компьютеров и развитие параллельных архитектур [3]. На рис. 1 приведено распределение количества статей, индексируемых в базе данных Scopus, в которых упоминалось слово “metaheuristic”, а на рис. 2 указано распределение статей по отраслям знаний.

Метаэвристические методы классифицируются по различным основаниям. Методы, формирующие последующее решение на основе единственного предыдущего решения, относятся к траекторным методам, типичными представителями которых являются алгоритм имитации отжига [4] и поиск с запретами [5]. Популяционные методы формируют решение на основе своего предыдущего опыта и информации о лучших решениях в популяции. Траекторные методы в большей степени ориентированы на интенсификацию решений, тогда как базовая метаэвристика, основанная на популяциях, в большей степени ориентирована на диверсификацию [3]. Популяционные методы разделены на четыре класса [6]: эволюционные алгоритмы, основанные на процессах эволюции; алгоритмы, имитирующие физические процессы; роевые алгоритмы, основанные на поведенческих процессах социальных животных; алгоритмы, имитирующие поведение человека.

Концепция отсутствия универсального алгоритма оптимизации, оформленная в виде “теоремы о бесплатных завтраках” [7], фактически констатирует отсутствие необходимости оценивать эффективность алгоритмов оптимизации *в среднем* для всех проблем. Теорема указывает на важность понимания взаимосвязи между компонентами алгоритма оптимизации и спецификой решаемой проблемы и побуждает специалистов к проведению новых исследований в области роевых алгоритмов. Эти исследования можно разделить на три большие категории [8]. Первая категория связана с исследованием и

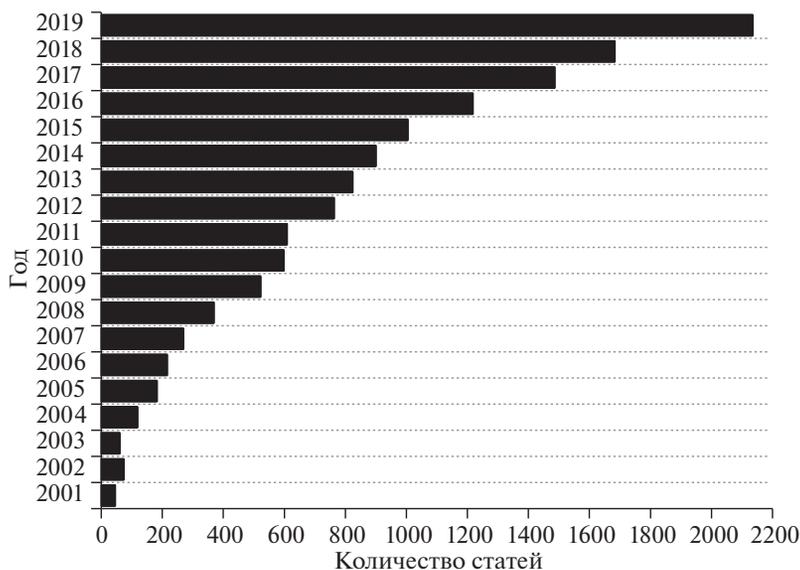


Рис. 1. Распределение количества статей, индексируемых в базе данных Scopus, в которых упоминалось слово “metaheuristic”.

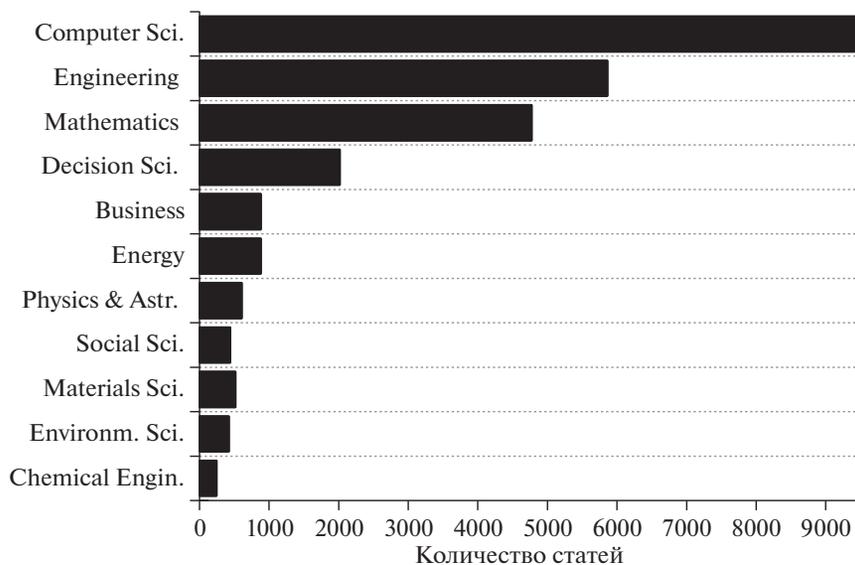


Рис. 2. Распределение по отраслям знаний статей, в которых упоминалось слово “metaheuristic”.

модификацией роевого алгоритма для улучшения его сходимости, точности решения, уменьшения времени выполнения. Вторая категория относится к исследованию типов проблем, которые могут быть решены с помощью данного алгоритма. Большинство роевых алгоритмов изначально предназначались для решения однокритериальных задач безусловной оптимизации. После того как алгоритм показал свою эффективность в решении указанных задач,

проводится его модификация для решения других типов задач, таких как задачи одноцелевой оптимизации с ограничениями, задачи многоцелевой оптимизации, задачи многоцелевой оптимизации с ограничениями, комбинаторные задачи оптимизации и т.д. Третья категория связана с исследованиями практических приложений алгоритмов.

Цель обзора связана с первой категорией исследований. Существующие обзоры, посвященные метаэвристической оптимизации, по большей части сосредоточены на эволюционных алгоритмах и в меньшей степени на роевых алгоритмах. Обзоры, за редким исключением, сосредоточены на рассмотрении метафор, лежащих в основе метаэвристических алгоритмов, и не раскрывают общего понимания механизмов генерации и модификации решений. В представленном обзоре роевые алгоритмы оптимизации рассматриваются как совокупность операторов, без детального обсуждения каждого алгоритма, основное внимание сосредоточено на анализе ключевых компонентов роевых алгоритмов.

Статья организована следующим образом. В разделе 2 дается краткое описание основных понятий и терминологии роевых алгоритмов. Так как алгоритм роящихся частиц является прототипом для реализации многих последующих роевых алгоритмов, его описанию посвящен раздел 3. Процедуры инициализации рассматриваются в разделе 4. В разделе 5 приведены некоторые меры популяционного разнообразия, а механизмы поддержки разнообразия — в разделе 6. Проблемы настройки и регулировки значений параметров рассматриваются в разделе 7. В разделе 8 дается краткое описание основных подходов к гибридизации роевых алгоритмов. Открытые проблемы кратко рассмотрены в разделе 9. В разделе 10 подведены итоги.

2. Основные понятия и терминология

Алгоритмы оптимизации ищут точку в пространстве поиска, в которой значение целевой функции максимально или минимально в зависимости от задачи оптимизации.

Рассмотрим задачу поиска минимального значения функции

$$\min f(\mathbf{x}), \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max},$$
$$\mathbf{x}_{\min} = (x_{1,\min}, x_{2,\min}, \dots, x_{D,\min}), \quad \mathbf{x}_{\max} = (x_{1,\max}, x_{2,\max}, \dots, x_{D,\max}),$$

\mathbf{x}_{\min} и \mathbf{x}_{\max} определяют пространство поиска, $f : \mathbb{R}^D \rightarrow \mathbb{R}$ — целевая функция D -мерного векторного аргумента, которая оценивает решение $\mathbf{x} = (x_1, x_2, \dots, x_D)$.

Большинство традиционных методов оптимизации требуют выполнения определенных условий, таких как выпуклость, непрерывность и т.д. Принимая во внимание ограничения традиционных методов, исследователи предложили использовать методы стохастической оптимизации, в частности роевые алгоритмы.

Большинство роевых алгоритмов оптимизации основаны на метафоре природных или техногенных процессов. Однако анализ работы метаэвристического алгоритма должен быть выполнен с использованием общепринятой тер-

минологии в области оптимизации, а не на языке метафоры. Реальная ценность использования таких метафор в алгоритмах оптимизации часто неясна и вызывает ряд критических замечаний [9, 10]. Основное замечание связано с отсутствием алгоритмических различий в так называемых новых метаэвристиках по сравнению с ранее опубликованными. Автор публикации [11], проведя строгий анализ гармонического поиска [12], приходит к выводу, что эта метаэвристика является частным случаем хорошо известной эволюционной стратегии. В [13] показано, что оптимизация на основе черных дыр [14] является упрощенной версией алгоритма роящихся частиц. Анализ алгоритма под названием “интеллектуальные капли воды” [15] позволил авторам [10] сделать вывод о том, что все основные алгоритмические компоненты этой метаэвристики являются упрощениями или частными случаями алгоритма муравьиной колонии. Другое важное критическое замечание связано с сомнением по поводу полезности использования некоторых метафор при разработке алгоритмов оптимизации. Авторы публикаций [9, 10, 13] указывают на то, что, хотя некоторые метафоры имеют хорошее математическое описание, созданные на их основе метаэвристики либо существенно модифицируют указанные описания, либо вообще не соответствуют исходной метафоре. В [9] утверждается, что в большинстве случаев метафоры совершенно не нужны для описания алгоритма оптимизации, роль метафор часто преувеличивается их авторами.

Роевые алгоритмы оптимизации могут рассматриваться как итеративные процедуры поиска, где рой — это множество взаимодействующих частиц, которые определяют потенциальное решение задачи оптимизации в D -мерном пространстве. Итоговое решение может быть представлено либо отдельной частицей (координаты частицы в пространстве поиска), либо группой частиц, где каждая частица представляет собой часть решения. Существуют две основные схемы кодирования возможных решений: векторное и представление в виде графа. В векторном типе представления каждое возможное решение кодируется строкой значений:

$$(1) \quad \mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD}) \quad i = 1, 2, \dots, N,$$

где N — размер популяции, D — размерность пространства поиска.

Многие задачи комбинаторной оптимизации, например задача коммивояжера, могут быть представлены в виде графа. В таком представлении частицы перемещаются от одного узла к другому, пока не пройдут все узлы, и не будет найдено полное решение проблемы (путь, соединяющий все города).

Схемы кодирования во многом определяют способ формирования новых вариантов решения. В большинстве роевых алгоритмов новые решения генерируются случайным образом или с использованием эвристики, которая объединяет существующие решения для получения нового. Для векторного представления типичный метод обновления выполняется на основе уравнения

$$(2) \quad \mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \Delta_i(t+1),$$

где $\mathbf{x}_i(t)$ — текущее решение, $\mathbf{x}_i(t+1)$ — обновленное решение, $\Delta_i(t)$ — модифицирующий вектор, t — текущая итерация. Модифицирующий вектор

может быть сформирован так: 1) на основе градиента; 2) перестановкой отдельных частей нескольких решений; 3) внесением направленного возмущения, вычисленного как разность двух решений, умноженная на равномерно распределенное случайное число; 4) внесением случайного возмущения, основанного на распределениях Гаусса, Коши, Леви.

В силу ограниченности объема статьи будет рассмотрен только векторный тип представления решения.

Большинство роевых алгоритмов имеют сходную структуру, в которой можно выделить ключевые компоненты и функции:

1) инициализация — задание значений параметрам алгоритма, формирование начальной популяции;

2) диверсификация (глобальный поиск) — генерация новых решений или исследование всего пространства поиска, когда модифицирующий вектор может принимать большие значения, обеспечивая разнообразие решений и не позволяя застрять в локальном оптимуме;

3) интенсификация (локальный поиск) — улучшение ранее найденных решений, выполняемое на ограниченном пространстве поиска, в этом случае модифицирующий вектор принимает малые значения.

Перечисленные выше компоненты реализуют вычислительные процедуры, например диверсификация почти всегда выполняется путем рандомизации, а интенсификация реализуется путем незначительного изменения лучших решений. Роевые алгоритмы можно классифицировать на две группы. В первую группу входят алгоритмы, в которых явно присутствуют компоненты диверсификации и интенсификации, например меметические алгоритмы [16], алгоритм на основе теории электромагнетизма [17], алгоритм ворон [18], прямой алгоритм муравьиной колонии [19]. Ко второй группе относятся алгоритмы с одним выражением, отвечающим как за диверсификацию, так и за интенсификацию, например алгоритм роящихся частиц [20], алгоритм “кукушкин поиск” [21]. Для соблюдения баланса между диверсификацией и интенсификацией необходима мера измерения популяционного разнообразия, определив которую, можно поддерживать баланс путем изменения параметров алгоритма или гибридизации.

Типичным представителем алгоритмов с одним выражением является алгоритм роящихся частиц, являющийся прототипом для реализации многих последующих алгоритмов и примером для иллюстрации методов повышения эффективности. Далее приведено краткое описание этого алгоритма.

3. Стандартный алгоритм роящихся частиц

Алгоритм роящихся частиц имеет мало настраиваемых параметров, небольшое требование к памяти, что делает его вычислительно эффективным. Алгоритм роящихся частиц [20] — это стохастический метод поиска, основанный на итеративном взаимодействии частиц, образующих рой. Перемещение частицы в пространстве поиска определяют три фактора: инерция, память, сотрудничество. Инерция подразумевает, что частица не может мгновенно изменить свое направление движения. Каждая частица имеет память

и хранит свою лучшую позицию в пространстве поиска. Известна частице и лучшая позиция роя. Зная эти две позиции, частица динамически изменяет скорость согласно ее собственному опыту и опыту полета других частиц. Таким образом, движение каждой частицы задается ее лучшей позицией, ее текущей скоростью, ускорением, заданным предыдущей позицией, и ускорением, заданным лучшей частицей в рое. Рой прекращает движение при выполнении хотя бы одного из следующих условий: рой достиг состояния равновесия, найдено оптимальное решение (ошибка меньше заданной), выполнено заданное количество итераций. Пространство поиска — множество действительных чисел, рой состоит из N частиц-решений. Позиция i -й частицы определяется вектором (1). Лучшая позиция, которую занимала i -я частица, определяется вектором $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, модифицирующий вектор в выражении (2) определяется скоростью частицы $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. Скорость и положение i -й частицы на итерации $t + 1$ определяют уравнения

$$\begin{aligned} \mathbf{x}_i(t + 1) &= \mathbf{x}_i(t) + \mathbf{v}_i(t + 1), \\ \mathbf{v}_i(t + 1) &= w \cdot \mathbf{v}_i(t) + c_1 \cdot \text{rand} \cdot (\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_2 \cdot \text{Rand} \cdot (\mathbf{p}_g(t) - \mathbf{x}_i(t)), \end{aligned}$$

где $i = 1, 2, \dots, N$; $\mathbf{v}_i(t)$ — вектор скорости i -й частицы на итерации t ; $\mathbf{x}_i(t)$ — координаты i -й частицы на итерации t ; c_1, c_2 — положительные коэффициенты ускорения; $\mathbf{p}_i(t)$ — лучшая позиция i -й частицы на первых t итерациях; $\mathbf{p}_g(t)$ — глобально лучшая позиция частицы в рое на первых t итерациях (задается индексом g); w — эмпирический коэффициент инерции; rand, Rand — случайные числа из диапазона $[0; 1]$.

Если коэффициенты ускорения и инерции не меняются в процессе работы алгоритма, то для соблюдения баланса между диверсификацией и интенсификацией значения этих коэффициентов рекомендуется устанавливать $c_1 = c_2 = 1,419$, а $w = 0,721$ [22].

Рой имеет своего рода топологию, описывающую взаимосвязи между частицами. Каждая частица имеет множество соседей. Соседями частицы может быть весь рой или некоторое его подмножество. Две наиболее часто используемые топологии известны как *gbest* и *lbest*. Топология *gbest* является полносвязной топологией, в отличие от *lbest*, в которой частица взаимодействует с ограниченным числом соседей. Алгоритм с топологией *gbest* быстро сходится и имеет больше шансов застрять в локальных оптимумах. Алгоритм с топологией *lbest* медленнее сходится, но более полно исследует пространство поиска и, как правило, находит лучшее решение [3].

4. Инициализация

Если целевая функция имеет несколько локальных оптимумов, то традиционные алгоритмы сходятся к одному из них, положение которого определяет процедура инициализации. Алгоритмы могут найти глобальный оптимум, если начальные решения принадлежат области притяжения глобального оптимума. Процедура инициализации играет важную роль и в роевых алгоритмах, где отсутствие разнообразия в популяции может привести к преждевременной сходимости и попаданию в локальный оптимум. Таким образом, эффективность методов оптимизации может быть повышена за счет разработки

методов генерации перспективных начальных решений. Однако на сегодняшний день отсутствуют системные исследования процесса инициализации и того, как начальные распределения решений могут влиять на эффективность роевых алгоритмов при решении определенного класса проблем [1].

Процедура инициализации включает задание начального размера популяции и собственно создание начальной популяции. *Размер популяции* влияет на надежность и вычислительные затраты алгоритма. Небольшой размер популяции может привести к быстрой сходимости к локальному оптимуму, большой размер увеличит вычислительные затраты и может привести к медленной сходимости [23]. Размер популяции может быть увязан с размерностью пространства поиска D . В [8] предлагается определять размер популяции в алгоритме роящихся частиц как

$$[10 + 2\sqrt{D}],$$

где $[\cdot]$ — целая часть числа. Для задач большой размерности ($D \geq 50$) в [24] размер популяции в алгоритме “кукушкин поиск” предлагается устанавливать равным D . Авторы [25] определяют размер популяции в модифицированном алгоритме искусственной пчелиной колонии как $5D$, а в модифицированном алгоритме серого волка размер популяции задается равным $3D$ [26]. Однако чаще всего размер популяции выбирается экспериментальным путем. Другим подходом к решению этой проблемы является введение адаптивного метода формирования размера популяции, который делает роевый алгоритм менее чувствительным к начальному выбору размера популяции.

Можно выделить несколько различных стратегий *создания начальной популяции*: стохастические методы, квазислучайные последовательности, метод оппозиционного обучения, многошаговый метод. Группу стохастических методов можно разделить на две подгруппы: генераторы псевдослучайных чисел и генераторы хаотических чисел. В роевых алгоритмах при отсутствии априорных знаний генерация случайных чисел является распространенным способом создания начальной популяции, которая может быть определена как

$$x_{i,j} \sim U(low_j; up_j), \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, D,$$

где U представляет собой равномерное распределение, low_j , up_j — нижнее и верхнее значения j -й переменной.

Равномерное распределение начальных решений по пространству поиска не является единственно возможным, наряду с равномерным используется распределение Леви, бета-распределение, нормальное распределение, логнормальное распределение и др. [1]. Основным достоинством метода на основе псевдослучайных чисел является простота, поскольку средства генерации псевдослучайных чисел присутствуют в каждом языке программирования, с их помощью легко решается проблема инициализации начальной популяции.

Для создания стохастических начальных популяций используются хаотические методы, для которых характерны эргодичность, случайность и регулярность [27]. Чтобы создать хаотическую популяцию, нужна хаотическая

карта. В очень общем виде одномерные хаотические карты работают следующим образом [27]:

$$x_{i,j}^{t+1} = f_m(x_{i,j}^t),$$

где $x_{i,j}^t$ — j -я переменная i -й частицы на итерации t , m — определяемый пользователем параметр. Обычно $x_{i,j}^0$ выбирается случайным образом, последующие точки создаются путем многократного применения хаотической карты. Наиболее часто применяются следующие хаотические карты: логистическое отображение (Logistic Map), синусоидальное отображение (Sinusoidal iterator), отображение tent-map, гауссово отображение (Gauss Map).

Авторы [28] считают что, хотя метод генерации случайных чисел очень прост с точки зрения программирования, он не очень эффективен, потому что сгенерированные компьютером числа не покрывают равномерно пространство поиска. Квазислучайные последовательности относятся к детерминированным методам, генерирующим одну и ту же популяцию. Детерминированные методы инициализации разработаны для обеспечения равномерного распределения частиц на всем пространстве поиска. В последнее время эти методы привлекают больше внимания, потому что в отсутствие предварительных знаний о проблеме, однородность начальной популяции может повысить исследовательскую способность алгоритма на его ранних стадиях [27]. В квазислучайной последовательности разнообразие популяции, как правило, лучше, чем при псевдослучайной генерации. Для формирования начальной популяции алгоритма роящихся частиц использовались квазислучайные последовательности Ван-дер-Корпута [28] и Фора [29], а для инициализации популяции алгоритма гравитационного поиска использовали генератор Соболя [30].

Идея инициализации на основе оппозиционного обучения основана на противоположных отношениях между сущностями и заключается в одновременном формировании как текущих решений, так и решений, противоположных текущим, используя для этого противоположные числа. В [31] дано определение противоположной точки в D -мерном пространстве. Пусть $\mathbf{z} = (z_1, \dots, z_D)$ — точка в пространстве и $z_j \in [a_j; b_j]$, $j = 1, 2, \dots, D$. Противоположность \mathbf{z} определяется как $\check{\mathbf{z}} = (\check{z}_1, \dots, \check{z}_D)$ как

$$(3) \quad \check{z}_j = a_j + b_j - z_j.$$

Одновременное использование случайного решения и его противоположности дает больше шансов найти оптимальное решение [31]. Можно предположить, если текущие случайно сгенерированные решения далеки от оптимального решения, то их противоположности будут ближе к этому решению. Приведенное выше определение противоположной точки называется противоположностью типа I, она определяется в соответствии с отношением между точками в пространстве поиска без учета их функциональных значений. Противоположность типа II определяется с учетом функционального пространства задачи следующим образом [31]. Предположим, что для функции $f(x_1, x_2, \dots, x_D)$ предопределены или могут быть оценены y_{\min} и y_{\max} . Пусть $y = f(x_1, x_2, \dots, x_D) \in \mathfrak{R}$ и $y \in [y_{\min}; y_{\max}]$. Для каждой точки

$\mathbf{x} = (x_1, x_2, \dots, x_D)$ противоположная точка типа II $\check{\mathbf{x}} = (\check{x}_1, \check{x}_2, \dots, \check{x}_D)$ определяется как

$$(4) \quad \check{x}^{\Pi} = \{x \mid \check{y} = y_{\min} + y_{\max} - y\}.$$

Авторы [31] теоретически доказали, что в общем случае противоположные числа с большей вероятностью будут ближе к оптимальному решению, чем чисто случайные; в [32] теоретические исследования были подтверждены экспериментально применительно к метаэвристике “дифференциальная эволюция”. Применением инициализации на основе оппозиционного обучения улучшены алгоритм роящихся частиц [33], алгоритм на основе стада криля [34], алгоритм на основе электромагнетизма [17]. На первом этапе методы оппозиционного обучения генерируют набор точек, называемых исходной популяцией. Исходная популяция может быть сгенерирована с использованием любого метода инициализации, например генератором равномерно распределенных чисел [32] или хаотической картой [35]. Затем используются некоторые простые эвристические правила (например выражения 3 или 4) для создания другой популяции такого же размера. Итоговая популяция формируется путем объединения обеих популяций на основе значений целевых функций входящих в них решений.

Примером многошагового метода является центроидальная тесселяция Вороного. С помощью этого метода можно разделить пространство поиска на равные объемы, не используя целевую функцию. Временная популяция создается с использованием генератора равномерно распределенных чисел или более сложных методов. Затем с помощью множества случайно сгенерированных вспомогательных точек пространство поиска разделяется на несколько областей. Эти области итеративно расширяются, пока не будут выполнены некоторые критерии. Центры областей используются в качестве начальной популяции [36]. Другим примером многошагового метода является подход к поиску перспективных регионов, названный Smart Sampling [37], который генерирует начальные решения, оценивает их, используя целевую функцию, и разделяет их на два класса — “хорошие” и “плохие” — на основе порога, применяемого к значению целевой функции каждого решения. На размеченной выборке начальных решений строится классификатор, который позволяет проверять, является ли вновь сгенерированное решение хорошим без использования целевой функции. Если новое решение классифицируется как “хорошее”, оно может быть оценено целевой функцией. Подход Smart Sampling эффективен только в том случае, если вычислительно затратным является процесс определения значения целевой функции.

В [1] предложен метод инициализации на основе латинского гиперкуба, суть которого заключается в разделении значений каждой координаты поиска на N равных интервалов, в каждый из которых попадает по одному начальному решению. Такой метод позволяет избежать проблемы чрезмерного скопления решений в одной точке пространства поиска. Исследования показывают, что такой метод может обеспечить лучшее покрытие, чем равномерное распределение [1].

Автор обзора [1] отмечает следующие проблемы рассмотренных методов инициализации: стохастические методы просты и легко реализуемы, однако

они плохо масштабируемы; хаотические методы сильно зависят от начальных условий; методы оппозиционного обучения генерируют популяцию удвоенного размера, что приводит к удваиванию вычислительных затрат; метод латинского гиперкуба очень эффективен при небольших размерностях пространства поиска, но он не дает явного преимущества при решении задач высокой размерности [1]. Как не существует единственного эффективного алгоритма оптимизации (“теорема о бесплатных завтраках” [7]), так не существует единственного эффективного алгоритма инициализации, в [1] на примерах показано, что для разных задач следует использовать разные алгоритмы.

5. Меры популяционного разнообразия

Эффективность роевого алгоритма во многом определяет баланс между фазами диверсификации и интенсификации. О том, в какой фазе находится алгоритм в каждый момент выполнения, можно судить по значению популяционного разнообразия. Разнообразие определяется как расстояние между решениями в пространстве поиска или как различие между значениями целевых функций найденных решений. Недостаточное популяционное разнообразие приводит к быстрой сходимости к потенциально неоптимальному решению. Для управления поиском решений необходимо иметь метрику для определения популяционного разнообразия. Метрики, основанные на расстоянии Минковского, наиболее часто используются для определения разнообразия. В [38] популяционное разнообразие определяется как

$$Div = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{j=1}^D (x_{ij}(t) - \bar{x}_j(t))^2},$$

где $\bar{x}_j(t)$ — среднее значение по j -му признаку на итерации t .

В [35, 39] разнообразие популяции алгоритма роящихся частиц и алгоритма искусственной пчелиной колонии рассчитывается по формуле

$$Div = \frac{1}{N} \frac{\sum_{i=1}^N \|\mathbf{x}_i - \bar{\mathbf{x}}\|}{\max_{1 \leq i, j \leq N} \|\mathbf{x}_i - \mathbf{x}_j\|}.$$

В [40] разнообразие популяции основано на норме $L1$:

$$Div = \frac{1}{N \cdot D} \sum_{i=1}^N \sum_{j=1}^D |x_{ij}(t) - \bar{x}_j(t)|.$$

По мнению авторов [41], медиана более эффективно указывает на центр j -го измерения, поэтому в [41] среднее значение $\bar{x}_j(t)$ заменено на медиану.

В [42] разнообразие определяется через Евклидово расстояние относительно лучшего решения в популяции

$$Div = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{j=1}^D (x_{ij}(t) - p_{jg}(t))^2}.$$

Разнообразие может быть определено как значение расстояния между лучшими решениями. В [43] это расстояние между всеми лучшими решениями, найденными каждой из частиц,

$$Div = \frac{2}{N(N-1)} \sum_{i=1}^{N-2} \sum_{j=i+1}^{N-1} |\mathbf{p}_i - \mathbf{p}_j|.$$

В [44] разнообразие определяется как значение расстояния между текущим и усредненным лучшим решением каждой частицы:

$$Div = \frac{1}{N} \sum_{i=1}^N |\mathbf{p}_i - \bar{\mathbf{p}}_i|.$$

В качестве показателя преждевременной сходимости в [45] рассматривается разность целевых функций лучших решений на двух итерациях:

$$\delta(t) = f(\mathbf{p}_g(t-1)) - f(\mathbf{p}_g(t)).$$

Констатируется преждевременная сходимость, если для заданного числа последовательных итераций значение $\delta(t)$ меньше заданного порога.

Мера разнообразия, основанная на относительном расстоянии частиц до глобально лучшей частицы, предложена в [45]:

$$Div = \max_{i=1, \dots, N} \frac{\|\mathbf{x}_i - \mathbf{p}_g\|}{\sqrt{\sum_{j=1}^D (x_{\max,j} - x_{\min,j})^2}}.$$

Популяционное разнообразие может быть определено на основе индекса квази-энтропии, который вычисляется по формулам [46]

$$QE(t) = - \sum_{i=1}^N P_i(t) \log P_i(t),$$

$$P_i(t) = \frac{f(\mathbf{p}_i(t))}{\sum_{i=1}^N f(\mathbf{p}_i(t))},$$

где $f(\mathbf{p}_i(t))$ — лучшее значение целевой функции i -й частицы на первых t итерациях. Авторы [46] предполагают, что на ранних стадиях оптимизации ни одна из частиц не находится в области глобального оптимального решения и индекс QE изменяется незначительно. На более поздних стадиях, когда все больше частиц попадают в область глобального оптимального решения, индекс QE резко уменьшается и, когда он достигнет определенной доли от начального значения, запускается локальный поиск.

Разнообразие в [44] определяется через распределение текущих скоростей частиц

$$Div = \frac{1}{N} \sum_{i=1}^N |\mathbf{v}_i - \bar{\mathbf{v}}_i|,$$

где $\mathbf{v}_i, \bar{\mathbf{v}}_i$ — текущее и среднее значения скорости i -й частицы соответственно.

Популяционное разнообразие может быть определено на основе значения целевой функции. В [1] популяционное разнообразие определяется по формуле

$$Div = \frac{\min(f_{best}, f_{avg})}{\max(f_{best}, f_{avg})},$$

где $f_{best}(t)$, $f_{worst}(t)$, $f_{avg}(t)$ — лучшее, худшее и среднее значения целевой функции.

В [16, 47] проведено исследование метрик разнообразия при поиске оптимума различных математических функций. Для исследования были выбраны метрики

$$\begin{aligned} \xi = Div &= \min \left\{ \left| \frac{f_{best} - f_{avg}}{f_{best}} \right|, 1 \right\}, \\ \nu = Div &= \min \left\{ \left| \frac{\sigma_f}{f_{avg}} \right|, 1 \right\}, \\ \psi = Div &= 1 - \left| \frac{f_{avg} - f_{best}}{f_{worst} - f_{best}} \right|, \\ \chi = Div &= 1 - \frac{|f_{best} - f_{avg}|}{\max |f_{best} - f_{avg}|_k}, \\ \varphi = Div &= \frac{\sigma_f}{|f_{worst} - f_{best}|}, \end{aligned} \tag{5}$$

где σ_f — стандартное отклонение от среднего значения целевой функции. Недостатками метрики ξ являются слабая устойчивость и сильная зависимость от области изменения целевой функции. Однако с ее помощью удается получить хороший результат для многомерных и сложных ландшафтов целевых функций, которые имеют ограниченный диапазон изменчивости и минимум у которых находится около нуля (например в инженерных задачах для минимизации ошибки). Метрика ν зависит от стандартного отклонения и, следовательно, от распределения значений целевой функции по всем элементам популяции. Эта метрика также как и ξ -метрика хорошо работает с целевыми функциями, имеющими ограниченный диапазон изменчивости, но она менее чувствительна к разбросам значений этих функций. В отличие от ξ - и ν -метрик, ψ не зависит от диапазона изменения значений целевой функции. Эта метрика очень чувствительна к небольшим изменениям и поэтому особенно подходит для ландшафтов целевых функций, содержащих плато и

области с небольшим градиентом. Метрика χ характеризует не только меру разнообразия, но и является оценкой лучшего результата по сравнению с другими решениями. Метрика φ может рассматриваться как комбинация ν - и ψ -метрик [16, 47].

Метрик разнообразия столько же, сколько проблем, т.е. метрики разнообразия специфичны для конкретной проблемы [48]. Не существует “наилучшей” метрики в целом, но есть “наиболее подходящая” метрика, зависящая не только от ландшафта целевой функции, но и от алгоритма оптимизации [16]. Автор [16] утверждает, что эффективная метрика разнообразия для эволюционной стратегии вряд ли будет эффективной для определения разнообразия для дифференциальной эволюции; это соображение автор [16] рассматривает как следствие теоремы об отсутствии бесплатных завтраков [7].

Разнообразие характеризует состояние популяции или распределение частиц. Большое значение Div означает, что поиск находится в состоянии диверсификации, малое значение свидетельствует о том, что поиск находится в состоянии интенсификации [40]. С помощью измерения разнообразия рассчитывается доля диверсификации и интенсификации на каждой итерации:

$$X_{dv} = \left(\frac{Div}{Div_{\max}} \right),$$

$$X_{it} = \left(\frac{|Div - Div_{\max}|}{Div_{\max}} \right).$$

Используя приведенные выше соотношения, можно поддерживать разнообразие популяции на достаточном уровне, не допуская ранней сходимости алгоритма. Механизмы поддержки популяционного разнообразия рассмотрены в разделе 6.

6. Механизмы поддержки популяционного разнообразия

Результаты поиска, выполняемого роевым алгоритмом, зависят от установленного или динамического баланса между двумя компонентами алгоритма: диверсификацией и интенсификацией. Успешно работающий компонент диверсификации обеспечит поиск на всем пространстве поиска, однако этот компонент может замедлить процесс сходимости алгоритма. Существуют несколько основных механизмов поддержания разнообразия: 1) ре-инициализация; 2) введение новых операторов: хаотических процедур и операторов кроссовера и мутации; 3) разделение популяции на субпопуляции.

6.1. Ре-инициализация

Ре-инициализация частиц является эффективным способом поддержания разнообразия, такой механизм способствует выходу частиц из локальных оптимумов и дает алгоритму возможность находить “достаточно хорошее” решение. Различают случайную ре-инициализацию, когда повторно генерируются большинство частиц, и элитную ре-инициализацию, когда сохраняются решения с лучшей целевой функцией [44].

В [49] новое решение генерируется на основе предыдущего решения путем добавления в него шума. По количеству повторно инициализированных решений определено три стратегии.

1. Количество ре-инициализированных решений уменьшается в процессе поиска. Более половины решений ре-инициализируются в начале поиска, а затем количество ре-инициализированных решений линейно уменьшается при каждой ре-инициализации. Эта стратегия заключается в том, чтобы сначала сосредоточиться на диверсификации, а в конце поиска — на интенсификации.

2. Часть решений повторно инициализируется после заданного числа итераций. Количество ре-инициализированных решений не меняется в процессе поиска.

3. Количество ре-инициализированных решений увеличивается в процессе поиска. Менее половины решений ре-инициализируются в начале поиска, и число ре-инициализированных решений линейно увеличивается при каждой ре-инициализации. Эта стратегия заключается в том, чтобы сначала сосредоточиться на интенсификации, а в конце поиска — на диверсификации.

В [50] ре-инициализация выполняется следующим образом: через заданное количество итераций в текущую популяцию из N решений добавляется еще N случайно сгенерированных решений; из полученных $2N$ решений выбираются N лучших; процесс повторяется заданное количество раз. В [51] предлагаются две стратегии ре-инициализации алгоритма роящихся частиц. Первая стратегия предполагает повторную инициализацию отдельных частиц, чье лучшее положение не улучшается после заданного числа итераций. Вторая стратегия подразумевает ре-инициализацию всей популяции, когда число лучших частиц в рое, имеющих значение целевой функции, близкое к значению целевой функции глобально лучшей частицы, превысит заданное пороговое значение. В [52] предлагается элитная схема ре-инициализации, когда случайным образом генерируются все решения за исключением лучшего, полученного на последней итерации, при условии, что после заданного числа итераций относительное улучшение значения целевой функции меньше порогового. В [45] активируется процедура ре-инициализации, если подтверждается состояние преждевременной сходимости или недостаточного популяционного разнообразия. В этом случае каждое решение заменяется на его противоположность, используя метод оппозиционного обучения.

6.2. Введение новых операторов

6.2.1. Эволюционные операторы

Операторы кроссовера и мутации, применяемые в эволюционных алгоритмах, все чаще применяются и в роевых алгоритмах для увеличения популяционного разнообразия [34, 53–56]. Оператор мутации, вносящий возмущение в решение $\mathbf{x} \in \mathbb{R}^D$, может быть определен как

$$\mathbf{x}' = \mathbf{x} + \mathbf{Z},$$

где \mathbf{Z} — вектор случайных чисел, распределенных по заданному закону. Примерами являются операторы мутации на основе распределения Гаусса

Таблица 1. Описание операторов мутации

Схема	Оператор мутации
De/Best/1	$\mathbf{x} = \mathbf{p}_g + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2})$
De/Best/2	$\mathbf{x} = \mathbf{p}_g + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2} + \mathbf{x}_{r3} - \mathbf{x}_{r4})$
De/CurrToBest/1	$\mathbf{x} = \mathbf{x}_i + K \cdot (\mathbf{p}_g - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2})$
De/CurrToRand/1	$\mathbf{x} = \mathbf{x}_i + K \cdot (\mathbf{x}_{r3} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2})$
De/Rand/1	$\mathbf{x} = \mathbf{x}_{r1} + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3})$
De/Rand/2	$\mathbf{x} = \mathbf{x}_{r5} + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2} + \mathbf{x}_{r3} - \mathbf{x}_{r4})$
De/RandToBest/1	$\mathbf{x} = \mathbf{x}_{r3} + K \cdot (\mathbf{p}_g - \mathbf{x}_{r3}) + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2})$

($Z \sim N(\mu, \sigma^2)$) и Коши ($Z \sim C(x_0, \gamma)$). Динамическая регулировка параметров σ^2 и γ в процессе выполнения роевого алгоритма позволяет изменять баланс между диверсификацией и интенсификацией. Большие значения указанных параметров могут привести к появлению сильно возмущенных решений, малые значения параметров способствуют выполнению локального поиска.

Оператор мутации может быть основан не только на законах распределения. Популяционное разнообразие алгоритма роящихся частиц и гравитационного поиска в [57, 58] регулируется оператором мутации, построенным на основе вейвлета Морле:

$$x_{ij}(t+1) = \begin{cases} x_{ij}(t+1) + (x_{\max,j} - x_{ij}(t+1)), & \sigma > 0, \\ x_{ij}(t+1) + (x_{ij}(t+1) - x_{\min,j}), & \sigma \leq 0, \end{cases}$$

$$\sigma = \frac{1}{\sqrt{a}} e^{\left(\frac{z}{2}\right)} \cos\left(5\frac{z}{a}\right),$$

где a — масштабный коэффициент, $z \in [-2,5a; 2,5a]$.

Авторы [59] вводят в алгоритм гравитационного поиска знаковую мутацию:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{s}_i \cdot \Delta_i(t+1),$$

$$s_{ij} = \begin{cases} -1, & rand < P_s, \\ 1, & rand \geq P_s, \end{cases}$$

где Δ_i — модифицирующий вектор, вычисленный по оригинальной схеме гравитационного поиска, \mathbf{s}_i — вектор мутации, P_s — вероятность мутации.

В роевых алгоритмах для увеличения популяционного разнообразия часто применяется оператор мутации, заимствованный из алгоритма дифференциальной эволюции [60–63]. В табл. 1 приведены операторы мутации [64], здесь \mathbf{p}_g — лучшее решение в текущей популяции; F и K — коэффициенты масштабирования, действительные положительные числа; индексы $r_1, r_2, r_3, r_4, r_5 = \overline{1, N}$ — случайно сгенерированные целые числа, причем $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$.

Оператор кроссовера на основе исходного списка решений генерирует новые решения путем “смешивания” информации о решениях из исходного списка. Оператор кроссовера $O(\mathbf{x}_i, \mathbf{x}_k, j)$ применяется для формирования нового решения путем изменения j -го элемента x_{ij} и x_{kj} в выбранных решениях. Новый элемент решения x_{lj} с использованием оператора арифметического кроссовера вычисляется по формуле

$$x_{lj} = \alpha x_{ij} + (1 - \alpha)x_{kj},$$

где α — коэффициент масштабирования, случайно выбранный из некоторого интервала. Элемент решения с применением геометрического кроссовера определяется по формуле

$$x_{lj} = (x_{ij} \cdot x_{kj})^\alpha, \quad \alpha \in [0, 1].$$

Случайный кроссовер формирует элемент решения по схеме

$$x_{lj} = \begin{cases} x_{ij}, & \text{rand} \leq 0,5, \\ x_{kj}, & \text{rand} > 0,5. \end{cases}$$

В [55] предложен оператор самоадаптирующегося кроссовера:

$$x_{lj}(t+1) = x_{ij}(t) \cdot (1 - Cr) + x_{kj}(t) \cdot Cr, \\ Cr = 0,8 + 0,2 \cdot \frac{f(\mathbf{x}_i(t)) - f(\mathbf{p}_g(t))}{f(\mathbf{p}_w(t)) - f(\mathbf{p}_g(t))},$$

где \mathbf{p}_w и \mathbf{p}_g — худшее и лучшее решения в популяции соответственно.

В [65] приведена классификация операторов кроссовера, применяемых в решении задач с непрерывно меняющимися параметрами, а также проведено эмпирическое исследование 14 операторов кроссовера при решении задач нахождения минимума математических функций. В [56] описан двухточечный полный кроссовер, в котором каждое из двух решений (оригинальное и его противоположность, полученная методом оппозиционного обучения) делится на три части двумя случайно выбранными точками. Новое решение формируется путем обмена частями исходных решений. Итоговые решения получаются после упорядочения по значению целевой функции шести новых решений и двух исходных.

Изменяя частоту (вероятность) применения операторов кроссовера и мутации в роевом алгоритме, можно регулировать баланс диверсификация–интенсификация.

6.2.2. Хаотические операторы

Существуют два способа включения хаоса в роевые алгоритмы оптимизации. Первый способ состоит в том, чтобы использовать последовательно, сгенерированные хаотическими системами, для замены случайных чисел, используемых в оригинальном алгоритме. Другой способ заключается

во включении хаотического поиска в качестве локального поиска в оригинальный алгоритм. Оба способа авторы [66] применяют в алгоритме гравитационного поиска для повышения скорости сходимости и решения проблемы выхода из локальных оптимумов. Двенадцать хаотических карт применяются для настройки значений коэффициентов инерции (ω_n, ω_f) в алгоритме стадо криля [54]. Хаотические последовательности используются в качестве генераторов псевдослучайных чисел в алгоритме искусственной пчелиной колонии [67], алгоритме роящихся частиц [68], алгоритме кузнечика [53].

6.3. Разделение популяции на субпопуляции

Многопопуляционные методы за последние десять лет стали одними из часто используемых методов оптимизации. Концепция этого метода следующая: исходная популяция делится на несколько субпопуляций; над элементами субпопуляции выполняются некоторые эволюционные операции; взаимодействуя между собой, субпопуляции обмениваются элементами, дробятся, поглощают друг друга, что способствует предотвращению преждевременной сходимости и поддержанию популяционного разнообразия путем распределения элементов по всему пространству поиска. Многопопуляционные методы сравнительно легко интегрируются в роевые алгоритмы оптимизации и часто работают эффективнее, чем алгоритмы с одной популяцией [69].

Многопопуляционные алгоритмы можно классифицировать на два типа: конкурирующие и кооперативные. Для алгоритмов конкурирующего типа каждая субпопуляция ищет решения в области, выделенной только для нее [70]. Кооперативные алгоритмы ведут поиск в одной и той же области, взаимодействуя и обмениваясь информацией [71].

В [69] обозначены проблемы и решения, характерные для многопопуляционных методов оптимизации. Первая проблема заключается в том, как определить количество субпопуляций: большое их количество связано с высокими вычислительными затратами, малое количество может не привести к желаемому эффекту. Указанная проблема решается двумя способами. Первый способ — использовать фиксированное количество субпопуляций. По этому способу функционирует большинство многопопуляционных методов. Его преимущество заключается в простоте реализации. Однако число субпопуляций определяется эмпирически, и трудно получить единые и эффективные правила для определения этого числа при решении различных практических задач. Второй способ заключается в динамическом изменении числа субпопуляций во время оптимизации. Выдвигается предположение, что на начальных этапах требуется большое количество субпопуляций для увеличения популяционного разнообразия и распределения возможных решений по всему пространству поиска. На поздних этапах небольшое количество субпопуляций способствует уменьшению разнообразия, что может быстрее приблизить решение к глобальному оптимуму.

Вторая проблема заключается в определении параметров коммуникации (миграции) между субпопуляциями. К таким параметрам относятся: 1) количество решений, подлежащих обмену; 2) коммуникационная (миграционная) политика, которая определяет, какие решения должны быть заменены реше-

ниями других субпопуляций; 3) частота коммуникаций; 4) топология коммуникаций. Чаще других применяется коммуникационная политика, основанная либо на жадном алгоритме, либо на случайном отборе, например элитарная политика миграции предполагает замену худших решений в субпопуляции лучшими решениями другой субпопуляции. Среди множества топологий можно выделить иерархическую и случайную топологии, а также линейные (цепочечные), кольцевые и решетчатые (клеточные) топологии, которые могут быть однонаправленными или двунаправленными.

Одной из широко применяемых многопопуляционных стратегий является модель островов [72], в которой популяция делится на несколько субпопуляций, называемых островами, эволюционирующие независимо друг от друга, но периодически взаимодействующие друг с другом посредством миграций. Схема взаимодействия, состоящая из последовательно происходящих процессов миграции и эволюции, позволяет создать баланс между диверсификацией и интенсификацией [18]. В [73] приводится динамическая островная модель, в которой размер популяции фиксирован, но размер каждой субпопуляции может меняться; топология задается полносвязным неориентированным графом. На каждой итерации алгоритма каждое решение мигрирует с исходного острова на остров назначения в соответствии с заданными вероятностями и динамически изменяющейся матрицей миграции, которая формируется с использованием алгоритма QLearning [74]; вместо единственной миграционной политики используется несколько миграционных политик.

В [18] предлагается иерархическая модель с 16 островами и четырьмя топологиями миграции. Острова нижних уровней ответственны за глобальный поиск, острова верхнего уровня выполняют локальный поиск. После каждой итерации лучшее решение мигрирует в соседнюю субпопуляцию верхнего уровня. Миграция между субпопуляциями одного уровня поддерживается путем замены лучшего решения текущей субпопуляции худшим решением соседнего острова.

Коммуникации могут быть синхронными или асинхронными. В первом случае все субпопуляции обновляют свои решения одновременно, тогда как во втором случае решения обновляются по наступлению какого-либо события. Обе модели имеют свои преимущества и недостатки. Синхронные островные модели проще в реализации, тогда как асинхронные модели являются более гибкими и эффективными [75]. Важным улучшением модели островов является слабосвязанная модель с виртуальным хранилищем совместно используемых данных, получившая название “океан” [76]. “Океан” позволяет легко реализовать гетерогенность и динамическое изменение числа островов, уменьшить задержку на обмен решениями за счет того, что “океан” хранит решения, которые были отправлены островом, но еще не приняты другим островом, — таким образом реализован асинхронный режим обмена решениями. В изложенном авторами [77] подходе на островах независимо работают четыре алгоритма: алгоритм пчелиной колонии для генерации правил нечеткого классификатора, алгоритм пчелиной колонии для оптимизации параметров, алгоритм адаптивной эволюционной стратегии и метод наименьших квадратов. “Океан” реализует обмен только множеством лучших решений, нахо-

дящихся в нем. Моменты отправки и приема лучших решений из “океана” выбирают сами алгоритмы, выполняющиеся на островах.

В клеточной топологии популяция делится на множество субпопуляций малого размера, обычно состоящих только из одного решения. Коммуникации происходят только в пределах небольшой окрестности, определенной вокруг отдельного решения или малой субпопуляции. При такой топологии снижается риск преждевременной сходимости по причине медленного распространения информации от соседа к соседу [78]. В качестве топологий окрестности в [79] рассматриваются три клеточные структуры — кубическая, тригональная и гексагональная, — интегрированные в алгоритм роящихся частиц. В [80] описан алгоритм роящихся частиц с клеточно-структурированной популяцией, частицы в которой распределены в двумерной сетке и взаимодействуют со своими соседями в соответствии с заданной окрестностью; авторы исследуют шесть типов окрестности: две линейные (L5, L9) и четыре компактных окрестности (C9, C25, C13, C21); для обработки краевых ячеек используется тороидальная структура.

Третья проблема многопопуляционных методов — как определить область поиска каждой субпопуляции. Одним из решений этой проблемы является кластеризация. Авторы [81] используют метод кластеризации для разделения популяции в алгоритме гравитационного поиска. При инициализации размер субпопуляции выбирается случайным образом из множества $T = \{5, 10, 25\}$. После каждого цикла работы алгоритма размер субпопуляции изменяется по результатам поиска. Если лучшее решение в субпопуляции не улучшается, из множества T случайным образом выбирается новое значение, в противном случае размер не меняется. В [82] предлагается многопопуляционный алгоритм роящихся частиц, который использует метод иерархической кластеризации, позволяющий адаптивно корректировать количество необходимых субпопуляций. Алгоритм кластеризации на основе нечетких s -средних в [83] используется для адаптивного разделения популяции на кластеры в алгоритме роящихся частиц. Для каждого кластера вычисляется его адаптивная ценность, полученная в результате конкурентного обучения. Кластеры ранжируются в соответствии с их ценностью. Кластеры с малой адаптивной ценностью поглощаются соседними кластерами, имеющими большую ценность.

Роевый алгоритм со структурированной популяцией имеет больше параметров, чем его аналог с единой популяцией. Дополнительно введенные параметры существенно влияют на работу алгоритма, поэтому настройка и регулировка этих параметров имеют важное значение для повышения эффективности алгоритма [75].

7. Настройка и регулировка значений параметров

Роевые алгоритмы характеризуются наличием стохастических компонентов и свободных параметров, которые могут быть установлены пользователем. Основная проблема здесь в том, что нет никакой гарантии, что хорошо настроенный алгоритм, хорошо работающий для одного типа проблем, может хорошо работать для другого типа проблем. Эффективность роевых алгоритмов во многом зависит от значений параметров, задаваемых в начале работы

и изменяемых в процессе выполнения алгоритма. По сути, это две проблемы: проблема автономной настройки и проблема регулировки значений параметров “на лету”. Процедура настройки параметров требует большого количества прогонов алгоритма для анализа его эффективности на одном или нескольких экземплярах проблем с различными настройками параметров. Процедура эта является вычислительно затратной, что является основным недостатком настройки параметров. Преимущество настройки параметров заключается в ее универсальности, хороший метод настройки применим для задания параметров многих различных метаэвристик. Недостатком регулировки параметров является ее узкая специализация, хорошо подобранные стратегии управления для одного алгоритма обычно не подходят для другого алгоритма [84].

7.1. Автономная настройка параметров

Автономная настройка параметров выполняется перед началом работы алгоритма. Цель состоит в том, чтобы априори найти лучшее сочетание значений параметров. Автономная настройка параметров является сложной проблемой из-за большого количества значений, которые может принимать каждый параметр. При этом необходимо учитывать, что параметры некоторых алгоритмов являются взаимозависимыми. Теоретический анализ, проведенный авторами [85], показал, что для обеспечения стабильной работы алгоритма роящихся частиц необходимо выполнение следующего соотношения между параметрами алгоритма:

$$(c_1 + c_2) < 24(1 - \omega^2) / (7 - 5\omega).$$

А в [86] условие сходимости алгоритма роящихся частиц определено как

$$1 > \omega > ((c_1 + c_2)/2 - 1) \geq 0.$$

На ранних этапах исследований метаэвристик настройка параметров осуществлялась либо “вручную” путем проведения серий экспериментов с различными конфигурациями параметров и выбора среди них лучшей на основе полученных результатов, либо “по аналогии” с применением настроек, показавших хорошие результаты при решении подобных проблем [87].

В последние два десятилетия проблеме автоматической настройки параметров стали уделять больше внимания [84, 87–91]. В [92] утверждается, что при разработке и создании нового метаэвристического алгоритма только 10 % времени тратится на разработку алгоритма, а 90 % времени уходит на тонкую настройку его параметров. Актуальность этой проблемы определяется следующим [89]:

1) использование методов автоматической настройки параметров может сократить затраты времени и позволит достичь лучших результатов по сравнению с ручными методами;

2) методы автоматической настройки могут помочь решить вопрос, превосходит ли один алгоритм другой, потому что он принципиально лучше или потому что его разработчики более успешно подобрали его параметры;

3) конечные пользователи часто не знают о влиянии настроек параметров алгоритма на его эффективность, поэтому просто используют настройки по умолчанию, однако конфигурация параметров по умолчанию может не работать должным образом для конкретных проблемных ситуаций, с которыми сталкивается пользователь;

4) автоматическая настройка параметров позволяет устранить необходимость пользователю устанавливать связь между настраиваемыми параметрами и решаемой проблемой.

Проблема настройки параметров может рассматриваться как проблема оптимизации, часто называемая мета-оптимизацией. Здесь алгоритм настройки является мета-оптимизатором, который ищет оптимальный (или по крайней мере хорошо работающий) набор параметров для целевого (настраиваемого) алгоритма. Сложность проблемы автоматической настройки обусловлена стохастической природой целевого алгоритма и метрики его эффективности [84, 90].

Конечной целью оптимизации является решение конкретных проблем, возникающих в реальном мире, но чтобы не решать каждую конкретную задачу отдельно и заново не начинать исследование с нуля, когда появляется новая задача, можно предположить, что некоторые из задач имеют общую структуру и могут рассматриваться как различные примеры общей проблемы. Этот шаг абстракции является фундаментальным для формального развития оптимизации и открывает путь к разработке алгоритмов и формальному анализу их свойств [88]. Если настройка алгоритма выполняется на одном примере проблемы, показателем эффективности может быть среднее значение показателя эффективности целевого алгоритма на нескольких прогонах. Показатель эффективности целевого алгоритма для множества примеров проблемы может быть определен как средняя эффективность алгоритма по данному множеству. Однако в некоторых случаях показатели эффективности могут сильно различаться, тогда показатели нормализуют для каждого примера [88]. Для нахождения оптимальной настройки параметров (или конфигурации) обычно требуется большое количество прогонов целевого алгоритма с разными конфигурациями на различных примерах. Таким образом, проблема настройки параметров является вычислительно дорогой и трудоемкой задачей [84].

Существующие методы автоматической настройки работают по принципу генерация–оценка, т.е. путем генерации различных конфигураций параметров и их оценки с помощью метрик эффективности [90, 91]. В качестве методов генерации часто используются методы планирования эксперимента [92–94], а также кригинг модели [95] и метод прямого поиска на адаптивных сетках [96]. Простым является метод повторной оценки параметров, который оценивает каждую конфигурацию несколько раз и возвращает среднее значение [90]. В методе грубой силы сначала генерируется семейство конфигураций параметров, как правило, с помощью методов планирования эксперимента, затем оценивается эффективность каждой конфигурации. В этом методе затраты на тестирование низкоэффективных конфигураций равны затратам на тестирование высокоэффективных конфигураций. Кроме того, здесь нет критерия, который определяет, сколько прогонов каждой конфигурации на

каждом экземпляре должно быть выполнено для обработки стохастичности целевого алгоритма [84]. Метод F-Race [88] поэтапно оценивает возможные конфигурации на примерах проблемы. После каждого этапа с использованием непараметрического теста Фридмана проверяется, есть ли статистические значимые отличия в показателях эффективности среди конфигураций-кандидатов. Если обнаружены значимые отличия, то после ранжирования кандидатов худшие отбрасываются и не участвуют в оценке на следующем этапе. Ранжирование в методе F-Race является еще и способом нормализации показателей эффективности, определяемых для разных примеров проблемы. В методе F-Race вычислительные ресурсы используются более эффективно, чем в методе грубой силы. Однако если целевой алгоритм имеет большое количество параметров и/или каждый параметр имеет большой диапазон изменений, тогда для получения хорошего результата необходимо оценить большое количество возможных конфигураций, что может стать практически невыполнимой задачей с точки зрения выполненных вычислений. К достоинствам методов F-Race и грубой силы относятся их понятность и простота реализации, с помощью этих методов можно настраивать как числовые, так и категориальные параметры [84].

Метод ParamILS [89] — это один из самых современных методов автоматической настройки параметров, с его помощью можно настраивать как числовые, так и категориальные параметры. В этом методе новая конфигурация параметров сравнивается с лучшей на текущий момент конфигурацией последовательности примеров проблемы. Если новая конфигурация хуже лучшей конфигурации, она исключается, иначе она оценивается на следующем примере и снова сравнивается с лучшей. Процесс продолжается до тех пор, пока новая конфигурация не будет оценена на всех примерах последовательности и будет определена новая лучшая конфигурация.

Перспективным подходом к решению проблемы настройки параметров является применение моделей поверхности отклика для описания зависимости эффективности целевого алгоритма от настроек его параметров. Процедура SPO [84] настраивает только числовые параметры и строит модель поверхности отклика, называемую моделью Кригинга. Метод SMAC работает как с числовыми, так и категориальными параметрами целевого алгоритма. Для построения поверхности отклика здесь используется случайный лес. Экспериментальное исследование показало, что с помощью SMAC удается получить лучшие результаты по сравнению с ParamILS. Метод SMAC особенно хорош при решении задач настройки со многими категориальными параметрами [84].

Подводя итог рассмотрения методов настройки параметров, можно утверждать, что эти методы выполняются до запуска настраиваемого алгоритма и выполняют поиск значений параметров, которые останутся фиксированными во время выполнения алгоритма. Кроме того, процессы настройки обычно включают несколько прогонов алгоритма, чтобы проанализировать его эффективность на разных значениях параметров с учетом его стохастической природы [97].

7.2. Регулировка значений параметров

Проблема регулировки значений параметров в роевых алгоритмах появилась одновременно с самими этими алгоритмами. В [98] отмечается, что изменение параметров в процессе выполнения алгоритма позволяет: искать решения, двигаясь в пространстве поиска большими шагами в начале работы алгоритма, и вести поиск оптимальных решений малыми шагами на последних этапах; алгоритму приспосабливаться к изменяющемуся ландшафту целевой функции; алгоритму накапливать информацию о целевой функции в процессе поиска и использовать эту информацию для повышения эффективности на поздних этапах и освобождает пользователя от необходимости выполнения этапа настройки значений параметров.

Методы регулировки значений параметров выполняются во время работы роевого алгоритма и изменяют значения его параметров во время выполнения алгоритма. Указанные методы могут быть разделены на две категории [99]: детерминированные и адаптивные. Детерминированные методы работают без обратной связи и следуют заранее установленному правилу при задании новых значений параметров. Адаптивные методы изменяют значения параметров на основе информации, получаемой после очередной итерации работы алгоритма.

Детерминированные методы регулировки значений параметров основаны исключительно на количестве пройденных итераций. В качестве примера рассмотрим методы регулировки коэффициента инерции алгоритма роящихся частиц, поскольку этот параметр в большей степени ответственен за соблюдение баланса между диверсификацией и интенсификацией. Несмотря на значительный объем исследований, многие стратегии изменения коэффициента инерции не изучены ни аналитически, ни эмпирически [100]. В табл. 2 приведены детерминированные методы регулировки коэффициента инерции; здесь

Таблица 2. Детерминированные методы изменения значения коэффициента инерции

Метод	Выражение	Ссылка
Случайное изменение	$\omega(t) = 0,5 + r(t)/2$	[101]
Линейное изменение	$\omega(t) = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \cdot (t/T)$	[102]
Нелинейное изменение	$\omega(t) = \omega_{\max} - (\omega_{\max} - \omega_{\min})(t/T)^{1/\pi^2}$	[103]
	$\omega(t) = \omega \cdot u^{-t}$	[104]
	$\omega(t) = (2/t)^{0,3}$	[105]
	$\omega(t) = \exp(-\exp((T-t)/T))$	[106]
	$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min})e^{-10t/T}$	[107]
	$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min})e^{-16t/T^2}$	
	$\omega(t) = \omega_{\max} + (\omega_{\min} - \omega_{\max}) \cdot \log_{10}(1 + 10t/T)$	[108]
Хаотическое изменение	$\omega(t) = 0,5z(t) + 0,5r(t)$	[109]
	$\omega(t) = z(t)\omega_{\min} + (\omega_{\max} - \omega_{\min}) \cdot (T-t)/T$	
Осциллирующее изменение	$\omega(t) = \begin{cases} ((\omega_{\min} + \omega_{\max})/2 + (\omega_{\max} - \omega_{\min})/2) \times \\ \times \cos(2\pi t(4k+6)/3T), & t < 3T/4, \\ \omega_{\min} & \text{иначе} \end{cases}$	[110]

Таблица 3. Адаптивные методы изменения значения коэффициента инерции

Метод	Выражение	Ссылка
Саморегуляция	$\omega(t) = \begin{cases} \omega(t-1) + \Delta\omega & \text{для лучшей частицы,} \\ \omega(t-1) - \Delta\omega & \text{для остальных частиц,} \end{cases}$ $\Delta\omega = \frac{\omega_{\max} - \omega_{\min}}{T}$	[112]
Адаптация на основе информации о скорости	$\omega(t) = \begin{cases} \max\{\omega(t-1) - \Delta\omega\}, & \bar{v}(t-1) \geq v_{ideal}(t), \\ \omega(t-1) - \Delta\omega, & \bar{v}(t-1) < v_{ideal}(t), \end{cases}$ $\bar{v}(t) = \frac{1}{N \cdot D} \sum_{i=1}^N \sum_{j=1}^D v_{ij} ,$ $v_{ideal}(t) = v_{\max} \left(\left(1 + \cos \left(\pi \frac{t}{T_{0,95}} \right) \right) / 2 \right)$	[113]
Адаптация на основе информации об успешности роя	$\omega(t) = 0,5r(t) + 0,5ssr(t-1)$	[114]
	$\omega(t) = (\omega_{\max} - \omega_{\min})(T-t)/T + \omega_{\min}ssr(t-1)$ $ssr(t) = \frac{\sum_{i=1}^N \begin{cases} 1, & f(\mathbf{p}_i(t)) < f(\mathbf{p}_i(t-1)), \\ 0, & f(\mathbf{p}_i(t)) \geq f(\mathbf{p}_i(t-1)) \end{cases}}{N}$	
Адаптация на основе факторов скорости изменения и степени агрегации	$\omega_i(t) = \omega_i(t-1) - (\omega_i(t-1) \exp(d(\mathbf{p}_g, \mathbf{p}_i) \cdot t/T))$ $\omega(t) = \exp(-\exp(-d(\mathbf{p}_g, \mathbf{p}_i) \cdot (T-t)/T))$	[106]
	$\omega_i(t) = \omega_{\max} - \alpha(1 - h_i(t)) + \beta \cdot s$ $h_i(t) = \frac{\min(f(\mathbf{p}_i(t-1)), f(\mathbf{p}_i(t)))}{\max(f(\mathbf{p}_i(t-1)), f(\mathbf{p}_i(t)))}$ $s = \frac{\min(f(\mathbf{p}_g(t)), \bar{f}(\mathbf{x}(t)))}{\max(f(\mathbf{p}_g(t)), f(\mathbf{x}(t)))}$ $\alpha, \beta \in [0, 1], \bar{f}(\bullet) - \text{среднее значение}$	
		[115]

Таблица 4. Стратегии регулировки значений параметров c_1 и c_2

Стратегии	Состояния	c_1	c_2
Стратегия 1	Диверсификация	Увеличение	Уменьшение
Стратегия 2	Интенсификация	Незначительное увеличение	Незначительное снижение
Стратегия 3	Сходимость	Незначительное увеличение	Незначительное увеличение
Стратегия 4	Прыжок	Уменьшение	Увеличение

приняты обозначения: t — текущая итерация; T — максимальное число итераций; $r(t) \in [0, 1]$ — равномерно распределенное случайное число; $\omega \in [0, 1]$, $u \in [1,0001; 1,005]$; $z(t)$ — значение хаотического логистического отображения (Logistic Map).

