

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ

*На правах рукописи*

Сергеенко Анна Николаевна

**УПРАВЛЕНИЕ СЕНСОРНОЙ СЕТЬЮ НА  
ОСНОВЕ РАНДОМИЗИРОВАННОГО И  
МУЛЬТИАГЕНТНОГО ПОДХОДОВ**

Научная специальность 1.2.3. Теоретическая информатика, кибернетика

**Диссертация  
на соискание ученой степени  
кандидата физико-математических наук**

Научный руководитель:  
доктор физико-математических наук, профессор  
Олег Николаевич Граничин

Санкт-Петербург  
2023

# Оглавление

Введение	4
<b>Глава 1. Модель управления сенсорной сетью на основе рандомизированного и мультиагентного подходов при ограничениях на коммуникации и наличии существенных неопределенностей</b>	<b>15</b>
1.1 Обозначения и определения . . . . .	15
1.2 Сетевая модель наблюдения за движущимися объектами . . . . .	18
1.3 Рандомизированный и мультиагентный подходы . . . . .	26
1.4 Задача управления сенсорной сетью . . . . .	39
1.4.1 Оптимизация выбора целей и сети сенсоров . . . . .	40
1.4.2 Рандомизация топологии . . . . .	42
1.5 Выводы . . . . .	46
<b>Глава 2. Распределенный рандомизированный алгоритм стохастической аппроксимации, совмещенный с протоколом локального голосования</b>	<b>48</b>
2.1 Оптимизация нестационарного функционала среднего риска	48
2.2 Распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, для оценки движущихся целей . .	51
2.3 Взвешенная версия распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования . . . . .	57

2.4	Распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, для оценки положения статически объектов . . . . .	63
2.5	Выводы . . . . .	68
<b>Глава 3. Отслеживание движения объектов при наличии неопределенностей и ограничений на количество связей между агентами</b>		<b>69</b>
3.1	Численные эксперименты . . . . .	69
3.2	Прототип системы для отслеживания целей распределенной сетью сенсоров . . . . .	75
3.2.1	Архитектура системы . . . . .	75
3.2.2	Графический интерфейс . . . . .	77
3.2.3	Примеры работы системы . . . . .	81
3.3	Практическое применение рандомизированного алгоритма стохастической аппроксимации, совмещенного с протоколом локального голосования . . . . .	88
3.4	Выводы . . . . .	89
<b>Заключение</b>		<b>91</b>
<b>Литература</b>		<b>92</b>

## Введение

В настоящее время задача управления сенсорной сетью в режиме реального времени при существующих ограничениях на коммуникацию имеет множество практических приложений, таких как управление в коммуникационных сетях, например, [2] (Б. Р. Андриевский, А. С. Матвеев, А. Л. Фрадков), мобильная робототехника [36] (О. Н. Граничин), [45] (Ф. Булло (F. Bullo)) и управление в транспортных и логистических сетях [101] (Г. А. Ржевский и П. О. Скобелев), [42] (С.С. Блэкман (S. S. Blackman)), [4] (В. И. Городецкий).

Во многих работах для проблемы отслеживания изменения параметров динамических систем в режиме реального времени активно используются централизованные подходы, при которых каждый сенсор передает данные в общий центр, где происходит обработка всех данных. Стоит учитывать, что чем больше размер системы и количество наблюдаемых характеристик, тем больше должна быть сенсорная сеть, чтобы процесс отслеживания был надежным и быстрым. Соответственно, каналы, по которым передаются данные, становятся все более загруженными по мере роста сенсорной сети и замедляют передачу данных. Кроме этого, перегружается центр обработки данных, куда поступают все наблюдения и где производится расчет искомым характеристик системы. Благодаря развитию информационных технологий все больший интерес сейчас представляют распределенные вычисления, применимые для задач, в которых есть возможность перенести обработку данных ближе к их источнику. Такие вычисления являются более надежными по сравнению с централизованными вычислениями, так как нарушения в работе одной вычислительной единицы не приостанавливают работу всей системы. Смещение фокуса исследований со специализированных централизованных комплексов к системам с децентрализацией можно проследить по работам по распределенным вычислениям Н. А. Линча (N. A. Lynch) [79], теории принятия оптимальных управленческих решений М. Х. ДеГрута (M. H. DeGroot) [49], коллективного поведения Т. Висека



(Т. Vicsek) [88,119], распределенным методам принятия решений Дж. Н. Тситсиклиса (J. N. Tsitsiklis) [116,117] в области теории управления.

Прогресс в области электроники и информационных технологий вызывает особый интерес к мультиагентным технологиям, использующим сети взаимосвязанных автономных единиц (*агентов*), которые могут быть распределены на большие расстояния и играть роль датчиков, процессоров данных, исполнительных механизмов и сенсоров. Таким образом, в настоящее время сенсоры могут быть наделены вычислительными возможностями и иметь доступ к измерениям других сенсоров. На практике, однако, часто возникают ограничения различного рода: каждый сенсор обычно может взаимодействовать лишь с несколькими соседними узлами сети, а связь между сенсорами может быть лимитирована, например, из-за ограниченной пропускной способности каналов связи или задержек в передаче данных. Математически такие ограничения можно смоделировать, как ограниченную скорость передачи данных или меняющийся во времени коммуникационный граф, естественным образом возникающий в ситуации, когда сенсорам приходится делить несколько линий связи или узкую полосу частот. В работах Р. П. Агаева [33], Б. Р. Андриевского [2], Р. В. Берда (R. W. Beard) [97,98], Ф. Булло (F. Bullo) [45], А. Джадбабаи (A. Jadbabaie) [73], В. В. Захарова [107], М. И. Карпова [13], А. Ю. Крылатова [75], Ф. Л. Льюиса (F. L. Lewis) [48], А. С. Морса (A.S. Morse) [73], Р. М. Мюррея (R. M. Murray) [61, 90], Р. Олфати-Сабера (R. Olfati-Saber) [90], Л. А. Петросяна [13], В. Рена (W. Ren) [97,98], П. Ю. Чеботарева [33], Ю. В. Цыгановой [118], и других заложены фундаментальные принципы построения распределенных алгоритмов мультиагентной (многоагентной) координации и управления движением, обсуждаются сферы практического применения разработанных подходов к организации коллективного поведения наблюдателей. Вышеописанные проблемы актуализируют исследование мультиагентного подхода к рассматриваемой проблеме.

Во время измерения характеристики параметров динамических систем в режиме реального времени обычно возникают различные меры

неопределенности. Часто такие проблемы сводятся к оптимизации некоторых функционалов среднего риска, например, к оптимизации среднеквадратичной ошибки слежения. Во многих случаях используются метод эмпирического функционала, метод максимального правдоподобия или байесовское оценивание, однако такие алгоритмы опираются на существенные предположения и статистические свойства помех наблюдения. Обычно предполагаются их центрированность и некоррелированность. Наиболее интересна постановка задачи о выборе в том или ином смысле наилучших оценок. Известно, что метод наименьших квадратов является оптимальным в случае независимых одинаково распределенных гауссовых помех в наблюдениях, но его применение не обосновано, если такие условия не выполняются. В частности, трудной проблемой является оценивание, когда наблюдаемые характеристики параметров могут меняться с произвольными неизвестными, но ограниченными возмущениями, а при измерениях появляются систематические ошибки (ошибки модели), которые зачастую очень сложно исключить. В работах А. С. Матвеева [95], А. Недич (A. Nedic) [84, 86], Р. Олфати-Сабера (R. Olfati-Saber) [89], А. В. Проскурникова [87, 94, 95] представлены результаты исследования алгоритмов распределенной оптимизации и консенсусного управления. Для задачи достижения консенсуса на графах при наличии зашумленных измерений о состояниях соседей в работах М. Дж. Вайнрайта (M. J. Wainwright) [96], Д. Вергадоса (Vergados D.J.) с соавторами [115], О. Н. Граничина с соавторами [109], М. Хуанга (M. Huang) [69], А. Л. Фрадкова и Н. О. Амелиной [38] рассматривалось применение алгоритмов типа стохастической оптимизации. Этот вид алгоритмов является одним из важнейших классов среди подходов к решению задач оптимизации с неопределенностями. Например, при наличии произвольных внешних возмущений возможно найти решение вышеописанной проблемы с помощью алгоритмов рандомизированной стохастической оптимизации, описанных в статьях О.Н. Граничина с соавторами [64, 65, 108]. В том числе, в работе [108] был предложен рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального го-

лосования, способный эффективно решать оптимизационные задачи при наличии неизвестных, но ограниченных помех в измерениях наблюдаемой функции, а также было проведено исследование сходимости предложенного алгоритма. Актуальной задачей является применение рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, для задачи управления сенсорной сетью при отслеживании параметров динамической системы.

В диссертации рассматривается задача управления сенсорной сетью в контексте распределенного *сопровождения цели* сетями подключенных датчиков (радаров, гидролокаторов, камер и т.д.), что является классической задачей обработки сигналов [67] (О. Хлинка (O. Hlinka) и соавторы). Такая проблема распределенного слежения за несколькими целями широко изучались в литературе в связи с многочисленными приложениями в управлении воздушным [78], морским [80] и дорожным движением [71], а также в видеонаблюдении [83], однако, как правило, рассматривались неопределенности, обладающие заданными традиционными статистическими свойствами центрированности и независимости, что часто не выполняется на практике. В частности, в многообразии возможных вариантов поведения целей недостаточно статистики, так что траектории маневрирующих целей могут быть описаны как произвольные неизвестные, но ограниченные возмущения. В качестве решения такой задачи предлагается применить модернизированный распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, для сетевой модели наблюдения за движущимися объектами.

Обозначенные проблемы и тенденции подтверждают актуальность темы диссертационного исследования.

*Цель работы* — разработка алгоритмов управления распределенной сенсорной сетью в режиме реального времени при ограничениях на коммуникации и наличии существенных неопределенностей в описании исследуемой системы.

Для достижения этой цели были поставлены и решены следующие задачи:

- 1) разработать модель управления сенсорной сетью на основе рандомизированного и мультиагентного подходов при ограничениях на коммуникации и наличии существенных неопределенностей в описании исследуемой системы;
- 2) модернизировать распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, и исследовать свойства его оценок для задачи трекинга (отслеживания изменения параметров) при использовании разработанной модели управления сенсорной сетью;
- 3) исследовать условия применимости модернизированного распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, для сетевой модели наблюдения за движущимися объектами при наличии неопределенностей и ограничений на количество связей между сенсорами, а также на количество наблюдаемых объектов.

*Методы исследования.* В диссертации используются методы теорий оценивания, оптимизации, управления, графов, вероятностей и математической статистики; применяются методы стохастической аппроксимации, рандомизированные алгоритмы, имитационное моделирование.

*Основные результаты.* В ходе выполнения работы получены следующие научные результаты:

- 1) разработана модель управления сенсорной сетью на основе рандомизированного и мультиагентного подходов;
- 2) модернизирован распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального

голосования, и исследованы свойства его оценок для задачи трекинга (отслеживания изменения параметров) при использовании разработанной модели управления сенсорной сетью;

- 3) исследованы условия применимости модернизированного распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, для сетевой модели наблюдения. Результаты апробированы в задаче наблюдения за движущимися объектами при наличии неопределенностей и ограничений на количество связей между сенсорами.

*Научная новизна.* Все основные научные результаты диссертации являются новыми.

*Теоретическая ценность и практическая значимость.* Теоретическая ценность результатов заключается в разработке модели слежения объектами за группой наблюдателей, модернизации распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, исследовании свойств его оценок, а также условий применимости для сетевой модели наблюдения. Предложенные методы и подходы могут использоваться при решении ряда практических задач, в том числе для отслеживания летательных объектов в воздушном пространстве и координации их перемещений.

*Апробация работы.* Результаты диссертации докладывались на семинарах кафедр системного программирования математико-механического факультета СПбГУ, на конференции «Неделя Науки СПбПУ 2018» (19-24 ноября 2018 г., Санкт-Петербург, Россия), на международном семинаре «International Workshop Navigation and Motion Control» (NMC 2019) (16-20 октября 2019 г., Ленинградская обл., Россия), на XXI конференции молодых ученых «Навигация и управление движением» (с международным участием) (19-22 Марта 2019 г., г. Санкт-Петербург, Россия), на конференции The 9th International Scientific Conference on Physics and Control (8-11 сентября 2019 г., г. Иннополис, Россия), на XXII конферен-

ции молодых ученых «Навигация и управление движением» (с международным участием) (17-20 Марта 2020 г., г. Санкт-Петербург, Россия), на конференции 27th Mediterranean Conference on Control and Automation (MED 2019) (1-4 июля 2019 г., г. Акко, Израиль), 13-ой мультikonференции по проблемам управления (Математическая теория управления и ее приложения) (6–8 октября 2020 г., г. Санкт-Петербург, Россия), на 18-ой Национальной конференции по искусственному интеллекту (10-16 октября 2020 г., г. Москва, Россия), на конференции 59th IEEE Conference on Decision and Control (CDC 2020) (14-18 декабря 2020 г., г. Джеджу, Республика Корея), на 63-й Всероссийской научной конференции МФТИ (23–29 ноября 2020 г., г. Москва, Россия), на конференции IFAC World Congress (11-17 июля 2020 г., г. Берлин, Германия), на конференции 19th IFAC Symposium on System Identification (SYSID 2021)(13-16 июля 2021 г., г. Падова, Италия), на XXIII конференции молодых ученых «Навигация и управление движением» (с международным участием) (23-29 Марта 2021 г., г. Санкт-Петербург, Россия), на конференции 5th Scientific School Dynamics of Complex Networks and their Applications (DCNA 2021) (13-15 сентября, 2021 г., г. Калининград, Россия), на конференции 60th IEEE Conference on Decision and Control (CDC 2021) (13-17 декабря 2021 г., г. Остин, США), на конференции European Control Conference (ECC 2021) (29 июня-2 июля 2021 г., г. Роттердам, Нидерланды), на XIV конференции молодых ученых «Навигация и управление движением» (с международным участием) (15-18 Марта 2022 г., Санкт-Петербург, Россия), на конференции Americal Control Conference (ACC 2022) (8-10 июня 2022 г., г. Атланта, США), на конференции 6th Scientific School Dynamics of Complex Networks and their Applications (DCNA 2022) (14-16 сентября 2022 г., г. Калининград, Россия).

Результаты диссертации были использованы в работах по грантам РФФ 16-19-00057 «Адаптивное управление с прогнозирующими моделями при переменной структуре пространства состояний с применением к системам сетевого управления движением и автоматизации медицинского оборудования», РФФИ 20-01-00619 «Рандомизированные ал-

горитмы многоагентной оптимизации, распознавания образов и оценивания в условиях существенных неопределенностей», РНФ 21-19-00516 «Мультиагентное адаптивное управление в сетевых динамических системах с применением к группам робототехнических устройств в условиях неопределенностей». Проект «Разработка технологии отслеживания летательных аппаратов в условиях неопределенностей», включающий результаты исследования, отмечен дипломом победителя молодежного научно-инновационного конкурса (УМНИК-2019). Проект «Разработка распределенных алгоритмов отслеживания траекторий множества объектов массивом мобильных сенсоров» в 2020 году отмечен дипломом победителя конкурса для студентов и аспирантов вузов, отраслевых академических институтов, расположенных на территории Санкт-Петербурга.

*Публикация результатов.* Основные результаты исследований отражены в работах [11, 12, 18, 19, 21–32, 34, 46, 47, 54, 55, 59, 60, 72, 102–106, 121]. Соискателем опубликовано 30 научных работ, из которых одна публикация – свидетельство о регистрации программы на ЭВМ, 14 опубликованы в изданиях, входящих в РИНЦ, 14 опубликованы в изданиях, индексируемых в базе данных Scopus и 1 опубликована в издании, входящая в перечень ВАК.

Работы [11, 12, 18, 19, 31, 32, 34, 46, 47, 54, 55, 59, 60, 72, 102–106, 121] написаны в соавторстве. В работе [32] А. Н. Сергеенко принадлежит доказательство теорем и результаты имитационного моделирования, соавтору – общая постановка задачи. В [18] А. Н. Сергеенко принадлежит модификация протокола локального голосования для задачи управления движением группы динамических объектов (роботов, агентов), соавторам – общая постановка задачи. В [11] А. Н. Сергеенко принадлежит имитационное моделирование, соавторам – общая постановка задачи и выбор методов решения. В [19] А. Н. Сергеенко принадлежит описание подхода к рандомизации связей в сенсорной сети для удовлетворения стоимостных ограничений, соавторам – общая постановка задачи. В [12] А. Н. Сергеенко принадлежит общая постановка задачи, соавтору – выбор методов решения, имитационное моделирование. В [31] А. Н. Серге-

енко принадлежат выбор методов решения и результат экспериментов, соавтору – общая постановка задачи. В [34] А. Н. Сергееенко принадлежит имитационное моделирование, соавторам – общая постановка задачи, выбор методов решения, доказательство теоремы. В [46] А. Н. Сергееенко принадлежит доказательство теоремы, соавторам – общая постановка задачи, выбор методов решения и имитационное моделирование. В [47] А. Н. Сергееенко принадлежит доказательство теоремы и имитационное моделирование, соавторам – общая постановка задачи, выбор методов решения. В [54] А. Н. Сергееенко принадлежит выбор метода решения, соавторам – общая постановка задачи, результаты экспериментов. В [55] А. Н. Сергееенко принадлежит описание эмерджентного интеллекта, соавторам – общая постановка задачи, выбор методов решения и имитационное моделирование. В [59] А. Н. Сергееенко принадлежит имитационное моделирование, соавторам – общая постановка задачи, выбор методов решения. В [72] А. Н. Сергееенко принадлежит имитационное моделирование, соавтору – общая постановка задачи, выбор методов решения. В [60] А. Н. Сергееенко принадлежит имитационное моделирование, соавторам – общая постановка задачи, выбор методов решения. В [102] А. Н. Сергееенко принадлежит имитационное моделирование, соавторам – общая постановка задачи, выбор методов решения, доказательство теорем. В [104] А. Н. Сергееенко принадлежит доказательство теорем, соавторам – общая постановка задачи, выбор методов решения и имитационное моделирование В [103] А. Н. Сергееенко принадлежат описание истории возникновения рандомизированных и мультиагентных алгоритмов, выбор методов решения, доказательство теорем и имитационное моделирование, соавтору – общая постановка задачи. В [106] А. Н. Сергееенко принадлежит выбор метода решения, соавторам – общая постановка задачи, результаты экспериментов. В [105] А. Н. Сергееенко принадлежит общая постановка задачи и имитационное моделирование, соавторам – выбор методов решения. В [121] А. Н. Сергееенко принадлежит доказательство теоремы, соавторам – общая постановка задачи, выбор методов решения.

*Структура и объем диссертации.* Диссертация состоит из введения,



трех глав, заключения, списка литературы, 122 источника. Текст занимает 104 страницы, содержит 23 рисунка и 0 таблиц.

*Краткое содержание работы.*

Во **введении** обосновывается актуальность диссертационной работы, формулируется цель, ставятся задачи исследования и кратко излагаются основные результаты.

В **первой главе** приводится описание проблемы оценивания движущихся объектов сетью сенсоров, сопровождаемое обзором литературы по теме исследования. В Разделе 1.1 вводятся обозначения и основные определения. В Разделе 1.2 описываются особенности разработки сетевой модели наблюдения за движущимися объектами (целями). Вводятся предположения о скорости движения целей, а также о сенсорах, точках наблюдения целей и помехах измерения. В Разделе 1.3 приводится описание рандомизированных и мультиагентных подходов и обосновывается их использование в разработке сетевой модели наблюдения за целями. В Разделе 1.4 формулируется постановка задачи отслеживания целей сетью сенсоров. В Подразделах описываются различные подходы к выбору соседей для каждого сенсора с учетом топологических ограничений. В Подразделе 1.4.1 описывается подход, основанный на оптимизации выбора целей с помощью поиска максимального пересечения доверительных эллипсоидов при минимальной нагрузке на сенсоры. В Подразделе 1.4.2 приводится другой подход, основанный на рандомизации топологии, и описываются его преимущества. В Разделе 1.5 сформулированы выводы из первой главы.

Во **второй главе** формулируются модификации распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, и исследуются свойства его оценок. В Разделе 2.1 описывается распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, для общего вида функции, приводятся условия на эту функцию и показывается, что функция, разработанная в первом Раз-

деле, удовлетворяет этим условиям. В Разделе 2.2 приводится модифицированный распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, для оценки движущихся целей, формулируется и доказывается Теорема 1, отражающая среднеквадратическое качество оценок, получаемых по предложенному алгоритму, а также формулируется и доказывается Теорема 2 для выбора субоптимального шага алгоритма. В Разделе 2.3 описывается взвешенная версия алгоритма, применимая для целей с различным характером движения. Также формулируется и доказывается Теорема 3, которая показывает сходимость матрицы ковариации невязок, полученных с помощью взвешенной версии алгоритма. В Разделе 2.4 приводится алгоритм с переменным шагом для оценки положения статистических объектов, формулируется и доказывается Теорема 4, показывающая скорость сходимости матрицы ковариации невязок, получаемых по описанному алгоритму. В Разделе 2.5 сформулированы выводы из второй главы.

В **третьей главе** приводятся результаты имитационного моделирования, иллюстрирующие работу предложенных методов и подходов. В Разделе 3.1 приводятся результаты численных экспериментов решения задачи отслеживания целей распределенной сетью сенсоров с помощью разработанной модели сети сенсоров из Раздела 1.4 и алгоритмов из Разделов 2.2-2.4. В Разделе 3.2 описывается прототип системы для отслеживания целей распределенной сетью сенсоров. В Подразделе 3.2.1 приводится архитектура системы, в Подразделе 3.2.2 – графический интерфейс системы, а в Подразделе 3.2.3 – примеры работы системы. В Разделе 3.3 описывается применение распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, а именно слежение за летательными аппаратами в режиме реального времени. В Разделе 3.4 сформулированы выводы из третьей главы.

В **заключении** формулируются основные результаты диссертации.

# Глава 1. Модель управления сенсорной сетью на основе рандомизированного и мультиагентного подходов при ограничениях на коммуникации и наличии существенных неопределенностей

## 1.1 Обозначения и определения

В тексте диссертации приняты следующие обозначения:

$t, k$  – дискретное время;

$i, j, k, m, n, d, p, l, q$  – целые числа (обычно неотрицательные);

$\alpha, \beta, \gamma, \delta, \epsilon, a, b, c, z, \sigma$  – скалярные величины;

$\mathbf{a}, \mathbf{b}, \mathbf{x}, \mathbf{r}, \mathbf{s}, \mathbf{u}, \Delta$  – векторные переменные;

$y$  – наблюдаемые скалярные и векторные переменные;

$v, w, \xi$  – помехи (шумы) в наблюдениях (измерениях);

$A, B, C, D, \Psi, I, \Gamma, H, \Phi, S, Z, W$  – матрицы;

$M, Pr, \nabla$  – операторы;

$J$  – матрица или набор чисел;

$\mathcal{N}, \mathcal{M}$  – множества натуральных чисел;

$\mathbb{R}$  – множество вещественных чисел;

$\theta$  – оцениваемое (оптимальное значение);

$\hat{\theta}$  – вектор (иногда матрица) в пространстве оцениваемых параметров (оценка);

$\langle \cdot, \cdot \rangle$  – скалярное произведение векторов;

$f(\cdot, \cdot), F(\cdot), \rho(\cdot, \cdot)$  – вещественные функции;

$\cdot^T$  – транспонирование вектора или матрицы;

$\cdot^{-1}$  – обращение матрицы;

$\cdot'$  – псевдообращение матрицы;

$\|A\|$  – норма Фробениуса:  $\|A\| = \sqrt{\sum_i \sum_j (a_{i,j})^2}$ ;

$\mathbf{0}$  – нулевой вектор;

$\mathbf{1}_d = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^d$  – вектор, состоящий из всех единиц;

$\mathbf{e}_i = \begin{pmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} \in \mathbb{R}^d$  – канонический базисный вектор, где  $i$ -ый элемент

равен 1;

$I_d \in \mathbb{R}^{d \times d}$  – единичная матрица;

$J_d = \mathbf{1}_d \mathbf{1}_d^T \in \mathbb{R}^{d \times d}$  – матрица, состоящая из всех единиц;

$A \otimes B$  – произведение Кронекера, определенное для любых матриц  $A$  и  $B$ ;

$A < B, A \leq B$  – упорядоченность матриц в смысле квадратичных форм;

$\mathcal{N} = \{1, \dots, n\}$  – множество вершин;

$\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$  – множество ребер;

$\mathcal{G}_A = (\mathcal{N}, \mathcal{E})$  – ориентированный граф;

$i \in \mathcal{N}$  – идентификатор  $i$ -й вершины и  $(j, i) \in \mathcal{E}$ , если существует направленное ребро из вершины  $j$  в вершину  $i$ ;

$\mathcal{N}^i = \{j \in \mathcal{N} : (j, i) \in \mathcal{E}\}$  – множество *соседей* для узла  $i \in \mathcal{N}$ ;

$|\cdot|$  – мощность множества;

верхние индексы – индексы агентов;

$a^{i,j} > 0$  – вес, связанный с ребром  $(j, i) \in \mathcal{E}$ ,  $a^{i,j} = 0$  всякий раз, когда  $(j, i) \notin \mathcal{E}$ ;

$\deg_i^+(A)$  – взвешенная степень захода  $i \in \mathcal{N}$   $\deg_i^+(A) = \sum_{j=1}^n a^{i,j}$  в графе  $\mathcal{G}_A$ ;

$\deg_{\max}^+(A)$  – максимальная степень захода среди всех вершин, содержащихся в графе  $\mathcal{G}_A$ ;

$\text{col}\{\cdot\}$  – вектор-столбец, полученный путем наложения его элементов друг на друга;

$\text{diag}_n(\mathbf{b})$  – диагональная матрица с элементами вектора  $\mathbf{b}$  на диагонали и других элементов, равных нулю;

$\mathcal{D}(A)$  – диагональная матрица  $\mathcal{D}(A) = \text{diag}_n(\deg_1^+(A), \dots, \deg_n^+(A))$ ;

$\mathcal{L}(A) = \mathcal{D}(A) - A$  – лапласиан графа  $\mathcal{G}_A$ ;

$\Omega$  – вероятностное пространство;

$\omega$  – элемент вероятностного пространства;

$\mathcal{F}$  – множество всех событий;

$P$  – вероятностная мера;

$(\Omega, \mathcal{F}, P)$  – базовое вероятностное пространство;

$\mathbb{E}$  – математическое ожидание;

$\forall$  – квантор всеобщности;

$\exists$  – квантор существования;

$\mathcal{F}_t$  –  $\sigma$ -алгебра всех вероятностных событий, произошедших до момента времени  $t = 1, 2, \dots$ ;

$\mathbb{E}_{\mathcal{F}_t}$  – условное математическое ожидание относительно  $\sigma$ -алгебры  $\mathcal{F}_t$ . Математически, эта  $\sigma$ -алгебра порождается значениями всех случайных функций (положения цели, помехи, смены графа смежности) в моменты времени  $\tau = \{1, 2, \dots, t\}$ .

**О п р е д е л е н и е 1.** *Ориентированный граф  $\mathcal{G}_A$  называется сильно связным, если для каждой пары вершин  $j, i \in \mathcal{N}$  существует путь направленных ребер, идущий из  $j$  до  $i$ .*

Обозначим расположенные в порядке возрастания действительных частей собственные значения лапласиана  $\mathcal{L}(A)$  через  $\lambda_1, \dots, \lambda_n$ ,

$$0 \leq \operatorname{Re}(\lambda_1) \leq \operatorname{Re}(\lambda_2) \leq \dots \leq \operatorname{Re}(\lambda_n).$$

Известно, что если граф сильно связан, то  $\lambda_1 = 0$  и все остальные собственные значения  $\mathcal{L}$  лежат в открытой правой половине комплексной плоскости (см. [1, 48]). Собственное значение матрицы  $A$  с максимальной абсолютной величиной обозначается как  $\lambda_{\max}(A)$ .

## 1.2 Сетевая модель наблюдения за движущимися объектами

Рассмотрим систему, состоящую из сети из  $n$  сенсоров (измерителей), следящих за движением  $m$  целей (объектов слежения). Будем считать,

что сенсоры и цели размещены в вещественном пространстве  $\mathbb{R}^d$  (например,  $d = 2$  – на плоскости,  $d = 3$  – в воздухе или под водой).

Пусть

$\mathcal{N} = \{1, 2, \dots, n\}$  – множество всех сенсоров,

$\mathcal{M} = \{1, 2, \dots, m\}$  – множество всех целей,

$\mathbf{s}_t^i = \begin{pmatrix} s_t^{i,1} \\ \dots \\ s_t^{i,d} \end{pmatrix} \in \mathbb{R}^d$  – вектор координат сенсора  $i, i \in \mathcal{N}$  в момент времени  $t$ ,

$\mathbf{r}_t^l = \begin{pmatrix} r_t^{l,1} \\ \dots \\ r_t^{l,d} \end{pmatrix} \in \mathbb{R}^d$  – вектор координат цели  $l, l \in \mathcal{M}$ , в момент времени  $t$ ,

$\theta_t = \begin{pmatrix} \mathbf{r}_t^1 \\ \dots \\ \mathbf{r}_t^m \end{pmatrix} \in \mathbb{R}^{dm}$  – общий вектор состояния всех целей в момент времени  $t$ .

Введем предположение о скорости движения целей.

**Предположение 1.** Обозначим разность  $\xi_t^l = \mathbf{r}_t^l - \mathbf{r}_{t-1}^l, l \in \mathcal{M}$ . Скорость изменения положения целей равномерно ограничена:  $\forall l \in \mathcal{M} \|\xi_t^l\| \leq \delta < \infty$ , или же  $\mathbb{E}\|\xi_t^l\|^2 \leq \delta^2, \mathbb{E}\|\xi_t^l\|\|\xi_{t-2}^l\| \leq \delta^2, \mathbb{E}[\tilde{\xi}_t \tilde{\xi}_t^T] \leq Q_\delta$  и  $\mathbb{E}[\tilde{\xi}_t \tilde{\xi}_{t-1}^T] \leq Q_\delta$ , если последовательность  $\{\xi_t^l\}$  случайна.

Будем считать, что в каждый момент времени  $t$  сенсор  $i$  может измерять расстояние до любой из цели  $l$  из некоторого подмножества всех целей  $\mathcal{M}_t^i \subset \mathcal{M}$ :

$$z_t^{i,l} = \rho(i, l) + w_t^{i,l},$$

где  $w_t^{i,l}$  – помехи измерения, а  $\rho(i, l)$  – квадрат расстояния между сенсором  $i$  и целью  $l$  в момент времени  $t$ :

$$\rho(i, l) = \|\mathbf{r}_t^l - \mathbf{s}_t^i\|^2 = \sum_{d'=1}^d (r_t^{l,d'} - s_t^{i,d'})^2.$$

Пусть сеть сенсоров, состоящая из  $n$  узлов, описывается ориентированным графом  $\mathcal{G}_A$  с взвешенной матрицей смежности  $A = [a^{i,j}]$  (матрица связности, связанная с графом  $\mathcal{G}_A$ ). Обозначим подмножество сенсоров  $\mathcal{N}_t^i \subset \mathcal{N}$  с соседями сенсора  $i$  в момент времени  $t$ , а  $\bar{\mathcal{N}}_t^i \subset \mathcal{N}_t^i$  – подмножество соседей  $J = \{j_1, \dots, j_p\} \subset \bar{\mathcal{N}}_t^i$  сенсора  $i$ , каждый из которых в момент времени  $t$  измеряет расстояние до той же цели  $l \in \mathcal{M}_t^i$ , что и сенсор  $i$  ( $p$  – количество одновременных согласованных наблюдений за одной целью разными сенсорами) ([12, 72]). Составим набор троек  $\mathcal{U}_t$ , состоящий из возможных комбинаций сенсора  $i \in \mathcal{N}$ , его соседей  $J \in \bar{\mathcal{N}}_t^i$  и некоторой цели  $l \in \mathcal{M}_t^i$ , за которой они согласовано наблюдают в момент времени  $t$ . Обозначим любую тройку из этого набора  $\mathbf{u} \in \mathcal{U}_t$ .

С практической точки зрения наиболее интересными являются случаи, когда количество соседей минимально для каждого сенсора. В частности, далее будет рассмотрено два варианта, при которых количество одновременных согласованных наблюдений за одной целью либо равно одному  $p = 1$ , либо равно размерности пространства  $p = d$ . Во втором случае, при условии измерения без ошибок сенсорами координат целей, можно воспользоваться стандартными приемами триангуляции.

Для каждого набора  $\mathbf{u} = (i, J, l) \in \mathcal{U}_t$  и некоторого  $j_q \in J$  определим разницу  $\bar{z}_t^q(\mathbf{u}) = z(i, l) - z(j_q, l)$  между измерениями расстояний до одной и той же цели  $l$  сенсора  $i$  и сенсора  $j_q \in J$ . Также введем  $\bar{w}_t^q(\mathbf{u}) = w_t^{i,l} - w_t^{j_q,l}$  как разницу между помехами измерения расстояний до цели  $l$  сенсора  $i$  и сенсора  $j_q \in J$ . Используя формулу разности квадратов, можно для всех  $j_q \in J$  получить  $p$  уравнений для каждого сенсора  $i \in \mathcal{N}$ :



$$\begin{aligned}
\bar{z}_t^q(\mathbf{u}) &= z_t^{i,l} - z_t^{j_q,l} = \rho(i,l) - \rho(j_q,l) + w_t^{i,l} - w_t^{j_q,l} = \\
&= \sum_{d'=1}^d \left( (r_t^{l,d'} - s_t^{i,d'})^2 - (r_t^{l,d'} - s_t^{j_q,d'})^2 \right) + \bar{w}_t^q(\mathbf{u}) = \\
&= \sum_{d'=1}^d (s_t^{j_q,d'} - s_t^{i,d'}) (2r_t^{l,d'} - s_t^{j_q,d'} - s_t^{i,d'}) + \bar{w}_t^q(\mathbf{u}) = \\
&= \sum_{d'=1}^d 2r_t^{l,d'} (s_t^{j_q,d'} - s_t^{i,d'}) - \sum_{d'=1}^d \left( (s_t^{j_q,d'})^2 - (s_t^{i,d'})^2 \right) + \bar{w}_t^q(\mathbf{u}),
\end{aligned}$$

$q = 1, \dots, p$ . Эти уравнения в матричной форме имеют вид

$$2 \begin{bmatrix} (\mathbf{s}_t^{j_1} - \mathbf{s}_t^i)^T \\ \dots \\ (\mathbf{s}_t^{j_p} - \mathbf{s}_t^i)^T \end{bmatrix} \mathbf{r}_t^l = \begin{bmatrix} \bar{z}_t^1(\mathbf{u}) + \|\mathbf{s}_t^{j_1}\|^2 - \|\mathbf{s}_t^i\|^2 \\ \dots \\ \bar{z}_t^p(\mathbf{u}) + \|\mathbf{s}_t^{j_p}\|^2 - \|\mathbf{s}_t^i\|^2 \end{bmatrix} - \begin{bmatrix} \bar{w}_t^1(\mathbf{u}) \\ \dots \\ \bar{w}_t^p(\mathbf{u}) \end{bmatrix}. \quad (1.1)$$

Введем  $C_t^{\mathbf{u}} = 2 \begin{bmatrix} (\mathbf{s}_t^{j_1} - \mathbf{s}_t^i)^T \\ \dots \\ (\mathbf{s}_t^{j_p} - \mathbf{s}_t^i)^T \end{bmatrix}$ ,  $D_t^{\mathbf{u}} = \begin{bmatrix} \bar{z}_t^1(\mathbf{u}) + \|\mathbf{s}_t^{j_1}\|^2 - \|\mathbf{s}_t^i\|^2 \\ \dots \\ \bar{z}_t^p(\mathbf{u}) + \|\mathbf{s}_t^{j_p}\|^2 - \|\mathbf{s}_t^i\|^2 \end{bmatrix}$ ,

$W_t^{\mathbf{u}} = \begin{bmatrix} \bar{w}_t^1(\mathbf{u}) \\ \dots \\ \bar{w}_t^p(\mathbf{u}) \end{bmatrix}$ . Уравнение (1.1) представляет собой линейную систему

относительно вектора  $\mathbf{r}_t^l$ .

Будем считать, что измерения проводятся без помех, то есть  $W_t^{\mathbf{u}} = \mathbf{0}$ . Выразим вектор  $\mathbf{r}_t^l$  через  $C_t^{\mathbf{u}}$  и  $D_t^{\mathbf{u}}$ . В общем случае, при  $p < d$  матрица  $C_t^{\mathbf{u}}$  не является квадратной, поэтому вместо операции обращения матрицы будем использовать операцию псевдообращения:

$$\begin{aligned}
C_t^{\mathbf{u}} \mathbf{r}_t^l &= D_t^{\mathbf{u}}, \\
C_t^{\mathbf{u}T} C_t^{\mathbf{u}} \mathbf{r}_t^l &= C_t^{\mathbf{u}T} D_t^{\mathbf{u}}, \\
[C_t^{\mathbf{u}T} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}} \mathbf{r}_t^l &= [C_t^{\mathbf{u}T} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}T} D_t^{\mathbf{u}}.
\end{aligned}$$

В итоге, получаем

$$I_t^{\mathbf{u}} \mathbf{r}_t^l = H_t^{\mathbf{u}}, \quad (1.2)$$

где  $I_t^{\mathbf{u}} = [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}$ ,  $H_t^{\mathbf{u}} = [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} D_t^{\mathbf{u}}$ .

**З а м е ч а н и е 1.** Если  $p = d$  и матрица  $C_t^{\mathbf{u}}$  невырожденная, то  $I_t^{\mathbf{u}}$  – единичная матрица, а  $H_t^{\mathbf{u}} = [C_t^{\mathbf{u}}]^{-1} D_t^{\mathbf{u}}$ . Если  $p = 1$ , то все элементы матрицы  $I_t^{\mathbf{u}}$  являются нулями, кроме одного элемента на диагонали, который равен единице.

Цель сетевой системы сенсоров состоит в формировании в каждый момент времени  $t$  состоятельной оценки местоположения всех объектов

слежения  $\hat{\theta}_t = \begin{pmatrix} \hat{\mathbf{r}}_t^{1,1} \\ \vdots \\ \hat{\mathbf{r}}_t^{m,n} \end{pmatrix} \in \mathbb{R}^{nmd}$ , которая минимизируют функцию потерь

$$\hat{\theta}_t = \underset{\hat{\theta}_t}{\operatorname{argmin}} \mathbb{M} \sum_{i \in \mathcal{N}} F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) \quad (1.3)$$

где

$$F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \sum_{\mathbf{u} \in \mathcal{U}_t, i=i(\mathbf{u})} f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}),$$

$\hat{\mathbf{r}}_t^{l,i}$  – оценка местоположения цели  $l$  сенсором  $i$ ,  $F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i})$  – функционал, в который входят цели, за которыми следит сенсор  $i$ ,  $f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \|I_t^{\mathbf{u}} \hat{\mathbf{r}}_t^{l,i} - H_t^{\mathbf{u}}\|^2$  – функция потерь (или качества), а  $\mathbb{M}$  – либо оператор усреднения, либо оператор максимизации. Далее будет рассматриваться только случай с усреднением:

$$\hat{\theta}_t = \underset{\hat{\theta}_t}{\operatorname{argmin}} \frac{1}{|\mathcal{U}_t|} \sum_{\mathbf{u} \in \mathcal{U}_t} f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \underset{\hat{\theta}_t}{\operatorname{argmin}} \frac{1}{|\mathcal{U}_t|} \sum_{i \in \mathcal{N}} F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}), \quad (1.4)$$

где

$$F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \sum_{\mathbf{u} \in \mathcal{U}_t, i=i(\mathbf{u})} f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}).$$

На практике любые измерения, полученные сенсорами, всегда искажены помехами [25, 29]. Решение задачи оптимизации усложняется при измерениях с помехами. Для случая ненулевых помех  $W_t^{\mathbf{u}}$  измерения сенсоров формируются по зашумленным измерениям оптимизируемой функции

$$y_t^i = \|I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}} + [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2.$$

Применяя формулу квадрата суммы, последовательно получаем

$$\begin{aligned} y_t^i &= \\ & \|I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}}\|^2 + 2(I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} + \|[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2 = \\ & f_t^i(\mathbf{u}, \mathbf{x}_t^{l,i}) + 2(I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} + W_t^{\mathbf{u}\top} C_t^{\mathbf{u}} C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} = \\ & f_t^i(\mathbf{u}, \mathbf{x}_t^{l,i}) + 2(I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} + \|C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2 = \\ & f_t^i(\mathbf{u}, \mathbf{x}_t^{l,i}) + v_t^i, \end{aligned} \quad (1.5)$$

где  $f_t^i(\mathbf{u}, \mathbf{x}_t^{l,i}) = \|I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}}\|^2$ ,  $\mathbf{x}_t^{l,i}$  – текущая точка, в которой производится измерение функции (точка наблюдения), например,  $\mathbf{x}_t^{l,i} = \hat{\mathbf{r}}_t^{l,i}$ , а  $v_t^i = 2(I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} + \|C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2$  – некоторая помеха.

Сформулируем Предположения о сенсорах, точках наблюдения и помехах измерения.

## Предположение 2.

1. Расстояние между сенсором  $\mathbf{s}_t^i$  и сенсорами  $\mathbf{s}_t^{j_1}, \dots, \mathbf{s}_t^{j_p}$  ограничено сверху и снизу  $\forall i \in \mathcal{N}, j_q \in J, J \subset \bar{\mathcal{N}}_t^i, c_s \leq \|\mathbf{s}_t^i - \mathbf{s}_t^{j_q}\| \leq c_S, q = 1, \dots, p$ .
2. Известны ограничения на область нахождения целей. Другими словами, для всех  $l \in \mathcal{M}$  точки наблюдения  $\mathbf{x}_t^{l,i}$  выбираются таким образом, чтобы они находились в выпуклом замкнутом ограниченном подмножестве, которые содержат искомый вектор  $\mathbf{r}_t^l$ .

3. Для всех  $i \in \mathcal{N}, l \in \mathcal{M}$  помехи  $w_t^{i,l}$  – независимые, симметрично распределенные центрированные помехи, равномерно ограниченные в среднеквадратическом смысле  $\mathbb{E}\|w_t^{i,l}\| \leq \sigma_w$  и с равномерно ограниченными четвертыми моментами  $\mathbb{E}\|w_t^{i,l}\|^4 \leq \sigma_4$ .

**Л е м м а 1.** Если выполняется Предположение 2, то

1.  $\|C_t^{\mathbf{u}\mathbf{T}}\|^4 \leq 4p^2c_S^4$ ;
2.  $\|[C_t^{\mathbf{u}\mathbf{T}}C_t^{\mathbf{u}}]'C_t^{\mathbf{u}\mathbf{T}}\|^2 \leq \frac{c_S^2}{2pc_S^2}$ ;
3.  $\|I_t^{\mathbf{u}}\mathbf{x}_t^{l,i} - H_t^{\mathbf{u}}\|^2 \leq c_f$ ;
4.  $\mathbb{E}\|W_t^{\mathbf{u}}\|^2 \leq p\sigma_w^2$ ,  $\mathbb{E}\|W_t^{\mathbf{u}}\|^4 \leq p\sigma_4$ ;
5. неизвестные помехи

$$v_t^i = 2(I_t^{\mathbf{u}}\mathbf{x}_t^{l,i} - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\mathbf{T}}C_t^{\mathbf{u}}]'C_t^{\mathbf{u}\mathbf{T}}W_t^{\mathbf{u}} + \|C_t^{\mathbf{u}\mathbf{T}}W_t^{\mathbf{u}}\|^2$$

являются ограниченными в среднеквадратичном смысле  $\mathbb{E}(v_t^i)^2 \leq c_v^2$ ;

6. последовательные разности  $\tilde{v}_k^i = v_{2k}^i - v_{2k-1}^i$  помех ограничены:  $|\tilde{v}_k^i| \leq c_{\tilde{v}} < \infty$  или  $\mathbb{E}(\tilde{v}_k^i)^2 \leq c_{\tilde{v}}^2$ , если последовательность  $\{\tilde{v}_k^i\}$  случайна.