Авторами [111] предложен детерминированный способ регулировки значений коэффициентов c_1 и c_2 алгоритма роящихся частиц:

$$c_1(t) = c_{1i} + (c_{1f} - c_{1i}) \cdot t/T,$$

$$c_2(t) = c_{2i} + (c_{2f} - c_{2i}) \cdot t/T,$$

где c_{1i} и c_{2i} — начальные значения коэффициентов, c_{1f} и c_{2f} — конечные значения коэффициентов. В [111] рекомендовано использовать следующие начальные значения этих коэффициентов: $c_{1i} = c_{2f} = 2,5$, $c_{2i} = c_{1f} = 0,5$.

Адаптивные методы динамически регулируют параметры роевых алгоритмов в процессе их работы. В качестве средств адаптации используются такие характеристики, как средняя скорость, компактность роя, расстояние до лучшей частицы в рое. В табл. 3 приведены адаптивные методы регулировки коэффициента инерции.

Быстрая сходимость и уход из локальных оптимумов — две наиболее важные цели в исследовании роевых алгоритмов. Для достижения этих целей в [116] предложены четыре стратегии адаптивной регулировки коэффициентов ускорения алгоритма роящихся частиц (табл. 4).

Оценка состояния популяции основана на вычислении суммарного расстояния между частицами роя

$$f = (d_g - d_{\min}) / (d_{\max} - d_{\min}),$$

где d_{\min} , d_{\max} , d_g — минимальное, максимальное суммарное расстояние и суммарное расстояние до лучшей частицы в рое. Значение f относительно велико в состоянии диверсификации или прыжка и относительно мало в состоянии интенсификации или сходимости.

Коэффициент инерции ω уравнивает возможности глобального и локального поиска, в [116] значение ω адаптивно регулируется в соответствии с выражением:

$$\omega(f) = 1 / (1 + 1,5 \exp(-2,6f)).$$

Регулировка значений параметров c_1 и c_2 выполняется с помощью нечеткой системы типа сингтон, на вход которой подается значение f .

В последнее время увеличивается количество исследований, связанных с встраиванием хаоса в алгоритмы оптимизации. Использование хаоса усиливает диверсификационные и интенсификационные свойства алгоритмов оптимизации [53]. Стадии диверсификации и интенсификации в роевых алгоритмах регулируются хаотическим изменением их параметров: шага полетов Леви в алгоритме кукушкин поиск [117], гравитационных констант в гравитационном поиске [118], коэффициента c в алгоритме кузнечика [53], двух коэффициентов — γ и β — в алгоритме светлячков [119], четырех параметров в алгоритме летучих мышей [120].

Нечеткие системы достаточно часто используются для регулировки параметров роевых алгоритмов. В [121] адаптивная нечеткая схема построена на вычислении разности значений целевой функции лучших решений на двух последних итерациях алгоритма роящихся частиц (df). Предлагаются следующие нечеткие правила регулировки значений параметров c_1 и c_2 :

ЕСЛИ df маленькое, ТО c_1 большое, c_2 маленькое,
ЕСЛИ df среднее, ТО c_1 среднее, c_2 среднее,
ЕСЛИ df большое, ТО c_1 маленькое, c_2 большое.

Входная переменная df и выходные переменные c_1 и c_2 представлены тремя нечеткими термами: маленькое, среднее, большое.

В [122] для модификации параметров c_1 и c_2 построены три нечеткие системы типа Мамдани с разными входными переменными, в качестве которых используются нормированные значения: а) популяционного разнообразия, б) усредненной разности между значением целевой функций каждой частицы и значением целевой функции лучшей частицы, в) итерации. Все входные переменные представлены тремя равномерно распределенными на интервале $[0; 1]$ нечеткими термами с треугольными функциями принадлежности. Выходные переменные (c_1 и c_2) представлены пятью равномерно распределенными на интервале $[0,5; 2,5]$ нечеткими термами с треугольными функциями принадлежности. Входными переменными для первой нечеткой системы являются итерации и разнообразие, для второй — итерации и усредненная разность, для третьей — итерации, разнообразие и усредненная разность. Первые две системы имеют по девять нечетких правил, третья — 27.

В модифицированном алгоритме роящихся частиц FAIPSO [123] коэффициенты c_1 и c_2 корректируются с применением нечеткой системы, которая имеет шесть входных переменных, содержащих информацию о расстояниях между частицами, два выхода (c_1 и c_2) и десять нечетких правил. Компромисс между диверсификацией и интенсификацией при решении задач оптимизации с использованием гравитационного алгоритма достигается изменением значения ускорения, для этого авторы [124] применяют нечеткую систему типа 2. Системы этого же типа используются в [125] для регулировки параметров алгоритма летучих мышей, а в [126] — для динамической адаптации параметров алгоритма пчелиной колонии.

Размер популяции — параметр, присутствующий во всех роевых алгоритмах и влияющий на их эффективность. В [23] предложено адаптивное изменение размера популяции в зависимости от популяционного разнообразия;

максимальное количество итераций делится на равные периоды, каждый из которых содержит равное количество итераций; в конце каждого периода вычисляется разнообразие популяции и в зависимости от его значения размер популяции либо увеличивается, либо уменьшается.

Методы регулировки значений параметров включаются в исходный алгоритм, что приводит к увеличению вычислительной и временной сложности алгоритма. Одним из вариантов решения этой проблемы является разработка роевых алгоритмов без параметров, но этот подход требует детального изучения [6]. Существует еще ряд открытых вопросов, связанных с настройкой и регулировкой параметров. Необходимо теоретическое обоснование существования зависимости эффективности роевого алгоритма от его параметров. Необходимы разработки схем адаптации параметров для сложных задач оптимизации, таких как многоцелевая оптимизация, оптимизация с ограничениями, крупномасштабная оптимизация, оптимизация в динамических и неопределенных средах. Необходимы исследования применения методов машинного обучения для решения проблем настройки и регулировки параметров роевых алгоритмов, частично эта проблема рассматривается в разделе 8 обзора.

8. Гибридизация

Создатели метаэвристических алгоритмов сталкиваются с трудно разрешимой проблемой: как найти хороший баланс между диверсификацией и интенсификацией; в одном алгоритме практически невозможно сбалансировать указанные компоненты [127]. Задачи оптимизации могут быть более эффективно решены путем гибридизации существующих алгоритмических структур. Основная цель гибридизации заключается в использовании уникальных особенностей и преимуществ каждого алгоритма для достижения компромисса между диверсификацией и интенсификацией, а также для предотвращения преждевременной сходимости путем использования сильных сторон каждого из компонентов в соответствующем алгоритме. Однако создание эффективного гибридного алгоритма является сложным процессом, который требует обширных знаний и большого опыта в создании систем оптимизации [128].

В [129] приведена следующая классификация гибридных алгоритмов:

1) портфельные алгоритмы — это библиотека последовательно работающих алгоритмов поиска;

2) гиперэвристика — это библиотека алгоритмов поиска, снабженная механизмом координации, который выбирает и активизирует различные алгоритмы или генерирует специальные компоненты поиска;

3) меметические алгоритмы — это структура, которая содержит основной решатель и несколько алгоритмов локального поиска; 4) ансамбль алгоритмов — алгоритмическая структура, в которую входят стратегии поиска и общие компоненты решения проблем оптимизации.

Авторы [129] указывают на то, что классификация скорее отражает исторический путь гибридизации, чем выявляет алгоритмические различия в подходах.

Роевые алгоритмы показывают хорошие результаты на стадии диверсификации, но существенно уступают традиционным методам на стадии интенсификации; при приближении к оптимуму метаэвристические методы уступают градиентным. Комбинирование глобального поиска на уровне популяции и локального поиска на уровне отдельного решения получило развитие в области меметических алгоритмов [49]. Меметические алгоритмы — класс алгоритмов оптимизации, в структуру которых включены компоненты популяционных метаэвристик и процедуры локального поиска. Термин “меметический алгоритм” вытекает из того факта, что процедура локального поиска сродни мему, представляющему некоторую форму предметно-специфических априорных знаний человека-эксперта о том, как можно улучшить решение. В гибридных алгоритмах мемы воспринимаются как инструкции, правила, стратегии, априорные знания [130].

Авторы [131] разделяют меметические алгоритмы на три основных класса:

1) алгоритмы с одним фиксированным мемом или локальным оператором поиска, применяемым к решению, сформированному популяционной метаэвристикой;

2) алгоритмы с пулом заданных локальных операторов поиска (мемов), которые конкурируют между собой; выбор мема может быть основан на различных критериях, таких как абсолютное значение целевой функции решений, связанных с мемом, или улучшение значения целевой функции в результате прошлых применений мема;

3) эволюционирующие мемы с априори неизвестным пулом мемов, которые автоматически адаптируются к ландшафту целевой функции.

Процедура локального поиска, используемая в меметических алгоритмах, имеет как детерминированный, так и стохастический характер. В качестве детерминированных процедур часто используются алгоритмы Хука–Дживса [132] и Нелдера–Мида [133], а также градиентные и квазиньютоновские методы [134]. Стохастические процедуры в меметических алгоритмах представлены поиском с запретами [135], имитационным отжигом [6, 133], хаотическим локальным поиском [136]. Глобальный и локальный поиск в гибриде, предложенном в [127], выполняется двумя операторами, основанными на полетах Леви и поисковых выражениях алгоритма крилья соответственно.

В большинстве меметических алгоритмов локальный поиск применяется только к текущему глобально лучшему решению по причине того, что 1) окрестность такого решения может быть наиболее перспективной для поиска глобального оптимума; 2) экономится время выполнения по сравнению со схемами, которые применяют локальный поиск ко всем решениям.

В [137] предложены три схемы, комбинирующие алгоритм роящихся частиц с локальным поиском:

Схема 1. Локальный поиск применяется к глобально лучшему решению \mathbf{p}_g ;

Схема 2. Локальный поиск применяется к лучшему решению i -й частицы \mathbf{p}_i при условии, что случайно сгенерированное число $r < \varepsilon$, где $\varepsilon > 0$ — заданный порог;

Схема 3,а. Локальный поиск применяется как к глобально лучшему решению \mathbf{p}_g , так и к некоторым случайно выбранным решениям \mathbf{p}_i ;

Схема 3,б. Локальный поиск применяется как к глобально лучшему решению \mathbf{p}_g , так и к некоторым случайно выбранным решениям \mathbf{p}_i , для которых $|\mathbf{p}_g - \mathbf{p}_i| > c(S)$, где $A \in (0, 1)$ и (S) — размер поискового пространства S .

Приведенные три схемы могут применяться либо на каждой итерации алгоритма, либо на некоторых итерациях. Существуют и другие схемы, например применяющие локальный поиск ко всем частицам, однако такие схемы являются вычислительно затратными, на практике локальный поиск должен применяться только к небольшому количеству частиц (порядка 5 % от общего числа) [137].

В [138] предложен алгоритм роящихся частиц с двумя адаптивными методами локального поиска. Первый метод выполняет случайное перемещение от текущего решения, уменьшая на каждой итерации длину шага. Во втором методе процедуру локального поиска можно рассматривать как итеративное локальное движение частицы, когда поиск направляется лучшим решением i -й частицы без учета глобально лучшего решения:

$$\mathbf{x}_i(t + 1) = \mathbf{x}_i(t) + w \cdot \mathbf{v}_i(t) + c_1 \cdot \text{rand} \cdot (\mathbf{p}_i(t) - \mathbf{x}_i(t)).$$

В [139] предложен меметический алгоритм на основе алгоритма искусственной пчелиной колонии, методов Нелдера–Мида и случайного блуждания. Для поддержания баланса диверсификация/интенсификация используется адаптивная схема:

$$\begin{cases} \text{если } \text{rand}(0,1) > p(\psi), \text{ то использовать Метод Нелдера–Мида,} \\ \text{в противном случае использовать метод случайных блужданий.} \end{cases}$$

Значения $p(\psi)$ находятся из экспоненциального распределения

$$p(\psi) = e^{-\frac{(\psi - \mu_p)}{2\sigma_p^2}},$$

где ψ — значение меры разнообразия (5), μ_p , σ_p — среднее значение и стандартное отклонение меры разнообразия соответственно.

Оптимальный баланс между глобальным и локальным поиском явно зависит от решаемой задачи оптимизации, применяемого меметического алгоритма и настроек его параметров. Поэтому невозможно разработать универсальные принципы проектирования роевых алгоритмов, всегда приводящие к лучшим решениям [140]. Существование таких универсальных принципов противоречит теореме о бесплатных завтраках [7].

8.2. Ансамбли

Одним из подходов к повышению эффективности роевых алгоритмов является ансамблевая стратегия. В контексте проблемы оптимизации понятие “ансамбль” связано с использованием нескольких операторов и стратегий поиска, субпопуляций, алгоритмов, значений параметров для решения задачи

оптимизации. Ансамбли могут быть разделены на низкоуровневые и высокоуровневые. В ансамбль низкого уровня объединены различные типы операторов и стратегий поиска. Высокоуровневые ансамбли относятся к методам, которые для данной задачи адаптивно выбирают лучший алгоритм оптимизации из набора алгоритмов-кандидатов [141].

В [22] предлагается ансамбль стратегий адаптивного изменения коэффициента инерции в алгоритме роящихся частиц. Линейное или нелинейное уменьшение значения коэффициента инерции для каждой частицы определяется предысторией; если стратегия, выбранная на предыдущей итерации, приводит к лучшему решению, тогда назначенная стратегия сохраняется, в противном случае меняется. В [68] описан ансамбль из шести хаотических карт для генерации псевдослучайных чисел, используемых для формирования вектора скорости в алгоритме роящихся частиц. В начале работы алгоритма каждой частице случайным образом назначается одна из шести карт; если частица не улучшает свое положение в течение 20 итераций, ей назначается другая хаотическая карта. В [142] описан алгоритм роящихся частиц, в котором скорости частицы вычисляются с помощью одного из четырех операторов, ответственных за сходимость, интенсификацию, диверсификацию и “выпрыгивание” из локальных оптимумов. Выбор одного из четырех операторов основан на предложенной авторами адаптивной стратегии.

В модифицированном алгоритме искусственной пчелиной колонии MEABC используется ансамбль из трех стратегий поиска решений [143]:

1) стратегия, принятая в оригинальном алгоритме, когда новое решение генерируется путем перемещения старого решения к случайно выбранному из популяции решению;

2) стратегия ABC/best, повторяющая первую, в которую добавлена информация о лучшем решении в популяции;

3) модифицированная стратегия ABC/best/1, основанная на мутации DE/best/1 [64].

Указанные стратегии динамически меняются в процессе поиска, соревнуясь между собой при генерации нового решения, что позволяет сбалансировать процессы диверсификации и интенсификации. Эффективность MEABC проверялась экспериментально на множестве тестовых функций. По сравнению с оригинальным алгоритмом искусственной пчелиной колонии и двумя его модификациями MEABC показал более высокую скорость сходимости и более точные решения. Алгоритм MEABC оказался лучшим при сравнении с пятью модификациями алгоритма роящихся частиц, равным по эффективности третьей модификации при решении сложных мультимодальных задач и уступил этой модификации при поиске оптимума на нескольких математических функциях. Однако авторы утверждают, что MEABC проще и требует настройки меньшего количества параметров при одинаковой вычислительной временной сложности [143].

К высокоуровневым методам относится ансамбль алгоритмов роящихся частиц с самоадаптивным механизмом выбора лучшего алгоритма [144]. Ансамбль состоит из пяти разных алгоритмов роящихся частиц, каждый из которых используется для выполнения определенных задач, а именно: обес-

печения высокой скорости сходимости, для предотвращения преждевременной сходимости, для поддержания популяционного разнообразия, для увеличения возможностей локального поиска, для поиска нескольких локальных оптимумов при мультимодальной оптимизации. Выбор наиболее подходящего алгоритма на текущей итерации выполняется на основе механизма самоадаптации.

Хотя за последнее десятилетие были достигнуты впечатляющие результаты в разработке и исследовании ансамблевых стратегий в оптимизации, существует ряд открытых проблем, требующих решения, далее приведены некоторые из них [141].

1. В настоящее время количество элементов, включенных в ансамбль, относительно невелико, было бы полезно включить в ансамбль большее количество элементов с различными характеристиками. Однако пока неизвестно, как реализовать крупномасштабный ансамбль, насколько он будет надежен и эффективен.

2. Нерешенной проблемой остается определение характеристики алгоритмических компонентов, входящих в ансамбль. Необходимо разработать адекватные метрики описания компонентов, а также методы формирования ансамбля на основе этих метрик.

3. Эффективность стратегии ансамбля обычно проверяется с помощью эксперимента. Необходим теоретический анализ компонентов ансамбля и описание эффекта от каждого из них.

4. Наличие множества различных алгоритмических компонентов может вызвать нестабильность ансамбля по причине случайности, присущей этим компонентам. В реализацию ансамбля должны быть включены оценки изменения целевой функции и популяционного разнообразия, а также механизмы взаимодействия между различными компонентами.

8.3. Гиперэвристика

Эффективность метаэвристических алгоритмов зависит от решаемых задач. В общем случае неизвестно, какой алгоритм лучше всего подходит для данной задачи. Как следствие, выбор наиболее подходящей метаэвристики для конкретной задачи обычно является непростой задачей. Альтернативой экспертному поиску лучшего алгоритма для конкретной задачи оптимизации является гиперэвристика, которая обеспечивает универсальный эвристический подход к решению сложных задач оптимизации. Гиперэвристика ориентирована не на решение конкретной задачи, а на поиск (мета)эвристики или их последовательностей, которые будут решать конкретную задачу. Отличие метаэвристики от гиперэвристики в том, что первая работает в пространстве решений, а вторая в пространстве (мета)эвристик.

С точки зрения пространства поиска различают гиперэвристики, которые либо выбирают, либо генерируют (мета)эвристики низкого уровня, использующиеся в дальнейшем для решения конкретной задачи оптимизации. Гиперэвристику можно рассматривать как алгоритм обучения либо в режиме офлайн, либо в режиме онлайн. Онлайн обучение проводится в процессе решения проблемы, офлайн обучение заключается в формировании правил из

множества обучающих примеров. Среди подходов, основанных на онлайн-обучении, можно выделить подходы, основанные на обучении с подкреплением для выбора эвристики нижнего уровня, и подходы, которые применяют метаэвристику для поиска эвристики, работающей в пространстве решений [145].

Гиперэвристика на основе выбора строится следующим образом [146]:

1) выбрать множество эвристик низкого уровня для решения задач оптимизации, причем, не обязательно, чтобы каждая из них была высокоэффективной при решении поставленной задачи, но их число должно быть достаточно большим, чтобы генерировать большое пространство поиска;

2) определить алгоритм высокого уровня, который может быть обучающим автоматом или метаэвристикой (эволюционный алгоритм, алгоритм муравьиной колонии, роевый алгоритм);

3) определить процедуру кодирования для поиска низкоуровневых эвристик.

Подавляющее большинство гиперэвристик разработаны для дискретных пространств поиска. Далее приведены несколько гиперэвристик, работающих в непрерывных пространствах. Гиперэвристика DEWCO [147], используя на верхнем уровне алгоритм дифференциальной эволюции, формирует оптимальную начальную популяцию алгоритма китов. В выборе участвуют метод оппозиционного обучения и множество хаотических карт. Гиперэвристика H3AD [148] сочетает в себе два подхода: генерацию и выбор. В основе генерации лежит база знаний, которая содержит примеры алгоритмов, созданных для набора репрезентативных задач, вместе с описанием характерных признаков этих задач. Решение новой задачи начинается с выбора из базы знаний по представленным признакам наиболее похожей задачи и алгоритма, ранее сгенерированного для решения этой задачи. Гиперэвристика H3AD применяется для определения адекватных конфигураций параметров и компонентов алгоритма роящихся частиц. Для формирования компонентов алгоритма используется генетическое программирование, которое определяет такую процедуру, как, например, стратегия инициализации популяции. Выбор алгоритма роящихся частиц выполняется с использованием метода рассуждений по прецедентам.

В гиперэвристику HyperSPAM [129] включены три алгоритма решения задач оптимизации в непрерывном пространстве поиска. Алгоритм эволюционной стратегии с ковариационной матрицей адаптации активируется в начале процесса оптимизации в качестве компонента предварительной обработки. Далее полученное решение подается на два других алгоритма поиска. Первый выполняет перемещения по осям, а второй использует матричную ортогонализацию для выполнения диагональных перемещений. Четыре адаптивные стратегии используются для координации работы алгоритмов поиска. Первая стратегия является простым случайным выбором, а остальные три применяют механизм вознаграждения, основанный на успехах алгоритма поиска на предыдущих итерациях.

9. Открытые проблемы

Хотя за последние десятилетия исследование роевых алгоритмов продвинулось далеко вперед, остается еще ряд открытых проблем. Часть этих проблем затронута в обзоре, но в большинстве своем решаются эти проблемы на эмпирическом уровне. Далее рассмотрены некоторые актуальные проблемы в исследовании роевых алгоритмов [141, 149].

1. Эффективность роевого алгоритма напрямую связана с установлением оптимального баланса между диверсификацией и интенсификацией. Однако ни один алгоритм не может претендовать на решение проблемы достижения такого баланса. По сути, сам баланс является проблемой гипероптимизации, потому что это оптимизация алгоритма оптимизации.

2. Отсутствие единой математической структуры для всестороннего изучения роевых алгоритмов, позволяющей анализировать сложность, устойчивость, сходимость, соотношение точности решения и вычислительных затрат, настройку и регулировку параметров. Разработка такой структуры потребует междисциплинарного подхода для объединения различных математических, стохастических и численных методов.

3. Отсутствие унифицированной тестовой среды с комплексным набором метрик для сравнения результатов работы различных роевых алгоритмов. Показателем эффективности работы алгоритма чаще всего является точность. Однако если не учитывать время работы, то точность не будет адекватной оценкой работы алгоритма. Практика современного тестирования использует набор математических функций с различными свойствами, однако реальные задачи гораздо более разнообразны, они могут иметь много нелинейных ограничений, образованных несколькими изолированными областями. Таким образом, алгоритмы, хорошо работающие на тестовых функциях, могут быть неэффективными при решении реальных задач.

4. Масштабируемость роевых алгоритмов. Открытой остается проблема, как наилучшим образом изменить алгоритмы, хорошо работающие на задачах небольшой размерности (с числом переменных от нескольких единиц до нескольких сотен), для эффективного решения задач большой (больше тысячи переменных) и очень большой размерности.

5. Проблемой отдаленного будущего является создание интеллектуальных алгоритмов и систем, способных распознавать особенности решаемой задачи и выбирать или проектировать подходящий алгоритм.

10. Заключение

В статье рассмотрены некоторые подходы к повышению эффективности роевых алгоритмов оптимизации. Модульная структура роевых алгоритмов позволяет изменять входящие в них компоненты. Для задания популяции начальных решений используется метод инициализации или комбинация этих методов. Новое решение формируется как операторами оригинального алгоритма, так и эволюционными или хаотическими операторами. Лучшие решения получены путем гибридизации роевых алгоритмов с использованием различных стратегий. В разработке и исследовании роевых алгоритмов оп-

тимизации существует ряд нерешенных проблем, некоторые из них указаны в обзоре.

К настоящему времени в литературе представлено большое количество результатов исследований, невозможно охватить все соответствующие публикации в пределах заданного объема статьи. В обзоре не рассмотрены критерии остановки работы роевых алгоритмов [150], их квантовые модификации [151], а также аппаратная реализация роевых алгоритмов на основе облаков, вычислительных кластеров и графических процессоров [152–154].

СПИСОК ЛИТЕРАТУРЫ

1. *Li Q., Liu S.-Y., Yang X.-S.* Influence of Initialization on the Performance of Metaheuristic Optimizers // *Appl. Soft Comput.* 2020. V. 91. 106193.
2. *Luke S.* Essentials of Metaheuristics [Электронный ресурс]. Режим доступа: <https://cs.gmu.edu/~sean/book/metaheuristics/>
3. *Boussaid I., Lepagnot J., Siarry P.* A Survey on Optimization Metaheuristics // *Inf. Sci.* 2013. V. 237. P. 82–117.
4. *Kirkpatrick S., Gelatt C.D., Vecchi M.P.* Optimization by Simulated Annealing // *Science.* 1983. V. 220. P. 671–680.
5. *Glover F.* Future Paths for Integer Programming and Links to Artificial Intelligence // *Comput. Oper. Res.* 1986. V. 13. P. 533–549.
6. *Nematollahi A.F., Rahiminejad A., Vahidi B.* A Novel Meta-Heuristic Optimization Method Based on Golden Ratio in Nature // *Soft Comput.* 2020. V. 24. P. 1117–1151.
7. *Wolpert D., Macready W.* No Free Lunch Theorems for Optimization // *IEEE Trans. Evol. Comput.* 1997. V. 1. P. 67–82.
8. *Clerc M.* From Theory to Practice in Particle Swarm Optimization / *Handbook of Swarm Intelligence.* V. 8. Berlin: Springer, 2011. P. 3–36.
9. *Sorensen K.* Metaheuristics—The Metaphor Exposed // *Int. Trans. Oper. Res.* 2015. V. 22. No. 1. P. 3–18.
10. *Camacho-Villalón C.L., Dorigo M., Stützle T.* The Intelligent Water Drops Algorithm: Why It Cannot Be Considered a Novel Algorithm // *Swarm Intell.* 2019. V. 13. P. 173–192.
11. *Weyland D.* A Rigorous Analysis of the Harmony Search Algorithm: How The Research Community Can Be Misled by a “Novel” Methodology // *Int. J. Appl. Metaheuristic Computing.* 2010. V. 1. P. 50–60.
12. *Geem Z.W., Kim J.H., Loganathan G.V.* A New Heuristic Optimization Algorithm: Harmony Search // *Simulation.* 2001. V. 76. P. 60–68.
13. *Piotrowski A.P., Napiorkowski J.J., Rowinski P.M.* How Novel is the “Novel” Black Hole Optimization Approach? // *Inf. Sci.* 2014. V. 267. P. 191–200.
14. *Hatamlou A.* Black Hole: A New Heuristic Optimization Approach for Data Clustering // *Inf. Sci.* 2013. V. 222. P. 175–184.
15. *Shah-Hosseini H.* The Intelligent Water Drops Algorithm: A Nature-Inspired Swarm-Based Optimization Algorithm // *Int. J. Bio-Inspired Computation.* 2009. V. 1. No. 1/2. P. 71–79.
16. *Neri F.* Diversity Management in Memetic Algorithms // *Handbook of Memetic Algorithms.* 2012. SCI. V. 379. P. 153–165.

17. Cuevas E., Oliva D., Zaldivar D., Perez-Cisneros M., Pajares G. Opposition-Based Electromagnetism-Like for Global Optimization // Int. J. Innov. Comput. I. 2012. V. 8. No. 12. P. 8181–8198.
18. Turgut M.S., Turgut O.E., Eliyyi D.T. Island-based Crow Search Algorithm for Solving Optimal Control Problems // Appl. Soft Comput. 2020. V. 90. 106170.
19. Ходашинский И.А., Дудин П.А. Идентификация нечетких систем на основе прямого алгоритма муравьиной колонии // Искусственный интеллект и принятие решений. 2011. № 3. С. 26–33.
20. Kennedy J., Eberhart R. Particle Swarm Optimization // Proc. 1995 IEEE Int. Conf. Neural Networks. Perth: IEEE Service Center, 1995. P. 1942–1948.
21. Yang X.-S., Deb S. Engineering Optimisation by Cuckoo Search // Int. J. Math. Modelling and Numerical Optimisation. 2010. V. 1. No. 4. P. 330–343.
22. Shirazi M.Z., et al. Particle Swarm Optimization with Ensemble of Inertia Weight Strategies // ICSI 2017. LNCS. V. 10385. Heidelberg: Springer, 2017. P. 140–147.
23. Chen D.B., Zhao C.X. Particle Swarm Optimization with Adaptive Population Size and Its Application // Appl. Soft Comput. 2009. V. 9. No. 1. P. 39–48.
24. Chen L., Lu H., Li H., Wang G., Chen L. Dimension-by-Dimension Enhanced Cuckoo Search Algorithm for Global Optimization // Soft Comput. 2019. V. 23. P. 11297–11312.
25. Singh A., Deep K. Artificial Bee Colony Algorithm with Improved Search Mechanism // Soft Comput. 2019. V. 23. P. 12437–12460.
26. Gupta S., Deep K. Cauchy Grey Wolf Optimiser for Continuous Optimisation Problems // J. Exp. Theor. Artif. Intell. 2018. V. 30. P. 1051–1075.
27. Kazimipour B., Omidvar M.N., Li X., Qin A. A Novel Hybridization of Opposition-Based Learning and Cooperative Co-Evolutionary for Large-Scale Optimization // Proc. IEEE Congr. on Evolutionary Computation. Beijing, 2014. P. 2833–2840.
28. Pant M., Thangaraj R., Abraha A. Low Discrepancy Initialized Particle Swarm Optimization for Solving Constrained Optimization Problems // Fundamenta Informaticae. 2009. V. 95. P. 511–531.
29. Brits R., Engelbrecht A.P., van den Bergh F. A Nicheing Particle Swarm Optimizer // Proc. 4th Asia-Pacific Conf. Simulated Evol. Learning. 2002. P. 692–696.
30. Altinoz O.T., Yilmaz A.E., Weber G. Improvement of the Gravitational Search Algorithm by Means of Low-Discrepancy Sobol Quasi Random-Number Sequence Based Initialization // Adv. Electr. Comp. Eng. 2014. V. 14. No. 3. P. 55–62.
31. Mahdavi S., Rahnamayan S., Deb K. Opposition Based Learning: A Literature Review // Swarm Evol. Comput. 2018. V. 39. P. 1–23.
32. Rahnamayan S., Tizhoosh H.R., Salama M. Opposition-Based Differential Evolution // IEEE Trans. Evol. Comput. 2008. V. 12. No. 1. P. 64–79.
33. Farooq M.U., Ahmad A., Hameed A. Opposition-Based Initialization and a Modified Pattern for Inertia Weight (IW) in PSO // Proc. IEEE Int. Conf. Innov. Intel. Sys. Appl (INISTA). 2017. Number: 17083603.
34. Wang G.-G., Deb S., Gandomi A.H., Alavi A.H. Opposition-Based Krill Herd Algorithm with Cauchy Mutation and Position Clamping // Neurocomputing. 2016. V. 177. P. 147–157.
35. Gao W.-f., Liu S.-y., Huang L.-l. Particle Swarm Optimization with Chaotic Opposition-Based Population Initialization and Stochastic Search Technique // Commun. Nonlinear Sci. Numer. Simulat. 2012. V. 17. P. 4316–4327.

36. *Kazimipour B., Li X., Qin A.K.* A Review of Population Initialization Techniques for Evolutionary Algorithms // Proc. IEEE Congr. Evol. Comput. Beijing, 2014. P. 2585–2592.
37. *Melo V.V., Botazzo Delbem A.C.* Investigating Smart Sampling as a Population Initialization Method for Differential Evolution in Continuous Problems // Inf. Sci. 2012. V. 193. P. 36–53.
38. *Grobler J., Engelbrecht A.P.* A Scalability Analysis of Particle Swarm Optimization Roaming Behaviour // ICSI 2017. LNCS. V. 10385. Heidelberg: Springer, 2017. P. 119–130.
39. *Gao W.-f., Liu S.-y., Huang L.-l.* Enhancing Artificial Bee Colony Algorithm using More Information-Based Search Equations // Inf. Sci. 2014. V. 270. P. 112–133.
40. *Cheng S., Shi Y., Qin Q.* Experimental Study on Boundary Constraints Handling in Particle Swarm Optimization: From Population Diversity Perspective // Int. J. Swarm Intell. Res. 2011. V. 2. P. 43–69.
41. *Salleh M.N.M., et al.* Exploration and Exploitation Measurement in Swarm-Based Metaheuristic Algorithms: An Empirical Analysis / Recent Advances on Soft Computing and Data Mining. Springer, 2018. P. 24–32.
42. *Chaudhary R., Banati H.* Swarm Bat Algorithm with Improved Search (SBAIS) // Soft Comput. 2019. V. 23. P. 11461–11491.
43. *Blackwell T., Kennedy J.* Impact of Communication Topology in Particle Swarm Optimization // IEEE Trans. Evol. Comput. 2019. V. 23. No. 4. P. 689–702.
44. *Cheng S., Shi Y., Qin Q.* Promoting Diversity in Particle Swarm Optimization to Solve Multimodal Problem // Neural Inf. Proc. LNCS, V. 7063. Heidelberg: Springer, 2011. P. 228–237.
45. *Kaucic M.* A Multi-Start Opposition-Based Particle Swarm Optimization Algorithm with Adaptive Velocity for Bound Constrained Global Optimization // J. Glob. Optim. 2013. V. 55. P. 165–188.
46. *Cao Y., et al.* Comprehensive Learning Particle Swarm Optimization Algorithm With Local Search for Multimodal Functions // IEEE Trans. Evol. Comput. 2019. V. 23. No. 4. P. 718–731.
47. *Neri F., Tirronen V., Karkkainen T., Rossi T.* Fitness Diversity Based Adaptation in Multimeme Algorithms: A Comparative Study // IEEE Congr. Evol. Comput. Singapore: IEEE, 2007. INSPEC Accession Number: 9889441.
48. *Ghannami A., Li J., Hawbani A., Alhusaini N.* Diversity Metrics for Direct-Coded Variable-Length Chromosome Shortest Path Problem Evolutionary Algorithms // Computing. 2020.
49. *Cheng S., Shi Y., Qin Q., Zhang Q., Bai R.* Population Diversity Maintenance in Brain Storm Optimization Algorithm // J. Artif. Intell. Soft Comput. Res. 2014. V. 4. No. 2. P. 83–97.
50. *Simon D., Omran M.G., Clerc M.* Linearized Biogeography-Based Optimization with Re-Initialization and Local Search // Inf. Sci. 2014. V. 267. P. 140–157.
51. *Kennedy D.D., Zhang H., Rangaiah G.P., Bonilla-Petriciolet A.* Particle Swarm Optimization with Re-Initialization Strategies for Continuous Global Optimization / Global Optimization: Theory, Developments and Applications. Nova Science Publishers, 2013. P. 1–42.
52. *Merrikh-Bayat F.* The Runner-Root Algorithm: A metaheuristic for Solving Unimodal and Multimodal Optimization Problems Inspired by Runners and Roots of Plants in Nature // Appl. Soft Comput. 2015. V. 33. P. 292–303.

53. *Saxena A.* A Comprehensive Study of Chaos Embedded Bridging Mechanisms and Crossover Operators for Grasshopper Optimisation Algorithm // *Expert Syst. Appl.* 2019. V. 132. P. 166–188.
54. *Wang G.-G., et. al.* Chaotic krill herd algorithm // *Inf. Sci.* 2014. V. 274. P. 17–34.
55. *Wang G.-G., Deb S., Zhao X., Cui Z.* A New Monarch Butterfly Optimization with an Improved Crossover Operator // *Oper. Res. Int. J.* 2018. V. 18. P. 731–755.
56. *Zhou L., Ma M., Ding L., Tang W.* Centroid Opposition with a Two-Point Full Crossover for the Partially Attracted Firefly Algorithm // *Soft Comput.* 2019. V. 23. P. 12241–12254.
57. *San P.P., Ling S.H., Nguyen H.T.* Hybrid PSO-based Variable Translation Wavelet Neural Network and Its Application to Hypoglycemia Detection System // *Neural Comput. Applic.* 2013. V. 23. P. 2177–2184.
58. *Saha S.K., Kar R., Mandal D., Ghoshal S.P.* Optimal IIR Filter Design using Gravitational Search Algorithm with Wavelet Mutation // *J. King Saud Univ. Comput. Inf. Sci.* 2015. V. 27. P. 25–39.
59. *Nobahari H., Nikusokhan M., Siarry P.* A Multi-Objective Gravitational Search Algorithm Based on Non-Dominated Sorting // *Int. J. Swarm. Intell. Res.* 2012. V. 3. P. 32–49.
60. *Chen Y., et al.* Dynamic Multi-Swarm Differential Learning Particle Swarm Optimizer // *Swarm Evol. Comput.* 2018. V. 39. P. 209–221.
61. *Yazdani S., Hadavandi E.* LMBO-DE: A Linearized Monarch Butterfly Optimization Algorithm Improved with Differential Evolution // *Soft Comput.* 2019. V. 23. P. 8029–8043.
62. *Luo J., Liu Z.* Novel Grey Wolf Optimization Based on Modified Differential Evolution for Numerical Function Optimization // *Appl. Intell.* 2020. V. 50. P. 468–486.
63. *Zou F., et al.* Teaching-Learning-Based Optimization with Differential and Repulsion Learning for Global Optimization and Nonlinear Modeling // *Soft Comput.* 2018. V. 22. P. 7177–7205.
64. *Мех М.А., Ходашинский И.А.* Сравнительный анализ применения методов дифференциальной эволюции для оптимизации параметров нечетких классификаторов // *Изв. РАН. Теория и системы управления.* 2017. № 4. С. 65–75.
65. *Herrera F., Lozano M., Sanchez A.M.* A Taxonomy for the Crossover Operator for Real-Coded Genetic Algorithms: An Experimental Study // *Int. J. Intell. Syst.* 2003. V. 18. P. 309–338.
66. *Gao S., et al.* Gravitational Search Algorithm Combined with Chaos for Unconstrained Numerical Optimization // *Appl. Math. Comput.* 2014. V. 231. P. 48–62.
67. *Metlicka M., Davendra D.* Chaos Driven Discrete Artificial Bee Algorithm for Location and Assignment Optimisation Problems // *Swarm Evol. Comput.* 2015. V. 25. P. 15–28.
68. *Pluhacek M., Senkerik R., Davendra D.* Chaos Particle Swarm Optimization with Ensemble of Chaotic Systems // *Swarm Evol. Comput.* 2015. V. 25. P. 29–35.
69. *Ma H., et al.* Multi-Population Techniques in Nature Inspired Optimization Algorithms: A Comprehensive Survey // *Swarm Evol. Comput.* 2019. V. 44. P. 365–387.
70. *Blackwell T., Branke J.* Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments // *IEEE Trans. Evol. Comput.* 2006. V. 10. P. 459–472.
71. *Lung R.I., Dumitrescu D.* Evolutionary Swarm Cooperative Optimization in Dynamic Environments // *Nat. Comput.* 2010. V. 9. P. 83–94.

72. *Corcoran A.L., Wainwright R.L.* A Parallel Island Model Genetic Algorithm for the Multiprocessor Scheduling Problem // Proc. ACM Symp. Appl. Comput. ACM. 1994. P. 483–487.
73. *Lardeux F., Maturana J., Rodriguez-Tello E., Saubion F.* Migration Policies in Dynamic Island Models // Nat. Comput. 2019. V. 18. P. 163–179.
74. *Sutton R., Barto A.* Reinforcement Learning: An Introduction. London: MIT Press, 1998.
75. *Gong Y.J., et al.* Distributed Evolutionary Algorithms and Their Models: A Survey of the State-of-the-Art // Appl. Soft Comput. 2015. V. 34. P. 286–300.
76. *Raidl G.R.* A Unified View on Hybrid Metaheuristics // Proc. Hybrid Metaheuristics, Third Int. Workshop. LNCS V. 4030. Heidelberg: Springer, 2006. P. 1–12.
77. *Ходашинский И.А., Горбунов И.В.* Гибридный метод построения нечетких систем на основе модели островов // Информатика и системы управления. 2014. № 3. С. 114–120.
78. *Lynn N., Ali M.Z., Suganthan P.N.* Population Topologies for Particle Swarm Optimization and Differential Evolution // Swarm Evol. Comput. 2018. V. 39. P. 24–35.
79. *Shi Y., Liu H., Gao L., Zhang G.* Cellular Particle Swarm Optimization // Inf. Sci. 2011. V. 181. P. 4460–4493.
80. *Fang W., Sun J., Chen H., Wu X.* A Decentralized Quantum-Inspired Particle Swarm Optimization Algorithm with Cellular Structured Population // Inf. Sci. 2016. V. 330. P. 19–48.
81. *Huang L., Qin C.* A Novel Modified Gravitational Search Algorithm for the Real World Optimization Problem // Int. J. Mach. Learn. Cybern. 2019. V. 10. P. 2993–3002.
82. *Li C., Yang S.* A General Framework of Multipopulation Methods with Clustering in Undetectable Dynamic Environments // IEEE Trans. Evol. Comput. 2012. V. 16. P. 556–577.
83. *Xia L., Chu J., Geng Z.* A Multiswarm Competitive Particle Swarm Algorithm for Optimization Control of an Ethylene Cracking Furnace // Appl. Artif. Intell. 2014. V. 28. P. 30–46.
84. *Huang C., Li Y., Yao X.* A Survey of Automatic Parameter Tuning Methods for Metaheuristics // IEEE Trans. Evol. Comput. 2020. V. 24. No. 2. P. 201–216.
85. *Poli R.* Mean and Variance of the Sampling Distribution of Particle Swarm Optimizers during Stagnation // IEEE Trans. Evol. Comput. 2009. V. 13. No. 4. P. 712–721.
86. *Sengupta S., Basak S., Peters R.A.* Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives // Mach. Learn. Knowl. Extr. 2019. V. 1. P. 157–191.
87. *Calvet L., Juan A.A., Serrat C., Ries J.* A Statistical Learning Based Approach for Parameter Fine-Tuning of Metaheuristics // SORT – Stat. Oper. Res. Trans. 2016. V. 40. P. 201–224.
88. *Birattari M.* Tuning Metaheuristics: A Machine Learning Perspective // SCI. 2009. V. 197.
89. *Hutter F., Hoos H.H., Leyton-Brown K., Stützle T.* ParamILS: An Automatic Algorithm Configuration Framework // J. Artif. Intell. Res. 2009. V. 36. P. 267–306.
90. *Yuan Z., de Oca M.M.A., Birattari M., Stützle T.* Continuous Optimization Algorithms for Tuning Real and Integer Parameters of Swarm Intelligence Algorithms // Swarm Intell. 2012. V. 6. P. 49–75.

91. *Eiben A.E., Smit S.K.* Evolutionary Algorithm Parameters and Methods to Tune Them / Autonomous Search. Springer, 2012. P. 15–36.
92. *Adenso-Diaz B., Laguna M.* Fine-tuning of Algorithms using Fractional Experimental Designs and Local Search // Oper. Res. 2006. V. 54. P. 99–114.
93. *Barbosa E.B.M., Senne E.L.F.* Improving the Fine-Tuning of Metaheuristics: An Approach Combining Design of Experiments and Racing Algorithms // J. Optim. 2017. V. 2017. P. 1–7.
94. *Fallahi M., Amiri S., Yaghini M.* A Parameter Tuning Methodology for Metaheuristics Based on Design of Experiments // Int. J. Eng. Techn. Sci. 2014. V. 2. P. 497–521.
95. *Huang D., Allen T.T., Notz W.I., Zeng N.* Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models // J. Glob. Optim. 2006. V. 34. No. 3. P. 441–466.
96. *Audet C., Dennis J.E.* Mesh Adaptive Direct Search Algorithms for Constrained Optimization // SIAM J. Optim. 2006. V. 17. No. 1. P. 188–217.
97. *Montero E., Riff M.-C., Neveu B.* A Beginner’s Guide to Tuning Methods // Appl. Soft. Comput. 2014. V. 17. P. 39–51.
98. *Karafotias G., Hoogendoorn M., Eiben A.E.* Parameter Control in Evolutionary Algorithms: Trends and Challenges // IEEE Trans. Evol. Comput. 2015. V. 19. No. 2. P. 167–187.
99. *Zhang J., Chen W.-N., Zhan Z.-H., Yu W.-J., Li Y.-L., Chen N., Zhou Q.* A Survey on Algorithm Adaptation in Evolutionary Computation // Front. Electr. Electron. Eng. 2012. V. 7. No. 1. P. 16–31.
100. *Harrison K.R., Engelbrecht A.P., Ombuki-Berman B.M.* Inertia Weight Control Strategies for Particle Swarm Optimization // Swarm. Intell. 2016. V. 10. No. 4. P. 267–305.
101. *Eberhart R.C., Shi Y.* Tracking and Optimizing Dynamic Systems with Particle Swarms // Proc. IEEE Congr. Evol. Comput. V. 1. Seoul, South Korea, 2001. P. 94–100.
102. *Shi Y., Eberhart R.C.* Empirical Study of Particle Swarm Optimization // Proc. IEEE Congr. on Evol. Comput. 1999. V. 3. P. 1945–1950.
103. *Yang C., Gao W., Liu N., Song C.* Low-Discrepancy Sequence Initialized Particle Swarm Optimization Algorithm with High-Order Nonlinear Time-Varying Inertia Weight // Appl. Soft Comput. 2015. V. 29. P. 386–394.
104. *Jiao B., Lian Z., Gu X.* A Dynamic Inertia Weight Particle Swarm Optimization Algorithm // Chaos Solitons Fractals. 2008. V. 37. No. 3. P. 698–705.
105. *Fan S.K.S., Chiu Y.Y.* A Decreasing Inertia Weight Particle Swarm Optimizer // Eng. Optim. 2007. V. 39. No. 2. P. 203–228.
106. *Chauhan P., Deep K., Pant M.* Novel Inertia Weight Strategies for Particle Swarm Optimization // Memetic Comput. 2013. V. 5. No. 3. P. 229–251.
107. *Chen G., Huang X., Jia J., Min Z.* Natural Exponential Inertia Weight Strategy in Particle Swarm Optimization // Proc. Sixth World Congr. Intelligent Control and Automation. V. 1. IEEE. 2006. P. 3672–3675.
108. *Gao Y.-l., An X.-h., Liu J.-m.* A Particle Swarm Optimization Algorithm With Logarithm Decreasing Inertia Weight and Chaos Mutation // Int. Conf. Comput. Intell. Security. V. 1. IEEE, 2008. P. 61–65.
109. *Feng Y., Teng G.F., Wang A.X., Yao Y.M.* Chaotic Inertia Weight in Particle Swarm Optimization // Proc. Second Int. Conf. Innovative Computing, Information and Control. Kumamoto: IEEE, 2007. P. 475–479.

110. *Kentzoglanakis K., Poole M.* Particle Swarm Optimization with an Oscillating Inertia Weight // Proc. 11th Annual Conf. Genetic Evol. Comput. ACM, 2009. P. 1749–1750.
111. *Ratnaweera A., Halgamuge S.K., Watson H.C.* Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients // IEEE Trans. Evol. Comput. 2004. V. 8. P. 240–255.
112. *Tanweer M.R., Suresh S., Sundararajan N.* Self Regulating Particle Swarm Optimization Algorithm // Inf. Sci. 2015. V. 294. P. 182–202.
113. *Xu G.* An Adaptive Parameter Tuning of Particle Swarm Optimization Algorithm // Appl. Math. Comput. 2013. V. 219. P. 4560–4569.
114. *Adewumi A.O., Arasomwan A.M.* An Improved Particle Swarm Optimiser Based on Swarm Success Rate for Global Optimisation Problems // J. Exp. Theoret. Artif. Intell. 2014. V. 28. P. 441–483.
115. *Yang X., Yuan J., Mao H.* A Modified Particle Swarm Optimizer with Dynamic Adaptation // Appl. Math. Comput. 2007. V. 189. P. 1205–1213.
116. *Zhan Z.-H., Zhang J., Li Y., Chung H.S.-H.* Adaptive Particle Swarm Optimization // IEEE Trans. Syst. Man Cyber. B. 2009. V. 39. P. 1362–1381.
117. *Huang L., Ding S., Yu S., Wang J., Lu K.* Chaos-Enhanced Cuckoo Search Optimization Algorithms for Global Optimization // Appl. Math. Model. 2016. V. 40. P. 3860–3875.
118. *Mirjalili S., Gandomi A.H.* Chaotic Gravitational Constants for the Gravitational Search Algorithm // Appl. Soft Comput. 2017. V. 53. P. 407–419.
119. *Gandomi A., Yang X.-S., Talatahari S., Alavi A.* Firefly Algorithm with Chaos // Commun. Nonlinear Sci. Numer. Simulat. V. 18. 2013. P. 89–98.
120. *Gandomi A.H., Yang X.-S.* Chaotic Bat Algorithm // J. Comput. Sci. 2014. V. 5. P. 224–232.
121. *Juang Y.-T., Tung S.-L., Chiu H.-C.* Adaptive Fuzzy Particle Swarm Optimization for Global Optimization of Multimodal Functions // Inf. Sci. 2011. V. 181. P. 4539–4549.
122. *Melin P., et al.* Optimal Design of Fuzzy Classification Systems using PSO with Dynamic Parameter Adaptation through Fuzzy Logic // Expert Syst. Appl. 2013. V. 40. P. 3196–3206.
123. *Neshat M.* FAIPSO: Fuzzy Adaptive Informed Particle Swarm Optimization // Neural Comput. Applic. 2013. V. 23. P. 95–116.
124. *Gonzalez B., Melin P., Valdez F., Prado-Arechiga G.* A Gravitational Search Algorithm using Fuzzy Adaptation of Parameters for Optimization of Ensemble Neural Networks in Medical Imaging // Proc. Inter. Conf. on Artif. Intell. Las Vegas: CSREA Press, 2017. P. 54–59.
125. *Perez J., et al.* Interval Type-2 Fuzzy Logic for Dynamic Parameter Adaptation in the Bat Algorithm // Soft Comput. 2017. V. 21. P. 667–685.
126. *Amador-Angulo L., Castillo O.* Statistical Analysis of Type-1 and Interval Type-2 Fuzzy Logic in Dynamic Parameter Adaptation of the BCO // Proc. 9th Conf. Europ. Society for Fuzzy Logic and Technology. Atlantis Press, 2015. P. 776–783.
127. *Abdel-Basset M., Wang G.-G., Sangaiah A.K., Rushdy E.* Krill Herd Algorithm Based on Cuckoo Search for Solving Engineering Optimization Problems // Multimed. Tools Appl. 2019. V. 78. P. 3861–3884.
128. *Galvez J., Cuevas E., Becerra H., Avalos O.* A Hybrid Optimization Approach Based on Clustering and Chaotic Sequences // Int. J. Mach. Learn. Cybern. 2020. V. 11. P. 359–401.

129. *Caraffini F., Neri F., Epitropakis M.* HyperSPAM: A Study on Hyper-Heuristic Coordination Strategies in the Continuous Domain // *Inf. Sci.* 2019. V. 477. P. 186–202.
130. *Chen X., Ong Y.-S., Lim M.-H., Tan K.C.* A Multi-Facet Survey on Memetic Computation // *IEEE Trans. Evol. Comput.* 2011. V. 15. P. 591–607.
131. *Bartoccini U., Carpi A., Poggioni V., Santucci V.* Memes Evolution in a Memetic Variant of Particle Swarm Optimization // *Mathematics.* 2019. V. 7. 423.
132. *Duan Q., Liao T.W., Yi H.Z.* A Comparative Study of Different Local Search Application Strategies in Hybrid Metaheuristics // *Appl. Soft Comput.* 2013. V. 13. P. 1464–1477.
133. *Lopez-Garcia M., Garcia-Rodenas R., Gonzalez A.G.* Hybrid Meta-Heuristic Optimization Algorithms for Time-Domain-Constrained Data Clustering // *Appl. Soft Comput.* 2014. V. 23. P. 319–332.
134. *de Oca M.A.M., Cotta C., Neri F.* Local Search // *Handbook of Memetic Algorithms.* 2012. SCI. V. 379. P. 29–41.
135. *Lai X., Hao J.-K.* A Tabu Search Based Memetic Algorithm for the Max-Mean Dispersion Problem // *Comput. Oper. Res.* 2016. V. 72. P. 118–127.
136. *Yu Y., et al.* CBSO: A Memetic Brain Storm Optimization with Chaotic Local Search // *Memetic Comput.* 2018. V. 10. P. 353–367.
137. *Petalas Y.G., Parsopoulos K.E., Vrahatis M.N.* Memetic Particle Swarm Optimization // *Ann. Oper. Res.* 2007. V. 156. P. 99–127.
138. *Wang H., Moon I., Yang S., Wang D.* A Memetic Particle Swarm Optimization Algorithm for Multimodal Optimization Problems // *Inf. Sci.* 2012. V. 197. P. 38–52.
139. *Fister I., Fister I. Jr., Brest J., Zumer V.* Memetic Artificial Bee Colony Algorithm for Large-Scale Global Optimization // 2012 IEEE Congr. Evol. Comput. Brisbane: IEEE, 2012. P. 1–8.
140. *Sudholt D.* Parametrization and Balancing Local and Global Search // *Handbook of Memetic Algorithms.* SCI. 2012. V. 379. P. 55–72.
141. *Wu G., Mallipeddi R., Suganthan P.N.* Ensemble Strategies for Population-Based Optimization Algorithms—A Survey // *Swarm Evol. Comput.* 2019. V. 44. P. 695–711.
142. *Li C., Yang S., Nguyen T.T.* A Self-Learning Particle Swarm Optimizer for Global Optimization Problems // *IEEE Trans. Syst. Man Cybern. Part B.* 2012. V. 42. P. 627–646.
143. *Wang H., et al.* Multi-Strategy Ensemble Artificial Bee Colony Algorithm // *Inf. Sci.* 2014. V. 279. P. 587–603.
144. *Lynn N., Suganthan P.N.* Ensemble Particle Swarm Optimizer // *Appl. Soft Comput.* 2017. V. 55. P. 533–548.
145. *Pillay N., Qu R.* Hyper-Heuristics: Theory and Applications. Cham: Springer, 2018.
146. *Del Ser J., et al.* Bio-inspired Computation: Where We Stand and what's Next // *Swarm Evol. Comput.* 2019. V. 48. P. 220–250.
147. *Elaziz M.A., Mirjalili S.* A Hyper-Heuristic for Improving the Initial Population of Whale Optimization Algorithm // *Knowl. Based Syst.* 2019. V. 172. P. 42–63.
148. *Miranda P., Prudencio R., Pappa G.* H3ad: A Hybrid Hyper-Heuristic for Algorithm Design // *Inf. Sci.* 2017. V. 414. P. 340–354.
149. *Yang X.-S.* Nature-Inspired Optimization Algorithms: Challenges and Open Problems // *J. Comput. Sci.* 2020. 101104.
150. *Bassimir B., Schmitt M., Wanka R.* Self-Adaptive Potential-Based Stopping Criteria for Particle Swarm Optimization with Forced Moves // *Swarm. Intell.* 2020. V. 14. P. 285–311.

151. *Li P., Zhao Y.* A Quantum-Inspired Vortex Search Algorithm with Application to Function Optimization // *Nat. Comput.* 2019. V. 18. P. 647–674.
152. *Nedjah N., Mourelle L.M.* Hardware for Soft Computing and Soft Computing for Hardware // *SCI.* V. 529. Cham: Springer, 2014.
153. *Li D., et al.* A General Framework for Accelerating Swarm Intelligence Algorithms on FPGAs, GPUs and Multi-Core CPUs // *IEEE Access.* 2018. V. 6. P. 72327–72344.
154. *Damaj I., Elshafei M., El-Abd M., EminAydin M.* An Analytical Framework for High-Speed Hardware Particle Swarm Optimization // *Microprocess Microsyst.* 2020. V. 72. 102949.

Статья представлена к публикации членом редколлегии В.М. Вишневым.

Поступила в редакцию 25.06.2020

После доработки 31.10.2020

Принята к публикации 08.12.2020

© 2021 г. И.Р. БЕЛОВ (ivanb1993@mail.ru)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва)

**АНИЗОТРОПИЙНАЯ ФИЛЬТРАЦИЯ ДЛЯ ЛИНЕЙНЫХ
ДИСКРЕТНЫХ НЕСТАЦИОНАРНЫХ СИСТЕМ
С МУЛЬТИПЛИКАТИВНЫМИ ШУМАМИ
НА КОНЕЧНОМ ИНТЕРВАЛЕ ВРЕМЕНИ¹**

Рассматривается задача анизотропийной фильтрации на конечном интервале времени для линейной дискретной нестационарной системы с мультипликативными шумами. На вход системы поступает случайное возмущение с неточно известным распределением, неопределенность в описании которого характеризуется ограничением анизотропийного функционала. Оценка выхода исходной системы производится с помощью выбранного фильтра определенного вида по принципу минимизации критерия качества в виде анизотропийной нормы системы в ошибках фильтрации. На основании леммы об ограниченности анизотропийной нормы системы в ошибках фильтрации выводятся соотношения на неизвестные матрицы фильтра. В качестве частных случаев рассмотрены задачи анизотропийной фильтрации для системы с отказом в измерениях, а также аналогичная задача для системы с фильтром специального вида. Приведен численный пример использования полученного метода синтеза анизотропийного фильтра.

Ключевые слова: анизотропия, анизотропийная норма, задача фильтрации, системы с отказом в измерениях, мультипликативный шум, дискретная система.

DOI: 10.31857/S0005231021060027

1. Введение

Задача фильтрации состояния или выходного сигнала системы на основании доступных измерений является одной из фундаментальных задач теории управления. Технические системы чаще всего имеют достаточно сложную структуру, и поэтому точно измерить все их параметры не представляется возможным. Кроме того, те измерения параметров объекта, которые можно провести с помощью разнообразных технических средств, таких как датчики и сенсоры, чаще всего содержат в себе случайные или постоянные шумы, которые могут сильно влиять на конечное значение измеряемой величины и динамику системы в целом. Наличие подобных проблем и является предпосылкой появления целого класса задач обработки цифровых сигналов и фильтрации.

¹ Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 19-31-90060).

Проблема обработки данных, приходящих с разнообразных технических объектов, актуальна до сих пор, и наблюдается повышенный интерес к данному разделу теории управления. Универсального метода решения задачи фильтрации для любой технической системы и любых внешних условий не существует, поэтому при разработке новых методов используются определенные допущения о виде модели рассматриваемого объекта, фильтра, о свойствах шумов, входящих в измеряемый выходной сигнал систем или действующих на нее извне, и т.д.