*Доказательство.*

1. Если выполнено Предположение 2.1, то  $\|C_t^{\mathbf{u}\mathbf{T}}\|^2 \leq 2pc_S^2$  и  $\|C_t^{\mathbf{u}\mathbf{T}}\|^4 \leq 4p^2c_S^4$ .
2. Если выполнено Предположение 2.1, то

$$4p^2c_s^2 \leq \|[C_t^{\mathbf{u}\mathbf{T}}C_t^{\mathbf{u}}]\|^2 \leq 4p^2c_S^2,$$

$$\frac{1}{4p^2c_S^2} \leq \|[C_t^{\mathbf{u}\mathbf{T}}C_t^{\mathbf{u}}]'\|^2 \leq \frac{1}{4p^2c_s^2},$$

$$\frac{2pc_s^2}{4p^2c_s^2} \leq \|[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top}\|^2 \leq \frac{2pc_s^2}{4p^2c_s^2},$$

$$\|[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top}\|^2 \leq \frac{c_s^2}{2pc_s^2}.$$

3. Если выполнено Предположение 2.2, то  $\|I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}}\|^2 \leq c_f$ .
4. Если выполнено Предположение 2.3, то  $\mathbb{E}\|\bar{w}_t^q(\mathbf{u})\|^2 \leq \sigma_w^2$ ,  $\mathbb{E}\|\bar{w}_t^q(\mathbf{u})\|^4 \leq \sigma_4$  и  $\mathbb{E}\|W_t^{\mathbf{u}}\|^2 \leq p\sigma_w^2$ ,  $\mathbb{E}\|W_t^{\mathbf{u}}\|^4 \leq p\sigma_4$ .
5. Если выполнены Предположения 2.1-2.3, то

$$\mathbb{E}v_t^i = 2(I_t^{\mathbf{u}} x_t^l - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} \mathbb{E}W_t^{\mathbf{u}} + \mathbb{E}\|C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2 = \mathbb{E}\|C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2,$$

$$\begin{aligned} \mathbb{E}(v_t^i)^2 &= 4\mathbb{E}\|(I_t^{\mathbf{u}} x_t^l - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2 + \\ &4(I_t^{\mathbf{u}} x_t^l - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} \mathbb{E}\left(W_t^{\mathbf{u}}\|C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2\right) + \mathbb{E}\|C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^4 = \\ &4\mathbb{E}\|(I_t^{\mathbf{u}} x_t^l - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2 + \mathbb{E}\|C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^4 \leq \\ &16p^3 c_f c_s^2 \sigma_w^2 + 4p^3 c_s^2 \sigma_4. \end{aligned}$$

Таким образом, помеха  $v_t^i$  не является центрированной и  $\mathbb{E}(v_t^i)^2 \leq c_v^2$ , где  $c_v^2 = 16p^3 c_f c_s^2 \sigma_w^2 + 4p^3 c_s^2 \sigma_4$ . Другими словами,  $v_t^i$  является *неизвестной, но ограниченной помехой*.

6. Если выполнены Предположения 2.1-2.3, то

$$\mathbb{E}(\tilde{v}_k^i)^2 = \mathbb{E}(v_{2k}^i - v_{2k}^i)^2 \leq c_v^2,$$

где  $c_v^2 = 2c_v^2 = 8pc_f c_s^2 \sigma_w^2 + 2pc_s^2 \sigma_4$ .

□

## 1.3 Рандомизированный и мультиагентный подходы

Рандомизация зарекомендовала себя как эффективный инструмент для решения задач с неопределенностями [6], которые часто возникают при сборе данных с помощью сенсорных сетей. Более того, она значительно сокращает размерность задачи [111]. Мультиагентность, в свою очередь, является подходом, развивающимся на стыке распределенных вычислений и эмерджентного интеллекта, что позволяет включать коммуникационные ограничения в описанную сетевую модель для наблюдения за движущимися объектами.

Развитие и доступность вычислительной техники оказали влияние на классические области математической статистики, способствуя развитию и приоритетному использованию повторяющихся схем оценки. В основе относительно нового подхода к решению задач оценивания и оптимизации в неблагоприятных условиях (например, при наличии вырожденных последовательностей наблюдений) лежит использование пробных возмущений [6]. Если при решении задачи через входные каналы системы или алгоритма можно внести новое возмущение с заданными экспериментатором или хорошо известными статистическими свойствами, то оно может быть использовано для “обогащения” информации в канале наблюдения. Иногда роль пробного возмущения может играть измеримый случайный процесс, присутствующий в системе. В системах управления испытания могут быть добавлены через контрольный канал, а в других случаях роль пробного воздействия может выполнять рандомизированный дизайн эксперимента. Изучение обновленной системы с пробным возмущением, даже с использованием традиционных методов, часто приводит к обнадеживающим результатам относительно сходимости и применимости новых алгоритмов. Одним из замечательных свойств таких алгоритмов является сохранение оценок при “почти произвольных” возмущениях. Алгоритмы, в которых один или несколько

шагов основаны на случайном выборе правил, называются рандомизированными алгоритмами.

Проект Манхэттен [110] сыграл важную роль в развитии компьютеров. В XVIII веке точные предсказания возвращения комет и затмений были обеспечены благодаря бумажным расчетам. Однако до 1940-х годов выбор оргтехники, такой как счетные машины, сортировщики карт и простые механические калькуляторы, доступной для помощи людям, был ограничен. В рамках проекта Манхэттен в Лос-Аламосе, где работали ученые и инженеры, требовалась большая точность вычислений, что привело к использованию аналоговых компьютеров, включая компьютеры с перфокартами производства IBM. Этот сдвиг в развитии компьютеров и возможность точных вычислений спровоцировали появления метода Монте-Карло [82], названного в честь города Монте-Карло в княжестве Монако, где рулетка являлась популярным генератором случайных чисел. Таким образом, проект Манхэттен и метод Монте-Карло стали важными этапами в развитии компьютерной техники и применении случайных чисел в вычислениях.

Существует класс задач, сложность которых растет экспоненциально с размерностью задачи. Некоторые задачи позволяют найти алгоритмы, сложность которых растет медленнее, но есть и множество задач, для которых это невозможно, например, вычисление объема выпуклого тела в  $n$ -мерном евклидовом пространстве или вычисление  $n$ -мерных интегралов. В таких случаях метод Монте-Карло является единственным способом получить достаточно точные результаты за приемлемое время. Современные исследования в основном направлены на разработку эффективных алгоритмов Монте-Карло для различных физических, химических и социальных процессов, а также для параллельных вычислительных систем. Метод Монте-Карло также получил развитие в квантовой теории. Существует квантовый метод Монте-Карло [50], который широко применяется в исследовании сложных молекул и твердых тел.

На математико-механическом факультете СПбГУ методом Монте-

Карло с 1956 года занимается Сергей Михайлович Ермаков [10]. Его научные интересы включают теорию метода Монте-Карло, оценку многомерных интегралов, исследование генераторов псевдослучайных чисел, а также вероятностное решение линейных и нелинейных интегральных уравнений. Его работы способствовали формализации этого вычислительного метода и превращению его из набора полуэмпирических приемов в строго очерченную область математики.

Многие практические задачи, возникающие в различных областях, могут быть сформулированы как задачи поиска корня функции или поиска точки минимума, однако аналитическое решение таких задач часто оказывается невозможным из-за сложности функций или ограничений, с которыми они связаны. В таких случаях методы случайного поиска представляется как альтернативный подход, который может быть применен для нахождения приближенного решения. Метод случайного поиска основан на итеративном построении последовательности оценок решения. На каждом шаге новая оценка выбирается путем сдвига предыдущей оценки в случайно выбранном направлении. Этот подход является развитием метода проб и ошибок, при котором решение ищется случайным образом, и если оно удовлетворяет заданным критериям, принимается, в противном случае отвергается с целью продолжения поиска новой возможности. Основная идея заключается в использовании случайности для исследования различных областей пространства параметров, включая искомое решение. Метод случайного поиска нашел широкое применение в различных областях, включая оптимизацию функций, компьютерное моделирование и статистику. Этот метод позволяет исследовать пространство параметров, учитывая случайность, и достичь оптимального или приближенного решения задачи. Подробное описание метода случайного поиска можно найти в книге Растригина “Статистические методы поиска” [20], где представлены теоретические основы метода, его различные варианты и примеры его применения в практических задачах.

Одним из примеров рандомизированных алгоритмов, который получил значительное внимание, является генетический алгоритм оптими-



зации. Основы этого алгоритма были заложены Нилсом Ааллом Барричелли, который в 1954 году провел первые работы по моделированию эволюции на компьютерах в Институте продвинутых исследований в Принстонском университете [39]. В 1957 году австралийский генетик Алекс Фрейзер начал публиковать серию статей о моделировании искусственного отбора среди организмов с несколькими измеримыми характеристиками. Работа Фрейзера послужила переломной точкой в развитии компьютерных моделей эволюционных процессов и методов. С течением времени, с растущим интересом и прогрессом в области вычислительной мощности, генетические алгоритмы нашли практическое применение в различных областях. По мере того, как персональные компьютеры развивались и их вычислительные возможности значительно увеличивались, улучшилась возможность реализации генетических алгоритмов для решения сложных задач оптимизации. Генетические алгоритмы имитируют процессы естественного отбора и генетики, используя такие техники, как отбор, скрещивание и мутация, чтобы исследовать пространство решений и сходиться к оптимальным или близким к оптимальным решениям. Генетические алгоритмы демонстрируют свою эффективность в широком спектре приложений, включая задачи оптимизации в инженерии, финансах, логистике и искусственном интеллекте [40].

Метод отжига – это метод оптимизации, тоже основанный на упорядоченном случайном поиске. История этого метода начинается в 1953 году, когда Н. Метрополис разработал алгоритм для симуляции достижения равновесия в системе с несколькими степенями свободы при заданной температуре [74]. Затем, в начале 1980-х годов, С. Киркпатрик предложил идею использования этого алгоритма для решения различных задач оптимизации, выходящих за пределы моделирования физических систем [57]. Один из основных преимуществ метода отжига заключается в его способности избегать локальных минимумов оптимизируемой функции и продолжать поиск глобального минимума.

Поиск с запретами, также известный как табу-поиск, представляет собой мета-алгоритм поиска, использующий методы локального поиска

для решения задач математической оптимизации. Этот алгоритм был создан Фредом У. Гловером в 1986 году [62]. Все дело в том, что локальный поиск проверяет потенциальное решение задачи и его непосредственных соседей (то есть решения, которые отличаются только небольшими деталями) в надежде найти улучшенное решение. Однако методы локального поиска могут застрять в субоптимальных областях или на плато, где многие решения одинаково подходят. Поиск с запретами улучшает производительность локального поиска, ослабляя его основное правило. На каждом шаге может быть принято ухудшение, если нет никакого улучшения (аналогично ситуации, когда поиск застревает на локальном минимуме). Кроме того, запреты (табу) вводятся для предотвращения поиска по уже посещенным решениям. При реализации поиска с запретами используются структуры, которые описывают посещенные решения или пользовательские наборы правил. Если потенциальное решение было посещено в течение некоторого короткого периода времени или нарушает правило, оно помечается как “табу”, чтобы алгоритм не рассматривал решение повторно.

Протокол сплетен (Gossip), связанный с рандомизированными алгоритмами, происходит от алгоритмов эпидемической репликации, описанных в исследовании “Эпидемические алгоритмы обслуживания реплицированных баз данных” (1987) [56]. С того времени применение такого протокола стало объектом значительного интереса в области вычислений, и первые практические применения были замечены в системах маршрутизации компьютерных сетей, которые предшествовали сети Интернет. Протоколы сплетен – это децентрализованные протоколы обмена информацией, основанные на идее распространения информации через случайные связи между узлами системы. В таких протоколах узлы передают информацию друг другу, опираясь на случайные контакты или случайно выбираемых соседей. Это позволяет эффективно распространять информацию в сети и обнаруживать глобальные свойства или решать задачи, основываясь на локальных обменах информацией. Протоколы сплетен получили широкое применение в различных областях. В сетях связи они

использовались для маршрутизации пакетов данных, обнаружения и исправления ошибок, управления ресурсами и принятия коллективных решений. Также протоколы сплетен широко исследовались в области социальных сетей, где они использовались для моделирования распространения информации и влияния в сообществах. Преимущества протоколов сплетен заключаются в их устойчивости к отказам и динамическим изменениям в системе. Благодаря случайным связям и распространению информации через соседей, они способны эффективно адаптироваться к изменениям в сети и продолжать функционировать при отказах отдельных узлов. Кроме того, протоколы сплетен не требуют глобального знания о состоянии системы и могут работать в условиях ограниченной доступности информации. В целом, протоколы сплетен представляют собой важный класс рандомизированных алгоритмов, широко применяемых для обмена информацией и координации в децентрализованных системах.

Любые измерения, полученные сенсорами, всегда искажены помехами. В литературе, как правило, рассматриваются помехи, обладающие некоторыми известными нормальными статистическими свойствами [42, 77, 78]. Задача управления сенсорной сетью значительно усложняется при рассмотрении произвольных внешних по отношению к системе неопределенностей, например, при ограниченных, но в остальном неизвестных [64]. В связи с этим наибольший интерес представляют исследования в области стохастической оптимизации, направленные на уменьшение необходимых для работоспособности алгоритмов оптимизации априорных знаний относительно неопределенностей в задаче [63].

Сегодня стохастическая аппроксимация имеет широкий спектр приложений в таких областях как: адаптивная обработка сигналов, адаптивное размещение ресурсов в коммуникационной сети, идентификация систем, адаптивное управление и других. В работах Б. Т. Поляка [17], Дж. С. Спалла (J. C. Spall) [112, 113], В. С. Боркара (V. S. Borkar) [43], А. Б. Цыбакова [17], Х. Кушнера (H. Kushner) и Г. Г. Ина (G. G. Yin) [76] стохастическая аппроксимация используется с убывающими со временем

до нуля размерами шагов. Сейчас возрастает использование алгоритмов стохастической аппроксимации для оптимизации нестационарных функционалов качества (изменяющихся со временем). В таких задачах отслеживания изменений параметров часто используют достаточно малый, но постоянный размер шага. Д. П. Деревицкий и А. Л. Фрадков в [8, 9] при анализе динамики алгоритмов адаптации, основанном на построении приближенных усредненных моделей, обосновали возможность использования алгоритмов стохастической аппроксимации с неубывающим до нуля размером шага. Позже исследование оптимизации нестационарных функционалов рассматривалось в работах О. Н. Граничина [5, 7], Н. О. Амелиной [38, 64], Дж. С. Спалла (J. C. Spall) [122], В. С. Боркара (V. S. Borkar) [43].

Еще одним рандомизированным алгоритмом является рандомизированный алгоритм стохастической оптимизации, известный также как Simultaneous Perturbation Stochastic Approximation (SPSA) [111]. Он отличается тем, что способен решать задачи оптимизации [54] в присутствии произвольных, но ограниченных помех и изменяющихся со временем параметров системы [5, 7, 17]. Эти неопределенности могут быть неслучайными, и даже если они являются случайными, их статистические характеристики необязательно известны [108]. Основная особенность SPSA заключается в том, что он использует одновременные возмущения на входе, а также то, что вместо оценки градиента, как в традиционных методах оптимизации, SPSA оценивает аппроксимацию градиента на основе случайных возмущений функции и соответствующих изменений целевой функции. SPSA обладает несколькими преимуществами. Во-первых, он эффективно работает в условиях ограниченных, но в целом неизвестных помех, так как не требует информации об их точных свойствах. Во-вторых, SPSA позволяет сократить вычислительные затраты, так как оценка градиента производится только на основе двух функции значений, полученных с помощью случайных возмущений.

Оригинальная идея следующего метода – муравьиного алгоритма – исходит от наблюдения за муравьями в процессе поиска кратчайшего

пути от колонии до источника питания [53].

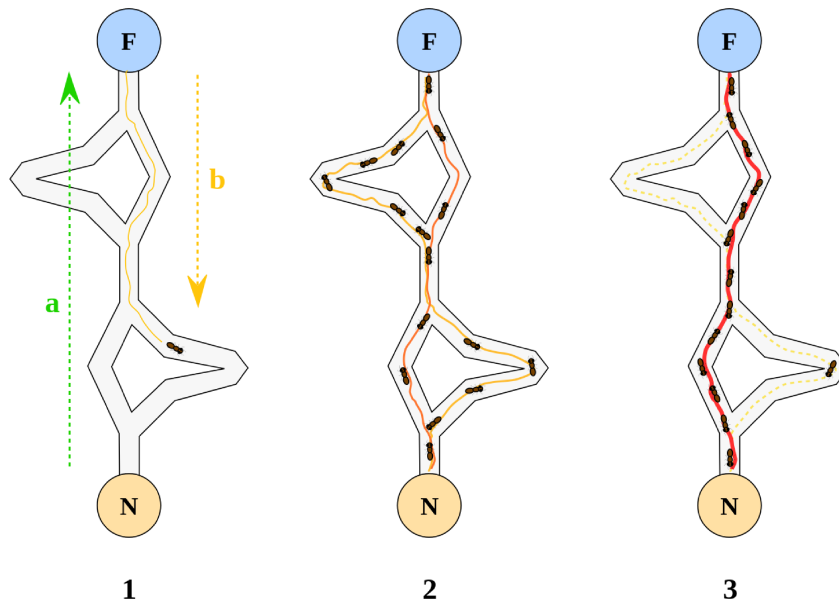


Рис. 1.1: Муравьиный алгоритм

Главное задачей муравьев является поиск пищи. Таким образом, первый муравей находит источник пищи (F) любым способом (a), а затем возвращается к муравейнику (N), оставив за собой тропу из феромонов (b) (Рис. 1.1). Затем муравьи выбирают один из нескольких возможных путей, укрепляют его и делают привлекательным. Муравьи выбирают кратчайший маршрут, так как феромоны с более длинных путей быстрее испаряются. Среди экспериментов по выбору между двумя путями неравной длины, ведущих от колонии к источнику питания, биологи заметили, что, как правило, муравьи используют кратчайший маршрут [53]. Модель такого поведения заключается в следующем.

Колония муравьев функционирует посредством механизма, известного как стигмергия, который включает обмен информацией с использованием феромонов. Когда муравей отправляется в поисках пищи, он оставляет за собой след феромонов по пути возвращения к муравейнику. Эти феромоны привлекают других муравьев, находящихся вблизи, и увеличивают вероятность их последовательности по этому же пути. При возвращении в муравейник эти муравьи укрепляют след феромо-

нов. Со временем, если существует несколько путей, большее количество муравьев проходит по более короткому пути, делая его более привлекательным и приводя к исчезновению более длинных путей из-за испарения феромонов. Этот метод коммуникации, использующий окружающую среду, демонстрирует самоорганизацию [55] и применим к различным социальным животным, таким как, например, термитам.

Система колонии муравьев отражает как положительную, так и отрицательную обратную связь. Присутствие других муравьев укрепляет след феромонов (положительная обратная связь), в то время как испарение феромонов действует как отрицательная обратная связь. Без обратной связи было бы невозможно выбрать путь, если количество феромонов оставалось неизменным со временем. Однако небольшие колебания, усиленные обратной связью, приводят к укреплению одного пути, в конечном итоге стабилизируя систему в сторону выбора самого короткого пути. Следует отметить, что существуют современные вариации алгоритма колонии муравьев, включая элитарную систему муравьев, MMAS (Max-Min Ant System), ранжированную систему муравьев и длинную ортогональную колонию муравьев.

Многие алгоритмы мультиагентной координации, которые используются сейчас, например, в блокчейне [114], мобильной робототехнике [18, 55] и эмерджентном интеллекте [55, 100] также вдохновлены динамикой природных, социальных и экономических систем [41, 44, 93].

Основу мультиагентной системы составляют интеллектуальные агенты, которые обладают свойством автономности (способность работать для достижения поставленной цели без вмешательства человека или других компонент системы) и способностью к проактивному поведению (при управлении своим поведением агент принимает во внимание не только информацию, полученную им онлайн из внешнего мира, но и предысторию своих действий и состояний внешней среды, которые нашли отражение в его текущем внутреннем состоянии, например, сообщения, полученные от других агентов) [3]. В мультиагентной архитектуре динамически

обновляемая информация, получаемая агентом извне и формируемая самим агентом, вместе с локально доступной информацией от ограниченного числа соседей обрабатывается им или пересылается другому агенту. При этом существенно сокращаются как ресурсные и временные затраты на коммуникации в сети, так и время на обработку и принятие решений, затрачиваемые всей системой (в центре, если он есть в системе). Более того, мультиагентные системы также являются гибкими: такая система может быть дополнена и модифицирована без переписывания значительной части программы [32, 103].

Недавнее исследование [99] показало, что бактерии способны взаимодействовать друг с другом, что значительно изменило наше представление о поведении простых организмов, населяющих окружающий нас мир. Бактерии используют молекулярную сигнализацию для измерения плотности своей популяции. Термин “чувство кворума” (Quorum Sensing, QS) относится к процессу, при котором бактериальная клетка способна ощущать активность других бактерий посредством концентрации популяции. Различные виды бактерий, находящиеся в одной среде, используют разнообразные сигнальные молекулы, что позволяет им взаимодействовать с другими видами бактерий. В настоящее время эти системы кворума активно исследуются для различных категорий бактерий. Существуют значительные достижения в понимании генетики, геномики, биохимии и разнообразия сигнальных молекул QS, включая взаимосвязь между чувством кворума и социальным поведением бактерий. Поведение и эволюция таких бактериальных сообществ рассматриваются как естественные примеры мультиагентных систем, поскольку взаимодействие между бактериями происходит локально, и измерение плотности бактерий может быть выполнено без необходимости собирать все данные в одном центре обработки информации. Бактерии успешно решают глобальную задачу, такую как измерение плотности населения, используя только локальное взаимодействие, что является примером мультиагентного алгоритма.

Еще одним важным примером самоорганизации являются ДНК-вы-

числения [35]. Для объяснения концепции ДНК-вычислений рассмотрим известную комбинаторную задачу поиска Гамильтонова пути в графе. Как известно, эта задача состоит в том, чтобы найти такой путь в неориентированном или ориентированном графе, чтобы он проходил через каждую вершину графа ровно один раз. Вычисления ДНК, основанные на заложенных в природе принципах самоорганизации [105], способны решать такие задачи с линейными затратами по времени.

Напомним, что молекулы ДНК обычно образуют двойную спираль, состоящую из нуклеотидов, содержащих фосфатную группу, сахарную группу и азотистое основание. Существует четыре различных нуклеотида: аденин (“А”), тимин (“Т”), гуанин (“Г”), цитозин (“С”). В спирали нуклеотиды соединяются парами в соответствии с фундаментальным *правилom комплементарности*: “А” всегда комплементарно “Т”, а “Г” всегда комплементарно “С” [120]. Также с помощью нагревания возможно разделять двухцепочечную ДНК на две отдельные нити или совершать обратное действие (*лигирование*) с помощью особого фермента *лигазы*.

Рассмотрим ориентированный граф  $G = (\mathcal{N}, E)$  с  $|\mathcal{N}| = n$  узлов. Пусть случайные последовательности ДНК, состоящие из двадцати нуклеотидов, представляют собой узлы (вершины). Ребро связано с последовательностью ДНК, полученной совмещением комплементарной последовательности второй половины начального узла и комплементарной последовательности первой половины конечного узла. Например, если первый узел имеет код ДНК

$$TCAGTACCAG \quad TACAGTCACA$$

комплементарная последовательность :

$$(AGTCATGGTC \quad ATGTCAGTGT),$$

где А – аденин, Т – тимин, G – гуанин, С – цитозин, второй узел имеет



код ДНК

*TAGGTATGCT CAGATAAAGG*

комплементарная последовательность :

*(ATCCATACGA GTCTATTTCC),*

и есть ребро от первого узла до второго, то это ребро кодируется как

*ATGTCAGTGT ATCCATACGA.*

Эта процедура повторяется с каждым доступным ребром [26, 27]. Необходимо отметить, что направления нитей ДНК опущено для простоты.

Далее, все описанные коды ДНК синтезируются и смешиваются в одной пробирке для проведения лигирования (реакции, соединяющей молекулы по правилу комплементарности). Таким образом, все возможные пути создаются одновременно и параллельно. Пример такого пути (молекулы) показан на рисунке 1.2.

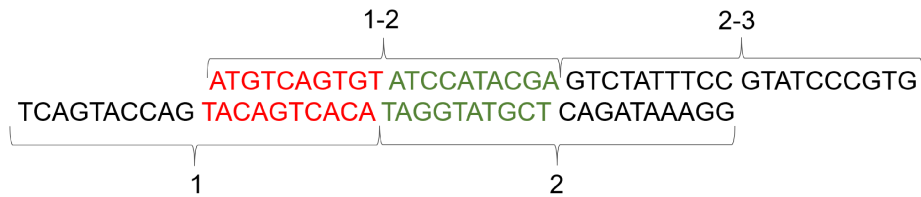


Рис. 1.2: Пример случайной молекулы после реакции лигирования

В этом процессе просматривается аналогия с мультиагентными технологиями. Путем формирования нитей и звеньев в массивные молекулы ДНК, как если бы они были агентами каскадного процесса, нуклеотиды обеспечивают решение проблемы. Последующие шаги предназначены для проверки того, находятся ли молекулы, кодирующие Гамильтонов путь, внутри пробирки или нет. Затраты времени на вычисления ДНК являются результатом локальных взаимодействий и кластеризации, линейно увеличивающихся по мере увеличения количества узлов [22–24]. Далее в [106] эта характеристика сравнивалась с той же, что и в методе ветвей и границ. Выяснилось, что ДНК-вычисления работают намного лучше при параметрах разреженности матрицы смежности, равных

0.85, 0.9, 0.8, и увеличении числа узлов графа с 37, 43, 45 соответственно. В [31] также рассматривалась задача коммивояжера, решенная с помощью ДНК-вычислений.

Мультиагентные алгоритмы также возникают в социальных сетях. В [93] описываются модели социальных процессов, которые были разработаны одновременно с теорией мультиагентных систем.

Протокол локального голосования является еще одним примером мультиагентного подхода, который обеспечивает децентрализованное принятие решений и координацию между агентами в сети [37, 38, 68]. Согласно этому протоколу, задачи, поступающие на узлы сети, могут быть равномерно распределены между всеми узлами путем их коммуникации (обмена данными между соседними узлами). Он широко применяется в различных областях. В контексте управления крыльями самолета во время турбулентности [92] протокол локального голосования доказывает свою эффективность в уменьшении влияния турбулентности. Путем использования сети маленьких вращающихся блоков, оснащенных датчиками давления, протокол обеспечивает локальную обратную связь и обмен информацией между соседними блоками. Это позволяет блокам регулировать свое положение на основе сравнения давлений, в конечном итоге достигая состояния согласия, при котором интегрированные силы давления становятся одинаковыми для всех блоков. Это демонстрирует способность мультиагентных систем к адаптации и самоорганизации для решения сложных задач, связанных с турбулентностью. Аналогично, протокол локального голосования показал свою эффективность в балансировке нагрузки в крупномасштабных сетях [38, 115]. Путем перераспределения задач между агентами на основе локального обмена информацией протокол стремится достичь равномерного распределения нагрузки. Это оптимизирует общее время выполнения задач и уменьшает задержки, повышая эффективность сети.

Одним из возможных решений задачи управления сенсорной сетью с учетом топологических ограничений является распределение наблюдае-

мых объектов между сенсорами, что было описано в [102]. Существенными недостатками этого метода являются большая трудоемкость и необходимость априорного знания о том, какой объект ближе к какому сенсору. Альтернативный подход – использование рандомизации в выборе наблюдаемого объекта, а также рандомизации и мультиагентного подхода внутри сенсорной сети для выбора соседних сенсоров (передающих данные между собой) [32, 103].

Описанные выше примеры подтверждают актуальность и необходимость использования рандомизированного и мультиагентного подходов в задаче управления сенсорной сетью, которая будет описана далее.

## 1.4 Задача управления сенсорной сетью

Существуют различные способы формирования оценок местоположения движущихся целей на основании измерений сенсоров. Традиционно используются централизованные алгоритмы, в которых предполагается, что все сенсоры передают измерения расстояний до целей в общий центр обработки данных для оценки неизвестного вектора (см. Рис. 1.3 слева). Если наблюдаемых целей небольшое количество, то такой подход работает достаточно хорошо. Однако если количество целей растёт, то сбор данных, их передача в центр и их обработка становятся “узким местом” [58, 85]. Альтернативным способом является использование распределенного подхода, в котором сенсоры рассматриваются как автономные вычислительные центры (агенты), способные обрабатывать локально доступную информацию от ограниченного числа соседей и обмениваться данными с соседями. При этом могут существенно сокращаться как затраты на коммуникации в сети, так и время на обработку и принятие решений, затрачиваемые всей системой (см. Рис. 1.3 справа).

При распределенном подходе отдельной задачей является выбор соседей для коммуникации. С одной стороны, сенсор не может постоянно обмениваться данными со всеми узлами сети, так как это снова при-

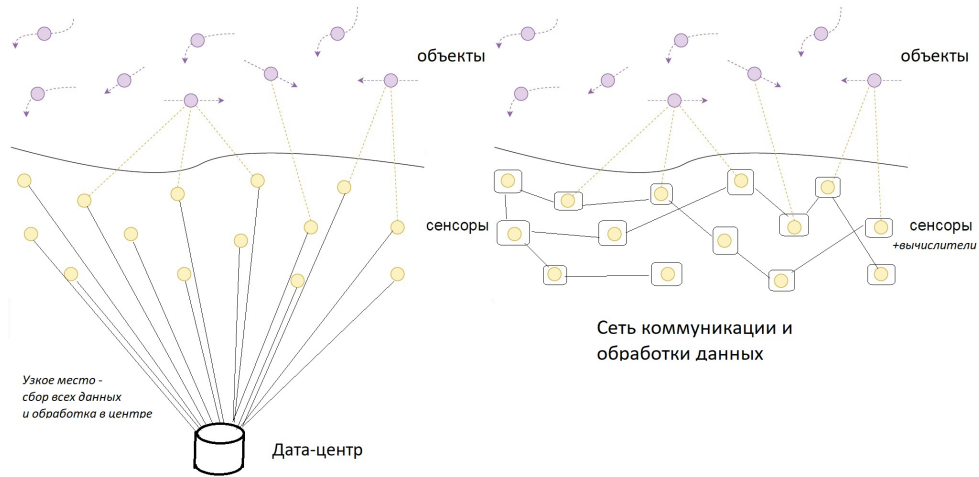


Рис. 1.3: Централизованный (слева) и распределенный (справа) способ решения задачи управления сенсорной сетью для отслеживания движущихся целей

ведет к ресурсозатратной обработке данных и не будет отличаться от централизованного подхода. Кроме того, на практике часто существуют технические ограничения на количество каналов связи (коммуникационные ограничения). С другой стороны, коммуницирование только с постоянным малым количеством соседних сенсоров может привести к потере важной информации обо всей системе в целом из-за ограниченности обзора сенсоров или даже к вырожденной задаче оценивания. В связи с этим является актуальной задачей оптимизация выбора соседей для каждого сенсора.

### 1.4.1 Оптимизация выбора целей и сети сенсоров

Одним из способов оптимизации распределения целей между сенсорами с учетом нагрузки на каждый сенсор и точности определения местоположения целей представлен в [102].

Введем  $\bar{\mathbf{r}}_t^{l,i}$  полученное измерение сенсором  $i$  местоположение цели  $l$ . Предположим, что такая оценка выглядит следующим образом:  $\bar{\mathbf{r}}_t^{l,i} = \mathbf{r}_t^l + \xi_t^{l,i}$ , где  $\mathbf{r}_t^l$  – истинное местоположение цели  $l$  в момент времени  $t$ , а  $\xi_t^{l,i}$  – неизвестная, но ограниченная помеха, которая может быть описана

эллипсоидом вокруг оценки  $\bar{\mathbf{r}}_t^{l,i}$ . В этом случае можно предполагать, что эллипсоид гарантированно содержит истинное значение  $\mathbf{r}_t^l$ . Это означает, что если существует ограничивающий эллипсоид, то ошибка оценивания  $\|\mathbf{r}_t^l - \bar{\mathbf{r}}_t^{l,i}\|$  не выходит за его границы и значение  $\mathbf{r}_t^l$  лежит внутри этого эллипсоида:

$$\mathcal{E}_t^{l,i} = \{\Xi_t^{l,i} x + \bar{\mathbf{r}}_t^{l,i} \mid \|x\| \leq 1\},$$

где  $\Xi_t^{l,i}$  – матрица формы эллипсоида, а  $\bar{\mathbf{r}}_t^{l,i}$  – центр эллипсоида. Если  $\Xi_t^{l,i}(\Xi_t^{l,i})^T > 0$ , то предыдущая запись эквивалентна:

$$\mathcal{E}_t^{l,i} = \{x : (x - \bar{\mathbf{r}}_t^{l,i})^T (\Xi_t^{l,i})^{-2} (x - \bar{\mathbf{r}}_t^{l,i}) \leq 1\}. \quad (1.6)$$

Поскольку рассматривается сенсорная сеть, то для каждой цели  $l$  имеется набор оценок  $\tilde{\mathbf{r}}_t^l = \{\bar{\mathbf{r}}_t^{l,1}, \dots, \bar{\mathbf{r}}_t^{l,n}\}$  и соответствующие им эллипсоиды  $\tilde{\mathcal{E}}_t^{l,i} = \{\mathcal{E}_t^{l,1}, \dots, \mathcal{E}_t^{l,n}\}$ . Так как эллипсоиды гарантированно содержат истинное значение  $\mathbf{r}_t^l$ , то возможно уменьшить неопределенность местоположения целей с помощью нахождения области пересечения эллипсоидов, полученных для разных сенсоров. Чем меньше будет эта область пересечения, тем точнее возможно будет определить истинный вектор  $\mathbf{r}_t^l$ . Однако стоит отметить, что найти объем такого пересечения может быть достаточно сложной задачей. Одно из возможных решений – аппроксимация области пересечения эллипсоидом [81].

Пусть  $\hat{\mathcal{E}}_t^l$  – эллипсоид, которые аппроксимирует пересечение эллипсоидов, полученных для цели  $l$ , а  $\bar{\mathcal{E}}_t = \{\hat{\mathcal{E}}_t^1, \dots, \hat{\mathcal{E}}_t^m\}$  – множество таких эллипсоидов, полученных для каждой цели. Тогда, необходимо найти такое множество  $\bar{\mathcal{E}}_t$ , чтобы сумма объемов эллипсоидов была минимальна:

$$\bar{\mathcal{E}}_t = \operatorname{argmin}_{\bar{\mathcal{E}}_t} \sum_{l \in \mathcal{M}} \operatorname{vol}(\hat{\mathcal{E}}_t^l).$$

Чтобы уменьшить нагрузку на обработку и связь, также необходимо минимизировать для каждой цели количество сенсоров, которые за ними

следят. Обозначим  $P_t$  матрицу распределения целей между сенсорами. Элементы этой матрицы указывают, следит ли сенсор  $i$  за целью  $l$  или нет. Каждой матрице  $P_t$  соответствует свое множество эллипсоидов  $\bar{\mathcal{E}}_t$ . Оптимизация распределения целей между сенсорами соответствует как можно более разреженной матрице  $P_t$ , так как чем меньше сенсоров следят за одной и той же целью, тем меньше затрат на коммуникацию и обработку измеренных данных. Типичным подходом, дающим разреженное решение, является регуляризация  $l_0$ - (квази) нормы. Так как оптимизация  $l_0$ - (квази) нормы – это невыпуклая и NP-сложная задача, в [91] предлагается использовать  $l_1$ -норму, которая дает наилучшую аппроксимацию разреженного решения. В итоге задача минимизации матрицы распределения целей между сенсорами сформулируется в следующем виде:

$$P_t = \underset{P_t}{\operatorname{argmin}} \left( \sum_{l \in \mathcal{M}} \operatorname{vol}(\hat{\mathcal{E}}_t^l) + \alpha \sum_{l \in \mathcal{M}} \|P^{(l,\cdot)}_t\|_1 + \beta \sum_{i \in \mathcal{N}} \|P_t^{(\cdot,i)}\|_1 \right), \quad (1.7)$$

где  $\alpha$  – параметр, регулирующий количество сенсоров, которые следят за одной целью,  $\beta$  – параметр, регулирующий количество целей, за которыми следит один сенсор,  $P^{(l,\cdot)}$  –  $l$ -ая строка матрицы  $P_t$ ,  $P_t^{(\cdot,i)}$  –  $j$ -ый столбец матрицы  $P_t$ ,  $\|\cdot\|_1$  –  $l_1$ -норма.

В [102] оптимизационная задача (1.7) была переписана в терминах линейных матричных неравенств и решена с использованием пакета CVX [66].

## 1.4.2 Рандомизация топологии

В предыдущем подразделе описан способ выбора целей и сети сенсоров. Недостатками такого подхода является долгое время расчета пакетом CVX, а также то, что при существенном движении целей или при выходе сенсоров из строя оптимизационную задачу распределения ресурсов приходится решать заново, что является достаточно трудоемким процессом. Альтернативой может служить использование рандомизации

топологии [19].

Предположим, что заданы следующие *топологические ограничения*: каждый сенсор  $i \in \mathcal{N}$  в каждый момент времени  $t$  способен измерять расстояние до одной цели и собирать данные только от  $p$  соседей, где  $p$  – заранее заданное значение. На практике из-за определенных ограничений количество возможных каналов связи обычно меньше или равно размерности пространства ( $p \leq d$ ).

Удовлетворить коммуникационные ограничения возможно при помощи рандомизации топологии по аналогии с протоколом сплетен [52]. Предлагается выбирать каналы связи между сенсорами случайным образом в каждый момент времени. Другими словами, необходимо *рандомизировать граф связи*.

Более формально, чтобы удовлетворить топологическим ограничениям с максимально допустимым количеством каналов  $p$  для каждого сенсора, топологический граф рандомизируется  $\mathcal{G}_A$  в каждый момент времени  $t$ . Для этого используется случайно выбранный подграф  $\mathcal{G}_{B_t} \subset \mathcal{G}_A$  с матрицей смежности  $B_t = [b_t^{i,j}]$ , строки которого содержат не более  $p$  ненулевых элементов, то есть граф  $\mathcal{G}_A$  описывает всевозможные связи между сенсорами, а граф  $\mathcal{G}_{B_t}$  – связи между сенсорами в конкретный момент времени  $t$ . Например, если матрица связности графа  $\mathcal{G}_A$  полная (все могут коммуницировать со всеми)

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

то при  $p = 1$  матрица связности графа  $\mathcal{G}_{B_t}$  в момент времени  $t = t_1$  может быть равна

$$B_{t_1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

а в момент времени  $t = t_2$  может быть равна

$$B_{t_2} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

На практике такая рандомизация означает, что каждый сенсор получает данные только от малого числа  $p$  других сенсоров, причем эти сенсоры постоянно меняются: в примере, описанном выше, в момент времени  $t = t_1$  первый сенсор получает данные от второго сенсора, а в момент времени  $t = t_2$  этот же сенсор получает данные от четвертого сенсора. Пример графов, описывающих сенсорную сеть, представлен на Рис. 1.4. Стоит отметить, что соседние агенты для коммуникации выбираются независимо и равномерно как в протоколе сплетен [52]. Фактически, топологический граф  $\mathcal{G}_A$  рандомизируется так же, как в [51].

Введем предположение на топологию такой рандомизированной сети.

**Предположение 3.** а) Для всех  $i \in \mathcal{N}, j \in \mathcal{N}_t^i$  веса  $b_t^{i,j}$  являются независимыми случайными величинами со средними значениями (математическими ожиданиями):  $\mathbb{E}b_t^{i,j} = b_{av}^{i,j}$ , ограниченными дисперсиями:  $\mathbb{E}\|\mathcal{L}(B_t) - \mathcal{L}(B_{av})\|^2 \leq \sigma_B^2$ , где  $B_{av} = [b_{av}^{i,j}]$  и ограниченными матрицами ковариации:  $\mathbb{E}[(\mathcal{L}(B_t) - \mathcal{L}(B_{av}))(\mathcal{L}(B_t) - \mathcal{L}(B_{av}))^T] \leq Q_B$ , а также  $b_{\max}$  – максимальный элемент  $Q_B$ .

в) Граф  $\mathcal{G}_{B_{av}}$  сильно связан.

В описанном подходе при рандомизации топологии выбор описанного



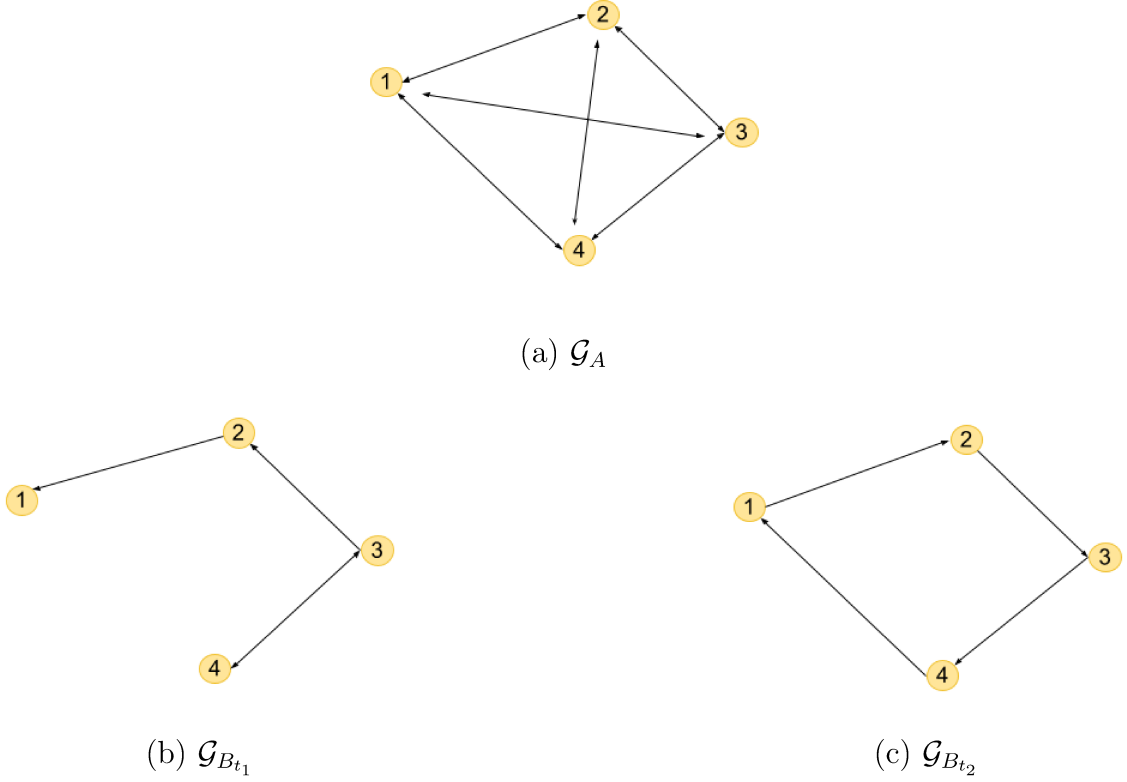


Рис. 1.4: Примеры топологий

ранее набора  $\mathbf{u} \in \mathcal{U}_t$  является случайным. Составим такой набор

$$\mathbf{u} = (i, J, l),$$

где  $i \in \mathcal{N}$ ,  $J \subset \bar{\mathcal{N}}_t^i$ ,  $l \in \mathcal{M}$ . Возвращаясь к примеру выше, один из таких наборов для первого сенсора может выглядеть следующим образом при  $t = t_1$ , если учесть, что первый и второй сенсоры согласованно наблюдают за пятой целью:  $\mathbf{u} = (1, 2, 5)$ .

Введем  $\text{Pr}_l(\mathbf{x})$  – вырезка (проекция) компонент вектора  $\mathbf{x}$ , соответствующая наблюдениям за целью с номером  $l$ .

Задача (1.4) отслеживания целей сетью сенсоров может быть переформулирована в виде следующей задачи стохастической оптимизации: в каждый момент времени  $t$  необходимо найти общую оценку местоположения всех целей  $\hat{\theta}_t = \begin{pmatrix} \hat{\mathbf{r}}_t^{1,1} \\ \vdots \\ \hat{\mathbf{r}}_t^{m,n} \end{pmatrix} \in \mathbb{R}^{nmd}$ , которая минимизирует функцию

потерь

$$\hat{\theta}_t = \operatorname{argmin}_{\hat{\theta}_t} \mathbb{E}_{\mathcal{F}_{t-1}} \sum_{i \in \mathcal{N}} F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \operatorname{argmin}_{\hat{\theta}_t} \sum_{i \in \mathcal{N}} \bar{F}_t^i(\hat{\theta}_t), \quad (1.8)$$

где

$$\bar{F}_t^i(\hat{\theta}_t) = \mathbb{E}_{\mathcal{F}_{t-1}} \{f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) \mid i(\mathbf{u}) = i\}$$

по зашумленным измерениям оптимизируемой функции

$$y_t^i = f_t^i(\mathbf{u}_t^i, \operatorname{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i)) + v_t^i,$$

где  $v_t^i$  – неизвестные, но ограниченные помехи. При равномерной выборке  $\mathbf{u}_t^i \in \mathcal{U}_t |_{i(\mathbf{u})=i}$  решение задачи (1.8) соответствует заявленной ранее цели (1.4).

Для решения такой задачи предполагается использование метода нулевого порядка. Этот класс методов подразумевает использование измерений оптимизируемой функции и не задействует вычисление градиента и производных более высоких порядков.

## 1.5 Выводы

В первой главе составлена сетевая модель наблюдения за движущимися объектами, основанная на рандомизированном подходе. Сформулированы предположения о скорости движения целей (Предположение 1), сенсорах, точках наблюдения и о помехах измерения (Предположение 2), а также о топологии рандомизированной сети (Предположение 3). Определена задача оптимизации для формирования в каждый момент времени состоятельных оценок местоположения всех целей с учетом зашумленных измерений оптимизируемой функции: в каждый момент времени  $t$  необходимо найти общую оценку местоположения всех

целей  $\hat{\theta}_t = \begin{pmatrix} \hat{\mathbf{r}}_t^{1,1} \\ \vdots \\ \hat{\mathbf{r}}_t^{m,1} \\ \hat{\mathbf{r}}_t^{1,2} \\ \vdots \\ \hat{\mathbf{r}}_t^{m,n} \end{pmatrix} \in \mathbb{R}^{nmd}$ , которая минимизирует функцию потерь

$$\hat{\theta}_t = \operatorname{argmin}_{\hat{\theta}_t} \sum_{i \in \mathcal{N}} \bar{F}_t^i(\hat{\theta}_t),$$

где

$$\bar{F}_t^i(\hat{\theta}_t) = \mathbb{E}_{\mathcal{F}_{t-1}} \{f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) \mid i = i(\mathbf{u})\}$$

по зашумленным измерениям оптимизируемой функции

$$y_t^i = f_t^i(\mathbf{u}_t^i, \operatorname{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i)) + v_t^i,$$

где  $v_t^i$  – неизвестные, но ограниченные помехи, причем  $\mathbb{E}(v_t^i)^2 \leq c_v^2$ , где  $c_v$  вычисляется с помощью констант из Леммы 1.

# Глава 2. Распределенный рандомизированный алгоритм стохастической аппроксимации, совмещенный с протоколом локального голосования

## 2.1 Оптимизация нестационарного функционала среднего риска

В работе [108] был предложен рандомизированный алгоритм стохастической оптимизации, совмещенный с алгоритмом консенсуса, для решения задачи оптимизации (1.8) для общего вида функции

$$f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i)).$$

Стоит отметить, что в работе [108] рассматривалась функция  $f_{\xi_t}^i(\mathbf{x}_t^{l,i})$ , где  $\{\xi\}, \xi_t \in \Xi$  – неконтролируемая детерминированная или случайная последовательность, а  $\Xi$  – множество. Функция  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$  из Раздела 1 является частным случаем функции  $f_{\xi_t}^i(\mathbf{x}_t^i)$ , при котором  $\xi_t$  состо-

ит из всех векторов  $\mathbf{u}_t^i$ , сгенерированных в момент времени  $t$ . Приведем алгоритм из [108].

Пусть  $\Delta_k^i$ ,  $k = 1, 2, \dots, i \in \mathcal{N}$  – наблюдаемые последовательности независимых случайных векторов из  $\mathbb{R}^{md}$  с распределением Бернулли, в котором каждый компонент независимо принимает значения  $\pm \frac{1}{\sqrt{d}}$  с вероятностью  $\frac{1}{2}$ . Такую последовательность обычно называют *одновременным пробным возмущением*. Также пусть  $\mathbf{K}_k^i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ ,  $k = 1, 2, \dots, i \in \mathcal{N}$  – множество векторных функций. Далее будем предполагать следующие свойства векторов  $\xi_k^i$  и  $\Delta_k^i$  (а также  $\mathbf{u}_k^i$  и  $\Delta_k^i$ ).

**Предположение 4.** Для любых  $i \in \mathcal{N}$  а) векторы  $\xi_k^i$ ,  $\Delta_k^i$ ,  $k = 1, 2, \dots, i \in \mathcal{N}$ , взаимно независимы; б)  $\xi_k^i$ ,  $\Delta_k^i$ ,  $\mathbf{s}_{2k-1}^i$ ,  $\mathbf{s}_{2k}^i$ , если они случайны, не зависят от  $\sigma$ -алгебры  $\mathcal{F}_{2k-2}$ ; в) если  $\tilde{v}_k^i$  случайны, то случайные векторы  $\xi_k^i$ ,  $\Delta_k^i$  и элементы  $\mathbf{s}_{2k-1}^i$ ,  $\mathbf{s}_{2k}^i$ ,  $\tilde{v}_k^i$  являются независимыми. г) векторы  $\mathbf{u}_k^i$ ,  $\Delta_k^i$ ,  $k = 1, 2, \dots, i \in \mathcal{N}$ , взаимно независимы; д)  $\mathbf{u}_k^i$ ,  $\Delta_k^i$ ,  $\mathbf{s}_{2k-1}^i$ ,  $\mathbf{s}_{2k}^i$ , если они случайны, не зависят от  $\sigma$ -алгебры  $\mathcal{F}_{2k-2}$ ; е) если  $\tilde{v}_k^i$  случайны, то случайные векторы  $\mathbf{u}_k^i$ ,  $\Delta_k^i$  и элементы  $\mathbf{s}_{2k-1}^i$ ,  $\mathbf{s}_{2k}^i$ ,  $\tilde{v}_k^i$  являются независимыми; ж)  $\mathbb{E}\|\Delta_k^i\|^2 \leq \sigma_\Delta^2$ ,  $\mathbb{E}[\Delta_k^i(\Delta_k^i)^T] \leq \sigma_\Delta^2 I_{md}$

Возьмем фиксированный неслучайный начальный вектор  $\hat{\theta}_0^i \in \mathbb{R}^{md}$ ,  $i \in \mathcal{N}$ , положительный размер шага  $\alpha$ , коэффициент усиления  $\gamma$  и выберем такие последовательности  $\{\beta_k^+\}$  и  $\{\beta_k^-\}$ , что  $\beta_k = \beta_k^+ + \beta_k^- > 0$ . Будем считать, что функция  $f_{\xi_k}^i(\mathbf{x}_k^i)$  доступна для наблюдения в каждый момент времени  $k$  с неизвестной, но ограниченной помехой  $v_k^i$ :  $y_k^i = f_{\xi_k}^i(\mathbf{x}_k^i) + v_k^i$ . Рассмотрим алгоритм с двумя наблюдениями распределенных подфункций  $f_{\xi_t}^i(\mathbf{x}_t^i)$  для каждого агента  $i \in \mathcal{N}$  для построения последовательностей точек наблюдения  $\{\mathbf{x}_t^i\}$  и оценок  $\{\hat{\theta}_t^i\}$  общего вектора состояния всех целей:

$$\begin{cases} \mathbf{x}_{2k}^i = \widehat{\theta}_{2k-2}^i + \beta_k^+ \Delta_k^i, \mathbf{x}_{2k-1}^i = \widehat{\theta}_{2k-2}^i - \beta_k^- \Delta_k^i, \\ \widehat{\theta}_{2k-1}^i = \widehat{\theta}_{2k-2}^i, \\ \widehat{\theta}_{2k}^i = \widehat{\theta}_{2k-1}^i - \alpha \left[ \frac{y_{2k}^i - y_{2k-1}^i}{\beta_k} \mathbf{K}(\Delta_k^i) + \right. \\ \left. \gamma \sum_{j \in \mathcal{N}_{2k-1}^i} b_{2k-1}^{i,j} (\widehat{\theta}_{2k-1}^i - \widehat{\theta}_{2k-1}^j) \right], \end{cases} \quad (2.9)$$

Рассмотрим последнее уравнение алгоритма (2.9): первая часть аналогична SPSA из [64], а вторая совпадает с протоколом локального голосования (LVP) из [38], где он был изучен для стохастических сетей в контексте проблемы балансировки нагрузки. Часть SPSA представляет собой стохастический градиентный спуск подфункций  $\bar{F}_t^i(\widehat{\theta}_t)$ , а часть LVP определяется для каждого агента  $i$  взвешенной суммой разностей между информацией о текущей оценке  $\widehat{\theta}_{2k-1}^i$  агента  $i$  и имеющейся информацией об оценках его соседей.

Основным результатом работы [108] является теорема о сходимости рандомизированного алгоритма стохастической оптимизации, совмещенного с алгоритмом консенсуса, к множеству возможных значений для общего вида функции  $f_{\xi_t}^i(\mathbf{x}_t^i)$ . Приведем предположения, введенные в [108], и, учитывая конкретный вид функции  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i)) = \|I_t^{\mathbf{u}_t^i} \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i) - H_t^{\mathbf{u}_t^i}\|^2$ , проверим выполнение условий теоремы для сетевой модели наблюдения за движущимися целями.

Условие 1 из [108]. Функции  $\bar{F}_t^i(\widehat{\theta}_t)$  выпуклые и у них есть общая точка минимума  $\theta_t$ , а также:

$$\forall \mathbf{x} \in \mathbb{R}^d \langle \mathbf{x} - \mathbf{r}_t^l, \mathbb{E}_{\mathcal{F}_{t-1}} \nabla f_{\xi_t}^i(\mathbf{x}) \rangle \geq 0.$$

Проверка выполнения Условия 1:

$$\begin{aligned} \langle \mathbf{x} - \mathbf{r}_t^l, \mathbb{E}_{\mathcal{F}_{t-1}} \nabla f_t^i(\mathbf{u}_t^i, \mathbf{x}) \rangle &= 2\mathbb{E}_{\mathcal{F}_{t-1}} (\mathbf{x} - \mathbf{r}_t^l)^\top (I_t^{\mathbf{u}_t^i})^\top (I_t^{\mathbf{u}_t^i} \mathbf{x} - H_t^{\mathbf{u}_t^i}) = \\ &= 2\mathbb{E}_{\mathcal{F}_{t-1}} \left( (I_t^{\mathbf{u}_t^i} (\mathbf{x} - \mathbf{r}_t^l))^\top (I_t^{\mathbf{u}_t^i} \mathbf{x} - H_t^{\mathbf{u}_t^i}) \right) = 2(I_t^{\mathbf{u}_t^i} \mathbf{x} - H_t^{\mathbf{u}_t^i})^\top (I_t^{\mathbf{u}_t^i} \mathbf{x} - H_t^{\mathbf{u}_t^i}) \geq 0. \end{aligned}$$

Условие 2 из [108]. Градиент  $\nabla f_{\xi_t}^i(\mathbf{x})$  непрерывен по Липшицу с одинаковой константой  $M > 0$ :

$$\forall \mathbf{x}', \mathbf{x}'' \in \mathbb{R}^d \quad \|\nabla f_{\xi_t}^i(\mathbf{x}') - \nabla f_{\xi_t}^i(\mathbf{x}'')\| \leq M \|\mathbf{x}' - \mathbf{x}''\|.$$

Проверка выполнения Условия 2:

$$\|\nabla f_t^i(\mathbf{u}_t^i, \mathbf{x}') - \nabla f_t^i(\mathbf{u}_t^i, \mathbf{x}'')\| = 2\|(I_t^{\mathbf{u}_t^i})^\top I_t^{\mathbf{u}_t^i}(\mathbf{x}' - \mathbf{x}'')\| \leq M \|\mathbf{x}' - \mathbf{x}''\|,$$

где  $M = 2 \max_i \|(I_t^{\mathbf{u}_t^i})^\top I_t^{\mathbf{u}_t^i}\|$ .

Условие 3 из [108]. Градиент  $\nabla f_{\xi_t}^i(\mathbf{x})$  равномерно ограничен в средне-квадратичном смысле в точке минимума  $\mathbf{r}_t^l$ :

$$\mathbb{E}\|\nabla f_{\xi_t}^i(\mathbf{r}_t^l)\|^2 \leq g_2^2,$$

$$\mathbb{E}\langle \nabla f_{\xi_t}^i(\mathbf{r}_t^l), \nabla f_{\xi_t}^i(\mathbf{r}_{t-1}^l) \rangle \leq g_2^2.$$

Проверка выполнения Условия 3: так как  $\nabla f_t^i(\mathbf{u}_t^i, \mathbf{r}_t^l) = 0$ , то  $g_2 = 0$ .

Условия 4-8 из [108] выполняются автоматически с учетом введенных ранее Предположений 1-4.

## 2.2 Распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, для оценки движущихся целей

Специфика задачи позволяет адаптировать (модернизировать) алгоритм (2.9) и переписать теорему о сходимости алгоритма к множеству возможных значений в терминах задачи, сформулированной в Разделе 1.

Более того, уточненные формулировки теоремы позволяют вывести субоптимальный шаг модернизированного алгоритма.

Рассмотрим модернизированный распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, для разработанной в Разделе 1 модели наблюдения за движущимися целями [46]. Необходимо заметить, что в отличие от работы [108] далее будет использоваться постоянный масштаб возмущения  $\beta > 0$  и  $\mathbf{K}_k^i(\Delta_k^i) = \Delta_k^i$ .

Выберем наблюдаемые последовательности независимых случайных векторов  $\Delta_k^i \in \mathbb{R}^{md}$  с распределением Бернулли, в котором каждый компонент независимо принимает значения  $\pm \frac{1}{\sqrt{d}}$  с вероятностью  $\frac{1}{2}$ , а также  $\mathbf{u}_k^i \in \mathcal{U}_t$  согласно рандомизированному графу топологии  $\mathcal{G}_{B_k}$ . Возьмем фиксированный неслучайный начальный вектор  $\hat{\theta}_0^i \in \mathbb{R}^{md}$ ,  $i \in \mathcal{N}$ , положительный размер шага  $\alpha$ , коэффициент усиления  $\gamma$  и масштаб возмущения  $\beta$ . Рассмотрим алгоритм с двумя наблюдениями распределенных подфункций  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$  для каждого сенсора  $i \in \mathcal{N}$  для построения последовательностей точек наблюдения  $\{\mathbf{x}_t^i\}$  и оценок  $\{\hat{\theta}_t^i\}$  общего вектора состояния всех целей:

$$\begin{cases} \mathbf{x}_{2k}^i = \hat{\theta}_{2k-2}^i + \beta \hat{\Delta}_k^i, & \mathbf{x}_{2k-1}^i = \hat{\theta}_{2k-2}^i - \beta \hat{\Delta}_k^i, \\ \hat{\theta}_{2k-1}^i = \hat{\theta}_{2k-2}^i, \\ \hat{\theta}_{2k}^i = \hat{\theta}_{2k-1}^i - \alpha \left[ \hat{\Delta}_k^i \frac{y_{2k}^i - y_{2k-1}^i}{2\beta} + \right. \\ \left. \gamma \sum_{j \in \mathcal{N}_t^i} b_t^{i,j} (\hat{\theta}_{2k-1}^i - \hat{\theta}_{2k-1}^j) \right]. \end{cases} \quad (2.10)$$

Введем следующие обозначения:

$$\bar{\mathbf{y}}_t = \begin{pmatrix} y_t^1 \\ \vdots \\ y_t^n \end{pmatrix}$$



$$\bar{\Delta}_{t\div 2} = \text{diag}_{nmd} \begin{pmatrix} \hat{\Delta}_{t\div 2}^1 \\ \vdots \\ \hat{\Delta}_{t\div 2}^n \end{pmatrix}.$$

Используя такие обозначения, алгоритм (2.10) может быть переписан в следующей форме:

$$\begin{aligned} \bar{\theta}_{2k} = \bar{\theta}_{2k-1} - \alpha[\bar{\Delta}_k \left( \frac{\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}}{2\beta} \otimes \mathbf{1}_{md} \right) + \\ \gamma(\mathcal{L}(B_{2k-1}) \otimes I_{md})\bar{\theta}_{2k-1}]. \end{aligned} \quad (2.11)$$

Для формулировки основного результата необходимо ввести следующие обозначения и определения констант относительно Предположений 1-4. Обозначим  $\bar{\mathcal{L}} = \mathcal{L}(B_{av})$ ,  $\bar{\lambda}_2 = \text{Re}(\lambda_2(\bar{\mathcal{L}}))$ ,  $\bar{\lambda}_{\max} = \lambda_{\max}^{\frac{1}{2}}(\bar{\mathcal{L}}^T \bar{\mathcal{L}})$ ,  $\mathbb{E} \text{diag}\{\dots, I_t^{u_t}, \dots\} = I_{av}$ ,

$$\Lambda_{\max} = \lambda_{\max}(I_{av}), \Lambda_{\min} = \begin{cases} 1/\lambda_{\min}(I_{av}) \text{ if } \lambda_{\min}(I_{av}) > 0, \\ 0 \text{ if } \lambda_{\min}(I_{av}) \leq 0, \end{cases},$$

$$c_1 = 2\gamma\bar{\lambda}_2 + 4\Lambda_{\min},$$

$$c_2 = \gamma^2(\bar{\lambda}_{\max}^2 + \sigma_B^2) + 4 + 4\gamma\bar{\lambda}_{\max}\Lambda_{\max} + \frac{2\bar{q}\delta^2\Lambda_{\max}^2}{\beta^2},$$

$$c_3 = \delta(2\gamma\sqrt{nm}\bar{\lambda}_{\max} + 6\sqrt{m}\Lambda_{\max} + \sqrt{\bar{q}}),$$

$$c_4 = \delta(\gamma\bar{\lambda}_{\max}(\sqrt{\bar{q}} + 2\sqrt{m}\Lambda_{\max}) + \frac{\bar{q}\delta^2}{\beta^2}(2\sqrt{m}\Lambda_{\max}^2 + \sqrt{\bar{q}}\Lambda_{\max}) + 2(2\sqrt{m}\Lambda_{\max}^2 + \sqrt{\bar{q}}\Lambda_{\max})),$$

$$c_5 = n(8m\delta^2\Lambda_{\max} + 4\sqrt{m}\delta^2\sqrt{\bar{q}}),$$

$$c_6 = n\left(\frac{c_v^2}{\beta^2} + \left(1 + \frac{\bar{q}\delta^4}{2\beta^2}\right)(4m\Lambda_{\max}^2 + 4\sqrt{m}\sqrt{\bar{q}} + \bar{q})\right).$$

**О п р е д е л е н и е 2.** Последовательность оценок  $\{\bar{\theta}_{2k}\}$  имеет асимптотически эффективную верхнюю границу  $\bar{L} > 0$  среднеквадратичной ошибки трекинга, если  $\forall \varepsilon > 0 \exists \bar{k}$  такое, что  $\forall k > \bar{k}$

$$\sqrt{\mathbb{E}\|\bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta_{2k}\|^2} \leq \bar{L} + \varepsilon.$$

Следующая теорема показывает асимптотически эффективную верх-

нюю границу среднеквадратичной ошибки трекинга, обеспечиваемую алгоритмом (2.10) [30].

**Т е о р е м а 1.** *Если Предположения 1–4 выполнены и положительная константа  $\alpha$  достаточно мала:  $\alpha < \min(c_1/c_2; 1/\mu)$ , тогда последовательность оценок алгоритма (2.10) имеет асимптотически эффективную верхнюю границу равную*

$$\bar{L} = \frac{1}{\mu} \left( h + \sqrt{h^2 + l\mu} \right), \quad (2.12)$$

где  $\mu = c_1 - \alpha c_2$ ,  $h = c_3 + \alpha c_4$ ,  $l = c_5 + \alpha c_6 + \frac{4nm\delta^2}{\alpha}$ .

*Доказательство.*

Доказательство во многом повторяет доказательство теоремы из [108].

Пусть  $\mathbf{d}_t^i = \hat{\theta}_{2\lceil \frac{t-1}{2} \rceil}^i - \theta_t$ ,  $\bar{\mathbf{d}}_t = \text{col}\{\mathbf{d}_t^1, \dots, \mathbf{d}_t^n\}$ , где  $\lceil \cdot \rceil$  – «потолочная» функция,  $\nu_k = \|\bar{\mathbf{d}}_{2k}\|$ ,  $\bar{\mathbf{s}}_k = \frac{\alpha}{2\beta}((\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}) \otimes \mathbf{I}_{md}) \bar{\Delta}_k$ ,  $\bar{\mathbf{v}}_k = \text{col}\{\tilde{v}_k^1, \dots, \tilde{v}_k^n\}$ ,  $\bar{\mathbf{u}}_{t \div 2} = \text{col}\{\mathbf{u}_{t \div 2}^1, \dots, \mathbf{u}_{t \div 2}^n\}$ .

Пусть  $\bar{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k, \bar{\Delta}_k\}$  – это  $\sigma$ -алгебра всех событий, сгенерированная  $\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k, \bar{\Delta}_k$ ,  $\tilde{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k\}$ , и  $\hat{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}\}$ :

$$\mathcal{F}_{k-1} \subset \hat{\mathcal{F}}_{k-1} \subset \tilde{\mathcal{F}}_{k-1} \subset \bar{\mathcal{F}}_{k-1} \subset \mathcal{F}_k.$$

Учитывая алгоритм (2.10), Предположение 3 и условное математическое ожидание по  $\sigma$ -алгебре  $\bar{\mathcal{F}}_{k-1}$ , получаем

$$\mathbb{E}_{\bar{\mathcal{F}}_{k-1}} \nu_k^2 = \|\bar{\mathbf{g}}_k\|^2 + \alpha^2 \gamma^2 \sigma_B^2 \nu_{k-1}^2 - 2\langle \bar{\mathbf{g}}_k, \mathbb{E}_{\bar{\mathcal{F}}_{k-1}} \bar{\mathbf{s}}_k \rangle + \mathbb{E}_{\bar{\mathcal{F}}_{k-1}} \|\bar{\mathbf{s}}_k\|^2, \quad (2.13)$$

где  $\bar{\mathbf{g}}_k = (I_{mnd} - \alpha\gamma\bar{\mathcal{L}} \otimes I_{md})\bar{\mathbf{d}}_{2k-2} + \mathbf{1}_n \otimes (\theta_{2k-2} - \theta_{2k})$ ,  $\bar{\mathcal{L}} = \mathcal{L}(B_{av})$ .

При выполнении Предположения 3  $\bar{\lambda}_2 > 0$  (см. [90]). Следовательно, для первого члена в (2.13) получаем

$$\|\bar{\mathbf{g}}_k\|^2 \leq (1 - 2\alpha\gamma\bar{\lambda}_2 + \alpha^2\gamma^2\bar{\lambda}_{\max}^2)\nu_{k-1}^2 + 4\alpha\gamma\sqrt{nm}\bar{\lambda}_{\max}\delta\nu_{k-1} + 4nm\delta^2.$$

В силу Предположений 1, 4 и условного математического ожидания над  $\sigma$ -алгеброй  $\tilde{\mathcal{F}}_{k-1}$ , а затем и над  $\sigma$ -алгеброй  $\hat{F}_{k-1}$ , можем оценить третий член в (2.13):

$$\begin{aligned} 2\langle \bar{\mathbf{g}}_k, \mathbb{E}_{\hat{\mathcal{F}}_{k-1}} \bar{\mathbf{s}}_k \rangle &\leq (-4\alpha\Lambda_{\min} + 4\alpha^2\gamma\bar{\lambda}_{\max}\Lambda_{\max})\nu_{k-1}^2 + \\ &(4\alpha\sqrt{m}\delta\Lambda_{\max} + 2\alpha\sqrt{\bar{q}}\delta + 2\alpha^2\gamma\bar{\lambda}_{\max}(\sqrt{\bar{q}}\delta + 2\sqrt{m}\delta\Lambda_{\max}) \\ &+ 8\sqrt{m}\alpha\delta\Lambda_{\max})\nu_{k-1} + 8m\alpha\delta^2\Lambda_{\max} + 4\sqrt{m}\alpha\delta^2\sqrt{\bar{q}}. \end{aligned}$$

Беря условное математическое ожидание над  $\sigma$ -алгеброй  $\mathcal{F}_{k-1}$  для последнего члена в (2.13) и учитывая Предположение 2, получаем

$$\begin{aligned} \mathbb{E}_{\mathcal{F}_{k-1}} \|\bar{\mathbf{s}}_k\|^2 &\leq \frac{\alpha^2}{4\beta^2} (2nc_v^2 + (2\bar{q}\delta^2 + 4\beta^2)(4\Lambda_{\max}^2\nu_{k-1}^2 + \\ &(8\sqrt{m}\delta\Lambda_{\max}^2 + 4\sqrt{\bar{q}}\delta\Lambda_{\max})\nu_{k-1} + n(4m\delta^2\Lambda_{\max}^2 + 4\sqrt{m}\delta\sqrt{\bar{q}}\delta + \bar{q}\delta^2))). \end{aligned}$$

Взяв условное математическое ожидание над  $\sigma$ -алгеброй  $\mathcal{F}_{k-1}$ , получаем следующее из (2.13)

$$\mathbb{E}_{\mathcal{F}_{k-1}} \nu_k^2 \leq (1 - \mu\alpha)\nu_{k-1}^2 + 2\alpha h\nu_{k-1} + \alpha l. \quad (2.14)$$

В силу условия Теоремы 1  $\alpha\mu < 1$ . Взяв безусловное математическое ожидание обеих частей (2.14), видим, что все условия Леммы из [64] выполняются для  $e_k = \sqrt{\mathbb{E}\nu_k^2}$ .  $\square$

**З а м е ч а н и е 2.** Помехи наблюдения  $v_t$  в Теореме 1 можно назвать почти произвольным, поскольку они могут быть как неслучайным, но ограниченным, так и реализацией некоторого случайного процесса с произвольными внутренними зависимостями. В частности, для доказательства результатов Теоремы 1 нет необходимости считать, что  $v_t$  и  $\mathcal{F}_{t-1}$  независимы.

**З а м е ч а н и е 3.** Следствие Теоремы 1 показывает, что для случая  $\delta = 0$  (все цели не меняют положение со временем)  $\bar{L} = \frac{c_v\sqrt{n\alpha}}{\beta\sqrt{2\mu}}$ . При

любом уровне помех  $c_\nu$  эту оценку можно сделать бесконечно малой путем выбора достаточно малых  $\alpha$ . В то же время, в случае движущихся целей большая норма изменений  $\delta$  может быть компенсирована выбором большего размера шага  $\alpha$ . Это приводит к компромиссу между уменьшением  $\alpha$  из-за зашумленных наблюдений и увеличением  $\alpha$  из-за дрейфа оптимальных точек.

**З а м е ч а н и е 4.** В случае  $\Lambda_{\min} = 0$  традиционные условия строгой выпуклости, обычно накладываемые при обосновании корректности таких алгоритмов, не выполняются.

**З а м е ч а н и е 5.** Алгоритм можно ускорить, добавив метод ускорения по Нестерову [11, 34, 60] или метод тяжелого шарика [59].

Уточненные формулировки Теоремы 1 позволяют вывести субоптимальный шаг модернизированного алгоритма (2.10) ([104]).

**Т е о р е м а 2.** При условии выполнения Предположений из Теоремы 1  $\alpha = \sqrt{\frac{C_2}{C_1}}$  – это субоптимальный шаг алгоритма (2.10), и соответствующее значение  $\hat{L}$  равно

$$\hat{L} = 2\sqrt{C_1 C_2} + C_3. \quad (2.15)$$

*Доказательство.*

Для фиксированных  $\beta$  и  $\gamma$  верхняя граница оценок (2.12) может быть оценена в следующем виде:

$$\bar{L}^2 \leq \tilde{L} = \frac{h}{\mu} \left( 2 + \frac{l\mu}{2h^2} \right) = 2\frac{h}{\mu} + \frac{l}{2h} = \hat{L} + \mathcal{O}(\alpha^2),$$

где  $\hat{L} = \alpha C_1 + \frac{C_2}{\alpha} + C_3$ ,  $C_1 = (c_1(c_1 c_3^3 c_6 - c_1 c_3 c_4 c_5 + 4nt\delta^2 c_1 c_4^2 + 4c_3^3 c_4) + 4c_2 c_3^4) / (2c_1^2 c_3^3)$ ,  $C_2 = (4nt\delta^2) / (2c_3)$ ,  $C_3 = (4c_3^3 + c_3 c_5 c_1 - 4nt\delta^2 c_4 c_1) / (2c_1 c_3)$ , и  $\mathcal{O}(\alpha^2)$  порядок малости, эквивалентный  $\alpha^2$ . Далее, находя частную про-

изводную по  $\alpha$ , получаем:

$$\frac{\partial \hat{L}}{\partial \alpha} = 0 \Rightarrow \alpha^* = \sqrt{\frac{C_2}{C_1}}.$$

□

## **2.3 Взвешенная версия распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования**

При практическом использовании того или иного алгоритма возможна ситуация, когда определенные наблюдения реализации случайной величины дают очень большие значения. Как правило, в реальных системах при больших значениях параметров малоизвестны модели динамики. В таких ситуациях модель часто определяют из соображений удобства использования. Однако нежелательно, чтобы одно большое по величине наблюдение существенно сказалось на вычисление текущей оценки. Для этого необходимо использовать робастные процедуры по аналогии с процедурами робастного статистического анализа [6]. Например, если несколько целей имеют отличный от других целей характер движения, целесообразно выделить их в отдельную группу. Для того, чтобы учитывать разный характер движения для разных групп целей, далее будет приведена взвешенная версия распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования. Более того, будет доказана теорема об асимптотически эффективной границе матрицы ковариации невязок, а не дисперсии невязок как в Разделе 2.2, что позволяет провести более детальный анализ сходимости алгоритма. Стоит отметить, что далее будет рассмат-

риваться только случай  $p = d$ , то есть когда количество каждого сенсора равно размерности пространства.

Определим диагональную матрицу  $\Psi = [\psi_{ij}]$ , где  $\psi_{ij} > 0$ , если  $i = j$ , и  $\psi_{ij} = 0$  в противном случае. Выберем наблюдаемые последовательности независимых случайных векторов  $\Delta_k^i \in \mathbb{R}^{md}$  с распределением Бернулли, в котором каждый компонент независимо принимает значения  $\pm \frac{1}{\sqrt{d}}$  с вероятностью  $\frac{1}{2}$ , а также  $\mathbf{u}_k^i \in \mathcal{U}_t$  согласно рандомизированному графу топологии  $\mathcal{G}_{B_k}$ . На этапе инициализации для каждого  $i \in \mathcal{N}$  выбираем начальный вектор  $\hat{\theta}_0^i \in \mathbb{R}^{md}$ , положительный размер шага  $\alpha$ , матрица  $\Psi$ , коэффициент усиления  $\gamma$  и масштаб возмущения  $\beta > 0$ .

Чтобы получить оценки  $\{\hat{\theta}_t^i\}$  общих векторов состояния  $\{\theta_t^i\}$  на основе точек измерения  $\{\mathbf{x}_t^i\}$  предлагается использовать взвешенный алгоритм с двумя измерениями распределенных подфункций  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$ :

$$\begin{cases} \mathbf{x}_{2k}^i = \hat{\theta}_{2k-2}^i + \beta \hat{\Delta}_k^i, & \mathbf{x}_{2k-1}^i = \hat{\theta}_{2k-2}^i - \beta \hat{\Delta}_k^i, \\ \hat{\theta}_{2k-1}^i = \hat{\theta}_{2k-2}^i, \\ \hat{\theta}_{2k}^i = \hat{\theta}_{2k-1}^i - \alpha \Psi \left[ \hat{\Delta}_k^i \frac{y_{2k}^i - y_{2k-1}^i}{2\beta} + \right. \\ \left. \gamma \sum_{j \in \bar{\mathcal{N}}_{2k-1}^i} b_{2k-1}^{i,j} (\hat{\theta}_{2k-1}^i - \hat{\theta}_{2k-1}^j) \right]. \end{cases} \quad (2.16)$$

Используя раннее введенные обозначения и  $\bar{\Psi} = I_n \otimes \Psi$ , алгоритм (2.16) может быть переписан в виде

$$\begin{aligned} \bar{\theta}_{2k} = \bar{\theta}_{2k-1} - \alpha \bar{\Psi} \left[ \bar{\Delta}_k \left( \frac{\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}}{2\beta} \otimes \mathbf{1}_{md} \right) + \right. \\ \left. \gamma (\mathcal{L}(B_{2k-1}) \otimes I_{md}) \bar{\theta}_{2k-1} \right]. \end{aligned} \quad (2.17)$$

Для формулировки основного результата необходимо ввести следующие обозначения и определения констант, отличных от введенных ранее в Разделе 2.2:

$$\begin{aligned} c_1 &= 1 - \varepsilon_1 - \alpha \sigma_{\Delta}^2 \varepsilon_2, \\ c_2 &= \alpha \frac{4\sigma_{\Delta}^4}{c_1} + 2\sigma_{\Delta}^2 \varepsilon_3, \end{aligned}$$

$$C_1 = \gamma \bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md}),$$

$$C_2 = \bar{\Psi} \bar{R}_k,$$

$$C_3 = \frac{1}{\beta} \sigma_{\Delta}^2 [(4\beta^2 \sigma_{\Delta}^2 + \delta^2)(4\nu_{k-1}^T R_k \nu_{k-1} - 12\delta e \nu_{k-1} + 9\delta^2) - 2c_v \delta (2e \nu_{k-1} - 3\delta) + c_v^2] I_{nmd},$$

$$C_4 = \sigma_{\Delta}^2 [(-\frac{1}{\varepsilon_1} + 1) \bar{\Psi}(\text{diag}_{nmd}(e)(9\mathbb{J}_n \otimes Q_{\delta}) \text{diag}_{nmd}(e) \bar{\Psi}^T + (-2\frac{1}{\varepsilon_2} + 1)(4\mathbb{J}_n \otimes Q_{\delta})],$$

$$C_5 = (1 - \frac{1}{\varepsilon_3}) 4\mathbb{J}_n \otimes Q_{\delta},$$

$$C_6 = C_1 + \frac{2\sigma_{\Delta}^2}{c_1} C_2.$$

**О п р е д е л е н и е 3.** Матрица ковариации невязок имеет асимптотически эффективную верхнюю границу  $\Sigma > 0$ , если  $\exists \bar{k}$  такое, что  $\forall k > \bar{k}$

$$\mathbb{E}[(\bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta_{2k})(\bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta_{2k})^T] \leq \Sigma + E_k,$$

где  $E_k \rightarrow 0$ .

Следующая теорема показывает асимптотически эффективную верхнюю границу ковариационной матрицы невязок, обеспечиваемую алгоритмом (2.16).

**Т е о р е м а 3.** Если Предположения 1-4 выполнены и  $\alpha > 0$ , тогда ковариационная матрица невязки, обеспечиваемая алгоритмом (2.16), имеет асимптотически эффективную верхнюю границу  $\Sigma$ , которая является решением следующего уравнения

$$\Sigma = c_1([I_{nmd} - \alpha C_6] \Sigma [I_{nmd} - \alpha C_6^T]) - \alpha c_2 C_2 \Sigma C_2^T + \alpha^2 \gamma^2 \bar{\Psi} \Sigma (Q_B \otimes I_{md}) \bar{\Psi}^T + \alpha^2 C_3 + \alpha C_4, \quad (2.18)$$

*Доказательство.*

Обозначим  $\mathbf{d}_t^i = \hat{\theta}_{2\lceil \frac{t-1}{2} \rceil}^i - \theta_t$ ,  $\bar{\mathbf{d}}_t = \text{col}\{\mathbf{d}_t^1, \dots, \mathbf{d}_t^n\}$ , где  $\lceil \cdot \rceil$  – потолочная функция,  $\nu_k = \bar{\mathbf{d}}_{2k}$ ,  $\Sigma_k = \mathbb{E}[\nu_k \nu_k^T]$ ,  $\bar{\mathbf{s}}_k = \frac{\alpha}{2\beta} \bar{\Delta}_k ((\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}) \otimes \mathbf{1}_{md})$ ,  $\bar{\mathbf{v}}_t = \text{col}\{\tilde{v}_t^1, \dots, \tilde{v}_t^n\}$ ,  $\bar{\mathbf{u}}_{t \div 2} = \text{col}\{\mathbf{u}_{t \div 2}^1, \dots, \mathbf{u}_{t \div 2}^n\}$ ,  $\bar{\Psi} = I_n \otimes \Psi$ .

Пусть  $\bar{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k, \bar{\Delta}_k\}$  –  $\sigma$ -алгебра всех событий, сгенерированных  $\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k, \bar{\Delta}_k, \bar{\mathcal{F}}_{k-1} =$

$\sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k\}$ , и  $\hat{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}\}$ :  $\mathcal{F}_{k-1} \subset \hat{\mathcal{F}}_{k-1} \subset \tilde{\mathcal{F}}_{k-1} \subset \bar{\mathcal{F}}_{k-1} \subset \mathcal{F}_k$ .

Учитывая, что  $\bar{\theta}_{2k-1} = \bar{\theta}_{2k-2}$  и  $\mathcal{L}(B_{2k-2})\mathbf{1}_n = 0$ , получаем

$$\nu_k = \bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta_{2k} = \bar{\mathbf{g}}_k - \bar{\Psi}\bar{\mathbf{s}}_k - \alpha\gamma\bar{\Psi}[(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]\nu_{k-1},$$

где  $\bar{\mathbf{g}}_k = [I_{nmd} - \alpha\gamma\bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md})]\nu_{k-1} + \mathbf{1}_n \otimes (\theta_{2k-2} - \theta_{2k})$ . Тогда

$$\begin{aligned} D_k = \nu_k \nu_k^T &= \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^T - \bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T - \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^T + \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T - \\ &\alpha\gamma(\bar{\mathbf{g}}_k - \bar{\Psi} \bar{\mathbf{s}}_k)[(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]^T \bar{\Psi}^T - \\ &\alpha\gamma\bar{\Psi}[(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]\nu_{k-1}(\bar{\mathbf{g}}_k^T - \bar{\mathbf{s}}_k^T \bar{\Psi}^T) + \\ &\alpha^2\gamma^2\bar{\Psi}[(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]D_{k-1}[(\mathcal{L}(B_{2k-2}) - \\ &\mathcal{L}(B_{av})) \otimes I_{md}]^T \bar{\Psi}^T. \end{aligned}$$

Взяв условное математическое ожидание  $\sigma$ -алгебра  $\bar{\mathcal{F}}_{k-1}$  и применив Предположение 3, получаем:

$$\begin{aligned} \mathbb{E}_{\bar{\mathcal{F}}_{k-1}}[D_k] &\leq \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^T - \bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T - \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^T + \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T + \\ &\alpha^2\gamma^2 b_{\max} \bar{\Psi} D_{k-1} \bar{\Psi}^T, \end{aligned} \quad (2.19)$$

где  $b_{\max}$  – максимальный элемент  $Q_B$ .

Далее, возьмем условное математическое ожидание над  $\sigma$ -алгеброй  $\tilde{\mathcal{F}}_{k-1}$ :

$$\begin{aligned} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[D_k] &\leq \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^T - \bar{\mathbf{g}}_k \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k^T] \bar{\Psi}^T - \bar{\Psi} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k] \bar{\mathbf{g}}_k^T + \\ &\bar{\Psi} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T] \bar{\Psi}^T + \alpha^2\gamma^2 b_{\max} \bar{\Psi} D_{k-1} \bar{\Psi}^T. \end{aligned} \quad (2.20)$$

Для первого слагаемого в (2.20), используя Предположение 1, можно



ВЫВЕСТИ:

$$\begin{aligned} \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^\top &\leq (1 - \varepsilon_3)([I_{nmd} - \alpha\gamma\bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md})]\Sigma_{k-1} \cdot \\ &\cdot [I_{nmd} - \alpha\gamma\bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md})]^\top) + (1 - \frac{1}{\varepsilon_3})4\mathbb{J}_n \otimes Q_\delta. \end{aligned}$$

Обозначим  $\mathbf{r}_t^i = \mathbf{e}_{l(\mathbf{u}_{t+2}^i)} \otimes [C_t^{\mathbf{u}_{t+2}^i}]^{-1} D_t^{\mathbf{u}_{t+2}^i}$ ,  $\hat{\mathbf{r}}_t^i = \text{diag}_{\mathfrak{g}_{md}}(\mathbf{e}_{l(\mathbf{u}_{t+2}^i)} \otimes I_d) \hat{\theta}_t^i$  и  $\Xi_t^i = \mathbf{r}_t^i - \mathbf{r}_{t-1}^i$ , тогда  $\forall i \in \{1, \dots, n\}$ :

$$\begin{aligned} y_{2k}^i - y_{2k-1}^i &= (2\beta\hat{\Delta}_k^i - \Xi_{2k}^i)^\top (2\text{diag}_{\mathfrak{g}_{md}}(\mathbf{e}_{l(\mathbf{u}_k^i)} \otimes I_d) \mathbf{d}_{2k-2}^i \\ &\quad - 2\Xi_{2k-1}^i - \Xi_{2k}^i) + v_{2k}^i - v_{2k-1}^i. \end{aligned}$$

При условии выполнения Предположения 4, домножая на  $\hat{\Delta}_k^i$  и взяв условное математическое ожидание над  $\sigma$ -алгеброй  $\tilde{\mathcal{F}}_{k-1}$ , получаем

$$\mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[(y_{2k}^i - y_{2k-1}^i) \hat{\Delta}_k^i] = 2\beta\sigma_\Delta^2 \mathbf{q}_{2k-2}^i,$$

где  $\mathbf{q}_{2k-2}^i = 2\text{diag}_{\mathfrak{g}_{md}}(\mathbf{e}_{l(\mathbf{u}_k^i)} \otimes I_d) \mathbf{d}_{2k-2}^i - 2\Xi_{2k-1}^i - \Xi_{2k}^i$ .

Пусть  $\sigma_\Delta = \frac{1}{\sqrt{d}}$ . Обозначим  $\bar{R}_t = \text{diag}_{nmd}(\text{col}\{\mathbf{e}_{l(\mathbf{u}_{t+2}^1)} \otimes I_d, \dots, \mathbf{e}_{l(\mathbf{u}_{t+2}^n)} \otimes I_d\})$ ,  $\bar{\Xi}_t = \text{col}\{\Xi_t^1, \dots, \Xi_t^n\}$ ,  $e = \text{col}\{e_{l(\mathbf{u}_t^1)}, \dots, e_{l(\mathbf{u}_t^n)}\}$ . Согласно Предположению 1, взяв условное математическое ожидание над  $\sigma$ -алгеброй  $\hat{\mathcal{F}}_{k-1}$  и используя  $\forall \varepsilon > 0 : AB^\top + BA^\top \leq \varepsilon AA^\top + \frac{1}{\varepsilon} BB^\top$ , получаем следующие выражения для второго и третьего слагаемого в (2.20):

$$\begin{aligned}
& -\bar{\mathbf{g}}_k \mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k^{\text{T}}] \bar{\Psi}^{\text{T}} - \bar{\Psi} \mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k] \bar{\mathbf{g}}_k^{\text{T}} \leq \\
& \alpha \sigma_{\Delta}^2 [-2(I_{nmd} - 2\alpha\gamma \bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md})) \Sigma_{k-1} \bar{R}_k \bar{\Psi}^{\text{T}} - \\
& 2\bar{\Psi} \bar{R}_k \Sigma_{k-1} (I_{nmd} - 2\alpha\gamma(\mathcal{L}(B_{av})^{\text{T}} \otimes I_{md}) \bar{\Psi}^{\text{T}}) - \\
& \varepsilon_1 (I_{nmd} - 2\alpha\gamma \bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md})) \Sigma_{k-1} (I_{nmd} - \\
& 2\alpha\gamma(\mathcal{L}(B_{av})^{\text{T}} \otimes I_{md}) \bar{\Psi}^{\text{T}}) + (-\frac{1}{\varepsilon_1} + 1) \bar{\Psi}(\text{diag}_{nmd}(e) \\
& (9\mathbb{J}_n \otimes Q_{\delta}) \text{diag}_{nmd}(e) \bar{\Psi}^{\text{T}} - \\
& 2\varepsilon_2 \bar{\Psi} \bar{R}_k \Sigma_{k-1}^{\text{T}} \bar{R}_k \bar{\Psi}^{\text{T}} + (-2\frac{1}{\varepsilon_2} + 1)(4\mathbb{J}_n \otimes Q_{\delta})].
\end{aligned}$$

Согласно Предположению 1 и 4, используя то, что  $\Delta_k^i$  получено из симметричного распределения, в качестве четвертого члена в (2.20) последовательно возьмем условное математическое ожидание над  $\sigma$ -алгеброй  $\tilde{\mathcal{F}}_{k-1}$ , а затем над  $\sigma$ -алгеброй  $\hat{\mathcal{F}}_{k-1}$ . В результате получаем

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^{\text{T}}] & \leq \frac{\alpha^2}{\beta} \sigma_{\Delta}^2 [(4\beta^2 \sigma_{\Delta}^2 + \delta^2)(4\nu_{k-1}^{\text{T}} R_k \nu_{k-1} - \\
& 12\delta e \nu_{k-1} + 9\delta^2) - 2c_v \delta (2e \nu_{k-1} - 3\delta) + c_v^2] I_{nmd}.
\end{aligned}$$

Складывая верхние границы и взяв безусловное математическое ожидание, получаем

$$\begin{aligned}
\Sigma_k & \leq c_1 ([I_{nmd} - \alpha C_6] \Sigma_{k-1} [I_{nmd} - \alpha C_6^{\text{T}}]) - \\
& \alpha c_2 C_2 \Sigma_{k-1} C_2^{\text{T}} + \alpha^2 \gamma^2 b_{\max} \bar{\Psi} \Sigma_{k-1} \bar{\Psi}^{\text{T}} + \alpha^2 C_3 + \alpha C_4 + C_5.
\end{aligned}$$

Введем замену:  $\Sigma_k = \Sigma + E_k$ , где  $\Sigma$  – решение уравнения (2.18).

Тогда,

$$\begin{aligned}
E_k & \leq c_1 ([I_{nmd} - \alpha C_6] E_{k-1} [I_{nmd} - \alpha C_6^{\text{T}}]) - \\
& \alpha c_2 C_2 E_{k-1} C_2^{\text{T}} + \alpha^2 \gamma^2 b_{\max} \bar{\Psi} E_{k-1} \bar{\Psi}^{\text{T}} + C_5.
\end{aligned}$$

Согласно [16]  $E_k \rightarrow 0$ . Следовательно, теорема доказана. □

## 2.4 Распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, для оценки положения статических объектов

В предыдущих разделах рассматривались случаи, в которых предполагалось, что цели и сенсоры могут менять свое местоположение с течением времени. В качестве области сходимости при исследовании алгоритмов с такой нестационарной постановкой задачи было получено множество возможных значений. Стоит заметить, что возможно построить алгоритм для оценки местоположения статических объектов, который сходится к точке как в [6] (оптимальный вектор состояния всех целей  $\theta$  не будет зависеть от времени, так как цели стационарны). Для этого модернизируем распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования. Основное его отличие от алгоритма (2.16) состоит в замене постоянного шага алгоритма  $\alpha$  на переменный шаг алгоритма  $\alpha_k$ .