При постановке задачи фильтрации для исследуемого объекта вводится некоторый критерий эффективности искомого фильтра, который чаще всего представляется в форме квадратичного функционала от ошибки фильтрации. Поэтому задача фильтрации часто сводится к задаче поиска фильтра, при котором функционал качества системы с данным фильтром достигает своего экстремального значения (оптимальная фильтрация), или к задаче поиска фильтра, обеспечивающего ограниченность значения заданного критерия некоторой величиной. Наиболее известными методами решения задач фильтрации являются методы \mathcal{H}_2 - и \mathcal{H}_∞ -теорий [1–3]. В самом простом случае LQG/ \mathcal{H}_2 -оптимальный фильтр, или фильтр Калмана, используется при допущении, что входное возмущение системы является гауссовским белым шумом. Также предполагается, что точно известны параметры модели исследуемого объекта, т.е. матрицы линейной системы объекта являются детерминированными. Подобный фильтр обеспечивает минимальный след ковариационной матрицы ошибки оценивания состояния системы и эффективен для систем с указанными допущениями. Однако фильтр Калмана оказывается неэффективным с точки зрения подавления возмущений, если указанные допущения не выполняются (к примеру, если входное возмущение не является белым шумом или параметры объекта не точно известны). Если статистические или иные свойства шумов точно не известны, то используется \mathcal{H}_∞ -фильтр. При его использовании предполагается, что входное возмущение является квадратично суммируемым или возмущением с ограниченной мощностью. При построении \mathcal{H}_∞ -фильтра математически конструируется наихудшее внешнее возмущение и решается стандартная задача фильтрации при данном возмущении, в виду чего в реальных ситуациях данный фильтр оказывается излишне консервативным, а в некоторых случаях практической реализации — неоправданно энергозатратным.

Поскольку \mathcal{H}_2 - и \mathcal{H}_∞ -фильтры являются наиболее эффективными только при выполнении условий задач, существует множество научных работ по поиску методов синтеза фильтров, более устойчивых к возмущениям со статистическими характеристиками, отличными от характеристик гауссовского белого шума, чем \mathcal{H}_2 -, и менее консервативных, чем \mathcal{H}_∞ . В 90-е годы с целью обобщения \mathcal{H}_2 - и \mathcal{H}_∞ -теории И.Г. Владимировым была предложена анизотропная теория управления и фильтрации в [4–7]. Фундаментальными понятиями этой теории являются анизотропия случайного вектора, средняя анизотропия (стационарной эргодической) последовательности случайных векторов и анизотропная норма системы. Понятие анизотропии случайного вектора было введено на основе таких понятий теории информации, как энтропия и количество информации, и представляет собой меру отличия

распределения случайного вектора от нормального распределения с нулевым математическим ожиданием и единичной ковариационной матрицей.

Для линейных нестационарных систем существует множество публикаций на тему анизотропийного анализа и задачи анизотропийной фильтрации. В [7] было показано, что для решения задачи оптимальной анизотропийной фильтрации в случае линейной дискретной нестационарной системы на конечном интервале времени необходимо решить разностное уравнение Риккати с дополнительным условием в виде алгебраического уравнения специального вида. В [8] приведена формулировка леммы об ограниченности анизотропийной нормы для линейных нестационарных систем. Эта лемма является основой для многих работ по анизотропийной фильтрации для различных классов нестационарных систем. Задача субоптимальной анизотропийной фильтрации для линейных нестационарных систем с неслучайными матрицами была решена в [9]. Также в [10] была решена задача субоптимальной фильтрации для линейных систем с условием равенства размерностей векторов оцениваемого выхода и внешнего возмущения. Задача субоптимальной фильтрации заключается в поиске фильтра, который обеспечивает ограниченность анизотропийной нормы сверху заданным числом. Это число можно менять для достижения требуемого уровня качества фильтрации. В отличие от задачи оптимальной фильтрации, в субоптимальной фильтрации условия существования искомого фильтра выражены в виде линейных матричных неравенств, решение которых проще с точки зрения программной реализации алгоритма решения задачи.

С середины двадцатого столетия появилось немало научных работ на тему задач управления и фильтрации для частного случая стохастических систем, а именно систем, содержащих мультипликативные шумы. Преимущество описания стохастических систем подобным образом заключается в том, что они достаточно эффективно аппроксимируют линейные дискретные системы со случайными матрицами и для них проще найти подход к решению задач анализа, управления и фильтрации. Также системы с мультипликативными шумами часто используются для математического описания разнообразных физических и финансовых процессов, технических объектов и т.д., как показано в [11–13]. Вследствие большого научного интереса к таким системам уже существует множество статей на темы задач робастного управления [14], оценивания состояния системы [15], линейного квадратичного управления [16] и многие другие.

В [17] представлена лемма об ограниченности анизотропийной нормы системы с мультипликативными шумами. Эта лемма является основой метода решения задачи анизотропийной фильтрации для систем с мультипликативными шумами, который представлен в данной статье. Помимо общего случая задачи анизотропийной фильтрации для линейных дискретных нестационарных систем с мультипликативными шумами, в статье рассмотрены частные случаи построения фильтров и решения исходных задач в рамках сделанных предположений.

2. Обозначения и предварительные сведения

Перед описанием класса рассматриваемых систем введем необходимые обозначения. Пусть \mathbb{R}^n — множество n -мерных векторов с элементами из \mathbb{R} ; $\mathbb{R}^{n \times m}$ — множество $(n \times m)$ -мерных матриц с элементами из \mathbb{R} ; \mathbb{L}_2^n — множество \mathbb{R}^n -значных интегрируемых с квадратом случайных векторов; $\mathbb{L}_2^{n \times m}$ — множество $\mathbb{R}^{n \times m}$ -значных интегрируемых с квадратом случайных матриц; $\mathbf{E}(x)$ — математическое ожидание случайной величины x . В выкладках статьи будут использоваться следующие матричные и векторные нормы:

- 1) $\|x\|_2 = \sqrt{\mathbf{E}(x^\top x)}$ — норма случайного вектора x из \mathbb{L}_2^n ;
- 2) $\|X\|_2 = \sqrt{\text{tr}(\mathbf{E}(X^\top X))}$ — 2-норма случайной матрицы X из $\mathbb{L}_2^{n \times m}$;
- 3) $\|X\|_\infty = \sqrt{\max_i \lambda_i(\mathbf{E}(X^\top X))}$ — ∞ -норма случайной матрицы X

из $\mathbb{L}_2^{n \times m}$, где $\lambda_i(\mathbf{E}(X^\top X))$ — собственные значения матрицы $\mathbf{E}(X^\top X)$.

- 4) $\Phi_M(X)$ — блочно-диагональная матрица из M матриц X .

Для сокращения записи линейных матричных неравенств (ЛМН) будет использоваться символ $*$ на месте элемента ЛМН, равного симметричному ему элементу относительно главной диагонали под знаком транспонирования.

В следующих подразделах приведены основные понятия анизотропийной теории, определение системы с мультипликативными шумами и лемма об ограниченности анизотропийной нормы для таких систем.

2.1. Класс рассматриваемых систем

Рассмотрим линейную дискретную нестационарную систему F на ограниченном интервале $[0, N]$ следующего вида:

$$(2.1) \quad F : \begin{cases} x(k+1) = A(k)x(k) + B(k)w(k), \\ z(k) = C(k)x(k) + D(k)w(k), \end{cases} \\ x(0) = 0, \quad k \in [0, N],$$

где $x(k) \in \mathbb{L}_2^{n_x}$ — вектор состояния системы, $w(k) \in \mathbb{L}_2^{m_w}$ — вектор входа системы, $z(k) \in \mathbb{L}_2^{p_z}$ — вектор выхода системы. Матрицы системы $A(k)$, $B(k)$, $C(k)$, $D(k)$ являются случайными матрицами специального вида

$$A(k) = \sum_{i=0}^M \xi_{1i}(k) A_i(k) \in \mathbb{L}_2^{n_x \times n_x}, \quad B(k) = \sum_{i=0}^M \xi_{2i}(k) B_i(k) \in \mathbb{L}_2^{n_x \times m_w}, \\ C(k) = \sum_{i=0}^M \xi_{3i}(k) C_i(k) \in \mathbb{L}_2^{p_z \times n_x}, \quad D(k) = \sum_{i=0}^M \xi_{4i}(k) D_i(k) \in \mathbb{L}_2^{p_z \times m_w},$$

где $\xi_{ji}(k)$, $j = \overline{1, 4}$, $i = \overline{1, M}$, $k = \overline{0, N}$ являются независимыми в совокупности по i, j, k скалярными случайными величинами с нулевыми математическими ожиданиями и единичными дисперсиями. Случайные величины $\xi_{j0}(k)$, $j = \overline{1, 4}$, $k = \overline{0, N}$ обладают единичными математическими ожиданиями и дисперсиями. Следовательно, выполняются равенства $\mathbf{E}A(k) = A_0(k)$, $\mathbf{E}B(k) = B_0(k)$, $\mathbf{E}C(k) = C_0(k)$, $\mathbf{E}D(k) = D_0(k)$.

Соотношение вход-выход системы (2.1) можно записать в следующей форме:

$$Z_{0:N} = F_{0:N}W_{0:N},$$

где $Z_{0:N} = (z^\top(0), \dots, z^\top(N))^\top$ — последовательность выходных сигналов системы, $W_{0:N} = (w^\top(0), \dots, w^\top(N))^\top$ — последовательность входных сигналов системы, $Z_{0:N} \in \mathbb{L}_2^{p_z(N+1)}$, $W_{0:N} \in \mathbb{L}_2^{m_w(N+1)}$. Для сокращения записи дальнейших выкладок введем обозначения $l_z = p_z(N+1)$, $l_w = m_w(N+1)$. Матрица связи в $Z_{0:N} = F_{0:N}W_{0:N}$ имеет вид $F_{0:N} = \text{block}_{0 \leq k, \kappa \leq N}(f(k, \kappa))$, где

$$(2.2) \quad f(k, \kappa) = \begin{cases} C(k)T(k, \kappa+1)B(\kappa), & \text{если } k > \kappa, \\ D(k), & \text{если } k = \kappa, \\ 0, & \text{если } k < \kappa. \end{cases}$$

Матрица перехода $T(k, \kappa)$ для $k > \kappa$ удовлетворяет рекуррентной формуле $T(k, \kappa) = A(k-1)T(k-1, \kappa) = T(k, \kappa+1)A(\kappa)$ при граничном условии $T(k, k) = I_{n_x}$. Стоит отметить, что случайная матрица $F_{0:N}$ и случайный вектор $W_{0:N}$ являются независимыми в силу независимости случайных величин $\xi_{ji}(k)$. Нормой системы (2.1) будем считать норму соответствующей матрицы $F_{0:N}$. Рассмотрим следующие виды нормы случайной матрицы $F_{0:N}$:

- \mathcal{H}_2 -норма системы F вида (2.1)

$$\|F\|_2 \triangleq \sqrt{\text{tr}(\mathbf{E}[F_{0:N}^\top F_{0:N}])} = \|F_{0:N}\|_2;$$

- \mathcal{H}_∞ -норма системы F вида (2.1).

Меру отклика системы T_{zw} на входной вектор $W_{0:N}$ можно описать с помощью среднеквадратичного коэффициента усиления, который определяется по формуле

$$(2.3) \quad \mathbf{Q}(F_{0:N}, W_{0:N}) = \frac{\|Z_{0:N}\|_2}{\|W_{0:N}\|_2}.$$

С учетом независимости $F_{0:N}$ и $W_{0:N}$ среднеквадратичный коэффициент усиления можно записать следующим образом:

$$(2.4) \quad \mathbf{Q}(F_{0:N}, W_{0:N}) = \sqrt{\frac{\text{tr}(\Lambda\Sigma)}{\text{tr}(\Sigma)}},$$

где $\Lambda = \mathbf{E}[F_{0:N}^\top F_{0:N}] \in \mathbb{R}^{l_w \times l_w}$, $\Sigma = \mathbf{E}[W_{0:N}W_{0:N}^\top] \in \mathbb{R}^{l_w \times l_w}$. Для системы F \mathcal{H}_∞ -норма определяется как супремум среднеквадратичного коэффициента усиления $\mathbf{Q}(F_{0:N}, W_{0:N})$ системы по всем последовательностям входных векторов $W_{0:N} \in \mathbb{L}_2^{l_w}$, т.е.

$$\|F\|_\infty \triangleq \sup_{W_{0:N} \in \mathbb{L}_2^{l_w}} \mathbf{Q}(F_{0:N}, W_{0:N}) = \|F_{0:N}\|_\infty.$$

Далее приведены понятия анизотропии случайного вектора и анизотропной нормы системы, с помощью которых можно отказаться от предположения о наилучшем возмущении и таким образом устранить консервативность \mathcal{H}_∞ -метода.

2.2. Анизотропия случайного вектора

Впервые определение анизотропии случайного вектора введено И.Г. Владимировым в [4]. Впоследствии, в [7] определение сформулировано следующим образом

Определение 1. Анизотропией $\mathbf{A}(W)$ m -мерного случайного вектора W называется неотрицательная величина, определенная по формуле

$$(2.5) \quad \mathbf{A}(W) = \min_{\lambda > 0} \mathbf{D}(f \| p_{m,\lambda}) = \frac{m}{2} \ln \left(\frac{2\pi e}{m} \|W\|^2 \right) - \mathbf{h}(W),$$

где $\mathbf{D}(f \| p_{m,\lambda})$ — относительная энтропия (расстояние Кульбака–Лейблера) функции плотности распределения вероятности f вектора W относительно функции плотности гауссовского распределения $p_{m,\lambda}$ с нулевым математическим ожиданием и скалярной ковариационной матрицей λI_m , $\lambda > 0$; $\mathbf{h}(W) = - \int_{\mathbb{R}^m} f(w) \ln f(w) dw$ — дифференциальная энтропия W .

2.3. Анизотропийная норма

На основании определений среднеквадратичного коэффициента усиления $\mathbf{Q}(F_{0:N}, W_{0:N})$ и понятия анизотропии случайного вектора $\mathbf{A}(W_{0:N})$ приведем определение анизотропийной нормы.

Определение 2. Анизотропийной нормой системы (2.1), которая определяется матрицей $F_{0:N}$ на временном интервале $k \in [0, N]$, называется неотрицательная величина следующего вида:

$$(2.6) \quad \|F\|_a = \sup_{W_{0:N} \in \mathbb{W}_a} \mathbf{Q}(F_{0:N}, W_{0:N}),$$

где $\mathbb{W}_a = \{W \in \mathbb{L}_2^{l_w} : \mathbf{A}(W) \leq a\}$ — множество l_w -мерных случайных векторов с анизотропией, ограниченной сверху числом $a \geq 0$.

В [7] показано, что супремум среднеквадратичного коэффициента усиления $\mathbf{Q}(F_{0:N}, W_{0:N})$ в (2.6) достигается при входном возмущении в виде случайного вектора, распределенного по нормальному закону с нулевым математическим ожиданием и ковариационной матрицей вида

$$\Sigma(q) = (I_{l_w} - q\Lambda)^{-1},$$

где q — скалярный параметр, являющийся решением уравнения

$$-\frac{1}{2} \ln \det \frac{m\Sigma(q)}{\text{tr}\Sigma(q)} = a$$

на интервале $q \in [0; \|F_{0:N}\|_{\infty}^{-2})$.

Анизотропийная норма является неубывающей функцией при любом $a \geq 0$, удовлетворяющей неравенствам

$$\|F\|_2 / \sqrt{l_w} = \lim_{a \rightarrow 0^+} \|F\|_a \leq \|F\|_a,$$

$$\|F\|_a \leq \lim_{a \rightarrow +\infty} \|F\|_a = \|F\|_{\infty}.$$

Эффективное использование математического аппарата анизотропийной теории имеет смысл при выполнении строгого неравенства $\|F\|_2/\sqrt{l_w} < \|F\|_\infty$.

В [17] показано, что для вычисления анизотропийной нормы системы с мультипликативными шумами необходимо решить разностное уравнение Риккати в обратном времени на интервале $k = 0 \dots N + 1$ и разностное уравнение Ляпунова в прямом времени. Для решения задачи анизотропийной фильтрации для систем с мультипликативными шумами, которое обеспечивает ограниченность анизотропийной нормы системы в ошибках фильтрации, запишем лемму об ограниченности анизотропийной нормы для данного класса систем.

Лемма 1. Дана линейная дискретная нестационарная система F с мультипликативными шумами вида (2.1) и задано максимальное значение анизотропии $a \geq 0$ входного сигнала. Тогда анизотропийная норма системы $\|F\|_a$ ограничена заданной константой $\gamma \geq 0$ тогда и только тогда, когда существуют значение параметра $q \geq 0$ и решения разностных уравнений Риккати $R_1(k) \succ 0$, $R_2(k) \succ 0$ такие, что

$$(2.7) \quad R_1(k) = \sum_{i=0}^M A_i^\top(k) R_1(k+1) A_i(k) + q \sum_{i=0}^M C_i^\top(k) C_i(k),$$

$$(2.8) \quad R_2(k) = A_0^\top(k) R_2(k+1) A_0(k) + L^\top(k) S^{-1}(k) L(k),$$

$$(2.9) \quad S(k) = \left(I_{m_z} - \sum_{i=0}^M (q D_i^\top(k) D_i(k) + B_i^\top(k) R_1(k+1) B_i(k)) - \right. \\ \left. - B_0^\top(k) R_2(k+1) B_0(k) \right)^{-1} \succ 0,$$

$$(2.10) \quad L(k) = S(k) \left(B_0^\top(k) R_1(k+1) A_0(k) + B_0^\top(k) R_2(k+1) A_0(k) + \right. \\ \left. + q D_0^\top(k) C_0(k) \right),$$

$$(2.11) \quad \sum_{k=0}^N \ln \det S(k) \geq 2a + l_w \ln(1 - q\gamma^2).$$

3. Постановка задачи фильтрации

Задана линейная дискретная нестационарная стохастическая система F на конечном временном интервале

$$(3.1) \quad F : \begin{cases} x(k+1) = A(k)x(k) + B(k)w(k), \\ z(k) = C_z(k)x(k) + D_z(k)w(k), \end{cases}$$

где $x(k)$ является вектором состояния системы, $w(k)$ — вектором входного возмущения, а $z(k)$ — вектором оцениваемого выхода системы. В системе (3.1) вход $w(k)$ является случайным процессом, удовлетворяющим усло-

вию $w(k) \in \mathbb{L}_2^{m_w}$, $\mathbf{E}[w(k)] = 0$. Предполагается, что известна априорная информация о других статистических характеристиках входной последовательности $W = w(k)$, на основе которой можно ввести ограничение анизотропии $A(W) \leq a$, где параметр a является известным положительным числом.

Целью работы является получение оценки выхода $z(k)$ на основе измеряемого выхода $y(k)$, описываемого следующим уравнением наблюдения:

$$(3.2) \quad y(k) = C_y(k)x(k) + D_y(k)w(k).$$

Ниже приведена формулировка задачи анизотропийной фильтрации

Задача 1. Получить такую оценку выхода $z(k)$ системы (3.1) на основе измеряемого выхода $y(k)$ вида (3.2), что анизотропийная норма $\|T_{\hat{z}w}\|_a$ системы в ошибках фильтрации $T_{\hat{z}w}$ будет ограничена сверху заданным числом $\gamma > 0$.

Выбрана следующая модель фильтра $T_{\hat{z}w}$:

$$(3.3) \quad T_{\hat{z}w} : \begin{cases} \hat{x}(k+1) = W(k)\hat{x}(k) + H(k)(y(k) - \hat{y}(k)), \\ \hat{z}(k) = F(k)\hat{x}(k) + G(k)(y(k) - \hat{y}(k)), \\ \hat{y}(k) = \Gamma(k)\hat{x}(k), \\ \hat{x}(0) = 0, \quad k = \overline{0, N}. \end{cases}$$

Матрицы фильтра (3.3) являются детерминированными матрицами, являющимися решением задачи фильтрации.

Обозначим ошибку оценивания выхода как $\tilde{z}(k) = z(k) - \hat{z}(k)$. Переписав систему (3.3) относительно ошибки оценивания $\tilde{z}(k)$ и ошибки оценивания состояния $e(k) = x(k) - \hat{x}(k)$, получим систему в ошибках фильтрации. Вывод уравнений этой системы подробно разобран в следующем разделе. Переформулируем задачу анизотропийной фильтрации следующим образом:

Задача 2. Для заданных системы F вида (3.1), уровня анизотропии $a \geq 0$ входного возмущения W и верхней границы анизотропийной нормы $\gamma \geq 0$ найти фильтр вида (3.3) такой, чтобы для системы в ошибках фильтрации выполнялось неравенство

$$(3.4) \quad \|T_{\hat{z}w}\|_a \leq \gamma.$$

4. Решение задачи фильтрации

В первую очередь, необходимо построить систему в ошибках фильтрации, соответствующую исходной системе объекта (3.1) и выбранной модели фильтра (3.3). Подставив выражение (3.2) измеряемого выхода $y(k)$ в модель (3.3), получим

$$(4.1) \quad \begin{cases} \hat{x}(k+1) = (W(k) - H(k)\Gamma(k))\hat{x}(k) + H(k)C_y(k)x(k) + \\ + H(k)D_y(k)w(k), \\ \hat{z}(k) = (F(k) - G(k)\Gamma(k))\hat{x}(k) + G(k)C_y(k)x(k) + \\ + G(k)D_y(k)w(k). \end{cases}$$

Для упрощения дальнейших выкладок вводим замену переменных

$$X(k) = W(k) - H(k)\Gamma(k), \quad Y(k) = F(k) - G(k)\Gamma(k).$$

Далее запишем выражение для ошибки оценивания выхода $z(k)$ системы (3.1)

$$(4.2) \quad \begin{aligned} \tilde{z}(k) &= (C_z(k) - G(k)C_y(k))x(k) + \\ &+ (D_z(k) - G(k)D_y(k))w(k) - Y(k)\hat{x}(k). \end{aligned}$$

Для записи системы в ошибках фильтрации введем обозначение

$$(4.3) \quad e(k) = x(k) - \hat{x}(k).$$

Из (3.1), (3.3) с учетом (4.3) получим динамику ошибки оценивания состояния:

$$(4.4) \quad \begin{aligned} e(k+1) &= (A(k) - X(k) - H(k)C_y(k))x(k) + \\ &+ X(k)e(k) + (B(k) - H(k)D_y(k))w(k). \end{aligned}$$

В итоге получим следующий вид системы в ошибках фильтрации $T_{\tilde{z}w}$:

$$(4.5) \quad T_{\tilde{z}w} : \begin{cases} e(k+1) = (A(k) - X(k) - H(k)C_y(k))x(k) + \\ \quad + X(k)e(k) + (B(k) - H(k)D_y(k))w(k), \\ \tilde{z}(k) = (C_z(k) - Y(k) - G(k)C_y(k))x(k) + \\ \quad + (D_z(k) - G(k)D_y(k))w(k) + Y(k)e(k). \end{cases}$$

Для сокращенной записи системы (4.5) введем расширенный вектор состояния

$$(4.6) \quad \bar{x}(k) = \begin{pmatrix} x(k) \\ e(k) \end{pmatrix}.$$

Запишем систему в ошибках фильтрации $T_{\tilde{z}w}$ вида (4.5) в терминах расширенного вектора состояния

$$(4.7) \quad T_{\tilde{z}w} : \begin{cases} \bar{x}(k+1) = \mathcal{A}(k)\bar{x}(k) + \mathcal{B}(k)w(k), \\ \tilde{z}(k) = \mathcal{C}(k)\bar{x}(k) + \mathcal{D}(k)w(k), \\ \bar{x}(0) = 0, \quad k = \overline{0, N}, \end{cases}$$

где матрицы $\mathcal{A}(k)$, $\mathcal{B}(k)$, $\mathcal{C}(k)$, $\mathcal{D}(k)$ определяются следующим образом:

$$(4.8) \quad \mathcal{A}(k) = \begin{pmatrix} A(k) & 0_{n_x \times n_x} \\ A(k) - X(k) - H(k)C_y(k) & X(k) \end{pmatrix},$$

$$(4.9) \quad \mathcal{B}(k) = \begin{pmatrix} B(k) \\ B(k) - H(k)D_y(k) \end{pmatrix},$$

$$(4.10) \quad \mathcal{C}(k) = \left(C_z(k) - Y(k) - G(k)C_y(k) \quad Y(k) \right),$$

$$(4.11) \quad \mathcal{D}(k) = D_z(k) - G(k)D_y(k).$$

Для решения задачи анизотропийной фильтрации в данной работе необходимо, чтобы система в ошибках фильтрации (4.7) была системой с мультипликативными шумами. Следовательно, должны быть выполнены следующие условия:

- 1) матрицы системы (4.7) $\mathcal{A}(k)$, $\mathcal{B}(k)$, $\mathcal{C}(k)$, $\mathcal{D}(k)$ являются независимыми в совокупности,
- 2) матрицы системы (3.1) $A(k)$, $B(k)$, $C_z(k)$, $D_z(k)$, $C_y(k)$, $D_y(k)$ являются независимыми в совокупности,
- 3) матрицы системы (4.7) $\mathcal{A}(k)$, $\mathcal{B}(k)$, $\mathcal{C}(k)$, $\mathcal{D}(k)$ удовлетворяют следующим выражениям:

$$(4.12) \quad \mathcal{A}(k) = \mathcal{A}_0(k) + \sum_{i=1}^M \zeta_{1i}(k)\mathcal{A}_i(k), \quad \mathcal{B}(k) = \mathcal{B}_0(k) + \sum_{i=1}^M \zeta_{2i}(k)\mathcal{B}_i(k),$$

$$(4.13) \quad \mathcal{C}(k) = \mathcal{C}_0(k) + \sum_{i=1}^M \zeta_{3i}(k)\mathcal{C}_i(k), \quad \mathcal{D}(k) = \mathcal{D}_0(k) + \sum_{i=1}^M \zeta_{4i}(k)\mathcal{D}_i(k).$$

Случайные величины $\zeta_{ij}(k)$, $i = \overline{1, 4}$, $j = \overline{1, M_i}$, $k = \overline{0, N}$, в выражениях (4.12)–(4.13) обладают следующими свойствами:

$$\mathbf{E}[\zeta_{ij}(k)] = 0, \quad \mathbf{E}[\zeta_{ij}^2(k)] = 1.$$

Заметим, что в матрицы системы $\mathcal{A}(k)$, $\mathcal{B}(k)$, $\mathcal{C}(k)$, $\mathcal{D}(k)$ входят матрицы $C_y(k)$ и $D_y(k)$, в общем случае являющиеся случайными. Поскольку матрицы $\mathcal{A}(k)$, $\mathcal{B}(k)$, $\mathcal{C}(k)$, $\mathcal{D}(k)$ должны быть независимыми в совокупности, необходимо выполнение следующего условия:

$$(4.14) \quad H(k)G^\top(k) = 0.$$

Это условие, к примеру, выполняется в следующих случаях:

- 1) $H(k) = 0$ или $G(k) = 0$;
- 2) $H(k) = \begin{pmatrix} H'(k) & 0 \end{pmatrix}$, $G(k) = \begin{pmatrix} 0 & G'(k) \end{pmatrix}$,
где $H'(k)$ — ненулевая часть неизвестной матрицы $H(k)$, $G'(k)$ — ненулевая часть неизвестной матрицы $G(k)$.

Частные случаи фильтра $H(k) = 0$ и $G(k) = 0$ будут рассмотрены в следующих разделах статьи. Пока предполагаем, что условие (4.14) выполняется при $H(k) \neq 0$ и $G(k) \neq 0$.

Стоит отметить, что матрица $\Gamma(k)$ влияет на динамику изменения переменной фильтра $\hat{x}(k)$ и на связь между оценкой выхода $\hat{z}(k)$ и переменной $\hat{x}(k)$. В зависимости от значения этой матрицы оценка выхода $\hat{z}(k)$ может корректироваться на основе значения измеряемого выхода $y(k)$ исходной системы или на основе “ошибки оценки” измеряемого выхода. Поскольку в общем случае матрица $C_y(k)$ может быть случайной, а все искомые матрицы

фильтра являются детерминированными, выражение $\hat{y}(k)$ является, скорее, прогнозом значения измеряемого выхода системы. Это выражение вводится дополнительно к основной модели фильтра. Значение матрицы $\Gamma(k)$ выбирается вручную вне процесса решения задачи. Частные случаи фильтра при различных значениях матрицы $\Gamma(k)$ будут рассмотрены в соответствующем разделе.

4.1. Условия ограниченности анизотропийной нормы системы в ошибках фильтрации

В данном разделе воспользуемся леммой 1 для системы (4.7) для сведения исходной задачи субоптимальной анизотропийной фильтрации к задаче выпуклой оптимизации с условиями в виде линейных матричных неравенств. Объединим уравнения (2.7)–(2.10) в одно уравнение Риккати. Для этого введем замену переменных $R(k) = R_1(k) + R_2(k)$:

$$\begin{aligned}
 & \sum_{i=0}^M A_i^\top(k) R_1(k+1) A_i(k) + \sum_{i=0}^M q C_i^\top(k) C_i(k) + \\
 & + A_0^\top(k) R_2(k+1) A_0(k) + L^\top(k) S^{-1}(k) L(k) = \\
 & = \sum_{i=0}^M A_i^\top(k) R(k+1) A_i(k) + \sum_{i=0}^M q C_i^\top(k) C_i(k) + \\
 & + A_0^\top(k) R(k+1) A_0(k) + L^\top(k) S^{-1}(k) L(k) - \\
 (4.15) \quad & - \sum_{i=0}^M A_i^\top(k) R_2(k+1) A_i(k) - A_0^\top(k) R_1(k+1) A_0(k),
 \end{aligned}$$

где матрицы $S(k)$ и $L(k)$ принимают следующий вид:

$$\begin{aligned}
 (4.16) \quad S(k) &= I - B_0^\top(k) R(k+1) B_0(k) - \sum_{i=0}^M q D_i^\top(k) D_i(k) - \\
 & - \sum_{i=0}^M \left(B_i^\top(k) (R(k+1) - R_2(k+1)) B_i(k) \right) + B_0^\top(k) R_1(k+1) B_0(k),
 \end{aligned}$$

$$(4.17) \quad L(k) = S(k) \left(B_0^\top(k) R(k+1) A_0(k) + q D_0^\top(k) C_0(k) \right).$$

В итоге получаем матричное уравнение

$$\begin{aligned}
 (4.18) \quad R(k) &= \sum_{i=0}^M A_i^\top(k) R(k+1) A_i(k) + \sum_{i=0}^M q C_i^\top(k) C_i(k) + \\
 & + A_0^\top(k) R(k+1) A_0(k) + L^\top(k) S^{-1}(k) L(k) + Q(k),
 \end{aligned}$$

$$\begin{aligned}
(4.19) \quad S(k) &= I - B_0^\top(k)R(k+1)B_0(k) - \sum_{i=0}^M qD_i^\top(k)D_i(k) - \\
&\quad - \sum_{i=0}^M \left(B_i^\top(k)(R(k+1) - R_2(k+1))B_i(k) \right) + \\
&\quad + B_0^\top(k)R_1(k+1)B_0(k)
\end{aligned}$$

$$(4.20) \quad L(k) = S(k) \left(B_0^\top(k)R(k+1)A_0(k) + qD_0^\top(k)C_0(k) \right),$$

где

$$Q(k) = - \sum_{i=0}^M A_i^\top(k)R_2(k+1)A_i(k) - A_0^\top(k)R_1(k+1)A_0(k).$$

Уравнение (4.18) не является разностным уравнением Риккати относительно новой переменной $R(k)$ по причине наличия членов с матрицами $R_1(k)$ и $R_2(k)$, которые содержатся в (4.18)–(4.20). Стоит отметить, что слагаемые $-Q(k)$, $\sum_{i=0}^M (B_i^\top(k)R_2(k+1)B_i(k))$ и $B_0^\top(k)R_1(k+1)B_0(k)$ являются неотрицательно определенными матрицами. Второе и третье слагаемые из перечисленных содержатся под знаком обратной матрицы, а в данном случае справедливо следующее утверждение

$$\text{Если } A \succ 0, X \succeq 0, \text{ то } (A + X)^{-1} \preceq A^{-1}.$$

Введем матрицу $\mathcal{R} \succ 0$, являющуюся решением матричного неравенства вида

$$\begin{aligned}
(4.21) \quad \mathcal{R}(k) &\succ \mathcal{A}_0^\top(k)\mathcal{R}(k+1)\mathcal{A}_0(k) + \sum_{i=1}^{M_1} \mathcal{A}_i^\top(k)\mathcal{R}(k+1)\mathcal{A}_i(k) + \\
&\quad + q \sum_{i=0}^{M_3} \mathcal{C}_i^\top(k)\mathcal{C}_i(k) + \mathcal{L}^\top(k)\mathcal{S}^{-1}(k)\mathcal{L}(k), \\
&\quad \mathcal{R}(N+1) = 0,
\end{aligned}$$

$$\begin{aligned}
(4.22) \quad \mathcal{S}(k) &= \left(I_{m_w} - \left(q \sum_{i=0}^{M_4} \mathcal{D}_i^\top(k)\mathcal{D}_i(k) + \right. \right. \\
&\quad \left. \left. + \sum_{i=0}^{M_2} \mathcal{B}_i^\top(k)\mathcal{R}(k+1)\mathcal{B}_i(k) \right) \right)^{-1} \succ 0,
\end{aligned}$$

$$(4.23) \quad \mathcal{L}(k) = \mathcal{S}(k) \left(\mathcal{B}_0^\top(k)\mathcal{R}(k+1)\mathcal{A}_0(k) + q\mathcal{D}_0^\top(k)\mathcal{C}_0(k) \right),$$

$$(4.24) \quad \sum_{k=0}^N \ln \det \mathcal{S}^{-1}(k) \geq 2a + l_w \ln(1 - q\gamma^2).$$

Разность правых частей неравенства (4.21) и уравнения (4.18) является неотрицательно определенной матрицей. Поэтому в силу свойства монотонности решения дискретного уравнения Риккати, описанного в [18, 19], решение $\mathcal{R}(k)$ неравенства (4.21) удовлетворяет условию $\mathcal{R}(k) \succ R(k)$ при всех $k \in [0, N]$. Таким образом, получена система матричных неравенств (4.21)–(4.23) с неравенством специального вида (4.24), которая в соответствии с изложенными преобразованиями позволяет получить субоптимальное решение поставленной задачи.

Запишем теорему об условиях существования фильтра вида (3.3), который обеспечивает ограниченность анизотропийной нормы системы вида (4.7).

Теорема 1. *Задаана линейная дискретная нестационарная система F на конечном интервале времени с реализацией в пространстве состояний (3.1) и выбраны вещественные числа $\gamma \geq 0$, $a \geq 0$. Фильтр $T_{\bar{z}w}$ с реализацией в пространстве состояний (3.3), гарантирующий выполнение неравенства $\|T_{\bar{z}w}\|_a \leq \gamma$ для системы в ошибках фильтрации с реализацией в пространстве состояний (4.7), существует, если найдутся матрицы $\mathcal{P}(k) \succ 0$, $\mathcal{P}(N+1) = 0$, $W(k)$, $H(k)$, $F(k)$, $G(k)$, $\Gamma(k)$ и $\bar{\Psi}(k)$, при которых выполняются неравенства*

$$(4.25) \quad \begin{pmatrix} \mathcal{P}(k) & * & * & * & * & * \\ 0 & \eta I_{m_w} & * & * & * & * \\ \mathcal{P}(k+1)\mathcal{A}_0(k) & \mathcal{P}(k+1)\mathcal{B}_0(k) & \mathcal{P}(k+1) & * & * & * \\ \mathcal{P}(k+1)\bar{\mathcal{A}}^\top(k) & 0 & 0 & \ddots & * & * \\ \mathcal{C}_0(k) & \mathcal{D}_0(k) & 0 & 0 & \mathcal{P}(k+1) & * \\ \bar{\mathcal{C}}^\top(k) & 0 & 0 & 0 & 0 & I \end{pmatrix} \succ 0,$$

$$(4.26) \quad \begin{pmatrix} \mathcal{P}(N) & * & * & * \\ 0 & \eta I_{m_w} & * & * \\ \mathcal{C}_0(N) & \mathcal{D}_0(N) & I_{p_z} & * \\ \bar{\mathcal{C}}^\top(N) & 0 & 0 & I \end{pmatrix} \succ 0,$$

$$(4.27) \quad \begin{pmatrix} \eta I_{m_w} - \bar{\Psi}(k) & * & * & * & * \\ \mathcal{P}(k+1)\mathcal{B}_0^\top(k) & \mathcal{P}(k+1) & * & * & * \\ \vdots & 0 & \ddots & * & * \\ \mathcal{P}(k+1)\mathcal{B}_M^\top(k) & 0 & 0 & \mathcal{P}(k+1) & * \\ \bar{\mathcal{D}}^\top(k) & 0 & \dots & 0 & I \end{pmatrix} \succ 0,$$

$$(4.28) \quad \begin{pmatrix} \eta I_{m_w} - \bar{\Psi}(N) & * \\ \mathcal{D}(N) & I_{p_z} \end{pmatrix} \succ 0,$$

$$(4.29) \quad \prod_{k=0}^N \det \bar{\Psi}(k) \geq e^{2a(\eta - \gamma^2)m_w(N+1)},$$

$$\mathcal{P}(k) \succ 0, \quad k = N - 1, \dots, 0.$$

Доказательство теоремы приведено в Приложении 1.

Стоит отметить, что в произведении матриц $\mathcal{A}_i \mathcal{P}(k+1)$, $\mathcal{B}_i^\top \mathcal{P}(k+1)$ могут содержаться нелинейные члены в виде произведений неизвестных матриц фильтра $W(k)$, $H(k)$, $F(k)$, $G(k)$, $\Gamma(k)$ и компонент матрицы $\mathcal{P}(k+1)$. В зависимости от частных случаев задачи фильтрации вводятся дополнительные замены переменных и к неравенствам (4.25)–(4.29) добавляются новые условия в соответствии с введенными заменами переменных. В частных случаях задачи анизотропийной фильтрации это будет показано наглядно.

4.2. Частные случаи задачи фильтрации

В данном разделе рассмотрены частные случаи задачи анизотропийной фильтрации для разных типов стохастических систем и разных конфигураций фильтра.

4.2.1. Система с отказами в измерениях

Рассмотрим систему F вида (3.1)

$$(4.30) \quad F : \begin{cases} x(k+1) = A(k)x(k) + B(k)w(k), \\ z(k) = C_z(k)x(k) + D_z(k)w(k), \\ y(k) = \lambda(k)C_y(k)x(k) + D_y(k)w(k), \end{cases}$$

где матрицы системы $A(k)$, $B(k)$, $C_z(k)$, $D_z(k)$, $C_y(k)$, $D_y(k)$ являются детерминированными матрицами соответствующих размерностей, $\lambda(k)$ — случайная величина с распределением вероятностей Бернулли:

$$P(\lambda(k) = 1) = p, \quad P(\lambda(k) = 0) = 1 - p.$$

Рассмотрим модель фильтра $T_{\hat{z}w}$ вида (3.3) со следующей конфигурацией:

$$(4.31) \quad T_{\hat{z}w} : \begin{cases} \hat{x}(k+1) = W(k)\hat{x}(k) + H(k)(y(k) - \hat{y}(k)), \\ \hat{z}(k) = C_z(k)\hat{x}(k), \\ \hat{y}(k) = pC_y(k)\hat{x}(k), \\ \hat{x}(0) = 0, \quad k = \overline{0, N}. \end{cases}$$

Для системы (4.30) и фильтра (4.31) рассмотрим задачу анизотропийной фильтрации. Необходимо найти матрицы $W(k)$, $H(k)$ фильтра, при которых анизотропийная норма системы в ошибках фильтрации удовлетворяет неравенству $\| \|T_{\hat{z}w}\| \|_a$ для фиксированного $a > 0$. В первую очередь, запишем систему в ошибках фильтрации $T_{\hat{z}w}$ по формуле (4.7)

$$(4.32) \quad T_{\hat{z}w} : \begin{cases} X(k+1) = \mathcal{A}(k)X(k) + \mathcal{B}(k)w(k), \\ \tilde{z}(k) = \mathcal{C}(k)X(k) + \mathcal{D}(k)w(k), \end{cases}$$

где

$$(4.33) \quad \begin{aligned} \mathcal{A}(k) &= \mathcal{A}_0(k) + (\lambda(k) - p)\mathcal{A}_1(k), \\ \mathcal{A}_0(k) &= \begin{pmatrix} A(k) & 0_{n_x \times n_x} \\ A(k) - W(k) & W(k) - pH(k)C_y(k) \end{pmatrix}, \\ \mathcal{A}_1(k) &= \begin{pmatrix} 0_{n_x \times n_x} & 0_{n_x \times n_x} \\ -H(k)C_y(k) & 0_{n_x \times n_x} \end{pmatrix}, \\ \mathcal{B}(k) &= \begin{pmatrix} B(k) \\ B(k) - H(k)D_y(k) \end{pmatrix}, \\ \mathcal{C}(k) &= (0_{p_z \times n_x} \quad C_z(k)), \quad \mathcal{D}(k) = D_z(k). \end{aligned}$$

Далее запишем сформулированные в предыдущем разделе условия ограниченности анизотропийной нормы в форме системы неравенств (4.25)–(4.29) для системы в ошибках фильтрации вида (4.32)

$$(4.34) \quad \begin{pmatrix} \mathcal{P}(k) & * & * & * & * \\ 0 & \eta I_{m_w} & * & * & * \\ \mathcal{P}(k+1)\mathcal{A}_0(k) & \mathcal{P}(k+1)\mathcal{B}_0(k) & \mathcal{P}(k+1) & * & * \\ \sqrt{p(1-p)}\mathcal{P}(k+1)\mathcal{A}_1^\top(k) & 0 & 0 & \mathcal{P}(k+1) & * \\ \mathcal{C}(k) & \mathcal{D}(k) & 0 & 0 & I \end{pmatrix} \succ 0,$$

$$(4.35) \quad \begin{pmatrix} \mathcal{P}(N) & * & * \\ 0 & \eta I_{m_w} & * \\ \mathcal{C}_0(N) & \mathcal{D}(N) & I_{p_z} \end{pmatrix} \succ 0,$$

$$(4.36) \quad \begin{pmatrix} \eta I_{m_w} - \bar{\Psi}(k) & * & * \\ \mathcal{P}(k+1)\mathcal{B}^\top(k) & \mathcal{P}(k+1) & * \\ \bar{\mathcal{D}}^\top(k) & 0 & I \end{pmatrix} \succ 0,$$

$$(4.37) \quad \begin{pmatrix} \eta I_{m_w} - \bar{\Psi}(N) & * \\ \mathcal{D}(N) & I_{p_z} \end{pmatrix} \succ 0,$$

$$(4.38) \quad \begin{aligned} \prod_{k=0}^N \det \bar{\Psi}(k) &\geq e^{2a}(\eta - \gamma^2)^{m_w(N+1)}, \\ \mathcal{P}(k) &\succ 0, \quad k = N-1, \dots, 0. \end{aligned}$$

Видно, что в матричных неравенствах (4.34), (4.36) есть произведения матриц $\mathcal{P}(k+1)\mathcal{A}_0(k)$, $\mathcal{P}(k+1)\mathcal{A}_1(k)$ и $\mathcal{P}(k+1)\mathcal{B}^\top(k)$. Матрицы $\mathcal{A}_0(k)$, $\mathcal{A}_1(k)$, $\mathcal{B}_0(k)$ удовлетворяют условиям (4.33). Следовательно, в $\mathcal{P}(k+1)\mathcal{A}_0(k)$, $\mathcal{P}(k+1)\mathcal{A}_1(k)$ и $\mathcal{P}(k+1)\mathcal{B}^\top(k)$ есть произведения неизвестных матриц

$\mathcal{P}(k+1)$, $W(k)$ и $H(k)$. С целью устранения нелинейных членов в матричных неравенствах введем следующие замены переменных системы матричных неравенств:

$$(4.39) \quad \mathcal{X}(k) = \begin{pmatrix} 0_{n_x \times n_x} & 0_{n_x \times p_y} \\ W(k) & H(k) \end{pmatrix},$$

$$(4.40) \quad \mathcal{P}(k+1)\mathcal{A}_0(k) = \mathcal{P}(k+1)\tilde{\mathcal{A}}_0^0(k) + \mathcal{P}(k+1)\mathcal{X}(k)\tilde{\mathcal{A}}_0^1(k),$$

$$\tilde{\mathcal{A}}_0^0(k) = \begin{pmatrix} A & 0_{n_x \times n_x} \\ A & 0_{n_x \times n_x} \end{pmatrix}, \quad \tilde{\mathcal{A}}_0^1(k) = \begin{pmatrix} -I_{n_x} & I_{n_x} \\ 0_{n_x \times n_x} & -pC_y \end{pmatrix},$$

$$\mathcal{P}(k+1)\mathcal{A}_1(k) = \mathcal{P}(k+1)\mathcal{X}(k)\tilde{\mathcal{A}}_1^1(k),$$

$$\tilde{\mathcal{A}}_1^1(k) = \begin{pmatrix} 0_{n_x \times n_x} & 0_{n_x \times n_x} \\ -C_y & 0_{p_y \times n_x} \end{pmatrix},$$

$$\mathcal{P}(k+1)\mathcal{B}(k) = \mathcal{P}(k+1)\tilde{\mathcal{B}}^0(k) + \mathcal{P}(k+1)\mathcal{X}(k)\tilde{\mathcal{B}}^1(k),$$

$$\tilde{\mathcal{B}}^0(k) = \begin{pmatrix} B \\ B \end{pmatrix}, \quad \tilde{\mathcal{B}}^1(k) = \begin{pmatrix} 0_{n_x \times m_w} \\ -D_y \end{pmatrix},$$

$$(4.41) \quad Q(k) = \mathcal{P}(k+1)\mathcal{X}(k).$$

Применим замену переменных (4.39)–(4.41) к неравенствам (4.34)–(4.37) и получим выражения следующего вида:

$$(4.42) \quad \begin{pmatrix} \mathcal{P}(k) & * & * & * & * \\ 0 & \eta I_{m_w} & * & * & * \\ \mathcal{P}(k+1)\tilde{\mathcal{A}}_0^0(k) + Q(k)\tilde{\mathcal{A}}_0^1(k) & \mathcal{P}(k+1)\tilde{\mathcal{B}}^0(k) + Q(k)\tilde{\mathcal{B}}^1(k) & \mathcal{P}(k+1) & * & * \\ \sqrt{p(1-p)}Q(k)\tilde{\mathcal{A}}_1^1(k) & 0 & 0 & \mathcal{P}(k+1) & * \\ \mathcal{C}(k) & \mathcal{D}(k) & 0 & 0 & I \end{pmatrix} \succ 0,$$

$$(4.43) \quad \begin{pmatrix} \eta I_{m_w} - \bar{\Psi}(k) & * & * \\ \mathcal{P}(k+1)\tilde{\mathcal{B}}^0(k) & \mathcal{P}(k+1) & * \\ \mathcal{D}(k) & 0_{p_z \times 2n_x} & I_{p_z} \end{pmatrix} + \begin{pmatrix} 0_{m_w \times m_w} & * & * \\ Q(k)\tilde{\mathcal{B}}^1(k) & 0_{2n_x \times 2n_x} & * \\ 0_{p_z \times m_w} & 0_{p_z \times 2n_x} & 0_{p_z \times p_z} \end{pmatrix} \succ 0,$$

$$\begin{pmatrix} \mathcal{P}(N) & * & * \\ 0_{m_w \times 2n_x} & \eta I_{m_w} & * \\ \mathcal{C}(N) & \mathcal{D}(N) & I_{2p_z} \end{pmatrix} \succ 0,$$

$$\begin{pmatrix} \eta I_{m_w} - \bar{\Psi}(N) & * \\ \mathcal{D}(N) & I_{p_z} \end{pmatrix} \succ 0,$$

$$\prod_{k=0}^N \det \bar{\Psi}(k) \geq e^{2a}(\eta - \gamma^2)^{m_w(N+1)},$$

$$\mathcal{P}(k) \succ 0, \quad k = N-1, \dots, 0.$$

Вместо неизвестных матриц фильтра $W(k)$, $H(k)$ переменными системы линейных матричных неравенств становятся матрицы $\mathcal{X}(k)$ и $Q(k)$. Стоит обратить внимание, что при решении неравенств (4.42)–(4.43) необходимо учитывать условия (4.39)–(4.41), ограничивающие область значений матриц $\mathcal{X}(k)$, $Q(k)$. Численный пример решения задачи анизотропийной фильтрации для системы с отказами представлен в разделе Пример.

4.2.2. Фильтр с $H(k) = 0$

Заданы система вида (3.1) и фильтр (3.3). Если матрица $H(k) = 0$, то с учетом условия $\hat{x}(0) = x(0) = 0$ получаем

$$\hat{x}(k+1) = W(k)\hat{x}(k), \quad \hat{x}(0) = 0 \Rightarrow \hat{x}(k) \equiv 0.$$

Следовательно, оценка выхода системы $z(k)$ имеет вид $\hat{z}(k) = G(k)y(k)$. Имеем статический фильтр с одной неизвестной матрицей $G(k)$.

Матрицы фильтра в терминах ошибок имеют следующий вид:

$$(4.44) \quad \begin{aligned} \mathcal{A}(k) &= A(k), & \mathcal{B}(k) &= B(k), \\ \mathcal{C}(k) &= C_z(k) - G(k)C_y(k), & \mathcal{D}(k) &= D_z(k) - G(k)D_y(k). \end{aligned}$$

Подставив (4.44) в (4.25)–(4.29), получим систему линейных матричных неравенств относительно матриц $\mathcal{P}(k)$, $\overline{\Psi}(k)$ и матрицы фильтра $G(k)$, для которых не требуется дополнительных замен переменных.

4.2.3. Фильтр с $G(k) = 0$

Заданы система вида (3.1) и фильтр (3.3). Если $G(k) = 0$, то оценка выхода системы $\hat{z}(k)$ имеет вид

$$(4.45) \quad \hat{z}(k) = F(k)\hat{x}(k).$$

Если исходная система (3.1) с учетом (4.45) является детерминированной, то можно принять $F(k) = C_z(k)$. В противном случае можно принять $F(k) = \mathbf{E}[C_z(k)]$. Таким образом, задача анизотропийной фильтрации сводится к поиску матриц фильтра $W(k)$, $H(k)$. В зависимости от вида матриц $A(k)$ и $B(k)$ будет необходимо ввести замену переменных для устранения произведений компонент неизвестной матрицы $\mathcal{P}(k+1)$ и искомых матриц фильтра $W(k)$, $H(k)$.

4.2.4. Фильтр с $\Gamma(k) = 0$

Заданы система вида (3.1) и фильтр (3.3). Рассмотрим случай $\Gamma(k) = 0$. Поскольку необходимо, чтобы система в ошибках фильтрации принадлежала классу систем с мультипликативными шумами, матрицы (4.8) должны удовлетворять условиям (4.12)–(4.13). Поэтому запишем матрицу $\mathcal{A}(k)$ в следующем виде:

$$(4.46) \quad \begin{aligned} \mathcal{A}(k) &= \begin{pmatrix} \mathcal{A}_{11}(k) & \mathcal{A}_{12}(k) \\ \mathcal{A}_{21}(k) & \mathcal{A}_{22}(k) \end{pmatrix} = \mathcal{A}_0(k) + \sum_{i=1}^M \zeta_{1i}(k)\mathcal{A}_i(k), \\ \mathcal{A}_i(k) &= \begin{pmatrix} \mathcal{A}_{i11}(k) & \mathcal{A}_{i12}(k) \\ \mathcal{A}_{i21}(k) & \mathcal{A}_{i22}(k) \end{pmatrix}, \quad i = \overline{0, M}. \end{aligned}$$

Далее запишем условия для каждой компоненты матрицы (4.46) с учетом выражений матриц системы в ошибках фильтрации вида (4.8):

$$(4.47) \quad A(k) = \mathcal{A}_{011}(k) + \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i11}(k),$$

$$(4.48) \quad 0 = \mathcal{A}_{012}(k) + \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i12}(k),$$

$$(4.49) \quad A(k) - W(k) - H(k)C_y(k) = \mathcal{A}_{021}(k) + \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i21}(k),$$

$$(4.50) \quad W(k) = \mathcal{A}_{022}(k) + \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i22}(k).$$

Из равенства (4.47) видно, что $A(k)$ является матрицей специального вида. Из (4.50) и того факта, что матрица фильтра $W(k)$ не может быть случайной, следует, что у уравнения (4.50) нет решений. Следовательно, условием наличия решения является равенство нулю всех матриц $\mathcal{A}_{i22}(k)$ при случайных величинах $\lambda_{1i}(k)$. Предположим, что $W(k) = \mathcal{A}_{022}(k)$.

С учетом вышесказанного, перепишем (4.49) в следующем виде:

$$(4.51) \quad \begin{aligned} \mathcal{A}_{011}(k) - W(k) &= \mathcal{A}_{021}(k), \\ \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i11}(k) - H(k)C_y(k) &= \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i21}(k). \end{aligned}$$

В общем случае $\mathbf{E}[C_y(k)] \neq 0$. Следовательно, равенство (4.49) является уравнением относительно случайных величин с разными математическими ожиданиями. Необходимо свести (4.49) к уравнению относительно случайных величин с нулевыми математическими ожиданиями. Для этого проведем следующие преобразования (4.51):

$$(4.52) \quad \begin{aligned} \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i11}(k) - H(k)C_y(k) + H(k)\mathbf{E}[C_y(k)] - \\ - H(k)\mathbf{E}[C_y(k)] &= \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i21}(k). \end{aligned}$$

Видно, что слагаемые $H(k)C_y(k)$ и $H(k)\mathbf{E}[C_y(k)]$ можно сгруппировать, получив при матрице $H(k)$ случайную матрицу с нулевым математическим ожиданием. Перепишем (4.52) в виде

$$(4.53) \quad \begin{aligned} \mathcal{A}_{021}(k) &= \mathcal{A}_{011}(k) - W(k) - H(k)\mathbf{E}[C_y(k)], \\ \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i21}(k) &= \sum_{i=1}^{M_1} \lambda_{1i}(k) \mathcal{A}_{i11}(k) + H(k)(\mathbf{E}[C_y(k)] - C_y(k)). \end{aligned}$$

Равенство (4.53) выполняется, если матрица $C_y(k)$ имеет следующий вид:

$$(4.54) \quad C_y(k) = C_{y_0}(k) + \sum_{i=1}^M \lambda_{1i}(k) C_{y_i}(k),$$

где

$$C_{y_0}(k) = \mathbf{E}[C_y(k)], \quad H(k)C_{y_i}(k) = \mathcal{A}_{i11}(k) - \mathcal{A}_{i21}(k).$$

Предположив, что все матрицы $A_i(k)$, $C_{y_i}(k)$, $i = \overline{0, M}$ известны, имеем

$$(4.55) \quad \mathcal{A}(k) = \begin{pmatrix} A_0(k) & 0 \\ A_0(k) - W(k) - H(k)C_{y_0}(k) & W(k) \end{pmatrix} + \sum_{i=1}^M \lambda_{1i}(k) \begin{pmatrix} A_i(k) & 0 \\ A_i(k) - H(k)C_{y_i}(k) & 0 \end{pmatrix}.$$

Далее, в соответствии с условиями (4.8), (4.12) для матриц системы с мультипликативными шумами матрица $\mathcal{B}(k)$ должна удовлетворять выражению

$$(4.56) \quad \mathcal{B}(k) = \mathcal{B}_0(k) + \sum_{i=1}^M \lambda_{2i}(k) \mathcal{B}_i(k).$$

Предположим, что выполняются соотношения

$$(4.57) \quad \begin{aligned} B(k) &= B_0(k) + \sum_{i=1}^M \lambda_{2i}(k) B_i(k), \\ D_y(k) &= D_{y_0}(k) + \sum_{i=1}^M \lambda_{2i}(k) D_{y_i}(k), \\ H(k)D_{y_i}(k) &= \mathcal{B}_{1i} - \mathcal{B}_{2i}, \end{aligned}$$

где все матрицы $B_i(k)$, $D_{y_i}(k)$, $i = \overline{0, M}$ известны. Таким образом, (4.56) с учетом (4.57) имеет вид

$$(4.58) \quad \mathcal{B}(k) = \begin{pmatrix} B_0(k) \\ B_0(k) - H(k)D_{y_0}(k) \end{pmatrix} + \sum_{i=1}^M \lambda_{2i}(k) \begin{pmatrix} B_i(k) \\ B_i(k) - H(k)D_{y_i}(k) \end{pmatrix}.$$

При выполнении (4.54), (4.55) и (4.58) система в ошибках фильтрации является системой с мультипликативными шумами. Запишем теорему об условиях ограниченности анизотропийной нормы этой системы для случая $\Gamma(k) = 0$:

Теорема 2. Заданы линейная дискретная нестационарная система T_{zw} с реализацией в пространстве состояний (4.7) и вещественные числа $a > 0$ и $\gamma > 0$. Тогда a -анизотропийная норма системы на интервале $[0, N]$ удовлетворяет неравенству $\|T_{zw}\|_a \leq \gamma$, если существуют $q = \eta^{-1} > 0$ и матрицы $\mathcal{P}(k)$, $\Psi(k)$, $\mathcal{Y}(k)$, которые являются решением системы линейных матричных неравенств

$$(4.59) \quad \begin{bmatrix} \mathcal{P}(k) & 0 & \mathcal{A}_0^{0\top} \mathcal{P}(k+1) & \overline{\mathcal{A}}^0(k) \mathcal{P}(k+1) & \overline{\mathcal{C}}(k) \\ * & \eta I & \mathcal{B}_0^{0\top}(k) \mathcal{P}(k+1) & 0 & 0 \\ * & * & \mathcal{P}(k+1) & 0 & 0 \\ * & * & * & \Phi_M(\mathcal{P}(k+1)) & 0 \\ * & * & * & * & I \end{bmatrix} +$$

$$+ \begin{bmatrix} 0 & 0 & \mathcal{A}_0^{1\top} \mathcal{Y}(k) & \overline{\mathcal{A}}^1(k) \mathcal{Y}(k) & 0 \\ * & 0 & \mathcal{B}_0^{1\top}(k) \mathcal{Y}(k) & 0 & 0 \\ * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ * & * & * & * & 0 \end{bmatrix} \succ 0,$$

$$(4.60) \quad \begin{bmatrix} \eta I - \overline{\Psi}(k) & \mathcal{B}_0^{0\top}(k) \mathcal{P}(k+1) & \overline{\mathcal{B}}^0(k) \mathcal{P}(k+1) \\ * & \mathcal{P}(k+1) & 0 \\ * & * & \Phi_M(\mathcal{P}(k+1)) \end{bmatrix} +$$

$$+ \begin{bmatrix} 0 & \mathcal{B}_0^{1\top}(k) \mathcal{Y}(k) & \overline{\mathcal{B}}^1(k) \mathcal{Y}(k) \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix} \succ 0,$$

$$(4.61) \quad \begin{bmatrix} \mathcal{P}(N) & 0 & \overline{\mathcal{C}}(N) \\ * & \eta I & 0 \\ * & * & I \end{bmatrix} \succ 0,$$

$$(4.62) \quad \begin{bmatrix} \eta I - \overline{\Psi}(N) & 0 \\ 0 & I \end{bmatrix} \succ 0,$$

$$(4.63) \quad \prod_{k=0}^N \det \overline{\Psi}(k) \geq e^{2a} (\eta - \gamma^2)^{m_w(N+1)},$$

$$\mathcal{Y}(k) = \begin{pmatrix} 0 & 0 \\ \mathcal{Y}_{21} & \mathcal{Y}_{22} \end{pmatrix}.$$

Доказательство теоремы приведено в Приложении 2.

Аналогично рассмотренным выше частным случаям фильтров, при решении неравенств (4.59)–(4.63) необходимо учитывать вид неизвестной матрицы $\mathcal{Y}(k)$ в соответствии с заменой переменных (см. док-во теоремы в Приложении). Решив полученные неравенства относительно матриц $\mathcal{P}(k)$, $\overline{\Psi}(k)$ и $\mathcal{Y}(k)$, используем обратную замену переменных и находим матрицы фильтра $W(k)$ и $H(k)$.