Пусть есть диагональная матрица  $\Psi = [\psi_{ij}]$ , где  $\psi_{ij} > 0$ , если  $i = j$ , и  $\psi_{ij} = 0$  в других случаях. Выберем наблюдаемые последовательности независимых случайных векторов  $\Delta_k^i \in \mathbb{R}^{md}$  с распределением Бернулли, в котором каждый компонент независимо принимает значения  $\pm \frac{1}{\sqrt{d}}$  с вероятностью  $\frac{1}{2}$ , а также  $\mathbf{u}_k^i \in \mathcal{U}_t$  согласно рандомизированному графу топологии  $\mathcal{G}_{B_k}$ . Выберем для каждого  $i \in \mathcal{N}$   $\hat{\theta}_0^i \in \mathbb{R}^{md}$ , положительный шаг алгоритма  $\alpha_k$ , матрицу  $\Psi$ , коэффициент усиления  $\gamma$  и масштаб возмущения  $\beta > 0$ .

Чтобы получить оценки  $\{\widehat{\theta}_t^i\}$  общих векторов состояния  $\{\theta^i\}$  на основе точек измерения  $\{\mathbf{x}_t^i\}$  предлагается использовать взвешенный алгоритм с двумя измерениями распределенных подфункций  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$ :

$$\begin{cases} \mathbf{x}_{2k}^i = \widehat{\theta}_{2k-2}^i + \beta \widehat{\Delta}_k^i, & \mathbf{x}_{2k-1}^i = \widehat{\theta}_{2k-2}^i - \beta \widehat{\Delta}_k^i, \\ \widehat{\theta}_{2k-1}^i = \widehat{\theta}_{2k-2}^i, \\ \widehat{\theta}_{2k}^i = \widehat{\theta}_{2k-1}^i - \alpha_k \Psi \left[ \widehat{\Delta}_k^i \frac{y_{2k}^i - y_{2k-1}^i}{2\beta} + \right. \\ \left. \gamma \sum_{j \in \bar{\mathcal{N}}_{2k-1}^i} b_{2k-1}^{i,j} (\widehat{\theta}_{2k-1}^i - \widehat{\theta}_{2k-1}^j) \right]. \end{cases} \quad (2.21)$$

Используя обозначения, введенные ранее, алгоритм (2.21) может быть представлен в виде

$$\begin{aligned} \bar{\theta}_{2k} = \bar{\theta}_{2k-1} - \alpha_k \bar{\Psi} \left[ \bar{\Delta}_k \left( \frac{\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}}{2\beta} \otimes \mathbf{1}_{md} \right) + \right. \\ \left. \gamma (\mathcal{L}(B_{2k-1}) \otimes I_{md}) \bar{\theta}_{2k-1} \right]. \end{aligned} \quad (2.22)$$

Следующая теорема показывает асимптотически эффективную верхнюю границу ковариационной матрицы остатков, обеспечиваемую алгоритмом (2.21).

**Т е о р е м а 4.** *Если выполняются Предположения 2–4,  $\alpha_k = \frac{1}{k}$  и  $-(\gamma \bar{\lambda}_2 + \frac{2}{m}) \bar{\Psi} + \frac{1}{2} I_{nmd}$  устойчива (все ее собственные значения лежат в левой полуплоскости), тогда асимптотическая скорость сходимости алгоритма (2.21) характеризуется неравенством*

$$\mathbb{E}[(\widehat{\theta}_t - \theta)(\widehat{\theta}_t - \theta)^T] \leq \frac{1}{n} S + o\left(\frac{1}{n}\right), \quad (2.23)$$

в котором матрица  $S$  является решением матричного уравнения

$$\begin{aligned} S \left( \left( \gamma \bar{\lambda}_2 + \frac{2}{m} \right) \bar{\Psi}^T - \frac{1}{2} I_{nmd} \right) + \\ \left( \left( \gamma \bar{\lambda}_2 + \frac{2}{m} \right) \bar{\Psi} - \frac{1}{2} I_{nmd} \right) S = 4nc_v^2 \bar{\Psi} \bar{\Psi}^T. \end{aligned} \quad (2.24)$$

*Доказательство.*

Обозначим  $\mathbf{d}_t^i = \hat{\theta}_{2^{\lceil \frac{t-1}{2} \rceil}}^i - \theta$ ,  $\bar{\mathbf{d}}_t = \text{col}\{\mathbf{d}_t^1, \dots, \mathbf{d}_t^n\}$ , где  $[\cdot]$  – “потолочная функция”,  $\nu_k = \bar{\mathbf{d}}_{2k}$ ,  $D_k = \nu_k \nu_k^T$ ,  $\Sigma_k = \mathbb{E}[D_k]$ ,  $\bar{\mathbf{s}}_k = \frac{\alpha_k}{2\beta} \bar{\Delta}_k ((\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}) \otimes \mathbf{1}_{md})$ ,  $\bar{\mathbf{v}}_t = \text{col}\{\tilde{v}_t^1, \dots, \tilde{v}_t^n\}$ ,  $\bar{\mathbf{u}}_k = \text{col}\{\mathbf{u}_k^1, \dots, \mathbf{u}_k^n\}$ ,  $\bar{\Psi} = I_n \otimes \Psi$ .

Пусть  $\bar{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \bar{\mathbf{u}}_k, \bar{\Delta}_k\}$  –  $\sigma$ -алгебра вероятностных событий, сгенерированных  $\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \bar{\Delta}_k$ ,  $\tilde{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \bar{\mathbf{u}}_k\}$ , и  $\hat{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}\}$ :  $\mathcal{F}_{k-1} \subset \hat{\mathcal{F}}_{k-1} \subset \tilde{\mathcal{F}}_{k-1} \subset \bar{\mathcal{F}}_{k-1} \subset \mathcal{F}_k$ .

Используя  $\bar{\theta}_{2k-1} = \bar{\theta}_{2k-2}$  and  $\mathcal{L}(B_{2k-2})\mathbf{1}_n = 0$ , получим

$$\nu_k = \bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta = \bar{\mathbf{g}}_k - \bar{\Psi}\bar{\mathbf{s}}_k - \alpha_k \gamma \bar{\Psi}[(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]\nu_{k-1},$$

где  $\bar{\mathbf{g}}_k = [I_{nmd} - \alpha_k \gamma \bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md})]\nu_{k-1}$ . Тогда

$$\begin{aligned} D_k &= \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^T - \bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T - \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^T + \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T - \\ &\alpha_k \gamma (\bar{\mathbf{g}}_k - \bar{\Psi} \bar{\mathbf{s}}_k) \nu_{k-1}^T [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]^T \bar{\Psi}^T - \\ &\alpha_k \gamma \bar{\Psi} [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}] \nu_{k-1} (\bar{\mathbf{g}}_k^T - \bar{\mathbf{s}}_k^T \bar{\Psi}^T) + \\ &\alpha_k^2 \gamma^2 \bar{\Psi} [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}] D_{k-1} [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]^T \bar{\Psi}^T. \end{aligned}$$

Теперь возьмем условное математическое ожидание над  $\sigma$ -алгеброй  $\bar{\mathcal{F}}_{k-1}$  и применим Предположение 3:

$$\begin{aligned} \mathbb{E}_{\bar{\mathcal{F}}_{k-1}}[D_k] &\leq \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^T - \bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T - \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^T + \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T + \\ &\alpha_k^2 \gamma^2 b_{\max} \|D_{k-1}\|^2 \bar{\Psi} \bar{\Psi}^T, \end{aligned} \quad (2.25)$$

где  $b_{\max}$  – максимальный элемент  $Q_B$ .

Последовательно возьмем условное математическое ожидание над  $\sigma$ -алгеброй  $\tilde{\mathcal{F}}_{k-1}$ :

$$\begin{aligned} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[D_k] &\leq \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^T - \bar{\mathbf{g}}_k \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k^T] \bar{\Psi}^T - \bar{\Psi} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k] \bar{\mathbf{g}}_k^T + \\ &\bar{\Psi} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T] \bar{\Psi}^T + \alpha_k^2 \gamma^2 b_{\max} \|D_{k-1}\|^2 \bar{\Psi} \bar{\Psi}^T. \end{aligned} \quad (2.26)$$

При выполнении Предположения 3  $\bar{\lambda}_2 > 0$  (см. [90]). Следовательно, для первого члена в (2.26) получаем

$$\bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^T \leq D_{k-1} - \alpha_k \gamma \bar{\lambda}_2 (\bar{\Psi} D_{k-1} + D_{k-1} \bar{\Psi}^T) + \alpha_k^2 \gamma^2 \bar{\lambda}_{\max}^2 \|D_{k-1}\|^2 \bar{\Psi} \bar{\Psi}^T.$$

Обозначим  $\mathbf{r}^{l(\mathbf{u}_k^i)} = \mathbf{e}_{l(\mathbf{u}_k^i)} \otimes [C^{\mathbf{u}_k^i}]^{-1} D^{\mathbf{u}_k^i}$ ,  $\hat{\mathbf{r}}_t^{l(\mathbf{u}_k^i)} = \text{diag}_{md}(\mathbf{e}_{l(\mathbf{u}_k^i)} \otimes I_d) \hat{\theta}_t^i$ ,  $\tilde{v}_k^i = v_{2k}^i - v_{2k-1}^i$ , тогда  $\forall i \in \{1, \dots, n\}$ :

$$y_{2k}^i - y_{2k-1}^i = 4\beta (\hat{\Delta}_k^i)^T (\hat{\mathbf{r}}_{2k-2}^{l(\mathbf{u}_k^i)} - \mathbf{r}^{l(\mathbf{u}_k^i)}) + \tilde{v}_k^i.$$

По Предположению 4 получаем  $\mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\tilde{v}_k^i \hat{\Delta}_k^i] = 0$ . Обозначим

$$\bar{R}_t = \text{diag}_{nmd}(\text{col}\{\mathbf{e}_{l(\mathbf{u}_{t+2}^1)} \otimes I_d, \dots, \mathbf{e}_{l(\mathbf{u}_{t+2}^n)} \otimes I_d\}).$$

По Предположению 4, используя то, что  $\Delta_k^i$  взято из симметричного распределения, для четвертого члена в (2.26) получаем

$$\begin{aligned} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T] &\leq 4\alpha_k^2 \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\Delta}_k (\bar{\Delta}_k)^T \bar{R}_k D_{k-1} \bar{R}_k \bar{\Delta}_k^i (\bar{\Delta}_k)^T] + \\ &\quad \frac{\alpha_k^2}{4\beta^2} \sum_{i \in \mathcal{N}} (\tilde{v}_k^i)^2 \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\Delta}_k^i (\bar{\Delta}_k)^T]. \end{aligned}$$

Обозначим  $Z_k = \mathbb{E}[\bar{\Delta}_k (\bar{\Delta}_k)^T]$ . Суммируя второе и третье слагаемое из (2.26) и принимая условное математическое ожидание по  $\sigma$ -алгебре  $\hat{\mathcal{F}}_{k-1}$ , получаем следующий:

$$\begin{aligned} &-\mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^T] \bar{\Psi}^T - \bar{\Psi} \mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^T] \leq \\ &\quad -2 \frac{\alpha_k}{m} (D_{k-1} Z_k^T \bar{\Psi}^T + \bar{\Psi} Z_k D_{k-1}) + \\ &\quad 2\alpha_k^2 \gamma \frac{1}{m} \bar{\lambda}_{\max} \bar{\Psi} (D_{k-1} Z_k^T + Z_k D_{k-1}) \bar{\Psi}^T. \end{aligned}$$

Возьмем условное математическое ожидание по  $\sigma$ -алгебре  $\hat{\mathcal{F}}_{k-1}$ :

$$\begin{aligned} \mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T] &\leq 4 \frac{\alpha_k^2}{m^2} \|D_{k-1}\| \mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\|\bar{\Delta}_k\|^4] + \\ &\quad \frac{\alpha_k^2}{4\beta^2} \sum_{i \in \mathcal{N}} (\tilde{v}_k^i)^2 Z_k. \end{aligned}$$

Наконец, взяв условное математическое ожидание над  $\sigma$ -алгеброй  $\mathcal{F}_{k-1}$  в силу Предположения 2 для четвертого слагаемого в (2.26) получаем

$$\begin{aligned} \bar{\Psi} \mathbb{E}_{\mathcal{F}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T] \bar{\Psi}^T &\leq 4 \frac{\alpha_k^2}{m^2} \|D_{k-1}\| \bar{\Psi} \mathbb{E}_{\mathcal{F}_{k-1}}[\|\bar{\Delta}_k\|^4] \bar{\Psi}^T + \\ &\quad \frac{\alpha_k^2}{4\beta^2} n c_v^2 \bar{\Psi} Z_k. \end{aligned}$$

В итоге, взяв безусловное математическое ожидание и учитывая, что  $\|z_k^{-1} Z_k - I_{nmd}\| = \mathcal{O}(k^{-1})$ ,  $\alpha_k z_k = k^{-1}$  получаем следующее из (2.26)

$$\begin{aligned} \Sigma_k &\leq \Sigma_{k-1} - \left( \alpha_k \gamma \bar{\lambda}_2 + \frac{1}{k} \frac{2}{m} \right) (\Sigma_{k-1} \bar{\Psi}^T + \bar{\Psi} \Sigma_{k-1}) + \\ &\quad 4 \alpha_k^2 z_k n c_v^2 \bar{\Psi} \bar{\Psi}^T + \frac{1}{k} \kappa_k \mathcal{O}(\|\Sigma_{k-1}\|) + o(k^{-2}), \end{aligned}$$

где  $\{\kappa_k\} : \kappa_k \rightarrow 0$  if  $k \rightarrow \infty$ .

Пусть  $\alpha_k = \frac{1}{k}$ ,  $W_k = \frac{1}{\frac{1}{k} \gamma \bar{\lambda}_2 + \frac{1}{k} \frac{2}{m}} (\Sigma_k - \frac{1}{k} S)$ . Рассмотрим уравнение (2.24): если  $-(\gamma \bar{\lambda}_2 + \frac{2}{m}) \bar{\Psi} + \frac{1}{2} I_{nmd}$  устойчива, то существует положительно определенная матрица  $S$ , являющаяся решением этого уравнения Ляпунова.

Оценим  $W_k$  сверху:

$$\begin{aligned} W_k &\leq W_{k-1} - \frac{1}{k-1} W_{k-1} \left( (\gamma \bar{\lambda}_2 + \frac{2}{m}) \bar{\Psi}^T - \frac{1}{2} I_{nmd} \right) - \\ &\quad \frac{1}{k-1} \left( (\gamma \bar{\lambda}_2 + \frac{2}{m}) \bar{\Psi} - \frac{1}{2} I_{nmd} \right) W_{k-1} + \frac{1}{k} \kappa_k \mathcal{O}(\|W_{k-1}\|) + o(k^{-1}). \end{aligned}$$

Тогда по лемме 9 из [6]  $W_k \xrightarrow[k \rightarrow \infty]{} 0$ . Следовательно, теорема доказана.  $\square$

## 2.5 Выводы

Во второй главе показано, что благодаря специфике задачи распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, возможно модернизировать с учетом разработанной модели управления сенсорной сетью. Представлено три варианта модернизации такого алгоритма: алгоритм для оценки движущихся целей (2.10), взвешенная версия алгоритма для учета различного характера движения целей (2.16) и алгоритм для оценки местоположения статических объектов (2.21). Также исследованы условия применимости распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, из [108] для сетевой модели наблюдения. Для первого алгоритма доказана Теорема 1 о сходимости к множеству возможных значений. Эта теорема во многом повторяет теорему из [108], однако уточненные формулировки позволяют вывести субоптимальный шаг алгоритма (Теорема 2). Для второго алгоритма доказана Теорема 3 об асимптотической эффективной границе матрицы ковариаций невязок. Для третьего алгоритма доказана Теорема 4 об асимптотической скорости сходимости алгоритма.



# Глава 3. Отслеживание движения объектов при наличии неопределенностей и ограничений на количество связей между агентами

## 3.1 Численные эксперименты

В этом разделе рассматриваются численные эксперименты, иллюстрирующие работу предложенных алгоритмов (2.10), (2.16), (2.21).

Пусть есть распределенная сеть из 10-и планарных ( $d = 2$ ) интеллектуальных сенсоров (агентов), у которых есть возможность сбора данных только от одного своего соседа. Датчики имеют в зоне видимости 25 планарных целей, векторы состояния которых подлежат оценке модернизированным распределенным рандомизированным алгоритмом стохастической оптимизации, совмещенным с протоколом локального голосования, для оценки движущихся целей (2.10). В момент времени  $t$   $\mathbf{s}_t^i = [s_t^{i,1}, s_t^{i,2}]^T \in \mathbb{R}^2$  — вектор текущего состояния датчика  $i, i \in \mathcal{N} =$

$\{1, 2, \dots, 10\}$ ,  $\mathbf{r}_t^l = [r_t^{l,1}, r_t^{l,2}]^T \in \mathbb{R}^2$  — вектор состояния цели  $l, l \in \mathcal{M} = \{1, 2, \dots, 25\}$ ,  $\theta_t = \text{col}\{\mathbf{r}_t^1, \dots, \mathbf{r}_t^{25}\}$  — общий вектор состояния всех целей.

Рассмотрим три типа помех: равномерно распределенная случайная величина, попадающая в интервал  $[-1; 1]$ , неизвестная константа и гибридный шум, равномерно распределенный вокруг констант, изменяющихся во времени, например  $v_k^i = \pm 1 + 0.1 * \sin(k)$ , где знак перед 1 переключается каждую 50-ю итерацию. В моделировании, представленном далее, приведены оценки, характерные для каждого вида помех.

Цели  $l = 1, \dots, m$  начинают свое движение с позиции, состоящей из случайно выбранных координат из интервала  $[0; 120]$ . Динамика движения целей следующая:  $\mathbf{r}_t^l = \mathbf{r}_{t-1}^l + \chi_{t-1}^l$ . Пусть  $\chi_{t-1}^l$  — случайный вектор, равномерно распределенный на шаре радиусом 0.25. Наблюдатели не двигаются и их координаты представляют собой случайные величины, равномерно распределенные в интервале  $[200; 250]$ .

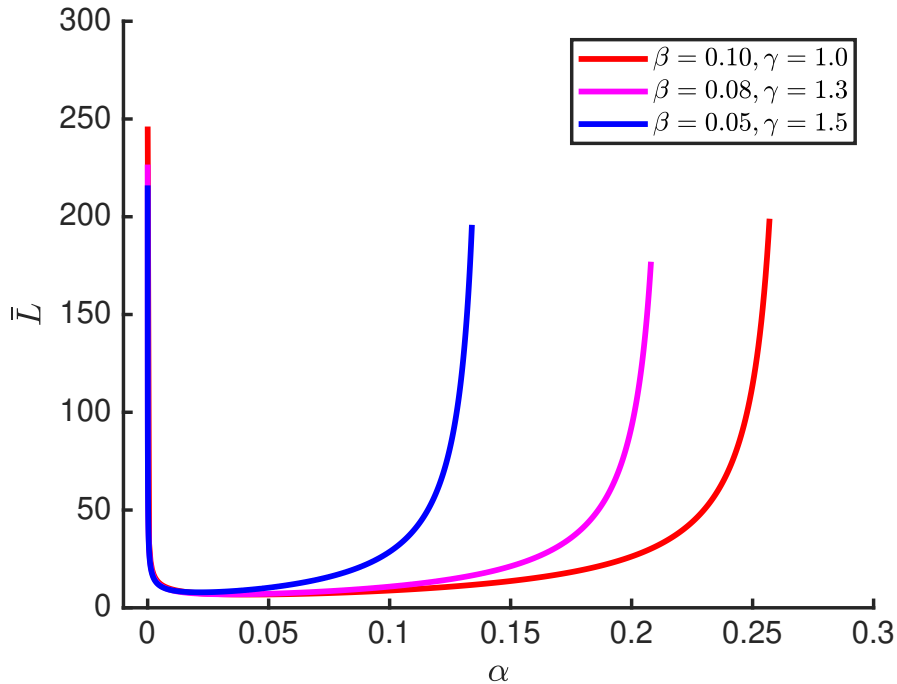


Рис. 3.5: Зависимость верхней границы среднеквадратичных ошибок от шага алгоритма  $\alpha$  при различных параметрах  $\beta$  и  $\gamma$

На Рис. 3.5 представлена зависимость верхней границы среднеквадратичных ошибок от разных параметров  $\beta$  и  $\gamma$ . Из графика видно, что

выбор  $\beta$  и  $\gamma$  практически не влияет на сходимость алгоритма при условии, что шаг алгоритма  $\alpha$  выбран оптимальным. Таким образом, имеет смысл проводить настройку только одного параметра алгоритма, а именно шага алгоритма  $\alpha$ , что и было реализовано в Теореме 2.

Размера шага  $\alpha = 0.05$  был выбран из Теоремы 2 при  $\beta = 0.1$  и  $\gamma = 1.0$ . Рис. 3.5 показывает, что эта  $\alpha$  является субоптимальным минимумом  $\bar{L}$ .

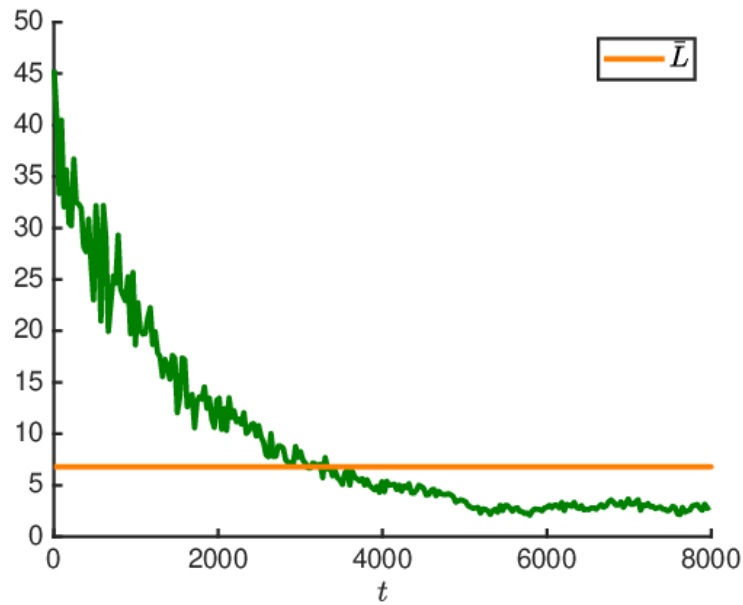


Рис. 3.6: Невязки  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$ , усредненные по сенсорам и целям, и верхняя граница  $\bar{L}$  среднего квадрата ошибки трекинга

На рис. 3.6 показано, как среднее значение наблюдений

$$f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$$

меняется с течением времени. Также показано, что существует момент времени  $t$ , начиная с которого оценки сходятся к действительному значению и колеблются рядом с ним. Кроме того, хорошо видно, что невязки находятся ниже рассчитанной согласно Теореме 1 верхней границы среднеквадратичной ошибки отслеживания  $\bar{L} = 6.8$ .

Далее рассмотрим численные результаты, полученные с помощью взвешенной версии распределенного рандомизированного алгоритма сто-

хастической оптимизации, совмещенного с протоколом локального голосования (2.16).

Пусть есть распределенная сеть из трех планарных ( $d = 2$ ) интеллектуальных сенсоров (агентов), у которых есть возможность сбора данных от двух своих соседей. В интересующей области имеется 6 движущихся целей. Сенсоры должны оценивать местоположение целей в режиме реального времени.

Также, как и в предыдущем моделировании, рассматривается три типа помех: равномерно распределенная случайная величина, попадающая в интервал  $[-1; 1]$ , неизвестная константа и гибридный шум, равномерно распределенный вокруг констант, изменяющихся во времени, например  $v_k^i = \pm 1 + 0.1 * \sin(k)$ , где знак перед 1 переключается каждую 50-ю итерацию. В моделировании, представленном далее, приводятся оценки, характерные для каждого типа помех.

Алгоритм (2.16), работающий на каждом узле (сенсоре), имеет следующие параметры:  $\alpha = 0.05$ ,  $\beta = 0.1$ ,  $\gamma = 1.0$ . Цели начинают свое движение в позиции, состоящей из случайно выбранных компонент из интервала  $[0; 100]$ . Динамика движения целей следующая:  $\mathbf{r}_t^l = \mathbf{r}_{t-1}^l + \chi_{t-1}^l$ . Пусть  $\chi_{t-1}^l$  – случайный вектор, равномерно распределенный по шару радиусом 0.2 для целей с нечетными номерами и 0.6 для целей с четными номерами. Это означает, что некоторые цели двигаются быстрее, чем другие цели. Пусть сенсоры неподвижны, а их координаты представляют собой случайные величины, равномерно распределенные в интервале  $[100; 120]$ .

Рассмотрим для каждой цели  $l$  и датчика  $i$  в каждый момент времени  $t$  ковариационную матрицу невязок  $\tilde{\Sigma}_t^{i,l} \in \mathbb{R}^{d \times d}$ , которая представлена как часть общей ковариационной матрицы. В этом моделировании алгоритм (2.16) сравнивается с алгоритмом (2.10). Чтобы проиллюстрировать, как матрица  $\Psi$  влияет на сходимость алгоритма, рассмотрим два

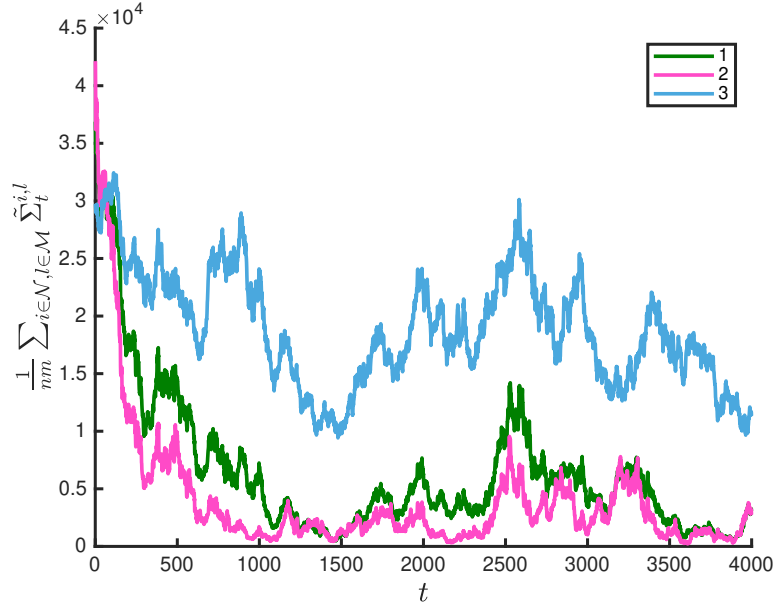


Рис. 3.7: Усредненное по всем сенсорам и целям значение первого элемента ковариационной матрицы невязок  $\tilde{\Sigma}_t^{i,l}$ , где (1) — предыдущий алгоритм (2.10) с  $\Psi = I_{md}$ , (2) и (3) — алгоритм (2.16), в котором  $\Psi$  равно  $\Psi^*$  и  $\Psi^{**}$  соответственно

разных случая:

$$\Psi^* = \text{diag}_{12}(\text{col}\{2, 2, 10, 10, 2, 2, \dots, 10, 10\}),$$

$$\Psi^{**} = \text{diag}_{12}(\text{col}\{0.2, 0.2, 100, 100, 0.2, 0.2, \dots, 100, 100\}).$$

На Рис. 3.7 показано, как изменяется усредненный первый диагональный элемент ковариационной матрицы невязок  $\tilde{\Sigma}_t^{i,l}$  через некоторое время. Хорошо видно, что новый алгоритм с  $\Psi = \Psi^*$  сходится быстрее, чем предыдущий алгоритм. Однако скорость сходимости зависит от  $\Psi$ .

Рассмотрим численный эксперимент, иллюстрирующий работу распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, для оценки местоположения статических объектов (2.21).

Пусть есть распределенная сеть, состоящая из пяти сенсоров. Пусть  $\mathcal{N} = \{1, 2, 3, 4, 5\}$  — множество датчиков. Каждый датчик имеет не бо-

лее двух активных каналов связи в каждый момент времени. В пределах интересующей области есть 10 стационарных целей. Задача сенсоров состоит в определении местоположения этих целей. В момент времени  $t$ ,  $\mathbf{s}^i = [s^{i,1}, s^{i,2}]^T \in \mathbb{R}^2$  – текущее состояние датчика  $i \in \mathcal{N}$ ,  $\mathbf{r}^l = [r^{l,1}, r^{l,2}]^T \in \mathbb{R}^2$  – состояние цели  $l \in \mathcal{M} = \{1, 2, \dots, 10\}$ ,  $\theta = \text{col}\{\mathbf{r}^1, \dots, \mathbf{r}^{10}\}$  – общее состояние всех целей.

В этом моделировании рассматривается только гибридный шум, равномерно распределенный вокруг констант, которые изменяются со временем, например,  $v_k^i = \pm 1 + 0.1 * \sin(k)$ , где знак перед 1 переключается каждую 50-ю итерацию.

Согласно Теореме 4 параметр размера шага  $\alpha$  должен быть равен  $\frac{1}{k}$ . Однако алгоритм (2.21), работающий на каждом узле с параметром  $\alpha_k = \frac{1}{k^{1-\rho}}$ ,  $\forall \rho > 0$ , имеет более последовательную сходимость. В этом моделировании были выбраны следующие параметры:  $\alpha_k = \frac{1}{k^{3/5}}$ ,  $\beta = 0.1$ ,  $\gamma = 1.0$ . Координаты целей и сенсоров представляют собой случайные величины, равномерно распределенные в интервалах  $[0; 100]$  и  $[100; 120]$  соответственно.

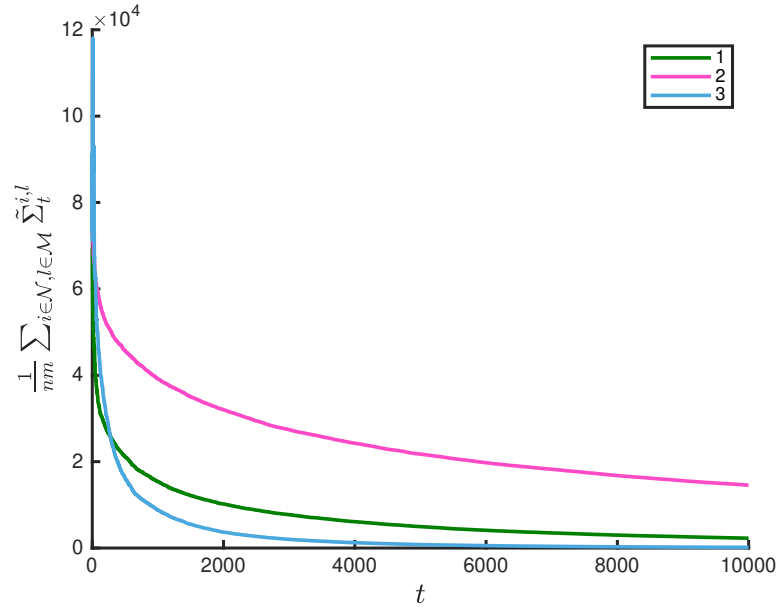


Рис. 3.8: Усредненное по всем целям и сенсорам значение первого элемента ковариационной матрицы остатков  $\tilde{\Sigma}_t^{i,l}$ , где (1):  $\Psi_1 = I_{md}$ , (2):  $\Psi_1 = 0.5I_{md}$ , (3):  $\Psi_3 = 2I_{md}$

Рассмотрим для каждой цели  $l$  и датчика  $i$  в каждый момент времени  $t$  ковариационную матрицу невязок  $\tilde{\Sigma}_t^{i,l} \in \mathbb{R}^{d \times d}$ , которая представлена как часть общей ковариационной матрицы. На Рис. 3.8 показано, как изменяется усредненный первый диагональный элемент ковариационной матрицы остатков  $\tilde{\Sigma}_t^{i,l}$  в зависимости от различных матриц  $\Psi$  через некоторое время. Из Рис. 3.8 видно, что новый алгоритм сходится к точке.

## 3.2 Прототип системы для отслеживания целей распределенной сетью сенсоров

### 3.2.1 Архитектура системы

Для реализации программного продукта, реализующего алгоритмы, описанные в прошлой главе, было разработано программное обеспечение<sup>1</sup> с использованием следующих технологий:

- язык программирования – MATLAB;
- инструмент для создания графического интерфейса – MATLAB App Designer;
- инструмент для создания независимого приложения – MATLAB Application Compiler.

Система представляет собой независимое пользовательское приложение, архитектура которого представлена на Рис. 3.9. Пользователь вводит параметры через графический интерфейс MATLAB App Designer.

---

<sup>1</sup>На прототип системы, реализующий алгоритм, получено свидетельство о государственной регистрации программы для ЭВМ №2022611354 “Прототип системы для отслеживания целей множеством сенсоров в условиях неизвестных, но ограниченных помех” от 24.01.2022 [21]

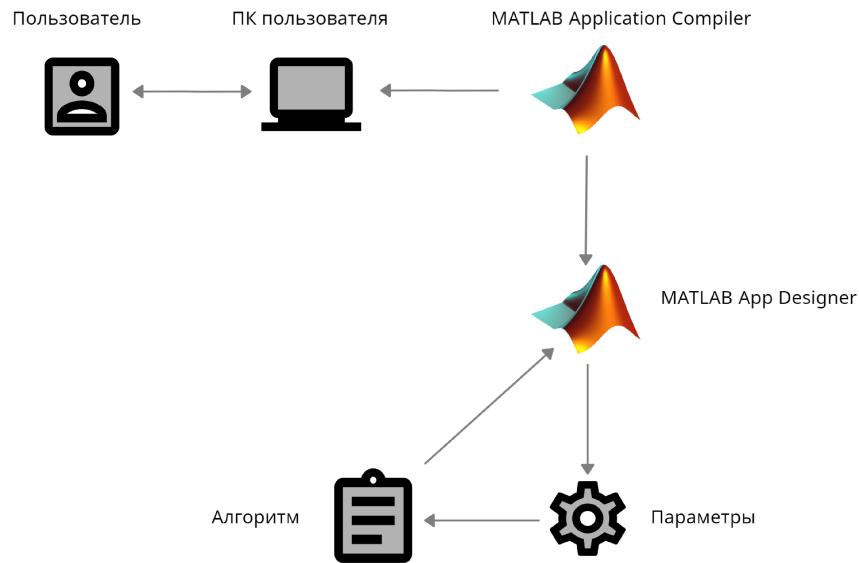


Рис. 3.9: Диаграмма компонентов архитектуры системы

Далее, сформированная модель подается на вход алгоритма. После выполнения вычислений MATLAB App Designer выдает результат работы алгоритма через графический интерфейс в виде графика зависимости дисперсии от количества итераций, графика зависимости элементов матрицы ковариации от количества итераций, графика, показывающего движение целей и сенсоров и графика, показывающего распределение помех. Использование MATLAB Application Compiler позволяет упаковать все необходимые библиотеки для создания независимого приложения, которое не требует установки MATLAB для запуска на персональном компьютере (ПК) конечного пользователя.

Общая структура работы в системе изображена на Рис. 3.10 в виде диаграммы активностей. Системе подаются на вход параметры, заданные в графическом интерфейсе. Далее проверяется, выбрано ли пользователем задание произвольной топологии сети сенсоров. Если да, то появляется интерфейс для выбора такой топологии, который в дальнейшем считывается. Если нет, то используется рандомизированная топология. После этого проверяется, выбрал ли пользователь вычисление оптимального параметра (шага алгоритма)  $\alpha$ . Если да, то производится поиск такого параметра. Если нет, то считывается введенное пользова-



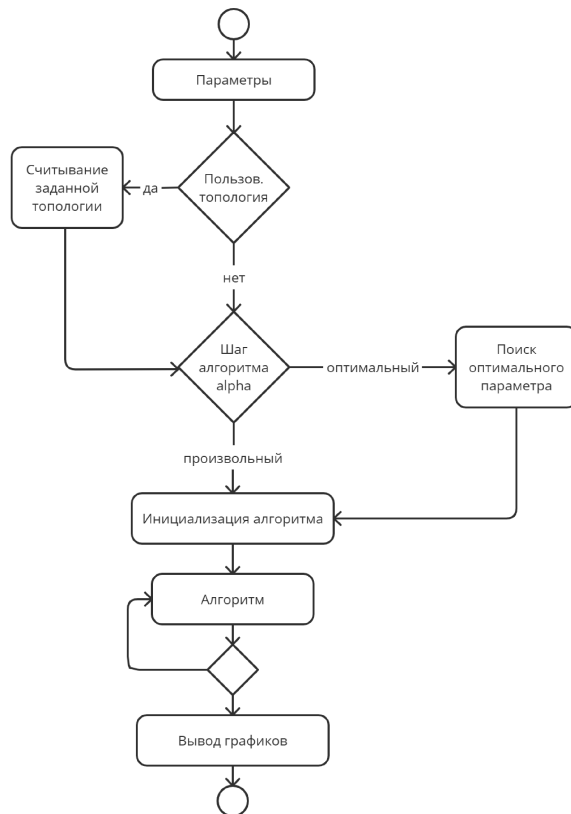


Рис. 3.10: Диаграмма активностей системы

телем значение. Далее происходит инициализация самого алгоритма и итеративное вычисление значения оценки точки минимума, которое повторяется заданное количество раз (число итераций). На каждой итерации вычисляется среднеквадратичная ошибка и матрица ковариации ошибок. После завершения цикла результат выводится в виде графиков среднеквадратичных ошибок оценок алгоритма и элементов матрицы ковариации ошибок.

### 3.2.2 Графический интерфейс

Внешний вид системы представлен на Рис. 3.11, 3.12, 3.13, 3.14. Графический интерфейс выполнен в виде четырех вкладок с общей фиксированной областью. Подробное описание реализации графического интерфейса системы приведено далее.

Общая область разделена по смыслу на четыре блока. В первом бло-

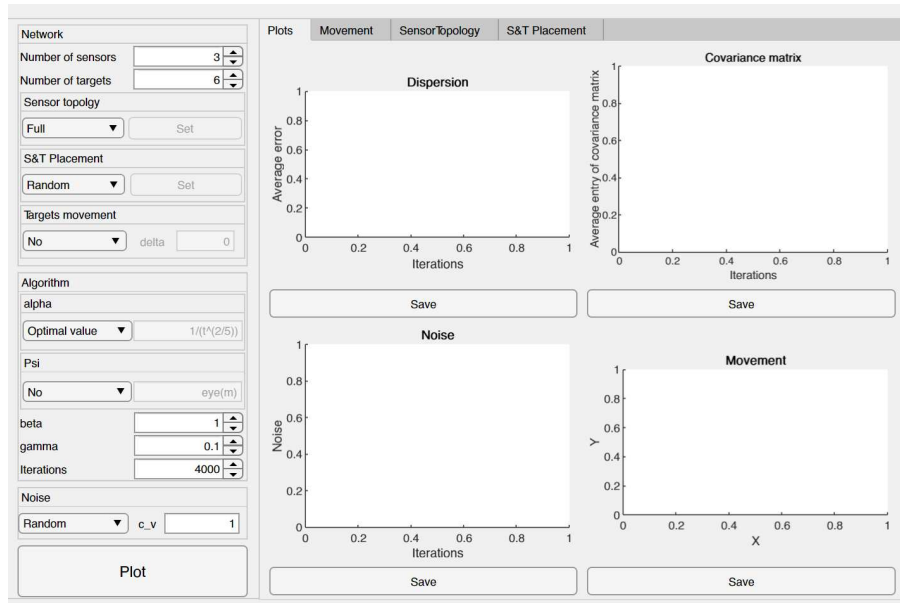


Рис. 3.11: Графический интерфейс системы. Изображение первой вкладки

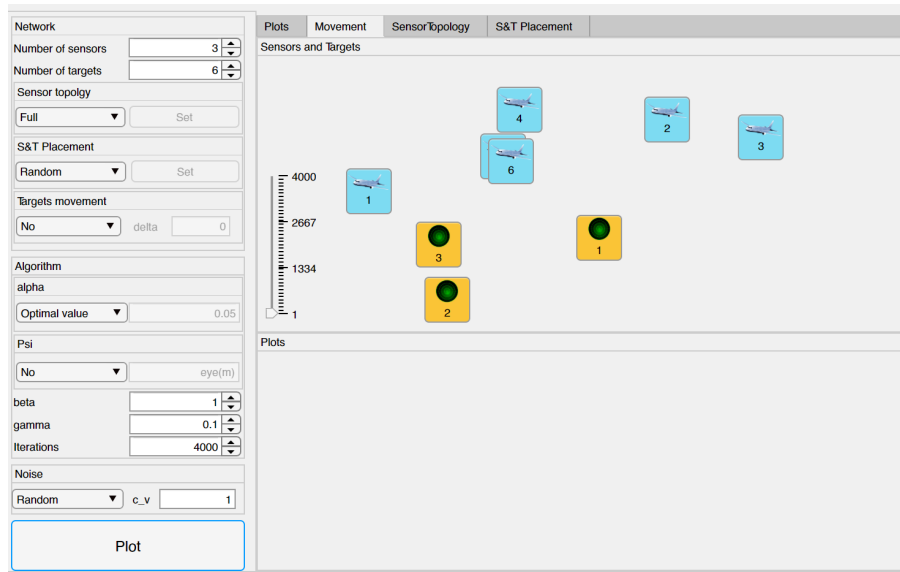


Рис. 3.12: Графический интерфейс системы. Изображение второй вкладки

ке “Network” пользователю предлагается задать количество сенсоров в системе (“Number of sensors”) и количество целей (“Number of targets”). Также в этом блоке возможно выбрать топологию сети сенсоров. Она может быть как полной (“Full”), так и пользовательской (“Custom”). Кнопка “Set”, расположенная в панели “Sensor topology”, становится активной при

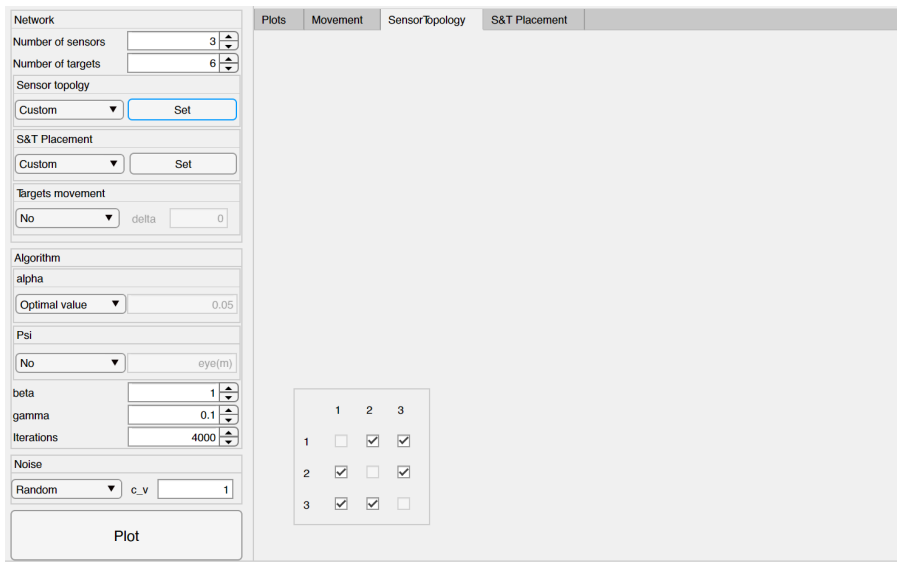


Рис. 3.13: Графический интерфейс системы. Изображение третьей вкладки

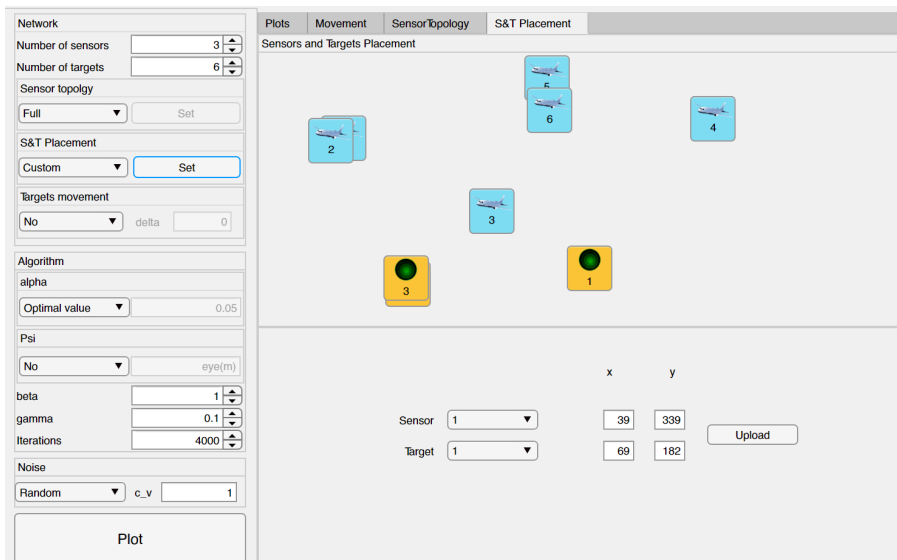


Рис. 3.14: Графический интерфейс системы. Изображение четвертой вкладки

выборе пользовательской топологии. При ее нажатии в третьей вкладке “Sensor Topology” (Рис. 3.13) появляется поле для задания топологии. Если  $\checkmark$  стоит на пересечении  $i$ -го столбца и  $j$ -й строки, то это означает, что сенсор  $i$  может получать данные от сенсора  $j$ . Стоит отметить, что количество соседей для каждого сенсора должно быть меньше, чем размерность пространства ( $d = 2$ ), но не может быть равно нулю. Та-

ким образом, если количество выбранных соседей становится нулевым, то система выдает ошибку. В первом блоке у пользователя также есть возможность задать модели движения целей: без перемещения (“No”); дрейфт с заданным ограничением  $\delta$  (“Drift”):  $r_t^l = r_{t-1}^l + \chi_{t-1}^l$ , где  $\chi_{t-1}^l$  – это случайный вектор, равномерно распределенный на шаре радиуса  $\delta$ ; произвольная функция, зависящая от количества целей  $m$  и момента времени  $t$  (“Custom”). Также в первом блоке пользователь может сам задать начальное расположение сенсоров и целей, либо выбрать ее случайной. Кнопка “Set”, расположенная в панели “S&T Placement”, становится активной при выборе первого варианта. После ее нажатия пользователь может перейти на четвертую вкладку “S&T Placement” (Рис. 3.14), где будет отрисован интерфейс текущего местоположения сенсоров и целей. Далее, выбирая номер сенсора и цели, вводя значения по осям OX и OY в соответствующие поля и нажимая на кнопку “Upload”, пользователь загружает данные в систему.

Параметры алгоритма задаются во втором блоке “Algorithm”. Значение параметра  $\alpha$  можно выбрать либо оптимальным (“Optimal”), либо произвольным (“Custom”), зависящим от момента времени  $t$ . При решении стационарной задачи оптимальное альфа рассчитывается как  $\alpha(t) = \frac{1}{t^{2/5}}$  [6, 47]. При решении нестационарной задачи  $\alpha$  вычисляется согласно Теореме 2. Рассчитанное значение помещается в соответствующее окно. Матрицу  $\Psi$  можно либо задать как единичную (“No”), либо ввести произвольно значение (“Custom”), зависящее от количества целей  $m$ . Пользователь также может задавать в этом блоке такие параметры, как  $\beta$ ,  $\gamma$  и количество итераций.

В третьем блоке производится задание помех. Есть возможность выбора трех видов помех: случайные (“Random”), произвольная константа (“Constant”), осцилляции (“Oscillations”). Также можно задавать пользовательский вид помех (“Custom”), зависящий от момента времени  $t$ . Помехи будут ограничены числом, введенным в поле “c\_v”.

В четвертом блоке расположена кнопка “Plot”, которая запускает вы-

полнение алгоритма с учетом заданных параметров.

На первой вкладке (Рис. 3.11) представлены графики результаты работы алгоритма. В верхнем левом углу находится график зависимости усредненной по всем сенсорам и целям среднеквадратичной ошибки от количества итераций, в верхнем правом углу – график зависимости элементов усредненной по всем сенсорам и целям матрицы ковариации от количества итераций, в нижнем левом углу – график зависимости помех от количества итераций, в нижнем правом углу – график, показывающий местоположение сенсоров (окружности), движение целей (точки), а также оценки местоположения целей различными сенсорами (плюсы). Каждый из вышеуказанных графиков можно сохранить в формате EPS (Encapsulated PostScript), нажав на кнопку “Save”.

Вторая вкладка (Рис. 3.12) состоит из двух частей. Верхняя часть предназначена для отрисовки местоположения сенсоров и целей. В качестве движущихся целей использовано изображение самолета, в качестве сенсоров – изображение радара. Слева находится ползунок, который принимает в качестве значения количество итераций. В системе возможно отследить движение целей (узнать их местоположение в конкретный момент времени), передвигая этот ползунок. Более того, в системе реализован следующий интерактивный режим: при нажатии на конкретную цель, в нижней части вкладки появляется график, показывающий зависимость среднеквадратичной ошибки от количества итераций для каждого сенсора. Аналогичный режим также реализован для сенсоров – при выборе определенного сенсора появляется график, показывающий зависимость среднеквадратичной ошибки от количества итераций для каждой цели.

### 3.2.3 Примеры работы системы

В этом подразделе показаны результаты работы системы на некоторых примерах.

## 1. Стационарная задача без учета помех.

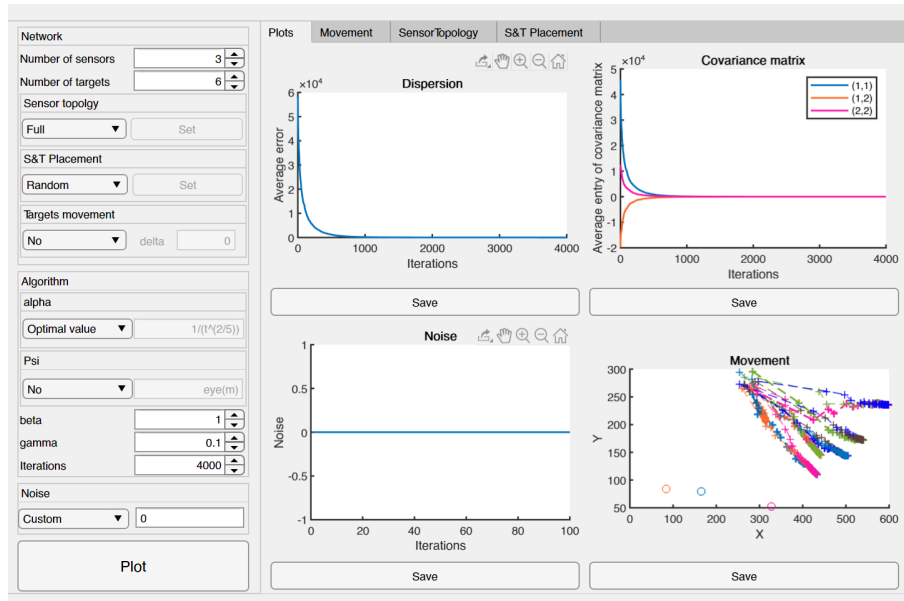


Рис. 3.15: Пример работы системы при решении стационарной задачи без учета помех. Изображение первой вкладки

В качестве первого примера на Рис. 3.15 представлен результат работы системы при отсутствии движения целей и каких-либо помех, полученных сенсорами при наблюдении за целями. В этом моделировании три сенсора оценивают местоположение шести стационарных целей. Топология сенсоров выбрана как полная, а местоположение сенсоров и целей выбирается случайно. Шаг алгоритма  $\alpha$  выбран как оптимальный и рассчитывается согласно Теореме 2. Другие параметры алгоритма заданы следующим образом:  $\beta = 1$ ,  $\gamma = 0.1$ , а количество итераций равно 4000. Так как это случай стационарной детерминированной оптимизации, из графика зависимости среднеквадратичной ошибки видно, что алгоритм сходится к нулю, то есть сенсоры точно определяют местоположение целей.

## 2. Стационарная задача с учетом помех в виде осцилляций.

На Рис. 3.16 представлен результат работы системы при решении стационарной задачи. В отличие от предыдущего примера, в этом случае сенсоры получают наблюдения о целях с помехами в виде периодиче-

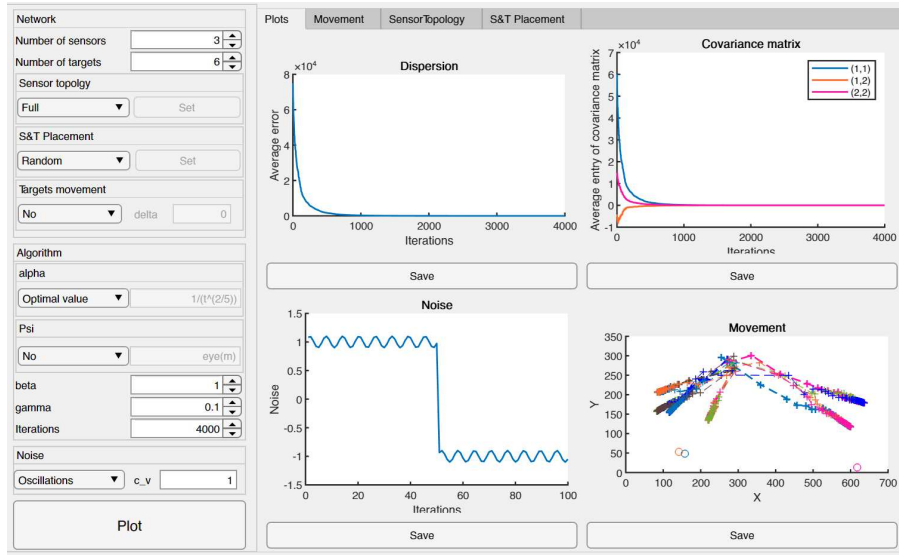


Рис. 3.16: Пример работы системы при решении стационарной задачи с учетом помех в виде осцилляций. Изображение первой вкладки

ских осцилляций, ограниченными по модулю  $c_v = 1$ . Распределение таких помех показано на нижнем левом графике на Рис. 3.16. Параметры алгоритма совпадали с параметрами из предыдущего примера. Исходя из графика зависимости среднеквадратичной ошибки также видно, что алгоритм сходится даже при условии таких помех.

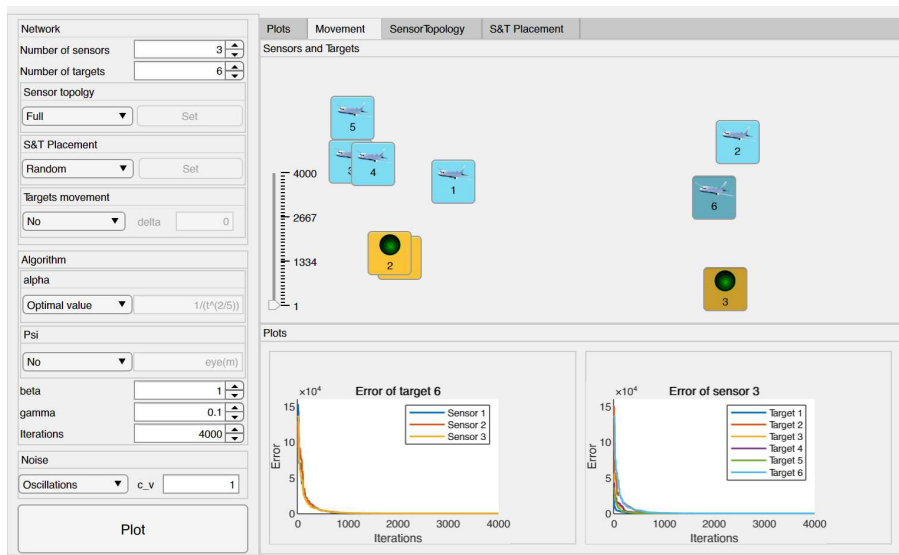


Рис. 3.17: Пример работы системы при решении стационарной задачи с учетом помех в виде осцилляций. Изображение второй вкладки

Внешний вид второй вкладки представлен на Рис. 3.17. На верхней

панели сенсоры и цели расположены согласно их настоящему местоположению, выбранному случайно. В этом примере нажаты кнопки, соответствующие третьему сенсору и шестой цели и отрисованы соответствующие графики среднеквадратичной ошибки для выбранных объектов. Как уже описывалось ранее, добавление такого интерактивного режима необходимо для последующего исследования алгоритма, а именно для того, чтобы понимать, какому сенсору лучше следить за какой целью.

3. Нестационарная задача с заданием пользовательской топологии сенсоров с учетом случайных помех.

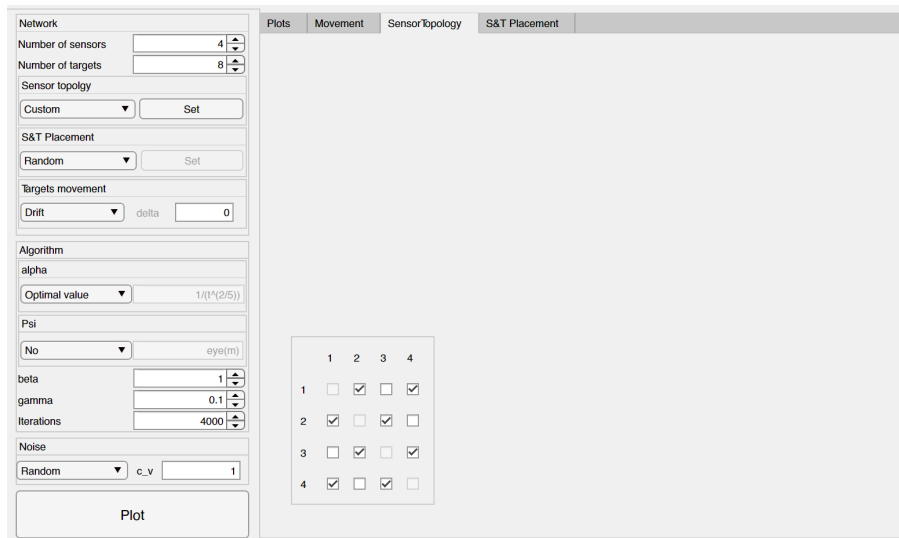


Рис. 3.18: Пример работы системы при решении нестационарной задачи с заданием пользовательской топологии сенсоров с учетом случайных помех. Изображение третьей вкладки

В этом примере четыре сенсора отслеживают движение восьми целей. Топология сети сенсоров задана согласно Рис. 3.18. Далее, в каждый момент времени сенсоры случайно выбирают согласно этой топологии не более чем одного соседа для общения. Модель движения целей задана в виде дрейфа с  $\delta = 2$ . Сенсоры получают наблюдения о местоположениях целей со случайными помехами. Из графиков, представленных на Рис. 3.19, видно, что алгоритм сходится.

4. Нестационарная задача с заданием первоначальных местоположений сенсоров и целей с учетом помех в виде константы.



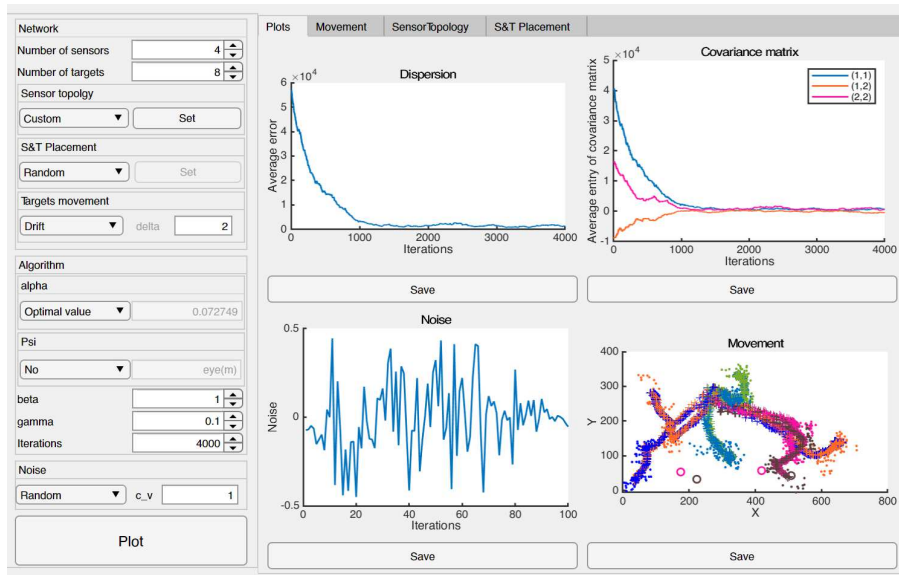


Рис. 3.19: Пример работы системы при решении нестационарной задачи с заданием пользовательской топологии сенсоров с учетом случайных помех. Изображение первой вкладки

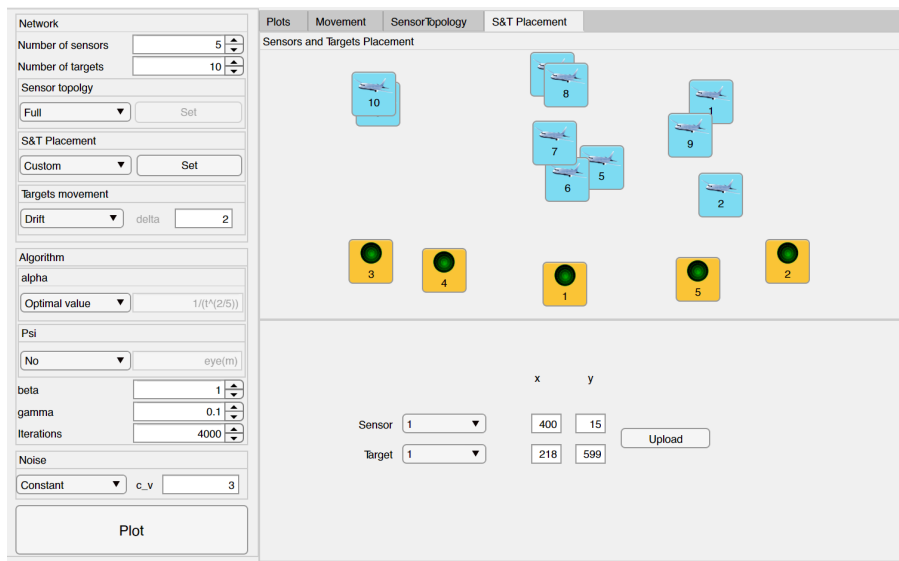


Рис. 3.20: Пример работы системы при решении нестационарной с заданием первоначальных местоположений сенсоров и целей с учетом помех в виде константы. Вид четвертой вкладки

В следующем примере пять сенсоров следят за десятью целями. Сенсоры получают наблюдения с помехами в виде случайной константы, ограниченной по модулю  $c_v = 3$ . Первоначальное местоположение сенсоров и целей задано с помощью графического интерфейса (Рис. 3.20).

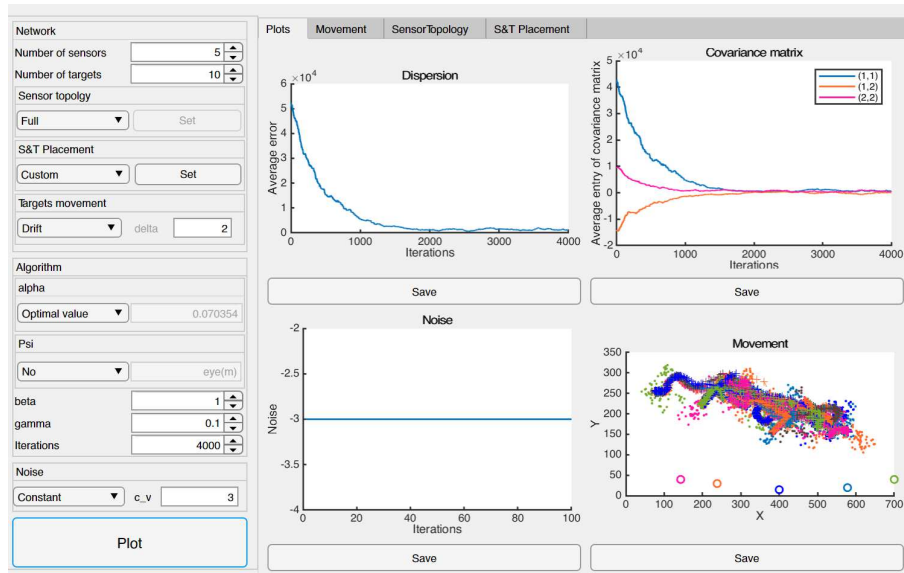


Рис. 3.21: Пример работы системы при решении нестационарной с заданием первоначальных местоположений сенсоров и целей с учетом помех в виде константы. Вид первой вкладки

Результаты работы системы представлены на Рис. 3.21. Как и во всех предыдущих случаях, графики подтверждают сходимость алгоритма. Необходимо отметить, что нижний правый график “Movement” перестает быть информативным, если количество целей и сенсоров больше четырех и восьми соответственно. Для более детального изучения движения целей и оценок их местоположения пользователю предлагается пользоваться второй вкладкой “Movement”.

5. Нестационарная задача с заданием произвольной функции движения целей и матрицы  $\Psi$  с учетом случайных помех.

В этом примере пять сенсоров наблюдают за девятью целями. Они получают наблюдения со случайными помехами, ограниченными по модулю  $c_v = 1$ . В [121] описано, что алгоритм с добавлением матрицы  $\Psi$  будет сходиться быстрее, если цели имеют разный характер движения. Для того, чтобы проверить это предположение, движение целей задано в виде функции `custom_target_sensor_movement(m, Time)` таким образом, чтобы первые четыре цели имели модель движения в виде дрефта с  $\delta = 3$ , а другие пять — с  $\delta = 5$ .

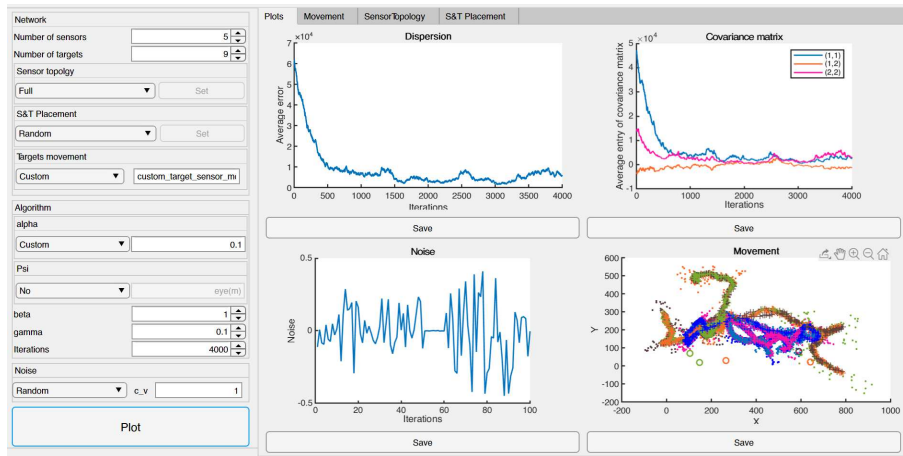


Рис. 3.22: Пример работы системы при решении нестационарной задачи с заданием произвольной функции движения целей и матрицы  $\Psi$ . В этом моделировании матрица  $\Psi$  — единичная матрица

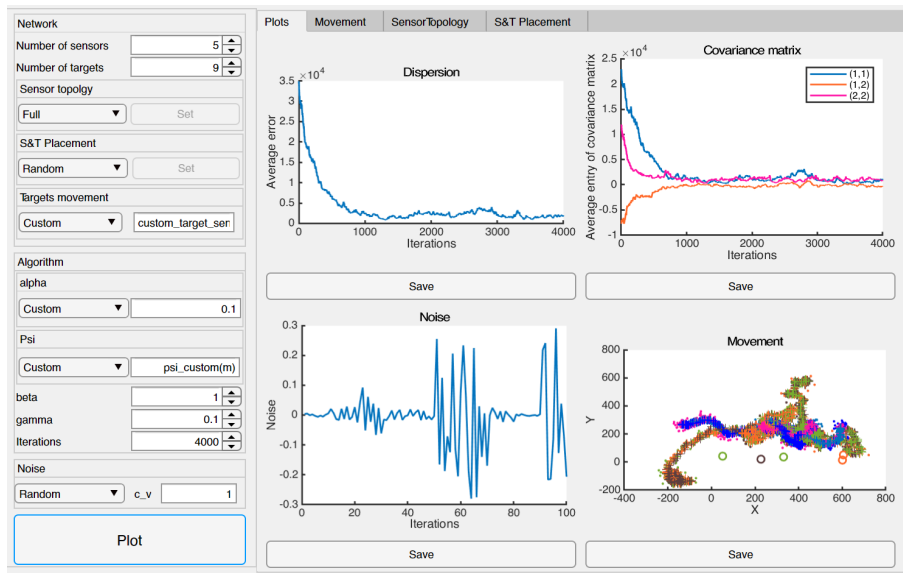


Рис. 3.23: Пример работы системы при решении нестационарной задачи с заданием произвольной функции движения целей и матрицы  $\Psi$

На Рис. 3.22 представлен результат работы системы с единичной матрицей  $\Psi$ , а на Рис. 3.23 с матрицей  $\Psi$ , заданной пользователем. Как видно из графиков среднеквадратичных ошибок и элементов матриц ковариаций, алгоритм с добавлением матрицы  $\Psi$  действительно имеет более быструю сходимость.

### **3.3 Практическое применение рандомизированного алгоритма стохастической аппроксимации, совмещенного с протоколом локального голосования**

Одним из приложений алгоритма является задача трекинга, а именно отслеживание движения летательных аппаратов. Для того, чтобы обеспечить безопасность воздушного движения, необходимо слежение за всеми летательными аппаратами. Еще некоторое время назад для наблюдения за самолетами использовались первичные радиолокаторы, которые могли получать только расстояние до фюзеляжа самолета. На сегодняшний день как основа наблюдения для управления воздушным движением применяются вторичные радиолокаторы. Их основная проблема — это высокая стоимость локатора и его позиции (капитального строительства), которая часто превосходит, а иногда и в разы, стоимость самого локатора. В связи с этим, расставить необходимое количество вторичных радиолокаторов для решения различных задач не представляется возможным. Также в связи с их большой массой такие радиолокаторы не являются мобильными.

В последнее время в российских аэропортах активно внедряются многопозиционные системы наблюдения (МПСН) [15]. МПСН представляет собой распределённую систему, которая использует для определения местоположения воздушных судов информацию о времени прихода сигналов от них на установленные в зоне действия приёмные станции системы [14]. Они имеют следующие основные преимущества [70]: зона действия МПСН конструируется сплошной, в отличие от вторичного радиолокатора, у которого зона действия имеет вид “воронки”; низкая стоимость отдельных позиций, что позволяет практически разворачивать системы, обеспечивающие наблюдения, в том числе на малых и предель-

но малых высотах; небольшой вес (5-8 кг). В связи с этим, такие системы теоретически являются мобильными.

Принимая во внимание непрерывный рост числа летательных аппаратов и тот объем данных, который каждый из них передает на обработку, к примеру, в случае с пассажирскими самолетами – это позывной воздушного судна, приборная скорость, число Маха и прочие, можно сделать вывод, что передача такого объема данных приводит к сильной загруженности каналов связи. Также, учитывая особенности рельефа местности, произвольные помехи и невозможность точного определения местонахождения летательного аппарата, некоторые станции передают в центр обработки заведомо ложную информацию, что приводит к необходимости начинать заново вычисления расположения этих летательных аппаратов.

Разработанная система для отслеживания целей сетью сенсоров с помощью распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, может существенно увеличить точность оценивания местоположения летательных аппаратов. В то же время, за счет использования распределенной сети сенсоров, количество передаваемых данных может значительно увеличиться. Все эти факторы в дальнейшем могут оказать заметное влияние на развитие слежения за летательными аппаратами [28].

Кроме того, в связи с тем, что в большинстве систем оценивания (идентификации) неизвестных параметров и стратегий управления сетевыми системами заложены те или иные методы оптимизации предложенный алгоритм возможно использовать для решения таких проблем.

### 3.4 Выводы

В третьей главе приведены результаты имитационного моделирования, иллюстрирующие работу версий модернизированного рандомизированного алгоритма стохастической аппроксимации, совмещенного с про-

токолом локального голосования. Также описан прототип системы для распределенного отслеживания, на который получено свидетельство о регистрации программы на ЭВМ, реализованный на языке программирования MATLAB. Основным приложением рандомизированного алгоритма стохастической аппроксимации, совмещенного с протоколом локального голосования, является отслеживание движения летательных аппаратов, однако алгоритм также может быть использован в идентификации неизвестных параметров, методах управления сетевыми системами и др.

## Заключение

Основные научные результаты диссертационного исследования, полученные в рамках выполнения поставленных задач:

- 1) разработана модель управления сенсорной сетью на основе рандомизированного и мультиагентного подходов (Глава 1);
- 2) модернизирован распределенный рандомизированный алгоритм стохастической оптимизации, совмещенный с протоколом локального голосования, и исследованы свойства его оценок для задачи трекинга (отслеживания изменения параметров) при использовании разработанной модели управления сенсорной сетью (Глава 2);
- 3) исследованы условия применимости модернизированного распределенного рандомизированного алгоритма стохастической оптимизации, совмещенного с протоколом локального голосования, для сетевой модели наблюдения. Результаты апробированы в задаче наблюдения за движущимися объектами при наличии неопределенностей и ограничений на количество связей между сенсорами (Главы 2, 3).

# Литература

- [1] Агаев Р.П., Чеботарев П.Ю. Матрица максимальных исходящих лесов орграфа и ее применения // Автоматика и телемеханика. — 2000. — № 9. — С. 15–43.
- [2] Андриевский Б. Р., Матвеев А. С., Фрадков А. Л. Управление и оценивание при информационных ограничениях: к единой теории управления, вычислений и связи // Автоматика и телемеханика. — 2010. — Т. 2010, № 4. — С. 34–99.
- [3] Городецкий В.И., Граничин О.Н., Скобелев П.О. Децентрализация, самоорганизация и эмерджентный интеллект – цифровой взрыв умных технологий // МКПУ-2022, С.-Петербург. — 2022.
- [4] Городецкий В.И., Скобелев П.О. Многоагентные технологии для промышленных приложений: реальность и перспектива // Труды СПИИРАН. — 2017. — Т. 55. — С. 11–45.
- [5] Граничин О.Н. Процедура стохастической аппроксимации с возмущением на входе // Автоматика и телемеханика. — 1992. — Т. 2. — С. 97–104.
- [6] Граничин О.Н., Поляк Б.Т. Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах. — Российская Федерация : Издательство “Наука”, 2003.
- [7] Граничин О.Н., Фомин В.Н. Адаптивное управление с использованием пробных сигналов в канале обратной связи // Автоматика и телемеханика. — 1986. — Т. 2. — С. 100–112.
- [8] Деревницкий Д.П. Фрадков А.Л. Две модели для анализа динамики алгоритмов адаптации // Автоматика и телемеханика. — 1974. — № 1. — С. 59–67.



- [9] Дерезицкий Д.П. Фрадков А.Л. Прикладная теория дискретных адаптивных систем управления. — М.: Наука, 1981. — С. 216.
- [10] Ермаков С.М. Метод Монте-Карло и смежные вопросы. — М.: Наука, 1971. — С. 471.
- [11] Ерофеева В.А., Сергеенко А.Н., Турсунова М.Б. Ускоренный рандомизированный алгоритм стохастической аппроксимации для задачи трекинга // Навигация и управление движением. Материалы XXIV конференции молодых ученых с международным участием. — 2022. — С. 143–145.
- [12] Иванский Ю.В., Сергеенко А.Н. Распределенные алгоритмы отслеживания траекторий множества объектов массивом мобильных сенсоров // Навигация и управление движением. сборник тезисов докладов Международного семинара. Под общей редакцией И.В. Белоконова. — 2020. — С. 22.
- [13] Карпов М.И., Петросян Л.А. Кооперативные решения в коммуникационных сетях // Вестник Санкт-Петербургского Университета. Серия 10: Прикладная математика, информатика, процессы управления. — 2012. — С. 37–45.
- [14] Международная организация гражданской авиации. — Руководство по авиационному наблюдению, издание второе, 2017.
- [15] Научно-производственное предприятие «Цифровые радиотехнические системы». — URL: npp-crts.ru.
- [16] Поляк Б.Т. Сходимость и скорость сходимости итеративных стохастических алгоритмов. II. Линейный случай // Автомат. и телемех. — 1977. — Т. 38, № 4. — С. 537–542.
- [17] Поляк Б.Т., Цыбаков А.Б. Оптимальные порядки точности поисковых алгоритмов стохастической оптимизации // Проблемы передачи информации. — 1990. — Т. 26, № 2. — С. 45–53.
- [18] Протокол кооперативной самоорганизации группового поведения роботов / К.С. Амелин, В.А. Ерофеева, О.Н. Граничин и др. // XIV Всероссийская мультиконференция по проблемам управления мкпу-2021. Материалы XIV мультиконференции. — 2021. — С. 94–96.

- [19] Рандомизация связей в мультиагентных системах для удовлетворения стоимостных ограничений / В.А. Ерофеева, О.Н. Граничин, Ю.В. Иванский, А.Н. Сергеевко // XIV Всероссийская мультиконференция по проблемам управления мкпу-2021. Материалы XIV мультиконференции. — 2021. — С. 48–50.
- [20] Растрингин Л.А. Статистические методы поиска. — М.: Наука, 1968. — С. 376.
- [21] Сергеевко А.Н. Прототип системы для отслеживания целей множеством сенсоров в условиях неизвестных, но ограниченных помех. — Свидетельство о регистрации программы для ЭВМ №2022611354 от 24.01.22.
- [22] Сергеевко А.Н. Задача о коммивояжере: сравнение трудоемкости метода перебора с алгоритмом, основанным на ДНК-вычислениях // Неделя науки СПбПУ. Материалы научной конференции с международным участием. Институт физики, нанотехнологий и телекоммуникаций. — 2018. — С. 376–378.
- [23] Сергеевко А.Н. Сравнение трудоемкостей метода перебора и алгоритма, основанного на ДНК-вычислениях, на примере решения задачи о коммивояжере // Неделя науки СПбПУ. Материалы научной конференции с международным участием. Лучшие доклады. — 2018. — С. 167–169.
- [24] Сергеевко А.Н. Применение алгоритма, основанного на дезоксирибонуклеиновых вычислениях, для решения оптимизационной задачи // Навигация и управление движением. Материалы XXI конференции молодых ученых с международным участием. Под общей редакцией В. Г. Пешехонова. — 2019. — С. 274–276.
- [25] Сергеевко А.Н. Алгоритм отслеживания распределенной сетью мобильных сенсоров траекторий множества объектов // Математическая теория управления и ее приложения (МТУиП-2020). Материалы конференции. Государственный научный центр Российской Федерации АО «Концерн «ЦНИИ «Электроприбор». — 2020. — С. 130–132.
- [26] Сергеевко А.Н. ДНК-вычисления для задачи нахождения гамильтонова пути // Самоорганизация и искусственный интеллект в

- группах автономных роботов: методология, теория, практика. Коллективная монография. Под редакцией О.Н. Граничина, С.Ф. Сергеева. — 2020. — С. 43–50.
- [27] Сергеевко А.Н. ДНК-вычисления для задачи нахождения гамильтонова пути // Стохастическая оптимизация в информатике. — 2020. — Т. 16, № 1. — С. 40–47.
- [28] Сергеевко А.Н. Разработка технологии отслеживания летательных аппаратов в условиях неопределенностей // Молодой ученый. — 2020. — Т. 35, № 325. — С. 3–6.
- [29] Сергеевко А.Н. Распределенное отслеживание большого количества летательных аппаратов в условиях неопределенностей // Навигация и управление движением. Материалы XXII конференции молодых ученых с международным участием. — 2020. — С. 319–321.
- [30] Сергеевко А.Н. Верхняя граница оценок, полученных совмещенным рандомизированным алгоритмом стохастической аппроксимации и протоколом локального голосования для задачи трекинга // Навигация и управление движением. Материалы XXIII конференции молодых ученых с международным участием. — 2021. — С. 255–257.
- [31] Сергеевко А.Н., Граничин О.Н. ДНК-вычисления как способ решения задачи коммивояжера // Восемнадцатая Национальная конференция по искусственному интеллекту с международным участием КИИ-2020. Труды конференции. Под ред. В.В. Борисова, О.П. Кузнецова. — 2020. — С. 137–144.
- [32] Сергеевко А.Н., Граничин О.Н. Задача управления сенсорной сетью на основе рандомизированного и многоагентного подходов и ее приложения // Компьютерные инструменты в образовании. — 2022. — С. 94–107.
- [33] Чеботарев П.Ю., Агаев Р.П. Согласование характеристик в многоагентных системах и спектры лапласовских матриц орграфов // Автоматика и телемеханика. — 2009. — № 3. — С. 136–151.
- [34] Accelerated Simultaneous Perturbation Stochastic Approximation for Tracking Under Unknown-but-Bounded Disturbances / V. Erofeeva, O. Granichin, Munira Tursunova et al. // 2022 American Control Conference (ACC). — 2022. — P. 1582–1587.

- [35] Adleman L.M. Molecular computation of solutions to combinatorial Problems // Science, New Series. — 1994. — Vol. 266, no. 5187. — P. 1021–1024.
- [36] Amelin K.S., Granichin O.N. Multiagent network control for a group of light UAVs // Neurocomputers: Design & Application. — 2011. — Vol. 2011, no. 6. — P. 64–72.
- [37] Amelina N., Fradkov A. Approximate consensus in the dynamic stochastic network with incomplete information and measurement delays // Automation and Remote Control. — 2012. — Vol. 73, no. 11. — P. 1765–1783.
- [38] Approximate consensus in stochastic networks with application to load balancing / N. Amelina, A. Fradkov, Y. Jiang, D.J. Vergados // IEEE Transactions on Information Theory. — 2015. — Vol. 61, no. 4. — P. 1739–1752.
- [39] Barricelli N. A. Esempi numerici di processi di evoluzione // Methodos. — 1954. — P. 45–68.
- [40] Barricelli N. A. Numerical testing of evolution theories. Part II. Preliminary tests of performance, symbiogenesis and terrestrial life // Acta Biotheoretica. — 1963. — P. 99–126.
- [41] Bhattacharya K., Vicsek T. Collective decision making in cohesive flocks // New Journal of Physics. — 2010. — Vol. 12, no. 9. — P. 093019.
- [42] Blackman S. S. Multiple hypothesis tracking for multiple target tracking // IEEE Aerospace and Electronic Systems Magazine. — 2004. — Vol. 19, no. 1. — P. 5–18.
- [43] Borkar Vivek S. Stochastic Approximation: A Dynamical Systems Viewpoint. — Cambridge : Cambridge University Press, 2008.
- [44] Bullo F. Lectures on Network Systems. — published online at <http://motion.me.ucsb.edu/book-lns>, 2016. — With contributions by J. Cortes, F. Dorfler, and S. Martinez.
- [45] Bullo F., Cortés J., Martínez S. Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms. — 2009. — 07. — P. 333.

- [46] Consensus-based Distributed Algorithm for Multisensor-Multitarget Tracking under Unknown-but-Bounded Disturbances / N. Amelina, V. Erofeeva, O. Granichin et al. // IFAC-PapersOnLine. — 2020. — Vol. 53, no. 2. — P. 3589–3595. — 21th IFAC World Congress.
- [47] Convergence Analysis of Weighted SPSA-based Consensus Algorithm in Distributed Parameter Estimation Problem / A. Sergeenko, V. Erofeeva, O. Granichin et al. // Proc. of the 19th IFAC SYSID: learning models for decision and control. — Vol. 54. — 2021. — P. 126–131.
- [48] Cooperative Control of Multi-agent Systems: Optimal and Adaptive Design Approaches / F. L. Lewis, H. Zhang, K. Hengster-Movric, A. Das. — Springer Science & Business Media, 2013.
- [49] DeGroot M. H. Reaching a Consensus // Journal of the American Statistical Association. — 1974. — Vol. 69. — P. 118–121.
- [50] Del Moral P., Doucet A. Particle Motions in Absorbing Medium with Hard and Soft Obstacles // Stochastic Analysis and Applications. — 2004. — Vol. 22, no. 5. — P. 1175–1207.
- [51] Differentiated consensus in decentralized load balancing problem with randomized topology, noise, and delays / N. Amelina, O. Granichin, O. Granichina, Y. Jiang // 53rd IEEE Conference on Decision and Control / IEEE. — 2014. — P. 6969–6974.
- [52] Distributed optimization and statistical learning via the alternating direction method of multipliers / S. Boyd, N. Parikh, E. Chu et al. // Foundations and Trends® in Machine learning. — 2011. — Vol. 3, no. 1. — P. 1–122.
- [53] Dorigo M., Maniezzo V., Coloni A. Ant system: optimization by a colony of cooperating agents // IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). — 1996. — Vol. 26, no. 1. — P. 29–41.
- [54] Dynamic Voltage-Frequency Optimization using Simultaneous Perturbation Stochastic Approximation / E. Bogdanov, A. Bozhnyuk, D. Bykov et al. // 2021 60th IEEE Conference on Decision and Control (CDC). — 2021. — P. 3774–3779.

- [55] Emergent intelligence via self-organization in group of robotics devices / K. Amelin, O. Granichin, A. Sergeenko, Z.V. Volkovich // Mathematics. — 2021. — Vol. 9, no. 12. — P. 1314.
- [56] Epidemic Algorithms for Replicated Database Maintenance / A. Demers, D. Greene, C. Hauser et al. // Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing. — PODC '87. — New York, NY, USA : Association for Computing Machinery, 1987. — P. 1–12.
- [57] Equation of State Calculations by Fast Computing Machines / N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth et al. // The Journal of Chemical Physics. — 1953. — Vol. 21, no. 6. — P. 1087–1092.
- [58] Erofeeva V., Granichin O., Granichina O. Multi-sensor task assignment using linear matrix inequalities in the multiple target tracking problem // IFAC-PapersOnLine. — 2018. — Vol. 51, no. 15. — P. 880–885.
- [59] Erofeeva V., Granichin O., Sergeenko A. Distributed Stochastic Optimization with Heavy-Ball Momentum Term for Parameter Estimation // 2021 5th Scientific School Dynamics of Complex Networks and their Applications (DCNA). — 2021. — P. 69–72.
- [60] Erofeeva V., Sergeenko A., Granichin O. Accelerated Online Distributed Optimization for Parameter Estimation Under Uncertainties // 2022 6th Scientific School Dynamics of Complex Networks and their Applications (DCNA). — 2022. — P. 244–247.
- [61] Fax J.A., Murray R.M. Information flow and cooperative control of vehicle formations // IEEE Transactions on Automatic Control. — 2004. — Vol. 49, no. 9. — P. 1465–1476.
- [62] Glover F. Future Paths for Integer Programming and Links to Artificial Intelligence // Computers and Operations Research. — 1986. — Vol. 13, no. 5.
- [63] Granichin O. Linear regression and filtering under nonstandard assumptions (arbitrary noise) // IEEE Transactions on Automatic Control. — 2004. — Vol. 49, no. 10. — P. 1830–1837.

- [64] Granichin O., Amelina N. Simultaneous perturbation stochastic approximation for tracking under unknown but bounded disturbances // IEEE Transactions on Automatic Control. — 2015. — Vol. 60, no. 6. — P. 1653–1658.
- [65] Granichin O., Volkovich V., Toledano-Kitai D. Randomized algorithms in automatic control and data mining. — Springer, 2015. — Vol. 67.
- [66] Grant M., Boyd S. CVX: Matlab software for disciplined convex programming, version 2.1. — 2014.
- [67] Hlinka O., Hlawatsch F., Djuric P. M. Distributed particle filtering in agent networks: A survey, classification, and comparison // IEEE Signal Processing Magazine. — 2013. — Vol. 30, no. 1. — P. 61–81.
- [68] Huang M. Stochastic approximation for consensus: a new approach via ergodic backward products // IEEE Transactions on Automatic Control. — 2012. — Vol. 57, no. 12. — P. 2994–3008.
- [69] Huang M., Manton J. H. Coordination and Consensus of Networked Agents with Noisy Measurements: Stochastic Algorithms and Asymptotic Behavior // SIAM Journal on Control and Optimization. — 2009. — Vol. 48, no. 1. — P. 134–161.
- [70] ICAO. — Multilateration (MLAT) Concept of use, Edition 1.0 edition, 2007.
- [71] Isaksson A. J., Gustafsson F. Comparison of some Kalman filter based methods for manoeuvre tracking and detection // IEEE Conference on Decision and Control. — Vol. 2. — 1995. — P. 1525–1531.
- [72] Ivanskiy Y., Sergeenko A. Distributed algorithms for tracking the trajectories of many objects by the set of mobile sensors // IOP Conference Series: Materials Science and Engineering. — 2020. — Vol. 984, no. 1. — P. 012003.
- [73] Jadbabaie A., Lin Jie, Morse A.S. Coordination of groups of mobile autonomous agents using nearest neighbor rules // IEEE Transactions on Automatic Control. — 2003. — Vol. 48, no. 6. — P. 988–1001.