4.2.5. Фильтр с $\Gamma(k) \neq 0$

Заданы система вида (3.1) и фильтр (3.3). Рассмотрим случай $\Gamma(k) \neq 0$. Матрицы системы и фильтра должны удовлетворять следующим условиям:

$$(4.64) \quad A(k) = \mathcal{A}_{011}(k) + \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i11}(k),$$

$$(4.65) \quad 0 = \mathcal{A}_{012}(k) + \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i12}(k),$$

$$(4.66) \quad A(k) - W(k) + H(k)(\Gamma(k) - C_y(k)) = \mathcal{A}_{021}(k) + \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i21}(k),$$

$$(4.67) \quad W(k) - H(k)\Gamma(k) = \mathcal{A}_{022}(k) + \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i22}(k),$$

$$(4.68) \quad B(k) = \mathcal{B}_{01}(k) + \sum_{i=1}^M \lambda_{2i}(k) \mathcal{B}_{i1}(k).$$

Так же, как и в случае $\Gamma(k) = 0$, равенство (4.66) имеет место, если математические ожидания случайных величин обеих сторон выражения (4.66) равны. Предположив, что $\Gamma(k) = \mathbf{E}[C_y(k)]$, преобразуем (4.66)–(4.67) к следующему виду:

$$\mathcal{A}_{021}(k) = \mathcal{A}_{011}(k) - W(k),$$

$$(4.69) \quad \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i21}(k) = \sum_{i=1}^M \lambda_{1i}(k) \mathcal{A}_{i11}(k) + H(k)(\mathbf{E}[C_y(k)] - C_y(k)).$$

Если условие (4.69) выполняется, то условия ограниченности анизотропийной нормы системы в ошибках фильтрации (4.7) выводятся тем же образом, как и в случае $\Gamma(k) = 0$. Применив аналогичные преобразования и замены переменных, получим условия ограниченности анизотропийной нормы для рассматриваемого частного случая фильтра:

Теорема 3. Заданы линейная дискретная нестационарная система $T_{\bar{z}w}$ с мультипликативными шумами и реализацией в пространстве состояний (4.7) и вещественные числа $a \geq 0$, $\gamma \geq 0$. Тогда a -анизотропийная норма системы на интервале $k \in [0, N]$ удовлетворяет условию $\| \|T_{\bar{z}w}\| \|_a \leq \gamma$, если

существуют $q = \eta^{-1} > 0$ и матрицы $\mathcal{P}(k)$, $\Psi(k)$, $\mathcal{Y}(k)$, при которых выполняются неравенства

$$\begin{aligned}
& \begin{bmatrix} \mathcal{P}(k) & 0 & \mathcal{A}_0^{0\top} \mathcal{P}(k+1) & \overline{\mathcal{A}}^0(k) \mathcal{P}(k+1) & \overline{\mathcal{C}}(k) \\ * & \eta I & \mathcal{B}_0^{0\top}(k) \mathcal{P}(k+1) & 0 & 0 \\ * & * & \mathcal{P}(k+1) & 0 & 0 \\ * & * & * & \Phi_M(\mathcal{P}(k+1)) & 0 \\ * & * & * & * & I \end{bmatrix} + \\
& + \begin{bmatrix} 0 & 0 & \mathcal{A}_0^{1\top} \mathcal{Y}(k) & \overline{\mathcal{A}}^1(k) \mathcal{Y}(k) & 0 \\ * & 0 & \mathcal{B}_0^{1\top}(k) \mathcal{Y}(k) & 0 & 0 \\ * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ * & * & * & * & 0 \end{bmatrix} \succ 0, \\
& \begin{bmatrix} \eta I - \overline{\Psi}(k) & \mathcal{B}_0^{0\top}(k) \mathcal{P}(k+1) & \overline{\mathcal{B}}^0(k) \mathcal{P}(k+1) \\ * & \mathcal{P}(k+1) & 0 \\ * & * & \Phi_M(\mathcal{P}(k+1)) \end{bmatrix} + \\
& + \begin{bmatrix} 0 & \mathcal{B}_0^{1\top}(k) \mathcal{Y}(k) & \overline{\mathcal{B}}^1(k) \mathcal{Y}(k) \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix} \succ 0, \\
& \begin{bmatrix} \mathcal{P}(N) & 0 & \overline{\mathcal{C}}(N) \\ * & \eta I & 0 \\ * & * & I \end{bmatrix} \succ 0, \\
& \begin{bmatrix} \eta I - \overline{\Psi}(N) & 0 \\ 0 & I \end{bmatrix} \succ 0, \\
& \prod_{k=0}^N \det \overline{\Psi}(k) \geq e^{2a} (\eta - \gamma^2)^{m_w(N+1)},
\end{aligned}$$

$$\mathcal{Y}(k) = \begin{pmatrix} 0 & 0 \\ \mathcal{Y}_{21} & \mathcal{Y}_{22} \end{pmatrix}.$$

Стоит отметить, что единственным отличием частных случаев фильтров $\Gamma(k) = 0$ и $\Gamma(k) \neq 0$ является компонента $H(k)\mathbf{E}[C_y(k)]$ в матрице $\mathcal{A}_{021}(k)$. Даже при отсутствии в фильтре слагаемых $H(k)\Gamma(k)$ компонента $H(k)\mathbf{E}[C_y(k)]$ все равно появляется для центрирования случайной матрицы $C_y(k)$ и тем самым обеспечения выполнения ограничений на матрицы системы с мультипликативными шумами. В следующем разделе приведен численный пример решения задачи анизотропной фильтрации для системы с отказами в измерениях.

5. Пример

Рассматривается система с отказами в измерениях вида

$$(5.1) \quad F : \begin{cases} x(k+1) = A(k)x(k) + B(k)w(k) + r(k), \\ z(k) = C_z(k)x(k) + D_z(k)w(k), \\ y(k) = \lambda(k)C_y(k)x(k) + D_y(k)w(k), \end{cases}$$

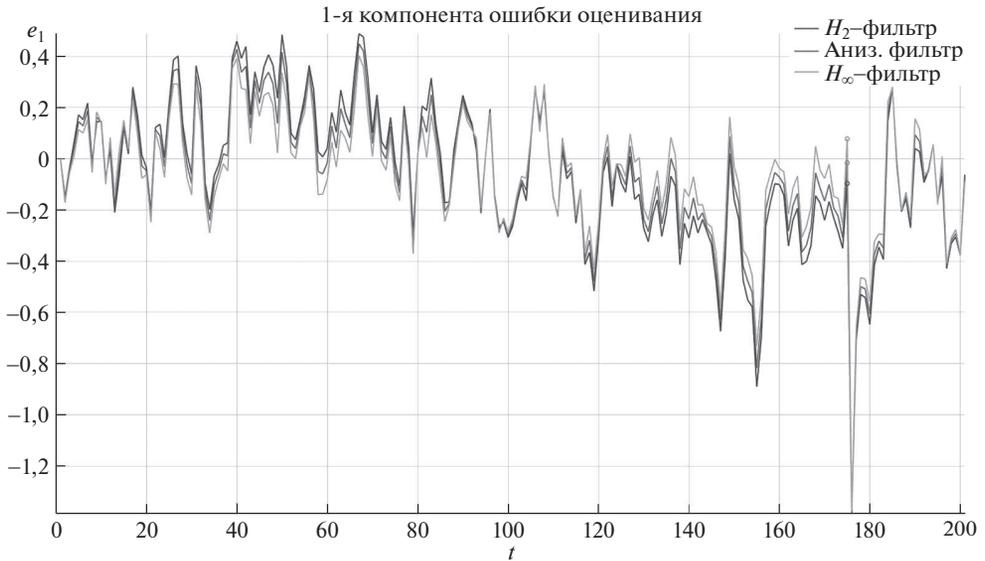


Рис. 1. Ошибка оценивания $e_1(t)$.



Рис. 2. Ошибка оценивания $e_2(t)$.

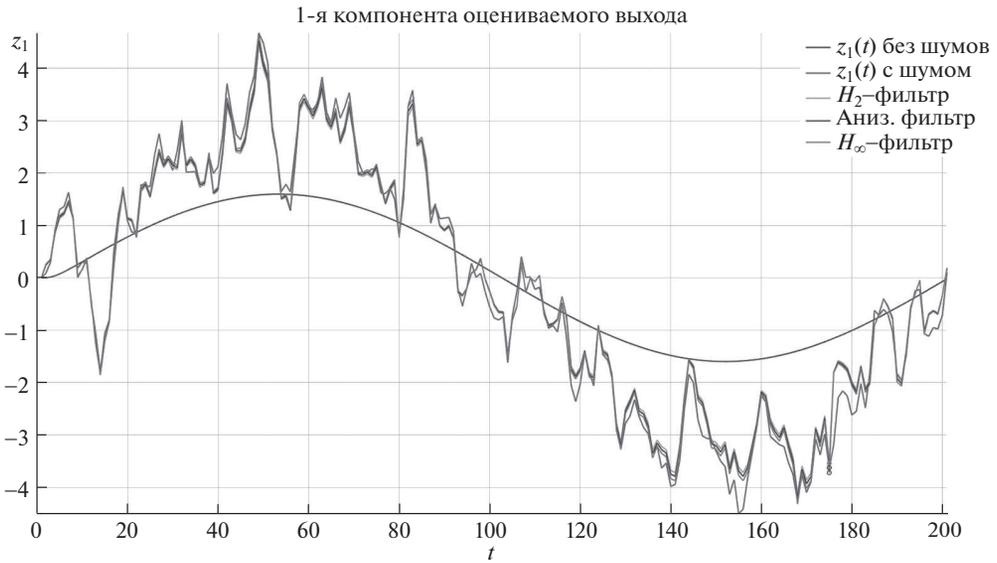


Рис. 3. Оцениваемый выход $z_1(t)$.

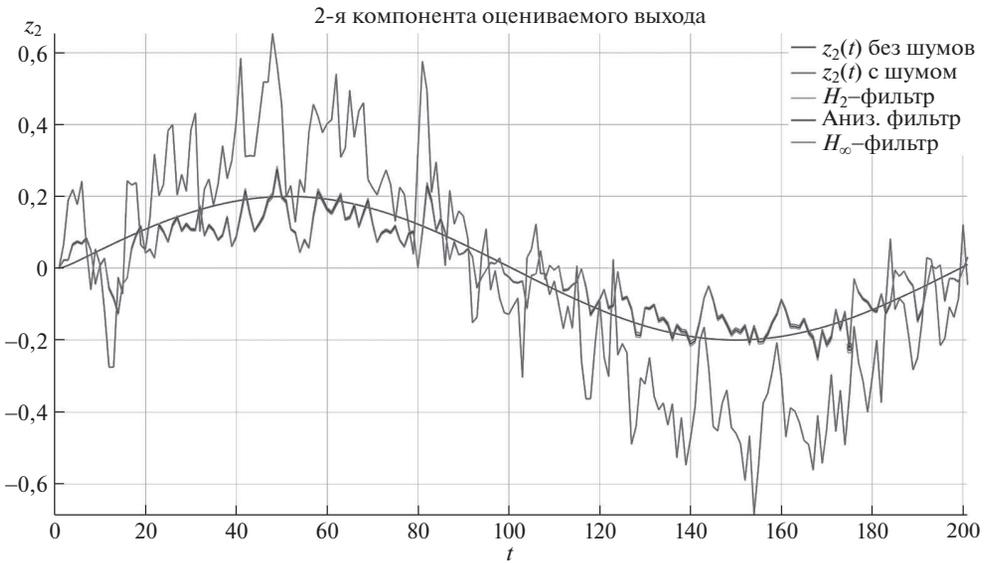


Рис. 4. Оцениваемый выход $z_2(t)$.

с матрицами

$$\begin{aligned}
 A &= A_0 = \begin{pmatrix} 0,5 & 4 \\ 0 & 0,5 \end{pmatrix}, & B &= B_0 = \begin{pmatrix} 0 & 0,8 \\ 1 & 0 \end{pmatrix}, \\
 C_z &= C_{z_0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & D_z &= D_{z_0} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \\
 C_y &= (0 \ 1), & \sigma(\lambda_k) &= \sqrt{p(1-p)}, & D_y &= D_{y_0} = (0 \ 0,64), \\
 N &= 200, & r(k) &= \left(0, 0, 2 \sin \left(\frac{k}{5} \frac{1}{2\pi} \right) \right)^\top,
 \end{aligned}$$

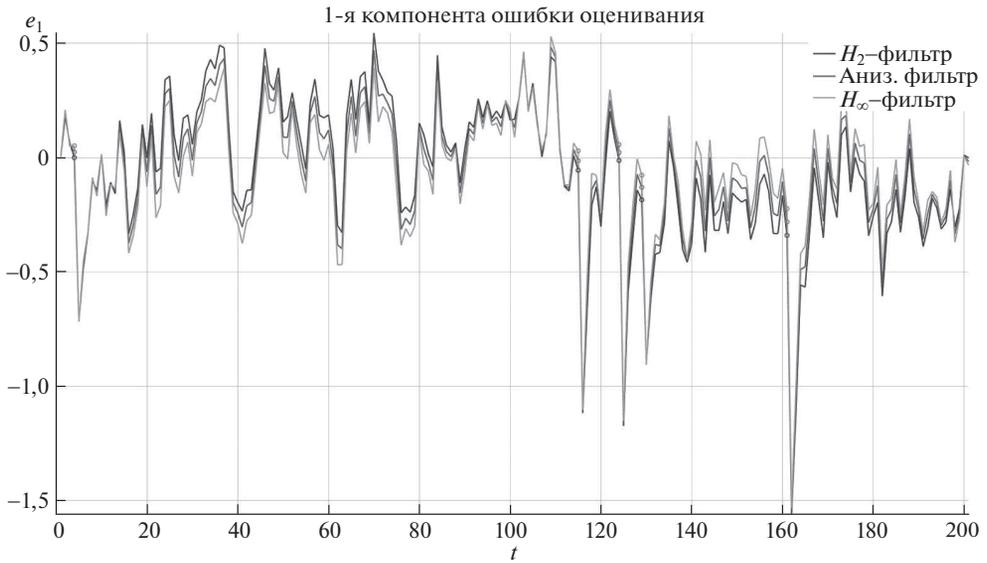


Рис. 5. Ошибка оценивания $e_1(t)$.



Рис. 6. Ошибка оценивания $e_2(t)$.

где $\sigma(\lambda_k)$ — среднеквадратичное отклонение случайной величины $\lambda(k)$ при матрице C_y , $r(k)$ — синусоидальное задающее воздействие, подаваемое на вход системы. По предложенному алгоритму для данной системы синтезированы три a -анизотропных фильтра при фиксированном параметре анизотропии a и различных значениях вероятности отказа $1 - p$. Каждый из этих анизотропных фильтров сравнивается с субоптимальным \mathcal{H}_2 -фильтром и субоптимальным \mathcal{H}_∞ -фильтром, построенными для той же системы с соответствующими параметрами вероятности отказа.

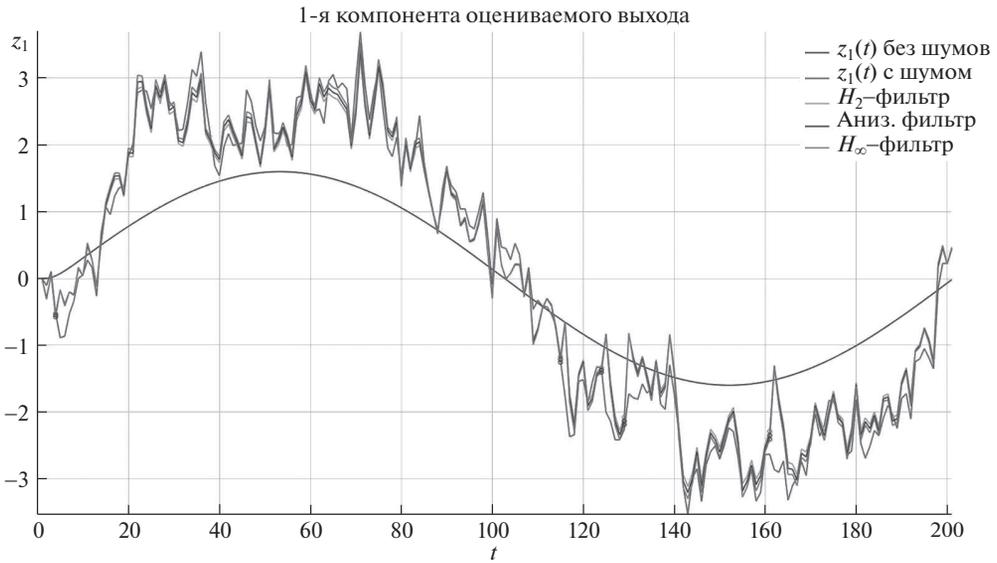


Рис. 7. Оцениваемый выход $z_1(t)$.

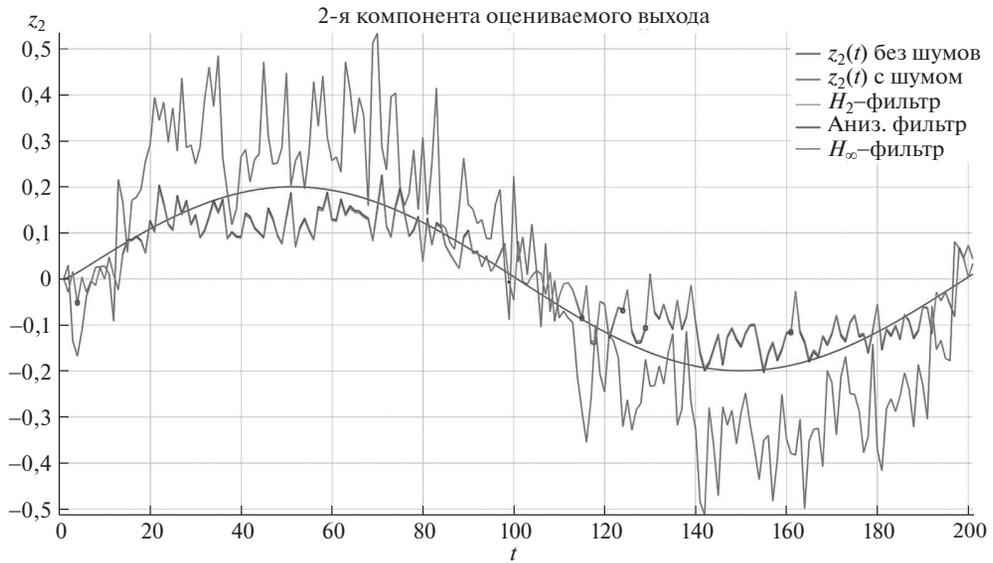


Рис. 8. Оцениваемый выход $z_2(t)$.

В результате решения системы линейных матричных неравенств (4.39)–(4.40) для системы (5.1) были получены последовательности значений матриц фильтра $W(k)$ и $H(k)$. При решении системы линейных матричных неравенств использовались пакеты Yalmip Matlab toolbox и SeDuMi optimization package.

Результаты моделирования представлены в виде графиков значений оцениваемого выхода $z(t)$ и ошибок оценивания выхода $e(t) = z(t) - \hat{z}(t)$ для трех значений параметра p . Точками на графиках обозначены моменты отказа в



Рис. 9. Ошибка оценивания $e_1(t)$.



Рис. 10. Ошибка оценивания $e_2(t)$.

измерениях, т.е. $\lambda(k) = 0$. Ниже приведены значения параметра вероятности отказа, при которых производилось численное решение поставленной задачи фильтрации:

- Вероятность отказа в измерениях $1 - p = 0,01$. Графики ошибки фильтрации и оцениваемого выхода представлены на рис. 1–4.
- Вероятность отказа в измерениях $1 - p = 0,03$. Графики ошибки фильтрации и оцениваемого выхода представлены на рис. 5–8.

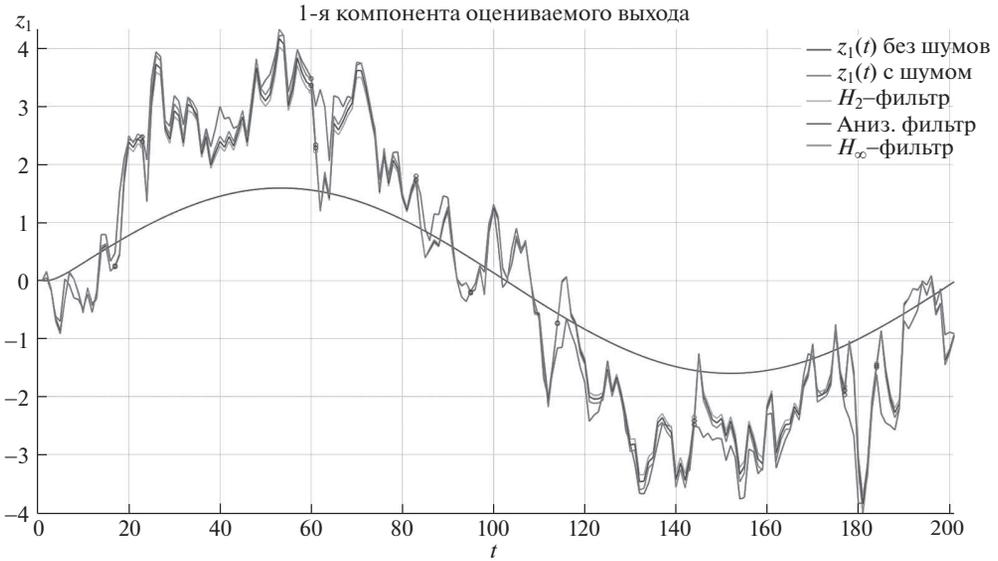


Рис. 11. Оцениваемый выход $z_1(t)$.

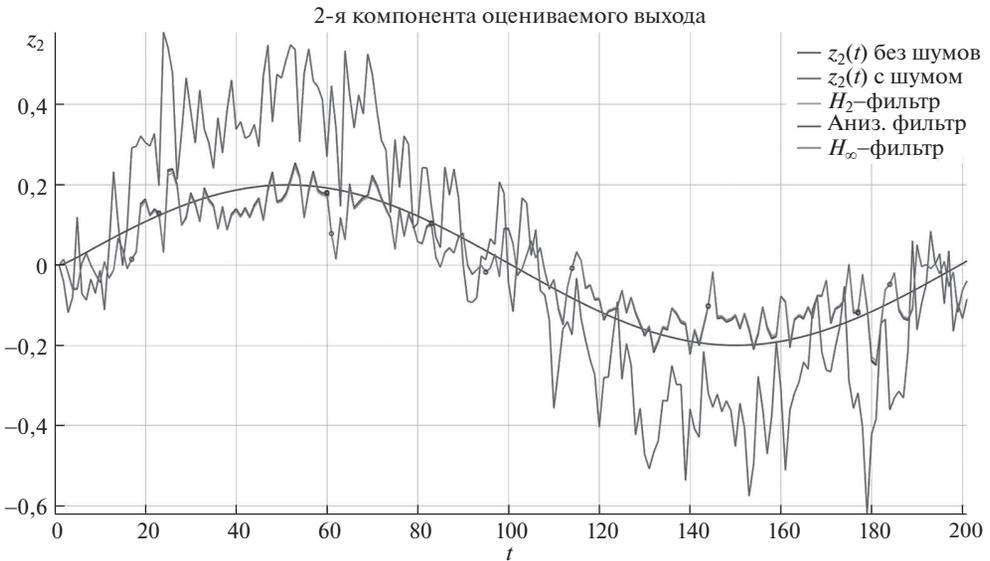


Рис. 12. Оцениваемый выход $z_2(t)$.

- Вероятность отказа в измерениях $1 - p = 0,05$. Графики ошибки фильтрации и оцениваемого выхода представлены на рис. 9–12.

На графиках видно, как при увеличении числа отказов в измерениях увеличивается ошибка оценивания выходов $z(t)$ рассматриваемой системы. Но после каждого отказа оценка постепенно сходится к реальному значению выхода системы, т.е. ошибка оценивания продолжает колебаться около нуля. Также видно, что субоптимальный анизотропный фильтр эффективнее

\mathcal{H}_2 -субоптимального фильтра, но менее эффективен, чем \mathcal{H}_∞ -субоптимальный фильтр, что обусловлено свойствами анизотропийной нормы. Однако, как было сказано ранее, анизотропийный фильтр может менять свои свойства в зависимости от параметра a , что позволяет решать задачу фильтрации для конкретного случая входного возмущения менее энергозатратно с технической точки зрения, чем с помощью \mathcal{H}_∞ -субоптимального фильтра.

6. Заключение

Решена задача анизотропийной фильтрации на конечном временном интервале для линейной дискретной нестационарной системы с мультипликативными шумами. Качество решения задачи фильтрации характеризуется анизотропийной нормой системы в ошибках оценивания наблюдаемого выхода исходной системы. На основании леммы об ограниченности анизотропийной нормы системы с мультипликативными шумами выведены линейные матричные неравенства относительно неизвестных матриц искомого фильтра. Рассмотрены частные случаи задачи анизотропийной фильтрации, включая задачу фильтрации для системы с отказами в измерениях. Использование предложенного метода решения задачи анизотропийной фильтрации продемонстрировано с помощью численного примера задачи для системы с отказами в измерениях с различными значениями вероятности отказа.

ПРИЛОЖЕНИЕ 1

Доказательство теоремы 1. Приведем выражение (4.22) к виду неравенства с учетом положительной определенности матрицы $\mathcal{S}(k)$:

$$\Theta(k) \succ 0 : \Theta(k) \succ \mathcal{S}(k), \quad \Theta^{-1}(k) \prec \mathcal{S}^{-1}(k),$$

$$(П.1.1) \quad \Theta^{-1}(k) = \Psi(k) \prec I_{m_w} - \left(q \sum_{i=0}^M \mathcal{D}_i^\top(k) \mathcal{D}_i(k) + \sum_{i=0}^M \mathcal{B}_i^\top(k) \mathcal{R}(k+1) \mathcal{B}_i(k) \right), \\ \Psi^{-1}(k) \succ \mathcal{S}(k).$$

Применив (П.1.1) к системе (4.21)–(4.24), получим систему неравенств следующего вида:

$$(П.1.2) \quad \widehat{\mathcal{R}}(k) \succ \mathcal{R}(k), \quad \widehat{\mathcal{R}}(N+1) = 0, \\ \widehat{\mathcal{R}}(k) \succ \mathcal{A}_0^\top(k) \widehat{\mathcal{R}}(k+1) \mathcal{A}_0(k) + \sum_{i=1}^M a_i \mathcal{A}_i^\top(k) \widehat{\mathcal{R}}(k+1) \mathcal{A}_i(k) + q \sum_{i=0}^M c_i \mathcal{C}_i^\top(k) \mathcal{C}_i(k) + \\ + \left(\mathcal{B}_0^\top \widehat{\mathcal{R}}(k+1) \mathcal{A}_0 + q \mathcal{D}_0^\top \mathcal{C}_0 \right)^\top \Psi^{-1}(k) \left(\mathcal{B}_0^\top \widehat{\mathcal{R}}(k+1) \mathcal{A}_0 + q \mathcal{D}_0^\top \mathcal{C}_0 \right),$$

$$(II.1.3) \quad \Psi(k) \prec I_{m_w} - \left(q \sum_{i=0}^M \mathcal{D}_i^\top(k) \mathcal{D}_i(k) + \sum_{i=0}^M \mathcal{B}_i^\top(k) \widehat{\mathcal{R}}(k+1) \mathcal{B}_i(k) \right),$$

$$(II.1.4) \quad \sum_{k=0}^N \ln \det \Psi(k) \geq 2a + m_w(N+1) \ln(1 - q\gamma^2).$$

Заметим, что в неравенстве (II.1.4) есть произведение двух переменных системы $q\gamma^2$. Для решения системы неравенств необходимо устранить данную нелинейную компоненту с помощью преобразования самого выражения или замены переменных. Следовательно, введем следующие новые переменные:

$$(II.1.5) \quad \begin{aligned} \eta &= q^{-1}, \quad \overline{\Psi} = \eta \Psi, \\ \mathcal{P}(k) &= \eta \widehat{\mathcal{R}}(k), \quad \mathcal{P}(N+1) = \eta \widehat{\mathcal{R}}(N+1). \end{aligned}$$

Для компактности записи дальнейших выкладок с линейными матричными неравенствами введем обозначения

$$(II.1.6) \quad \overline{\mathcal{A}}(k) = (\mathcal{A}_1^\top \ \dots \ \mathcal{A}_M^\top), \quad \overline{\mathcal{B}}(k) = (\mathcal{B}_1^\top \ \dots \ \mathcal{B}_M^\top),$$

$$(II.1.7) \quad \overline{\mathcal{C}}(k) = (\mathcal{C}_1^\top \ \dots \ \mathcal{C}_M^\top), \quad \overline{\mathcal{D}}(k) = (\mathcal{D}_1^\top \ \dots \ \mathcal{D}_M^\top).$$

Для приведения неравенств (II.1.2), (II.1.3) к форме системы линейных матричных неравенств (ЛМН) потребуется лемма Шура (в англоязычной литературе — Schur complement lemma). Формулировку данной леммы можно найти в [20].

Применим лемму Шура к системе неравенств Риккати (II.1.2)–(II.1.3) с учетом замены переменных (II.1.5) и обозначений (II.1.6)–(II.1.7):

$$(II.1.8) \quad \begin{pmatrix} \mathcal{P}(k) & * & * & * & * & * \\ 0 & \eta I_{m_w} & * & * & * & * \\ \mathcal{A}_0(k) & \mathcal{B}_0(k) & \mathcal{P}^{-1}(k+1) & * & * & * \\ \overline{\mathcal{A}}^\top(k) & 0 & 0 & \ddots & * & * \\ \mathcal{C}_0(k) & \mathcal{D}_0(k) & 0 & 0 & \mathcal{P}^{-1}(k+1) & * \\ \overline{\mathcal{C}}^\top(k) & 0 & 0 & 0 & 0 & I \end{pmatrix} \succ 0,$$

$$(II.1.9) \quad \begin{pmatrix} \mathcal{P}(N) & * & * & * \\ 0 & \eta I_{m_w} & * & * \\ \mathcal{C}_0(N) & \mathcal{D}_0(N) & I_{p_z} & * \\ \overline{\mathcal{C}}(N) & 0 & 0 & I \end{pmatrix} \succ 0,$$

$$(II.1.10) \quad \begin{pmatrix} \eta I_{m_w} - \overline{\Psi}(k) & * & * & * & * \\ \mathcal{B}_0^\top(k) & \mathcal{P}^{-1}(k+1) & * & * & * \\ \vdots & 0 & \ddots & * & * \\ \mathcal{B}_M^\top(k) & 0 & 0 & \mathcal{P}^{-1}(k+1) & * \\ \overline{\mathcal{D}}^\top(k) & 0 & \dots & 0 & I \end{pmatrix} \succ 0,$$

$$(П.1.11) \quad \begin{pmatrix} \eta I_{m_w} - \overline{\Psi}(N) & * \\ \mathcal{D}(N) & I_{p_z} \end{pmatrix} \succ 0,$$

$$(П.1.12) \quad \prod_{k=0}^N \det \overline{\Psi}(k) \geq e^{2a} (\eta - \gamma^2)^{m_w(N+1)},$$

$$\mathcal{P}(k) \succ 0, \quad k = N-1, \dots, 0.$$

Для устранения обратных матриц в неравенствах (П.1.8), (П.1.10) сделаем конгруэнтные преобразования неравенств с помощью блочно-диагональных матриц вида

$$\text{blockdiag} (I_{2n_x}, I_{m_w}, \mathcal{P}(k+1), \dots, \mathcal{P}(k+1), I_{p_z(M)}) \quad \text{и}$$

$$\text{blockdiag} (I_{2n_x}, \mathcal{P}(k+1), \dots, \mathcal{P}(k+1), I_{p_z(M)}).$$

Для упрощения записи преобразованных ЛМН введем обозначения

$$(П.1.13) \quad \overline{\mathcal{A}}(k)\mathcal{P}(k+1) = \begin{pmatrix} \mathcal{A}_1^\top \mathcal{P}(k+1) & \dots & \mathcal{A}_M^\top \mathcal{P}(k+1) \end{pmatrix},$$

$$(П.1.14) \quad \overline{\mathcal{B}}(k)\mathcal{P}(k+1) = \begin{pmatrix} \mathcal{B}_1^\top \mathcal{P}(k+1) & \dots & \mathcal{B}_M^\top \mathcal{P}(k+1) \end{pmatrix}.$$

Применив обозначения (П.1.13)–(П.1.14) и необходимые конгруэнтные преобразования к неравенствам (П.1.8)–(П.1.12), получим искомые линейные матричные неравенства (4.25)–(4.29).

ПРИЛОЖЕНИЕ 2

Доказательство теоремы 2. Запишем условия ограниченности анизотропийной нормы для системы в ошибках фильтрации с учетом приведенных выше выкладок:

$$\begin{bmatrix} \mathcal{P}(k) & 0 & \mathcal{A}_0^\top \mathcal{P}(k+1) & \overline{\mathcal{A}}(k)\mathcal{P}(k+1) & \overline{\mathcal{C}}(k) \\ * & \eta I & \mathcal{B}_0^\top(k)\mathcal{P}(k+1) & 0 & 0 \\ * & * & \mathcal{P}(k+1) & 0 & 0 \\ * & * & * & \Phi_M(\mathcal{P}(k+1)) & 0 \\ * & * & * & * & I \end{bmatrix} \succ 0,$$

$$\begin{bmatrix} \eta I - \overline{\Psi}(k) & \mathcal{B}_0^\top(k)\mathcal{P}(k+1) & \overline{\mathcal{B}}(k)\mathcal{P}(k+1) \\ * & \mathcal{P}(k+1) & 0 \\ * & * & \Phi_M(\mathcal{P}(k+1)) \end{bmatrix} \succ 0,$$

$$\begin{bmatrix} \mathcal{P}(N) & 0 & \overline{\mathcal{C}}(N) \\ * & \eta I & 0 \\ * & * & I \end{bmatrix} \succ 0,$$

$$\begin{bmatrix} \eta I - \overline{\Psi}(N) & 0 \\ 0 & I \end{bmatrix} \succ 0,$$

$$\prod_{k=0}^N \det \bar{\Psi}(k) \geq e^{2a(\eta - \gamma^2)m_w(N+1)}.$$

С учетом (4.55), (4.58) очевидно, что в неравенствах есть нелинейные компоненты в виде произведений неизвестных матриц в произведениях $\mathcal{A}_i(k)\mathcal{P}(k+1)$ и $\mathcal{B}_i^\top(k)\mathcal{P}(k+1)$.

Для устранения нелинейных членов в неравенствах используется следующая замена переменных неравенств:

$$(П.2.1) \quad \mathcal{X}(k) = \begin{pmatrix} 0 & 0 \\ W(k) & H(k) \end{pmatrix}, \quad \mathcal{Y}(k) = \mathcal{X}(k)\mathcal{P}(k+1).$$

Применив замену переменных (П.2.1) к системе неравенств (4.55), (4.58), получим

$$(П.2.2) \quad \mathcal{A}(k) = \mathcal{A}_0^0(k) + \mathcal{X}(k)\mathcal{A}_0^1(k) + \sum_{i=1}^M \lambda_{1i}(k) (\mathcal{A}_i^0(k) + \mathcal{X}(k)\mathcal{A}_i^1(k)),$$

$$(П.2.3) \quad \mathcal{B}(k) = \mathcal{B}_0^0(k) + \mathcal{X}(k)\mathcal{B}_0^1(k) + \sum_{i=1}^M \lambda_{1i}(k) (\mathcal{B}_i^0(k) + \mathcal{X}(k)\mathcal{B}_i^1(k)),$$

где

$$(П.2.4) \quad \mathcal{A}_0^0(k) = \begin{pmatrix} A_0(k) & 0 \\ A_0(k) & 0 \end{pmatrix}, \quad \mathcal{A}_0^1(k) = \begin{pmatrix} -I & I \\ -C_{y_0}(k) & 0 \end{pmatrix},$$

$$(П.2.5) \quad \mathcal{A}_i^0(k) = \begin{pmatrix} A_i(k) & 0 \\ A_i(k) & 0 \end{pmatrix}, \quad \mathcal{A}_i^1(k) = \begin{pmatrix} 0 & 0 \\ -C_{y_i}(k) & 0 \end{pmatrix},$$

$$(П.2.6) \quad \mathcal{B}_0^0(k) = \begin{pmatrix} B_0(k) \\ B_0(k) \end{pmatrix}, \quad \mathcal{B}_0^1(k) = \begin{pmatrix} 0 \\ -D_{y_0} \end{pmatrix},$$

$$(П.2.7) \quad \mathcal{B}_i^0(k) = \begin{pmatrix} B_i(k) \\ B_i(k) \end{pmatrix}, \quad \mathcal{B}_i^1(k) = \begin{pmatrix} 0 \\ -D_{y_i} \end{pmatrix}.$$

Подставив (П.2.1)–(П.2.7) в матричные неравенства, устраняем нелинейные компоненты из системы матричных неравенств. В итоге получаем искомые линейные матричные неравенства (4.59)–(4.63).

СПИСОК ЛИТЕРАТУРЫ

1. *Anderson B.D.O., Moore J.B.* Optimal Filtering. New Jersey: Prentice Hall, 1979.
2. *Hassibi B., Sayed A., Kailath T.* Indefinite Quadratic Estimation and Control: A Unified Approach to \mathcal{H}_2 and \mathcal{H}_∞ Theories. Philadelphia: SIAM, 1999.
3. *Simon D.* Optimal State Estimation: Kalman, \mathcal{H}_∞ , and Nonlinear Approaches. New Jersey: Wiley, 2006.

4. *Vladimirov I.G., Kurdjukov A.P., Semyonov A.V.* Anisotropy Of Signals And The Entropy Of Linear Stationary Systems // Dokl. Math. 1995. V. 51. P. 388–390.
5. *Vladimirov I.G., Kurdjukov A.P., Semyonov A.V.* On Computing The Anisotropic Norm Of Linear Discrete-Time-Invariant Systems // Proc. 13 IFAC World Congress. 1996. P. 179–184.
6. *Vladimirov I.G., Diamond P., Kloeden P.* Anisotropy-based robust performance analysis of linear discrete time varying systems. CADSMAP Research Report 01-01. The University of Queensland, Australia. 2001.
7. *Vladimirov I.G., Diamond P.* Robust Filtering in Finite Horizon Linear Discrete Time Varying Systems by Minimum Anisotropic Norm Criterion. CADSMAP Res. Report 01-05. 2001 [completed in 2001].
8. *Maximov E.A., Kurdyukov A.P., Vladimirov I.G.* Anisotropic Norm Bounded Real Lemma for Linear Time-Varying System // Preprint 18th IFAC World Congr. Milano. 2011. P. 4701–4706.
9. *Тимин В.Н., Курдюков А.П.* Субоптимальная анизотропная фильтрация на конечном горизонте // АиТ. 2016. № 1. С. 5–29.
Timin V.N., Kurdyukov A.P. Suboptimal Anisotropic Filtering in a Finite Horizon // Autom. Remote Control. 2016. V. 77. No. 1. P. 1–20.
10. *Yaesh I., Stoica A.-M.* Linear Time-varying Anisotropic Filtering its Application to Nonlinear Systems State Estimation // Proc. Eur. Control Conf. Strasbourg. 2014. June 24–27. P. 975–980.
11. *Gershon E., Shaked U., Yaesh I.* \mathcal{H}_∞ -control And Filtering Of Discrete-Time Stochastic Systems With Multiplicative Noise // Automatica. 2001. No. 37. P. 409–417.
12. *Todorov E., Weiwei Li.* Optimal Control Methods Suitable For Biomechanical Systems // Proceedings of the 25th Annual International Conference of the IEEE Engineering in Biology and Medicine Society. Mexico. 2003. P. 1758–1761.
13. *Todorov E.* Stochastic Optimal Control And Estimation Methods Adapted To The Noise Characteristics Of The Sensorimotor System // Neural Comput. 2005. No. 17. P. 1084–1108.
14. *Semyonov A.V., Vladimirov I.G., Kurdyukov A.P.* Stochastic approach to H-infinity optimization // Proc. 33rd IEEE Conf. Decision and Control. 1994. V. 3. P. 2249–2250.
15. *Пакшин П.В.* Оценивание состояния и синтез управления для дискретных линейных систем с аддитивными и мультипликативными шумами // АиТ. 1978. № 4. С. 75–85.
Pakshin P. V. State Estimation And Control Design For Digital Linear Systems With Multiplicative Noises // Autom. Remote Control. 1978. V. 39. Is. 4. P. 526–636.
16. *Домбровский В.В., Ляшенко Е.А.* Линейно-квадратичное управление дискретными системами со случайными параметрами и мультипликативными шумами с применением к оптимизации инвестиционного портфеля // АиТ. 2003. № 10. С. 50–65.
Dombrovskii V.V., Lyashenko E.A. A Linear Quadratic Control for Discrete Systems with Random Parameters and Multiplicative Noise and Its Application to Investment Portfolio Optimization // Autom. Remote Control. 2003. V. 64. Is. 10. P. 1558–1570.
17. *Belov I.R., Yurchenkov A. V., Kustov A. Yu.* Anisotropy-Based Bounded Real Lemma for Multiplicative Noise Systems: the Finite Horizon Case. 27th Mediterranean Conference on Control and Automation. 2019.

18. *Freiling G., Ionescu V.* Monotonicity And Convexity Properties Of Matrix Riccati Equations // IMA Journal of Mathematical Control and Information. March 2001. V. 18. Is. 1. P. 61–72.
19. *Freiling G., Ionescu V.* Time-Varying Discrete Riccati Equation: Some Monotonicity Results // Linear Algebra and its Applications. 1999. No. 286. P. 135–148.
20. *Zhang F.* The Schur Complement and Its Applications // Springer, Boston, MA. 2005.

Статья представлена к публикации членом редколлегии М.В. Хлебниковым.

Поступила в редакцию 27.07.2020

После доработки 03.01.2021

Принята к публикации 15.01.2021

© 2021 г. Ю.Е. ОБЖЕРИН, д-р техн. наук (objsev@mail.ru),
С.М. СИДОРОВ (xaevec@mail.ru),
М.М. НИКИТИН (m.nikitin.1979@gmail.com)
(Севастопольский государственный университет),
С.Г. ГЛЕЧ, канд. техн. наук (sergeyglech@gmail.com)
(Севастопольский экономико-гуманитарный институт (филиал)
Крымского федерального университета им. В.И. Вернадского)

СКРЫТАЯ МАРКОВСКАЯ МОДЕЛЬ НА ОСНОВЕ СУПЕРПОЗИЦИИ ДВУХ ПРОЦЕССОВ ВОССТАНОВЛЕНИЯ¹

В процессе функционирования системы, для которой построена полумарковская модель, не всегда возможно при изменении ее состояний получить всю информацию, содержащуюся в кодах состояний, а есть возможность получить некоторый сигнал (информацию), связанный с состояниями полумарковской модели. В этом случае состояния полумарковской модели можно считать скрытыми (ненаблюдаемыми). В данной статье на примере суперпозиции двух независимых процессов восстановления рассматриваются подход к построению скрытой марковской модели на основе полумарковского процесса с фазовым пространством состояний общего вида и использование такой модели для анализа функционирования моделируемой системы на основе полученного вектора сигналов.

Ключевые слова: полумарковский процесс, полумарковская модель, скрытая марковская модель, суперпозиция процессов восстановления, вектор сигналов, оценка характеристик, прогнозирование состояний.

DOI: 10.31857/S0005231021060039

1. Введение

Полумарковские процессы широко используются для моделирования систем различного назначения: технических [1–3], информационных [4–6], энергетических [2, 7, 8], систем массового обслуживания [9–10], анализа финансовых рисков [5, 11] и других.

В процессе функционирования системы, для которой построена полумарковская модель, необходимо оценить, насколько построенная модель согласуется с полученными в процессе функционирования системы данными, уточнить ее, провести анализ функционирования системы и спрогнозировать ее состояния на основе полученного вектора сигналов. Решить эти задачи позволяют скрытые марковские модели [5, 12–15] и скрытые полумарковские модели [16–19].

При построении полумарковской модели необходимо ввести фазовое пространство состояний системы. В ряде случаев достаточно использовать ко-

¹ Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований в рамках научного проекта № 18-01-00392а.

нечное или счетное множества состояний, которые отражают физические состояния системы [2, 20]. Часто при построении полумарковской модели системы в фазовые состояния системы приходится вводить дополнительные непрерывные компоненты, которые необходимы для корректного построения модели [21–26]. Этими компонентами могут быть: время, прошедшее с начала работы элемента системы; время, оставшееся до окончания восстановления элемента системы; время, оставшееся до проведения контроля системы; величина оставшегося резерва времени и т.д. Отметим, что эти дополнительные непрерывные компоненты содержат важную информацию о функционировании системы. В этом случае приходится использовать дискретно-непрерывное фазовое пространство состояний, а для построения полумарковской модели и анализа функционирования системы необходимо использовать теорию полумарковских процессов с общим фазовым пространством состояний [21–25].

Важной составной частью полумарковского процесса является вложенная цепь Маркова (ВЦМ), которая отвечает за переходы между состояниями системы. Фазовое пространство состояний ВЦМ совпадает с фазовым пространством состояний полумарковского процесса. Второй важной составляющей полумарковского процесса являются времена пребывания в состояниях на переходах. При построении полумарковских моделей ряда систем переходные вероятности ВЦМ и функции распределения времен пребывания на переходах определяются общими параметрами. Так, для полумарковских моделей надежности систем они определяются распределениями времен безотказной работы и восстановления элементов системы.

При функционировании системы, для которой построена полумарковская модель, не всегда удастся при изменениях ее состояний полностью получить информацию, содержащуюся в кодировке состояний, а есть возможность получить некоторый сигнал (информацию), связанный с состояниями ВЦМ (полумарковского процесса). Например, в фазовом состоянии полумарковского процесса для каждого элемента системы указано, находится он в рабочем состоянии или на восстановлении, а при использовании системы можно получить сигнал только о числе работоспособных элементов. Для систем массового обслуживания, например, могут быть получены данные только о числе свободных приборов, а не о состоянии каждого прибора, содержащиеся в фазовых состояниях. При функционировании системы бывает сложно или невозможно получить значения дополнительных непрерывных компонент, которые, как было отмечено, несут важную информацию о функционировании системы. В этих случаях состояния ВЦМ и полумарковской модели можно считать скрытыми (ненаблюдаемыми). Поэтому возникает задача нахождения оценок характеристик и прогнозирования состояний ВЦМ и полумарковской модели на основе полученного вектора сигналов. Как отмечено выше, это можно сделать, используя аппарат теории скрытых марковских моделей.

В данной статье на примере суперпозиции двух независимых процессов восстановления [21–23] рассматривается подход к построению скрытой марковской модели на основе полумарковского процесса с фазовым пространством состояний общего вида, используя наличие у полумарковского процесса ВЦМ. Результаты, содержащиеся в [12, 13], обобщаются на случай ненаблюдаемой цепи Маркова с общим фазовым пространством состояний.

Скрытая марковская модель, построенная на основе суперпозиции двух независимых процессов восстановления, используется для нахождения оценок характеристик ВЦМ суперпозиции и сигналов на основе полученного вектора сигналов.

Для ВЦМ находятся условные вероятности состояний на $2 \leq k \leq n$, $(n + 1)$ -м и шагах при полученном векторе сигналов размера n . Для вектора сигналов получены условные вероятности значений вектора сигналов на $(n + 1)$ -м шаге, безусловные вероятности вектора сигналов и финальные вероятности для значений вектора сигналов.

Для полноты изложения описана суперпозиция двух независимых процессов восстановления, построенная в публикациях [21–23].

При реализации рассматриваемого подхода можно применять математические пакеты, допускающие аналитические преобразования, используя при этом рекуррентные формулы.

Эффективным методом решения проблемы размерности моделей является алгоритм стационарного фазового укрупнения полумарковских систем [21–23], разработанный В.С. Королюком и А.Ф. Турбиным. Используя этот алгоритм, можно перейти к укрупненной полумарковской модели, вплоть до полумарковской модели с конечным множеством состояний, а затем использовать рассматриваемый подход к построению скрытой марковской модели [15].

2. Построение скрытой марковской модели

Построим скрытую марковскую модель на основе суперпозиции двух независимых процессов восстановления.

Дадим, обобщая результаты, содержащееся в [12, 13] определение скрытой марковской модели, когда ненаблюдаемая цепь Маркова имеет фазовое пространство состояний общего вида.

Пусть $\{X_n, n = 1, 2, 3, \dots\}$ – однородная цепь Маркова, определенная в измеримом пространстве (E, \mathfrak{F}) , с переходными вероятностями

$$P(\mathbf{x}, B) = P(X_{n+1} \in B | X_n = \mathbf{x}), \quad \mathbf{x} \in E, \quad B \in \mathfrak{F},$$

и начальным распределением вероятностей

$$\pi(B) = P(X_1 \in B), \quad B \in \mathfrak{F}.$$

Жирными буквами \mathbf{x} , \mathbf{e} , \mathbf{y} , \mathbf{z} обозначаются элементы фазового пространства состояний E .

Предположим, что множество сигналов J конечно:

$$J = \{s_1, s_2, \dots, s_m\}.$$

Далее предположим, что когда цепь Маркова попадает в состояние $\mathbf{x} \in E$, то независимо от предыдущих состояний цепи Маркова и сигналов излучается сигнал $s \in J$ с вероятностью

$$(1) \quad R(s | \mathbf{x}) = P(S_n = s | X_n = \mathbf{x}), \quad \mathbf{x} \in E, \quad s \in J, \quad \sum_{s \in J} R(s | \mathbf{x}) = 1.$$

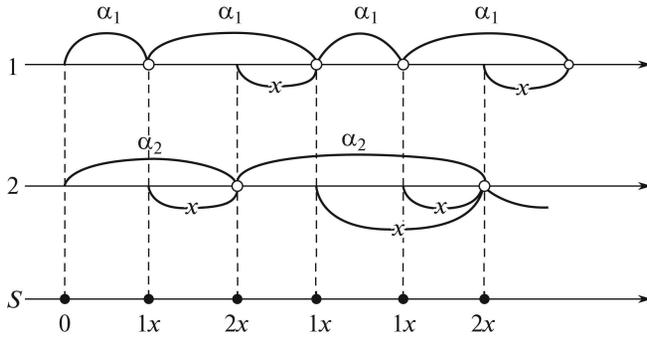


Рис. 1. Временная диаграмма функционирования системы.

Таким образом, если S_n представляет собой n -й сигнал, то

$$P(S_1 = s | X_1 = \mathbf{x}) = R(s | \mathbf{x}),$$

$$P(S_n = s | X_1, S_1, \dots, X_{n-1}, S_{n-1}, X_n = \mathbf{x}) = R(s | \mathbf{x}).$$

Модель такого типа, в которой последовательность сигналов $S_1, S_2, \dots, \dots, S_n, \dots$ является наблюдаемой, в то время как цепь Маркова $X_1, X_2, \dots, \dots, X_n, \dots$ является ненаблюдаемой, называется скрытой марковской моделью [5, 12, 13].

Опишем, следуя [21–23], суперпозицию двух независимых процессов восстановления, на основе которой строится скрытая марковская модель.

Рассмотрим систему S , состоящую из двух элементов, где функционирование каждого элемента описывается процессом восстановления [21–23] с временами восстановления α_1 и α_2 , имеющими функции распределения $G_1(x) = P(\alpha_1 \leq x)$, $G_2(x) = P(\alpha_2 \leq x)$ и плотности распределения $g_1(x)$, $g_2(x)$. Случайные величины α_1 и α_2 предполагаются независимыми, имеющими конечные математические ожидания.

Для описания функционирования системы S используется полумарковский процесс $\xi(t)$ со следующим дискретно-непрерывным фазовым пространством состояний:

$$E = \{0, 1x, 2x; x > 0\}.$$

Кодировка состояний следующая:

- 0 – оба элемента восстановлены, начальное состояние;
- 1x – восстановился первый элемент; время, оставшееся до восстановления второго элемента, осталось равно $x > 0$;
- 2x – восстановился второй элемент; время, оставшееся до восстановления первого элемента, осталось равно $x > 0$.

Временная диаграмма функционирования системы S представлена на рис. 1.

Плотности вероятностей переходов ВЦМ имеют вид [21–23]:

$$p_0^{1x} = \int_0^{\infty} g_2(x+t)g_1(t)dt, \quad x > 0, \quad p_0^{2x} = \int_0^{\infty} g_1(x+t)g_2(t)dt, \quad x > 0;$$

$$p_{1x}^{1y} = g_1(x-y), \quad 0 < y < x, \quad p_{1x}^{2y} = g_1(x+y), \quad y > 0;$$

$$p_{2x}^{1y} = g_2(x+y), \quad y > 0, \quad p_{2x}^{2y} = g_2(x-y), \quad 0 < y < x.$$

Времена пребывания системы S в состояниях определяются следующим образом:

$$\theta_0 = \alpha_1 \wedge \alpha_2, \quad \theta_{1x} = \alpha_1 \wedge x, \quad \theta_{2x} = \alpha_2 \wedge x, \quad \wedge - \text{знак минимума.}$$

В [21] показано, что ВЦМ имеет стационарное распределение, плотности которого определяются формулами:

$$(2) \quad \rho(1x) = \rho_0 \bar{G}_2(x), \quad \rho(2x) = \rho_0 \bar{G}_1(x),$$

где $\bar{G}_i(x) = 1 - G_i(x)$, $i = 1, 2$, $\rho_0 = \frac{1}{M\alpha_1 + M\alpha_2}$, $M\alpha_i$ – математическое ожидание случайной величины α_i .

Перейдем к построению скрытой марковской модели, определенной на основе суперпозиции двух независимых процессов восстановления. Предположим, что при использовании системы S состояния ВЦМ $1x$, $2x$ не наблюдаются (скрытые состояния), а наблюдается только номер элемента, который восстановился на следующем переходе.

Следовательно, множество сигналов имеет вид

$$J = \{1, 2\}.$$

Рассмотрим связь между состояниями ВЦМ и сигналами, т.е. определим функцию $\bar{R}(s|\mathbf{x})$.

1. Состояние 0:

$$P(S_n = 1 | X_n = 0) = P(\alpha_1 < \alpha_2) = \int_0^{\infty} \bar{G}_2(t)g_1(t)dt,$$

$$P(S_n = 2 | X_n = 0) = P(\alpha_2 < \alpha_1) = \int_0^{\infty} \bar{G}_1(t)g_2(t)dt.$$

2. Состояния $1x$:

$$P(S_n = 1 | X_n = 1x) = P(\alpha_1 < x) = G_1(x),$$

$$P(S_n = 2 | X_n = 1x) = P(\alpha_1 > x) = \bar{G}_1(x).$$

3. Состояния $2x$:

$$P(S_n = 1 | X_n = 2x) = P(\alpha_2 > x) = \bar{G}_2(x),$$

$$P(S_n = 2 | X_n = 2x) = P(\alpha_2 < x) = G_2(x).$$

Таблица 1. Функция $R(s | \mathbf{x})$ связи состояний ВЦМ с сигналами

Сигнал, s Состояние, \mathbf{x}	$s = 1$	$s = 2$
0	$p_1 = \int_0^\infty \bar{G}_2(t)g_1(t)dt$	$p_2 = \int_0^\infty \bar{G}_1(t)g_2(t)dt$
$1x$	$G_1(x)$	$\bar{G}_1(x)$
$2x$	$\bar{G}_2(x)$	$G_2(x)$

Функция $R(s | \mathbf{x})$ связи состояний ВЦМ с сигналами представлена в табл. 1.

3. Оценка характеристик на основе полученного вектора сигналов

Пусть $\bar{S}^n = (S_1, S_2, \dots, S_n)$ – случайный вектор первых n сигналов. Для фиксированной последовательности сигналов $\bar{s}_n = (s_1, s_2, \dots, s_n)$ пусть $\bar{s}_k = (s_1, s_2, \dots, s_k)$, $k \leq n$. Требуется оценить характеристики ВЦМ и вектора сигналов на основе полученного вектора сигналов \bar{s}_n .

I. Вначале определим условную вероятность ВЦМ в момент n при условии, что $\bar{S}^n = \bar{s}_n$, т.е. найдем условную вероятность

$$(3) \quad P(X_n \in B | \bar{S}^n = \bar{s}_n), \quad B \in \mathfrak{F}.$$

Для нахождения этой вероятности, следуя [12, 13], введем функции (прямые переменные), которые играют важную роль в дальнейшем:

$$F_n(B) = P(\bar{S}^n = \bar{s}_n, X_n \in B), \quad B \in \mathfrak{F},$$

или в дифференциальной форме

$$F_n(d\mathbf{x}) = P(\bar{S}^n = \bar{s}_n, X_n \in d\mathbf{x}).$$

Имеем, что

$$\begin{aligned} F_n(d\mathbf{x}) &= P(\bar{S}^{n-1} = \bar{s}_{n-1}, S_n = s_n, X_n \in d\mathbf{x}) = \\ &= \int_E P(\bar{S}^{n-1} = \bar{s}_{n-1}, X_{n-1} \in de, X_n \in d\mathbf{x}, S_n = s_n) = \\ &= \int_E F_{n-1}(de)P(X_n \in d\mathbf{x}, S_n = s_n | \bar{S}^{n-1} = \bar{s}_{n-1}, X_{n-1} = e) = \\ &= \int_E F_{n-1}(de)P(X_n \in d\mathbf{x}, S_n = s_n | X_{n-1} = e) = \\ &= \int_E F_{n-1}(de)P(e, d\mathbf{x})R(s_n | \mathbf{x}), \end{aligned}$$

где использовалось, что

$$\begin{aligned} & P(X_n \in d\mathbf{x}, S_n = s_n | X_{n-1} = \mathbf{e}) = \\ & = P(X_n \in d\mathbf{x} | X_{n-1} = \mathbf{e})P(S_n = s_n | X_n = \mathbf{x}, X_{n-1} = \mathbf{e}) = \\ & = P(X_n \in d\mathbf{x} | X_{n-1} = \mathbf{e})P(S_n = s_n | X_n = \mathbf{x}) = P(\mathbf{e}, d\mathbf{x})R(s_n | \mathbf{x}). \end{aligned}$$

Следовательно, для плотности $f_n(\mathbf{x})$ функции $F_n(B)$ справедлива рекуррентная формула

$$(4) \quad f_n(\mathbf{x}) = R(s_n | \mathbf{x}) \int_E f_{n-1}(\mathbf{e})p(\mathbf{e}, \mathbf{x})d\mathbf{e},$$

где $p(\mathbf{e}, \mathbf{x})$ – плотность вероятности переходов $P(\mathbf{e}, B)$ ненаблюдаемой цепи Маркова.

Найдем функции $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$, $f_4(\mathbf{x})$.

1. $F_1(d\mathbf{x}) = P(X_1 \in d\mathbf{x}, S_1 = s_1) = P(X_1 \in d\mathbf{x})P(S_1 = s_1 | X_1 = \mathbf{x}) = \pi(d\mathbf{x})R(s_1 | \mathbf{x})$, следовательно

$$(5) \quad f_1(\mathbf{x}) = \pi(\mathbf{x})R(s_1 | \mathbf{x}).$$

2. Применяя рекуррентную формулу (4), последовательно получаем:

$$(6) \quad f_2(\mathbf{x}) = R(s_2 | \mathbf{x}) \int_E R(s_1 | \mathbf{e})p(\mathbf{e}, \mathbf{x})\pi(\mathbf{e})d\mathbf{e},$$

$$(7) \quad f_3(\mathbf{x}) = R(s_3 | \mathbf{x}) \int_E R(s_2 | \mathbf{e})p(\mathbf{e}, \mathbf{x})d\mathbf{e} \int_E R(s_1 | \mathbf{y})p(\mathbf{y}, \mathbf{e})\pi(\mathbf{y})d\mathbf{y},$$

$$\begin{aligned} f_4(\mathbf{x}) &= R(s_4 | \mathbf{x}) \int_E R(s_3 | \mathbf{e})p(\mathbf{e}, \mathbf{x})d\mathbf{e} \int_E R(s_2 | \mathbf{z})p(\mathbf{z}, \mathbf{e})d\mathbf{z} \times \\ &\quad \times \int_E R(s_1 | \mathbf{y})p(\mathbf{y}, \mathbf{z})\pi(\mathbf{y})d\mathbf{y}, \end{aligned}$$

где $\pi(\mathbf{x})$ – плотность начального распределения $\pi(B)$.

Найдем функции $F_1(B)$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$ для построенной скрытой марковской модели в случае вектора сигналов $\bar{s}_3 = (s_1, s_2, s_3)$.

Используя формулы (5)–(7), получаем

$$(8) \quad F_1(B) = \begin{cases} 0, & \mathbf{x} = 0 \notin B, \\ R(s_1 | 0), & \mathbf{x} = 0 \in B, \end{cases}$$

$$(9) \quad f_2(\mathbf{x}) = \begin{cases} R(s_1 | 0)R(s_2 | 1E)\varphi_1(x), & \mathbf{x} = 1x, \\ R(s_1 | 0)R(s_2 | 2E)\varphi_2(x), & \mathbf{x} = 2x, \end{cases}$$

где

$$\varphi_1(x) = \int_0^{\infty} g_2(x+t)g_1(t)dt, \quad \varphi_2(x) = \int_0^{\infty} g_1(x+t)g_2(t)dt,$$

$$(10) \quad f_3(\mathbf{x}) = \begin{cases} R(s_1|0)R(s_3|1E) \left[\int_0^{\infty} R(s_2|2y)g_2(y+x)\varphi_2(y)dy + \right. \\ \left. + \int_0^{\infty} R(s_2|1y)g_1(y-x)\varphi_1(y)dy \right], & \mathbf{x} = 1x, \\ R(s_1|0)R(s_3|2E) \left[\int_0^{\infty} R(s_2|1y)g_1(y+x)\varphi_1(y)dy + \right. \\ \left. + \int_x^{\infty} R(s_2|2y)g_2(y-x)\varphi_2(y)dy \right], & \mathbf{x} = 2x, \end{cases}$$

значения $R(s|\mathbf{x})$ находятся из табл. 1.

Пусть полученный вектор сигналов $\bar{s} = (s_1, s_2, s_3)$ имеет вид: $\bar{s}_3 = (1, 2, 1)$, $s_1 = 1$, $s_2 = 2$, $s_3 = 1$. В этом случае, используя формулы (8)–(10) и табл. 1, получаем, что функции $F_1(\mathbf{B})$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$ имеют вид:

$$(11) \quad F_1(\mathbf{B}) = \begin{cases} 0, & \mathbf{x} = 0 \notin \mathbf{B}, \\ p_1, & \mathbf{x} = 0 \in \mathbf{B}, \end{cases}$$

$$(12) \quad f_2(\mathbf{x}) = \begin{cases} p_1 \bar{F}_1(x)\varphi_1(x), & \mathbf{x} = 1x, \\ p_1 F_2(x)\varphi_2(x), & \mathbf{x} = 2x, \end{cases}$$

$$(13) \quad f_3(\mathbf{x}) = \begin{cases} p_1 G_1(x) \left[\int_0^{\infty} G_2(y)g_2(y+x)\varphi_2(y)dy + \right. \\ \left. + \int_x^{\infty} \bar{G}_1(y)g_1(y-x)\varphi_1(y)dy \right], & \mathbf{x} = 1x, \\ p_1 \bar{G}_2(x) \left[\int_0^{\infty} \bar{G}_1(y)g_1(y+x)\varphi_1(y)dy + \right. \\ \left. + \int_x^{\infty} \bar{G}_1(y)g_2(y-x)\varphi_2(y)dy \right], & \mathbf{x} = 2x. \end{cases}$$

Аналогично можно найти вид функций $F_1(\mathbf{B})$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$ и для других векторов \bar{s}_3 сигналов.

В качестве конкретного примера рассмотрим случай, когда случайные величины α_1 и α_2 имеют экспоненциальное распределение:

$$G_1(x) = 1 - e^{-\lambda_1 x}, \quad g_1(x) = \lambda_1 e^{-\lambda_1 x}, \quad G_2(x) = 1 - e^{-\lambda_2 x}, \quad g_2(x) = \lambda_2 e^{-\lambda_2 x}.$$

Тогда

$$\varphi_1(x) = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} e^{-\lambda_2 x}, \quad \varphi_2(x) = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} e^{-\lambda_1 x}, \quad p_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2}, \quad p_2 = \frac{\lambda_2}{\lambda_1 + \lambda_2}.$$

Используя формулы (11)–(13), найдем выражения функций $F_1(\mathbf{B})$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$ для векторов сигналов $\bar{s}_3 = (1, 1, 1)$, $\bar{s}_3 = (1, 2, 1)$, $\bar{s}_3 = (1, 1, 2)$, $\bar{s}_3 = (1, 2, 2)$.

1. Для вектора $\bar{s}_3 = (1, 1, 1)$:

$$F_1(\mathbf{B}) = \begin{cases} 0, & \mathbf{x} = 0 \notin \mathbf{B}, \\ p_1, & \mathbf{x} = 0 \in \mathbf{B}, \end{cases}$$

$$f_2(\mathbf{x}) = \begin{cases} \lambda_2 p_1^2 (1 - e^{-\lambda_1 x}) e^{-\lambda_2 x}, & \mathbf{x} = 1x, \\ \lambda_2 p_1^2 e^{-(\lambda_1 + \lambda_2)x}, & \mathbf{x} = 2x, \end{cases}$$

$$f_3(\mathbf{x}) = \begin{cases} \lambda_2 p_1^2 (1 - e^{-\lambda_1 x}) e^{-\lambda_2 x} \left[p_1 - \frac{\lambda_1}{2\lambda_1 + \lambda_2} e^{-\lambda_1 x} + \frac{\lambda_2}{\lambda_1 + 2\lambda_2} \right], & \mathbf{x} = 1x, \\ \lambda_2 p_1^2 e^{-(\lambda_1 + \lambda_2)x} \left[p_1 - \frac{\lambda_1}{2\lambda_1 + \lambda_2} e^{-\lambda_1 x} + \frac{\lambda_2}{\lambda_1 + 2\lambda_2} e^{-\lambda_2 x} \right], & \mathbf{x} = 2x. \end{cases}$$

2. В случае вектора $\bar{s}_3 = (1, 2, 1)$:

$$F_1(\mathbf{B}) = \begin{cases} 0, & \mathbf{x} = 0 \notin \mathbf{B}, \\ p_1, & \mathbf{x} = 0 \in \mathbf{B}, \end{cases}$$

$$f_2(\mathbf{x}) = \begin{cases} \lambda_2 p_1^2 e^{-(\lambda_1 + \lambda_2)x}, & \mathbf{x} = 1x, \\ \lambda_2 p_1^2 (1 - e^{-\lambda_2 x}) e^{-\lambda_1 x}, & \mathbf{x} = 2x, \end{cases}$$

$$f_3(\mathbf{x}) = \begin{cases} \lambda_2 p_1^2 (1 - e^{-\lambda_1 x}) e^{-\lambda_2 x} \left[p_2 + \frac{\lambda_1}{2\lambda_1 + \lambda_2} e^{-\lambda_1 x} - \frac{\lambda_2}{\lambda_1 + 2\lambda_2} \right], & \mathbf{x} = 1x, \\ \lambda_2 p_1^2 e^{-(\lambda_1 + \lambda_2)x} \left[p_2 + \frac{\lambda_1}{2\lambda_1 + \lambda_2} e^{-\lambda_1 x} - \frac{\lambda_2}{\lambda_1 + 2\lambda_2} e^{-\lambda_2 x} \right], & \mathbf{x} = 2x. \end{cases}$$

3. Если $\bar{s}_3 = (1, 1, 2)$, то

$$F_1(\mathbf{B}) = \begin{cases} 0, & \mathbf{x} = 0 \notin \mathbf{B}, \\ p_1, & \mathbf{x} = 0 \in \mathbf{B}, \end{cases}$$

$$f_2(\mathbf{x}) = \begin{cases} \lambda_2 p_1^2 (1 - e^{-\lambda_1 x}) e^{-\lambda_2 x}, & \mathbf{x} = 1x, \\ \lambda_2 p_1^2 e^{-(\lambda_1 + \lambda_2)x}, & \mathbf{x} = 2x, \end{cases}$$

$$f_3(\mathbf{x}) = \begin{cases} p_1^2 \lambda_2 e^{-(\lambda_1 + \lambda_2)x} \left(p_1 - \frac{\lambda_1}{2\lambda_1 + \lambda_2} e^{-\lambda_1 x} + \frac{\lambda_2}{\lambda_1 + 2\lambda_2} \right), & \mathbf{x} = 1x, \\ p_1^2 \lambda_2 (1 - e^{-\lambda_2 x}) e^{-\lambda_1 x} \left(p_1 - \frac{\lambda_1}{2\lambda_1 + \lambda_2} + \frac{\lambda_2}{\lambda_1 + 2\lambda_2} e^{-\lambda_2 x} \right), & \mathbf{x} = 2x. \end{cases}$$

4. Для вектора $\bar{s}_3 = (1, 2, 2)$:

$$F_1(B) = \begin{cases} 0, & \mathbf{x} = 0 \notin B, \\ p_1, & \mathbf{x} = 0 \in B, \end{cases}$$

$$f_2(\mathbf{x}) = \begin{cases} \lambda_2 p_1^2 e^{-(\lambda_1 + \lambda_2)x}, & \mathbf{x} = 1x, \\ \lambda_2 p_1^2 (1 - e^{-\lambda_2 x}) e^{-\lambda_1 x}, & \mathbf{x} = 2x, \end{cases}$$

$$(14) \quad f_3(\mathbf{x}) = \begin{cases} p_1^2 \lambda_2 e^{-(\lambda_1 + \lambda_2)x} \left(p_2 + \frac{\lambda_1}{2\lambda_1 + \lambda_2} e^{-\lambda_1 x} - \frac{\lambda_2}{\lambda_1 + 2\lambda_2} \right), & \mathbf{x} = 1x, \\ p_1^2 \lambda_2 (1 - e^{-\lambda_2 x}) e^{-\lambda_1 x} \left(p_2 + \frac{\lambda_1}{2\lambda_1 + \lambda_2} - \frac{\lambda_2}{\lambda_1 + 2\lambda_2} e^{-\lambda_2 x} \right), & \mathbf{x} = 2x. \end{cases}$$

Для остальных векторов сигналов $\bar{s}_3 = (2, 2, 2)$, $\bar{s}_3 = (2, 1, 2)$, $\bar{s}_3 = (2, 2, 1)$, $\bar{s}_3 = (2, 1, 1)$ функции $F_1(B)$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$ могут быть определены с использованием функций, найденных для ранее рассмотренных векторов сигналов. Так, эти функции для вектора $\bar{s}_3 = (2, 1, 2)$ могут быть получены из функций для вектора $\bar{s}_3 = (1, 2, 1)$ заменой λ_1 на λ_2 , p_1 на p_2 , состояний $1x$ и $2x$ на $2x$ и $1x$ соответственно.

Используя функции $F_n(B)$, $f_n(\mathbf{x})$, перейдем к нахождению вероятности (3):

$$(15) \quad P(\bar{S}^n = \bar{s}_n) = \int_E P(\bar{S}^n = \bar{s}_n, X_n \in d\mathbf{x}) = \int_E F_n(d\mathbf{x}) = \int_E f_n(\mathbf{x}) d\mathbf{x}.$$

Введем функцию

$$H_n(B) = P(X_n \in B | \bar{S}^n = \bar{s}_n), \quad B \in \mathfrak{F},$$

или в дифференциальной форме

$$H_n(d\mathbf{x}) = P(X_n \in d\mathbf{x} | \bar{S}^n = \bar{s}_n).$$

Имеем, что

$$\begin{aligned} H_n(d\mathbf{x}) &= P(X_n \in d\mathbf{x} | \bar{S}^n = \bar{s}_n) = \frac{P(X_n \in d\mathbf{x}, \bar{S}^n = \bar{s}_n)}{P(\bar{S}^n = \bar{s}_n)} = \\ &= \frac{F_n(d\mathbf{x})}{F_n(E)} = \frac{F_n(d\mathbf{x})}{\int_E F_n(d\mathbf{e})}. \end{aligned}$$

Следовательно, плотность $h_n(\mathbf{x})$ функции $H_n(B)$ определяется формулой:

$$(16) \quad h_n(\mathbf{x}) = c_n \cdot f_n(\mathbf{x}), \quad c_n = \frac{1}{F_n(E)} = \frac{1}{\int_E f_n(\mathbf{x}) d\mathbf{x}}.$$

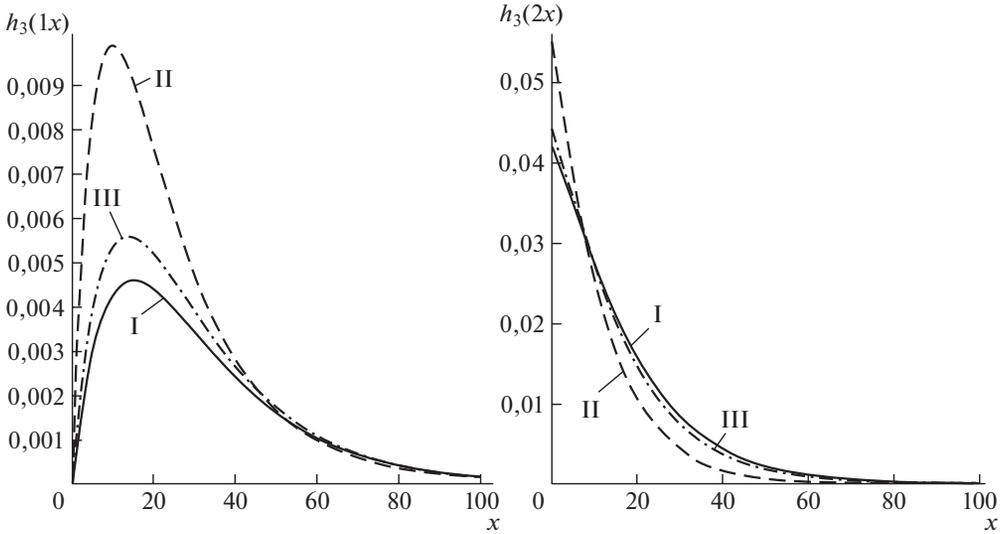


Рис. 2. Графики функций $h_3(1x)$, $h_3(2x)$.

В качестве примера, найдем $P(\bar{S}^3 = \bar{s}_3)$ и функцию $h_3(\mathbf{x})$ для вектора сигналов $\bar{s}_3 = (1, 2, 1)$. Используя формулы (14)–(16), получаем

$$\begin{aligned}
 (17) \quad \frac{1}{c_3} &= P(\bar{S}^3 = \bar{s}_3) = \int_{E_1} f_3(1x) dx + \int_{E_2} f_3(2x) dx = \\
 &= \lambda_2 p_1^2 \left[\frac{\lambda_1 p_1}{(2\lambda_1 + \lambda_2)^2} + \frac{p_1 p_2}{\lambda_1 + 2\lambda_2} + \frac{p_1 p_2 + 1}{2\lambda_1 + \lambda_2} - \frac{\lambda_2}{(\lambda_1 + 2\lambda_2)^2} \right], \\
 h_3(\mathbf{x}) &= \begin{cases} c_3 \cdot \lambda_2 p_1^2 (1 - e^{-\lambda_1 x}) e^{-\lambda_2 x} \left[p_2 + \frac{\lambda_1}{2\lambda_1 + \lambda_2} e^{-\lambda_1 x} - \frac{\lambda_2}{\lambda_1 + 2\lambda_2} \right], & \mathbf{x} = 1x, \\ c_3 \cdot \lambda_2 p_1^2 e^{-(\lambda_1 + \lambda_2)x} \left[p_2 + \frac{\lambda_1}{2\lambda_1 + \lambda_2} e^{-\lambda_1 x} - \frac{\lambda_2}{\lambda_1 + 2\lambda_2} e^{-\lambda_2 x} \right], & \mathbf{x} = 2x. \end{cases}
 \end{aligned}$$

Рассмотрим следующие наборы значений параметров λ_1 , λ_2 :

- I. ($\lambda_1 = 0,025$, $\lambda_2 = 0,05$),
- II. ($\lambda_1 = 0,05$, $\lambda_2 = 0,05$),
- III. ($\lambda_1 = 0,025$, $\lambda_2 = 0,0125$).

Графики функций $h_3(1x)$, $h_3(2x)$ при значениях параметров I, II, III представлены на рис. 2.

Введем следующие подмножества пространства состояний E :

$$E_1 = \{1x, x > 0\}, \quad E_2 = \{2x, x > 0\}.$$

В табл. 2 приведены вероятности $P(\bar{S}_3 = \bar{s}_3)$, $P(X_3 \in E_1 | \bar{s}_3)$ и $P(X_3 \in E_2 | \bar{s}_3)$ при значениях параметров I, II, III для различных вариантов вектора сигнала

Таблица 2. Значения вероятностей $P(\bar{S}_3 = \bar{s}_3)$, $P(X_3 \in E_1 | \bar{s}_3)$ и $P(X_3 \in E_2 | \bar{s}_3)$

$\lambda_1, \lambda_2 \backslash \bar{s}_s$	$\bar{s}_3 = (1, 1, 1)$	$\bar{s}_3 = (1, 2, 1)$	$\bar{s}_3 = (2, 1, 1)$	$\bar{s}_3 = (1, 1, 2)$
$\lambda_1 = 0,025$ $\lambda_2 = 0,05$	0,04648 0,48473 0,51527	0,06463 0,22445 0,77555	0,09296 0,48473 0,51527	0,06463 0,62559 0,37441
$\lambda_1 = 0,05$ $\lambda_2 = 0,05$	0,13889 0,65000 0,35000	0,11111 0,31250 0,68750	0,13889 0,65000 0,35000	0,11111 0,68750 0,31250
$\lambda_1 = 0,025$ $\lambda_2 = 0,0125$	0,31519 0,88125 0,11875	0,12926 0,39557 0,60443	0,15759 0,88125 0,11875	0,12926 0,86052 0,13948
$\lambda_1, \lambda_2 \backslash \bar{s}_s$	$\bar{s}_3 = (2, 2, 1)$	$\bar{s}_3 = (1, 2, 2)$	$\bar{s}_3 = (2, 1, 2)$	$\bar{s}_3 = (2, 2, 2)$
$\lambda_1 = 0,025$ $\lambda_2 = 0,05$	0,12926 0,22445 0,77555	0,15759 0,21347 0,78653	0,12926 0,62559 0,37441	0,31519 0,21347 0,78653
$\lambda_1 = 0,05$ $\lambda_2 = 0,05$	0,11111 0,31250 0,68750	0,13889 0,35000 0,65000	0,11111 0,68750 0,31250	0,13889 0,35000 0,65000
$\lambda_1 = 0,025$ $\lambda_2 = 0,0125$	0,06463 0,39557 0,60443	0,09296 0,67761 0,32239	0,06463 0,86052 0,13948	0,04648 0,67761 0,32239

лов $\bar{s}_3 = (s_1, s_2, s_3)$. В ячейках табл. 2 эти вероятности расположены в том же порядке сверху вниз.

II. Перейдем к нахождению условных вероятностей $P(X_4 \in B | \bar{s}_3)$, $P(S_4 = s | \bar{s}_3)$.

Имеем, что

$$\begin{aligned}
 P(X_4 \in d\mathbf{x} | \bar{s}_3) &= \int_E P(X_4 \in d\mathbf{x}, X_3 \in d\mathbf{y} | \bar{s}_3) = \\
 &= \int_E P(X_3 \in d\mathbf{y} | \bar{s}_3) P(X_4 \in d\mathbf{x} | X_3 = \mathbf{y}, \bar{s}_3) = \\
 &= \int_E P(X_3 \in d\mathbf{y} | \bar{s}_3) P(X_4 \in d\mathbf{x} | X_3 = \mathbf{y}).
 \end{aligned}$$

Таким образом,

$$\begin{aligned}
 P(X_4 \in d\mathbf{x} | \bar{s}_3) &= \int_E P(X_3 \in d\mathbf{y} | \bar{s}_3) P(X_4 \in d\mathbf{x} | X_3 = \mathbf{y}) = \\
 (18) \quad &= \int_E P(X_4 \in d\mathbf{x} | X_3 = \mathbf{y}) H_3(d\mathbf{y}).
 \end{aligned}$$

Таблица 3. Значения условных вероятностей $P(X_4 \in E_1 | \bar{s}_3)$, $P(X_4 \in E_2 | \bar{s}_3)$

$\lambda_1, \lambda_2 \backslash P$	$P(X_4 \in E_1 \bar{s}_3)$	$P(X_4 \in E_2 \bar{s}_3)$
$\lambda_1 = 0,025$ $\lambda_2 = 0,05$	0,53895	0,46105
$\lambda_1 = 0,05$ $\lambda_2 = 0,05$	0,62500	0,37500
$\lambda_1 = 0,025$ $\lambda_2 = 0,0125$	0,78677	0,21323

Выражение для плотности $p(X_4 = \mathbf{x} | \bar{s}_3)$ имеет вид:

$$(19) \quad \begin{aligned} p(X_4 = \mathbf{x} | \bar{s}_3) &= \int_E p(X_3 = \mathbf{y} | \bar{s}_3) p(X_4 = \mathbf{x} | X_3 = \mathbf{y}) d\mathbf{y} = \\ &= \int_E p(X_4 = \mathbf{x} | X_3 = \mathbf{y}) h_3(\mathbf{y}) d\mathbf{y}. \end{aligned}$$

В качестве примера, найдем выражение для $p(X_4 = \mathbf{x} | \bar{s}_3)$ в случае $\bar{s}_3 = (1, 2, 1)$:

$$\begin{aligned} p(X_4 = \mathbf{x} | \bar{s}_3) &= \\ &= \begin{cases} c_3 \lambda_2 p_1 e^{-\lambda_2 x} \left[\lambda_1 \left(\frac{\lambda_1 e^{-\lambda_1 x}}{2\lambda_1 + \lambda_2} \left(\frac{1}{2\lambda_1 + \lambda_2} - \frac{e^{-\lambda_1 x}}{3\lambda_1 + \lambda_2} \right) + \right. \right. \\ \left. \left. + \frac{\lambda_2 p_2}{\lambda_1 + 2\lambda_2} \left(\frac{1}{\lambda_1 + \lambda_2} - \frac{e^{-\lambda_1 x}}{2\lambda_1 + \lambda_2} \right) \right) + \right. \\ \left. + \lambda_2 \left(\left(\frac{\lambda_1}{2\lambda_1 + \lambda_2} + p_2 \right) \frac{1}{\lambda_1 + 2\lambda_2} - \frac{\lambda_2}{(\lambda_1 + 2\lambda_2)(\lambda_1 + 3\lambda_2)} \right) \right], & \mathbf{x} = 1x, \\ c_3 \lambda_2 p_1 e^{-\lambda_1 x} \left[\lambda_1 \left(\frac{\lambda_1^2}{(2\lambda_1 + \lambda_2)^2 (3\lambda_1 + \lambda_2)} + \frac{\lambda_2 p_1 p_2}{(\lambda_1 + 2\lambda_2)(2\lambda_1 + \lambda_2)} \right) + \right. \\ \left. + \lambda_2 e^{-\lambda_2 x} \left(\left(\frac{\lambda_1}{2\lambda_1 + \lambda_2} + p_2 \right) \frac{1}{\lambda_1 + 2\lambda_2} - \frac{\lambda_2 e^{-\lambda_2 x}}{(\lambda_1 + 2\lambda_2)(\lambda_1 + 3\lambda_2)} \right) \right], & \mathbf{x} = 2x. \end{cases} \end{aligned}$$

Значения условных вероятностей $P(X_4 \in E_1 | \bar{s}_3)$, $P(X_4 \in E_2 | \bar{s}_3)$ для вектора сигналов $\bar{s}_3 = (1, 2, 1)$ при значениях параметров I, II, III приведены в табл. 3.

Таблица 4. Значения условных вероятностей $P(S_4 = s | \bar{s}_3)$

$\lambda_1, \lambda_2 \backslash P$	$P(S_4 = 1 \bar{s}_3)$	$P(S_4 = 2 \bar{s}_3)$
$\lambda_1 = 0,025$ $\lambda_2 = 0,05$	0,42159	0,57841
$\lambda_1 = 0,05$ $\lambda_2 = 0,05$	0,54167	0,45833
$\lambda_1 = 0,025$ $\lambda_2 = 0,0125$	0,76997	0,23003

Далее,

$$\begin{aligned}
 P(S_4 = s | \bar{s}_3) &= \int_E P(S_4 = s, X_4 \in d\mathbf{x} | \bar{s}_3) = \\
 &= \int_E P(S_4 = s | X_4 = \mathbf{x}, \bar{s}_3) P(X_4 \in d\mathbf{x} | \bar{s}_3) = \\
 &= \int_E P(S_4 = s | X_4 = \mathbf{x}) P(X_4 \in d\mathbf{x} | \bar{s}_3) = \int_E R(s_4 | \mathbf{x}) P(X_4 \in d\mathbf{x} | \bar{s}_3).
 \end{aligned}$$

Следовательно,

$$(20) \quad P(S_4 = s | \bar{s}_3) = \int_E p(X_4 \mathbf{x} | \bar{s}_3) P(S_4 = s | X_4 = \mathbf{x}) d\mathbf{x}.$$

Условные вероятности $P(S_4 = s | \bar{s}_3)$ для вектора $\bar{s}_3 = (1, 2, 1)$ при значениях параметров I, II, III приведены в табл. 4.

III. Рассмотрим способы нахождения вероятности $P(\bar{S}^n = \bar{s}_n)$.

Вычисление вероятности $P(\bar{S}^n = \bar{s}_n)$ по формуле (15), основанное на использовании функций $F_n(\mathbf{B})(f_n(\mathbf{x}))$, называется прямым подходом [12, 13].

Существует также обратный подход [12, 13] к вычислению вероятности $P(\bar{S}^n = \bar{s}_n)$, основанный на использовании функций (обратные переменные) $B_k(\mathbf{x})$ [12, 13]:

$$(21) \quad B_k(\mathbf{x}) = P(S_{k+1} = s_{k+1}, S_n = s_n | X_k = \mathbf{x}), \quad k = \overline{1, n-1}, \quad \mathbf{x} \in E.$$

Найдем рекуррентное соотношение для функций $B_k(\mathbf{x})$.

$$\begin{aligned}
 B_k(\mathbf{x}) &= \int_E P(S_{k+1} = s_{k+1}, \dots, S_n = s_n, X_{k+1} \in d\mathbf{y} | X_k = \mathbf{x}) = \\
 &= \int_E P(S_{k+1} = s_{k+1}, \dots, S_n = s_n | X_k = \mathbf{x}, X_{k+1} = \mathbf{y}) P(X_{k+1} \in d\mathbf{y} | X_k = \mathbf{x}) =
 \end{aligned}$$

$$\begin{aligned}
&= \int_E P(S_{k+1} = s_{k+1}, \dots, S_n = s_n | X_{k+1} = \mathbf{y}) P(X_{k+1} \in d\mathbf{y} | X_k = \mathbf{x}) = \\
&= \int_E P(S_{k+1} = s_{k+1} | X_{k+1} = \mathbf{y}) P(S_{k+2} = s_{k+2}, \dots, S_n = s_n | S_{k+1} = \\
&\quad = s_{k+1}, X_{k+1} = \mathbf{y}) P(X_{k+1} \in d\mathbf{y} | X_k = \mathbf{x}) = \\
&= \int_E R(s_{k+1} | \mathbf{y}) P(S_{k+2} = s_{k+2}, \dots, S_n = s_n | X_{k+1} = \mathbf{y}) P(X_{k+1} \in d\mathbf{y} | X_k = \mathbf{x}) = \\
&\quad = \int_E R(s_{k+1} | \mathbf{y}) B_{k+1}(\mathbf{y}) P(X_{k+1} \in d\mathbf{y} | X_k = \mathbf{x}) = \\
&\quad = \int_E R(s_{k+1} | \mathbf{y}) B_{k+1}(\mathbf{y}) P(\mathbf{x}, d\mathbf{y}).
\end{aligned}$$

Следовательно,

$$(22) \quad B_k(\mathbf{x}) = \int_E B_{k+1}(\mathbf{y}) R(s_{k+1} | \mathbf{y}) P(\mathbf{x}, d\mathbf{y}), \quad k = \overline{1, n-2}.$$

Имеем, что

$$\begin{aligned}
B_{n-1}(\mathbf{x}) &= P(S_n = s_n | X_{n-1} = \mathbf{x}) = \int_E P(S_n = s_n, X_n \in d\mathbf{y} | X_{n-1} = \mathbf{x}) = \\
&= \int_E P(X_n \in d\mathbf{y} | X_{n-1} = \mathbf{x}) P(S_n = s_n | X_{n-1} = \mathbf{x}, X_n = \mathbf{y}) = \\
&= \int_E P(S_n = s_n | X_n = \mathbf{y}) P(X_n \in d\mathbf{y} | X_{n-1} = \mathbf{x}) = \int_E R(s_n | \mathbf{y}) P(\mathbf{x}, d\mathbf{y}).
\end{aligned}$$

Зная $B_{n-1}(\mathbf{x})$, используя рекуррентное соотношение (22), можно последовательно найти $B_{n-2}(\mathbf{x}), \dots, B_1(\mathbf{x})$.

Выражения для функций $B_2(\mathbf{x}), B_1(\mathbf{x})$ для вектора сигналов $\bar{s}_3 = (1, 2, 1)$ имеют вид:

$$B_2(\mathbf{x}) = \begin{cases} 1 - e^{-\lambda_1 x} (p_2 + \lambda_1 x), & \mathbf{x} = 1x, \\ e^{-\lambda_2 x} (p_1 + \lambda_2 x), & \mathbf{x} = 2x, \end{cases}$$

$$B_1(\mathbf{x}) = \begin{cases} \lambda_2 p_1 \left[\frac{1}{\lambda_1 + \lambda_2} - \frac{p_2}{2\lambda_1 + \lambda_2} - \frac{\lambda_1}{(2\lambda_1 + \lambda_2)^2} + \right. \\ \left. + \frac{p_1}{\lambda_1 + \lambda_2} + \frac{\lambda_2}{(\lambda_1 + \lambda_2)^2} - \frac{p_1}{\lambda_1 + 2\lambda_2} - \frac{\lambda_2}{(\lambda_1 + 2\lambda_2)^2} \right], & \mathbf{x} = 0, \\ 0, & \mathbf{x} = 1x, 2x. \end{cases}$$

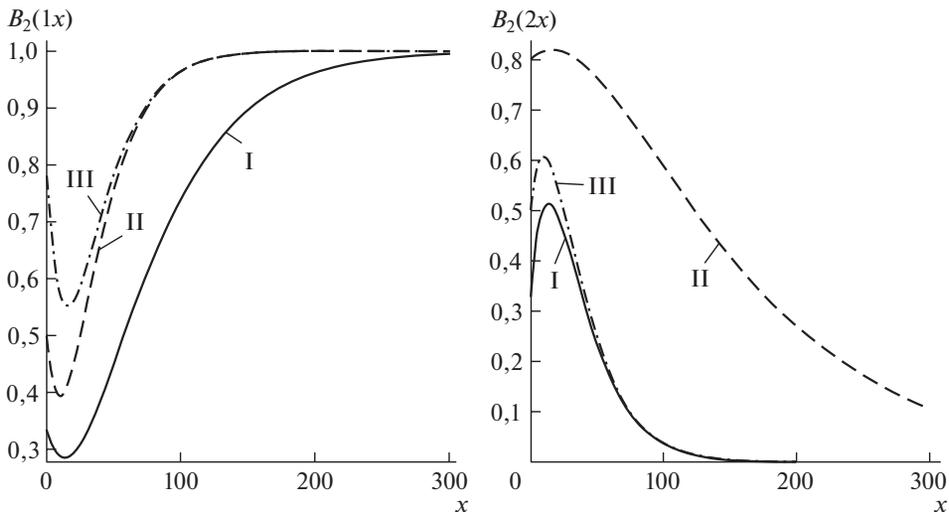


Рис. 3. Графики функций $B_2(1x)$, $B_2(2x)$.

Графики функций $B_2(1x)$, $B_2(2x)$ при значениях параметров I, II, III в случае вектора сигналов $\bar{s}_3 = (1, 2, 3)$ представлены на рис. 3.

Функция $B_1(\mathbf{x})$ позволяет найти вероятность $P(\bar{S}^n = \bar{s}_n)$.

$$\begin{aligned}
 P(\bar{S}^n = \bar{s}_n) &= \int_E P(S_1 = s_1, \dots, S_n = s_n, X_1 \in d\mathbf{x}) = \\
 &= \int_E P(X_1 \in d\mathbf{x}) P(S_1 = s_1, \dots, S_n = s_n | X_1 = \mathbf{x}) = \\
 &= \int_E P(S_1 = s_1, \dots, S_n = s_n | X_1 = \mathbf{x}) \pi(d\mathbf{x}) = \\
 &= \int_E P(S_1 = s_1 | X_1 = \mathbf{x}) P(S_2 = s_2, \dots, S_n = s_n | S_1 = s_1, X_1 = \mathbf{x}) \pi(d\mathbf{x}) = \\
 &= \int_E R(s_1 | \mathbf{x}) P(S_2 = s_2, \dots, S_n = s_n | X_1 = \mathbf{x}) \pi(d\mathbf{x}) = \\
 &= \int_E R(s_1 | \mathbf{x}) B_1(\mathbf{x}) \pi(d\mathbf{x}).
 \end{aligned}$$

Таким образом,

$$(23) \quad P(\bar{S}^n = \bar{s}_n) = \int_E R(s_1 | \mathbf{x}) B_1(\mathbf{x}) \pi(d\mathbf{x}).$$

Находя $P(\bar{S}^n = \bar{s}_n)$ по формуле (23) для вектора сигналов $\bar{s}_3 = (1, 2, 1)$, получаем выражение, совпадающее с (17).

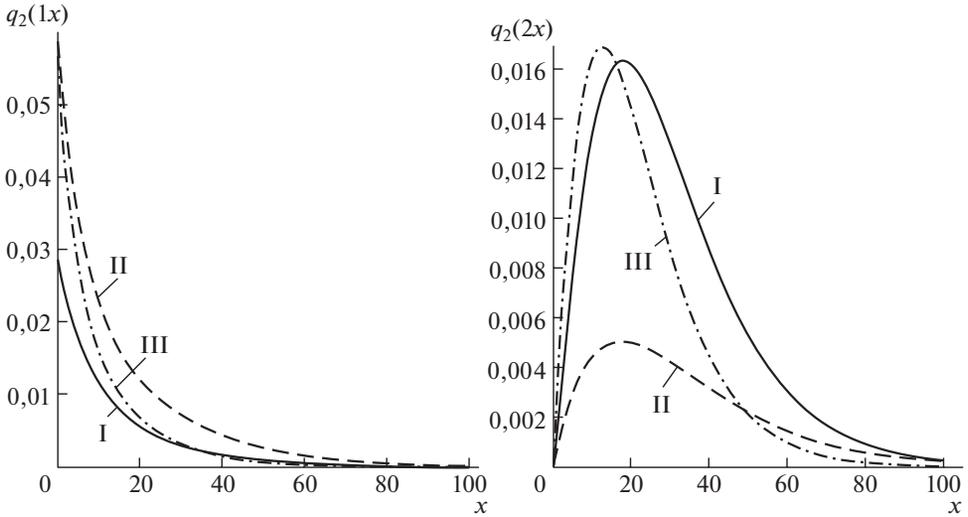


Рис. 4. Графики функций $q_2(1x)$, $q_2(2x)$.

Еще один способ нахождения вероятности $P(\bar{S}^n = \bar{s}_n)$ состоит в комбинировании прямого и обратного подходов [5, 12, 13]. Предположим, что для некоторого $1 \leq k \leq n - 1$ можно найти обе функции $F_k(B)(f_k(x))$ и $B_k(x)$. Тогда

$$\begin{aligned}
 & P(\bar{S}^n = \bar{s}_n, X_k \in d\mathbf{x}) = \\
 & = P(\bar{S}^k = \bar{s}_k, X_k \in d\mathbf{x})P(S_{k+1} = s_{k+1}, \dots, S_n = s_n | \bar{S}^k = \bar{s}_k, X_k = \mathbf{x}) = \\
 & = P(\bar{S}^k = \bar{s}_k, X_k \in d\mathbf{x})P(S_{k+1} = s_{k+1}, \dots, S_n = s_n | X_k = \mathbf{x}) = F_k(d\mathbf{x})B_k(\mathbf{x}).
 \end{aligned}$$

Следовательно,

$$(24) \quad P(\bar{S}^n = \bar{s}_n) = \int_E B_k(\mathbf{x})f_k(\mathbf{x})d\mathbf{x}.$$

IV. Использование скрытой марковской модели позволяет на основе полученного вектора сигналов \bar{s}_n прогнозировать состояния цепи Маркова, которые ненаблюдаемы [12, 13].

Предположим, что получен вектор $\bar{s}_n = (s_1, s_2, \dots, s_n)$ первых n сигналов и необходимо предсказать первые n состояний цепи Маркова, состояния которой ненаблюдаемы. Для этого, следуя [12, 13], при $k \leq n$ введем функции

$$(25) \quad Q_k(B) = P(X_k \in B | \bar{S}^n = \bar{s}_n), \quad B \in \mathfrak{F},$$

или в дифференциальной форме

$$(26) \quad Q_k(d\mathbf{x}) = P(X_k \in d\mathbf{x} | \bar{S}^n = \bar{s}_n), \quad \mathbf{x} \in E.$$

Таблица 5. Значения вероятностей $P(X_k = 0|\bar{s}_3)$, $P(X_k \in E_1|\bar{s}_3)$, $P(X_k \in E_2|\bar{s}_3)$

$\lambda_1, \lambda_2 \backslash k$	$k = 1$	$k = 2$	$k = 3$
$\lambda_1 = 0,025$ $\lambda_2 = 0,05$	1 0 0	0 0,35817 0,64183	0 0,22445 0,77555
$\lambda_1 = 0,05$ $\lambda_2 = 0,05$	1 0 0	0 0,50000 0,50000	0 0,31250 0,68750
$\lambda_1 = 0,025$ $\lambda_2 = 0,0125$	1 0 0	0 0,64183 0,35817	0 0,39557 0,60443

Имеем, что

$$Q_k(d\mathbf{x}) = P(X_k \in d\mathbf{x} | \bar{S}^n = \bar{s}_n) = \frac{P(\bar{S}^n = \bar{s}_n, X_k \in d\mathbf{x})}{P(\bar{S}^n = \bar{s}_n)} = \frac{F_k(d\mathbf{x})B_k(\mathbf{x})}{\int_E B_k(\mathbf{x})f_k(\mathbf{x})d\mathbf{x}}.$$

Следовательно, плотность $q_k(\mathbf{x})$ функции $Q_k(B)$ определяется формулой:

$$(27) \quad q_k(\mathbf{x}) = \frac{f_k(\mathbf{x})B_k(\mathbf{x})}{P(\bar{S}^n = \bar{s}_n)} = \frac{f_k(\mathbf{x})B_k(\mathbf{x})}{\int_E B_k(\mathbf{x})f_k(\mathbf{x})d\mathbf{x}} = c_n f_k(\mathbf{x})B_k(\mathbf{x}).$$

Графики функций $q_2(1x)$, $q_2(2x)$ для вектора сигналов $\bar{s}_3 = (1, 2, 1)$ при значениях параметров I, II, III представлены на рис. 4.

Знание функций $q_k(1x)$, $q_k(2x)$ позволяет найти прогноз состояний ненаблюдаемой цепи Маркова на k -м шаге, в том числе и значений непрерывных компонент, входящих в состав фазовых состояний системы.

В качестве примера, рассмотрим прогноз при $k = 1, 2, 3$ состояний ВЦМ суперпозиции двух независимых процессов восстановления при значениях параметров I, II, III в случае вектора $\bar{s}_3 = (1, 2, 1)$. В табл. 5 при $k = 1, 2, 3$ приведены вероятности $P(X_k = 0|\bar{s}_3)$, $P(X_k \in E_1|\bar{s}_3)$, $P(X_k \in E_2|\bar{s}_3)$, которые расположены в ячейках табл. 5 сверху вниз.

V. Следуя публикации [17], найдем $\lim_{n \rightarrow \infty} P(S_n = s | X_1 = \mathbf{x})$.

Имеем, что

$$\begin{aligned} P(S_n = s | X_1 = \mathbf{x}) &= \int_E P(S_n = s, X_n \in d\mathbf{y} | X_1 = \mathbf{x}) = \\ &= \int_E P(X_n \in d\mathbf{y} | X_1 = \mathbf{x})P(S_n = s | X_n = \mathbf{y}, X_1 = \mathbf{x}) = \\ &= \int_E P(S_n = s | X_n = \mathbf{y})P(X_n \in d\mathbf{y} | X_1 = \mathbf{x}) = \int_E R(s | \mathbf{y})P^n(\mathbf{x}, d\mathbf{y}). \end{aligned}$$

Таблица 6. Предельные вероятности $\lim_{n \rightarrow \infty} P(S_n = 1 | X_1 = \mathbf{x})$,
 $\lim_{n \rightarrow \infty} P(S_n = 2 | X_1 = \mathbf{x})$

λ_1, λ_2	$\lim_{n \rightarrow \infty} P(S_n = 1 X_1 = \mathbf{x})$	$\lim_{n \rightarrow \infty} P(S_n = 2 X_1 = \mathbf{x})$
$\lambda_1 = 0,025$ $\lambda_2 = 0,05$	0,33333	0,66667
$\lambda_1 = 0,05$ $\lambda_2 = 0,05$	0,50000	0,50000
$\lambda_1 = 0,025$ $\lambda_2 = 0,0125$	0,66667	0,33333

Следовательно,

$$\lim_{n \rightarrow \infty} P(S_n = s | X_1 = \mathbf{x}) = \lim_{n \rightarrow \infty} \int_E R(s | \mathbf{y}) P^n(\mathbf{x}, d\mathbf{y}) = \int_E R(s | \mathbf{y}) \rho(d\mathbf{y}),$$

где $\rho(d\mathbf{y})$ – стационарное распределение ВЦМ $\{X_n, n = 1, 2, \dots\}$.

Таким образом,

$$(28) \quad \lim_{n \rightarrow \infty} P(S_n = s | X_1 = \mathbf{x}) = \int_E R(s | \mathbf{y}) \rho(d\mathbf{y}), \quad \text{для любого } \mathbf{x} \in E.$$

Найдем эти предельные вероятности для ВЦМ суперпозиции двух процессов восстановления. Как отмечено выше, ВЦМ суперпозиции имеет стационарное распределение, определяемое формулой (2).

Применяя (2), (28) и табл. 1, получаем, что

$$\begin{aligned} \lim_{n \rightarrow \infty} P(S_n = 1 | X_1 = \mathbf{x}) &= \int_{E_1} R(1 | 1y) \rho(1dy) + \int_{E_2} R(1 | 2y) \rho(2dy) = \\ &= \rho_0 \left(\int_0^\infty G_1(y) \bar{G}_2(y) dy + \int_0^\infty \bar{G}_1(y) G_2(y) dy \right) = \\ &= \rho_0 \int_0^\infty \bar{G}_2(y) dy = \rho_0 M\alpha_2 = \frac{M\alpha_2}{M\alpha_1 + M\alpha_2}, \end{aligned}$$

$$\begin{aligned} \lim_{n \rightarrow \infty} P(S_n = 2 | X_1 = \mathbf{x}) &= \int_{E_1} R(2 | 1y) \rho(1dy) + \int_{E_2} R(2 | 2y) \rho(2dy) = \\ &= \rho_0 M\alpha_1 = \frac{M\alpha_1}{M\alpha_1 + M\alpha_2}, \end{aligned}$$

а в случае экспоненциального распределения случайных величин α_1 и α_2 с параметрами λ_1 и λ_2 :

$$\lim_{n \rightarrow \infty} P(S_n = 1 | X_1 = \mathbf{x}) = \frac{\lambda_1}{\lambda_1 + \lambda_2}, \quad \lim_{n \rightarrow \infty} P(S_n = 2 | X_1 = \mathbf{x}) = \frac{\lambda_2}{\lambda_1 + \lambda_2}.$$

В табл. 6 приведены предельные вероятности для значений параметров I, II, III.

Использование алгоритмов теории скрытых марковских моделей [5, 12, 13] позволяет находить оценки и других характеристик ненаблюдаемой модели и сигналов на основе полученного вектора сигналов.

4. Заключение

Большое число систем различного назначения допускает построение полумарковской модели. При построении полумарковской модели приходится строить достаточно сложное фазовое пространство состояний, отражающих физические состояния системы и обеспечивающих корректность построения полумарковской модели. Важной составной частью полумарковской модели является вложенная цепь Маркова, отвечающая за переходы между состояниями системы. Фазовое пространство состояний полумарковской модели совпадает с фазовым пространством состояний вложенной цепи Маркова.

В процессе функционирования системы, для которой построена полумарковская модель, в ряде случаев при изменении состояний системы не удается получить всю информацию, содержащуюся в кодах фазовых состояний полумарковской модели, а удается получить только некоторую информацию (сигнал), связанную с фазовыми состояниями. В этом случае фазовые состояния вложенной цепи Маркова (полумарковской модели) можно считать скрытыми, поэтому возникает задача оценки характеристик вложенной цепи Маркова, полумарковской модели и сигналов на основе полученного вектора сигналов. Решить эту задачу позволяют скрытые марковские модели.

В данной статье на примере суперпозиции двух независимых процессов восстановления, построенной в работах В.С. Королюка и А.Ф. Турбина, рассматривается подход к построению скрытой марковской модели на основе полумарковского процесса с фазовым пространством состояний общего вида. Обобщая на случай скрытой цепи Маркова с общим фазовым пространством состояний результаты, известные для скрытых цепей Маркова с конечным множеством состояний, проводится оценка характеристик вложенной цепи Маркова суперпозиции и сигналов на основе полученного вектора сигналов. Для ВЦМ находятся условные вероятности состояний на $2 \leq k \leq n$, $(n + 1)$ -м шагах на основе полученного вектора сигналов размера n . Для вектора сигналов получены условные вероятности значений вектора сигналов на $(n + 1)$ -м шаге, безусловные вероятности вектора сигналов и финальные вероятности для значений вектора сигналов.

Рассматриваемый подход может быть использован для анализа функционирования систем различного назначения, допускающих построение полумарковской модели, в том числе и в случае конечного фазового пространства состояний. При реализации рассматриваемого подхода можно применять математические пакеты, используя при этом рекуррентные формулы.

Эффективным методом решения проблемы размерности моделей является алгоритм стационарного фазового укрупнения полумарковских систем, разработанный В.С. Королюком и А.Ф. Турбиным, который можно использовать при построении скрытой марковской модели.

В дальнейшем предполагается использовать рассматриваемый подход для анализа функционирования систем массового обслуживания, контроля, обслуживания технических систем и анализа их надежности.

СПИСОК ЛИТЕРАТУРЫ

1. *Limnios N., Oprisan G.* Semi-Markov Processes and Reliability. N.Y.: Springer Science+Business Media, 2001. <https://doi.org/10.1007/978-1-4612-0161-8>
2. *Grabski F.* Semi-Markov Processes: Applications in System Reliability and Maintenance. Elsevier Science, 2014.
3. *Obzherin Yu.E., Boyko E.G.* Semi-Markov Models: Control of Restorable Systems with Latent Failures. London: Elsevier Academic Press, 2015.
4. *Jansen J., Limnios N.* (Eds.) Semi-Markov Models and Applications. Netherlands: Kluwer Academic Publishers, 1999. <https://doi.org/10.1007/978-1-4613-3288-6>
5. *Kobayashi H., Mark B., Turin W.* Probability, Random Processes, and Statistical Analysis: Applications to Communications. Signal Processing, Queueing Theory and Mathematical Finance. Cambridge: Cambridge University Press, 2011. <https://doi.org/10.1017/CBO9780511977770>
6. *Obzherin Y.E., Sidorov S.M., Nikitin M.M.* Reliability of the information system with intermediate storage devices // CCIS. 2018. V. 919. P. 432–444. http://doi-org-443.webvpn.fjmu.edu.cn/10.1007/978-3-319-99447-5_37
7. *Limnios N., Nikulin M.* Recent Advances in Reliability Theory: Methodology, Practice, and Inference, eds. N. Limnios, M. Nikulin. New York: Springer Science+Business Media, 2000.
8. *Руденко Ю.Н., Ушаков И.А.* Надежность систем энергетики, 2-е изд. Новосибирск: Наука, 1989.
9. *Обжерин Ю.Е., Песчанский А.И.* Об однолинейной системе обслуживания с потерями и абсолютным приоритетом // АиТ. 1990. № 10. С. 107–115.
Obzherin Yu.E., Peschanskii A.I. On One-line Service System with Losses and Absolute Priority // Autom. Remote Control. 1990. V. 51. No. 10. P. 1393–1400.
10. *Песчанский А.И.* Стационарные характеристики ненадежной многоканальной системы обслуживания с потерями и временным резервом // АиТ. 2019. № 4. С. 70–92. <https://doi.org/10.1134/S0005231019040044>
Peschansky A.I. Stationary Characteristics of an Unreliable Multi-Server Queueing System with Losses and Time Redundancy // Autom. Remote Control. 2019. V. 80. No. 4. P. 648–665. <https://doi.org/10.1134/S0005117919040040>
11. *Janssen J., Manca R.* Semi-Markov Risk Models for Finance, Insurance and Reliability. N.Y.: Springer Science & Business Media, 2007.
12. *Ross S.M.* Introduction to Probability Models, Ninth Edition. USA: Elsevier Academic Press, 2006.
13. *Rabiner L.R.* A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition // Proc. IEEE 77. 1989. V. 189. No. 2. P. 257–286. <https://doi.org/10.1109/5.18626>
14. *Cappe' O., Moulines E., Ryde'n T.* Inference in Hidden Markov Models. N.Y.: Springer Science+Business Media, 2005.
15. *Obzherin Y.E., Sidorov S.M., Nikitin M.M.* Hidden Markov Model of Information System with Component-Wise Storage Devices // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), LNCS, 2019. V. 11965. P. 354–364. https://doi.org/10.1007/978-3-030-36614-8_27.

16. *Yu S.-Z.* Hidden Semi-Markov Models: Theory, Algorithms and Applications. Elsevier Science, 2015.
17. *Barbu V.S., Limnios N.* Semi-Markov Chains and Hidden Semi-Markov Models toward Applications: Their Use in Reliability and DNA Analysis. N.Y.: Springer, 2008.
18. *Elliott R., Limnios N., Swishchuk A.* Filtering hidden semi-Markov chains // Stat. Probab. Lett., 2013. V. 83. P. 2007–2014. <https://doi.org/10.1016/j.spl.2013.05.007>
19. *Van der Hoek J., Elliott R.* Introduction to Hidden Semi-Markov Models. Cambridge: Cambridge University Press, 2018.
20. *Королюк В.С., Турбин А.Ф.* Полумарковские процессы и их приложения. Киев: Наук. думка, 1976.
21. *Королюк В.С., Турбин А.Ф.* Процессы марковского восстановления в задачах надежности систем. Киев: Наук. думка, 1982.
22. *Королюк В.С.* Стохастические модели систем. Киев: Наук. думка, 1989.
23. *Korolyuk V.S., Korolyuk V.V.* Stochastic Models of Systems. Dordrecht: Springer Science+Business Media, 1999.
24. *Korolyuk V.S., Limnios N.* Stochastic Systems in Merging Phase Space. World Scientific, Imperial Coledge Press, 2005.
25. *Корлат А.Н., Кузнецов В.Н., Новиков М.М., Турбин А.Ф.* Полумарковские модели восстанавливаемых систем и систем массового обслуживания. Кишинев: Штиинца, 1991.
26. *Obzherin Yu.E., Sidorov S.M.* Semi-Markov Model and Phase-Merging Scheme of a Multi-Component System with the Group Instantly Replenished Time Reserve // Int. J. of Reliability, Quality and Safety Engineering, 2019. V. 26. No. 3. Art. No. 1950014. <https://doi.org/10.1142/S0218539319500141>

Статья представлена к публикации членом редколлегии А.В. Назиньм.

Поступила в редакцию 20.06.2019

После доработки 13.01.2021

Принята к публикации 15.01.2021

Робастное, адаптивное и сетевое управление

© 2021 г. В.А. АЛЕКСАНДРОВ, канд. техн. наук (va.alexandrov@yandex.ru)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва)

ОПТИМИЗАЦИЯ РАЗМЕЩЕНИЯ ПОЛЮСОВ В ОДНОМЕРНОЙ СИСТЕМЕ УПРАВЛЕНИЯ

Рассматривается задача синтеза регулятора для линейного стационарного объекта управления. Находится линейный регулятор, минимизирующий некоторый критерий качества системы, например соотношение выхода объекта и внешнего возмущения как H_∞ норму соответствующей функции, при условии соблюдения требований к качеству системы: радиус запасов устойчивости, соотношение сигнала управления и помехи измерения выхода объекта, коэффициент демпфирования, быстродействие. Рассмотрены известные методы синтеза регулятора из пакета MATLAB Robust Control Toolbox для решения такой задачи. Предложен метод синтеза регулятора, использующий корни характеристического полинома замкнутой системы в качестве варьируемых переменных стандартной процедуры оптимизации и стандартную процедуру размещения полюсов для получения значений коэффициентов регулятора.

Ключевые слова: линейная система управления, робастная система, синтез регулятора, радиус запасов устойчивости, размещение полюсов, H_∞ оптимизация, глобальная оптимизация.

DOI: 10.31857/S0005231021060040

1. Введение

Существует много методов синтеза регулятора для линейного стационарного объекта управления [1–4]. Методы, разработанные для многомерных систем, применимы и для рассматриваемых объектов с одним управляющим входом и одним управляемым выходом. Одним из универсальных подходов к синтезу регулятора с учетом нескольких целей и ограничений стал метод H_∞ оптимизации [5, 6]. Так, например, классическое требование к запасам устойчивости в этом методе выражается как ограничение или минимизация H_∞ нормы функции чувствительности. Важно, что метод позволяет оптимизировать систему одновременно еще по нескольким H_∞ нормам функций: чувствительности к помехе измерений, передаточной функции замкнутой системы относительно возмущения и другим [3, 5], а также учитывать допустимую область полюсов замкнутой системы [6]. При этом регулятор, получаемый стандартным методом H_∞ оптимизации, который базируется на решении системы линейных матричных неравенств, сформированных из обобщенной

оптимизируемой системы, имеет порядок больший или равный порядку модели объекта управления. Развитием метода H_∞ оптимизации, направленным на простоту практического использования, стал подход к синтезу регулятора с заданной структурой [7–9], где коэффициенты регулятора находятся как решение задачи негладкой оптимизации. При этом можно находить регулятор заданного низкого порядка. Следует учитывать, что результат минимизации целевой функции при выборе структуры регулятора низкого порядка будет заведомо не лучше, чем при поиске регулятора стандартным методом H_∞ оптимизации без ограничений структуры. И, кроме того, для некоторых объектов может не существовать регулятора с порядком меньше порядка объекта более чем на единицу, обеспечивающего хотя бы устойчивость системы.

Практическая задача поиска регулятора с заданной структурой делает актуальным развитие и других подходов, таких как метод D-разбиения [10]. В публикации [11] решается задача построения регулятора методом D-разбиения с учетом ряда инженерных требований.

Необходимо заметить, что в большинстве методов используются некоторые весовые матрицы, весовые передаточные функции или другие параметры, характеризующие качество системы, выбор которых обычно не формализован и дает свободу проектировщику системы. Но если процедура синтеза регулятора работает в системе самонастройки, когда модель объекта получена в результате процедуры идентификации, и участие проектировщика не предусмотрено, то важно формулировать критерии синтеза так, чтобы они позволяли получить практически работоспособную систему, удовлетворяющую реальным требованиям к качеству. Так, в [12–15] решается задача синтеза регулятора по инженерным критериям качества с использованием метода H_∞ оптимизации. В [16, 17] для решения задачи синтеза регулятора, удовлетворяющего требованиям к точности, быстродействию и радиусу запасов устойчивости, применяется метод размещения полюсов и предложен способ формирования желаемого характеристического полинома замкнутой системы в соответствии с этими требованиями.

В настоящей статье рассматривается задача подавления внешних возмущений, т.е. минимизации H_∞ нормы функции чувствительности системы к возмущению, при соблюдении ряда ограничений, обуславливающих робастность системы и качество переходных процессов. Предлагается новый подход к решению этой задачи методом размещения полюсов, где желаемый характеристический полином ищется с помощью процедуры глобальной оптимизации. При этом корни характеристического полинома замкнутой системы, а не коэффициенты регулятора, являются варьируемыми переменными. Метод размещения полюсов позволяет найти регулятор, порядок которого на единицу меньше порядка модели объекта. Возможно построение регулятора более высокого порядка, но найти регулятор низкого порядка по сравнению с порядком модели объекта этот метод не позволяет. В случае заданной структуры регулятора можно редуцировать модель объекта до соответствующего порядка.

Кроме того, для сравнения рассматриваются реализации решения поставленной задачи с использованием стандартных процедур синтеза регулято-

ра методами H_∞ оптимизации из пакета MATLAB Robust Control Toolbox: `hinfsyn`, `h2hinfsyn` и `systune`.

2. Постановка задачи

Рассмотрим управляемый объект со скалярными входами и выходом, описываемый линейным дифференциальным уравнением

$$(1) \quad \begin{aligned} y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_1\dot{y} + a_0y = \\ = b_{n-1}u^{(n-1)} + \dots + b_1\dot{u} + b_0u + g_{n-1}f^{(n-1)} + \dots + g_1\dot{f} + g_0f, \end{aligned}$$

где правый верхний индекс обозначает порядок производной по непрерывному времени t , $y(t) \in \mathbb{R}$ — управляемый и измеряемый выход объекта, $u(t) \in \mathbb{R}$ — вход управления, $f(t) \in \mathbb{R}$ — неизмеряемое неизвестное возмущение, ограниченное по модулю, $n \in \mathbb{N}$ — известный порядок объекта, a_i, b_i, g_i ($i = 0, \dots, n-1$) — известные вещественные коэффициенты. Предполагается, что все функции удовлетворяют условиям применения преобразования Лапласа и начальные условия равны нулю. Тогда вместо уравнения (1) будем использовать передаточные функции относительно управления и возмущения соответственно

$$(2) \quad P(s) = \frac{b(s)}{a(s)} = \frac{b_{n-1}s^{n-1} + \dots + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_0},$$

$$(3) \quad P_f(s) = \frac{g(s)}{a(s)} = \frac{g_{n-1}s^{n-1} + \dots + g_0}{s^n + a_{n-1}s^{n-1} + \dots + a_0},$$

где s — переменная преобразования Лапласа. Принимая $s = j\omega$, где частота $\omega \in [0, \infty)$, будем использовать частотные передаточные функции.

Некоторые коэффициенты a_i, b_i, g_i ($i = 0, \dots, n-1$) могут быть равны нулю, поэтому степени полиномов $b(s)$ и $g(s)$ могут быть меньше $n-1$. Возможен случай, когда считается, что возмущение приложено вместе с управлением, и тогда $g(s) = b(s)$.

Рассмотрим систему, где объект управления (1) замкнут линейным регулятором с передаточной функцией

$$(4) \quad C(s) = \frac{d(s)}{c(s)} = \frac{d_ms^m + d_{m-1}s^{m-1} + \dots + d_0}{c_ms^m + c_{m-1}s^{m-1} + \dots + c_0},$$

где m — порядок регулятора, d_i, c_i ($i = 0, \dots, m$) — вещественные коэффициенты регулятора. Известно, что для любого управляемого объекта (1) существует стабилизирующий регулятор степени $m = n-1$, в то время как для $m < n-1$ такого регулятора может не существовать. Поэтому будем принимать $m \geq n-1$ в зависимости от метода синтеза регулятора. Задача нахождения регулятора порядка $m < n-1$ в этой статье не рассматривается.

Структура системы управления представлена на рис. 1. Задача слежения в этой статье не рассматривается, поэтому задающее воздействие r , присутствующее на рис. 1, будем считать равным нулю. Кроме того, в структуру

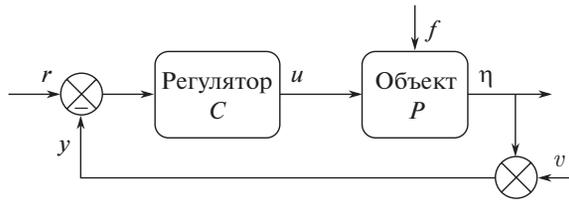


Рис. 1. Структура системы управления.

добавлена помеха измерения ν , свойства которой не оговариваются, так как действие помехи измерения в статье не рассматривается, но передаточная функция относительно этого входа будет использоваться. При этом измеряемый выход объекта y принимается равным сумме значений реального выхода объекта η и помехи измерения ν .

Задача состоит в том, чтобы найти регулятор (4), обеспечивающий устойчивость и некоторое желаемое качество системы (2), (3), (4). Известно, что система устойчива тогда и только тогда, когда отрицательны вещественные части всех корней характеристического полинома замкнутой системы:

$$(5) \quad a(s)c(s) + b(s)d(s).$$

Необходимо заметить, что уравнение (1) обычно является не абсолютно точной моделью реального объекта управления, а упрощенной линеаризованной моделью, причем коэффициенты определены неточно или могут меняться со временем, т.е. реальная динамика объекта может быть нелинейная и нестационарная. Поэтому при использовании модели (1) для синтеза регулятора требование устойчивости системы необходимо сопровождать требованием робастности системы, т.е. сохранения устойчивости при отличии реальной динамики объекта от исходной линейной модели. Один из критериев робастности — это запасы устойчивости. Широко используемые запасы по фазе и по амплитуде служат индикаторами удаленности амплитудно-фазовой частотной характеристики (АФЧХ) разомкнутой системы $P(j\omega)C(j\omega)$ от критической точки $(-1, j0)$ в соответствии с критерием устойчивости Найквиста. Но так как эти запасы учитывают значения АФЧХ только на двух частотах, то вместо них все чаще используется значение радиуса запасов устойчивости [2, 18, 19], который равен минимальному расстоянию АФЧХ разомкнутой системы от точки $(-1, j0)$ для всех частот, т.е. минимальному значению модуля функции обратной разности:

$$(6) \quad R = \inf_{\omega} |1 + P(j\omega)C(j\omega)|$$

или обратному значению H_{∞} нормы функции чувствительности $S(j\omega)$ [2, 19]:

$$(7) \quad S(j\omega) = \frac{1}{1 + P(j\omega)C(j\omega)},$$

$$R = \frac{1}{\|S(j\omega)\|_{\infty}}.$$

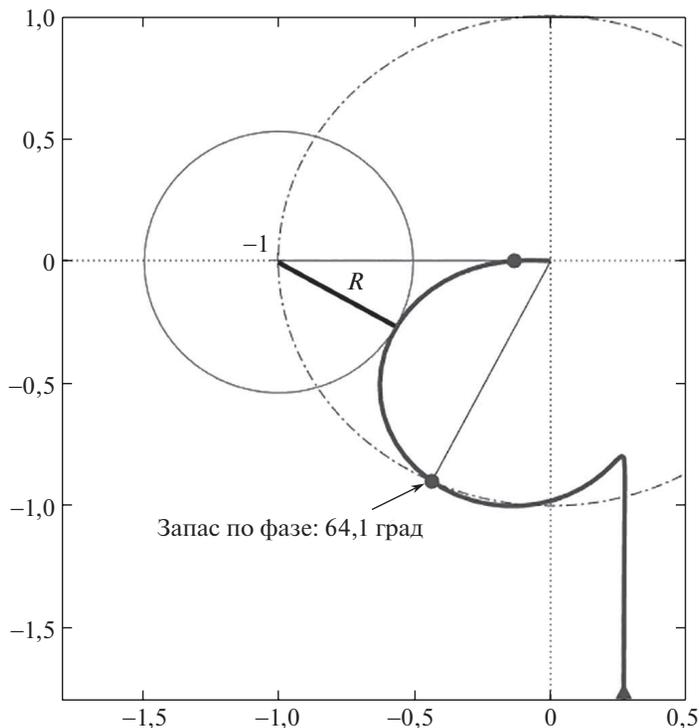


Рис. 2. Годограф Найквиста разомкнутой системы и радиус запасов устойчивости.