- [74] Kirkpatrick S., Gelatt Jr. C.D., Vecchi M.P. Optimization by simulated annealing // *Science*. — 1983. — Vol. 220, no. 4598. — P. 671–680.
- [75] Krylatov A. Yu. Network flow assignment as a fixed point problem // *Journal of Applied and Industrial Mathematics*. — 2016. — Vol. 10, no. 2. — P. 243–256.
- [76] Kushner H. J., Yin G. *Stochastic Approximation and Recursive Algorithms and Applications*. — New York, Springer-Verlag, 2003.
- [77] Leonard M. R., Zoubir A. M. Multi-target tracking in distributed sensor networks using particle PHD filters // *Signal Processing*. — 2019. — Vol. 159. — P. 130–146.
- [78] Li X. R., Bar-Shalom Y. Design of an interacting multiple model algorithm for air traffic control tracking // *IEEE Transactions on Control Systems Technology*. — 1993. — Vol. 1, no. 3. — P. 186–194.
- [79] Lynch N. A. *Distributed algorithms*. — Elsevier, 1996.
- [80] Maritime Surveillance Using Multiple High-Frequency Surface-Wave Radars / S. Maresca, P. Braca, J. Horstmann, R. Grasso // *IEEE Transactions on Geoscience and Remote Sensing*. — 2014. — Vol. 52, no. 8. — P. 5056–5071.
- [81] Matviychuk O. State estimation for bilinear impulsive control systems under uncertainties // *Cybernetics and Physics*. — 2018. — Vol. 7, no. 10. — P. 35–40.
- [82] Metropolis N., Ulam S. The Monte Carlo Method // *Journal of the American Statistical Association*. — 1949. — Vol. 44, no. 247. — P. 335–341.
- [83] Multi-target Tracking in Multiple Non-overlapping Cameras Using Fast-Constrained Dominant Sets / Y.T. Tesfaye, E. Zemene, A. Prati et al. // *Int. J. Computer Vision*. — 2019. — Vol. 127. — P. 1303–1320.
- [84] Nedic A., Olshevsky A. Distributed Optimization Over Time-Varying Directed Graphs // *IEEE Transactions on Automatic Control*. — 2015. — Vol. 60, no. 3. — P. 601–615.



- [85] Nedic A., Ozdaglar A. Distributed subgradient methods for multi-agent optimization // IEEE Transactions on Automatic Control. — 2009. — Vol. 54, no. 1. — P. 48.
- [86] Nedic A., Ozdaglar A.E., Parrilo P.A. Constrained Consensus and Optimization in Multi-Agent Networks // IEEE Transactions on Automatic Control. — 2010. — Vol. 55, no. 4. — P. 922–938.
- [87] Network science on belief system dynamics under logic constraints / N.E. Friedkin, A.V. Proskurnikov, R. Tempo, S.E. Parsegov // Science. — 2016. — Vol. 354, no. 6310. — P. 321–326.
- [88] Novel Type of Phase Transition in a System of Self-Driven Particles / T. Vicsek, A. Czirók, E. Ben-Jacob et al. // Phys. Rev. Lett. — 1995. — Aug. — Vol. 75. — P. 1226–1229.
- [89] Olfati-Saber R. Distributed Kalman filter with embedded consensus filters // Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on / IEEE. — 2005. — P. 8179–8184.
- [90] Olfati-Saber R., Murray R. M. Consensus problems in networks of agents with switching topology and time-delays // IEEE Transactions on automatic control. — 2004. — Vol. 49, no. 9. — P. 1520–1533.
- [91] Polyak B. T., Khlebnikov M. V., Shcherbakov P. S. Sparse feedback in linear control systems // Automation and Remote Control. — 2014. — Vol. 75, no. 12. — P. 2099–2111.
- [92] Proskurnikov A., Granichin O. Evolution of clusters in large-scale dynamical networks // Cybernetics and Physics. — 2018. — 11.
- [93] Proskurnikov A.V., Tempo R. A tutorial on modeling and analysis of dynamic social networks. Part I // Annual Reviews in Control. — 2017. — Vol. 43. — P. 65–79.
- [94] Proskurnikov A. V. Average consensus in networks with nonlinearly delayed couplings and switching topology // Automatica. — 2013. — Vol. 49, no. 9. — P. 2928–2932.

- [95] Proskurnikov A. V., Matveev A. S., Cao M. Opinion Dynamics in Social Networks With Hostile Camps: Consensus vs. Polarization // *IEEE Transactions on Automatic Control*. — 2016. — Vol. 61, no. 6. — P. 1524–1536.
- [96] Rajagopal R., Wainwright M. J. Network-based consensus averaging with general noisy channels // *IEEE Transactions on Signal Processing*. — 2011. — Vol. 59, no. 1. — P. 373–385.
- [97] Ren W., Beard R. W. Consensus seeking in multiagent systems under dynamically changing interaction topologies // *IEEE Transactions on Automatic Control*. — 2005. — Vol. 50, no. 5. — P. 655–661.
- [98] Ren W., Beard R. W., Atkins E. M. Information consensus in multivehicle cooperative control // *IEEE Control Systems Magazine*. — 2007. — Vol. 27, no. 2. — P. 71–82.
- [99] Rutherford S., Bassler B. Bacterial quorum sensing: its role in virulence and possibilities for its control // *Cold Spring Harbor Perspectives in Medicine*. — 2012. — 11. — Vol. 2.
- [100] Rzevski G., Skobelev P. Emergent Intelligence in Large Scale Multi-Agent Systems // *Int. J. Educ. Inf. Technol.* — 2007. — Vol. 1. — P. 64–71.
- [101] Rzevski G., Skobelev P. *Managing Complexity*. — WIT Press, 2014. — P. 217.
- [102] Sensor Selection under Unknown but Bounded Disturbances in Multi-Target Tracking Problem / V. Erofeeva, O. Granichin, O. Granichina et al. // *Proceedings of the 2019 27th Mediterranean Conference on Control and Automation (MED)*. — 2019. — P. 215–220.
- [103] Sergeenko A., Granichin O. Sensor network control based on randomized and multi-agent approaches // *Cybernetics and Physics*. — 2022. — Vol. 11, no. 2. — P. 94–105.
- [104] Sergeenko A., Granichin O., Proskurnikov A.V. Advanced SPSA-based Algorithm for Multi-Target Tracking in Distributed Sensor Networks // *2020 59th IEEE Conference on Decision and Control (CDC)*. — 2020. — P. 2424–2429.

- [105] Sergeenko A.N., Granichin O.N., Yakunina M.V. Hamiltonian path problem: the performance comparison deoxyribonucleic acid computing and the branch-and-bound method // Journal of Physics: Conference Series. “International Workshop Navigation and Motion Control”. — Vol. 1536. — 2020. — P. 012003.
- [106] Sergeenko A., Yakunina M., Granichin O. Hamiltonian path problem solution using DNA computing // Cybernetics and Physics. — 2020. — Vol. 9, no. 1. — P. 69–74.
- [107] Shchegryaev A. N., Zakharov V. V. Multi-period cooperative vehicle routing games // Contributions to Game Theory and Management. — 2014. — Vol. 7, no. 1. — P. 349–359.
- [108] Simultaneous Perturbation Stochastic Approximation-Based Consensus for Tracking Under Unknown-But-Bounded Disturbances / O. Granichin, V. Erofeeva, Y. Ivanskiy, Y. Jiang // IEEE Transactions on Automatic Control. — 2021. — Vol. 66, no. 8. — P. 3710–3717.
- [109] Simultaneous perturbation stochastic approximation in decentralized load balancing problem / N. Amelina, V. Erofeeva, O. Granichin, N. Malkovskii // IFAC-PapersOnLine. — 2015. — Vol. 48, no. 11. — P. 936–941.
- [110] Smyth H. D. Atomic Energy for Military Purposes (The Smyth Report). — 1945. — Online; accessed 03 May 2022.
- [111] Spall J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation // IEEE Transactions on Automatic Control. — 1992. — Vol. 37, no. 3. — P. 332–341.
- [112] Spall J. C. Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. — John Wiley & Sons, 2005. — P. 618.
- [113] Spall J. C. Identification for Systems With Binary Subsystems // IEEE Transactions on Automatic Control. — 2014. — Vol. 59, no. 1. — P. 3–17.
- [114] Survey on Blockchain Technology: Evolution, Architecture and Security / M. Nasir, M. Bhutta, A. K. Khwaja, et al. // IEEE Access. — 2021. — Vol. 10. — P. 61048–61073.

- [115] Toward Optimal Distributed Node Scheduling in a Multihop Wireless Network Through Local Voting / D. J. Vergados, N. Amelina, Y. Jiang et al. // IEEE Transactions on Wireless Communications. — 2018. — Vol. 17, no. 1. — P. 400–414.
- [116] Tsitsiklis J. N. Problems in Decentralized Decision Making and Computation : Ph. D. thesis / J. N Tsitsiklis ; PhD. dissertation, MIT. — 1984.
- [117] Tsitsiklis J. N., Bertsekas D., Athans M. Distributed asynchronous deterministic and stochastic gradient optimization algorithms // IEEE Transactions on Automatic Control. — 1986. — Vol. 31, no. 9. — P. 803–812.
- [118] Tsyganov A. V., Tsyganova Y. V., Golubkov A. V. Decentralized algorithm for detecting changes in the motion mode of an object based on multisensor data // 2022 VIII International Conference on Information Technology and Nanotechnology (ITNT) / IEEE. — 2022. — P. 1–5.
- [119] Vicsek T., Zafeiris A. Collective motion // Physics Reports. — 2012. — Vol. 517, no. 3. — P. 71–140.
- [120] Watson J.D., Crick F.H.C. A structure for deoxyribose nucleic acid // Nature. — 1953. — Vol. 171, no. 346. — P. 737–738.
- [121] Weighted SPSA-based Consensus Algorithm for Distributed Cooperative Target Tracking / V. Erofeeva, O. Granichin, A. Proskurnikov, A. Sergeenko // Proc. of the 2021 European Control Conference. — 2021. — P. 1074–1079.
- [122] Zhu H., Spall J. C. Tracking performance of stochastic gradient algorithms with constant step sizes // 2016 IEEE 55th Conference on Decision and Control (CDC) / IEEE. — 2016. — P. 5310–5315.

ST. PETERSBURG STATE UNIVERSITY

*Manuscript copyright*

Sergeenko Anna Nikolayevna

**SENSOR NETWORK CONTROL BASED  
ON RANDOMIZED AND MULTI-AGENT  
APPROACHES**

Scientific specialty 1.2.3. Theoretical informatics, cybernetics

**Thesis for the degree of  
Candidate of Physical and Mathematical  
Sciences**

Translation from Russian

Scientific supervisor:  
Doctor of Physical and Mathematical Sciences, Professor  
Oleg Nikolayevich Granichin

St. Petersburg  
2023

# Contents

<b>Introduction</b>	<b>108</b>
<b>Chapter 1. Control Model for Sensor Network Based on Randomized and Multi-Agent Approaches with Communication Constraints and Significant Uncertainties</b>	<b>118</b>
1.1 Notations and Definitions . . . . .	118
1.2 Network Model for Observing Moving Objects . . . . .	121
1.3 Randomized and Multi-agent Approaches . . . . .	129
1.4 Sensor Network Control Problem . . . . .	140
1.4.1 Optimization of Target and Sensor Network Selection .	141
1.4.2 Randomization of Topology . . . . .	143
1.5 Summary . . . . .	147
<b>Chapter 2. Distributed Randomized Stochastic Approximation Algorithm Combined with Local Voting Protocol</b>	<b>149</b>
2.1 Optimization of Nonstationary Mean Risk Functional . . . . .	149
2.2 Distributed Randomized Stochastic Optimization Algorithm Combined with Local Voting Protocol for Moving Target Estimation . . . . .	152
2.3 Weighted Version of the Distributed Randomized Stochastic Optimization Algorithm Combined with Local Voting Protocol	157
2.4 Distributed Randomized Stochastic Optimization Algorithm Combined with Local Voting Protocol for Estimating the Positions of Static Objects . . . . .	163
2.5 Summary . . . . .	168

<b>Chapter 3. Object Motion Tracking with Uncertainties and Constraints on Agent Connections</b>	<b>170</b>
3.1 Numerical Experiments . . . . .	170
3.2 Prototype of a System for Target Tracking by a Distributed Sensor Network . . . . .	175
3.2.1 System Architecture . . . . .	175
3.2.2 Graphical Interface . . . . .	178
3.2.3 Examples of System Operation . . . . .	181
3.3 Practical Application of Randomized Stochastic Approximation Algorithm Combined with Local Voting Protocol	188
3.4 Summary . . . . .	189
<b>Conclusion</b>	<b>191</b>
<b>Bibliography</b>	<b>192</b>

## Introduction

Currently, the problem of real-time control of a sensor network, given the existing constraints on communication, has numerous practical applications. Examples include network control in communication networks, as described in [2] (B. R. Andrievskiy, A. S. Matveev, A. L. Fradkov), mobile robotics as discussed in [36] (O. N. Granichin), [45] (F. Bullo), and control in transportation and logistics networks mentioned in [101] (G. A. Rzhevsky and P. O. Skobelev), [42] (S. S. Blackman), [9] (V. I. Gorodetsky).

In many studies addressing the problem of tracking changes in dynamic system parameters in real-time, centralized approaches are commonly used. In these approaches, each sensor transmits data to a central hub where all the data is processed. It should be noted that as the size of the system and the number of observed characteristics increase, the sensor network needs to be expanded to ensure reliable and fast tracking. Consequently, the channels through which data is transmitted become more congested as the sensor network grows, resulting in data transmission delays. Additionally, the central data processing hub becomes overloaded with incoming observations and the computation of desired system characteristics. With the advancement of information technologies, distributed computing has gained increasing interest and is applicable to tasks where data processing can be moved closer to its source. Such computations are more reliable compared to centralized computations since failures in one computing unit do not halt the entire system. A shift of research focus from specialized centralized systems to decentralized systems can be observed in works on distributed computing by N. A. Lynch [79], optimal decision-making theory by M. H. DeGroot [49], collective behavior by T. Vicsek [88, 119], and distributed decision-making methods by J. N. Tsitsiklis [116, 117] in the field of control theory.

The progress in electronics and information technology has sparked particular interest in multi-agent technologies that utilize networks of interconnected autonomous units (agents), which can be distributed over large distances and serve as sensors, data processors, actuators, and sensors. Thus,



sensors nowadays can possess computational capabilities and have access to measurements from other sensors. However, in practice, various limitations often arise. Each sensor typically interacts only with a few neighboring network nodes, and the communication between sensors may be limited due to factors such as limited channel bandwidth or data transmission delays. Mathematically, such constraints can be modeled as limited data transmission rates or a time-varying communication graph, which naturally arises when sensors have to share multiple communication lines or a narrow frequency band. In the works of R. P. Agayev [3], B. R. Andrievsky [2], R. W. Beard [97,98], F. Bullo [45], A. Jadbabaie [73], V. V. Zakharova [107], M. I. Karpova [15], A. Yu. Krylatov [75], F. L. Lewis [48], A. S. Morse [73], R. M. Murray [61,90], R. Olfati-Saber [90], L. A. Petrosyan [15], W. Ren [97,98], P. Yu. Chebotarev [3], Yu. V. Tsiganova [118], and others, fundamental principles of building distributed multi-agent coordination and motion control algorithms are established. These works also discuss practical application areas of the developed approaches for organizing collective behavior of observers. The aforementioned issues justify the investigation of a multi-agent approach to the considered problem.

During real-time measurement of dynamic system parameter characteristics, various measures of uncertainty typically arise. These problems are often formulated as the optimization of certain risk functionals, such as mean squared tracking error. In many cases, empirical functionals, maximum likelihood methods, or Bayesian estimation are employed. However, such algorithms rely on significant assumptions and statistical properties of observation noise, assuming their centeredness and uncorrelated nature. The problem of choosing the best estimates in one sense or another is particularly interesting. The least squares method is known to be optimal when dealing with independently and identically distributed Gaussian observation noise, but its application is not justified when such conditions are not met. In particular, a challenging problem arises when the observed parameter characteristics can change with arbitrary unknown but bounded disturbances, and systematic errors (model errors) occur in measurements, which are often

difficult to exclude. In the works of A. S. Matveev [95], A. Nedic [84, 86], R. Olfati-Saber [89], A. V. Proskurnikov [87, 94, 95], the results of research on distributed optimization algorithms and consensus control are presented. For the problem of achieving consensus on graphs in the presence of noisy measurements of neighbors' states, the works of M. J. Wainwright [96], D. J. Vergados et al. [115], O. N. Granichina et al. [109], M. Huang [69], A. L. Fradkov and N. O. Amelina [38] consider the application of stochastic approximation algorithms. This type of algorithm is one of the most important classes among approaches to solving optimization problems with uncertainties. For example, in the presence of arbitrary external disturbances, it is possible to find a solution to the aforementioned problem using randomized stochastic optimization algorithms described in the papers by O. N. Granichina et al. [64, 65, 108]. In particular, in the work [108], a randomized stochastic optimization algorithm combined with a local voting protocol was proposed, which can effectively solve optimization problems in the presence of unknown-but-bounded noise in the measurements of the observed function. Additionally, the convergence of the proposed algorithm was studied. An important task is the application of the stochastic approximation algorithm combined with the local voting protocol for the control of a sensor network in tracking the parameters of a dynamic system.

The dissertation addresses the problem of controlling a sensor network in the context of distributed *target tracking* with networks of connected sensors (radars, hydrolocators, cameras, etc.), which is a classical signal processing problem [67] (O. Hlinka and colleagues). Such distributed tracking problems for multiple targets have been extensively studied in the literature due to numerous applications in air traffic control [78], maritime surveillance [80], road traffic control [71], and video surveillance [83]. However, usually, uncertainties with predefined traditional statistical properties of centrality and independence were considered, which is often not valid in practice. In particular, in the variety of possible target behaviors, the available statistics are insufficient, and the trajectories of maneuvering targets can be described as arbitrary unknown but bounded disturbances. To address such a problem,

a modified distributed stochastic approximation algorithm combined with a local voting protocol is proposed for a network model of tracking moving objects.

The mentioned problems and trends confirm the relevance of the topic of the dissertation research.

*The goal of the study* is to develop algorithms for controlling a distributed sensor network in real-time, considering limitations in communication and significant uncertainties in the description of the system.

To achieve this goal, the following tasks were formulated and addressed:

- 1) develop a model for controlling the sensor network based on randomized and multi-agent approaches, taking into account communication constraints and significant uncertainties in the system description;
- 2) enhance the distributed randomized stochastic optimization algorithm combined with a local voting protocol and investigate the properties of its estimates for the tracking task (tracking changes in parameters) using the developed model for controlling the sensor network;
- 3) investigate the applicability conditions of the enhanced distributed randomized stochastic optimization algorithm combined with a local voting protocol for the networked model of object tracking, considering uncertainties and constraints on the number of connections between sensors and the number of observable objects.

*Research methods.* The dissertation employs methods from estimation theory, optimization theory, control theory, graph theory, probability theory, and mathematical statistics. Stochastic optimization methods, randomized algorithms, and simulation modeling are also used.

*Main results.* The following scientific results were obtained during the work:

- 1) a model of sensor network control based on randomized and multi-agent approaches was developed;

- 2) the distributed randomized stochastic optimization algorithm combined with a local voting protocol has been modified and its estimation properties have been investigated for the tracking (monitoring parameter changes) task using the developed sensor network control model;
- 3) the applicability conditions of the modified distributed randomized stochastic optimization algorithm combined with a local voting protocol for the network observation model have been studied. The results have been tested in the task of observing moving objects in the presence of uncertainties and constraints on the number of communications between sensors.

*Scientific novelty.* All the main scientific results of the dissertation are novel.

*Theoretical value and practical significance.* The theoretical value of the results lies in the development of a model for tracking objects by a group of observers, the modernization of a distributed randomized stochastic optimization algorithm combined with a local voting protocol, the investigation of the properties of its estimates, as well as the conditions for the applicability of a network observation model. The proposed methods and approaches can be used to solve a range of practical tasks. In particular, they can be applied for tracking aerial objects in airspace and coordinating their movements.

*Research Validation.* The results of the dissertation were presented at the seminars of the Department of System Programming of the Faculty of Mathematics and Mechanics at St. Petersburg State University, the conference «Week of Science SPbPU 2018» (November 19-24, 2018, St. Petersburg, Russia), the International Workshop «Navigation and Motion Control» (NMC 2019) (October 16-20, 2019, Leningrad Region, Russia), the XXI Conference of Young Scientists «Navigation and Motion Control» (with international participation) (March 19-22, 2019, St. Petersburg, Russia), the conference «The 9th International Scientific Conference on Physics and Control» (September 8-11, 2019, Innopolis, Russia), the XXII Conference of Young Scientists «Navigation and Motion Control» (with international participation)

(March 17-20, 2020, St. Petersburg, Russia), the conference «27th Mediterranean Conference on Control and Automation» (MED 2019) (July 1-4, 2019, Akko, Israel), the 13th Multiconference on Control Problems (Mathematical Control Theory and its Applications) (October 6-8, 2020, St. Petersburg, Russia), the 18th National Conference on Artificial Intelligence (October 10-16, 2020, Moscow, Russia), the conference «59th IEEE Conference on Decision and Control» (CDC 2020) (December 14-18, 2020, Jeju, Republic of Korea), the 63rd All-Russian Scientific Conference of MIPT (November 23-29, 2020, Moscow, Russia), the conference «IFAC World Congress» (July 11-17, 2020, Berlin, Germany), the conference «19th IFAC Symposium on System Identification» (SYSID 2021) (July 13-16, 2021, Padua, Italy), the XXIII Conference of Young Scientists «Navigation and Motion Control» (with international participation) (March 23-29, 2021, St. Petersburg, Russia), the conference «5th Scientific School Dynamics of Complex Networks and their Applications» (DCNA 2021) (September 13-15, 2021, Kaliningrad, Russia), the conference «60th IEEE Conference on Decision and Control» (CDC 2021) (December 13-17, 2021, Austin, USA), the conference «European Control Conference» (ECC 2021) (June 29 - July 2, 2021, Rotterdam, Netherlands), the XIV Conference of Young Scientists «Navigation and Motion Control» (with international participation) (March 15-18, 2022, St. Petersburg, Russia), the conference «Americas Control Conference» (ACC 2022) (June 8-10, 2022, Atlanta, USA), and the conference «6th Scientific School Dynamics of Complex Networks and their Applications» (DCNA 2022) (September 14-16, 2022, Kaliningrad, Russia).

The results of the dissertation were also utilized in research work supported by the following grants: Russian Science Foundation (RSF) grant 16-19-00057 «Adaptive Control with Predictive Models in Variable State Space Structure Applied to Networked Motion Control Systems and Medical Equipment Automation,» Russian Foundation for Basic Research (RFBR) grant 20-01-00619 «Randomized Algorithms for Multi-Agent Optimization, Pattern Recognition, and Estimation under Significant Uncertainties,» and RSF grant 21-19-00516 «Multi-Agent Adaptive Control in Networked Dynamic

Systems Applied to Groups of Robotic Devices under Uncertainties.» The project «Development of Technology for Tracking Aircraft under Uncertainties» received recognition through a diploma as the winner of the Youth Scientific and Innovative Competition (UMNIK-2019). The project «Development of Distributed Algorithms for Tracking Trajectories of Multiple Objects using an Array of Mobile Sensors» was awarded a diploma as the winner of the competition for students and graduate students from universities and academic institutions located in St. Petersburg in 2020.

*Publication of Results.* The main research results are reflected in the following works: [7, 14, 18, 19, 22–34, 46, 47, 54, 55, 59, 60, 72, 102–106, 121]. The applicant published 30 scientific papers, of which one publication is a certificate of registration of a computer program, 14 were published in publications included in the RSCI, 14 were published in publications indexed in the Scopus database, and 1 published in a publication included in the list of HAC.

The works [7, 14, 18, 19, 32–34, 46, 47, 54, 55, 59, 60, 72, 102–106, 121] were written in collaboration. In the work [33], A. N. Sergeenko was responsible for proving the theorems and the results of simulation modeling, while the co-authors were involved in the general problem formulation. In [18], A. N. Sergeenko contributed to the modification of the local voting protocol for controlling the motion of a group of dynamic objects (robots, agents), while the co-authors participated in the general problem formulation. In [7], A. N. Sergeenko conducted simulation modeling, while the co-authors contributed to the general problem formulation and method selection. In [19], A. N. Sergeenko described the approach to link randomization in a sensor network to satisfy cost constraints, with the co-authors involved in the general problem formulation. In [14], A. N. Sergeenko participated in the general problem formulation, while the co-author was responsible for the method selection and simulation modeling. In [32], A. N. Sergeenko contributed to the method selection and experimental results, with the co-author involved in the general problem formulation. In [34], A. N. Sergeenko conducted simulation modeling, while the co-authors were responsible for the general problem formulation, method selection, and theorem proof. In [46], A. N. Sergeenko

participated in proving the theorem, while the co-authors were involved in the general problem formulation, method selection, and simulation modeling. In [47], A. N. Sergeenko contributed to proving the theorem and simulation modeling, while the co-authors participated in the general problem formulation and method selection. In [54], A. N. Sergeenko was responsible for the method selection, while the co-authors contributed to the general problem formulation and experimental results. In [55], A. N. Sergeenko described emergent intelligence, with the co-authors involved in the general problem formulation, method selection, and simulation modeling. In [59], A. N. Sergeenko conducted simulation modeling, while the co-authors contributed to the general problem formulation and method selection. In [72], A. N. Sergeenko conducted simulation modeling, and the co-author participated in the general problem formulation and method selection. In [60], A. N. Sergeenko conducted simulation modeling, while the co-authors were involved in the general problem formulation and method selection. In [102], A. N. Sergeenko conducted simulation modeling, and the co-authors were involved in the general problem formulation, method selection, and theorem proof. In [104], A. N. Sergeenko was responsible for proving the theorems, while the co-authors contributed to the general problem formulation, method selection, and simulation modeling. In [103], A. N. Sergeenko described the history of the emergence of randomized and multi-agent algorithms, method selection, theorem proof, and simulation modeling, while the co-author participated in the general problem formulation. In [106], A. N. Sergeenko was responsible for the method selection, while the co-authors contributed to the general problem formulation and experimental results. In [105], A. N. Sergeenko participated in the general problem formulation and simulation modeling, with the co-authors involved in the method selection. In [121], A. N. Sergeenko proved the theorem, while the co-authors were responsible for the general problem formulation and method selection.

*Structure and volume of the dissertation.* The dissertation consists of an introduction, three chapters, a conclusion, a list of references, and 122 sources. The text spans 99 pages and includes 23 figures and 0 tables.

*Brief content of the work.*

The **introduction** justifies the relevance of the dissertation, states the objective, sets the research tasks, and provides a brief overview of the main results.

In the **first chapter**, the problem of estimating moving objects with a sensor network is described, accompanied by a literature review on the research topic. Section 1.1 introduces the notation and basic definitions. Section 1.2 describes the peculiarities of developing a network observation model for moving objects (targets). Assumptions about the target movement speed, sensors, target observation points, and measurement noise are introduced. Section 1.3 provides a description of randomized and multi-agent approaches and justifies their use in developing the network observation model for targets. Section 1.4 formulates the problem of tracking targets with a sensor network. Various approaches to selecting neighbors for each sensor, taking into account topological constraints, are described in subsections. Subsection 1.4.1 describes an approach based on optimizing target selection by maximizing the intersection of confidence ellipsoids with minimal sensor load. Subsection 1.4.2 presents another approach based on topology randomization and describes its advantages. Section 1.5 presents the conclusions from the first chapter.

In the **second chapter**, modifications of the distributed randomized stochastic optimization algorithm combined with a local voting protocol are formulated, and the properties of its estimates are investigated. Section 2.1 describes the distributed randomized stochastic optimization algorithm combined with a local voting protocol for a general class of functions, provides conditions on this function, and shows that the function developed in the first section satisfies these conditions. Section 2.2 presents the modified distributed randomized stochastic optimization algorithm combined with a local voting protocol for estimating moving targets. Theorem 1 is formulated and proven, reflecting the mean-square quality of the estimates obtained using the proposed algorithm. Theorem 2 is also formulated and proven for choos-



ing a suboptimal step of the algorithm. Section 2.3 describes a weighted version of the algorithm applicable to targets with different types of movement. Theorem 3 is formulated and proven, showing the convergence of the covariance matrix of residuals obtained using the weighted version of the algorithm. Section 2.4 presents a variable-step algorithm for estimating the positions of static objects. Theorem 4 is formulated and proven, showing the convergence rate of the covariance matrix of residuals obtained using the described algorithm. Section 2.5 presents the conclusions from the second chapter.

In the **third chapter**, the results of simulation modeling are presented to illustrate the operation of the proposed methods and approaches. Section 3.1 presents the results of numerical experiments on solving the problem of tracking targets with a distributed sensor network using the developed sensor network model from Section 1.4 and the algorithms from Sections 2.2-2.4. Section 3.2 describes a prototype system for tracking targets with a distributed sensor network. Subsection 3.2.1 presents the system architecture, Subsection 3.2.2 describes the graphical user interface of the system, and Subsection 3.2.3 provides examples of the system's operation. Section 3.3 describes the application of the distributed randomized stochastic optimization algorithm combined with a local voting protocol specifically for real-time tracking of aerial vehicles. Section 3.4 presents the conclusions from the third chapter.

In the **conclusion**, the main results of the dissertation are formulated.

# Chapter 1. Control Model for Sensor Network Based on Randomized and Multi-Agent Approaches with Communication Constraints and Significant Uncertainties

## 1.1 Notations and Definitions

In the text of the dissertation, the following notations are used:

$t, k$  is discrete time;

$i, j, k, m, n, d, p, l, q$  are integers (usually non-negative);

$\alpha, \beta, \gamma, \delta, \epsilon, c, z, \sigma$  are scalar variables;

$\mathbf{a}, \mathbf{b}, \mathbf{x}, \mathbf{r}, \mathbf{s}, \mathbf{u}, \mathbf{\Delta}$  are vector variables;

$a, b$  are scalar or vector variables;

$y$  is observed scalar and vector variable;

$v, w, \xi$  are- disturbances (noise) in observations (measurements);

$A, B, C, D, \Psi, I, \Gamma, H, \Phi, S, Z, W$  are matrices;

$M, Pr, \nabla$  are operators;

$J$  is a matrix or a set of numbers;

$\mathbb{R}$  is set of real numbers;

$\theta$  is an estimated (optimal) value;

$\hat{\theta}$  is a vector (sometimes matrix) in the space of estimated parameters (estimate);

$\langle \cdot, \cdot \rangle$  is a scalar product of vectors;

$f(\cdot, \cdot), F(\cdot), \rho(\cdot, \cdot)$  are real functions;

$\cdot^T$  is transposition of a vector or matrix;

$\cdot^{-1}$  is inverse of a matrix;

$\cdot'$  is pseudoinverse of a matrix;

$\|A\|$  is a Frobenius norm:  $\|A\| = \sqrt{\sum_i \sum_j (a_{i,j})^2}$ ;

$\mathbf{0}$  is a zero vector;

$\mathbf{1}_d = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^d$  is a vector consisting of all ones;

$\mathbf{e}_i = \begin{pmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} \in \mathbb{R}^d$  – canonical basis vector, where the  $i$ -th element is equal

to 1;

$I_d \in \mathbb{R}^{d \times d}$  is a identity matrix;

$J_d = \mathbf{1}_d \mathbf{1}_d^T \in \mathbb{R}^{d \times d}$  is a matrix consisting of all ones;

$A \otimes B$  is a Kronecker product, defined for any matrices  $A$  and  $B$ ;

$A < B, A \leq B$ : matrix ordering in terms of quadratic forms;

$\mathcal{N} = \{1, \dots, n\}$  is a set of vertices;

$\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$  is a set of edges;

$\mathcal{G}_A = (\mathcal{N}, \mathcal{E})$  is a directed graph;

$i \in \mathcal{N}$  is a identifier of vertex  $i$ , and  $(j, i) \in \mathcal{E}$  if there is a directed edge from vertex  $j$  to vertex  $i$ ;

$\mathcal{N}^i = \{j \in \mathcal{N} : (j, i) \in \mathcal{E}\}$  is a set of *neighbors* for node  $i \in \mathcal{N}$ ;

$|\cdot|$  is cardinality of a set;

upper indices are agent indices;

$a^{i,j} > 0$  is the weight associated with the edge  $(j, i) \in \mathcal{E}$ ,  $a^{i,j} = 0$  whenever  $(j, i) \notin \mathcal{E}$ ;

$\deg_i^+(A)$  is the weighted in-degree of vertex  $i \in \mathcal{N}$ ,  $\deg_i^+(A) = \sum_{j=1}^n a^{i,j}$  in the graph  $\mathcal{G}_A$ ;

$\deg_{\max}^+(A)$  is the maximum in-degree among all vertices in the graph  $\mathcal{G}_A$ ;

$\text{col}\{\cdot\}$  is a column vector obtained by stacking its elements on top of each other;

$\text{diag}_n(\mathbf{b})$  is a diagonal matrix with the elements of vector  $\mathbf{b}$  on the diagonal and other elements equal to zero;

$\mathcal{D}(A)$  is a diagonal matrix  $\mathcal{D}(A) = \text{diag}_n(\deg_1^+(A), \dots, \deg_n^+(A))$ ;

$\mathcal{L}(A) = \mathcal{D}(A) - A$  is the Laplacian of the graph  $\mathcal{G}_A$ ;

$\Omega$  is probability space;

$\omega$  is an element of the probability space;

$\mathcal{F}$  is a set of all events;

$P$  is probability measure;

$(\Omega, \mathcal{F}, P)$  is underlying probability space;

$\mathbb{E}$  is mathematical expectation;

$\forall$  is universal quantifier;

$\exists$  is existential quantifier;

$\mathcal{F}_t$  is  $\sigma$ -algebra of all probability events that have occurred up to time  $t = 1, 2, \dots$ ;

$\mathbb{E}_{\mathcal{F}_t}$  is conditional expectation with respect to the  $\sigma$ -algebra  $\mathcal{F}_t$ . Mathematically, this  $\sigma$ -algebra is generated by the values of all random functions (target positions, disturbances, adjacency graph changes) at time instants  $\tau = \{1, 2, \dots, t\}$ .

**Definition 1.** *A directed graph  $\mathcal{G}_A$  is called strongly connected if for every pair of vertices  $j, i \in \mathcal{N}$ , there exists a path of directed edges from  $j$  to  $i$ .*

Let us denote the eigenvalues of the Laplacian  $\mathcal{L}(A)$  arranged in ascending order of their real parts as  $\lambda_1, \dots, \lambda_n$ ,

$$0 \leq \operatorname{Re}(\lambda_1) \leq \operatorname{Re}(\lambda_2) \leq \dots \leq \operatorname{Re}(\lambda_n).$$

It is known that if the graph is strongly connected, then  $\lambda_1 = 0$ , and all other eigenvalues of  $\mathcal{L}$  are located in the open right half-plane of the complex plane (see [1, 48]). The eigenvalue of matrix  $A$  with the maximum absolute value is denoted as  $\lambda_{\max}(A)$ .

## 1.2 Network Model for Observing Moving Objects

Consider a system consisting of a network of  $n$  sensors observing the motion of  $m$  targets. We assume that the sensors and targets are located in the real space  $\mathbb{R}^d$  (e.g.,  $d = 2$  for a plane,  $d = 3$  for air or underwater).

Let

$\mathcal{N} = \{1, 2, \dots, n\}$  be the set of all sensors,

$\mathcal{M} = \{1, 2, \dots, m\}$  be the set of all targets,

$\mathbf{s}_t^i = \begin{pmatrix} s_t^{i,1} \\ \dots \\ s_t^{i,d} \end{pmatrix} \in \mathbb{R}^d$  be the coordinate vector of sensor  $i$  at time  $t$ ,

$$\mathbf{r}_t^l = \begin{pmatrix} r_t^{l,1} \\ \dots \\ r_t^{l,d} \end{pmatrix} \in \mathbb{R}^d \text{ be the coordinate vector of target } l \text{ at time } t,$$

$$\theta_t = \begin{pmatrix} \mathbf{r}_t^1 \\ \dots \\ \mathbf{r}_t^m \end{pmatrix} \in \mathbb{R}^{dm} \text{ be the overall state vector of all targets at time } t.$$

We introduce an assumption about the velocity of the targets.

**A s s u m p t i o n 1.** *Let  $\xi_t^l = \mathbf{r}_t^l - \mathbf{r}_{t-1}^l$  for  $l \in \mathcal{M}$ . The velocity of the targets is uniformly bounded:  $\forall l \in \mathcal{M}, \|\xi_t^l\| \leq \delta < \infty$ , or equivalently,  $\mathbb{E}\|\xi_t^l\|^2 \leq \delta^2$ ,  $\mathbb{E}\|\xi_t^l\|\|\xi_{t-2}^l\| \leq \delta^2$ ,  $\mathbb{E}[\tilde{\xi}_t \tilde{\xi}_t^T] \leq Q_\delta$ , and  $\mathbb{E}[\tilde{\xi}_t \tilde{\xi}_{t-1}^T] \leq Q_\delta$  if the sequence  $\{\xi_t^l\}$  is random.*

We assume that at each time  $t$ , sensor  $i$  can measure the distance to any target  $l$  from a subset of all targets  $\mathcal{M}_t^i \subset \mathcal{M}$ :

$$z_t^{i,l} = \rho(i, l) + w_t^{i,l},$$

where  $w_t^{i,l}$  is the measurement noise, and  $\rho(i, l)$  is the squared distance between sensor  $i$  and target  $l$  at time  $t$ :

$$\rho(i, l) = \|\mathbf{r}_t^l - \mathbf{s}_t^i\|^2 = \sum_{d'=1}^d (r_t^{l,d'} - s_t^{i,d'})^2.$$

Let a sensor network consisting of  $n$  nodes be described by a directed graph  $\mathcal{G}_A$  with a weighted adjacency matrix  $A = [a^{i,j}]$  (the connectivity matrix associated with the graph  $\mathcal{G}_A$ ). Let us denote the subset of sensors  $\mathcal{N}_t^i \subset \mathcal{N}$  with neighbors of sensor  $i$  at time  $t$ , and  $\bar{\mathcal{N}}_t^i \subset \mathcal{N}_t^i$  as the subset of neighbors  $J = \{j_1, \dots, j_p\} \subset \bar{\mathcal{N}}_t^i$  of sensor  $i$ , each of which measures the distance to the same target  $l \in \mathcal{M}_t^i$  as sensor  $i$  (where  $p$  is the number of simultaneous coordinated observations of the same target by different sensors) ([14, 72]).

Let us construct a set of triples  $\mathcal{U}_t$  consisting of possible combinations of sensor  $i \in \mathcal{N}$ , its neighbors  $J \in \bar{\mathcal{N}}_t^i$ , and a target  $l \in \mathcal{M}_t^i$  that they observe

in a coordinated manner at time  $t$ . Let any triple from this set be denoted as  $\mathbf{u} \in \mathcal{U}_t$ .

From a practical point of view, the most interesting cases are when the number of neighbors is minimal for each sensor. In particular, we will consider two cases where the number of simultaneous coordinated observations of the same target is either one,  $p = 1$ , or equal to the dimension of the space,  $p = d$ . In the latter case, assuming error-free measurements of target coordinates by sensors, standard triangulation techniques can be applied.

For each set  $\mathbf{u} = (i, J, l) \in \mathcal{U}_t$  and some  $j_q \in J$ , let us define the difference  $\bar{z}_t^q(\mathbf{u}) = z(i, l) - z(j_q, l)$  between the distance measurements to the same target  $l$  by sensor  $i$  and sensor  $j_q \in J$ . We also introduce  $\bar{w}_t^q(\mathbf{u}) = w_t^{i,l} - w_t^{j_q,l}$  as the difference between the measurement noises of distance to target  $l$  by sensor  $i$  and sensor  $j_q \in J$ . By using the formula for the difference of squares, we can obtain  $p$  equations for each sensor  $i \in \mathcal{N}$ :

$$\begin{aligned} \bar{z}_t^q(\mathbf{u}) &= z_t^{i,l} - z_t^{j_q,l} = \rho(i, l) - \rho(j_q, l) + w_t^{i,l} - w_t^{j_q,l} = \\ &= \sum_{d'=1}^d \left( (r_t^{l,d'} - s_t^{i,d'})^2 - (r_t^{l,d'} - s_t^{j_q,d'})^2 \right) + \bar{w}_t^q(\mathbf{u}) = \\ &= \sum_{d'=1}^d (s_t^{j_q,d'} - s_t^{i,d'}) (2r_t^{l,d'} - s_t^{j_q,d'} - s_t^{i,d'}) + \bar{w}_t^q(\mathbf{u}) = \\ &= \sum_{d'=1}^d 2r_t^{l,d'} (s_t^{j_q,d'} - s_t^{i,d'}) - \sum_{d'=1}^d \left( (s_t^{j_q,d'})^2 - (s_t^{i,d'})^2 \right) + \bar{w}_t^q(\mathbf{u}), \end{aligned}$$

$q = 1, \dots, p$ . These equations can be written in matrix form as follows:

$$2 \begin{bmatrix} (\mathbf{s}_t^{j_1} - \mathbf{s}_t^i)^\top \\ \dots \\ (\mathbf{s}_t^{j_p} - \mathbf{s}_t^i)^\top \end{bmatrix} \mathbf{r}_t^l = \begin{bmatrix} \bar{z}_t^1(\mathbf{u}) + \|\mathbf{s}_t^{j_1}\|^2 - \|\mathbf{s}_t^i\|^2 \\ \dots \\ \bar{z}_t^p(\mathbf{u}) + \|\mathbf{s}_t^{j_p}\|^2 - \|\mathbf{s}_t^i\|^2 \end{bmatrix} - \begin{bmatrix} \bar{w}_t^1(\mathbf{u}) \\ \dots \\ \bar{w}_t^p(\mathbf{u}) \end{bmatrix}. \quad (1.1)$$

$$\text{Let } C_t^{\mathbf{u}} = 2 \begin{bmatrix} (\mathbf{s}_t^{j_1} - \mathbf{s}_t^i)^\top \\ \dots \\ (\mathbf{s}_t^{j_p} - \mathbf{s}_t^i)^\top \end{bmatrix}, \quad D_t^{\mathbf{u}} = \begin{bmatrix} \bar{z}_t^1(\mathbf{u}) + \|\mathbf{s}_t^{j_1}\|^2 - \|\mathbf{s}_t^i\|^2 \\ \dots \\ \bar{z}_t^p(\mathbf{u}) + \|\mathbf{s}_t^{j_p}\|^2 - \|\mathbf{s}_t^i\|^2 \end{bmatrix},$$

$W_t^{\mathbf{u}} = \begin{bmatrix} \bar{w}_t^1(\mathbf{u}) \\ \dots \\ \bar{w}_t^p(\mathbf{u}) \end{bmatrix}$ . Equation (1.1) represents a linear system with respect to the vector  $\mathbf{r}_t^l$ .

Let's consider measurements conducted without noise, which means that  $W_t^{\mathbf{u}} = \mathbf{0}$ . We can express the vector  $\mathbf{r}_t^l$  in terms of  $C_t^{\mathbf{u}}$  and  $D_t^{\mathbf{u}}$ . In the general case, when  $p < d$ , the matrix  $C_t^{\mathbf{u}}$  is not square, so instead of matrix inversion, we'll use the pseudoinverse operation:

$$\begin{aligned} C_t^{\mathbf{u}} \mathbf{r}_t^l &= D_t^{\mathbf{u}}, \\ (C_t^{\mathbf{u}})^{\text{T}} C_t^{\mathbf{u}} \mathbf{r}_t^l &= (C_t^{\mathbf{u}})^{\text{T}} D_t^{\mathbf{u}}, \\ [(C_t^{\mathbf{u}})^{\text{T}} C_t^{\mathbf{u}}]' (C_t^{\mathbf{u}})^{\text{T}} C_t^{\mathbf{u}} \mathbf{r}_t^l &= [(C_t^{\mathbf{u}})^{\text{T}} C_t^{\mathbf{u}}]' (C_t^{\mathbf{u}})^{\text{T}} D_t^{\mathbf{u}}. \end{aligned}$$

As a result, we obtain:

$$I_t^{\mathbf{u}} \mathbf{r}_t^l = H_t^{\mathbf{u}}, \quad (1.2)$$

where  $I_t^{\mathbf{u}} = [(C_t^{\mathbf{u}})^{\text{T}} C_t^{\mathbf{u}}]' (C_t^{\mathbf{u}})^{\text{T}} C_t^{\mathbf{u}}$  and  $H_t^{\mathbf{u}} = [(C_t^{\mathbf{u}})^{\text{T}} C_t^{\mathbf{u}}]' (C_t^{\mathbf{u}})^{\text{T}} D_t^{\mathbf{u}}$ .