Известно, что радиус запасов устойчивости не является достаточным критерием робастности системы [19, 20]. В [2] отмечено, что для модели объекта, представляющей сумму $P(s) + \Delta(s)$ исходной и неизвестной дополнительной динамики, и для регулятора, обеспечивающего устойчивость исходной системы, система с дополнительной динамикой будет устойчива, если соблюдается ограничение

$$(8) \quad |\Delta(j\omega)| < \left| \frac{1 + P(j\omega)C(j\omega)}{C(j\omega)} \right|.$$

В правой части неравенства (8) записан модуль функции, обратной к функции чувствительности к помехе, связывающей помеху измерения, добавленную к выходу объекта, и сигнал управления

$$(9) \quad C(j\omega)S(j\omega) = \frac{C(j\omega)}{1 + P(j\omega)C(j\omega)},$$

т.е. значение $\|C(j\omega)S(j\omega)\|_\infty = \sup_\omega |C(j\omega)S(j\omega)|$ также является критерием робастности системы, и чем оно меньше, тем большее значение $|\Delta(j\omega)|$ не нарушит устойчивость системы. Кроме того, ограничение значения сигнала управления, возникающего при наличии помехи измерения и описываемого функцией чувствительности к помехе (9), необходимо для практической реализации системы, чтобы помехи не приводили к скачкам управления до предельных значений.

Таким образом, максимально допустимые значения $\|S(j\omega)\|_\infty$ и $\|C(j\omega)S(j\omega)\|_\infty$ должны задаваться при синтезе регулятора как строгие ограничения, которые не должны превышать. Значение радиуса запасов устойчивости находится в диапазоне $0 < R \leq 1$, и требуемое значение R^* обычно выбирается от 0,5 до 0,7, что соответствует максимальному значению H_∞ нормы функции чувствительности в диапазоне $1,43 \leq \|S(j\omega)\|_\infty \leq 2$. Необходимо учитывать, что, например, для неминимально-фазовых объектов может существовать максимально возможное значение радиуса запасов устойчивости меньше единицы, и тогда требуемое значение R^* должно выбираться в пределах достижимого. Выбор требуемого значения $N^* H_\infty$ нормы функции чувствительности к помехе $\|C(j\omega)S(j\omega)\|_\infty$ сложнее, так как зависит от соотношения масштабов измеряемого выхода и сигнала управления, а также их предельных значений и уровня помех.

Еще одно практическое ограничение — коэффициент демпфирования, который определяется соотношением вещественной и мнимой частей корней характеристического полинома замкнутой системы. Его значение может быть $0 < \zeta \leq 1$. Малая величина коэффициента демпфирования приводит к колебательному характеру переходных процессов. Поэтому стандартное требование при синтезе регулятора — это обеспечение коэффициента демпфирования в пределах $0,7 \leq \zeta \leq 1$. Кроме того, желательно ограничивать и близость к нулю вещественных частей корней характеристического полинома, так как их слишком малое по модулю значение может приводить к недопустимо большому времени переходных процессов.

В рамках этих, а возможно, и других ограничений можно оптимизировать остальные показатели качества. Одним из наиболее важных показателей качества системы управления является подавление внешних возмущений [21]. Передаточная функция замкнутой системы (2), (3), (4) относительно возмущения имеет вид

$$(10) \quad G_{yf}(s) = \frac{g(s)c(s)}{a(s)c(s) + b(s)d(s)},$$

и минимизация H_∞ нормы этой функции

$$\|G_{yf}(j\omega)\|_\infty = \sup_\omega |G_{yf}(j\omega)|$$

часто используется как цель оптимизации системы при синтезе регулятора. Заметим, что если возмущение приложено вместе с управлением, т.е. $g(s) = b(s)$, то

$$(11) \quad G_{yf}(j\omega) = \frac{P(j\omega)}{1 + P(j\omega)C(j\omega)} = P(j\omega)S(j\omega),$$

что представляет собой функцию чувствительности к возмущению [2].

Итак, задачу синтеза регулятора можно сформулировать так.

Задача. Найти стабилизирующий регулятор (4) такой, что в системе (2), (3), (4):

- 1) радиус запасов устойчивости (6) не меньше требуемого значения R^* ;
- 2) H_∞ норма функции чувствительности к помехе (9) не больше требуемого значения N^* ;
- 3) коэффициент демпфирования не меньше требуемого значения $\zeta_{\min} > 0$;
- 4) вещественные части корней характеристического полинома не превышают требуемого значения $(-\beta^*) < 0$;
- 5) H_∞ норма функции чувствительности к возмущению (10) будет минимальна при соблюдении перечисленных выше условий.

Следует внимательно подходить к выбору значений ограничений, так как может не существовать удовлетворяющего им решения. В этом случае ограничения должны быть пересмотрены.

3. Стандартные методы синтеза робастных регуляторов

3.1. H_∞ оптимальный регулятор

Стандартная процедура синтеза H_∞ оптимального регулятора (функция `hinfsyn` из пакета MATLAB Robust Control Toolbox) не решает поставленную в настоящей статье задачу полностью, так как в ней не предусмотрено введение ограничений на корни характеристического полинома. В процедурах, рассмотренных далее, такая возможность есть, но для сравнения рассмотрим и эту базовую процедуру, решая задачу без ограничений коэффициента демпфирования и близости к нулю вещественных частей корней характеристического полинома. Для процедуры синтеза регулятора нужно сформировать обобщенный объект управления [3]:

$$(12) \quad \begin{aligned} \dot{x}(t) &= Ax(t) + B_1w(t) + B_2u(t), \\ z(t) &= C_1x(t) + D_{11}w(t) + D_{12}u(t), \\ y(t) &= C_2x(t) + D_{21}w(t) + D_{22}u(t), \end{aligned}$$

где $x(t)$ — вектор состояния размерности n , $w(t)$ и $z(t)$ — векторы входов и выходов обобщенной системы, $y(t)$ и $u(t)$ — выход и вход объекта (1), A , B_1 , B_2 , C_1 , C_2 , D_{11} , D_{12} , D_{21} , D_{22} — матрицы и векторы коэффициентов соответствующих размерностей. Векторы входов и выходов обобщенной системы выбираются так, чтобы минимизация H_∞ нормы передаточной матрицы замкнутой системы, связывающей эти входы и выходы, решала поставленную задачу. Кроме того, можно находить субоптимальное решение, когда ищется регулятор, обеспечивающий значение H_∞ нормы передаточной матрицы замкнутой системы меньше заданного значения γ . В опциях процедуры можно выбирать метод решения: 2-Риккати формула или метод линейных матричных неравенств.

В [2] предлагается принять векторы выходов и входов обобщенной системы как

$$(13) \quad z = \begin{bmatrix} y \\ -u \end{bmatrix}, \quad w = \begin{bmatrix} f \\ \nu \end{bmatrix}.$$

Тогда процедура будет минимизировать или ограничивать H_∞ норму системы

$$(14) \quad z = \begin{bmatrix} G_{yf} & S \\ CG_{yf} & CS \end{bmatrix} w.$$

При этом в поставленной задаче нет требований к $\|CG_{yf}\|_\infty$, которая появилась в этой матрице, поэтому нужно будет проверять, что этот элемент не повлиял на полученное решение. Вторая проблема состоит в том, что процедура H_∞ оптимизации находит решение, которое обеспечивает значение всех составляющих передаточной матрицы меньше заданного значения γ . Для учета разных требований к передаточным функциям, связывающим разные входы и выходы, можно использовать [2, 3] весовые передаточные функции. Для решения поставленной Задачи будем использовать просто весовые коэффициенты W_1 и W_2 , тогда формируя векторы выходов и входов обобщенной системы как

$$(15) \quad z = \begin{bmatrix} y \\ -W_2 u \end{bmatrix}, \quad w = \begin{bmatrix} f/W_1 \\ \nu \end{bmatrix},$$

получим

$$(16) \quad z = \begin{bmatrix} W_1 G_{yf} & S \\ W_1 W_2 C G_{yf} & W_2 C S \end{bmatrix} w.$$

Принимая

$$(17) \quad \gamma = \frac{1}{R^*}, \quad W_2 = \frac{\gamma}{N^*}$$

и находя максимальное значение W_1 , для которого существует решение процедуры синтеза H_∞ оптимального регулятора, можно найти регулятор, обеспечивающий минимальное значение $\|CG_{yf}\|_\infty$ для заданных ограничений $\|S\|_\infty$ и $\|CS\|_\infty$. Если в результате получится неприемлемое значение $\|G_{yf}\|_\infty$, то необходимо пересмотреть требуемые значения R^* и N^* , т.е. изменить целевое значение γ и весового коэффициента W_2 . Надо также учитывать, что если значения $\|S\|_\infty$, $\|W_1 G_{yf}\|_\infty$ и $\|W_2 C S\|_\infty$ получились заметно меньше γ , а $\|W_1 W_2 C G_{yf}\|_\infty$ близка к заданному γ , то можно повторить процедуру, задавая большее значение γ , так как значение $\|W_1 W_2 C G_{yf}\|_\infty$ в поставленной задаче не ограничено.

3.2. H_∞ оптимальный регулятор с ограничением области размещения полюсов

Функция `h2hinfosyn` из пакета MATLAB Robust Control Toolbox отличается от `hinfosyn`, во-первых, тем, что можно задавать допустимую область размещения полюсов передаточной функции замкнутой системы [6]. Второе отличие состоит в том, что можно определять два вектора выходов обобщенной системы z_∞ и z_2 . При этом формируются разные передаточные матрицы

замкнутой системы, для одной из которых минимизируется или ограничивается H_∞ норма, а для другой — H_2 норма, что повышает гибкость формирования целей и ограничений синтеза регулятора. Решение основано на методе линейных матричных неравенств. Так как задача сформулирована как требования к H_∞ нормам, то в этой статье не рассматривается возможность использования комбинированного H_2/H_∞ подхода и формирование обобщенной системы может быть таким же, как описано в подразделе 3.1.

Возможность задавать область допустимого размещения полюсов позволяет решать задачу со всеми ограничениями. Ограничения на область размещения полюсов формируются как линейное матричное неравенство вида $L + zQ + \bar{z}Q^T < 0$, где матрицы L и Q можно определить с помощью функции `lmireg`, которая позволяет задавать полуплоскость, круг, сектор, эллипс, параболу, полосу и их пересечения. Для поставленной задачи нужно вместо значения коэффициента демпфирования задавать внутренний угол θ^* сектора. Значение $\theta^* = \pi/2$ соответствует значению $\zeta_{\min} = 0,707$. Задавая значение $(-\beta^*)$ как границу левой полуплоскости и формируя пересечение этой полуплоскости и сектора с углом θ^* , получим область размещения полюсов, соответствующую поставленной задаче.

Таким образом, функция `h2hinfosyn` предназначена для решения поставленной и аналогичных задач. Следует помнить, что так же, как и в случае применения функции `hinfosyn`, рассмотренной выше, значение γ нужно подбирать с учетом возможного влияния на полученный результат значения $\|W_1 W_2 C G_{yf}\|_\infty$. Кроме того, нужно учитывать, что ищется регулятор с порядком, равным порядку объекта. Полученный регулятор в пространстве состояний можно преобразовать к виду (4) с $m = n$. На примере в разделе 5 будет показано, что поиск регулятора с $m = n$ может не приводить к оптимальному решению, хотя, возможно, это проблемы конкретной программной реализации алгоритма.

3.3. Синтез регулятора с фиксированной структурой

При задании структуры регулятора как передаточной функции вида (4) с заданным порядком m или как, например, ПИД регулятора уже невозможно свести задачу к решению системы линейных матричных неравенств, поэтому в [7] предлагается искать коэффициенты регулятора заданной структуры как решение задачи негладкой оптимизации. На основе этого подхода построены процедуры `hinfstruct`, `looptune` [8] и `systune` [9] из пакета MATLAB Robust Control Toolbox.

Рассмотрим функцию `systune` как наиболее универсальную. Для ее использования не нужно формировать обобщенную матрицу передаточных функций, норма которой должна минимизироваться, а критерии оптимизации можно формировать независимо друг от друга из широкого списка целей (`TuningGoal`). Важно, что эти критерии можно разделить на две группы: минимизируемые цели оптимизации (`Soft goals`) и требуемые ограничения (`Hard goals`). Для решения задачи можно использовать `TuningGoal.Gain`, где вычисляется H_∞ норма передаточной функции от указанного входа к указанному выходу, чтобы сформировать $\|G_{yf}\|_\infty$, $\|S\|_\infty$ и $\|CS\|_\infty$, определив

H_∞ норму функции чувствительности к возмущению $\|G_{yf}\|_\infty$ как минимизируемую цель, а H_∞ нормы функций чувствительности $\|S\|_\infty$ и чувствительности к помехе $\|CS\|_\infty$ — как ограничения, а также добавить ограничения на корни характеристического полинома замкнутой системы, используя `TuningGoal.Poles`.

Кроме того, в `systune` реализованы в качестве целей оптимизации ограничения во временной области. Так, в `TuningGoal.StepRejection` можно задавать требуемое время переходного процесса и использовать этот критерий как цель оптимизации вместо $\|G_{yf}\|_\infty$ или добавить его к требуемым ограничениям.

Следует отметить, что в методах, описанных в подразделах 3.1 и 3.2, порядок регулятора m не задается и обычно находится регулятор с $m = n$, причем если используются весовые передаточные функции, то порядок регулятора увеличивается в соответствии с их порядком. В функции `systune` структура регулятора должна быть задана, и если ищется регулятор вида (4), то для задания его структуры можно использовать команду `tunableTF('Controller', m, m)`, выбирая $m = n - 1$. В `systune` можно устанавливать и $m < n - 1$, но тогда нужно помнить, что стабилизирующего регулятора может не существовать, а если существует, то он оптимален только в классе регуляторов выбранной структуры. Необходимо помнить, что процедура основана на локальной оптимизации. Поэтому для получения лучшего решения, особенно в случае, когда не найдено решение, удовлетворяющее заданным ограничениям, нужно при помощи команды `systuneOptions` установить значение параметра `RandomStart` больше единицы. Тогда процедура оптимизации выполнится указанное число раз для различных случайных наборов начальных значений варьируемых коэффициентов регулятора.

Таким образом, возможности функции синтеза регулятора с фиксированной структурой `systune` полностью соответствуют поставленной задаче и позволяют формулировать и решать широкий круг аналогичных задач, выбирая различные ограничения и цели оптимизации в соответствии с требованиями конкретной системы управления.

4. Оптимизация размещения полюсов замкнутой системы

Здесь предлагается новый подход к решению поставленной задачи, основанный на поиске при помощи процедуры оптимизации желаемых значений полюсов замкнутой системы, обеспечивающих выполнение целей и ограничений оптимизации, а не непосредственно коэффициентов регулятора, как в подразделе 3.3.

Процедура размещения полюсов [22–24] основана на том, что для любого заданного полинома $\delta(s)$ и известных полиномов $a(s)$ и $b(s)$ существует единственное решение $c(s)$ и $d(s)$ уравнения

$$(18) \quad a(s)c(s) + b(s)d(s) = \delta(s)$$

при условии $\deg(d(s)) < \deg(a(s))$ или $\deg(c(s)) < \deg(b(s))$. Кроме того, решение будет удовлетворять условию причинности управления [22], т.е.

$\deg(d(s)) \leq \deg(c(s))$ будет выполняться, если $\deg(\delta(s)) \geq 2\deg(a(s)) - 1$. Таким образом, регулятор вида (4), принимая $m = n - 1$, можно получить как решение уравнения (18), задавая желаемый характеристический полином замкнутой системы $\delta(s)$ степени $2n - 1$. При таких условиях можно, приравнявая коэффициенты при одинаковых степенях s левой и правой частей уравнения (18), получить систему $2n$ линейных алгебраических уравнений с $2n$ неизвестными и найти значения коэффициентов регулятора как ее решение:

$$(19) \quad \begin{bmatrix} c_{n-1} \\ \dots \\ c_0 \\ d_{n-1} \\ \dots \\ d_0 \end{bmatrix} = M^{-1} \begin{bmatrix} \delta_{2n-1} \\ \dots \\ \delta_0 \end{bmatrix},$$

где квадратная матрица M размера $2n$ сформирована из левой части уравнения (18), используя известные значения коэффициентов $a_i, b_i (i = 0, \dots, n - 1)$.

Таким образом, задавая полином $\delta(s)$ такой, что вещественные части всех его корней отрицательны, получим стабилизирующий регулятор. При этом остается проблема сформировать желаемый полином $\delta(s)$ так, чтобы решить поставленную задачу. В [16, 17] решается похожая задача, но не рассматривается функция чувствительности к помехе и не утверждается оптимальность подавления возмущения при заданном ограничении минимального радиуса запасов устойчивости. В [25] для дискретной системы получены формулы для выбора корней желаемого характеристического полинома, обеспечивающие заданный радиус запасов устойчивости и заданный уровень подавления возмущения, но результаты получены только для объектов с вещественными полюсами и носят достаточный характер.

В настоящей статье предлагается решать поставленную задачу, используя процедуру оптимизации, в которой вектором переменных будут корни характеристического полинома замкнутой системы. Задавая допустимый диапазон значений корней, можно ограничивать коэффициент демпфирования системы и длительность переходных процессов. Таким образом, в процедуре оптимизации не проверяются условия устойчивости системы и ограничения на область расположения полюсов, так как вектор переменных варьируется только в пределах допустимой области.

В соответствии с постановкой задачи H_∞ норма функции чувствительности к возмущению (10) — это цель оптимизации, а минимальное значение R^* радиуса запасов устойчивости (6) и максимальное значение N^* H_∞ нормы функции чувствительности к помехе (9) — это ограничения. Главное отличие предлагаемого подхода от процедуры синтеза оптимального регулятора для заданных целей и ограничений, описанной в подразделе 3.3, состоит в том, что вектор переменных оптимизации — это корни характеристического полинома замкнутой системы, а не коэффициенты регулятора. Хотя формально — это небольшое отличие, так как корни характеристического полинома однозначно определяют коэффициенты регулятора из уравнения (18), но факти-

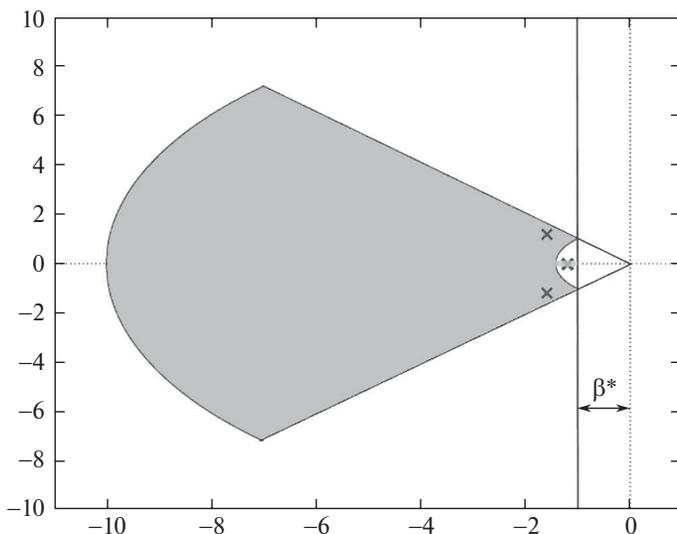


Рис. 3. Область допустимого расположения корней характеристического полинома.

чески оптимизация производится на другом множестве, что может приводить к другому результату.

Для реализации предлагаемого подхода может быть использована стандартная процедура глобальной оптимизации из пакета MATLAB Global Optimization Toolbox [26]. В этой процедуре вектор переменных — это вещественные числа. Поэтому зададим число $0 \leq n_2 < n$ комплексно-сопряженных пар корней полинома $\delta(s)$ и представим его в виде

$$(20) \quad \delta(s) = \prod_{i=1}^{2n-1-2n_2} (s + \lambda_i) \cdot \prod_{i=1}^{n_2} (s^2 + 2\zeta_i \omega_i s + \omega_i^2),$$

тогда вектор переменных будет $[\lambda_1, \dots, \lambda_{2n-1-2n_2}, \zeta_1, \dots, \zeta_{n_2}, \omega_1, \dots, \omega_{n_2}]$. Задавая границы $\zeta_{\min} \leq \zeta_i \leq 1$, получим ограничение колебаний, а границы $\lambda_i \leq -\beta^*$ и $\omega_i \geq \beta^*/\zeta_{\min}$ будут определять длительность переходных процессов. Необходимо отметить, что для процедуры глобальной оптимизации граничные значения переменных должны быть заданы в любом случае, даже если такие требования не сформулированы, т.е. $\zeta_{\min} \leq \zeta_i \leq 1$, $\lambda_{\min} \leq \lambda_i \leq -\beta^*$, $\beta^*/\zeta_{\min} \leq \omega_i \leq \omega_{\max}$, где можно принять $\omega_{\max} = -\lambda_{\min}$. Задаваемые интервалы должны быть замкнутыми, поэтому, например, вместо условия “больше нуля” нужно использовать условие “больше или равно некоторому минимальному приемлемому значению”. Вид области расположения корней характеристического полинома при таких ограничениях приведен на рис. 3.

Далее для процедуры глобальной оптимизации нужно сформировать две функции, одна из которых — функция цели — вычисляет вещественное значение, которое должно минимизироваться, а вторая — функция ограничений — вычисляет вектор значений, которые должны быть меньше заданных. В соответствии с поставленной задачей функция цели на каждом шаге процедуры

оптимизации для выбранного из допустимой области варианта вектора переменных $[\lambda_1, \dots, \lambda_{2n-1-2n_2}, \zeta_1, \dots, \zeta_{n_2}, \omega_1, \dots, \omega_{n_2}]$ выполняет следующие операции:

- 1) формирует желаемый характеристический полином (20);
- 2) решая систему (19), находит коэффициенты регулятора;
- 3) вычисляет значение $\|G_{yf}(j\omega)\|_\infty$.

Далее функция ограничений вычисляет значения $\|S(j\omega)\|_\infty$ и $\|C(j\omega)S(j\omega)\|_\infty$ и сравнивает их с заданными значениями $1/R^*$ и N^* .

Таким образом, такой подход решает поставленную задачу. Напомним, что при заданных ограничениях решения может не существовать или результат может оказаться неприемлемым, тогда нужно пересматривать ограничения. Предлагается следовать следующим рекомендациям по выбору ограничений. Минимальный радиус запасов устойчивости рекомендуется выбирать в диапазоне $0,5 \leq R^* \leq 0,7$, учитывая, что для систем, где такие значения недостижимы, нужно предварительно определить максимально возможное значение R , например, используя описанную процедуру оптимизации, где значение $\|S(j\omega)\|_\infty$ выбрано в качестве минимизируемой цели, и тогда можно принять $R^* = 0,8R_{\max}$. Значение β^* нужно согласовывать со значениями нулей объекта, так как при наличии устойчивых нулей они могут сокращаться в замкнутой системе и быстродействие получится высоким, несмотря на наличие медленных полюсов замкнутой системы. С другой стороны, как показано в [20], большое значение β^* может приводить к отсутствию решения для заданного R^* , поэтому значение β^* нужно согласовывать и со значениями полюсов объекта. Радиус запасов устойчивости должен иметь более высокий приоритет, поэтому значение β^* лучше выбирать не желаемым, а минимально допустимым, например равным минимальной собственной частоте объекта. Таким образом, выбор некоторых ограничений легко формализовать, исходя из свойств объекта, определяемых его известной передаточной функцией. А неформализованным остается только компромисс между желаемыми значениями H_∞ нормы функции чувствительности к помехе N^* и целевой функции. Можно рекомендовать увеличивать значение N^* , если решение задачи не существует при рекомендованных значениях остальных ограничений или полученное значение $\|G_{yf}(j\omega)\|_\infty$ слишком большое. А если, наоборот, получено решение с достаточно малыми значениями $\|G_{yf}(j\omega)\|_\infty$ и $\|C(j\omega)S(j\omega)\|_\infty$, то можно повторить процедуру, увеличивая значение β^* , если нужно повысить быстродействие системы.

Следует отметить, что предлагаемый подход позволяет формулировать и решать задачи с критериями, отличными от H_∞ норм. Например, если требуется минимизировать или ограничить время переходного процесса, то в качестве цели или ограничения оптимизации можно использовать результат функции MATLAB

```
data = stepinfo(Gyf, 'SettlingTimeThreshold', 0.05);
Objective_Value = data.SettlingTime;
```

которая находит время установления моделированием реакции на ступенчатый вход замкнутой системы G_{yf} .

5. Пример

В качестве примера будем синтезировать регулятор для двухмассовой системы [1, 27], представляющей собой две тележки массами m_1 и m_2 , соединенные пружиной с коэффициентом упругости k (рис. 4), трение не учитывается, управление приложено к левой тележке, положение правой тележки является измеряемым управляемым выходом, возмущение прикладывается к правой тележке.

Модель такого объекта имеет вид

$$(21) \quad \left(\frac{m_2}{k} s^4 + \frac{m_2 + m_1}{m_1} s^2 + \left(\frac{k}{m_1} - \frac{k}{m_2} \right) \right) y = \frac{1}{m_1} u + \left(\frac{1}{k} s^2 + \frac{1}{m_1} \right) f.$$

Номинальные значения параметров предполагаются $m_1 = m_2 = k = 1$. Тогда передаточные функции объекта будут

$$(22) \quad P(s) = \frac{1}{s^2(s^2 + 2)},$$

$$(23) \quad P_f(s) = \frac{s^2 + 1}{s^2(s^2 + 2)}.$$

Ставится задача синтеза регулятора, обеспечивающего длительность переходного процесса на импульсное возмущение не более 15 с и устойчивость системы при отклонении значения k от номинального в диапазоне $0,5 \leq k \leq 2,0$ [27]. Используя рассмотренные методы, будем искать регулятор для объекта (21) с номинальными значениями параметров и проверять устойчивость системы с полученным регулятором для значений $k = 0,5$ и $k = 2,0$. Если массы тележек не равны (например, $m_2 = 2,0$), то меняется структура передаточной функции объекта

$$(24) \quad P_m(s) = \frac{1}{2s^4 + 3s^2 + 0,5},$$

поэтому проверим также устойчивость системы с таким объектом.

В качестве импульса будем использовать сигнал единичной амплитуды и длительностью одна секунда. Зададим значения ограничений $R^* = 0,6$ и $N^* = 100$ для всех рассматриваемых методов.



Рис. 4. Пример объекта управления.

5.1. Синтез регулятора методом H_∞ оптимизации

Для объекта (21) принимаем обобщенные входы и выходы вида (15). Тогда получим обобщенную систему:

$$(25) \quad \dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ W_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f \\ \nu \\ u \end{bmatrix},$$

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -W_2 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} f \\ \nu \\ u \end{bmatrix}.$$

Для значений γ и W_2 , полученных по формулам (17), при $W_1 = 0,5$ функция `hinfsyn` с опцией LMI находит регулятор:

$$(26) \quad C_1(s) = \frac{2170s^3 + 354,3s^2 + 3506s + 1055}{s^4 + 53,04s^3 + 362,2s^2 + 1356s + 3032}.$$

Переходный процесс системы с этим регулятором при воздействии возмущения f в виде импульса единичной амплитуды и длительностью одна секунда показан на рис. 5. Этот регулятор обеспечивает значения $R = 0,6577$ и $\|CS\|_\infty = 47,27$, что лучше требуемых значений. При этом $\|G_{yf}\|_\infty = 2,87$. Тем не менее систему нельзя признать хорошей, так как переходный процесс колебательный и не сходится за 15 с. Это обусловлено тем, что характеристический полином замкнутой системы содержит корни $(-0,117 \pm j1,4)$ с коэффициентом демпфирования $\zeta = 0,083$ (полюса замкнутой системы показаны на рис. 6).

Кроме того, при значении коэффициента упругости $k < 0,8$, а также для объекта (24) при $m_2 = 2,0$, $k = 1,0$ система становится неустойчивой. На этом

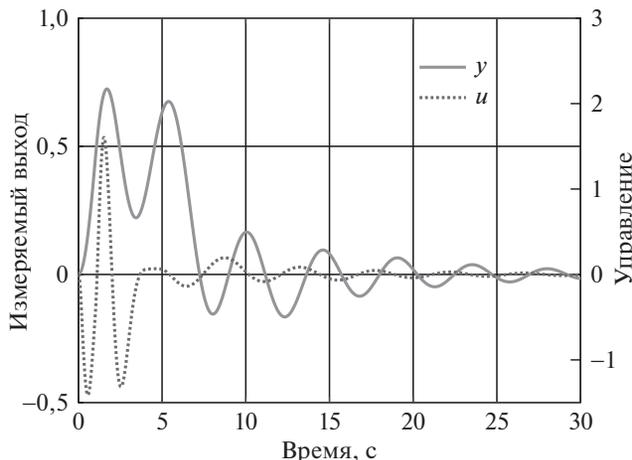


Рис. 5. Импульсное возмущение с регулятором (26).

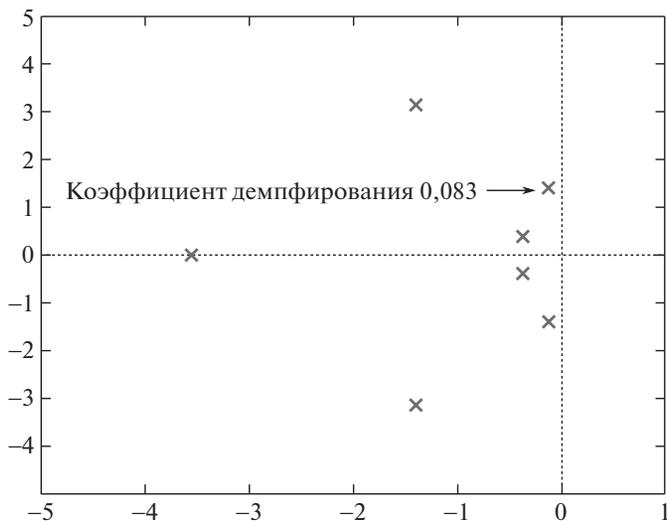


Рис. 6. Полюса замкнутой системы с регулятором (26).

примере видно, что значение коэффициента демпфирования замкнутой системы также должно быть критерием робастности, так как хотя радиус запасов устойчивости достаточно большой, система становится неустойчивой даже при не очень большом изменении параметров объекта.

5.2. Синтез H_∞ оптимального регулятора с ограничением области размещения полюсов

Попробуем получить регулятор для объекта (21) при тех же параметрах и ограничениях, используя функцию `h2hinfsv`. К обобщенной системе (25) добавим ограничение области размещения полюсов замкнутой системы: $\theta^* = \pi/2$ (что соответствует значению $\zeta = 0,707$) и $\beta^* = 0,1$. Функция не находит решения при таких условиях и заданном значении $\gamma = 1/R^*$. Поэтому проводим поиск оптимального регулятора, не задавая значения γ . Тогда получаем регулятор

$$(27) \quad C_2(s) = \frac{94,89s^4 + 9,545 \cdot 10^6 s^3 - 4,591 \cdot 10^6 s^2 + 4,907 \cdot 10^6 s + 7,221 \cdot 10^5}{s^4 + 3,414 \cdot 10^4 s^3 + 5,376 \cdot 10^5 s^2 + 3,23 \cdot 10^6 s + 8,946 \cdot 10^6},$$

который обеспечивает значения $R = 0,523$ и $\|CS\|_\infty = 279$, что существенно хуже требуемых, при этом значение $\|G_{yf}\|_\infty = 12,38$ тоже хуже, чем в системе с регулятором (26). Уменьшение значения весового коэффициента W_1 не приводит к получению регулятора, обеспечивающего заданное значение $R^* = 0,6$, поэтому проверяем работу регулятора (27). В переходном процессе на рис. 7 уже нет колебаний на выходе, но амплитуды сигналов увеличились.

Кроме того, если посмотреть нули и полюса регулятора (27)

$$C_2(s) = \frac{94,885(s + 1,006 \cdot 10^5)(s + 0,1278)(s^2 - 0,6088s + 0,5919)}{(s + 3,413 \cdot 10^4)(s + 8,027)(s^2 + 7,722s + 32,65)},$$

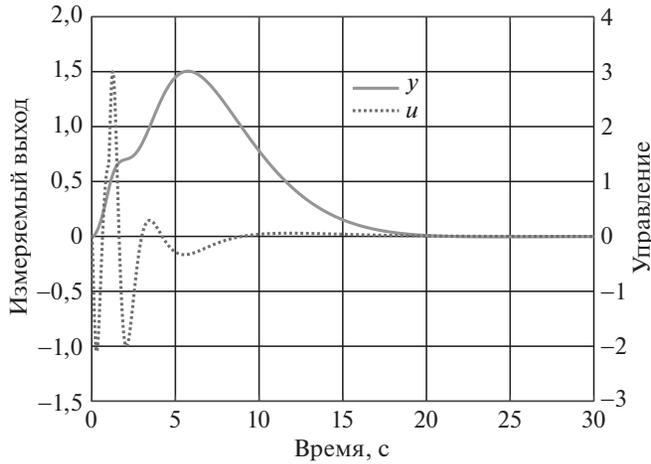


Рис. 7. Импульсное возмущение с регулятором (27).

то видно, что в регуляторе есть большие значения нуля и полюса, и скорее всего регулятор 3-го порядка был бы лучше, чем получившийся регулятор 4-го порядка, в чем убедимся далее. Тем не менее длительность переходного процесса получается около требуемых 15 с, и система с регулятором (27) остается устойчивой при значениях коэффициента упругости $k = 0,5$ и $k = 2,0$. Для объекта (24) при $m_2 = 2,0$, $k = 1,0$ устойчивость системы сохраняется, но процесс становится колебательным.

5.3. Синтез оптимального регулятора с фиксированной структурой

Для использования функции `systune` не нужно формировать обобщенную систему, а нужно

- 1) объявить регулятор как некоторую структуру с настраиваемыми параметрами, например при помощи команды

```
Controller3 = tunableTF('Controller', 3, 3),
```

которая задает структуру регулятора в виде (4) с $m = 3$;

- 2) сформировать командой `feedback` замкнутую систему с объектом, заданным передаточной функцией (22), и регулятором `Controller3`;
- 3) сформировать цель оптимизации, например как `TuningGoal.Gain` для H_∞ нормы функции чувствительности к возмущению $\|G_{yf}(j\omega)\|_\infty$ с ограничивающим профилем вида $W_2 = 1/(s^2 + 1)$, который учитывает, что возмущение приложено не там, где управление;
- 4) установить ограничения в соответствии с задачей как:
 - `TuningGoal.Sensitivity` для H_∞ нормы функции чувствительности $\|S(j\omega)\|_\infty$ с максимальным значением $1/R^*$;
 - `TuningGoal.Gain` для H_∞ нормы функции чувствительности к помехе $\|C(j\omega)S(j\omega)\|_\infty$ с максимальным значением N^* ;
 - `TuningGoal.Poles` для задания β^* , минимального коэффициента демпфирования ζ_{\min} и максимальной собственной частоты ω_{\max} .

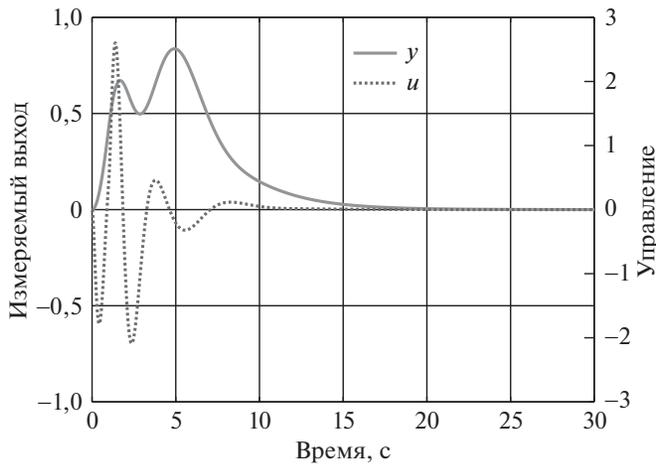


Рис. 8. Импульсное возмущение с регулятором (28).

Для значений $R^* = 0,6$, $N^* = 100$, $\beta^* = 0,1$, $\zeta_{\min} = 0,7$, $\omega_{\max} = 100$ получен регулятор

$$(28) \quad C_3(s) = \frac{99,99s^3 - 15,87s^2 + 96,81s + 18,28}{s^3 + 9,429s^2 + 42,74s + 96,99},$$

который обеспечивает все заданные ограничения и $\|G_{yf}\|_{\infty} = 5,3$. Переходный процесс системы с этим регулятором показан на рис. 8 и длится менее 15 с. Система с регулятором (28) остается устойчивой при значениях коэффициента упругости $k = 0,55$ и $k = 2,0$. При $k < 0,55$ устойчивость теряется. Для объекта (24) при $m_2 = 2,0$, $k = 1,0$ устойчивость системы сохраняется. Если, например, найти регулятор для других ограничений: $R^* = 0,606$ и $N^* = 70$, то система будет устойчивой и при $k = 0,5$, но при номинальных значениях длительность переходного процесса будет немного больше 15 с.

5.4. Синтез регулятора методом оптимизации размещения полюсов

В предлагаемом методе порядок регулятора (4) принимается $m = n - 1 = 3$. Желаемый характеристический полином замкнутой системы (20) порядка $2n - 1 = 7$ может содержать до трех пар комплексно-сопряженных корней. Для $n_2 = 3$ вектор переменных будет иметь вид $[\lambda_1, \zeta_1, \zeta_2, \zeta_3, \omega_1, \omega_2, \omega_3]$. Для применения процедуры глобальной оптимизации к решению поставленной задачи нужно определить:

- границы для каждого элемента вектора переменных, для чего будем использовать область размещения полюсов замкнутой системы такой же, как в подразделе 5.3: $\beta^* = 0,1$, $\zeta_{\min} = 0,7$, $\omega_{\max} = 100$, $\lambda_{\min} = -100$;
- функцию цели оптимизации, которая для текущего вектора переменных, определяющего коэффициенты желаемого характеристического полинома (20), и соответственно коэффициенты регулятора как решение системы (19), вычисляет значение $\|G_{yf}(j\omega)\|_{\infty}$;

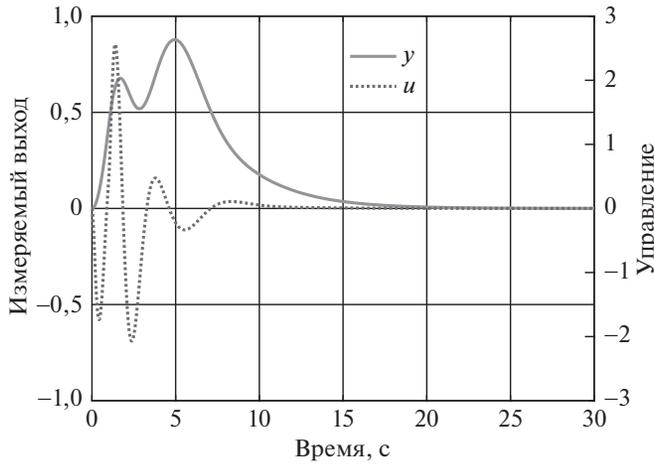


Рис. 9. Импульсное возмущение с регулятором (29).

- функцию ограничений, которая вычисляет значения $\|S(j\omega)\|_\infty$ и $\|C(j\omega)S(j\omega)\|_\infty$ и сравнивает их с заданными значениями $1/R^*$ и N^* .

Для значений $R^* = 0,6$, $N^* = 100$ процедура глобальной оптимизации находит регулятор

$$(29) \quad C_4(s) = \frac{98,38s^3 - 22,35s^2 + 86,06s + 14,8}{s^3 + 10,15s^2 + 43,96s + 98,33},$$

который обеспечивает все заданные ограничения и $\|G_{yf}\|_\infty = 6,64$. Видно, что полученный регулятор (29) практически совпадает с регулятором (28), полученным при помощи функции `syntune`. Переходный процесс системы с регулятором (29) показан на рис. 9, длится менее 15 с и также практически не отличается от процесса на рис. 8.

Система с регулятором (29) остается устойчивой при значениях коэффициента упругости $k = 0,5$ и $k = 2,0$, а также для объекта (24) при $m_2 = 2,0$, $k = 1,0$, т.е. регулятор (29) обеспечивает устойчивость в заданных границах изменения параметров объекта в отличие от регулятора (28), найденного функцией `syntune`, который не обеспечивал устойчивость при $k < 0,55$. Напомним, что регулятор (26), найденный с помощью функции `hinfsv`, не обеспечивал устойчивость системы уже при $k < 0,8$, а при помощи функции `h2hinfsv` не удалось найти регулятор, обеспечивающий заданные требования к радиусу запасов устойчивости и к H_∞ норме функции чувствительности к помехе.

6. Заключение

Классический метод H_∞ оптимизации, и особенно в версии с ограничением области размещения полюсов, казалось бы, дает все возможности для поиска оптимального регулятора, задавая необходимые для практической реализации ограничения. Но пример показывает, что не всегда полученные решения

будут действительно лучшими, в частности при прямом использовании реализации метода в пакете MATLAB Robust Control Toolbox. Следует учитывать, что специалисты, имеющие опыт использования этого метода, могут предложить дополнительные модификации подхода, существенно улучшающие результат. Тем не менее функция `sysstune`, основанная на процедуре оптимизации системы с регулятором фиксированной структуры, оказывается более гибкой при формировании целей и ограничений оптимизации, и в рассмотренном примере нашла решение, удовлетворяющее условиям поставленной задачи. При этом нужно учитывать, что решается невыпуклая задача оптимизации и не гарантируется оптимальность найденного решения. Поэтому наличие альтернативных подходов может оказаться полезным.

Предложенный в разделе 4 метод оптимизации размещения полюсов может предоставлять практически те же возможности и в рассмотренном примере дает примерно такой же результат, как и метод синтеза регулятора с фиксированной структурой, рассмотренный в подразделе 3.3. Так, регуляторы (29) и (28), построенные при помощи этих методов, очень близки. Отличие предлагаемого подхода в том, что варьируемые переменные процедуры оптимизации — это корни характеристического полинома замкнутой системы, а не коэффициенты регулятора. В каких-то случаях это может давать лучший результат, так как оптимизация производится на другом множестве варьируемых переменных. Используемая для реализации предлагаемого подхода стандартная процедура глобальной оптимизации не совсем подходит для решения задачи оптимизации размещения полюсов, так как, во-первых, варьируемые переменные должны быть вещественными, а корни характеристического полинома могут быть как вещественными, так и комплексно-сопряженными (хотя эта проблема решена заменой пары комплексно-сопряженных корней на пару вещественных коэффициентов соответствующего полинома второго порядка) и, во-вторых, процедура не учитывает, что последовательность корней в векторе не имеет значения. Тем не менее эта процедура может применяться для тестовых примеров.

СПИСОК ЛИТЕРАТУРЫ

1. Поляк Б.Т., Хлебников М.В., Рапопорт Л.Б. Математическая теория автоматического управления. М.: ЛЕНЛАНД, 2019.
2. Åström K.J., Murray R.M. Feedback Systems: an Introduction for Scientists and Engineers. New Jersey: Princeton University Press, 2008.
3. Skogestad S., Postlethwaite I. Multivariable Feedback Control. Analysis and Design. N.Y.: John Wiley and Sons, 2006.
4. Александров А.Г. Методы построения систем автоматического регулирования. М.: Физматлит, 2008.
5. Gahinet P., Apkarian P. A Linear Matrix Inequality Approach to H_∞ Control // Int. J. Robust Nonlinear Contr. 1994. V. 4. P. 421–448.
6. Chilali M., Gahinet P. H_∞ Design with Pole Placement Constraints: An LMI Approach // IEEE Trans. Autom. Control. 1996. V. 41. No. 3. P. 358–367.
7. Apkarian P., Noll D. Nonsmooth H_∞ Synthesis // IEEE Trans. Autom. Control. 2006. V. 51. No. 1. P. 71–86.

8. *Gahinet P., Apkarian P.* Decentralized and Fixed-Structure H_∞ Control in MATLAB // Proc. IEEE Conf. on Decision and Control. 2011. P. 8205–8210.
9. *Apkarian P., Gahinet P., Buhr C.* Multi-Model, Multi-Objective Tuning of Fixed-Structure Controllers // Proc. Eur. Control Conf. 2014. P. 856–861.
10. *Грязина Е.Н., Поляк В.Т., Тремба А.А.* Современное состояние метода D-разбиения // АиТ. 2008. № 12. С. 3–40.
Gryazina E.N., Polyak V.T., Tremba, A.A. D-decomposition Technique State-of-the-art // Autom. Remote Control. 2008. V. 69. No. 12. P. 1991–2026.
11. *Щербakov П.С.* Построение регуляторов заданной структуры при соблюдении инженерных требований // Проблемы управления. 2009. № 5. С. 9–17.
Shcherbakov P.S. Fixed Order Controller Design Subject to Engineering Specifications // Autom. Remote Control. 2010. V. 71. No. 6. P. 1217–1229.
12. *Александров А.Г., Честнов В.Н.* Синтез многомерных систем заданной точности. II. Применение процедур H_∞ -оптимизации // АиТ. 1998. № 8. С. 124–138.
Alexandrov A.G., Chestnov V.N. Synthesis of Multivariable Systems of Prescribed Accuracy. II. Use of Procedures of H-Infinity-Optimization // Autom. Remote Control. 1998. V. 59. No. 8. P. 1153–1164.
13. *Агафонов П.А., Честнов В.Н.* Синтез регуляторов по заданному радиусу запасов устойчивости с учетом внешних возмущений на основе H_∞ -подхода // АиТ. 2004. № 10. С. 101–108.
Agafonov P.A., Chestnov V.N. Controllers of a Given Radius of Stability Margin: Their Design by the H_∞ -Approach with Regard for External Disturbances // Autom. Remote Control. 2004. V. 65. No. 10. P. 1611–1617.
14. *Честнов В.Н.* Синтез робастных H_∞ -регуляторов многомерных систем по заданной степени устойчивости // АиТ. 2007. № 3. С. 199–205.
Chestnov V.N. Design of Robust H_∞ -Controllers of Multivariable Systems Based on the Given Stability Degree // Autom. Remote Control. 2007. V. 68. No. 3. P. 557–563.
15. *Честнов В.Н.* Синтез многомерных систем по инженерным критериям качества на основе H_∞ -оптимизации // АиТ. 2019. № 10. С. 132–152.
Chestnov V.N. Synthesis of Multivariable Systems According to Engineering Quality Criteria Based on H_∞ -Optimization // Autom. Remote Control. 2019. V. 80. No. 10. P. 1861–1877.
16. *Александров А.Г.* Синтез регуляторов по показателям точности и быстродействию. I. Минимально-фазовые одномерные объекты // АиТ. 2015. № 5. С. 27–42.
Aleksandrov A.G. Controller Design in Precision and Speed. I. Minimal Phase One-Dimensional Plants // Autom. Remote Control. 2015. V. 76. No. 5. P. 749–761.
17. *Александров А.Г.* Синтез регуляторов по показателям точности и быстродействию. II. Неминимально-фазовые объекты // АиТ. 2017. № 6. С. 3–17.
Aleksandrov A.G. Design of Controllers by Indices of Precision and Speed. II. Nonminimal-Phase Plants // Autom. Remote Control. 2017. V. 78. No. 6. P. 961–973.
18. *Александров А.Г.* Критерии грубости нестационарных систем автоматического регулирования // Аналитические методы синтеза регуляторов: Межвуз. научн. сб. Саратов: Саратов. политехн. ин-т, 1980. С. 3–14.
19. *Честнов В.Н.* Синтез регуляторов многомерных систем по заданному радиусу запасов устойчивости на базе процедуры оптимизации // АиТ. 1999. № 7. С. 100–109.
Chestnov V.N. Synthesis of Controllers for Multivariate Systems with a Given Radius of Stability Margin by the H-Infinity-Optimization Method // Autom. Remote Control. 1999. V. 60. No. 7. P. 986–993.

20. *Alexandrov V.A., Chestnov V.N., Shatov D.V.* Stability Margins for Minimum-Phase SISO Plants: A Case Study // Proc. Eur. Control Conf. 2020. P. 2068–2073.
21. *Поляк Б.Т., Хлебников М.В., Щербаков П.С.* Управление линейными системами при внешних возмущениях: Техника линейных матричных неравенств. М.: ЛЕНАНД, 2014.
22. *Острем К., Виттенмарк Б.* Системы управления с ЭВМ. М.: Мир, 1987.
23. *Гайдук А.Р.* Теория и методы аналитического синтеза систем автоматического управления (полиномиальный подход). М.: Физматлит, 2012.
24. *Ким Д.П.* Алгебраические методы синтеза систем автоматического управления. М.: Физматлит, 2014.
25. *Честнов В.Н., Александров В.А., Резков И.Г.* Синтез одномерных дискретных регуляторов по инженерным критериям качества на основе модального управления // Проблемы управления. 2019. № 6. С. 11–21.
Chestnov V.N., Alexandrov V.A., Rezkov I.G. Synthesis of Discrete Modal SISO Controllers by Engineering Performance Indices // Autom. Remote Control. 2020. V. 81. No. 6. P. 1107–1124.
26. *Ugray Z., Lasdon L., Plummer J., Glover F., Kelly J., Martí R.* Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization // INFORMS J. Computing. 2007. V. 19. No. 3. P. 328–340.
27. *Wie B., Bernstein D.S.* A Benchmark Problem for Robust Control Design // Proc. Amer. Control Conf. San Diego, CA, USA. 1990. P. 961–962.

Статья представлена к публикации членом редколлегии П.С. Щербаковым.

Поступила в редакцию 30.04.2020

После доработки 24.12.2020

Принята к публикации 15.01.2021

Интеллектуальные системы управления, анализ данных

© 2021 г. **В.В. КОНЕВ**, д-р физ.-мат. наук,
А.В. ПУПКОВ (andrewpupkov@gmail.com)
(Томский государственный университет)

ДОВЕРИТЕЛЬНОЕ ОЦЕНИВАНИЕ ПАРАМЕТРОВ АВТОРЕГРЕССИИ ПО ЗАШУМЛЕННЫМ ДАННЫМ¹

Рассматривается задача оценивания параметров процесса авторегрессии по наблюдениям с аддитивным шумом. Разработан последовательный метод построения доверительной области фиксированных размеров с заданным коэффициентом доверия для вектора неизвестных параметров по конечной выборке. Получены формулы для длительности процедуры, при которой достигается требуемое качество оценок неизвестных параметров в случае гауссовских шумов. Построение доверительных оценок параметров основано на использовании специальной последовательной модификации классических оценок Юла–Уокера, позволяющей оценить коэффициент доверия при малых и умеренных объемах выборки. Приводятся результаты численного моделирования предлагаемых оценок и их сравнение с оценками Юла–Уокера на примере доверительного оценивания спектральной плотности.

Ключевые слова: идентификация авторегрессии по зашумленным наблюдениям, последовательные оценки Юла–Уокера, доверительное оценивание, гарантированная точность.

DOI: 10.31857/S0005231021060052

1. Введение

Известно, что в теоретических и прикладных исследованиях, связанных с задачами адаптивного управления и регулирования, фильтрации и прогнозирования, анализа спектров и обработки временных рядов, большое внимание уделяется проблеме идентификации параметров динамических систем, описываемых стохастическими разностными и стохастическими дифференциальными уравнениями [1–3]. При этом для широкого круга используемых моделей неизвестные параметры входят линейно в уравнения системы, а идентификация параметров системы заключается в оценивании параметров детерминированной или стохастической регрессии [4–7]. Для идентификации линейных динамических систем разработаны различные эффективные методы: наименьших квадратов (МНК), максимального правдоподобия (МП),

¹ Работа выполнена при финансовой поддержке Российского научного фонда (проект № 17-11-01049).

стохастической аппроксимации и др. Теория оценивания параметров детерминированной регрессии разработана с исчерпывающей полнотой, позволяя строить оптимальные оценки по имеющимся данным и контролировать их точность в момент прекращения наблюдений динамической системы (см., например, [8]). Методы классической теории оценивания параметров стохастической регрессии, типичным примером которой являются временные ряды типа авторегрессии, позволили установить важные асимптотические свойства оценок при неограниченной длительности наблюдений. При практическом использовании оценок обычно исходят из того, что эти свойства сохраняются при малых и умеренных объемах данных. В последние годы в теоретических и прикладных исследованиях статистики случайных процессов возрастает роль последовательных методов анализа, которые отличаются тем, что объем данных, используемых в оценках, не фиксируется заранее, а определяется специальными правилами остановки наблюдений. Впервые целесообразность применения последовательного анализа для процессов с зависимыми значениями была установлена в публикациях [9, 10] в задаче оценивания коэффициента сноса диффузионного процесса. Было доказано, что оценка максимального правдоподобия, вычисленная в специальный (случайный) момент времени, превосходит классическую оценку максимального правдоподобия: она является несмещенной и гарантирует заданную среднеквадратическую точность. В дальнейшем последовательный анализ стал успешно применяться в задачах идентификации стохастических динамических систем с дискретным временем и более сложных моделей с непрерывным временем, позволяя улучшить как асимптотические [11, 12], так и неасимптотические свойства [13–16] классических оценок МНК и МП. Одним из важнейших результатов последовательных методов идентификации является построение точечных оценок неизвестных параметров уравнений с гарантированной среднеквадратической точностью в момент прекращения наблюдений. Последовательные оценки обычно строятся на основе классических оценок МНК и МП, используя специальные правила прекращения наблюдений и проводя дополнительные структурные изменения классических оценок. При этом конструкция последовательных оценок существенно зависит от числа неизвестных параметров: она использует одну классическую оценку МНК, если число неизвестных параметров не превышает размерности процесса, описывающего динамику системы [13, 14], и она включает целую серию базовых оценок (случайной длины) в случае, когда число неизвестных параметров больше, чем размерность процесса [15, 16].

В данной статье рассматривается задача идентификации параметров авторегрессии по наблюдениям, искаженным аддитивным шумом. Такая модель широко используется в задачах фильтрации и прогнозирования, при обработке сигналов в информационно-измерительных комплексах, анализе речи и др. Проблеме идентификации параметров зашумленной авторегрессии посвящено много публикаций [17–22] и др. Основные методы идентификации можно разбить на две группы. К первой группе относятся методы, использующие для описания зашумленной авторегрессии, модели авторегрессии и скользящего среднего. Затем параметры оцениваются применением модифицированных оценок Юла–Уокера, метода максимального правдоподобия или

метода рекуррентного прогноза ошибок. К другой группе методов относят улучшенные методы МНК [21]. Теоретические исследования свойств оценок носят асимптотический характер. Они позволили определить потенциальные возможности построения авторегрессионных моделей в условиях, когда авторегрессионный процесс недоступен прямым наблюдениям. При этом вопросы о точности оценок при малых и умеренных объемах данных остаются открытыми.

Цель данной статьи — разработать метод неасимптотического доверительного оценивания параметров зашумленной авторегрессии. Предполагается, что шумы, задающие динамику процесса авторегрессии, и шумы в канале измерений являются гауссовскими. Последовательная процедура идентификации строится на основе одной классической оценки Юла–Уокера, используя специальные правила остановки наблюдений и дополнительную модификацию структуры оценки с помощью весовых коэффициентов. Конструкция последовательных оценок параметров зашумленной авторегрессии и ее анализ существенно опираются на недавние результаты по последовательной идентификации гауссовской авторегрессии по прямым наблюдениям [23, 24]. Задача неасимптотического доверительного оценивания линейных параметров в многомерных динамических системах с условно-гауссовскими шумами изучалась в [25] в предположении, что число неизвестных параметров не превышает размерности процесса, а сам процесс доступен прямому наблюдению. Другие подходы к построению последовательных процедур идентификации динамических систем исследованы в [26, 27]. Отметим, что, помимо задач идентификации динамических систем, последовательные методы играют ключевую роль в теории оптимальной остановки [28] и в задачах скорейшего оптимального обнаружения разладок [29].

2. Постановка задачи. Построение доверительной оценки для авторегрессии первого порядка

Рассмотрим модель зашумленной авторегрессии $AR(p)$ p -го порядка, описываемого уравнениями

$$(2.1) \quad x_k = \theta_1 x_{k-1} + \dots + \theta_p x_{k-p} + \varepsilon_k,$$

$$(2.2) \quad y_k = x_k + \eta_k, \quad k \geq 1,$$

где $\{\varepsilon_k\}$ и $\{\eta_k\}$ — независимые последовательности гауссовских случайных величин (шумы) с нулевыми средними $\mathbf{E}\varepsilon_k = \mathbf{E}\eta_k = 0$ и дисперсиями $\mathbf{E}\varepsilon_k^2 = \sigma^2$ и $\mathbf{E}\eta_k^2 = \Delta^2$. Параметры $\theta_1, \dots, \theta_p$ неизвестны, и их требуется оценить по наблюдениям процесса $\{y_k\}$. Предполагается, что вектор начальных значений x_0, \dots, x_{1-p} процесса $AR(p)$ является случайным, не зависящим от процессов $\{\varepsilon_k\}$ и $\{\eta_k\}$.

Из (2.1), (2.2) получаем уравнение для наблюдаемого процесса

$$(2.3) \quad y_k = Y_{k-1}^T \theta + \xi_k,$$

где $Y_k = (y_k, \dots, y_{k-p+1})^T$, $\theta = (\theta_1, \dots, \theta_p)^T$, $\xi_k = \varepsilon_k + \eta_k - \theta_1 \eta_{k-1} - \dots - \theta_p \eta_{k-p}$; T обозначает транспонирование. Поскольку оценка МНК для вектора θ является смещенной, для оценивания θ часто используют оценку Юла–Уокера, которая вычисляется по наблюдениям y_p, \dots, y_n по формуле

$$(2.4) \quad \hat{\theta}(n) = G_n^{-1} \sum_{k=2p+1}^n Y_{k-p-1} y_k, \quad G_n = \sum_{k=2p+1}^n Y_{k-p-1} Y_{k-1}^T.$$

Подставляя y_k из (2.3) в (2.4), получаем уклонение оценки

$$(2.5) \quad \hat{\theta}(n) - \theta = G_n^{-1} \zeta(n), \quad \zeta(n) = \sum_{k=2p+1}^n Y_{k-p-1} \xi_k.$$

Оценка (2.4) является нелинейной функцией от наблюдений, ее свойства исследованы только при асимптотическом предположении, что объем выборки сколь угодно велик [2, 3]. Вопрос о качестве оценок (2.4) при малых и умеренных объемах данных остается открытым.

Цель статьи — получить решение задачи доверительного оценивания параметров $\theta_1, \dots, \theta_p$ в системе (2.1), (2.2) в неасимптотической постановке. На основе оценки Юла–Уокера (2.4), используя подход последовательного анализа, построим доверительные оценки для $\theta = (\theta_1, \dots, \theta_p)^T$, выбрав вместо n специальное правило остановки наблюдений в зависимости от требуемой точности оценки и проведя дополнительную модификацию структуры самой оценки (2.4).

Отложив до раздела 3 построение последовательных оценок Юла–Уокера и их анализ в общей модели (2.1), (2.2), рассмотрим сначала задачу оценивания параметра θ в зашумленном процессе авторегрессии первого порядка

$$(2.6) \quad x_k = \theta x_{k-1} + \varepsilon_k, \quad y_k = x_k + \eta_k, \quad k \geq 1.$$

При этом из (2.4)–(2.6) получаем, что

$$(2.7) \quad \hat{\theta}(n) = \frac{1}{\sum_{k=3}^n y_{k-2} y_{k-1}} \sum_{k=3}^n y_{k-2} y_k, \quad \hat{\theta}(n) - \theta = \frac{\zeta(n)}{\sum_{k=3}^n y_{k-2} y_{k-1}}, \quad n \geq 3,$$

$$(2.8) \quad \zeta(n) = \sum_{k=3}^n y_{k-2} \xi_k, \quad \xi_k = \eta_k - \theta \eta_{k-1} + \varepsilon_k.$$

Точечная последовательная оценка для θ строится в два этапа.

Этап 1. Представление процесса $\zeta(n)$ в виде суммы двух мартингалов.

Этап 2. Построение момента остановки наблюдений и выбор весовых коэффициентов в модифицированной оценке Юла–Уокера.

Множество индексов суммирования в (2.8) $T(n) = \{3, 4, \dots, n\}$ разобьем на нечетные и четные: $T(n) = T_1(n) \cup T_2(n)$, где

$$(2.9) \quad \begin{aligned} T_1(n) &= \{k = 2j + 1 : j = 1, 2, \dots; k \leq n\}, \\ T_2(n) &= \{k = 2(j + 1) : j = 1, 2, \dots; k \leq n\}. \end{aligned}$$

Заметим, что количества индексов во множествах $T_1(n)$ и $T_2(n)$ определяются величинами $d_1(n) = [(n - 1)/2]$, $d_2(n) = [(n - 2)/2]$, $n \geq 3$, где $[a]$ обозначает целую часть числа a . Далее запишем сумму (2.8) в виде $\zeta(n) = \zeta_1(n) + \zeta_2(n)$, где

$$(2.10) \quad \zeta_i(n) = \sum_{k=3}^n \chi_{\{k \in T_i(n)\}} y_{k-2} \xi_k, \quad i = 1, 2;$$

χ_A обозначает индикатор события A .

Обозначим через $m_1(n)$ и $m_2(n)$ максимальные индексы в $T_1(n)$ и $T_2(n)$, т.е.

$$\begin{aligned} m_1(n) &= \max \{2j + 1 : j = 1, 2, \dots; 2j + 1 \leq n\}, \\ m_2(n) &= \max \{2(j + 1) : j = 1, 2, \dots; 2(j + 1) \leq n\}. \end{aligned}$$

Введем две фильтрации $\{\mathcal{F}_n^{(1)}\}_{n \geq 0}$ и $\{\mathcal{F}_n^{(2)}\}_{n \geq 0}$, где

$$(2.11) \quad \begin{aligned} \mathcal{F}_0^{(i)} &= \sigma\{x_0\}, \\ \mathcal{F}_n^{(i)} &= \sigma\{x_0, \varepsilon_1, \dots, \varepsilon_{m_i(n)}, \eta_1, \dots, \eta_{m_i(n)}\}, \end{aligned} \quad i = 1, 2, \quad n \geq 1.$$

Первый этап завершается следующим результатом.

Лемма 1. Процессы $(\zeta_1(n), \mathcal{F}_n^{(1)})_{n \geq 3}$ и $(\zeta_2(n), \mathcal{F}_n^{(2)})_{n \geq 3}$ в разложении (2.10) являются мартингалами.

Доказательство леммы 1 приводится в Приложении.

Построим теперь последовательную оценку Юла–Уокера.

Пусть $h > 0$. Введем два момента остановки по нечетным и четным наблюдениям процесса y_k :

$$(2.12) \quad \begin{aligned} \tau_1(h) &= \inf \left\{ n \geq 3 : \sum_{j=1}^{d_1(n)} y_{2j-1}^2 \geq \frac{h}{\Delta^2 + \sigma^2} \right\}, \\ \tau_2(h) &= \inf \left\{ n \geq 1 : \sum_{j=1}^{d_2(n)} y_{2j}^2 \geq \frac{h}{\Delta^2 + \sigma^2} \right\}. \end{aligned}$$

Замечание 1. Известно (см., например, [11, 13]), что при построении последовательных оценок параметра $AR(1)$ по прямым наблюдениям моменты остановки наблюдений определяются с помощью выборочной информации Фишера. В модели (2.6) с дополнительной аддитивной помехой моменты

$\tau_1(h)$ и $\tau_2(h)$ выбираются так, чтобы выборочные информации по Фишеру $\sum_{j=1}^{d_1(n)} y_{2j-1}^2$ и $\sum_{j=1}^{d_2(n)} y_{2j}^2$, содержащиеся отдельно в нечетных и четных наблюдениях, достигли заданного порога $h/(\Delta^2 + \sigma^2)$.

Длительность наблюдений в последовательной процедуре оценивания θ определим равенством

$$(2.13) \quad \tau(h) = \max(\tau_1(h), \tau_2(h)).$$

Кроме замены в оценке (2.7) n на $\tau(h)$, потребуются некоторая модификация самой оценки, используя специальные весовые коэффициенты. Сначала найдем корректирующие множители $\alpha_1(h)$ и $\alpha_2(h)$, компенсирующие перекоки в (2.12), из уравнений

$$\begin{aligned} \sum_{j=1}^{d_1(\tau_1(h))-1} y_{2j-1}^2 + \alpha_1(h) y_{2d_1(\tau_1(h))-1}^2 &= \frac{h}{\Delta^2 + \sigma^2}, \\ \sum_{j=1}^{d_2(\tau_2(h))-1} y_{2j}^2 + \alpha_2(h) y_{2d_2(\tau_2(h))}^2 &= \frac{h}{\Delta^2 + \sigma^2}. \end{aligned}$$

Заметим, что $0 < \alpha_i(h) \leq 1$, $i = 1, 2$. Положим

$$\beta_{1,k} = \begin{cases} 1, & \text{если } k = 2j - 1 \text{ и } j < d_1(\tau_1(h)); \\ \alpha_1(h), & \text{если } k = 2d_1(\tau_1(h)) - 1; \\ 0, & \text{если } k > 2d_1(\tau_1(h)) - 1; \end{cases}$$

$$\beta_{2,k} = \begin{cases} 1, & \text{если } k = 2j + 2 \text{ и } j < d_2(\tau_2(h)); \\ \alpha_2(h), & \text{если } k = 2d_2(\tau_2(h)); \\ 0, & \text{если } k > 2d_2(\tau_2(h)). \end{cases}$$

Построим точечную последовательную оценку Юла–Уокера в виде

$$(2.14) \quad \hat{\theta}_{\tau(h)} = \left(\sum_{k=3}^{\tau(h)} \sqrt{\gamma_k} y_{k-2} y_{k-1} \right)^{-1} \sum_{k=3}^{\tau(h)} \sqrt{\gamma_k} y_{k-2} y_k,$$

где весовые коэффициенты γ_k определяются по формуле

$$(2.15) \quad \gamma_k = \begin{cases} \beta_{1,k}, & \text{если } k \in T_1(\tau_1(h)); \\ \beta_{2,k}, & \text{если } k \in T_2(\tau_2(h)); \\ 0, & \text{если } k \notin T(\tau(h)). \end{cases}$$

Заметим, что весовые коэффициенты γ_k в оценке (2.14) равны единице для наблюдений с нечетными номерами до момента останова $\tau_1(h)$, а также для наблюдений с четными номерами до момента $\tau_2(h)$; наблюдения в эти

моменты используются в суммах с корректирующими множителями $\alpha_1(h)$ и $\alpha_2(h)$; остальные коэффициенты в обеих суммах равны нулю.

Покажем, что последовательная оценка (2.14) дает возможность решить задачу доверительного оценивания параметра θ в зашумленной авторегрессии $AR(1)$ в неасимптотической постановке. Подставляя y_k из (2.6) в (2.14), получаем

$$(2.16) \quad \hat{\theta}_{\tau(h)} - \theta = \frac{\zeta(h)}{s(h)};$$

$$s(h) = \sum_{k=3}^{\tau(h)} \sqrt{\gamma_k} y_{k-2} y_{k-1}, \quad \zeta(h) = \sum_{k=3}^{\tau(h)} \sqrt{\gamma_k} y_{k-2} \xi_k.$$

Обозначим:

$$(2.17) \quad \tilde{\zeta}(h) = \varkappa_\theta \zeta(h) = \sum_{k=3}^{\tau(h)} \sqrt{\gamma_k} y_{k-2} \tilde{\xi}_k,$$

$$\tilde{\xi}_k = \varkappa_\theta \xi_k, \quad \varkappa_\theta = \sqrt{\frac{\Delta^2 + \sigma^2}{\Delta^2(1 + \theta^2) + \sigma^2}}.$$

Подставляя в (2.17) весовые коэффициенты (2.15), получаем разложение

$$(2.18) \quad \tilde{\zeta}(h) = \tilde{\zeta}_1(h) + \tilde{\zeta}_2(h);$$

$$(2.19) \quad \tilde{\zeta}_1(h) = \sum_{k=3}^{\tau_1(h)} \chi_{\{k \in T_1(\tau_1(h))\}} \sqrt{\beta_{1,k}} y_{k-2} \tilde{\xi}_k,$$

$$\tilde{\zeta}_2(h) = \sum_{k=3}^{\tau_2(h)} \chi_{\{k \in T_2(\tau_2(h))\}} \sqrt{\beta_{2,k}} y_{k-2} \tilde{\xi}_k.$$

Благодаря специальному выбору весовых коэффициентов (2.15) справедлив следующий результат.

Лемма 2. Пусть шумы $\{\varepsilon_k\}$ и $\{\eta_k\}$ в (2.6) являются гауссовскими, а случайные величины $\tilde{\zeta}_1(h)$ и $\tilde{\zeta}_2(h)$ определяются равенствами (2.19).

Тогда для любого $h > 0$ случайные величины $h^{-1/2} \tilde{\zeta}_1(h)$ и $h^{-1/2} \tilde{\zeta}_2(h)$ являются стандартными гауссовскими.

Доказательство леммы 2, основанное на теореме П.1, приводится в Приложении.

Построим доверительный интервал для θ с заданным коэффициентом доверия $0 < \rho < 1$. Используя (2.16), (2.18), получаем неравенство

$$(2.20) \quad h^{-1/2} \varkappa_* |\hat{\theta}_{\tau(h)} - \theta| s(h) \leq h^{-1/2} (|\tilde{\zeta}_1(h)| + |\tilde{\zeta}_2(h)|), \quad \varkappa_* = \min_{|\theta| \leq L} \varkappa_\theta.$$

Применяя лемму 2, находим оценку вероятности

$$\begin{aligned}
 & \mathbb{P}_\theta \left(\frac{|\tilde{\zeta}_1(h)|}{\sqrt{h}} + \frac{|\tilde{\zeta}_2(h)|}{\sqrt{h}} \geq \mu^* \right) \leq \\
 & \leq \mathbb{P}_\theta \left(\frac{|\tilde{\zeta}_1(h)|}{\sqrt{h}} \geq \frac{\mu^*}{2} \right) + \mathbb{P}_\theta \left(\frac{|\tilde{\zeta}_2(h)|}{\sqrt{h}} \geq \frac{\mu^*}{2} \right) = \\
 (2.21) \quad & = 2 \mathbb{P}_\theta \left(\frac{|\tilde{\zeta}_1(h)|}{\sqrt{h}} \geq \frac{\mu^*}{2} \right) = 4 \left(1 - \Phi \left(\frac{\mu^*}{2} \right) \right),
 \end{aligned}$$

$$\text{где } \Phi(z) = (2\pi)^{-1/2} \int_{-\infty}^z e^{-\frac{u^2}{2}} du.$$

Из уравнения $4(1 - \Phi(\mu^*/2)) = 1 - \rho$ определяем $\mu^* = \mu^*(\rho) = 2\Phi^{-1} \times \times ((3 + \rho)/4)$.

Используя неравенства (2.20), (2.21), приходим к следующему результату.

Теорема 1. Пусть шумы $\{\varepsilon_k\}$ и $\{\eta_k\}$ в (2.6) являются гауссовскими и точечная последовательная оценка $\hat{\theta}_{\tau(h)}$ определяется формулой (2.14). Тогда для любых $0 < \rho < 1$ и $h > 0$

$$(2.22) \quad \inf_{|\theta| \leq L} \mathbb{P}_\theta \left\{ \varkappa_* \frac{|s(h)|}{\sqrt{h}} |\hat{\theta}_{\tau(h)} - \theta| < \mu^* \right\} \geq \rho.$$

Неравенство (2.22) позволяет по заданным коэффициенту доверия ρ и параметру $h > 0$ построить доверительный интервал для θ . При этом ширина доверительного интервала зависит от уровней шумов σ^2 и Δ^2 , а также от статистики $s(h)$.

Асимптотические свойства длительности последовательной процедуры $\tau(h)$ и статистики $s(h)$ при $h \rightarrow \infty$ для устойчивого процесса авторегрессии (2.6) сформулируем в виде следующего утверждения.

Предложение 1. Пусть $|\theta| < 1$, $\tau(h)$ и $s(h)$ определяются равенствами (2.13), (2.16). Тогда

$$(2.23) \quad \lim_{h \rightarrow \infty} \frac{\tau(h)}{h} = \frac{2(1 - \theta^2)}{(\Delta^2 + \sigma^2)((1 - \theta^2)\Delta^2 + \sigma^2)} \quad \text{почти наверное (п.н.)},$$

$$(2.24) \quad \lim_{h \rightarrow \infty} \frac{s(h)}{h} = \frac{2\sigma^2\theta}{(\Delta^2 + \sigma^2)((1 - \theta^2)\Delta^2 + \sigma^2)} \quad \text{п.н.}$$

Утверждение предложения 1 следует из предложения 2, приведенного в разделе 3.

3. Доверительное оценивание параметров $AR(p)$ по зашумленным наблюдениям

Рассмотрим теперь общую задачу неасимптотического оценивания параметров $\theta_1, \dots, \theta_p$ в модели (2.1), (2.2). На основе оценки Юла-Уокера (2.4)

построим последовательную процедуру, гарантирующую точность оценок в момент прекращения наблюдений. При этом развивается подход, использованный в разделе 2. Во-первых, заметим, что шумовая часть $\zeta(n)$ в уклонении оценки (2.5) не является мартингалом, как и в случае одного неизвестного параметра.

На первом шаге покажем, что процесс $\zeta(n)$ допускает представление в виде суммы мартингалов. Начнем с разбиения множества индексов суммирования в (2.5):

$$(3.1) \quad T(n) = \{2p+1, 2p+2, \dots, n\} = \bigcup_{i=1}^{p+1} T_i(n),$$

$$T_i(n) = \{k = (p+1)j + p + i - 1 : j = 1, 2, \dots; k \leq n\}.$$

Обозначим через $d_i(n)$ число элементов в множестве $T_i(n)$:

$$d_i(n) = [(n - p - i + 1)/(p + 1)], \quad \text{если } T_i \neq \emptyset, \quad \text{и}$$

$$d_i(n) = 0, \quad \text{если } T_i = \emptyset.$$

При этом максимальный номер $t_i(n)$ в (3.1), если $T_i(n) \neq \emptyset$, есть $t_i(n) = 3d_i(n) + i + 1$.

Для каждого $i = \overline{1, p+1}$ введем фильтрацию $F^{(i)} = \left\{ \mathcal{F}_n^{(i)} \right\}_{n \geq 0}$, где

$$(3.2) \quad \mathcal{F}_0^{(i)} = \sigma\{x_0, x_{-1}, \dots, x_{-p+1}\}, \quad \text{если } t_i(n) = 0,$$

$$\mathcal{F}_n^{(i)} = \sigma\{x_0, x_{-1}, \dots, x_{-p+1}; \varepsilon_1, \dots, \varepsilon_{t_i(n)}, \eta_1, \dots, \eta_{t_i(n)}\}, \quad \text{если } t_i(n) \geq 1.$$

Далее векторный процесс $\zeta(n)$ в (2.5) представим в виде

$$(3.3) \quad \zeta(n) = \sum_{i=1}^{p+1} \zeta^{(i)}(n), \quad \zeta^{(i)}(n) = \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} Y_{k-p-1} \xi_k.$$

Используя координаты p -мерного вектора $\zeta^{(i)}(n)$, т.е.

$$(3.4) \quad \langle \zeta^{(i)}(n) \rangle_l = \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} y_{k-p-l} \xi_k, \quad l = 1, \dots, p,$$

введем процессы

$$(3.5) \quad M_l^{(i)}(n) = \varkappa_\theta \langle \zeta^{(i)}(n) \rangle_l = \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} y_{k-p-l} \tilde{\xi}_k;$$

$$(3.6) \quad \tilde{\xi}_k = \varkappa_\theta \xi_k, \quad \varkappa_\theta = (\Delta^2 + \sigma^2)^{1/2} / (\sigma^2 + \Delta^2(1 + \theta_1^2 + \dots + \theta_p^2))^{1/2}.$$

Учитывая (2.3) и гауссовость шумов ε_k и η_k в (2.1), (2.2), отметим, что случайная величина $\tilde{\xi}_k$ имеет гауссовское распределение с параметрами $(0, \Delta^2 + \sigma^2)$.

Следующие свойства процессов (3.5) играют ключевую роль при анализе рассматриваемой процедуры в Приложении.

Лемма 3. Для каждого $i = \overline{1, p+1}$ случайные процессы $M_l^{(i)}(n)$, $l = \overline{1, p}$, являются квадратично-интегрируемыми мартингалами с условно-гауссовскими приращениями, т.е.

$$\text{Law} \left(\Delta M_l^{(i)}(n) | \mathcal{F}_{n-1}^{(i)} \right) = \mathcal{N}(0, \sigma_{i,l}^2(n-1)),$$

причем

$$\sigma_{i,l}^2(n-1) = \mathbf{E}_\theta \left(\left(\Delta M_l^{(i)}(n) \right)^2 | \mathcal{F}_{n-1}^{(i)} \right) = \chi_{\{n \in T_i(n)\}} (\sigma^2 + \Delta^2) y_{n-p-l}^2.$$

Доказательство леммы 3 дано в Приложении.

На втором шаге построим последовательную модификацию оценки Юла–Уокера. Сначала для каждого вектора $\zeta^{(i)}(n)$ в разложении (3.3) введем p моментов остановки, связанных с мартингалами (3.5). Пусть $h > 0$. Положим $\tau_0^{(i)}(h) = 2p$,

$$(3.7) \quad \tau_l^{(i)}(h) = \inf \left\{ n > \tau_{l-1}^{(i)}(h) : \sum_{k=\tau_{l-1}^{(i)}(h)+1}^n \chi_{\{k \in T_i(n)\}} y_{k-p-l}^2 \geq \frac{h}{\Delta^2 + \sigma^2} \right\},$$

$$l = \overline{1, p}.$$

Длительность наблюдений в последовательной процедуре определим по формуле

$$(3.8) \quad \tau(h) = \max_{1 \leq i \leq p+1} \tau_p^{(i)}(h).$$

Введенные моменты остановки (3.7) дают возможность контролировать процесс накопления информации о неизвестных параметрах $\theta_1, \dots, \theta_p$. С их помощью выберем весовые коэффициенты в последовательной оценке Юла–Уокера. Во-первых, найдем корректирующие множители $\alpha_l^{(i)}(h)$, $1 \leq l \leq p$, компенсирующие перескоки в (3.7), из уравнений

$$\sum_{k=\tau_{l-1}^{(i)}+1}^{\tau_l^{(i)}(h)-1} \chi_{\{k \in T_i(\tau_l^{(i)}(h))\}} y_{k-p-l}^2 + \alpha_l^{(i)}(h) y_{\tau_l^{(i)}(h)-p-l}^2 = \frac{h}{\Delta^2 + \sigma^2},$$

$$\tau_0^{(i)}(h) = 2p.$$

Отметим, что моменты остановки $\tau_1^{(i)}(h), \dots, \tau_p^{(i)}(h)$ для каждого $i = \overline{1, p+1}$ образуют возрастающую последовательность, причем каждый из моментов строится по квадратичной характеристике $\sigma_{i,l}^2(n-1)$ своего мартингала (3.5).

Положим

$$(3.9) \quad \beta_{l,k}^{(i)}(h) = \begin{cases} 1, & \text{если } \tau_{l-1}^{(i)}(h) < k < \tau_l^{(i)}(h), \\ \alpha_1^{(i)}(h), & \text{если } k = \tau_l^{(i)}(h), \end{cases}$$

и определим весовые коэффициенты $\gamma_{l,k}$ равенствами

$$(3.10) \quad \gamma_{l,k} = \sum_{i=1}^{p+1} \mu_{k,l}^{(i)} \sqrt{\beta_{l,k}^{(i)}}, \quad \mu_{k,l}^{(i)} = \chi_{\{k \in T_i(k)\}} \chi_{(\tau_{l-1}^{(i)}(h), \tau_l^{(i)}(h)]}(k).$$

Построение последовательной оценки Юла–Уокера завершим определением.

Определение 1. Для каждого $h > 0$ последовательный план оценивания вектора параметров $\theta = (\theta_1, \dots, \theta_p)^\top$ в модели (2.1), (2.2) задается парой $(\tau(h), \theta^*(h))$, где $\tau(h)$ — длительность процедуры, определенная в (3.8), $\theta^*(h) = (\theta_1^*(h), \dots, \theta_p^*(h))^\top$ — вектор оценок, задаваемый формулами

$$(3.11) \quad \theta^*(h) = G^{-1}(h)\vartheta(h).$$

Здесь $\vartheta(h) = (\vartheta_1(h), \dots, \vartheta_p(h))^\top$ — p -мерный вектор с координатами

$$(3.12) \quad \vartheta_l(h) = \sum_{k=2p+1}^{\tau(h)} \gamma_{l,k} y_{k-p-l} y_k, \quad 1 \leq l \leq p;$$

$G(h)$ — матрица размера $p \times p$ с элементами

$$(3.13) \quad \langle G(h) \rangle_{l,s} = \sum_{k=2p+1}^{\tau(h)} \gamma_{l,k} y_{k-p-l} y_{k-s}.$$

Отметим, что в отличие от оценки (2.4) последовательная оценка (3.11) строится по выборке случайного объема. При этом длительность процедуры $\tau(h)$ и весовые коэффициенты $\gamma_{l,k}$ зависят от выбора значения параметра h , который позволяет контролировать точность оценивания. Используя точечную оценку (3.11), получаем решение задачи доверительного оценивания в неасимптотической постановке.

Теорема 2. Пусть шумы $\{\varepsilon_k\}$ и $\{\eta_k\}$ в уравнениях (2.1), (2.2) являются гауссовскими, $\theta \in \Theta \subset \mathbb{R}^p$ и последовательный план оценивания вектора параметров $\theta = (\theta_1, \dots, \theta_p)^\top$ определяется формулами (3.8), (3.11). Тогда для любого $0 < \rho < 1$ и $h > 0$

$$\inf_{\theta \in \Theta} \mathbb{P}_\theta \left(\frac{\varkappa_*}{\sqrt{h}} \|G(h)(\theta^*(h) - \theta)\| < \mu^* \right) \geq \rho,$$

где

$$\varkappa_* = \inf_{\theta \in \Theta} \varkappa_\theta,$$

$$(3.14) \quad \mu^* = (p+1) \Phi_p^{-1} \left(\frac{p+\rho}{p+1} \right),$$

$$\Phi_p(a) = \frac{1}{2^{\frac{p}{2}-1} \Gamma(\frac{p}{2})} \int_0^a z^{p-1} e^{-\frac{z^2}{2}} dz, \quad a \geq 0.$$

Доказательство теоремы 2, приведенное в Приложении, опирается на теорему П.2 о свойствах остановленных квадратично-интегрируемых мартингалов с условно-гауссовскими приращениями, установленных в [24]. Согласно теореме 2 для построения доверительной области для вектора параметров $\theta = (\theta_1, \dots, \theta_p)^T$ по заданному коэффициенту доверия ρ сначала находится пороговое значение μ^* по формуле (3.14). Размеры доверительной области для θ можно сократить, увеличивая параметр h , определяющий длительность наблюдений в процедуре.

Предложение 2. Пусть $AR(p)$ процесс (2.1) устойчив, моменты остановки $\tau(h)$ и матрица $G(h)$ определяются равенствами (3.8) и (3.13). Тогда

$$(3.15) \quad \lim_{h \rightarrow \infty} \frac{\tau(h)}{h} = \frac{p(p+1)}{(\Delta^2 + \sigma^2)(\langle F \rangle_{11} + \Delta^2)} \quad n.n.,$$

$$\lim_{h \rightarrow \infty} \frac{G(h)}{h} = \frac{(p+1)F(A^T)^p}{(\Delta^2 + \sigma^2)(\langle F \rangle_{11} + \Delta^2)} \quad n.n.,$$

где

$$F = \sum_{j \geq 0} A^j B(A^T)^j, \quad B = \|\sigma^2 \delta_{1,i} \delta_{1,j}\|, \quad A = \begin{pmatrix} \theta_1 & \dots & \theta_p \\ I_{p-1} & & 0 \end{pmatrix}.$$

Доказательство предложения 2 приводится в Приложении.

Замечание 2. Из предложения 2 следует, что в случае устойчивого процесса (2.1) диаметр доверительного эллипсоида, определяемый теоремой 2, уменьшается с ростом h как const/\sqrt{h} .

4. Численные результаты

Рассмотрим выборочные свойства построенных оценок и оценок Юла-Уокера для параметра θ в модели (2.6) с гауссовскими шумами $\{\varepsilon_k\}$ и $\{\eta_k\}$ с $\mathbf{E}\varepsilon_k = \mathbf{E}\eta_k = 0$, $\mathbf{E}\varepsilon_k^2 = 1$ и $\mathbf{E}\eta_k^2 = 0,01$. Заметим, что при $|\theta| < 1$ наблюдаемый процесс $\{y_k\}$ имеет спектральную плотность [2]

$$(4.1) \quad f(\lambda) = \frac{\sigma^2 + \Delta^2 (-2\theta \cos \lambda + 1 + \theta^2)}{2\pi (-2\theta \cos \lambda + 1 + \theta^2)}, \quad -\pi \leq \lambda \leq \pi,$$

если $\{x_k\}$ в (2.6) является стационарным. На рис. 1 для длительности $\tau(h)$ предлагаемой последовательной процедуры (2.14) приводится выборочное поведение отношения $\tau(h)/h$ в зависимости от параметра $-1 < \theta < 1$ при $h = 300$.

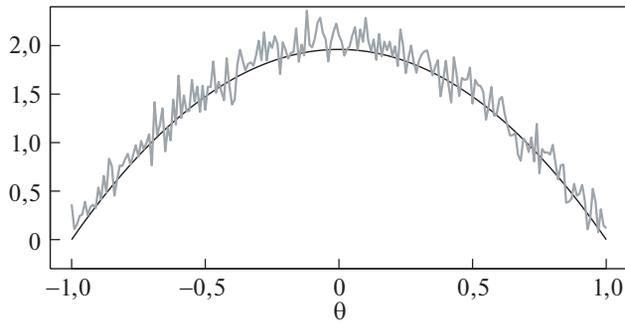


Рис. 1. Выборочное поведение отношения $\tau(h)/h$ (серая линия) и асимптотическая кривая (2.23) (черная линия).

В табл. 1 представлены средние значения последовательных оценок $\hat{\theta}_{\tau(h)}$, полученные с помощью процедуры (2.14) с $h = 1000$, а также средние значения классических оценок Юла–Уокера θ_n . Средние в обоих случаях вычислялись по 1000 повторений каждой процедуры. При этом объем выборки для оценок Юла–Уокера выбирался равным соответствующей средней длительности последовательной процедуры, указанной в столбце $\bar{\tau}$. Доверительный интервал на основе последовательной оценки (2.14) строился с помощью теоремы 1. При уровне доверия $\rho = 0,9$ и $|\theta| < 1$, используя (2.22), получаем

Таблица 1. Усредненные оценки параметра $AR(1)$ ($h = 1000$)

θ	$\hat{\theta}_{\tau(h)}$	θ_n	$\bar{\tau}$	θ	$\hat{\theta}_{\tau(h)}$	θ_n	$\bar{\tau}$
-1,0	-1,0002	-0,9798	91,5450	0,3	0,2979	0,2979	1832,2460
-0,8	-0,7969	-0,7976	736,0060	0,4	0,3985	0,3987	1693,9700
-0,6	-0,6005	-0,5974	1291,7620	0,6	0,5966	0,5992	1293,2380
-0,4	-0,3972	-0,3998	1690,6190	0,8	0,7984	0,7973	729,5970
-0,3	-0,2994	-0,2991	1829,6720	1,0	0,9993	0,9817	94,9450

Таблица 2. Выборочное поведение отношения $|s(h)|/h$ и предела $\lim |s(h)|/h$ в зависимости от θ ($h = 300$)

θ	-1	-0,8	-0,6	-0,4	-0,2	0	0,2	0,4	0,6	0,8	1
$\lim s(h) /h$	1,98	1,58	1,18	0,79	0,39	0	0,39	0,79	1,18	1,58	1,98
$ s(h) /h$	1,74	1,48	1,25	0,66	0,54	0,10	0,34	0,82	1,26	1,65	1,81

Таблица 3. Усредненная ширина полуинтервала интервальной оценки (4.2)

$\theta \backslash h$	-1	-0,8	-0,6	-0,4	-0,2	0	0,2	0,4	0,6	0,8	1
500	0,099	0,113	0,151	0,228	0,464	20,1	0,465	0,227	0,152	0,113	0,100
1000	0,068	0,080	0,106	0,161	0,326	13,5	0,330	0,160	0,106	0,080	0,068
2000	0,047	0,056	0,075	0,113	0,226	27,9	0,224	0,112	0,075	0,056	0,047
3000	0,038	0,046	0,061	0,092	0,185	69,6	0,183	0,092	0,061	0,046	0,038
5000	0,029	0,035	0,047	0,071	0,140	10,7	0,143	0,071	0,047	0,035	0,029

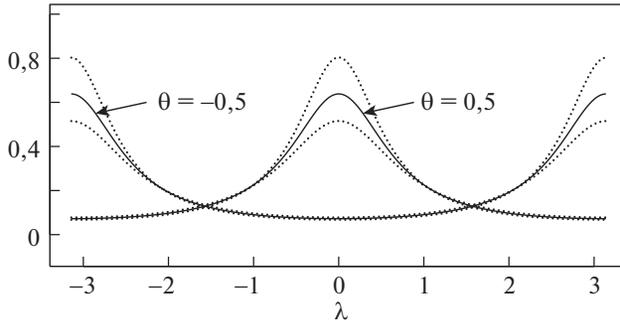


Рис. 2. Доверительные области для оценки спектральной плотности (4.1).

доверительный интервал для θ

$$(4.2) \quad \left(\hat{\theta}_{\tau(h)} - 3,94 \frac{h}{|s(h)|} \frac{1}{\sqrt{h}}, \hat{\theta}_{\tau(h)} + 3,94 \frac{h}{|s(h)|} \frac{1}{\sqrt{h}} \right).$$

Здесь множитель $|s(h)|/h$ выделен, учитывая равенство (2.24). В табл. 2 указаны значения модуля асимптотической кривой (2.24) из предложения 1 для различных значений θ , а также выборочное поведение этого отношения при $h = 300$.

В табл. 3 приводятся выборочные средние для ширины полуинтервала $\bar{\mu} = 3,94\sqrt{h}/|s(h)|$ доверительного интервала (4.2), полученные по 100 повторениям процедуры, для различных значений θ и h . Как видно из табл. 3, качество доверительных последовательных оценок существенно зависит от значений параметра θ , если параметр h , определяющий длительность наблюдений $\tau(h)$, остается постоянным. При значениях θ , близких к нулю, что соответствует ослаблению авторегрессионного сигнала x_k , требуется значительно увеличивать порог h , чтобы сохранить приемлемую точность оценивания θ . Это вполне объяснимо, если учесть в (4.2) асимптотическую зависимость (2.24) от θ в предложении 1. При отсутствии условия отделимости $|\theta|$ от нуля порог h , обеспечивающий приемлемое качество доверительного интервала (4.2), нельзя выбрать априорно, и его приходится подбирать эмпирически, повторяя процедуру оценивания (2.14) для больших значений h . На рис. 2 представлены доверительные области для оценки спектральной плотности (4.1), полученные с помощью доверительных оценок для значений $\theta = -0,5$ и $\theta = 0,5$ при $\rho = 0,9$ и $h = 5000$.

5. Заключение

В разделах 2 и 3 предложены и исследованы последовательные процедуры доверительного оценивания параметров гауссовской авторегрессии по наблюдениям с аддитивным гауссовским шумом. Последовательная оценка вектора неизвестных параметров для авторегрессии любого порядка строится на основе одной классической оценки Юла–Уокера, вводя специальное правило прекращения наблюдений и используя весовые коэффициенты в структуре

оценки. В теоремах 1 и 2 устанавливается, что предлагаемые последовательные оценки Юла–Уокера дают решение задачи идентификации зашумленной авторегрессии в неасимптотической постановке, позволяя определить длительность наблюдений в зависимости от требуемого качества оценивания. В разделе 4 рассматривается численный пример доверительного оценивания спектральной плотности зашумленной авторегрессии. Результаты могут использоваться в задачах идентификации и управления.

Авторы выражают признательность анонимным рецензентам за конструктивные замечания.

ПРИЛОЖЕНИЕ

Приведем теорему П.1 и теорему П.2 о свойствах остановленных мартингалов с условно-гауссовскими приращениями из публикаций [23] и [24], использованных в разделах 2 и 3 при выборе весовых коэффициентов (2.15) и (3.9), (3.10) в последовательных оценках Юла–Уокера (2.14) и (3.11), а также при определении длительности процедуры. Дадим доказательства некоторых технических результатов.

Теорема П.1. Пусть $(M_k, \mathcal{F}_k)_{k \geq 0}$ – квадратично интегрируемый мартингал [23] такой, что

а) его квадратическая характеристика удовлетворяет условию

$$P_\theta (\langle M \rangle_\infty = +\infty) = 1;$$

б) $\text{Law}(\Delta M_k | \mathcal{F}_{k-1}) = \mathcal{N}(0, \sigma_{k-1}^2)$, $k = 1, 2, \dots$, т.е. \mathcal{F}_{k-1} – условное распределение $\Delta M_k = M_k - M_{k-1}$ является гауссовским с параметрами 0 и $\sigma_{k-1}^2 = \mathbf{E}((\Delta M_k)^2 | \mathcal{F}_{k-1})$.

Для каждого h определим момент остановки

$$(П.1) \quad \tau = \tau(h) = \inf \left\{ n > 0 : \sum_{k=1}^n \sigma_{k-1}^2 \geq h \right\}, \quad \inf \{\emptyset\} = \infty,$$

и случайную величину

$$m(h) = \frac{1}{\sqrt{h}} \sum_{k=1}^{\tau(h)} \sqrt{\beta_k(h)} \Delta M_k, \quad \beta_k(h) = \begin{cases} 1, & \text{если } 1 \leq k < \tau(h), \\ \alpha(h), & \text{если } k = \tau(h); \end{cases}$$

$\alpha(h)$ – множитель, определяемый из уравнения

$$\sum_{k=1}^{\tau(h)-1} \sigma_{k-1}^2 + \alpha(h) \sigma_{\tau(h)-1}^2 = h.$$

Тогда для любого $h > 0$ величина $m(h)$ является стандартной гауссовской.

Доказательство леммы 1. Проверим, что $(\zeta_1(n), \mathcal{F}_n^{(1)})_{n \geq 3}$ — мартиггал. Измеримость $\zeta_1(n)$ относительно $\mathcal{F}_n^{(1)}$ следует из определений (2.10), (2.11). Покажем, что $\mathbf{E}(\zeta_1(n+1) | \mathcal{F}_n^{(1)}) = \zeta_n^{(1)}$. Рассмотрим, например, случай четного n . Пусть $n = 2l$, тогда

$$\zeta_1(2l+1) = \sum_{k=3}^{2l+1} \chi_{\{k \in T_1(2l+1)\}} y_{k-2} \xi_k.$$

Отсюда

$$\begin{aligned} \mathbf{E}(\zeta_1(2l+1) | \mathcal{F}_{2l-1}^{(1)}) &= \mathbf{E}\left(\sum_{k=3}^{2l+1} \chi_{\{k \in T_1(2l+1)\}} y_{k-2} \xi_k | \mathcal{F}_{2l-1}^{(1)}\right) = \\ &= \sum_{k=3}^{2l} \chi_{\{k \in T_1(2l+1)\}} y_{k-2} \xi_k + \chi_{\{(2l+1) \in T_1(2l+1)\}} y_{2l-1} \mathbf{E}(\xi_{2l+1} | \mathcal{F}_{2l-1}^{(1)}) = \zeta_1(2l). \end{aligned}$$

Проверка мартиггалности $\zeta_2(n)$ проводится аналогично.

Лемма 1 доказана.

Доказательство леммы 2. Покажем, что требуемый результат следует из теоремы П.1. Введем случайные процессы

$$(П.2) \quad M_n^{(1)} = \sum_{k=3}^n \chi_{\{k \in T_1(n)\}} y_{k-2} \tilde{\xi}_k, \quad M_n^{(2)} = \sum_{k=3}^n \chi_{\{k \in T_2(n)\}} y_{k-2} \tilde{\xi}_k, \quad n \geq 3.$$

Процессы $(M_n^{(1)}, \mathcal{F}_n^{(1)})_{n \geq 3}$ и $(M_n^{(2)}, \mathcal{F}_n^{(2)})_{n \geq 3}$ являются мартиггалами, причем

$$(П.3) \quad \begin{aligned} \mathbf{E}_\theta \left(\left(\Delta M_n^{(1)} \right)^2 | \mathcal{F}_{n-1}^{(1)} \right) &= y_{n-2}^2 \mathbf{E}_\theta \tilde{\xi}_n^2 \chi_{\{n \in T_1(n)\}}, \\ \mathbf{E}_\theta \left(\left(\Delta M_n^{(2)} \right)^2 | \mathcal{F}_{n-1}^{(2)} \right) &= y_{n-2}^2 \mathbf{E}_\theta \tilde{\xi}_n^2 \chi_{\{n \in T_2(n)\}}; \quad \mathbf{E}_\theta \tilde{\xi}_n^2 = \Delta^2 + \sigma^2. \end{aligned}$$

Проверим эти свойства для $M_n^{(1)}$. По определению $T_1(n)$ и $T_2(n)$ в (2.9) получаем

$$\begin{aligned} M_n^{(1)} &= \sum_{k=3}^{n-1} \chi_{\{k \in T_1(n)\}} y_{k-2} \tilde{\xi}_k + \chi_{\{n \in T_1(n)\}} y_{n-2} \tilde{\xi}_n = \\ &= M_{n-1}^{(1)} + \chi_{\{n \in T_1(n)\}} y_{n-2} \tilde{\xi}_n. \end{aligned}$$

Отсюда $\Delta M_n^{(1)} = \chi_{\{n \in T_1(n)\}} y_{n-2} \tilde{\xi}_n$. Поэтому

$$\mathbf{E}_\theta \left(\left(\Delta M_n^{(1)} \right)^2 | \mathcal{F}_{n-1}^{(1)} \right) = \chi_{\{n \in T_1(n)\}} \mathbf{E}_\theta \left(y_{n-2}^2 \tilde{\xi}_n^2 | \mathcal{F}_{n-1}^{(1)} \right).$$

Заметим, что если $n \in T_1(n)$, то $\mathcal{F}_{n-1}^{(1)} = \mathcal{F}_{n-2}^{(1)}$. Если, например, n нечетно, то число $n - 1$ четно. Следовательно, $m_1(n) = n$, $m_1(n - 1) = m_1(n - 2)$. По определению $\mathcal{F}_{n-1}^{(1)} = \mathcal{F}_{n-2}^{(1)}$. Так как ξ_n^2 не зависит от $\mathcal{F}_{n-2}^{(1)}$, то

$$\mathbf{E}_\theta \left(\left(\Delta M_n^{(1)} \right)^2 | \mathcal{F}_{n-1}^{(1)} \right) = y_{n-2}^2 \mathbf{E}_\theta \left(\xi_n^2 \chi_{\{n \in T_1(n)\}} \right).$$

Далее заметим, что мартингалы $(M_n^{(1)}, \mathcal{F}_n^{(1)})$ и $(M_n^{(2)}, \mathcal{F}_n^{(2)})$, определяемые равенствами (П.2), и их квадратические характеристики (П.3) удовлетворяют условиям теоремы П.1. При этом момент остановки $\tau_1(h)$ в (2.12) совпадает с (П.1). В силу теоремы П.1 случайные величины $\tilde{\zeta}_1(h)/\sqrt{h}$ и $\tilde{\zeta}_2(h)/\sqrt{h}$ являются стандартными гауссовскими. Лемма 2 доказана.

Доказательство предложения 2. Модель (2.1), (2.2) в векторной форме записывается в виде

$$(П.4) \quad X_k = AX_{k-1} + \nu_k, \quad Y_k = X_k + \zeta_k,$$

где $X_k = (x_k, \dots, x_{k-p+1})^\top$, $\nu_k = (\varepsilon_k, 0, \dots, 0)^\top$, $\zeta_k = (\eta_k, \dots, \eta_{k-p+1})^\top$,

$$A = \begin{pmatrix} \theta_1 & \dots & \theta_p \\ I_{p-1} & & 0 \end{pmatrix}.$$

Для анализа $\tau(h)$ потребуется асимптотическое поведение суммы

$$C_n^{(i)} = \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} Y_{k-p-1} Y_{k-p-1}^\top.$$

Подставив Y_k из (П.4), имеем разложение

$$(П.5) \quad C_n^{(i)} = U_n^{(i)} + V_n^{(i)} + R_n^{(i)};$$

$$U_n^{(i)} = \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} X_{k-p-1} X_{k-p-1}^\top,$$

$$V_n^{(i)} = \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} \zeta_{k-p-1} \zeta_{k-p-1}^\top,$$

$$R_n^{(i)} = \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} \zeta_{k-p-1} X_{k-p-1}^\top + \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} X_{k-p-1} \zeta_{k-p-1}^\top.$$

Учитывая (3.1), получаем

$$(П.6) \quad U_n^{(i)} = \sum_{j=1}^{d_i(n)} Z_j Z_j^\top, \quad Z_j = X_{(p+1)j+i-2}.$$

Последовательность $\{Z_j\}$ удовлетворяет векторному уравнению авторегрессии

$$Z_j = A^{p+1}Z_{j-1} + w_j, \quad w_j = \sum_{s=1}^{p+1} A^{p+1-s} \nu_{(p+1)(j-1)+i-2+s},$$

причем

$$\mathbf{E}w_j = 0, \quad \mathbf{E}w_j w_j^T = \sum_{l=0}^p A^l B (A^T)^l = \tilde{B}, \quad B = \|\sigma^2 \delta_{1,i} \delta_{1,j}\|.$$

Поскольку процесс Z_j устойчив, то (см., например, [2])

$$(II.7) \quad \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=1}^n Z_j Z_j^T = \tilde{F} \quad \text{п.н.};$$

$$\tilde{F} = \sum_{j \geq 0} A^{(p+1)j} \tilde{B} (A^T)^{(p+1)j} = \sum_{j \geq 0} A^j B (A^T)^j =: F.$$

Из (II.6) и (II.7) с учетом (3.1) находим

$$(II.8) \quad \lim_{n \rightarrow \infty} \frac{U_n^{(i)}}{n} = \lim_{n \rightarrow \infty} \frac{U_n^{(i)}}{d_i(n)} \frac{d_i(n)}{n} = \frac{F}{p+1} \quad \text{п.н.}$$

Далее непосредственно проверяется, что

$$\lim_{n \rightarrow \infty} \frac{V_n^{(i)}}{n} = \frac{\Delta^2}{p+1} I_p, \quad \lim_{n \rightarrow \infty} \frac{R_n^{(i)}}{n} = 0 \quad \text{п.н.}$$

Отсюда и из (II.5), (II.8) получаем

$$\lim_{n \rightarrow \infty} n^{-1} C_n^{(i)} = (p+1)^{-1} (F + \Delta^2 I_p).$$

Так как

$$\sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} y_{k-p-l}^2 = \langle C_n^{(i)} \rangle_{ll},$$

то

$$(II.9) \quad \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} y_{k-p-l}^2 = \frac{1}{p+1} (\langle F \rangle_{11} + \Delta^2), \quad l = \overline{1, p}.$$

Найдем теперь асимптотику для моментов остановки $\tau_1^{(i)}(h)$. По определению $\tau_1^{(i)}(h)$ в (3.7) имеем

$$\sum_{k=2p+1}^{\tau_1^{(i)}(h)-1} \chi_{\{k \in T_i(\tau_1^{(i)})\}} y_{k-p-1}^2 < \frac{h}{\Delta^2 + \sigma^2} \leq \sum_{k=2p+1}^{\tau_1^{(i)}(h)} \chi_{\{k \in T_i(\tau_1^{(i)})\}} y_{k-p-1}^2.$$

Отсюда, используя (П.9), получаем

$$\lim_{h \rightarrow \infty} \frac{h}{(\Delta^2 + \sigma^2) \tau_1^{(i)}(h)} = \frac{1}{p+1} (\langle F \rangle_{11} + \Delta^2),$$

т.е.

$$\lim_{h \rightarrow \infty} \frac{\tau_1^{(i)}(h)}{h} = \frac{p+1}{(\Delta^2 + \sigma^2) (\langle F \rangle_{11} + \Delta^2)}.$$

Аналогично находим, что

$$(П.10) \quad \lim_{h \rightarrow \infty} \frac{\tau_l^{(i)}(h)}{h} = \frac{l(p+1)}{(\Delta^2 + \sigma^2) (\langle F \rangle_{11} + \Delta^2)}, \quad 2 \leq l \leq p.$$

Принимая во внимание (3.8), получаем соотношение (3.15) для длительности последовательной процедуры. Далее рассмотрим асимптотическое поведение матрицы $G(h)$, определенной в (3.13). Для устойчивого процесса (2.1) справедливо свойство [2]

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=2p+1}^n Y_{k-p-1} Y_{k-1}^T = F(A^T)^p.$$

Аналогично (П.8) устанавливается, что матрица

$$D^{(i)}(n) = \sum_{k=2p+1}^n \chi_{\{k \in T_i(n)\}} Y_{k-p-1} Y_{k-1}^T$$

удовлетворяет предельному соотношению

$$(П.11) \quad \lim_{n \rightarrow \infty} \frac{1}{n} D^{(i)}(n) = \frac{1}{p+1} F(A^T)^p.$$

Далее, подставляя коэффициенты (3.10) в элемент матрицы (3.13) и учитывая, что $y_{k-p-l} = \langle Y_{k-p-1} \rangle_l$, $y_{k-s} = \langle Y_{k-1} \rangle_s$, получаем

$$(П.12) \quad \langle G(h) \rangle_{l,s} = \sum_{i=1}^{p+1} \sum_{k=\tau_{l-1}^{(i)}(h)+1}^{\tau_l^{(i)}(h)} \chi_{\{k \in T_i(k)\}} \sqrt{\beta_{l,k}^{(i)}} \langle Y_{k-p-1} Y_{k-1}^T \rangle_{l,s}.$$

Учитывая определение коэффициентов (3.9), заметим, что внутренняя сумма с точностью до одного слагаемого, отвечающего моменту $\tau_l^{(i)}(h)$, совпадает с суммой

$$\begin{aligned} & \sum_{k=\tau_{l-1}^{(i)}(h)+1}^{\tau_l^{(i)}(h)} \chi_{\{k \in T_i(k)\}} \langle Y_{k-p-1} Y_{k-1}^T \rangle_{l,s} = \\ & = \left\langle D^{(i)} \left(\tau_l^{(i)}(h) \right) \right\rangle_{l,s} - \left\langle D^{(i)} \left(\tau_{l-1}^{(i)}(h) \right) \right\rangle_{l,s}. \end{aligned}$$

Отсюда и из (П.10), (П.11) следует, что

$$\begin{aligned}
& \lim_{h \rightarrow \infty} \frac{1}{h} \sum_{k=\tau_{l-1}^{(i)}(h)+1}^{\tau_l^{(i)}(h)} \chi_{\{k \in T_i(k)\}} \langle Y_{k-p-1} Y_{k-1}^T \rangle_{l,s} = \\
& = \lim_{h \rightarrow \infty} \left(\frac{\tau_l^{(i)}(h)}{h} \frac{\langle D^{(i)}(\tau_l^{(i)}(h)) \rangle_{l,s}}{\tau_l^{(i)}(h)} - \frac{\tau_{l-1}^{(i)}(h)}{h} \frac{\langle D^{(i)}(\tau_{l-1}^{(i)}(h)) \rangle_{l,s}}{\tau_{l-1}^{(i)}(h)} \right) = \\
& = \frac{\langle F(A^T)^p \rangle_{l,s}}{(\Delta^2 + \sigma^2)(\langle F \rangle_{11} + \Delta^2)}.
\end{aligned}$$

Используя это равенство в (П.12), получаем

$$\lim_{h \rightarrow \infty} \frac{G(h)}{h} = \frac{(p+1)F(A^T)^p}{(\Delta^2 + \sigma^2)(\langle F \rangle_{11} + \Delta^2)}.$$

Предложение 2 доказано.

Теорема П.2. Пусть заданы [24]:

1) вероятностное пространство $(\Omega, \mathcal{F}, \mathbb{P})$ с фильтрацией $(\mathcal{F})_{k \geq 0}$;

2) семейство $(M_k^{(l)}, \mathcal{F}_k)_{k \geq 0}$, $l = \overline{1, p}$, квадратично интегрируемых мар-

тингалов с квадратическими характеристиками $\{\langle M^{(l)} \rangle_n\}_{n \geq 1}$, $l = \overline{1, p}$, та-
кими, что

а) $\mathbb{P}(\langle M^{(l)} \rangle_\infty = +\infty) = 1$, $l = \overline{1, p}$;

б) $\text{Law}(\Delta M_k^{(l)} | \mathcal{F}_{k-1}) = \mathcal{N}(0, \sigma_l^2(k-1))$, $k = 1, 2, \dots, l = \overline{1, p}$, т.е. \mathcal{F}_{k-1} —
условное распределение приращения $\Delta M_k^{(l)} = M_k^{(l)} - M_{k-1}^{(l)}$ является гауссов-
ским с параметрами 0 и $\sigma_l^2(k-1) = \mathbf{E} \left(\left(\Delta M_k^{(l)} \right)^2 | \mathcal{F}_{k-1} \right)$.

Для каждого $h > 0$ определим момент остановки

$$\begin{aligned}
\tau_l = \tau_l(h) &= \inf \left\{ n > \tau_{l-1}(h) : \sum_{k=\tau_{l-1}+1}^n \sigma_l^2(k-1) \geq h \right\}, \quad l = \overline{1, p}, \\
\tau_0 = \tau_0(h) &= 0, \quad \inf\{\emptyset\} = +\infty,
\end{aligned}$$

и случайные величины

$$\begin{aligned}
m_l(h) &= \frac{1}{\sqrt{h}} \sum_{k=\tau_{l-1}+1}^{\tau_l} \sqrt{\beta_k(h, l)} \Delta M_k^{(l)}, \quad l = \overline{1, p}; \\
\beta_k(h, l) &= \begin{cases} 1, & \text{если } \tau_{l-1}(h) < k < \tau_l(h), \\ \alpha_l(h), & \text{если } k = \tau_l(h); \end{cases}
\end{aligned}$$

$\alpha_l(h)$, $l = \overline{1, p}$, — корректирующие множители, определяемые из уравнений

$$\sum_{k=\tau_{l-1}+1}^{\tau_l-1} \sigma_l^2(k-1) + \alpha_l(h)\sigma_l^2(\tau_l(h)-1) = h.$$

Тогда для любого $h > 0$ случайный вектор $m(h) = (m_1(h), \dots, m_p(h))^T$ имеет стандартное нормальное распределение, т.е. $m(h) \sim \mathcal{N}(0, I_p)$, где I_p — единичная матрица размерности p .

Доказательство леммы 3. Запишем (3.4) в виде

$$\begin{aligned} M_l^{(i)}(n) &= \sum_{k=2p+1}^{n-1} \chi_{\{k \in T_i(n)\}} y_{k-p-l} \tilde{\xi}_k + \chi_{\{n \in T_i(n)\}} y_{n-p-l} \tilde{\xi}_n = \\ &= M_l^{(i)}(n-1) + \chi_{\{n \in T_i(n)\}} y_{n-p-l} \tilde{\xi}_n, \end{aligned}$$

т.е.

$$(II.13) \quad \Delta M_l^{(i)}(n) = \chi_{\{n \in T_i(n)\}} y_{n-p-l} \tilde{\xi}_n.$$

Отсюда имеем

$$(II.14) \quad \begin{aligned} \sigma_{i,l}^2(n-1) &= \mathbf{E}_\theta \left(\left(\Delta M_l^{(i)}(n) \right)^2 \middle| \mathcal{F}_{n-1}^{(i)} \right) = \\ &= \chi_{\{n \in T_i(n)\}} \mathbf{E}_\theta \left(y_{n-p-l}^2 \tilde{\xi}_n^2 \middle| \mathcal{F}_{n-1}^{(i)} \right), \end{aligned}$$

где $\mathcal{F}_{n-1}^{(i)}$ — σ -алгебра, определенная в (3.2). Далее заметим, что если $n \in T_i(n)$, то $t_i(n) = n$, $t_i(n-1) = n-p-1$. Поэтому $\mathcal{F}_{n-1}^{(i)} = \mathcal{F}_{n-p-1}^{(i)}$ и случайная величина y_{n-p-l}^2 измерима относительно $\mathcal{F}_{n-p-1}^{(i)}$, а $\tilde{\xi}_n$ не зависит от $\mathcal{F}_{n-p-1}^{(i)}$. В силу (3.6) и гауссовости $\tilde{\xi}_n$ приращение (II.13) имеет $\mathcal{F}_{n-1}^{(i)}$ — условно-гауссовское распределение, причем, учитывая (II.14), $\sigma_{i,l}^2(n-1) = \chi_{\{n \in T_i(n)\}} (\sigma^2 + \Delta^2) y_{n-p-l}^2$. Лемма 3 доказана.

Доказательство теоремы 2. Подставляя y_k из (2.1) в (3.12) и учитывая (3.13), получаем

$$(II.15) \quad \begin{aligned} \vartheta_l(h) &= \sum_{k=2p+1}^{\tau(h)} \gamma_{l,k} y_{k-p-l} \left(\sum_{s=1}^p y_{k-s} \theta_s + \xi_k \right) = \\ &= \sum_{s=1}^p \left(\sum_{k=2p+1}^{\tau(h)} \gamma_{l,k} y_{k-p-l} y_{k-s} \right) \theta_s + \sum_{k=2p+1}^{\tau(h)} \gamma_{l,k} y_{k-p-l} \xi_k = \langle G(h)\theta \rangle_l + \zeta_l(n), \end{aligned}$$

где

$$(II.16) \quad \zeta_l(h) = \sum_{k=2p+1}^{\tau(h)} \gamma_{l,k} y_{k-p-l} \xi_k, \quad 1 \leq l \leq p.$$

В векторной форме система уравнений (П.15) имеет вид

$$(П.17) \quad \vartheta(h) = G(h)\theta + \zeta(h); \quad \zeta(h) = (\zeta_1(h), \dots, \zeta_p(h))^T.$$

Подставляя весовые коэффициенты (3.10) в (П.16), получаем

$$(П.18) \quad \zeta_l(h) = \sum_{i=1}^{p+1} \zeta_l^{(i)}(h),$$

$$\zeta_l^{(i)}(h) = \sum_{k=\tau_{l-1}^{(i)}(h)+1}^{\tau_l^{(i)}(h)} \sqrt{\beta_{l,k}^{(i)}(h)} \chi_{\{k \in T_i(k)\}} y_{k-p-l} \xi_k.$$

Введя векторы $\zeta^{(i)}(h) = (\zeta_1^{(i)}(h), \dots, \zeta_p^{(i)}(h))^T$, $i = \overline{1, p+1}$, имеем разложение

$$\zeta(h) = \sum_{i=1}^{p+1} \zeta^{(i)}(h).$$

Далее рассмотрим вектор

$$(П.19) \quad m^{(i)}(h) = (m_1^{(i)}(h), \dots, m_p^{(i)}(h))^T := \varkappa_\theta h^{-1/2} \zeta^{(i)}(h).$$

Учитывая (3.5), координаты этого вектора запишем в виде

$$(П.20) \quad m_l^{(i)}(h) = \frac{1}{\sqrt{h}} \sum_{k=\tau_{l-1}^{(i)}(h)+1}^{\tau_l^{(i)}(h)} \sqrt{\beta_{l,k}^{(i)}(h)} \Delta M_l^{(i)}(k).$$

Используя (П.17), (П.18), (П.19), получаем

$$(П.21) \quad G(h)(\theta^*(h) - \theta) = \frac{\sqrt{h}}{\varkappa_\theta} \sum_{i=1}^{p+1} m^{(i)}(h).$$

К векторам (П.19) применима теорема П.2, поскольку мартингалы (3.5) и моменты остановки (3.7) в (П.20) удовлетворяют в силу леммы 3 всем условиям этой теоремы. Согласно теореме П.2 все векторы (П.19) имеют стандартное p -мерное нормальное распределение $\mathcal{N}(0, I_p)$. Отсюда следует, что квадраты норм этих векторов $\|m^{(i)}(h)\|^2$, $i = \overline{1, p+1}$, имеют χ^2 распределение с p степенями свободы, т.е. $P(\|m^{(i)}(h)\| > a) = 1 - \Phi_p(a)$, где $\Phi_p(a)$ — функция распределения, определенная в (3.14). Построим доверительную область для θ по заданной доверительной вероятности. Из (П.21) имеем неравенство

$$(П.22) \quad \frac{\varkappa_\theta}{\sqrt{h}} \|G(h)(\theta^*(h) - \theta)\| \leq \sum_{i=1}^{p+1} \|m^{(i)}(h)\|.$$

Используя оценку

$$(П.23) \quad \begin{aligned} P_{\theta} \left(\sum_{i=1}^{p+1} \|m^{(i)}(h)\| > \mu \right) &\leq \sum_{i=1}^{p+1} P_{\theta} \left(\|m^{(i)}(h)\| > \frac{\mu}{p+1} \right) = \\ &= (p+1) P_{\theta} \left(\|m^{(1)}(h)\| > \frac{\mu}{p+1} \right) = (p+1) \left(1 - \Phi_p \left(\frac{\mu}{p+1} \right) \right), \end{aligned}$$

найдем μ по заданному ρ из уравнения $(p+1)(1 - \Phi_p(\mu/(p+1))) = 1 - \rho$. Корень этого уравнения μ^* определяется формулой (3.14). Используя (П.22), (П.23), получаем

$$P_{\theta} \left(\frac{\varkappa_{\theta}}{\sqrt{h}} \|G(h)(\theta^*(h) - \theta)\| \leq \mu^* \right) \geq \rho.$$

Теорема 2 доказана.

СПИСОК ЛИТЕРАТУРЫ

1. *Ljung L., Söderstrom T.* Theory and Practice of recursive Identification. Cambridge, Massachusetts: London The MIT Press, 1986.
2. *Андерсон Т.* Статистический анализ временных рядов. М.: Мир, 1976.
3. *Brockwell P.J., Davis R.A.* Time Series: Theory and Methods. N.Y.: Springer Science+Business Media, 1991.
4. *Васильев В.А., Добровидов А.В., Кошкин Г.М.* Непараметрическое оценивание функционалов от распределений стационарных последовательностей. М.: Наука, 2004.
5. *Кашковский Д.В., Конев В.В.* Последовательная идентификация линейной динамической системы со случайными параметрами // *АиТ.* 2008. № 8. С. 82–95.
Kashkovskii D.V., Konev V.V. Successive Identification of the Random-parameter Linear Dynamic Systems // *Autom. Remote Control.* 2008. V. 69. No. 8. P. 1344–1356.
6. *Konev V.V., Pergamenshchikov S.M.* Robust Model Selection for a Semimartingale Continuous Time Regression from Discrete Data // *Stoch. Process. Their Appl.* 2015. V. 125. No. 1. P. 294–326.
7. *Емельянова Т.В., Конев В.В.* О последовательном оценивании параметров тригонометрической регрессии с непрерывным временем // *АиТ.* 2016. № 6. С. 61–80.
Emel'yanova T.V., Konev V.V. On Sequential Estimation of the Parameters of Continuous-time Trigonometric Regression // *Autom. Remote Control.* 2016. V. 77. No. 6. P. 992–1008.
8. *Себер Дж.* Линейный регрессионный анализ. М.: Мир, 1980.
9. *Новиков А.А.* Последовательное оценивание параметров диффузионных процессов // *Теория вероятн. и ее примен.* 1971. Т. 16. № 2. С. 394–396.
10. *Липцер Р.Ш., Ширяев А.Н.* Статистика случайных процессов. М.: Наука, 1974.
11. *Lai T.L., Siegmund D.* Fixed Accuracy Estimation of an Autoregressive Parameter // *Ann. Statist.* 1983. V. 11. P. 478–485.
12. *Galtchouk L., Konev V.* On Asymptotic Normality of Sequential LS-estimate for Unstable Autoregressive Process AR(2) // *J. Multivariate Anal., Academic Press.* 2010. V. 101. No. 10. P. 2616–2636.