**Remark 1.** *If  $p = d$  and the matrix  $C_t^{\mathbf{u}}$  is non-singular, then  $I_t^{\mathbf{u}}$  is an identity matrix, and  $H_t^{\mathbf{u}} = [C_t^{\mathbf{u}}]^{-1} D_t^{\mathbf{u}}$ . If  $p = 1$ , all elements of  $I_t^{\mathbf{u}}$  are zero, except for one element on the diagonal, which is equal to one.*

The goal of the sensor network system is to generate a consistent estimate of the location of all tracked objects at each time  $t$ , denoted as

$\hat{\theta}_t = \begin{pmatrix} \hat{\mathbf{r}}_t^{1,1} \\ \vdots \\ \hat{\mathbf{r}}_t^{m,n} \end{pmatrix} \in \mathbb{R}^{nmd}$ , which minimizes the loss function:

$$\hat{\theta}_t = \underset{\hat{\theta}_t}{\operatorname{argmin}} \mathbb{M} \sum_{i \in \mathcal{N}} F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}), \quad (1.3)$$



where

$$F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \sum_{\mathbf{u} \in \mathcal{U}_t, i=i(\mathbf{u})} f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}),$$

$\hat{\mathbf{r}}_t^{l,i}$  is the estimated location of target  $l$  by sensor  $i$ ,  $F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i})$  is the functional that includes the targets tracked by sensor  $i$ ,  $f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \|I_t^{\mathbf{u}} \hat{\mathbf{r}}_t^{l,i} - H_t^{\mathbf{u}}\|^2$  is the loss (or quality) function, and  $\mathbb{M}$  is either an averaging operator or a maximization operator. In the following, we will consider only the case of averaging:

$$\hat{\theta}_t = \underset{\hat{\theta}_t}{\operatorname{argmin}} \frac{1}{|\mathcal{U}_t|} \sum_{\mathbf{u} \in \mathcal{U}_t} f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \underset{\hat{\theta}_t}{\operatorname{argmin}} \frac{1}{|\mathcal{U}_t|} \sum_{i \in \mathcal{N}} F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}), \quad (1.4)$$

where

$$F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \sum_{\mathbf{u} \in \mathcal{U}_t, i=i(\mathbf{u})} f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}).$$

In practice, any measurements obtained from sensors are always corrupted by noise [28, 29]. Solving the optimization problem becomes more complex with noisy measurements. For measurements with non-zero noise  $W_t^{\mathbf{u}}$ , sensor measurements are formed based on the noisy measurements of the optimized function:

$$y_t^i = \|I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}} + [(C_t^{\mathbf{u}})^T C_t^{\mathbf{u}}]' (C_t^{\mathbf{u}})^T W_t^{\mathbf{u}}\|^2.$$

By applying the formula for the square of the sum, we obtain the following

sequence of expressions.

$$\begin{aligned}
y_t^i &= \\
\|I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}}\|^2 + 2(I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} + \|[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2 &= \\
f_t^i(\mathbf{u}, \mathbf{x}_t^{l,i}) + 2(I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} + W_t^{\mathbf{u}\top} C_t^{\mathbf{u}} C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} &= \\
f_t^i(\mathbf{u}, \mathbf{x}_t^{l,i}) + 2(I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} + \|C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}\|^2 &= \\
f_t^i(\mathbf{u}, \mathbf{x}_t^{l,i}) + v_t^i, & \tag{1.5}
\end{aligned}$$

where  $f_t^i(\mathbf{u}, \mathbf{x}_t^{l,i}) = |I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}}|^2$ ,  $\mathbf{x}_t^{l,i}$  is the current point at which the function is measured (observation point), for example,  $\mathbf{x}_t^{l,i} = \hat{\mathbf{r}}_t^{l,i}$ , and  $v_t^i = 2(I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} + |C_t^{\mathbf{u}\top} W_t^{\mathbf{u}}|^2$  is some noise.

Now, let's state the assumptions about the sensors, observation points, and measurement noise.

### A s s u m p t i o n 2.

1. *The distance between sensor  $\mathbf{s}_t^i$  and sensors  $\mathbf{s}_t^{j_1}, \dots, \mathbf{s}_t^{j_p}$  is bounded from above and below,  $\forall i \in \mathcal{N}, j_q \in J, J \subset \bar{\mathcal{N}}_t^i : c_s \leq \|\mathbf{s}_t^i - \mathbf{s}_t^{j_q}\| \leq c_S, q = 1, \dots, p$ .*
2. *The constraints on the target's location are known. In other words, for all  $l \in \mathcal{M}$ , the observation points  $\mathbf{x}_t^{l,i}$  are chosen such that they are within a convex closed bounded subset that contains the target vector  $\mathbf{r}_t^l$ .*
3. *For all  $i \in \mathcal{N}, l \in \mathcal{M}$ , the noise  $w_t^{i,l}$  is independent, symmetrically distributed centered noise that is uniformly bounded in the mean square sense  $\mathbb{E}\|w_t^{i,l}\| \leq \sigma_w$  and has uniformly bounded fourth moments  $\mathbb{E}\|w_t^{i,l}\|^4 \leq \sigma_4$ .*

**L e m m a 1.** *If Assumption 2 holds, then*

1.  $\|C_t^{\mathbf{u}\top}\|^4 \leq 4p^2 c_S^4$ ;

2.  $\| [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} \|^2 \leq \frac{c_s^2}{2pc_s^2}$ ;
3.  $\| I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}} \|^2 \leq c_f$ ;
4.  $\mathbb{E} \| W_t^{\mathbf{u}} \|^2 \leq p\sigma_w^2$ ,  $\mathbb{E} \| W_t^{\mathbf{u}} \|^4 \leq p\sigma_4$ ;
5. *The unknown noise*

$$v_t^i = 2(I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}}) [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} + \| C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} \|^2$$

is bounded in the mean square sense,  $\mathbb{E}(v_t^i)^2 \leq c_v^2$ ;

6. *The successive differences  $\tilde{v}_k^i = v_{2k}^i - v_{2k-1}^i$  of noise are bounded:  $|\tilde{v}_k^i| \leq c_{\tilde{v}} < \infty$  or  $\mathbb{E}(\tilde{v}_k^i)^2 \leq c_{\tilde{v}}^2$ , if the sequence  $\{\tilde{v}_k^i\}$  is random.*

*Proof.*

1. If Assumption 2.1 holds, then  $\| C_t^{\mathbf{u}\top} \|^2 \leq 2pc_s^2$  и  $\| C_t^{\mathbf{u}\top} \|^4 \leq 4p^2c_s^4$ .
2. If Assumption 2.1 holds, then

$$\begin{aligned} 4p^2c_s^2 &\leq \| [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}] \|^2 \leq 4p^2c_s^2, \\ \frac{1}{4p^2c_s^2} &\leq \| [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' \|^2 \leq \frac{1}{4p^2c_s^2}, \\ \frac{2pc_s^2}{4p^2c_s^2} &\leq \| [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} \|^2 \leq \frac{2pc_s^2}{4p^2c_s^2}, \\ &\| [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} \|^2 \leq \frac{c_s^2}{2pc_s^2}. \end{aligned}$$

3. If Assumption 2.2 holds, then  $\| I_t^{\mathbf{u}} \mathbf{x}_t^{l,i} - H_t^{\mathbf{u}} \|^2 \leq c_f$ .
4. If Assumption 2.3 holds, then  $\mathbb{E} \| \bar{w}_t^q(\mathbf{u}) \|^2 \leq \sigma_w^2$ ,  $\mathbb{E} \| \bar{w}_t^q(\mathbf{u}) \|^4 \leq \sigma_4$  and  $\mathbb{E} \| W_t^{\mathbf{u}} \|^2 \leq p\sigma_w^2$ ,  $\mathbb{E} \| W_t^{\mathbf{u}} \|^4 \leq p\sigma_4$ .
5. If Assumptions 2.1-2.3 hold, then

$$\mathbb{E} v_t^i = 2(I_t^{\mathbf{u}} x_t^l - H_t^{\mathbf{u}}) [C_t^{\mathbf{u}\top} C_t^{\mathbf{u}}]' C_t^{\mathbf{u}\top} \mathbb{E} W_t^{\mathbf{u}} + \mathbb{E} \| C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} \|^2 = \mathbb{E} \| C_t^{\mathbf{u}\top} W_t^{\mathbf{u}} \|^2,$$

$$\begin{aligned}
\mathbb{E}(v_t^i)^2 &= 4\mathbb{E}\|(I_t^{\mathbf{u}}x_t^l - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top}C_t^{\mathbf{u}}]'C_t^{\mathbf{u}\top}W_t^{\mathbf{u}}\|^2 + \\
&4(I_t^{\mathbf{u}}x_t^l - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top}C_t^{\mathbf{u}}]'C_t^{\mathbf{u}\top}\mathbb{E}\left(W_t^{\mathbf{u}}\|C_t^{\mathbf{u}\top}W_t^{\mathbf{u}}\|^2\right) + \mathbb{E}\|C_t^{\mathbf{u}\top}W_t^{\mathbf{u}}\|^4 = \\
&4\mathbb{E}\|(I_t^{\mathbf{u}}x_t^l - H_t^{\mathbf{u}})[C_t^{\mathbf{u}\top}C_t^{\mathbf{u}}]'C_t^{\mathbf{u}\top}W_t^{\mathbf{u}}\|^2 + \mathbb{E}\|C_t^{\mathbf{u}\top}W_t^{\mathbf{u}}\|^4 \leq \\
&16p^3c_f c_S^2\sigma_w^2 + 4p^3c_S^2\sigma_4.
\end{aligned}$$

Thus, the noise  $v_t^i$  is not centered and  $\mathbb{E}(v_t^i)^2 \leq c_v^2$ , where  $c_v^2 = 16p^3c_f c_S^2\sigma_w^2 + 4p^3c_S^2\sigma_4$ . In other words,  $v_t^i$  is an *unknown-but-bounded noise*.

6. If Assumptions 2.1-2.3 hold, then

$$\mathbb{E}(\tilde{v}_k^i)^2 = \mathbb{E}(v_{2k}^i - v_{2k}^i)^2 \leq c_v^2,$$

where  $c_v^2 = 2c_v^2 = 8pc_f c_c\sigma_w^2 + 2pc_s\sigma_4$ .

□

## 1.3 Randomized and Multi-agent Approaches

Randomization has proven to be an effective tool for dealing with uncertainties in solving problems [12], which often arise in data collection using sensor networks. Moreover, it significantly reduces the dimensionality of the problem [111]. Multi-agent systems, in turn, are an approach that develops at the intersection of distributed computing and emergent intelligence, allowing for the inclusion of communication constraints in the described network model for observing moving objects.

The development and availability of computing technology have influenced classical areas of mathematical statistics, contributing to the development and preferential use of repeated estimation schemes. The relatively new approach to solving estimation and optimization problems in adverse conditions (e.g., in the presence of degenerate sequences of observations) is based on the use of trial perturbations [12]. If it is possible to introduce a new perturbation through the input channels of the system or algorithm with specified properties by the experimenter or well-known statistical properties, it can be used to “enrich” the information in the observation channel. Sometimes, a measurable random process present in the system can play the role of a trial perturbation. In control systems, they can be added through a control channel, while in other cases, a randomized experimental design can serve as a trial intervention. Studying the updated system with a trial perturbation, even using traditional methods, often leads to encouraging results regarding the convergence and applicability of new algorithms. One remarkable property of such algorithms is the preservation of estimates under “almost arbitrary” perturbations. Algorithms in which one or more steps are based on random rule selection are called randomized algorithms.

The Manhattan Project [110] played an important role in the development of computers. In the 18th century, accurate predictions of the return of comets and eclipses were made through paper calculations. However, until the 1940s, the choice of office equipment, such as tabulating machines, card

sorters, and simple mechanical calculators, available to assist humans, was limited. Within the Manhattan Project at Los Alamos, where scientists and engineers worked, a high degree of calculation accuracy was required, leading to the use of analog computers, including IBM punched-card computers. This shift in computer development and the possibility of accurate calculations led to the emergence of the Monte Carlo method [82], named after the city of Monte Carlo in the Principality of Monaco, where roulette served as a popular random number generator. Thus, the Manhattan Project and the Monte Carlo method became important stages in the development of computer technology and the application of random numbers in computations.

There is a class of problems whose complexity grows exponentially with the dimensionality of the problem. Some problems allow finding algorithms whose complexity grows slower, but there are many problems for which this is impossible, such as computing the volume of a convex body in an  $n$ -dimensional Euclidean space or computing  $n$ -dimensional integrals. In such cases, the Monte Carlo method is the only way to obtain sufficiently accurate results in a reasonable amount of time. Modern research is mainly focused on developing efficient Monte Carlo algorithms for various physical, chemical, and social processes, as well as for parallel computing systems. The Monte Carlo method has also found development in quantum theory. There is a quantum Monte Carlo method [50], which is widely used in studying complex molecules and solids.

At the Faculty of Mathematics and Mechanics of St. Petersburg State University, Sergey Mikhailovich Ermakov has been working on the Monte Carlo method since 1956 [6]. His research interests include the theory of the Monte Carlo method, multidimensional integration, the study of pseudo-random number generators, as well as probabilistic solutions of linear and nonlinear integral equations. His work has contributed to the formalization of this computational method, transforming it from a set of semi-empirical techniques into a well-defined area of mathematics.

Many practical problems that arise in various fields can be formulated as

root-finding problems or minimization problems. However, finding analytical solutions to such problems is often impossible due to the complexity of the functions or constraints involved. In such cases, random search methods are considered as an alternative approach that can be applied to find approximate solutions. The random search method is based on the iterative construction of a sequence of solution estimates. At each step, a new estimate is chosen by shifting the previous estimate in a randomly selected direction. This approach is an extension of the trial and error method, where the solution is sought randomly, and if it satisfies the specified criteria, it is accepted; otherwise, it is rejected in order to continue the search for a new possibility. The main idea is to use randomness to explore different regions of the parameter space, including the desired solution. The random search method has found wide application in various fields, including function optimization, computer simulation, and statistics. This method allows for exploring the parameter space, taking into account randomness, and achieving optimal or approximate solutions to the problem. A detailed description of the random search method can be found in the book “Statistical Methods for Search” by Rastrigin [20], which presents the theoretical foundations of the method, its various variants, and examples of its application to practical problems.

One example of a randomized algorithm that has received significant attention is the genetic algorithm for optimization. The foundations of this algorithm were laid by Nils Aall Barricelli, who in 1954 conducted early work on simulating evolution on computers at the Institute for Advanced Study in Princeton University [39]. In 1957, Australian geneticist Alex Fraser began publishing a series of papers on modeling artificial selection among organisms with multiple measurable characteristics. Fraser’s work served as a breakthrough in the development of computer models of evolutionary processes and methods. Over time, with growing interest and advancements in computing power, genetic algorithms found practical applications in various fields. As personal computers developed and their computational capabilities significantly increased, it became feasible to implement genetic algorithms for solving complex optimization problems. Genetic algorithms simulate natu-

ral selection and genetics, using techniques such as selection, crossover, and mutation to explore the solution space and converge towards optimal or near-optimal solutions. Genetic algorithms have demonstrated their effectiveness in a wide range of applications, including optimization problems in engineering, finance, logistics, and artificial intelligence [40].

The Simulated Annealing method is an optimization method based on ordered random search. The history of this method dates back to 1953 when N. Metropolis developed an algorithm for simulating the attainment of equilibrium in a system with multiple degrees of freedom at a given temperature [74]. Then, in the early 1980s, S. Kirkpatrick proposed the idea of using this algorithm to solve various optimization problems beyond the modeling of physical systems [57]. One of the main advantages of the Simulated Annealing method is its ability to avoid local minima of the optimized function and continue searching for the global minimum.

Tabu Search, also known as taboo search, is a metaheuristic search algorithm that utilizes local search methods to solve mathematical optimization problems. This algorithm was created by Fred W. Glover in 1986 [62]. The idea is that local search examines a potential solution to the problem and its immediate neighbors (i.e., solutions that differ only in small details) in the hope of finding an improved solution. However, local search methods can get stuck in suboptimal regions or plateaus where many solutions are equally good. Tabu Search enhances the performance of local search by relaxing its main rule. At each step, a deterioration can be accepted if there is no improvement (similar to the situation when the search gets stuck at a local minimum). Additionally, tabus are introduced to prevent revisiting already visited solutions. When implementing Tabu Search, structures are used to describe visited solutions or user-defined sets of rules. If a potential solution has been visited within a short period of time or violates a rule, it is marked as “tabu” to prevent the algorithm from reconsidering that solution.

The Gossip protocol, associated with randomized algorithms, stems from the study “Epidemic Algorithms for Replicated Database Maintenance” (1987)



that described algorithms for epidemic replication [56]. Since then, the application of such a protocol has attracted significant interest in the field of computing, and the first practical implementations were observed in computer network routing systems that preceded the Internet. Gossip protocols are decentralized information exchange protocols based on the idea of spreading information through random links between nodes in the system. In such protocols, nodes transmit information to each other, relying on random contacts or randomly selected neighbors. This allows for efficient information dissemination in the network and the discovery of global properties or problem-solving based on local information exchanges. Gossip protocols have found wide application in various fields. In communication networks, they have been used for packet routing, error detection and correction, resource management, and collective decision-making. Gossip protocols have also been extensively studied in the field of social networks, where they have been used to model information diffusion and influence within communities. The advantages of gossip protocols lie in their resilience to failures and dynamic changes in the system. With random links and information propagation through neighbors, they can effectively adapt to changes in the network and continue to operate in the presence of individual node failures. Moreover, gossip protocols do not require global knowledge of the system state and can operate under conditions of limited information availability. Overall, gossip protocols represent an important class of randomized algorithms widely used for information exchange and coordination in decentralized systems.

Any measurements obtained from sensors are always distorted by noise. In the literature, disturbances with certain known normal statistical properties are usually considered [42, 77, 78]. The problem of controlling a sensor network becomes significantly more challenging when considering arbitrary uncertainties external to the system, such as limited but otherwise unknown uncertainties [64]. Therefore, research in the field of stochastic optimization, aimed at reducing the required a priori knowledge of uncertainties in the problem for the functionality of optimization algorithms, is of great interest [63].

Stochastic approximation has a wide range of applications today in areas such as adaptive signal processing, adaptive resource allocation in communication networks, system identification, adaptive control, and others. In the works of B. T. Polyak [17], J. C. Spall [112, 113], V. S. Borkar [43], A. B. Tsybakov [17], H. Kushner and G. G. Yin [76], stochastic approximation is used with time-varying step sizes converging to zero. There is an increasing use of stochastic approximation algorithms for optimizing time-varying objective functions. In such problems of parameter tracking, a sufficiently small but constant step size is often used. D. P. Derevitsky and A. L. Fradkov in [4, 5], in the analysis of the dynamics of adaptation algorithms based on the construction of approximate averaged models, justified the possibility of using stochastic approximation algorithms with non-increasing step sizes converging to zero. Later, the study of optimization of time-varying objective functions was considered in the works of O. N. Granichin [10, 11], N. O. Amelin [38, 64], J. C. Spall [122], V. S. Borkar [43].

Another randomized algorithm is the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm [111]. It is capable of solving optimization problems [54] in the presence of arbitrary but bounded disturbances and time-varying system parameters [10, 11, 17]. These uncertainties can be non-random, and even if they are random, their statistical characteristics are not necessarily known [108]. The main feature of SPSA is that it uses simultaneous perturbations at the input and, instead of estimating the gradient as in traditional optimization methods, SPSA estimates a gradient approximation based on random perturbations of the function and the corresponding changes in the objective function. SPSA has several advantages. Firstly, it effectively operates in conditions of bounded but generally unknown disturbances since it does not require information about their exact properties. Secondly, SPSA allows for computational cost reduction as the gradient estimation is based only on two function values obtained with random perturbations.

The original idea of the ant colony optimization algorithm comes from observing ants in their search for the shortest path from a colony to a food source [53].

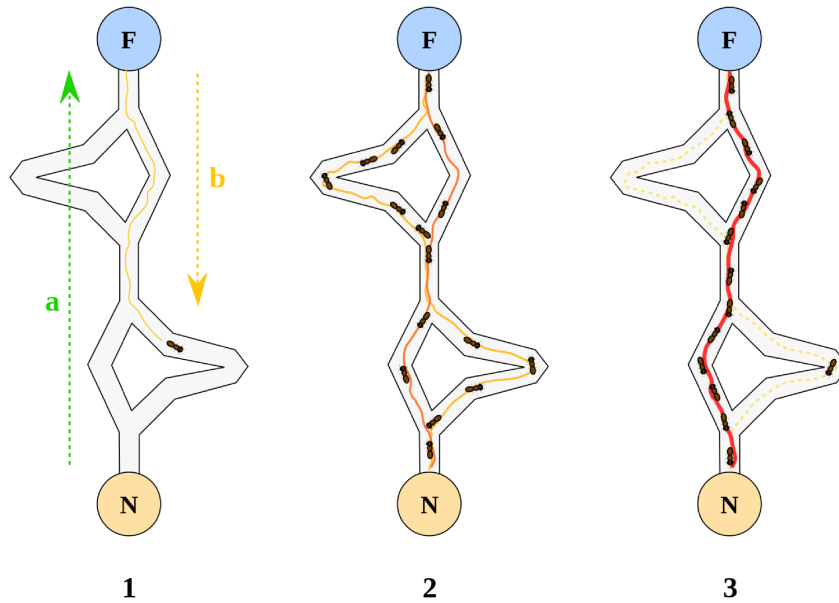


Figure 1.1: Ant Colony Optimization

The primary goal of ants is to find food. Thus, the first ant finds the food source (F) by any means (a), then returns to the nest (N) leaving a trail of pheromones (b) (Figure 1.1). Then, ants choose one of several possible paths, reinforcing and making it more attractive. Ants choose the shortest path as pheromones on longer paths evaporate faster. In experiments where there are two paths of unequal length from the colony to the food source, biologists noticed that ants generally choose the shortest path [53]. The model of such behavior is as follows.

The ant colony functions through a mechanism known as stigmergy, which involves information exchange using pheromones. When an ant sets out in search of food, it leaves a trail of pheromones on its return path to the nest. These pheromones attract other ants in the vicinity and increase the probability of following the same path. Upon returning to the nest, these ants reinforce the pheromone trail. Over time, if there are multiple paths, more ants will travel along the shorter path, making it more attractive and leading to the disappearance of longer paths due to pheromone evaporation. This method of communication using the environment demonstrates self-organization [55] and is applicable to various social animals, such as termites.

The ant colony system reflects both positive and negative feedback. The presence of other ants reinforces the pheromone trail (positive feedback), while pheromone evaporation acts as negative feedback. Without feedback, it would be impossible to choose a path if the amount of pheromones remained constant over time. However, small fluctuations, amplified by feedback, lead to the reinforcement of one path, ultimately stabilizing the system toward choosing the shortest path. It should be noted that there are modern variations of the ant colony algorithm, including the elite ant system, Max-Min Ant System (MMAS), ranked ant system, and long orthogonal ant colony.

Many algorithms for multi-agent coordination, currently used in domains such as blockchain [114], mobile robotics [18,55], and emergent intelligence [55, 100], are inspired by the dynamics of natural, social, and economic systems [41, 44, 93].

The foundation of a multi-agent system consists of intelligent agents that possess autonomy (the ability to work towards their goals without human intervention or interference from other system components) and proactive behavior (agents consider not only online information from the external world but also the history of their actions and the states of the external environment reflected in their current internal state, such as messages received from other agents) [8]. In a multi-agent architecture, dynamically updated information obtained by an agent from the external world, as well as locally available information from a limited number of neighbors, is processed by the agent itself or transmitted to another agent. This significantly reduces resource and time costs for communication in the network, as well as the time required for processing and decision-making by the entire system (if there is a central component in the system). Moreover, multi-agent systems are also flexible, allowing the system to be extended and modified without rewriting a significant portion of the program [33, 103].

Recent research [99] has shown that bacteria are capable of interacting with each other, fundamentally changing our understanding of the behavior of simple organisms inhabiting the world around us. Bacteria use molecular

signaling to measure the density of their population. The term “quorum sensing” (QS) refers to the process by which a bacterial cell can sense the activity of other bacteria through population concentration. Different species of bacteria coexisting in the same environment employ diverse signaling molecules, enabling them to interact with other bacterial species. Currently, these quorum sensing systems are actively studied for various categories of bacteria. There have been significant advancements in understanding the genetics, genomics, biochemistry, and diversity of QS signaling molecules, including the interplay between quorum sensing and the social behavior of bacteria. The behavior and evolution of such bacterial communities are considered natural examples of multi-agent systems since the interaction between bacteria occurs locally, and population density measurement can be achieved without gathering all the data in a central information processing center. Bacteria successfully solve global tasks, such as population density measurement, using only local interactions, which serves as an example of a multi-agent algorithm.

Another important example of self-organization is DNA computing [35]. To explain the concept of DNA computing, let’s consider the well-known combinatorial problem of finding a Hamiltonian path in a graph. This problem involves finding a path in an undirected or directed graph that visits each vertex exactly once. DNA computing, based on the principles of self-organization inherent in nature [105], can solve such problems with linear time complexity.

Recall that DNA molecules typically form a double helix composed of nucleotides containing a phosphate group, a sugar group, and a nitrogenous base. There are four different nucleotides: adenine (“A”), thymine (“T”), guanine (“G”), and cytosine (“C”). In the helix, nucleotides pair up according to the fundamental *complementarity rule*: “A

” always pairs with “T,” and “G” always pairs with “C” [120]. Additionally, through heating, the double-stranded DNA can be separated into two individual strands, or the reverse process (*ligation*) can be performed using

a special enzyme called *ligase*.

Consider a directed graph  $G = (\mathcal{N}, E)$  with  $|\mathcal{N}| = n$  nodes. Let random DNA sequences consisting of twenty nucleotides represent the nodes (vertices). An edge is associated with a DNA sequence obtained by aligning the complementary sequence of the second half of the initial node and the complementary sequence of the first half of the target node. For example, if the first node has the DNA code:

$$\begin{aligned} &TCAGTACCAG \quad TACAGTCACA \\ &\text{complementary sequence :} \\ &(AGTCATGGTC \quad ATGTCAGTGT), \end{aligned}$$

where A represents adenine, T represents thymine, G represents guanine, and C represents cytosine, and the second node has the DNA code:

$$\begin{aligned} &TAGGTATGCT \quad CAGATAAAGG \\ &\text{complementary sequence :} \\ &(ATCCATACGA \quad GTCTATTTCC), \end{aligned}$$

and there is an edge from the first node to the second, the edge is encoded as:

$$ATGTCAGTGT \quad ATCCATACGA.$$

This procedure is repeated for each available edge [26,27]. It should be noted that the directions of DNA strands are omitted for simplicity.

Furthermore, all the described DNA codes are synthesized and mixed in a single test tube to undergo ligation (a reaction that joins molecules based on complementarity). As a result, all possible paths are created simultaneously and in parallel. An example of such a path (molecules) is shown in Figure 1.2.

In this process, an analogy can be drawn with multi-agent technologies. By forming strands and links into complex DNA molecules, as if they were agents in a cascading process, nucleotides provide a solution to the problem. Subsequent steps involve checking whether the molecules encoding Hamilto-

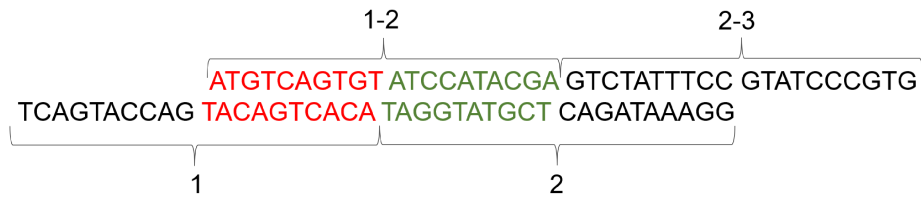


Figure 1.2: Example of a random molecule after ligation reaction

nian paths are present inside the test tube or not. The time complexity of DNA computing results from local interactions and clustering, which increase linearly with the number of graph nodes [23–25]. Furthermore, in [106], this characteristic was compared to the branch and bound method, revealing that DNA computing performs significantly better for adjacency matrix sparsity parameters of 0.85, 0.9, 0.8, and increasing numbers of graph nodes (37, 43, 45, respectively). In [32], the traveling salesman problem was also addressed using DNA computing.

Multi-agent algorithms also emerge in social networks. In [93], models of social processes are described that were developed concurrently with the theory of multi-agent systems.

The Local Voting Protocol is another example of a multi-agent approach that enables decentralized decision-making and coordination among agents in a network [37,38,68]. According to this protocol, tasks arriving at network nodes can be evenly distributed among all nodes through their communication (data exchange between neighboring nodes). It finds wide applications in various domains. In the context of aircraft wing control during turbulence [92], the Local Voting Protocol demonstrates its effectiveness in mitigating turbulence effects. By employing a network of small rotating blocks equipped with pressure sensors, the protocol enables local feedback and information exchange between neighboring blocks. This allows the blocks to adjust their positions based on pressure comparisons, ultimately reaching a consensus state where the integrated pressure forces become equal for all blocks. This showcases the ability of multi-agent systems to adapt and self-organize to solve complex tasks related to turbulence. Similarly, the Local

Voting Protocol has shown effectiveness in load balancing in large-scale networks [38,115]. By redistributing tasks among agents based on local information exchange, the protocol aims to achieve uniform task distribution. This optimizes the overall task completion time, reduces delays, and enhances network efficiency.

One possible solution to the problem of sensor network control considering topological constraints is the distribution of observed objects among sensors, as described in [102]. However, this method has significant drawbacks, including high computational complexity and the need for prior knowledge of which object is closer to which sensor. An alternative approach involves randomization in selecting observed objects and using a randomized and multi-agent approach within the sensor network to select neighboring sensors (for data transmission between them) [33,103].

The aforementioned examples confirm the relevance and necessity of using randomized and multi-agent approaches in sensor network control, which will be further described.

## 1.4 Sensor Network Control Problem

There are various ways to estimate the locations of moving targets based on sensor measurements. Traditionally, centralized algorithms are used, where it is assumed that all sensors transmit the measured distances to the targets to a central data processing center for estimating the unknown vector (see Fig. 1.3 left). This approach works well when there is a small number of observed targets. However, as the number of targets increases, data collection, transmission to the center, and processing become bottlenecks [58,85]. An alternative approach is to use a distributed approach, where sensors are treated as autonomous computational nodes (agents) capable of processing locally available information from a limited number of neighbors and exchanging data with them. This can significantly reduce communication costs in the network as well as the time required for processing and decision-making



by the entire system (see Fig. 1.3 right).

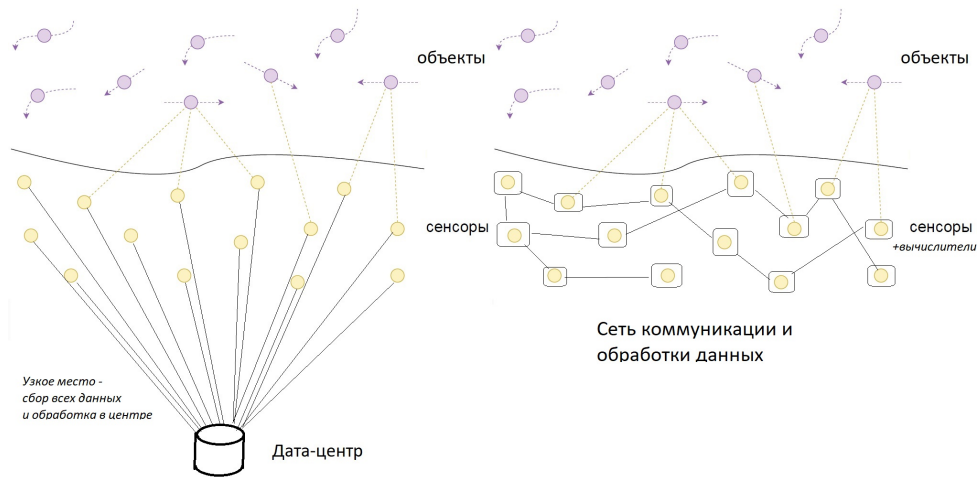


Figure 1.3: Centralized (left) and distributed (right) approaches for controlling a sensor network to track moving targets

In the distributed approach, one important task is to select neighbors for communication. On one hand, a sensor cannot constantly exchange data with all network nodes, as it would lead to resource-intensive data processing and would not differ much from the centralized approach. Additionally, there are often technical limitations on the number of communication channels (communication constraints) in practice. On the other hand, communicating only with a small, fixed number of neighboring sensors may lead to the loss of important information about the entire system due to limited sensor coverage or even result in a degenerate estimation problem. Therefore, optimizing the selection of neighbors for each sensor is a relevant task.

### 1.4.1 Optimization of Target and Sensor Network Selection

One way to optimize the allocation of targets among sensors, taking into account the load on each sensor and the accuracy of target localization, is presented in [102].

Let  $\bar{\mathbf{r}}_t^{l,i}$  be the measurement obtained by sensor  $i$  for target  $l$ . Assume that

such an estimate looks as follows:  $\bar{\mathbf{r}}_t^{l,i} = \mathbf{r}_t^l + \xi_t^{l,i}$ , where  $\mathbf{r}_t^l$  is the true location of target  $l$  at time  $t$ , and  $\xi_t^{l,i}$  is an unknown-but-bounded disturbance, which can be described by an ellipsoid around the estimate  $\bar{\mathbf{r}}_t^{l,i}$ . In this case, it can be assumed that the ellipsoid guarantees the containment of the true value  $\mathbf{r}_t^l$ . This means that if there exists a bounding ellipsoid, the estimation error  $\|\mathbf{r}_t^l - \bar{\mathbf{r}}_t^{l,i}\|$  does not exceed its boundaries, and the value  $\mathbf{r}_t^l$  lies inside this ellipsoid:

$$\mathcal{E}_t^{l,i} = \{\Xi_t^{l,i}x + \bar{\mathbf{r}}_t^{l,i} \mid \|x\| \leq 1\},$$

where  $\Xi_t^{l,i}$  is the matrix describing the shape of the ellipsoid, and  $\bar{\mathbf{r}}_t^{l,i}$  is the center of the ellipsoid. If  $\Xi_t^{l,i}(\Xi_t^{l,i})^T > 0$ , then the previous expression is equivalent to:

$$\mathcal{E}_t^{l,i} = \{x : (x - \bar{\mathbf{r}}_t^{l,i})^T (\Xi_t^{l,i})^{-2} (x - \bar{\mathbf{r}}_t^{l,i}) \leq 1\}. \quad (1.6)$$

Since we are considering a sensor network, for each target  $l$ , there is a set of estimates  $\tilde{\mathbf{r}}_t^l = \{\bar{\mathbf{r}}_t^{l,1}, \dots, \bar{\mathbf{r}}_t^{l,n}\}$  and their corresponding ellipsoids  $\tilde{\mathcal{E}}_t^{l,i} = \{\mathcal{E}_t^{l,1}, \dots, \mathcal{E}_t^{l,n}\}$ . Since the ellipsoids guaranteedly contain the true value  $\mathbf{r}_t^l$ , it is possible to reduce the uncertainty of the target location by finding the intersection region of the ellipsoids obtained for different sensors. The smaller this intersection region, the more accurately the true vector  $\mathbf{r}_t^l$  can be determined. However, it is worth noting that finding the volume of such an intersection can be a challenging task. One possible solution is to approximate the intersection region with an ellipsoid [81].

Let  $\hat{\mathcal{E}}_t^l$  be the ellipsoid that approximates the intersection of the ellipsoids obtained for target  $l$ , and  $\bar{\mathcal{E}}_t = \{\hat{\mathcal{E}}^1, \dots, \hat{\mathcal{E}}^m\}$  be the set of such ellipsoids obtained for each target. The goal is to find a set  $\bar{\mathcal{E}}_t$  such that the sum of the volumes of the ellipsoids is minimized:

$$\bar{\mathcal{E}}_t = \operatorname{argmin}_{\bar{\mathcal{E}}_t} \sum_{l \in \mathcal{M}} \operatorname{vol}(\hat{\mathcal{E}}_t^l).$$

To reduce the processing load and communication, it is also necessary to minimize the number of sensors tracking each target. Let  $P_t$  be the target allocation matrix among sensors. The elements of this matrix indicate whether sensor  $i$  is tracking target  $l$  or not. Each matrix  $P_t$  corresponds to its set of ellipsoids  $\bar{\mathcal{E}}_t$ . Optimizing the target allocation among sensors corresponds to a sparse matrix  $P_t$ , as reducing the number of sensors tracking the same target leads to lower communication and data processing costs. A typical approach to achieving sparsity is by regularizing the  $l_0$ -quasinorm. However, since optimizing the  $l_0$ -quasinorm is a non-convex and NP-hard problem, [91] proposes using the  $l_1$ -norm, which provides the best approximation of the sparse solution. As a result, the problem of minimizing the target allocation matrix among sensors can be formulated as follows:

$$P_t = \underset{P_t}{\operatorname{argmin}} \left( \sum_{l \in \mathcal{M}} \operatorname{vol}(\hat{\mathcal{E}}_t^l) + \alpha \sum_{l \in \mathcal{M}} \|P^{(l,\cdot)}_t\|_1 + \beta \sum_{i \in \mathcal{N}} \|P_t^{(\cdot,i)}\|_1 \right), \quad (1.7)$$

where  $\alpha$  is a parameter controlling the number of sensors tracking a single target,  $\beta$  is a parameter controlling the number of targets tracked by a single sensor,  $P_t^{(l,\cdot)}$  is the  $l$ -th row of the matrix  $P_t$ ,  $P_t^{(\cdot,i)}$  is the  $i$ -th column of the matrix  $P_t$ , and  $\|\cdot\|_1$  is the  $l_1$ -norm.

In [102], the optimization problem (1.7) was reformulated in terms of linear matrix inequalities and solved using the CVX package [66].

## 1.4.2 Randomization of Topology

The previous subsection describes a method for target and sensor network selection. However, this approach has drawbacks such as the long computation time using the CVX package, and the need to solve the resource allocation optimization problem again when targets move significantly or sensors are deployed or removed, which can be a computationally intensive process. An alternative approach is to use topology randomization [19].

Suppose the following *topological constraints* are given: each sensor  $i \in \mathcal{N}$

is able to measure the distance to only one target and collect data from  $p$  neighbors at each time  $t$ , where  $p$  is a predetermined value. In practice, due to certain limitations, the number of available communication channels is usually less than or equal to the dimension of the space ( $p \leq d$ ).

To satisfy the communication constraints, topology randomization can be used, similar to the gossip protocol [52]. The idea is to randomly select communication links between sensors at each time step, effectively *randomizing the communication graph*.

More formally, to satisfy the topological constraints with a maximum of  $p$  communication links for each sensor, the topological graph is randomized, denoted as  $\mathcal{G}_A$ , at each time step  $t$ . This is achieved by selecting a randomly chosen subgraph  $\mathcal{G}_{B_t} \subset \mathcal{G}_A$  with an adjacency matrix  $B_t = [b_t^{i,j}]$ , where each row of  $B_t$  contains at most  $p$  non-zero elements. In other words, the graph  $\mathcal{G}_A$  represents all possible connections between sensors, while the graph  $\mathcal{G}_{B_t}$  represents the specific connections between sensors at a given time  $t$ . For example, if the adjacency matrix of the graph  $\mathcal{G}_A$  is complete (all sensors can communicate with each other):

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

then for  $p = 1$ , the adjacency matrix of the graph  $\mathcal{G}_{B_t}$  at time  $t = t_1$  can be:

$$B_{t_1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

and at time  $t = t_2$ , it can be:

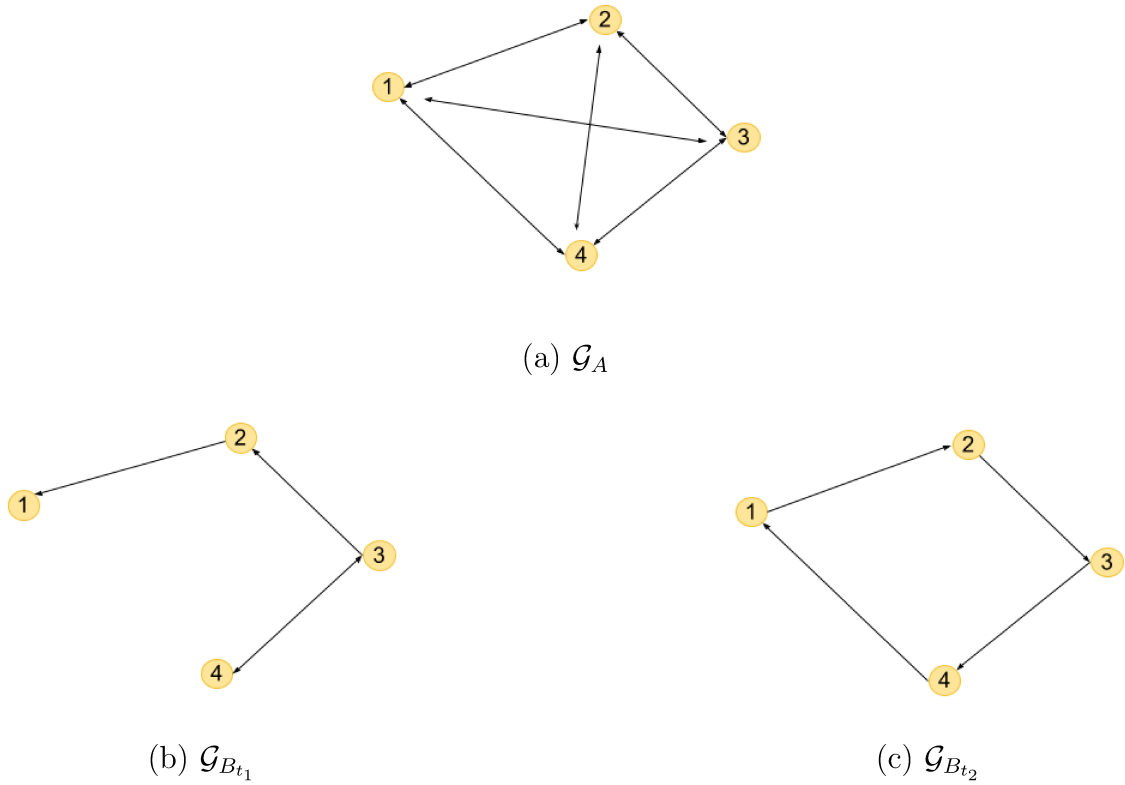


Figure 1.4: Examples of topology

$$B_{t_1} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

In practice, this randomization means that each sensor receives data from only a small number of  $p$  other sensors, and these sensors are constantly changing. In the example described above, at time  $t = t_1$ , the first sensor receives data from the second sensor, while at time  $t = t_2$ , the same sensor receives data from the fourth sensor. An example of graphs representing a sensor network is shown in Fig. 1.4. It is worth noting that neighboring agents for communication are independently and uniformly selected, similar to the gossip protocol [52]. Essentially, the topological graph  $\mathcal{G}_A$  is randomized in the same way as in [51].

The assumption on the topology of the randomized network is introduced.

**A s s u m p t i o n 3.** a) For all  $i \in \mathcal{N}, j \in \mathcal{N}_t^i$ , the weights  $b_t^{i,j}$  are independent random variables with mean values:  $\mathbb{E}b_t^{i,j} = b_{av}^{i,j}$ , bounded variances:  $\mathbb{E}\|\mathcal{L}(B_t) - \mathcal{L}(B_{av})\|^2 \leq \sigma_B^2$ , where  $B_{av} = [b_{av}^{i,j}]$ , and bounded covariance matrices:  $\mathbb{E}[(\mathcal{L}(B_t) - \mathcal{L}(B_{av}))(\mathcal{L}(B_t) - \mathcal{L}(B_{av}))^T] \leq Q_B$ , with  $b_{\max}$  being the maximum element of  $Q_B$ .  
b) The graph  $\mathcal{G}_{B_{av}}$  is strongly connected.

In the described approach, when randomizing the topology, the selection of the previously described set  $\mathbf{u} \in \mathcal{U}_t$  is random. We define such a set as

$$\mathbf{u} = (i, J, l),$$

where  $i \in \mathcal{N}, J \subset \bar{\mathcal{N}}_t^i, l \in \mathcal{M}$ . Referring back to the example above, one of such sets for the first sensor could be represented as follows at  $t = t_1$ , taking into account that the first and second sensors consistently observe the fifth target:  $\mathbf{u} = (1, 2, 5)$ .

Let  $\text{Pr}_l(\mathbf{x})$  denote the projection of the components of vector  $\mathbf{x}$  corresponding to observations of target  $l$ .

The target tracking problem with a sensor network can be reformulated as the following stochastic optimization problem: at each time step  $t$ , we

need to find a common estimate of the target locations  $\hat{\theta}_t = \begin{pmatrix} \hat{\mathbf{r}}_t^{1,1} \\ \vdots \\ \hat{\mathbf{r}}_t^{m,n} \end{pmatrix} \in \mathbb{R}^{nmd}$

that minimizes the loss function

$$\hat{\theta}_t = \underset{\hat{\theta}_t}{\text{argmin}} \mathbb{E}_{\mathcal{F}_{t-1}} \sum_{i \in \mathcal{N}} F_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) = \underset{\hat{\theta}_t}{\text{argmin}} \sum_{i \in \mathcal{N}} \bar{F}_t^i(\hat{\theta}_t), \quad (1.8)$$

where

$$\bar{F}_t^i(\hat{\theta}_t) = \mathbb{E}_{\mathcal{F}_{t-1}} \{f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) \mid i(\mathbf{u}) = i\}$$

over the noisy measurements of the objective function

$$y_t^i = f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i)) + v_t^i,$$

where  $v_t^i$  is unknown-but-bounded noise. Under uniform sampling  $\mathbf{u}_t^i \in \mathcal{U}_t|_{i(\mathbf{u})=i}$ , the solution of problem (1.8) corresponds to the previously stated objective (1.4).

To solve such a problem, the zeroth-order method is proposed. This class of methods utilizes measurements of the objective function and does not involve computing gradients or higher-order derivatives.