13. *Борисов В.З., Конеv В.В.* О последовательном оценивании параметров дискретных процессов // *АиТ.* 1977. № 10. С. 58–64.
Borisov V.Z., Konev V.V. On Sequential Parameter Estimation in Discrete-time Processes // *Autom. Remote Control.* 1977. V. 38. No. 10. P. 1475–1480.
14. *Воробейчиков С.Э., Конеv В.В.* О последовательной идентификации стохастических систем // *Изв. АН СССР. Техническая кибернетика.* 1980. № 4. С. 91–98.
15. *Конеv В.В., Пергаменищиков С.М.* Последовательные планы идентификации параметров динамических систем // *АиТ.* 1981. № 7. С. 84–92.
Konev V.V., Pergamenshchikov S.M. Sequential Plans of Parameter Identification in Dynamic Systems // *Autom. Remote Control.* 1981. V. 42. No. 7. Part 1. P. 917–924.
16. *Васильев В.А., Конеv В.В.* Последовательное оценивание параметров динамических систем при неполном наблюдении // *Изв. АН СССР. Техническая кибернетика.* 1982. № 6. С. 145–154.
17. *Xia Y., Zheng W.X.* Novel Parameter Estimation of Autoregressive Signals in the Presence of Noise // *Automatica.* 2015. V. 62. P. 98–105.
18. *Куликова М.В.* Максимально правдоподобное оценивание линейных стохастических систем в классе последовательных квадратно-корневых ортогональных методов фильтрации // *АиТ.* 2011. № 4. С. 99–120.
Kulikova M.V. Maximum Likelihood Estimation of Linear Stochastic Systems in the Class of Sequential Square-root Orthogonal Filtering Methods // *Autom. Remote Control.* 2011. V. 72. No. 4. P. 766–786.
19. *Diversi R., Guidorzi R., Soverini U.* Identification of Autoregressive Models in the Presence of Additive Noise // *Int. J. Adapt. Control Signal Process.* 2008. V. 22. No. 5. P. 465–481.
20. *Labarre D., Grivel E., Berthoumieu Y., Todini E., Najim M.* Consistent Estimation of Autoregressive Parameters from Noisy Observations Based on Two Interacting Kalman Filters // *Signal Processing.* 2006. V. 86. No. 10. P. 2863–2876.
21. *Zheng W.X.* Fast Identification of Autoregressive Signals from Noisy Observations // *IEEE Trans. Circuits and Systems–II: Express Briefs.* 2005. V. 52. No. 1. P. 43–48.
22. *Pagano M.* Estimation of Models of Autoregressive Signal Plus White Noise // *Ann. Stat.* 1974. V. 2. No. 1. P. 99–108.
23. *Конеv В.В.* Об одном свойстве мартингалов с условно-гауссовскими приращениями и его применении в теории неасимптотических выводов // *ДАН.* 2016. Т. 471. № 5. С. 523–527.
Konev V.V. On One Property of Martingales with Conditionally Gaussian Increments and Its Application in the Theory of Nonasymptotic Inference // *Doklady Mathematics.* 2016. V. 471. No. 5. P. 523–527.
24. *Konev V., Nazarenko B.* Sequential Fixed Accuracy Estimation for Nonstationary Autoregressive Processes // *AISM.* 2020. V. 72. No. 1. P. 235–264.
25. *Воробейчиков С.Э., Конеv В.В.* О доверительном последовательном оценивании параметров стохастических динамических систем с условно-гауссовскими шумами // *АиТ.* 2017. № 10. С. 90–108.
Vorobeichikov S.E., Konev V.V. On Sequential Confidence Estimation of Parameters of Stochastic Dynamical Systems with Conditionally Gaussian Noises // *Autom. Remote Control.* 2017. V. 78. No. 10. P. 1803–1818.
26. *Куржанский А.Б., Фурасов В.Д.* Идентификация билинейных систем. Гарантированные псевдоэллипсоидальные оценки // *АиТ.* 2000. № 1. С. 41–53.
Kurzhaniskii A.B., Furasov V.D. Identification of Bilinear Systems. Guaranteed Pseudoellipsoidal Estimates // *Autom. Remote Control.* 2000. V. 61. No. 1. P. 38–49.

27. *Konev V.V., Pergamenshchikov S.M.* General Model Selection Estimation of a Periodic Regression with a Gaussian Noise // AISM. 2010. V. 62. No. 6. P. 1083–1111.
28. *Ширяев А.Н.* Статистический последовательный анализ. М.: Наука, 1976.
29. *Tartakovsky A., Nikiforov I., Basseville M.* Sequential Analysis: Hypothesis Testing and Change-point Detection. Chapman & Hall/CRC Press, 2015.

Статья представлена к публикации членом редколлегии А.И. Кибзуном.

Поступила в редакцию 03.02.2020

После доработки 02.04.2020

Принята к публикации 15.01.2021

© 2021 г. А.Ю. ПОПКОВ, канд. техн. наук (apopkov@isa.ru)
(Федеральный исследовательский центр “Информатика и управление” РАН,
Москва)

РАНДОМИЗИРОВАННОЕ МАШИННОЕ ОБУЧЕНИЕ НЕЛИНЕЙНЫХ МОДЕЛЕЙ С ПРИМЕНЕНИЕМ К ПРОГНОЗИРОВАНИЮ РАЗВИТИЯ ЭПИДЕМИЧЕСКОГО ПРОЦЕССА¹

Развивается дискретный подход в теории рандомизированного машинного обучения, ориентированный на применение к нелинейным моделям. Формулируется задача энтропийного оценивания распределений вероятностей и шумов измерений для дискретных нелинейных моделей. Рассматриваются вопросы, связанные с применением таких моделей к задачам прогнозирования, в частности проблеме генерации энтропийно-оптимальных распределений. Демонстрация предложенных методов проводится на решении задачи прогнозирования общего количества инфицированных SARS-CoV-2 в Германии в 2020 г.

Ключевые слова: рандомизированное машинное обучение, энтропия, энтропийное оценивание, прогнозирование, рандомизированное прогнозирование, COVID-19, SARS-CoV-2.

DOI: 10.31857/S0005231021060064

1. Введение

Большое количество различных процессов, наблюдаемых в разных областях человеческой деятельности, природе и т.п., не могут быть эффективно описаны линейными математическими моделями. В этой связи разработка и развитие общих подходов нелинейного моделирования являются актуальными задачами. Однако необходимо отметить, что разработка и применение нелинейных моделей для конкретных задач сопряжены с определенными трудностями, связанными как с их обучением с использованием реальных данных, так и с выбором структуры модели.

Машинное обучение как раздел прикладной науки интегрирует в себе большое количество методов и подходов, накопленных в различных научных дисциплинах [1, 2], большой вклад в которые был сделан в таких направлениях, как теория вероятностей и математическая статистика [3, 4]. Методы машинного обучения успешно применяются к различным задачам, в частности к задачам классификации и регрессии, которые относятся к задачам обучения с учителем [1], основной особенностью которых является идея настройки (обучения) параметров требуемой модели с использованием реальных данных. Настраиваемую (обученную) модель предполагается использовать

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 20-07-00683).

для прогнозирования, т.е. получать от нее ответ, предъявляя ей входные данные, не участвующие ранее в обучении.

Этот подход является эффективным и известен, по крайней мере, с 1960-х гг. [5–8]. Однако большинство подходов, разработанных в данной области, ориентировано на использование линейных моделей. В частности, одним из эффективных подходов к решению задач классификации является линейная классификация, состоящая в поиске линейной разделяющей гиперплоскости и применяемая, например, в широко используемом методе опорных векторов [9]. Задачи восстановления регрессии давно и широко применяются в эконометрике [3], и также большинство из них в этой области ориентировано на восстановление линейной регрессии. Основные причины такого положения состоят в основном в том, что, во-первых, линейные модели легче исследовать и интерпретировать, во-вторых, численные и аналитические решения линейных задач возможно получать либо абсолютно точно (аналитически), либо с высокой точностью численно и, в-третьих, многие практические задачи часто можно свести к линейным постановкам, а следовательно, получать более качественное их решение с учетом отмеченных ранее свойств.

В то же время в некоторых прикладных задачах классификации и регрессии проявляются различные нелинейные эффекты, которые необходимо каким-то образом решать. Эффективным подходом к этой проблеме является, например, ядерный подход, состоящий в нелинейном переходе в пространство высокой размерности с последующим применением в нем уже линейного метода для решения задачи классификации или регрессии [1]. Этот подход демонстрирует свою эффективность во многих прикладных задачах и приводит к появлению различных «ядерных» версий известных линейных методов. Тем не менее часто вопрос о выборе модели (точнее, ядерной функции) продолжает возникать при применении этих методов на практике. Кроме этого, всякий переход в пространство высокой размерности при отсутствии большого количества данных в этом пространстве неизбежно приводит к нежелательным эффектам таким, например, как переобучение, а также ряду других.

Другим подходом борьбы с нелинейными эффектами предлагают методы, в которых не выделяется модель в явном виде, например методы, основанные на деревьях решений [10], нейросетевые модели [1, 11, 12] и ряд других.

Таким образом, с одной стороны, наблюдается ситуация, когда в практических задачах анализа данных явным образом наблюдается наличие “нелинейности”, требующей применения нелинейных моделей, с другой стороны, существующие методы, части используемые на практике, недостаточно развиты для эффективного решения задачи в нелинейном случае и требуют переформулирования или какой-либо адаптации задачи к их применению.

В настоящей работе предлагается универсальный подход к работе с нелинейными моделями в задачах анализа данных, в частности в задачах обучения регрессионных моделей. Этот подход основан на теории рандомизированного машинного обучения [13], основная идея которой состоит в искусственной рандомизации параметров модели, что позволяет перейти от модели с детерминированными параметрами к модели со случайными параметрами и определять в результате обучения не их точечные оценки, а их распреде-

ления. Распределения определяются таким образом, чтобы они доставляли максимум энтропийному функционалу при условии баланса со средним выходом модели.

Основным достоинством предлагаемого подхода является независимость от реальных характеристик используемых данных. Для корректного применения метода не требуется подтверждения или предположения о нормальности данных (или иных их вероятностных свойствах), а полученные в результате обучения распределения вычислены в условиях максимальной энтропии, таким образом отражая наиболее “плохой” сценарий развития исследуемого процесса, соответствующий его максимальной неопределенности. Данные свойства метода энтропийного оценивания восходят к работам Больцмана [14], Джейнса [15, 16], Шеннона [17]. Еще одной важной особенностью метода является получение энтропийно-оптимальных распределений шумов, содержащихся в данных, вместе с оптимальными распределениями параметров. Это свойство существенно отличает метод от классических подходов, в которых делаются различные предположения о характеристиках шумов и данных.

Оценки характеристик модели используются для прогнозирования моделируемого процесса. Стандартный подход к прогнозированию состоит в применении модели с точечными оценками параметров, полученных оцениванием по реальным данным, для неизвестных (“будущих”) точек данных [18–20].

С учетом того, что подобрать модель точно под данные не представляется возможным, вводится предположение о стохастической природе данных, точнее о том, что механизм порождения наблюдаемых данных, который неизвестен, содержит стохастическую компоненту. Следствием этого предположения становится то, что в наблюдаемых данных есть как детерминированная, так и стохастическая компонента, вероятностные характеристики которой неизвестны. Фактически, на моделирование этой стохастической компоненты и направлен весь разработанный к настоящему времени математический аппарат, применяемый в данной области. Существенную часть этого подхода составляет предположение о нормальности случайных компонент данных, из которого становится возможным установить свойства получаемых оценок параметров моделей.

Например, известный и широко применяемый метод максимума правдоподобия основан на идее максимизации распределения параметров модели при условии наблюдаемых данных: необходимо определить такие значения параметров, при которых будет достигаться максимум функции правдоподобия данных. Это означает, что вероятность того, что при найденных значениях параметров будут наблюдаться существующие данные, максимальна. Предположение о нормальности (или о наличии другого известного закона распределения) случайных компонент данных является основой этого метода, без этого предположения будет невозможно получить функцию правдоподобия в аналитическом виде [3].

Для вычисления оценок, максимизирующих правдоподобие, часто применяется метод наименьших квадратов (МНК), однако получаемые им оценки соответствуют оценкам максимального правдоподобия только для линейных

моделей. В нелинейном случае определение свойств этих оценок связано с существенными трудностями.

Предположение о нормальности стохастических компонент моделей, а значит и данных, с которыми ассоциирована модель, очевидно, не всегда является вполне корректным, так как не всегда удается выяснить или доказать требуемые факты о вероятностных характеристиках данных по имеющимся в наличии данным.

Энтропийно-оптимальные распределения, полученные на этапе обучения модели, могут быть использованы для прогнозирования несколькими способами, в частности, они могут быть сгенерированы для получения ансамбля выходов модели с последующим его анализом.

Работа посвящена развитию метода энтропийно-рандомизированного обучения и прогнозирования для нелинейным моделей с дискретными параметрами. Переход от непрерывных к дискретным моделям позволяет преодолеть трудности использования непрерывных моделей в условиях большого количества переменных, которые приводят к проблеме вычисления многомерных интегралов, произвести которое точно (аналитически) невозможно, а численное решение сопряжено с существенными вычислительными трудностями.

Предлагаемый в работе подход демонстрируется на примере задачи прогнозирования общего количества инфицированных в результате развития эпидемии COVID-19 в Германии. Проводится сравнение предлагаемого подхода с нелинейным методом наименьших квадратов [18, 21].

2. Нелинейная дискретная рандомизированная модель

Рассмотрим объект с n скалярными входами x_i , $i = \overline{1, n}$ и выходом \hat{y} , преобразование которых описывается в общем случае нелинейной функцией

$$(2.1) \quad \hat{y} = \Phi(\mathbf{x}, \mathbf{a}),$$

где $\mathbf{x} = (x_1, \dots, x_n)$ — вектор входов, $\mathbf{a} = (a_1, \dots, a_d)$ — вектор параметров модели.

Выход модели измеряется с некоторым шумом ξ , действующим аддитивно на выход, приводя к модели следующего вида:

$$(2.2) \quad v = \hat{y} + \xi = \Phi(\mathbf{x}, \mathbf{a}) + \xi.$$

Предположим, что значения каждого параметра сосредоточены на интервале $\mathcal{A}_k = [a_k^-, a_k^+]$, $k = \overline{1, d}$, и выход модели измеряется с шумом ξ_j , значения которого сосредоточены на интервале $\Xi_j = [\xi_j^-, \xi_j^+]$ для каждого заданного входа \mathbf{x}_j , $j = \overline{1, m}$.

Параметры a_k реализуются дискретной случайной величиной с M значениями на интервале \mathcal{A}_k , приводя к следующим распределениям:

$$(2.3) \quad a_{k\ell} \in \mathcal{A}_k, \quad p_{k\ell} \in [0, 1], \quad k = \overline{1, d}, \quad \ell = \overline{1, M},$$

где $a_{k\ell}$ — значения случайной величины, а $p_{k\ell}$ — вероятности их реализации.

Шумы измерений выхода реализуются дискретной случайной величиной ξ_j с L значениями на интервале Ξ_j для каждого входа x_j . Измерения выхода производятся независимо друг от друга, таким образом при m измерениях приводя к следующим распределениям:

$$(2.4) \quad \xi_{jh} \in \Xi_j, \quad q_{jh} \in [0, 1], \quad j = \overline{1, m}, \quad h = \overline{1, L},$$

где ξ_{jh} — значения случайной величины, а q_{jh} — вероятности их реализации.

С учетом m измерений получаем итоговый вид модели (2.2):

$$(2.5) \quad \mathbf{v} = \hat{\mathbf{y}} + \boldsymbol{\xi} = \Phi(\mathbf{x}_j, \mathbf{a}) + \boldsymbol{\xi}, \quad j = \overline{1, m},$$

где $\mathbf{v} = (v_1, \dots, v_m)$ — вектор измеренного выхода модели, $\boldsymbol{\xi} = (\xi_1, \dots, \xi_m)$ — вектор шумов, $\hat{\mathbf{y}} = (y_1, \dots, y_m)$ — вектор выхода модели.

Распределения параметров и шумов измерений модели подлежат оцениванию с использованием реальных данных об измерениях выхода объекта, модель которого описывается (2.5).

3. Обучение модели с использованием реальных данных

Рандомизированное машинное обучение базируется на принципах энтропийного оценивания параметров модели и шумов измерений ее выхода. Энтропийно-оптимальные распределения отражают наиболее неопределенную ситуацию, что в условиях полного отсутствия информации о реальных характеристиках является единственным доступным в этих условиях решением [13, 22, 23].

Для вычисления оптимальных распределений требуется решить задачу условной максимизации энтропии распределений параметров и шумов измерений при условиях нормировки соответствующих распределений и выполнении условий на баланс среднего выхода модели с измерением выхода объекта. Задача формулируется следующим образом:

$$(3.1) \quad H(P, Q) = - \sum_{k=1}^d \sum_{\ell=1}^M p_{k\ell} \ln p_{k\ell} - \sum_{j=1}^m \sum_{h=1}^L q_{jh} \ln q_{jh} \rightarrow \max_{P, Q},$$

где P и Q — распределения параметров и шумов (2.3) и (2.4), при условиях:

$$(3.2) \quad \sum_{\ell=1}^M p_{k\ell} = 1, \quad \sum_{h=1}^L q_{jh} = 1, \quad k = \overline{1, d}, \quad j = \overline{1, m},$$

$$(3.3) \quad \mathbb{E}[\mathbf{v}] = \mathbb{E}[\Phi(\mathbf{x}_j, \mathbf{a}) + \boldsymbol{\xi}] = \mathbf{y},$$

где $\mathbf{y} = (y_1, \dots, y_m)$ — вектор измерений выхода объекта (реальные данные выхода).

Условие (3.3) определяет баланс среднего выхода модели с реальными данными выхода:

$$\begin{aligned}
 \mathbb{E}[\mathbf{v}_j] &= \mathbb{E}[\Phi(\mathbf{x}_j, \mathbf{a}) + \xi_j] = \mathbb{E}[\Phi(\mathbf{x}_j, \mathbf{a})] + \mathbb{E}[\xi_j] = \\
 &= \sum_{\substack{\ell_k=1 \\ k=1, \bar{d}}}^M \Phi(\mathbf{x}_j, a_{1\ell_1}, \dots, a_{d\ell_d}) p_{1\ell_1} \cdots p_{d\ell_d} + \sum_{h=1}^L \xi_{jh} q_{jh} = \\
 (3.4) \quad &= \bar{\Phi}(\mathbf{x}_j) + \sum_{h=1}^L \xi_{jh} q_{jh} = y_j.
 \end{aligned}$$

Сумма в выражении для $\bar{\Phi}$ содержит M^d членов, суммирование осуществляется для всех комбинаций значений случайных величин $a_{k\ell}$. Решение задачи (3.1)–(3.3), подробно рассмотренное в Приложении, дает энтропийно-оптимальные распределения параметров и шумов измерений, что и является конечной целью обучения модели с использованием реальных данных.

4. Рандомизированное прогнозирование

В результате обучения модель оказывается снабжена энтропийно-оптимальными оценками распределений параметров и измерительных шумов, формируя таким образом *рандомизированную предсказательную модель* (РПМ). Такая модель определяет специальную методику прогнозирования — *рандомизированное прогнозирование*, элементы которого применялись для некоторых прикладных задач [24–26].

Рандомизированное прогнозирование базируется на генерации энтропийно-оптимальных распределений параметров (П.4) и измерительных шумов (П.5) с последующим построением ансамбля выхода модели для новых, не известных при обучении, входов модели.

Рассмотрим набор входов РПМ, для которых требуется построить прогноз, который может быть представлен в виде блочного вектора или матрицы, столбцами которой являются указанные входы:

$$X = \{\mathbf{x}_1, \dots, \mathbf{x}_s\} = \begin{bmatrix} x_{11} & \dots & x_{1s} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{ns} \end{bmatrix}.$$

Пусть имеется выборка параметров из распределения P объема S . Тогда ансамбль выхода модели для одного входа \mathbf{x} формируется согласно (2.1) и имеет вид

$$\hat{\mathcal{Y}} = \{\hat{y}_i = \Phi(\mathbf{x}, \mathbf{a}_i)\}, \quad i = \overline{1, S},$$

где \mathbf{a}_i — реализация параметров с распределением P . Ансамбль содержит S траекторий.

Теперь для каждого входа \mathbf{x}_j , $j = \overline{1, s}$ и каждой реализации параметров \mathbf{a}_i , $i = \overline{1, S}$ рассмотрим выборку шумов из распределения q объема U и сформируем итоговый ансамбль выхода модели согласно (2.2):

$$\mathcal{V} = \{\mathbf{v}_j = \Phi(\mathbf{x}_j, \mathbf{a}_i) + \boldsymbol{\xi}_j\}, \quad i = \overline{1, S}, \quad j = \overline{1, s},$$

где $\boldsymbol{\xi}_j = (\xi_{j1}, \dots, \xi_{jU})$ — вектор реализаций шумов для j -го входа, \mathbf{v}_j — вектор измеренного выхода модели для j -го входа.

Таким образом, при прогнозировании для каждого входа и каждой реализации параметров модели, генерируется шум в количестве U реализаций. В итоге ансамбль \mathcal{V} состоит из $W = SU$ траекторий, которые все вместе могут быть представлены блочным вектором или матрицей со строками, соответствующими прогнозируемому выходу модели для каждого входа

$$\mathcal{V} = [\mathbf{v}_1, \dots, \mathbf{v}_W] = \begin{bmatrix} v_{11} & \dots & v_{1s} \\ \vdots & \ddots & \vdots \\ v_{W1} & \dots & v_{Ws} \end{bmatrix}.$$

Для построения итоговой прогнозной траектории моделируемого процесса по ансамблю \mathcal{V} могут быть вычислены средняя и медианные траектории, область стандартного отклонения, а также другие выборочные вероятностные характеристики.

Как видно из выражений ансамблей, для их формирования необходимо иметь распределение шума для каждого входа. Распределения, полученные при обучении, не могут быть напрямую использованы для произвольного количества прогнозных входов, так как получены из известных на этом этапе данных, а количество и характеристики данных при прогнозировании заранее не известны. Выходов из этой ситуации может быть несколько.

Первый состоит в применении в качестве прогнозного распределения шума q распределения, определяемого выражением (П.5) для среднего значения параметра λ (множителей Лагранжа).

Второй подход состоит в использовании в качестве прогнозного распределение шума для одного из входов, используемых при обучении, например последнего, если постановка задачи допускает порядок входов в наборе. Идея этого способа состоит в следующем: если исходить из того, что измерения в каждой точке последовательно расположенных данных производятся одним “устройством”, то логично ожидать некоторой стабилизации характеристик этого измерительного устройства, которое достигается к последнему измерению из последовательности.

Третий подход основан на том предположении, что в результате энтропийного оценивания одновременно и параметров, и шумов можно рассматривать чистый выход модели без шума. Таким образом, можно говорить, что энтропийное оценивание осуществляет фильтрацию. В этом случае применение модели должно осуществляться в чистом виде, без шума.

Важной проблемой в применении энтропийно-рандомизированного подхода к прогнозированию является генерация оптимальных распределений параметров и шумов измерений, полученных при оценивании (обучении) модели.

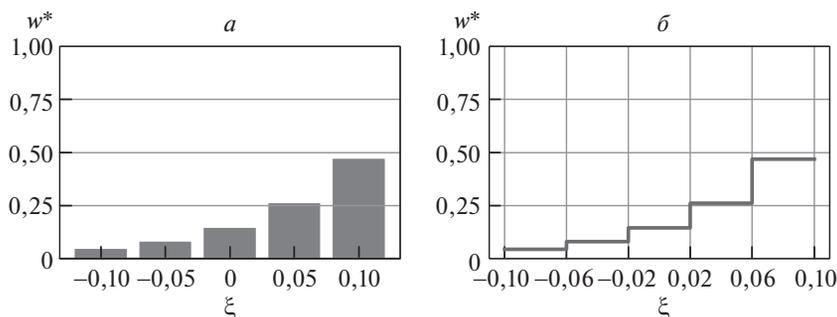


Рис. 1. Дискретное и кусочно-постоянное непрерывное распределение.

Для решения этой проблемы можно предложить два основных подхода, использующих генератор равномерно распределенных случайных чисел.

Первый подход состоит в генерации дискретного распределения. Для этого применяется стандартный подход, состоящий в случайном выборе значения случайной величины и затем соответствующей ему вероятности. В этом случае, очевидно, реализации, сделанные таким образом, будут представлять собой набор значений соответствующей случайной величины, многие из которых будут повторяться.

Второй подход основан на идее представления дискретного распределения в качестве кусочно-постоянной аппроксимации некоторого непрерывного распределения на соответствующем интервале. Для этого интервал значений соответствующей случайной величины разбивается на $L + 1$ подынтервалов (где L — количество значений дискретной случайной величины), левые и правые границы конечных подынтервалов соответствуют левым и правым границам интервала распределения. Внутри каждого подынтервала генерация происходит равномерно. В результате реализации такого подхода можно получить существенно больше различных значений соответствующих случайных величин. Пример построения непрерывных распределений для случайной величины $w^*(\xi)$, где $\xi \in [-0,1, 0,1]$ и $L = 5$, представлен на рис. 1.

5. Прогнозирование роста числа инфицированных COVID-19 в Германии

Предлагаемый в работе подход применяется для моделирования динамики развития эпидемии COVID-19 в Германии на основе данных Университета Джонса Хопкинса [27] начиная с сорокового дня эпидемии (8 марта 2020 г.), когда впервые общее количество инфицированных превысило 1000 человек.

Данные о развитии эпидемии (см. рис. 2) свидетельствуют о том, что сначала инфекция активно распространяется в популяции и наблюдается ее экспоненциальный рост. Далее наблюдается снижение числа зараженных, вероятно, вследствие ограничительных мер или увеличения количества иммунных членов популяции. При этом, как и в большинстве живых систем при рассмотрении их на относительно коротком промежутке времени, можно предположить, что объем популяции не меняется (например, можно пренебречь миграцией, воспроизводством и смертностью), а значит, существует ограниче-

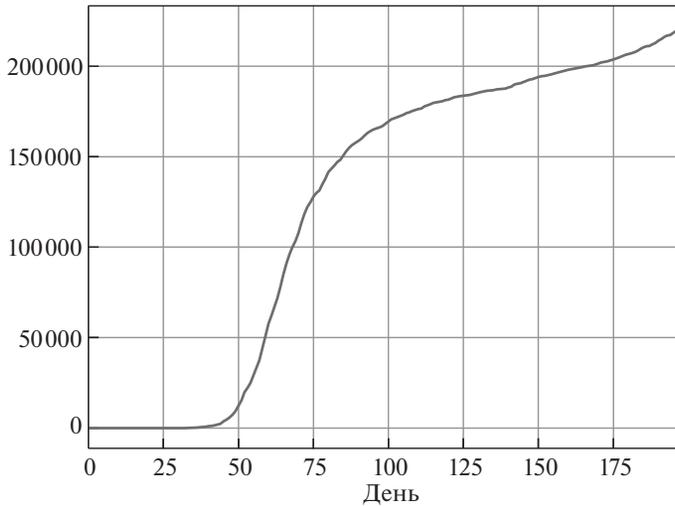


Рис. 2. Общее количество инфицированных в Германии по дням начиная с 8.03.2020.

ние на количество инфицированных. Динамика инфицированных членов популяции N в такой системе может быть описана следующим уравнением [28]:

$$(5.1) \quad \frac{dN}{dt} = \lambda N \left(1 - \frac{N}{K} \right),$$

где λ — скорость роста инфицированных, N — количество инфицированных, K — объем популяции. Решением этого уравнения является кривая Ферхюльста [29, 30]

$$(5.2) \quad N(t) = \frac{K}{1 + B e^{-\lambda t}}, \quad B = \frac{K - N_0}{N_0},$$

где N_0 — количество инфицированных в популяции в начальный момент времени [30].

Модель вида (5.2) активно использовалась в начале 2020 г. для предсказания общего количества заболевших [31–37] и показала свою эффективность на начальном этапе развития эпидемии. В этой связи представляется обоснованным использовать аналогичную модель для применения энтропийно-рандомизированного подхода к прогнозированию общего количества инфицированных. В качестве такой модели будем использовать трехпараметрическую логистическую модель роста (Logistic Growth Model, LGM), которая определяет преобразование скалярного входа x в выход \hat{y} с использованием логистической нелинейной функции

$$(5.3) \quad \hat{y} = \Phi(x, \mathbf{a}) = \frac{a_3}{1 + a_1 e^{-a_2 x}},$$

где $\mathbf{a} = (a_1, a_2, a_3)$ — вектор параметров модели. Данная модель является обобщением модели (5.2) и рассматривается здесь как абстрактная модель с параметрами без использования дополнительных связей между ними.

В контексте рассматриваемой задачи входом является порядковый номер (или индекс) дня, а выходом — накопленное (общее) количество инфицированных. Вход и выход являются целыми, однако в вычислениях целые числа преобразуются в числа с плавающей точкой.

Рандомизированная модель, выход которой искажен аддитивным шумом, а параметры и шумы реализуются дискретными случайными величинами со значениями из соответствующих интервалов, имеет вид

$$v = \hat{y} + \xi = \Phi(x, \mathbf{a}) + \xi,$$

$$a_{k\ell} \in A_k, \quad p_{k\ell} \in [0, 1], \quad k = \overline{1, d}, \quad \ell = \overline{1, M},$$

$$\xi_{jh} \in \Xi_j, \quad q_{jh} \in [0, 1], \quad j = \overline{1, m}, \quad h = \overline{1, L},$$

где $a_{k\ell}, \xi_{jh}$ — значения случайных величин, реализующих параметры и шумы, а $p_{k\ell}, q_{jh}$ — вероятности их реализации, m — количество точек данных, $d = 3$.

Для обучения (оценивания) предсказательной модели использовались данные за несколько дней (интервал обучения \mathcal{T}_{est}) начиная с 8 марта 2020 г. (40-й день эпидемии в Германии), когда впервые было зафиксировано общее количество инфицированных свыше 1000 человек.

Прогнозирование производится на следующие за интервалом оценивания дни вплоть до 120-го дня эпидемии (интервал прогнозирования \mathcal{T}_{pred}).

Полученные рандомизированные прогнозы сравнивались с подгонкой кривой по модели (5.3) с помощью нелинейного метода наименьших квадратов, реализованного функцией `curve_fit` библиотеки `scipy` на платформе Python 3.7.

После оценивания модели проводилась ее реализация на интервале оценивания (тестирование) и на интервале прогнозирования с вычислением следующих метрик качества для истинных (реальных, true) значений y и предсказанных (модельных, predicted) значений \hat{y} :

- 1) коэффициент детерминации R^2 , определяемый формулой

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

где \bar{y} — среднее значение по реальным данным позволяет оценить качество приближения кривой (goodness of fit (GoF)) и предсказательную способность модели через долю объясненной дисперсии. Максимум этого индикатора равен 1, чем его значение ближе к единице, тем выше качество модели;

- 2) средне-квадратичная ошибка (Mean Squared Error (MSE)), определяемая формулой

$$\text{MSE}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n-1} (y_i - \hat{y}_i)^2,$$

показывает ожидаемую (среднюю) квадратичную ошибку;

3) Norm Error (NE), определяемую формулой

$$\text{NE}(y, \hat{y}) = \frac{\|y - \hat{y}\|}{\|y\| + \|\hat{y}\|} = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n y_i^2 + \sum_{i=1}^n \hat{y}_i^2};$$

4) Rooted Norm Error (RNE), определяемую формулой

$$\text{RNE}(y, \hat{y}) = \frac{\sqrt{\|y - \hat{y}\|}}{\sqrt{\|y\|} + \sqrt{\|\hat{y}\|}} = \frac{\sqrt{\sum_{i=1}^n (y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^n y_i^2} + \sqrt{\sum_{i=1}^n \hat{y}_i^2}}.$$

Согласно теории метода энтропийного оценивания определяемые в результате оценивания распределения параметров и шумов являются интервальными. Таким образом, для применения метода необходимо задать эти интервалы. В данной работе интервалы для параметров задавались на основе оптимальных значений, полученных при оценивании методом наименьших квадратов. Границы интервалов устанавливались в пределах 20% от этих значений.

В экспериментах использовались данные, масштабированные на отрезок $[0, 1]$ на интервале оценивания.

Оценивание, тестирование и прогнозирование производилось для нескольких конфигураций:

- без шума;
- с шумом в пределах 10%;
- с шумом в пределах 30%.

При прогнозировании использовалось распределение шума, полученное для последней точки на интервале оценивания. Тестирование модели проводилось в конфигурации, соответствующей прогнозированию.

На рисунках изображены следующие результаты моделирования (траектории):

- метод наименьших квадратов (пунктирная линия с меткой `ols`);
- реальные данные (пунктирная линия с меткой `real`);
- рандомизированное прогнозирование со средними по распределению значениями параметров модели (линия с меткой `mean_params`);
- рандомизированное прогнозирование со средним по ансамблю (линия с меткой `mean`);
- рандомизированное прогнозирование с медианой по ансамблю (линия с меткой `med`).

Светло-серым цветом отмечены траектории, составляющие полученный ансамбль, темно-серым цветом — область стандартного отклонения по ансамблю. Все эксперименты производились для выборки из распределения параметров модели объемом 1000 и выборки из распределением шумов объемом 100 для каждого значения параметра. Генерация распределений шума

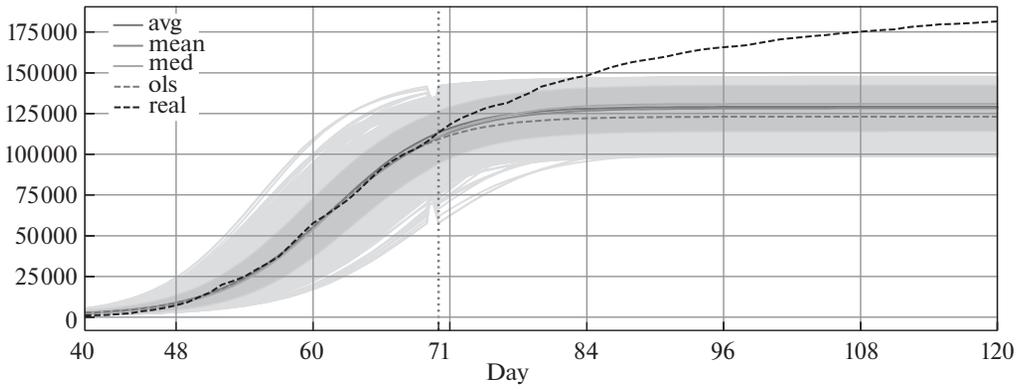


Рис. 3. Прогнозирование без шума (NN) для $\mathcal{T}_{est} = [40, 70]$ и $\mathcal{T}_{pred} = [71, 120]$.

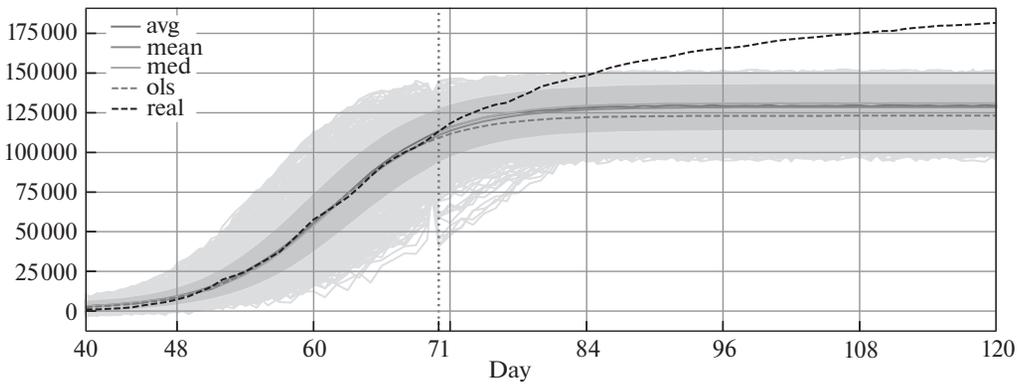


Рис. 4. Прогнозирование с шумом в пределах 10% (N1) для $\mathcal{T}_{est} = [40, 70]$ и $\mathcal{T}_{pred} = [71, 120]$.

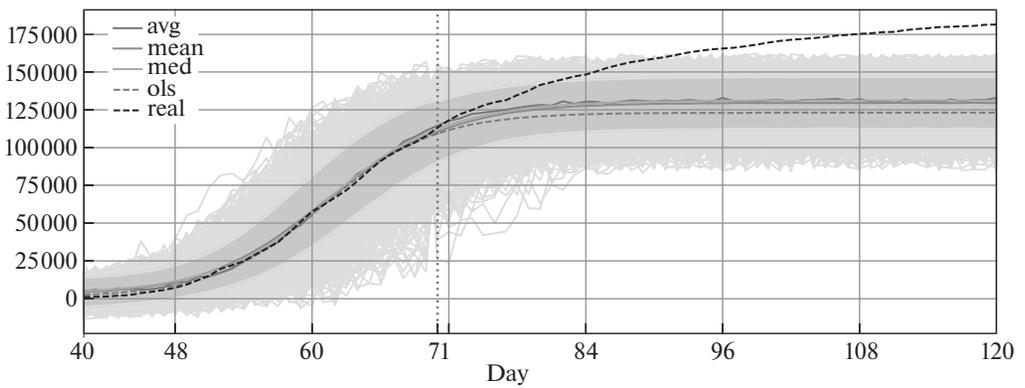


Рис. 5. Прогнозирование с шумом в пределах 30% (N3) для $\mathcal{T}_{est} = [40, 70]$ и $\mathcal{T}_{pred} = [71, 120]$.

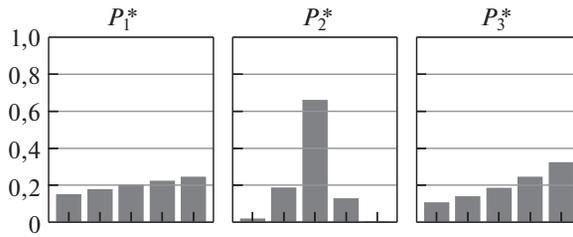


Рис. 6. Энтропийно-оптимальное распределение параметров P^* .

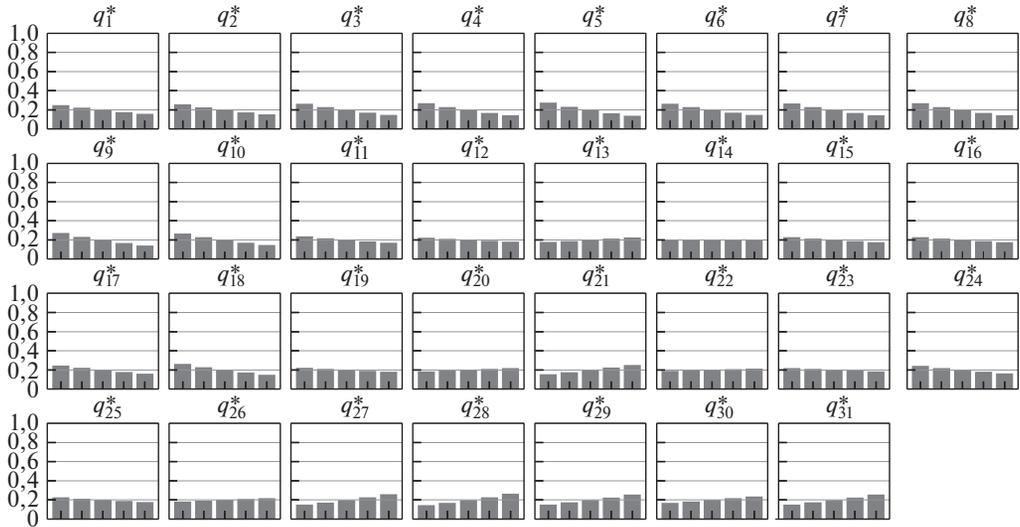


Рис. 7. Энтропийно-оптимальные распределения шумов Q^* .

проводилась для каждой точки соответствующего интервала (тестирования и прогнозирования). Таким образом, полученный ансамбль состоял из 10^5 траекторий. Вертикальная красная пунктирная линия нанесена в точке начала интервала прогнозирования. Эксперименты проводились на платформе Python 3.7 в среде Windows 10.

На рис. 3–5 приведены результаты реализации рандомизированной предсказательной модели на интервалах $\mathcal{T}_{est} = [40, 70]$ и $\mathcal{T}_{pred} = [71, 120]$ для трех вариантов прогнозирования: без шума (NN), с шумом 10% (N1) и с шумом 30% (N3).

На рис. 6–7 изображены энтропийно-оптимальные распределения параметров и шумов, полученные в результате обучения модели на интервале $\mathcal{T} = [40, 70]$ с шумом 30%.

На рис. 8–9 приведены результаты реализации рандомизированной предсказательной модели на интервалах $\mathcal{T}_{est} = [40, 76]$, $\mathcal{T}_{est} = [40, 79]$ и соответствующих интервалах прогнозирования.

Важно отметить, что на 77-й день произошел небольшой рост количества заболевших, что видно на графике. Однако при обучении модели этих данных

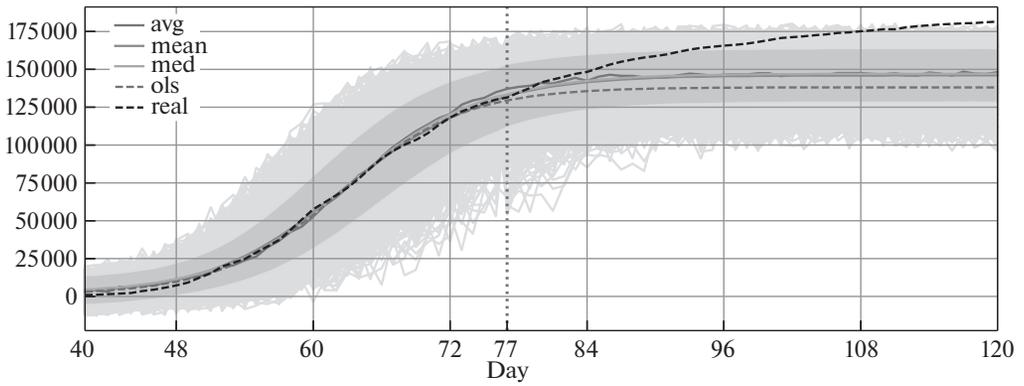


Рис. 8. Прогнозирование с шумом в пределах 30% (N3) для $\mathcal{T}_{est} = [40, 76]$ и $\mathcal{T}_{pred} = [77, 120]$.

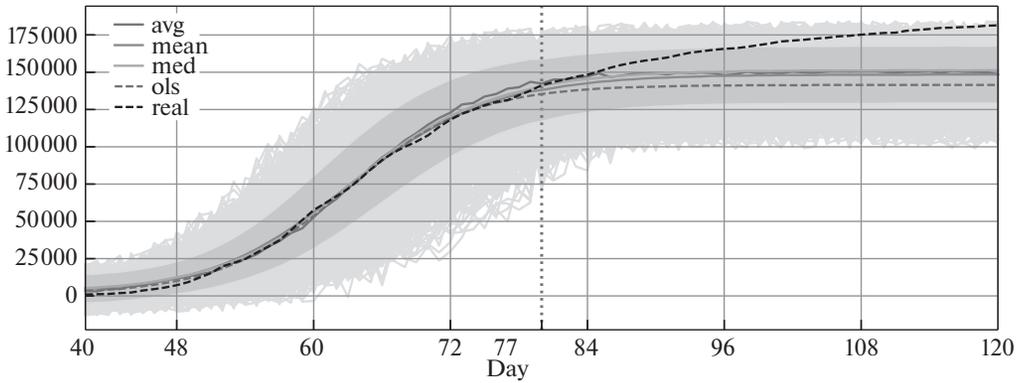


Рис. 9. Прогнозирование с шумом в пределах 30% (N3) для $\mathcal{T}_{est} = [40, 79]$ и $\mathcal{T}_{pred} = [80, 120]$.

еще не было. На 79-й день по-прежнему происходил рост количества заболевших, который начался ранее, но в этом случае уже была возможность учесть эти данные при обучении. На графиках видно, что во всех случаях рандомизированная модель предоставляет более реалистичный прогноз по сравнению с классической моделью.

Оценки параметров, полученные методом наименьших квадратов, и интервалы параметров рандомизированной модели указаны в табл. 1–2. В конфигурациях с шумом интервалы шумов задаются как $\Xi_j = [-0,1, 0,1]$ для N1 и $\Xi_j = [-0,3, 0,3]$ для N3 соответственно.

Значения показателей качества моделирования при тестировании на трех разных интервалах для варианта модели с шумом 30% указаны в табл. 3.

Анализируя полученные результаты, можно отметить, что стандартная методика прогнозирования, связанная с использованием подгонки кривой под данные методом наименьших квадратов, хоть и обладает определенной эф-

Таблица 1. Оценки параметров, полученные методом наименьших квадратов

\mathcal{T}_{test}	a_1	a_2	a_3
[40, 70]	297749	0,2065	1,1448
[40, 76]	1086539	0,1856	1,0624
[40, 79]	83517	0,1803	1,0278

Таблица 2. Конфигурации параметров рандомизированной модели

\mathcal{T}_{pred}	A_1	A_2	A_3
[40, 70]	[238199, 357299]	[0,1652, 0,2478]	[0,9158, 1,3738]
[40, 76]	[86923, 130384]	[0,1485, 0,2227]	[0,8499, 1,2749]
[40, 79]	[66813, 100220]	[0,1443, 0,2164]	[0,8223, 1,2334]

Таблица 3. Метрики качества на интервале оценивания

	R^2	MSE	NE	RNE
$\mathcal{T}_{test} = [40, 70]$				
ols	0,9984	0,0002	0,0004	0,0135
mean_params	0,9899	0,0011	0,0022	0,0335
mean	0,9997	0,0000	0,0001	0,0058
med	0,9980	0,0002	0,0004	0,0149
$\mathcal{T}_{test} = [40, 76]$				
ols	0,9982	0,0002	0,0004	0,0139
mean_params	0,9903	0,0012	0,0020	0,0314
mean	0,9994	0,0001	0,0001	0,0078
med	0,9985	0,0002	0,0003	0,0127
$\mathcal{T}_{test} = [40, 79]$				
ols	0,9982	0,0002	0,0004	0,0133
mean_params	0,9903	0,0012	0,0019	0,0306
mean	0,9997	0,0000	0,0001	0,0054
med	0,9986	0,0002	0,0003	0,0119

фективностью, не всегда способна качественно решить задачу построения корректного прогноза.

Из-за специфики рассматриваемой здесь эпидемии COVID-19 во всем мире наблюдается существенное искажение данных, связанных с ней. В этой связи представляется актуальной задача прогнозирования количества инфицированных с некоторым превышением. Из полученных результатов видно, что прогнозирование с помощью логистической модели, оцененной с использованием тех данных, которые были на момент прогноза, существенно недооценивает реальные данные. В то же время прогнозы, получаемые с использованием предлагаемого в работе подхода, показывают превышение прогнозных значений по сравнению с МНК. Необходимо также отметить, что использование шума в модели, который оценивается поточечно и потом используется при прогнозировании, позволяет построить более реалистичный прогноз.

6. Заключение

В работе развит метод рандомизированного машинного обучения и прогнозирования, основанный на использовании дискретных случайных величин, что приводит к задачам, более адаптированным к численному решению с использованием современной вычислительной техники. Демонстрация предложенного метода проведена на задаче прогнозирования общего количества инфицированных COVID-19 в Германии. Полученные результаты свидетельствуют о работоспособности и эффективности метода и его численной реализации, что определяется меньшей ошибкой при прогнозировании по сравнению со стандартной методикой, основанной на методе наименьших квадратов. Необходимо также отметить, что построенная рандомизированная модель показала хороший результат на интервале обучения, однако на интервале прогноза погрешность по сравнению с реальными данными составила существенную величину. Это связано, по всей видимости, с тем, что логистическая модель эффективна для прогноза не на всех этапах развития эпидемии, в частности, для обеспечения приемлемого уровня качества прогноза необходимо наличие признаков замедления эпидемии на интервале обучения, а также наличие реального затухания эпидемии на интервале прогноза. При проведении экспериментов для обучения использовались данные на этапе начала и активного развития эпидемии, что объясняет большую ошибку при прогнозировании.

ПРИЛОЖЕНИЕ

Рассмотрим решение задачи (3.1)–(3.3), которое осуществляется методом множителей Лагранжа. Функция Лагранжа будет иметь вид

$$L(P, Q, \alpha, \beta, \lambda) = -H(P, Q) + \sum_{k=1}^d \alpha_k \left(\sum_{\ell=1}^M p_{k\ell} - 1 \right) + \sum_{j=1}^m \beta_j \left(\sum_{h=1}^L q_{jh} - 1 \right) + \sum_{j=1}^m \lambda_j \left(\bar{\Phi}(\mathbf{x}_j) + \sum_{h=1}^L \xi_{jh} q_{jh} - y_j \right).$$

Для поиска экстремума функции Лагранжа вычислим производные по прямым переменным P и Q :

$$\begin{aligned} \frac{\partial L}{\partial P} &= \frac{\partial L}{\partial p_{k\ell}} = \ln p_{k\ell} + 1 + \alpha_k + \sum_{j=1}^m \lambda_j \frac{\partial \bar{\Phi}_j}{\partial p_{k\ell}}, \\ \frac{\partial L}{\partial Q} &= \frac{\partial L}{\partial q_{jh}} = \ln q_{jh} + 1 + \beta_j + \sum_{j=1}^m \lambda_j \xi_{jh}, \\ k &= \overline{1, d}, \quad j = \overline{1, m}, \quad \ell = \overline{1, M}, \quad h = \overline{1, L}, \end{aligned}$$

где $\bar{\Phi}_j = \bar{\Phi}(\mathbf{x}_j)$ и производная среднего значения модели по $p_{k\ell}$ определяется выражением

$$(II.1) \quad \frac{\partial \bar{\Phi}_j}{\partial p_{k\ell}} = \sum_{\substack{\ell_s=1 \\ s=1, \bar{d}}}^M \Phi(x_j, a_{1\ell_1}, \dots, a_{d\ell_d}) \prod_{r \neq k} p_{r\ell_r}.$$

Сумма в выражении для производной $\frac{\partial \bar{\Phi}}{\partial p_{k\ell}}$ содержит $M(d-1)$ членов.

Приравнивая к нулю производные функции Лагранжа по прямым переменным, получим выражения оптимальных распределений вероятностей параметров и шумов от множителей Лагранжа:

$$p_{k\ell}^*(\alpha, \lambda) = \exp \left(-1 - \alpha_k - \sum_{j=1}^m \lambda_j \frac{\partial \bar{\Phi}_j}{\partial p_{k\ell}} \right),$$

$$q_{jh}^*(\beta, \lambda) = \exp(-1 - \beta_j - \lambda_j \xi_{jh}).$$

Преобразуем эти выражения следующим образом:

$$(II.2) \quad p_{k\ell}^*(\alpha, \lambda) = \exp(-1 + \alpha_k) \exp \left(- \sum_{j=1}^m \lambda_j \frac{\partial \bar{\Phi}_j}{\partial p_{k\ell}} \right),$$

$$(II.3) \quad q_{jh}^*(\beta, \lambda) = \exp(-1 + \beta_j) \exp(-\lambda_j \xi_{jh}),$$

и, подставляя их в условия нормировки (3.2), получим выражения:

$$\exp(1 + \alpha_k) = \exp \left(- \sum_{\ell=1}^M \lambda_j \frac{\partial \bar{\Phi}_j}{\partial p_{k\ell}} \right),$$

$$\exp(1 + \beta_j) = \exp(-\lambda_j \xi_{jh}).$$

Подставим эти выражения обратно в (II.2)–(II.3), исключив таким образом множители α и β , получим финальные выражения энтропийно-оптимальных распределений параметров и шумов, зависящих от множителей λ :

$$(II.4) \quad p_{k\ell}^*(\lambda) = \frac{\exp \left(- \sum_{j=1}^m \lambda_j \frac{\partial \bar{\Phi}_j}{\partial p_{k\ell}} \right)}{\sum_{\ell=1}^M \exp \left(- \sum_{j=1}^m \lambda_j \frac{\partial \bar{\Phi}_j}{\partial p_{k\ell}} \right)}, \quad k = \overline{1, d}, \ell = \overline{1, M},$$

$$(II.5) \quad q_{jh}^*(\lambda) = \frac{\exp(-\lambda_j \xi_{jh})}{\sum_{h=1}^L \exp(-\lambda_j \xi_{jh})}, \quad j = \overline{1, m}, h = \overline{1, L}.$$

Множители λ определяются решением системы уравнений, получающихся подстановкой выражений (П.4)–(П.5) в балансовые соотношения (3.3):

$$(П.6) \quad \sum_{\substack{\ell_k=1 \\ k=1, \bar{d}}}^M \Phi(\mathbf{x}_j, a_{1\ell_1}, \dots, a_{d\ell_d}) \prod_{\substack{\ell_s=1 \\ s=1, \bar{d}}}^M p_{s\ell_s}^*(\lambda) + \sum_{h=1}^L \xi_{jh} q_{jh}^*(\lambda) = y_j, \quad j = \overline{1, m}.$$

Таким образом, решив систему (П.6), получим энтропийно-оптимальные распределения параметров и шумов измерений, что и является конечной целью обучения модели с использованием реальных данных.

Необходимо отметить, что для решения этой системы на практике необходимо привлекать какой-нибудь численный метод, так как ее решение аналитически сопряжено с существенными трудностями.

Вычисление левой части системы потребует вычисления среднего значения случайной функции $\bar{\Phi}$, а также ее производной по распределению P , определяемых выражениями (3.4) и (П.1). Суммирования в этих выражениях должны производиться по всем комбинациям индексов, таким образом количество операций суммирования растет как степень от количества параметров d .

СПИСОК ЛИТЕРАТУРЫ

1. *Bishop C.M.* Pattern Recognition and Machine Learning. Springer, Series: Information Theory and Statistics, 2006.
2. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning. Springer, 2001.
3. *Айвазян С.А., Мхитарян В.С.* Прикладная статистика и основы эконометрики. М.: Юнити, 1998.
4. *Мерков А.Б.* Распознавание образов. Введение в методы статистического обучения. М.: URSS, 2010.
5. *Аркадьев А.Г., Браверман Э.М.* Обучение машины распознаванию образов. М.: Наука, 1964.
6. *Цыпкин Я.З.* Основы теории обучающихся систем. М.: Наука, 1970.
7. *Вапник В.Н., Червоненкис А.Я.* Восстановление зависимостей по эмпирическим данным, М.: Наука, 1979.
8. *Вапник В.Н. Червоненкис А.Я.* Теория распознавания образов. М.: Наука, 1974.
9. *Cristianini N., Shawe-Taylor J.* An introduction to support vector machines, 2000.
10. *Breiman J.H., Friedman R., Olshen A., Stone C.J.* Classification and regression trees. 1984.
11. *Rosenblatt F.* The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory, 1957.
12. *Rumelhart D.E., Williams R.J., Hinton G.* Learning representations by back-propagating errors // Nature. 1986. V. 323. No. 6088. P. 533–538.
13. *Попков Ю.С., Попков А.Ю., Дубнов Ю.А.* Рандомизированное машинное обучение при ограниченных наборах данных: от эмпирической вероятности к энтропийной рандомизации. М.: ЛЕНАНД, 2019.

14. *Больцман Л.* О связи между вторым началом механической теории теплоты и теорией вероятностей в теоремах о тепловом равновесии / Больцман Л.Э. Избранные труды. под ред. Шлак Л.С. Классики науки. М.: Наука, 1984.
15. *Jaynes E.T.* Information theory and statistical mechanics // *Physical review*. 1957. V. 106. No. 4. P. 620–630.
16. *Jaynes E.T.* Probability theory: the logic of science. Cambridge university press, 2003.
17. *Shannon C.E.* Communication theory of secrecy systems // *Bell Labs Technical J.* 1949. V. 28. No. 4. P. 656–715.
18. *Diebold F.* Elements of Forecasting. Thomson, South-Western, Ohio, US, 4th edition, 2007.
19. *Gneiting T., Katzfuss M.* Probabilistic forecasting // *Annual Review of Statistics and Its Application*, 2014. No. 1. P. 125–151.
20. *Hong T., Fan S.* Probabilistic electric load forecasting: A tutorial review // *Int. J. Forecasting*. 2016. V. 32. No. 3. P. 914–938.
21. *Айвазян С.А., Мхитарян В.С.* Прикладная статистика. Классификация и снижение размерности. М.: Финансы и статистика, 1989.
22. *Golan A., Judge G., Miller D.* Maximum Entropy Econometrics: Robust Estimation with Limited Data. N.Y.: John Wiley & Sons, 1996.
23. *Golan A. et al.* Information and Entropy Econometrics — A Review and Synthesis // *Foundations and Trends in Economet.* 2008. V. 2. No. 1–2. P. 1–145.
24. *Popkov Y.S., Volkovich Z., Dubnov Y.A., Avros R., Ravve E.* Entropy 2-soft classification of objects // *Entropy*. 2017. V. 19. No. 4. P. 178.
25. *Попков Ю.С., Попков А.Ю., Дубнов Я.А.* Элементы рандомизированного прогнозирования и его применение для предсказания суточной электрической нагрузки энергетической системы // *АиТ*. 2020. № 7. С. 148–172.
Popkov Y.S., Popkov A.Y., Dubnov Y.A. Elements of Randomized Forecasting and Its Application to Daily Electrical Load Prediction in a Regional Power System // *Autom. Remote Control*. 2020. V. 81. P. 1286–1306.
26. *Popkov Y.S., Popkov A.Y., Dubnov Y.A., Solomatine D.* Entropy-randomized forecasting of stochastic dynamic regression models // *Mathematics*. 2020. V. 8. No. 7. P. 1119.
27. *Dong E., Du H., Gardner L.* An interactive web-based dashboard to track covid-19 in real time // *The Lancet Infectious Diseases*. 2020. V. 20. No. 5. P. 533–534.
28. *van den Driessche P.* Mathematical Epidemiology. In F. Brauer, P. van den Driessche & J. Wu (Eds.), *Lecture Notes in Mathematics*, 2008, <https://doi.org/10.1007/978-3-540-78911-6>.
29. *Verhulst P.-F.* Notice sur la loi que la population suit dans son accroissement // *Corresp. Math. Phys.*, 1893. No. 10. P. 113–126.
30. *Singer H.M.* The COVID-19 pandemic: growth patterns, power law scaling, and saturation // *Physical Biology*. 2020. Vol. 17. No. 5. P. 055001. <https://doi.org/10.1088/1478-3975/ab9bf5>
31. *Kumar J., Hembram K.P.S.S.* Epidemiological study of novel coronavirus (COVID-19) // *ArXiv*. 2020. <http://arxiv.org/abs/2003.11376>
32. *Yang W., Zhang D., Peng L., Zhuge C., Hong L.* Rational evaluation of various epidemic models based on the COVID-19 data of China // *ArXiv*. 2020. <http://arxiv.org/abs/2003.05666>

33. *Tatrai D., Varallyay Z.* COVID-19 epidemic outcome predictions based on logistic fitting and estimation of its reliability // ArXiv. 2020. <http://arxiv.org/abs/2003.14160>
34. *Morais A.F.* Logistic approximations used to describe new outbreaks in the 2020 COVID-19 pandemic. ArXiv. 2020. <http://arxiv.org/abs/2003.11149>
35. *Shen C.Y.* Logistic growth modelling of COVID-19 proliferation in China and its international implications // Int. J. Infectious Diseases. 2020. Vol. 96. P. 582–589. <https://doi.org/10.1016/j.ijid.2020.04.085>
36. *Wang P., Zheng X., Li J., Zhu B.* Prediction of epidemic trends in COVID-19 with logistic model and machine learning technics // Chaos Solitons & Fractals. 2020. Vol. 139. P. 110058. <https://doi.org/10.1016/j.chaos.2020.110058>
37. *Chen D.-G., Chen X., Chen J. K.* Reconstructing and forecasting the COVID-19 epidemic in the United States using a 5-parameter logistic growth model // Global Health Research and Policy. 2020. Vol. 5. No. 1. P. 25. <https://doi.org/10.1186/s41256-020-00152-5>

Статъя представена к публикации членом редколлегии А.И. Михальским.

Поступила в редакцию 15.10.2020

После доработки 12.01.2020

Принята к публикации 15.01.2021

СОДЕРЖАНИЕ

Обзоры

Ходашинский И.А. Методы повышения эффективности роевых алгоритмов оптимизации	3
--	---

Стохастические системы

Белов И.Р. Анизотропная фильтрация для линейных дискретных нестационарных систем с мультипликативными шумами на конечном интервале времени	46
Обжерин Ю.Е., Сидоров С.М., Никитин М.М., Глеч С.Г. Скрытая марковская модель на основе суперпозиции двух процессов восстановления	80

Робастное, адаптивное и сетевое управление

Александров В.А. Оптимизация размещения полюсов в одномерной системе управления	102
--	-----

Интеллектуальные системы управления, анализ данных

Конев В.В. , Пупков А.В. Доверительное оценивание параметров авторегрессии по зашумленным данным	124
Попков А.Ю. Рандомизированное машинное обучение нелинейных моделей с применением к прогнозированию развития эпидемического процесса	149

C O N T E N T S

Surveys

Hodashinsky I.A. Techniques to Improve the Efficiency of Swarm Optimization Algorithms	3
---	---

Stochastic Systems

Belov I.R. Anisotropy-based Filtering for Linear Discrete Time Varying Systems with Multiplicative Noises on the Finite Horizon	46
Obzherin Yu.E., Sidorov S.M., Nikitin M.M., Glech S.G. Hidden Markov Model Based on Two Renewal Processes Superposition	80

Robust, Adaptive and Network Control

Alexandrov V.A. Pole Placement Optimization for SISO Control System	102
--	-----

Intellectual Control Systems, Data Analysis

Konev V.V. , Pupkov A.V. Confidence Estimation of Autoregression Parameters from Noisy Data	124
Popkov A.Yu. Randomized Machine Learning of Nonlinear Models with Application to the Forecasting of Epidemic Development	149