## 1.5 Summary

In the first chapter, a network model for observing moving objects based on a randomized approach was developed. Assumptions were formulated regarding the velocity of targets (Assumption 1), sensors, observation points, and measurement noise (Assumption 2), as well as about the topology of the randomized network (Assumption 3). The optimization problem was defined to generate consistent estimates of the locations of all targets at each time step, taking into account noisy measurements of the objective function: at each time  $t$ , it is necessary to find a common estimate of the target locations

$$\hat{\theta}_t = \begin{pmatrix} \hat{\mathbf{r}}_t^{1,1} \\ \vdots \\ \hat{\mathbf{r}}_t^{m,1} \\ \hat{\mathbf{r}}_t^{1,2} \\ \vdots \\ \hat{\mathbf{r}}_t^{m,n} \end{pmatrix} \in \mathbb{R}^{nmd}$$

that minimizes the loss function

$$\hat{\theta}_t = \underset{\hat{\theta}_t}{\operatorname{argmin}} \sum_{i \in \mathcal{N}} \bar{F}_t^i(\hat{\theta}_t),$$

where

$$\bar{F}_t^i(\hat{\theta}_t) = \mathbb{E}_{\mathcal{F}_{t-1}} \{ f_t^i(\mathbf{u}, \hat{\mathbf{r}}_t^{l,i}) \mid i = i(\mathbf{u}) \}$$

based on noisy measurements of the objective function

$$y_t^i = f_t^i(\mathbf{u}_t^i, \Pr_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i)) + v_t^i,$$

where  $v_t^i$  are unknown-but-bounded disturbances, and  $\mathbb{E}(v_t^i)^2 \leq c_v^2$ , where  $c_v$  is computed using constants from Lemma 1.



# Chapter 2. Distributed Randomized Stochastic Approximation Algorithm Combined with Local Voting Protocol

## 2.1 Optimization of Nonstationary Mean Risk Functional

In the work [108], a randomized stochastic optimization algorithm combined with a consensus algorithm was proposed to solve the optimization problem (1.8) for a general class of functions

$$f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i)).$$

It is worth noting that the work [108] considered the function  $f_{\xi_t}^i(\mathbf{x}_t^{l,i})$ , where  $\{\xi\}$ ,  $\xi_t \in \Xi$  is an uncontrolled deterministic or random sequence, and  $\Xi$  is a set. The function  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$  from Section 1 is a special case of the function  $f_{\xi_t}^i(\mathbf{x}_t^i)$ , where  $\xi_t$  consists of all vectors  $\mathbf{u}_t^i$  generated at time  $t$ . Let us present the algorithm from [108].

Let  $\Delta_k^i$ ,  $k = 1, 2, \dots, i \in \mathcal{N}$  be observed sequences of independent random vectors from  $\mathbb{R}^{md}$  with a Bernoulli distribution, where each component

independently takes values  $\pm \frac{1}{\sqrt{d}}$  with a probability of  $\frac{1}{2}$ . Such a sequence is usually called a *synchronized probing perturbation*. Let  $\mathbf{K}_k^i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d, k = 1, 2, \dots, i \in \mathcal{N}$  be a set of vector-valued functions. We will assume the following properties of vectors  $\xi_k^i$  and  $\Delta_k^i$  (as well as  $\mathbf{u}_k^i$  and  $\Delta_k^i$ ).

**A s s u m p t i o n 4.** For any  $i \in \mathcal{N}$ : a) the vectors  $\xi_k^i, \Delta_k^i, k = 1, 2, \dots, i \in \mathcal{N}$ , are mutually independent; b)  $\xi_k^i, \Delta_k^i, \mathbf{s}_{2k-1}^i, \mathbf{s}_{2k}^i$ , if they are random, are independent of the  $\sigma$ -algebra  $\mathcal{F}_{2k-2}$ ; c) if  $\tilde{v}_k^i$  is random, then the random vectors  $\xi_k^i, \Delta_k^i$  and the elements  $\mathbf{s}_{2k-1}^i, \mathbf{s}_{2k}^i, \tilde{v}_k^i$  are independent; d) the vectors  $\mathbf{u}_k^i, \Delta_k^i, k = 1, 2, \dots, i \in \mathcal{N}$ , are mutually independent; e)  $\mathbf{u}_k^i, \Delta_k^i$

,  $\mathbf{s}_{2k-1}^i, \mathbf{s}_{2k}^i$ , if they are random, are independent of the  $\sigma$ -algebra  $\mathcal{F}_{2k-2}$ ; f) if  $\tilde{v}_k^i$  is random, then the random vectors  $\mathbf{u}_k^i, \Delta_k^i$  and the elements  $\mathbf{s}_{2k-1}^i, \mathbf{s}_{2k}^i, \tilde{v}_k^i$  are independent; g)  $\mathbb{E}\|\Delta_k^i\|^2 \leq \sigma_\Delta^2, \mathbb{E}[\Delta_k^i(\Delta_k^i)^\top] \leq \sigma_\Delta^2 I_{md}$

Take a fixed non-random initial vector  $\hat{\theta}_0^i \in \mathbb{R}^{md}, i \in \mathcal{N}$ , a positive step size  $\alpha$ , a gain coefficient  $\gamma$ , and choose sequences  $\{\beta_k^+\}$  and  $\{\beta_k^-\}$  such that  $\beta_k = \beta_k^+ + \beta_k^- > 0$ . Assume that the function  $f_{\xi_k}^i(\mathbf{x}_k^i)$  is available for observation at each time step  $k$  with the unknown-but-bounded noise  $v_k^i$ :  $y_k^i = f_{\xi_k}^i(\mathbf{x}_k^i) + v_k^i$ . Consider an algorithm with two observations of distributed subfunctions  $f_{\xi_t}^i(\mathbf{x}_t^i)$  for each agent  $i \in \mathcal{N}$  to construct sequences of observation points  $\{\mathbf{x}_t^i\}$  and estimates  $\{\hat{\theta}_t^i\}$  of the common state vector of all targets.

$$\begin{cases} \mathbf{x}_{2k}^i = \hat{\theta}_{2k-2}^i + \beta_k^+ \Delta_k^i, \mathbf{x}_{2k-1}^i = \hat{\theta}_{2k-2}^i - \beta_k^- \Delta_k^i, \\ \hat{\theta}_{2k-1}^i = \hat{\theta}_{2k-2}^i, \\ \hat{\theta}_{2k}^i = \hat{\theta}_{2k-1}^i - \alpha \left[ \frac{y_{2k}^i - y_{2k-1}^i}{\beta_k} \mathbf{K}(\Delta_k^i) + \right. \\ \left. \gamma \sum_{j \in \mathcal{N}_{2k-1}^i} b_{2k-1}^{i,j} (\hat{\theta}_{2k-1}^i - \hat{\theta}_{2k-1}^j) \right], \end{cases} \quad (2.9)$$

Let's consider the last equation of algorithm (2.9): the first part is analogous to SPSA from [64], and the second part corresponds to the Local Voting

Protocol (LVP) from [38], where it was studied for stochastic networks in the context of load balancing. The SPSA part represents the stochastic gradient descent of subfunctions  $\bar{F}_t^i(\hat{\theta}_t)$ , and the LVP part is defined for each agent  $i$  as a weighted sum of the differences between the information about the current estimate  $\hat{\theta}_{2k-1}^i$  of agent  $i$  and the available information about the estimates of its neighbors.

The main result of the work [108] is the theorem on the convergence of the randomized stochastic optimization algorithm combined with the consensus algorithm to the set of possible values for the general class of functions  $f_{\xi_t}^i(\mathbf{x}_t^i)$ . Let's present the assumptions introduced in [108], and considering the specific form of the function  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i)) = \|I_t^{\mathbf{u}_t^i} \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i) - H_t^{\mathbf{u}_t^i}\|^2$ , we will check the conditions of the theorem for the network observation model of moving targets.

Condition 1 from [108]. The functions  $\bar{F}_t^i(\hat{\theta}_t)$  are convex and have a common minimum point  $\theta_t$ , as well as:

$$\forall \mathbf{x} \in \mathbb{R}^d \langle \mathbf{x} - \mathbf{r}_t^l, \mathbb{E}_{\mathcal{F}_{t-1}} \nabla f_{\xi_t}^i(\mathbf{x}) \rangle \geq 0.$$

Checking Condition 1:

$$\begin{aligned} \langle \mathbf{x} - \mathbf{r}_t^l, \mathbb{E}_{\mathcal{F}_{t-1}} \nabla f_t^i(\mathbf{u}_t^i, \mathbf{x}) \rangle &= 2\mathbb{E}_{\mathcal{F}_{t-1}} (\mathbf{x} - \mathbf{r}_t^l)^\top (I_t^{\mathbf{u}_t^i})^\top (I_t^{\mathbf{u}_t^i} \mathbf{x} - H_t^{\mathbf{u}_t^i}) = \\ 2\mathbb{E}_{\mathcal{F}_{t-1}} \left( I_t^{\mathbf{u}_t^i} (\mathbf{x} - \mathbf{r}_t^l) \right)^\top (I_t^{\mathbf{u}_t^i} \mathbf{x} - H_t^{\mathbf{u}_t^i}) &= 2(I_t^{\mathbf{u}_t^i} \mathbf{x} - H_t^{\mathbf{u}_t^i})^\top (I_t^{\mathbf{u}_t^i} \mathbf{x} - H_t^{\mathbf{u}_t^i}) \geq 0. \end{aligned}$$

Condition 2 from [108]. The gradient  $\nabla f_{\xi_t}^i(\mathbf{x})$  is Lipschitz continuous with a common constant  $M > 0$ :

$$\forall \mathbf{x}', \mathbf{x}'' \in \mathbb{R}^d \|\nabla f_{\xi_t}^i(\mathbf{x}') - \nabla f_{\xi_t}^i(\mathbf{x}'')\| \leq M \|\mathbf{x}' - \mathbf{x}''\|.$$

Checking Condition 2:

$$\|\nabla f_t^i(\mathbf{u}_t^i, \mathbf{x}') - \nabla f_t^i(\mathbf{u}_t^i, \mathbf{x}'')\| = 2\|(I_t^{\mathbf{u}_t^i})^\top I_t^{\mathbf{u}_t^i}(\mathbf{x}' - \mathbf{x}'')\| \leq M \|\mathbf{x}' - \mathbf{x}''\|,$$

where  $M = 2 \max_i \|(I_t^{\mathbf{u}^i})^\top I_t^{\mathbf{u}^i}\|$ .

Condition 3 from [108]. The gradient  $\nabla f_{\xi_t}^i(\mathbf{x})$  is uniformly bounded in the mean square sense at the minimum point  $\mathbf{r}_t^l$ :

$$\mathbb{E}\|\nabla f_{\xi_t}^i(\mathbf{r}_t^l)\|^2 \leq g_2^2,$$

$$\mathbb{E}\langle \nabla f_{\xi_t}^i(\mathbf{r}_t^l), \nabla f_{\xi_t}^i(\mathbf{r}_{t-1}^l) \rangle \leq g_2^2.$$

Checking Condition 3: since  $\nabla f_t^i(\mathbf{u}_t^i, \mathbf{r}_t^l) = 0$ , we have  $g_2 = 0$ .

Conditions 4-8 from [108] are automatically satisfied based on the previously introduced Assumptions 1-4.

## 2.2 Distributed Randomized Stochastic Optimization Algorithm Combined with Local Voting Protocol for Moving Target Estimation

The nature of the problem allows us to adapt (modify) algorithm (2.9) and restate the convergence theorem of the algorithm to the set of possible values in terms of the problem formulated in Section 1. Moreover, the refined formulations of the theorem allow us to derive a suboptimal step for the modified algorithm.

Let's consider the modified distributed randomized stochastic optimization algorithm combined with the local voting protocol for the developed observation model of moving targets [46]. It should be noted that, unlike in the work [108], a constant perturbation scale  $\beta > 0$  and  $\mathbf{K}_k^i(\Delta_k^i) = \Delta_k^i$  will be used here.

We choose observed sequences of independent random vectors  $\Delta_k^i \in \mathbb{R}^{md}$  with a Bernoulli distribution, where each component independently takes

values  $\pm \frac{1}{\sqrt{d}}$  with a probability of  $\frac{1}{2}$ , and also  $\mathbf{u}_k^i \in \mathcal{U}_t$  according to the randomized graph topology  $\mathcal{G}_{B_k}$ . We take a fixed non-random initial vector  $\hat{\theta}_0^i \in \mathbb{R}^{md}$ ,  $i \in \mathbb{N}$ , a positive step size  $\alpha$ , a gain coefficient  $\gamma$ , and a perturbation scale  $\beta$ . Consider the algorithm with two observations of distributed subfunctions  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$  for each sensor  $i \in \mathbb{N}$  to construct the sequences of observation points  $\{\mathbf{x}_t^i\}$  and estimates  $\{\hat{\theta}_t^i\}$  of the common state vector of all targets:

$$\begin{cases} \mathbf{x}_{2k}^i = \hat{\theta}_{2k-2}^i + \beta \hat{\Delta}_k^i, & \mathbf{x}_{2k-1}^i = \hat{\theta}_{2k-2}^i - \beta \hat{\Delta}_k^i, \\ \hat{\theta}_{2k-1}^i = \hat{\theta}_{2k-2}^i, \\ \hat{\theta}_{2k}^i = \hat{\theta}_{2k-1}^i - \alpha \left[ \hat{\Delta}_k^i \frac{y_{2k}^i - y_{2k-1}^i}{2\beta} + \right. \\ \qquad \qquad \qquad \left. \gamma \sum_{j \in \mathcal{N}_t^i} b_t^{i,j} (\hat{\theta}_{2k-1}^i - \hat{\theta}_{2k-1}^j) \right]. \end{cases} \quad (2.10)$$

Let's introduce the following notations:

$$\bar{\mathbf{y}}_t = \begin{pmatrix} y_t^1 \\ \vdots \\ y_t^n \end{pmatrix},$$

$$\bar{\Delta}_{t \div 2} = \text{diag}_{nmd} \begin{pmatrix} \hat{\Delta}_{t \div 2}^1 \\ \vdots \\ \hat{\Delta}_{t \div 2}^n \end{pmatrix}.$$

Using these notations, the algorithm (2.10) can be rewritten in the following form:

$$\begin{aligned} \bar{\theta}_{2k} = \bar{\theta}_{2k-1} - \alpha \left[ \bar{\Delta}_k \left( \frac{\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}}{2\beta} \otimes \mathbf{1}_{md} \right) + \right. \\ \left. \gamma (\mathcal{L}(B_{2k-1}) \otimes I_{md}) \bar{\theta}_{2k-1} \right]. \end{aligned} \quad (2.11)$$

To formulate the main result, we need to introduce the following notations and define constants with respect to Assumptions 1-4. Let  $\bar{\mathcal{L}} = \mathcal{L}(B_{av})$ ,

where  $\mathcal{L}(\cdot)$  is a function defined on the average network connectivity matrix  $B_{av}$ .  $\bar{\lambda}_2 = \text{Re}(\lambda_2(\bar{\mathcal{L}}))$ ,  $\bar{\lambda}_{\max} = \lambda_{\max}^{\frac{1}{2}}(\bar{\mathcal{L}}^T \bar{\mathcal{L}})$ ,  $\mathbb{E} \text{diag}\{\dots, I_t^{u_i}, \dots\} = I_{av}$ ,

$$\Lambda_{\max} = \lambda_{\max}(I_{av}), \Lambda_{\min} = \begin{cases} 1/\lambda_{\min}(I_{av}) & \text{if } \lambda_{\min}(I_{av}) > 0, \\ 0 & \text{if } \lambda_{\min}(I_{av}) \leq 0, \end{cases},$$

$$c_1 = 2\gamma\bar{\lambda}_2 + 4\Lambda_{\min},$$

$$c_2 = \gamma^2(\bar{\lambda}_{\max}^2 + \sigma_B^2) + 4 + 4\gamma\bar{\lambda}_{\max}\Lambda_{\max} + \frac{2\bar{q}\delta^2\Lambda_{\max}^2}{\beta^2},$$

$$c_3 = \delta(2\gamma\sqrt{nm}\bar{\lambda}_{\max} + 6\sqrt{m}\Lambda_{\max} + \sqrt{\bar{q}}),$$

$$c_4 = \delta(\gamma\bar{\lambda}_{\max}(\sqrt{\bar{q}} + 2\sqrt{m}\Lambda_{\max}) + \frac{\bar{q}\delta^2}{\beta^2}(2\sqrt{m}\Lambda_{\max}^2 + \sqrt{\bar{q}}\Lambda_{\max}) + 2(2\sqrt{m}\Lambda_{\max}^2 + \sqrt{\bar{q}}\Lambda_{\max})),$$

$$c_5 = n(8m\delta^2\Lambda_{\max} + 4\sqrt{m}\delta^2\sqrt{\bar{q}}),$$

$$c_6 = n(\frac{c_v^2}{\beta^2} + (1 + \frac{\bar{q}\delta^4}{2\beta^2})(4m\Lambda_{\max}^2 + 4\sqrt{m}\sqrt{\bar{q}} + \bar{q})).$$

**Definition 2.** *The sequence of estimates  $\bar{\theta}_{2k}$  has an asymptotically efficient upper bound  $\bar{L} > 0$  on the mean-square tracking error if  $\forall \varepsilon > 0; \exists \bar{k}$  such that  $\forall k > \bar{k}$*

$$\sqrt{\mathbb{E}\|\bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta_{2k}\|^2} \leq \bar{L} + \varepsilon.$$

The following theorem demonstrates an asymptotically efficient upper bound on the mean-square tracking error provided by the algorithm (2.10) ([30]).

**Theorem 1.** *If Assumptions 1-4 are satisfied and the positive constant  $\alpha$  is sufficiently small:  $\alpha < \min(c_1/c_2, 1/\mu)$ , then the sequence of estimates produced by the algorithm (2.10) has an asymptotically efficient upper bound equal to*

$$\bar{L} = \frac{1}{\mu} \left( h + \sqrt{h^2 + l\mu} \right), \quad (2.12)$$

where  $\mu = c_1 - \alpha c_2$ ,  $h = c_3 + \alpha c_4$ ,  $l = c_5 + \alpha c_6 + \frac{4nm\delta^2}{\alpha}$ .

*Proof.*

The proof largely follows the proof of the theorem in [108].

Let  $\mathbf{d}_t^i = \hat{\theta}_{2\lceil \frac{t-1}{2} \rceil}^i - \theta_t$ ,  $\bar{\mathbf{d}}_t = \text{col}\{\mathbf{d}_t^1, \dots, \mathbf{d}_t^n\}$ , where  $\lceil \cdot \rceil$  is the ceiling function,  $\nu_k = \|\bar{\mathbf{d}}_{2k}\|$ ,  $\bar{\mathbf{s}}_k = \frac{\alpha}{2\beta}((\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}) \otimes I_{md})\bar{\Delta}_k$ ,  $\bar{\mathbf{v}}_k = \text{col}\{\tilde{v}_k^1, \dots, \tilde{v}_k^n\}$ ,  $\bar{\mathbf{u}}_{t\div 2} = \text{col}\{\mathbf{u}_{t\div 2}^1, \dots, \mathbf{u}_{t\div 2}^n\}$ .

Let  $\bar{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k, \bar{\Delta}_k\}$  is the  $\sigma$ -algebra generated by all events  $\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k, \bar{\Delta}_k$ ,  $\tilde{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k\}$ , and  $\hat{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}\}$ :

$$\mathcal{F}_{k-1} \subset \hat{F}_{k-1} \subset \tilde{\mathcal{F}}_{k-1} \subset \bar{\mathcal{F}}_{k-1} \subset \mathcal{F}_k.$$

Taking into account the algorithm (2.10), Assumption 3, and the conditional expectation with respect to the sigma-algebra  $\bar{\mathcal{F}}_{k-1}$ , we obtain

$$\mathbb{E}_{\bar{\mathcal{F}}_{k-1}} \nu_k^2 = \|\bar{\mathbf{g}}_k\|^2 + \alpha^2 \gamma^2 \sigma_B^2 \nu_{k-1}^2 - 2\langle \bar{\mathbf{g}}_k, \mathbb{E}_{\bar{\mathcal{F}}_{k-1}} \bar{\mathbf{s}}_k \rangle + \mathbb{E}_{\bar{\mathcal{F}}_{k-1}} \|\bar{\mathbf{s}}_k\|^2, \quad (2.13)$$

wehr  $\bar{\mathbf{g}}_k = (I_{mnd} - \alpha\gamma\bar{\mathcal{L}} \otimes I_{md})\bar{\mathbf{d}}_{2k-2} + \mathbf{1}_n \otimes (\theta_{2k-2} - \theta_{2k})$ ,  $\bar{\mathcal{L}} = \mathcal{L}(B_{av})$ .

Under fulfillment of Assumption 3, we have  $\bar{\lambda}_2 > 0$  (see [90]). Hence, for the first term in (2.13) we derive

$$\|\bar{\mathbf{g}}_k\|^2 \leq (1 - 2\alpha\gamma\bar{\lambda}_2 + \alpha^2\gamma^2\bar{\lambda}_{\max}^2)\nu_{k-1}^2 + 4\alpha\gamma\sqrt{nm}\bar{\lambda}_{\max}\delta\nu_{k-1} + 4nm\delta^2.$$

Due to Assumptions 1 and 4 and the conditional expectation over the sigma-algebra  $\tilde{\mathcal{F}}_{k-1}$ , and then over the sigma-algebra  $\hat{F}_{k-1}$ , we can estimate the third term in (2.13) as follows:

$$\begin{aligned} 2\langle \bar{\mathbf{g}}_k, \mathbb{E}_{\hat{\mathcal{F}}_{k-1}} \bar{\mathbf{s}}_k \rangle &\leq (-4\alpha\Lambda_{\min} + 4\alpha^2\gamma\bar{\lambda}_{\max}\Lambda_{\max})\nu_{k-1}^2 + \\ &(4\alpha\sqrt{m}\delta\Lambda_{\max} + 2\alpha\sqrt{q}\delta + 2\alpha^2\gamma\bar{\lambda}_{\max}(\sqrt{q}\delta + 2\sqrt{m}\delta\Lambda_{\max})) \\ &+ 8\sqrt{m}\alpha\delta\Lambda_{\max})\nu_{k-1} + 8m\alpha\delta^2\Lambda_{\max} + 4\sqrt{m}\alpha\delta^2\sqrt{q}. \end{aligned}$$

Taking the conditional expectation over the sigma-algebra  $\mathcal{F}_{k-1}$  for the

last term in (2.13) and considering Assumption 2, we obtain:

$$\mathbb{E}_{\mathcal{F}_{k-1}} \|\bar{\mathbf{s}}_k\|^2 \leq \frac{\alpha^2}{4\beta^2} (2nc_v^2 + (2\bar{q}\delta^2 + 4\beta^2)(4\Lambda_{\max}^2\nu_{k-1}^2 + (8\sqrt{m}\delta\Lambda_{\max}^2 + 4\sqrt{\bar{q}}\delta\Lambda_{\max})\nu_{k-1} + n(4m\delta^2\Lambda_{\max}^2 + 4\sqrt{m}\delta\sqrt{\bar{q}}\delta + \bar{q}\delta^2))).$$

Summarizing and taking the conditional expectation over the sigma-algebra  $\mathcal{F}_{k-1}$ , we obtain the following from (2.13):

$$\mathbb{E}_{\mathcal{F}_{k-1}} \nu_k^2 \leq (1 - \mu\alpha)\nu_{k-1}^2 + 2\alpha h\nu_{k-1} + \alpha l. \quad (2.14)$$

□

**R e m a r k 2.** *The observation noise  $v_t$  in Theorem 1 can be considered almost arbitrary since it can be either non-random but bounded or realizations of a random process with arbitrary internal dependencies. In particular, it is not necessary to assume that  $v_t$  and  $\mathcal{F}_{t-1}$  are independent in order to prove the results of Theorem 1.*

**R e m a r k 3.** *The consequence of Theorem 1 shows that for the case  $\delta = 0$  (all targets do not change their positions over time), the asymptotic upper bound is given by  $\bar{L} = \frac{c_v\sqrt{n\alpha}}{\beta\sqrt{2\mu}}$ . For any level of noise  $c_v$ , this bound can be made arbitrarily small by choosing sufficiently small  $\alpha$ . However, in the case of moving targets, a large norm of changes  $\delta$  can be compensated by choosing a larger step size  $\alpha$ . This leads to a trade-off between reducing  $\alpha$  due to noisy observations and increasing  $\alpha$  due to the drift of optimal points.*

**R e m a r k 4.** *In the case of  $\Lambda_{\min} = 0$ , the traditional conditions of strict convexity typically imposed when justifying the correctness of such algorithms are not satisfied.*

**R e m a r k 5.** *The algorithm can be accelerated by adding the Nesterov acceleration method [7, 34, 60] or heavy-ball momentum term [59].*

The refined formulations of Theorem 1 allow us to derive the suboptimal step size of the modified algorithm (2.10) ([104]).



**Theorem 2.** Under the assumptions of Theorem 1,  $\alpha = \sqrt{\frac{C_2}{C_1}}$  is the suboptimal step size of algorithm (2.10), and the corresponding value of  $\hat{L}$  is given by

$$\hat{L} = 2\sqrt{C_1 C_2} + C_3. \quad (2.15)$$

*Proof.* For fixed  $\beta$  and  $\gamma$ , the upper bound of the estimates (2.12) can be estimated as follows:

$$\bar{L}^2 \leq \tilde{L} = \frac{h}{\mu} \left( 2 + \frac{l\mu}{2h^2} \right) = 2\frac{h}{\mu} + \frac{l}{2h} = \hat{L} + \mathcal{O}(\alpha^2),$$

where  $\hat{L} = \alpha C_1 + \frac{C_2}{\alpha} + C_3$ ,  $C_1 = (c_1(c_1 c_3^3 c_6 - c_1 c_3 c_4 c_5 + 4nm\delta^2 c_1 c_4^2 + 4c_3^3 c_4) + 4c_2 c_3^4)/(2c_1^2 c_3^3)$ ,  $C_2 = (4nm\delta^2)/(2c_3)$ ,  $C_3 = (4c_3^3 + c_3 c_5 c_1 - 4nm\delta^2 c_4 c_1)/(2c_1 c_3)$ , and  $\mathcal{O}(\alpha^2)$  represents the order of magnitude equivalent to  $\alpha^2$ . Taking the partial derivative with respect to  $\alpha$ , we have:

$$\frac{\partial \hat{L}}{\partial \alpha} = 0 \Rightarrow \alpha^* = \sqrt{\frac{C_2}{C_1}}.$$

□

## 2.3 Weighted Version of the Distributed Randomized Stochastic Optimization Algorithm Combined with Local Voting Protocol

In practical applications of various algorithms, there may be situations where certain observations of a random variable realization yield very large values. Typically, in real-world systems, when the parameters have large values, the models of dynamics are poorly known. In such situations, the model is often determined based on convenience of use. However, it is un-

desirable for a single large observation to have a significant impact on the computation of the current estimate. To address this, it is necessary to use robust procedures similar to those used in robust statistical analysis [12]. For example, if some targets have a different motion pattern compared to other targets, it is reasonable to group them separately. To account for the different motion patterns of different target groups, we introduce a weighted version of the distributed randomized stochastic optimization algorithm with local voting protocol. Furthermore, we will prove a theorem on the asymptotically efficient covariance matrix bound of the residuals, rather than the residual variances as in Section 2.2, which allows for a more detailed analysis of the algorithm convergence. It is worth noting that we will only consider the case  $p = d$ , where the number of sensors is equal to the dimension of the space.

Let us define a diagonal matrix  $\Psi = [\psi_{ij}]$ , where  $\psi_{ij} > 0$  if  $i = j$ , and  $\psi_{ij} = 0$  otherwise. We choose independent random vector sequences  $\Delta_k^i \in \mathbb{R}^{md}$  with a Bernoulli distribution, where each component independently takes values  $\pm \frac{1}{\sqrt{d}}$  with a probability of  $\frac{1}{2}$ , and  $\mathbf{u}_k^i \in \mathcal{U}_t$  according to the randomized topology graph  $\mathcal{G}_{B_k}$ . During the initialization stage, for each  $i \in \mathcal{N}$ , we select an initial vector  $\hat{\theta}_0^i \in \mathbb{R}^{md}$ , a positive step size  $\alpha$ , the matrix  $\Psi$ , a gain coefficient  $\gamma$ , and a disturbance scale  $\beta > 0$ .

To obtain estimates  $\{\hat{\theta}_t^i\}$  of the common state vectors  $\{\theta_t^i\}$  based on measurement points  $\{\mathbf{x}_t^i\}$ , we propose using a weighted algorithm with two measurements of distributed subfunctions  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$ :

$$\begin{cases} \mathbf{x}_{2k}^i = \hat{\theta}_{2k-2}^i + \beta \hat{\Delta}_k^i, & \mathbf{x}_{2k-1}^i = \hat{\theta}_{2k-2}^i - \beta \hat{\Delta}_k^i, \\ \hat{\theta}_{2k-1}^i = \hat{\theta}_{2k-2}^i, \\ \hat{\theta}_{2k}^i = \hat{\theta}_{2k-1}^i - \alpha \Psi \left[ \hat{\Delta}_k^i \frac{y_{2k}^i - y_{2k-1}^i}{2\beta} + \right. \\ \left. \gamma \sum_{j \in \bar{\mathcal{N}}_{2k-1}^i} b_{2k-1}^{i,j} (\hat{\theta}_{2k-1}^i - \hat{\theta}_{2k-1}^j) \right]. \end{cases} \quad (2.16)$$

Using the previously introduced notation and  $\bar{\Psi} = I_n \otimes \Psi$ , algorithm (2.16)

can be rewritten as follows:

$$\begin{aligned} \bar{\theta}_{2k} = \bar{\theta}_{2k-1} - \alpha \bar{\Psi} \left[ \bar{\Delta}_k \left( \frac{\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}}{2\beta} \otimes \mathbf{1}_{md} \right) + \right. \\ \left. \gamma(\mathcal{L}(B_{2k-1}) \otimes I_{md}) \bar{\theta}_{2k-1} \right]. \end{aligned} \quad (2.17)$$

To formulate the main result, we need to introduce the following notation and define new constants that differ from those introduced earlier in Section

$$2.2: c_1 = 1 - \varepsilon_1 - \alpha \sigma_{\Delta}^2 \varepsilon_2,$$

$$c_2 = \alpha \frac{4\sigma_{\Delta}^4}{c_1} + 2\sigma_{\Delta}^2 \varepsilon_3,$$

$$C_1 = \gamma \bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md}),$$

$$C_2 = \bar{\Psi} \bar{R}_k,$$

$$C_3 = \frac{1}{\beta} \sigma_{\Delta}^2 [(4\beta^2 \sigma_{\Delta}^2 + \delta^2)(4\nu_{k-1}^{\top} R_k \nu_{k-1} - 12\delta e \nu_{k-1} + 9\delta^2) - 2c_v \delta (2e \nu_{k-1} - 3\delta) + c_v^2] I_{nmd},$$

$$C_4 = \sigma_{\Delta}^2 \left[ \left(-\frac{1}{\varepsilon_1} + 1\right) \bar{\Psi}(\text{diag}_{nmd}(e)(9\mathbb{J}_n \otimes Q_{\delta}) \text{diag}_{nmd}(e) \bar{\Psi}^{\top} + \left(-2\frac{1}{\varepsilon_2} + 1\right)(4\mathbb{J}_n \otimes Q_{\delta}) \right],$$

$$C_5 = \left(1 - \frac{1}{\varepsilon_3}\right) 4\mathbb{J}_n \otimes Q_{\delta},$$

$$C_6 = C_1 + \frac{2\sigma_{\Delta}^2}{c_1} C_2.$$

**Definition 3.** *The covariance matrix of the residuals has an asymptotically efficient upper bound  $\Sigma > 0$  if there exists  $\bar{k}$  such that  $\forall k > \bar{k}$*

$$\mathbb{E}[(\bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta_{2k})(\bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta_{2k})^{\top}] \leq \Sigma + E_k,$$

zde  $E_k \rightarrow 0$ .

The following theorem shows the asymptotically efficient upper bound on the covariance matrix of the residuals achieved by the algorithm (2.16).

**Theorem 3.** *If Assumptions 1-4 are satisfied and  $\alpha > 0$ , then the covariance matrix of the residuals obtained by the algorithm (2.16) has an asymptotically efficient upper bound  $\Sigma$ , which is the solution to the following*

equation:

$$\begin{aligned} \Sigma &= c_1([I_{nmd} - \alpha C_6]\Sigma[I_{nmd} - \alpha C_6^T]) - \\ &\alpha c_2 C_2 \Sigma C_2^T + \alpha^2 \gamma^2 \bar{\Psi} \Sigma (Q_B \otimes I_{md}) \bar{\Psi}^T + \alpha^2 C_3 + \alpha C_4, \end{aligned} \quad (2.18)$$

*Proof.*

Denote  $\mathbf{d}_t^i = \hat{\theta}_{2[\frac{t-1}{2}]}^i - \theta_t$ ,  $\bar{\mathbf{d}}_t = \text{col}\{\mathbf{d}_t^1, \dots, \mathbf{d}_t^n\}$ , where  $[\cdot]$  is the ceiling function,  $\nu_k = \bar{\mathbf{d}}_{2k}$ ,  $\bar{\Sigma}_k = \mathbb{E}[\nu_k \nu_k^T]$ ,  $\bar{\mathbf{s}}_k = \frac{\alpha}{2\beta} \bar{\Delta}_k ((\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}) \otimes \mathbf{1}_{md})$ ,  $\bar{\mathbf{v}}_t = \text{col}\{\tilde{v}_t^1, \dots, \tilde{v}_t^n\}$ ,  $\bar{\mathbf{u}}_{t\div 2} = \text{col}\{\mathbf{u}_{t\div 2}^1, \dots, \mathbf{u}_{t\div 2}^n\}$ ,  $\bar{\Psi} = I_n \otimes \Psi$ .

Let  $\bar{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k, \bar{\Delta}_k\}$  is the  $\sigma$ -algebra generated by  $\mathcal{F}_{k-1}$ ,  $\bar{\mathbf{v}}_{2k-1}$ ,  $\bar{\mathbf{v}}_{2k}$ ,  $\xi_{2k-1}$ ,  $\xi_{2k}$ ,  $\bar{\mathbf{u}}_k$ ,  $\bar{\Delta}_k$ ,  $\tilde{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}, \bar{\mathbf{u}}_k\}$ , and  $\hat{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \xi_{2k-1}, \xi_{2k}\}$ :  $\mathcal{F}_{k-1} \subset \hat{\mathcal{F}}_{k-1} \subset \tilde{\mathcal{F}}_{k-1} \subset \bar{\mathcal{F}}_{k-1} \subset \mathcal{F}_k$ .

Considering  $\bar{\theta}_{2k-1} = \bar{\theta}_{2k-2}$  and  $\mathcal{L}(B_{2k-2})\mathbf{1}_n = 0$ , we obtain

$$\nu_k = \bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta_{2k} = \bar{\mathbf{g}}_k - \bar{\Psi} \bar{\mathbf{s}}_k - \alpha \gamma \bar{\Psi} [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}] \nu_{k-1},$$

where

$$\bar{\mathbf{g}}_k = [I_{nmd} - \alpha \gamma \bar{\Psi} (\mathcal{L}(B_{av}) \otimes I_{md})] \nu_{k-1} + \mathbf{1}_n \otimes (\theta_{2k-2} - \theta_{2k})$$

. Then

$$\begin{aligned} D_k &= \nu_k \nu_k^T = \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^T - \bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T - \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^T + \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T \bar{\Psi}^T - \\ &\alpha \gamma (\bar{\mathbf{g}}_k - \bar{\Psi} \bar{\mathbf{s}}_k) [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]^T \bar{\Psi}^T - \\ &\alpha \gamma \bar{\Psi} [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}] \nu_{k-1} (\bar{\mathbf{g}}_k^T - \bar{\mathbf{s}}_k^T \bar{\Psi}^T) + \\ &\alpha^2 \gamma^2 \bar{\Psi} [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}] D_{k-1} [(\mathcal{L}(B_{2k-2}) - \\ &\mathcal{L}(B_{av})) \otimes I_{md}]^T \bar{\Psi}^T. \end{aligned}$$

By taking the conditional expectation with respect to the  $\sigma$ -algebra  $\bar{\mathcal{F}}_{k-1}$

and applying Assumption 3, we have:

$$\begin{aligned} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[D_k] &\leq \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^\top - \bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^\top \bar{\Psi}^\top - \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^\top + \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^\top \bar{\Psi}^\top + \\ &\quad \alpha^2 \gamma^2 b_{\max} \bar{\Psi} D_{k-1} \bar{\Psi}^\top, \end{aligned} \quad (2.19)$$

where  $b_{\max}$  is the maximum element of  $Q_B$ .

Next, let's take the conditional expectation over the  $\sigma$ -algebra  $\tilde{\mathcal{F}}_{k-1}$ :

$$\begin{aligned} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[D_k] &\leq \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^\top - \bar{\mathbf{g}}_k \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k^\top] \bar{\Psi}^\top - \bar{\Psi} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k] \bar{\mathbf{g}}_k^\top + \\ &\quad \bar{\Psi} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^\top] \bar{\Psi}^\top + \alpha^2 \gamma^2 b_{\max} \bar{\Psi} D_{k-1} \bar{\Psi}^\top. \end{aligned} \quad (2.20)$$

For the first term in (2.20), using Assumption 1, we can derive:

$$\begin{aligned} \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^\top &\leq (1 - \varepsilon_3) ([I_{nmd} - \alpha \gamma \bar{\Psi} (\mathcal{L}(B_{av}) \otimes I_{md})] \Sigma_{k-1} \cdot \\ &\quad \cdot [I_{nmd} - \alpha \gamma \bar{\Psi} (\mathcal{L}(B_{av}) \otimes I_{md})]^\top) + (1 - \frac{1}{\varepsilon_3}) 4 \mathbb{J}_n \otimes Q_\delta. \end{aligned}$$

Denote  $\mathbf{r}_t^i = \mathbf{e}_{l(\mathbf{u}_{t+2}^i)} \otimes [C_t^{\mathbf{u}_{t+2}^i}]^{-1} D_t^{\mathbf{u}_{t+2}^i}$ ,  $\hat{\mathbf{r}}_t^i = \text{diag}_{md}(\mathbf{e}_{l(\mathbf{u}_{t+2}^i)} \otimes I_d) \hat{\theta}_t^i$ , and  $\Xi_t^i = \mathbf{r}_t^i - \mathbf{r}_{t-1}^i$ , тогда  $\forall i \in \{1, \dots, n\}$ :

$$\begin{aligned} y_{2k}^i - y_{2k-1}^i &= (2\beta \hat{\Delta}_k^i - \Xi_{2k}^i)^\top (2 \text{diag}_{md}(\mathbf{e}_{l(\mathbf{u}_k^i)} \otimes I_d) \mathbf{d}_{2k-2}^i \\ &\quad - 2\Xi_{2k-1}^i - \Xi_{2k}^i) + v_{2k}^i - v_{2k-1}^i. \end{aligned}$$

Assuming the fulfillment of Assumption 4, by multiplying with  $\hat{\Delta}_k^i$  and taking the conditional expectation over the  $\sigma$ -algebra  $\tilde{\mathcal{F}}_{k-1}$ , we obtain:

$$\mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[(y_{2k}^i - y_{2k-1}^i) \hat{\Delta}_k^i] = 2\beta \sigma_\Delta^2 \mathbf{q}_{2k-2}^i,$$

where  $\mathbf{q}_{2k-2}^i = 2 \text{diag}_{md}(\mathbf{e}_{l(\mathbf{u}_k^i)} \otimes I_d) \mathbf{d}_{2k-2}^i - 2\Xi_{2k-1}^i - \Xi_{2k}^i$ .

Let  $\sigma_\Delta = \frac{1}{\sqrt{d}}$ . Denote  $\bar{R}_t = \text{diag}_{nmd}(\text{col}\{\mathbf{e}_{l(\mathbf{u}_{t+2}^1)} \otimes I_d, \dots, \mathbf{e}_{l(\mathbf{u}_{t+2}^n)} \otimes I_d\})$ ,  $\bar{\Xi}_t = \text{col}\{\Xi_t^1, \dots, \Xi_t^n\}$ ,  $e = \text{col}\{e_{l(\mathbf{u}_t^1)}, \dots, e_{l(\mathbf{u}_t^n)}\}$ . According to Assumption

1, by taking the conditional expectation over the  $\sigma$ -algebra  $\hat{\mathcal{F}}_{k-1}$  and using

$$\forall \varepsilon > 0 : AB^T + BA^T \leq \varepsilon AA^T + \frac{1}{\varepsilon} BB^T$$

, we obtain the following expressions for the second and third terms in (2.20):

$$\begin{aligned} & -\bar{\mathbf{g}}_k \mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k^T] \bar{\Psi}^T - \bar{\Psi} \mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k] \bar{\mathbf{g}}_k^T \leq \\ & \alpha \sigma_{\Delta}^2 [-2(I_{nmd} - 2\alpha\gamma \bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md})) \Sigma_{k-1} \bar{R}_k \bar{\Psi}^T - \\ & 2\bar{\Psi} \bar{R}_k \Sigma_{k-1} (I_{nmd} - 2\alpha\gamma (\mathcal{L}(B_{av})^T \otimes I_{md}) \bar{\Psi}^T) - \\ & \varepsilon_1 (I_{nmd} - 2\alpha\gamma \bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md})) \Sigma_{k-1} (I_{nmd} - \\ & 2\alpha\gamma (\mathcal{L}(B_{av})^T \otimes I_{md}) \bar{\Psi}^T) + (-\frac{1}{\varepsilon_1} + 1) \bar{\Psi}(\text{diag}_{nmd}(e) \\ & (9\mathbb{J}_n \otimes Q_{\delta}) \text{diag}_{nmd}(e) \bar{\Psi}^T - \\ & 2\varepsilon_2 \bar{\Psi} \bar{R}_k \Sigma_{k-1}^T \bar{R}_k \bar{\Psi}^T + (-2\frac{1}{\varepsilon_2} + 1)(4\mathbb{J}_n \otimes Q_{\delta})]. \end{aligned}$$

According to Assumptions 1 and 4, and using the fact that  $\Delta_k^i$  is obtained from a symmetric distribution, we can take the conditional expectation over the  $\sigma$ -algebra  $\tilde{\mathcal{F}}_{k-1}$  as the fourth term in (2.20), and then take the conditional expectation over the  $\sigma$ -algebra  $\hat{\mathcal{F}}_{k-1}$ . As a result, we obtain the following expression:

$$\begin{aligned} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T] & \leq \frac{\alpha^2}{\beta} \sigma_{\Delta}^2 [(4\beta^2 \sigma_{\Delta}^2 + \delta^2)(4\nu_{k-1}^T R_k \nu_{k-1} - \\ & 12\delta \varepsilon \nu_{k-1} + 9\delta^2) - 2c_v \delta (2\varepsilon \nu_{k-1} - 3\delta) + c_v^2] I_{nmd}. \end{aligned}$$

By summing up the upper bounds and taking the unconditional expectation, we obtain the following expression:

$$\begin{aligned} \Sigma_k & \leq c_1 ([I_{nmd} - \alpha C_6] \Sigma_{k-1} [I_{nmd} - \alpha C_6^T]) - \\ & \alpha c_2 C_2 \Sigma_{k-1} C_2^T + \alpha^2 \gamma^2 b_{\max} \bar{\Psi} \Sigma_{k-1} \bar{\Psi}^T + \alpha^2 C_3 + \alpha C_4 + C_5. \end{aligned}$$

Let's introduce the substitution:  $\Sigma_k = \Sigma + E_k$ , where  $\Sigma$  is the solution of equation (2.18).

Then,

$$E_k \leq c_1([I_{nmd} - \alpha C_6]E_{k-1}[I_{nmd} - \alpha C_6^T]) - \alpha c_2 C_2 E_{k-1} C_2^T + \alpha^2 \gamma^2 b_{\max} \bar{\Psi} E_{k-1} \bar{\Psi}^T + C_5.$$

According to [16]  $E_k \rightarrow 0$ . Therefore, the theorem is proven. □

## 2.4 Distributed Randomized Stochastic Optimization Algorithm Combined with Local Voting Protocol for Estimating the Positions of Static Objects

In the previous sections, we considered cases where the targets and sensors can change their positions over time. The convergence region for algorithms with such non-stationary problem settings was found to be a set of possible values. However, it is possible to construct an algorithm for estimating the positions of static objects that converges to a point, as in [12] (the optimal state vector of all targets  $\theta$  will not depend on time because the targets are stationary). To achieve this, we modify the distributed randomized stochastic optimization algorithm combined with the local voting protocol. The main difference from algorithm (2.16) is the replacement of the constant step size  $\alpha$  with a variable step size  $\alpha_k$ .

Let  $\Psi = [\psi_{ij}]$  be a diagonal matrix, where  $\psi_{ij} > 0$  if  $i = j$ , and  $\psi_{ij} = 0$  otherwise. We choose observed sequences of independent random vectors  $\Delta_k^i \in \mathbb{R}^{md}$  with a Bernoulli distribution, where each component independently takes values  $\pm \frac{1}{\sqrt{d}}$  with probability  $\frac{1}{2}$ , and  $\mathbf{u}_k^i \in \mathcal{U}_t$  according to the randomized graph topology  $\mathcal{G}_{B_k}$ . For each  $i \in \mathcal{N}$ , we choose  $\hat{\theta}_0^i \in \mathbb{R}^{md}$ , a

positive step size  $\alpha_k$ , a matrix  $\Psi$ , a gain coefficient  $\gamma$ , and a disturbance scale  $\beta > 0$ .

To obtain estimates  $\{\widehat{\theta}_t^i\}$  of the common state vectors  $\{\theta^i\}$  based on measurement points  $\{\mathbf{x}_t^i\}$ , we propose to use a weighted algorithm with two distributed function measurements  $f_t^i(\mathbf{u}_t^i, \text{Pr}_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$ :

$$\begin{cases} \mathbf{x}_{2k}^i = \widehat{\theta}_{2k-2}^i + \beta \widehat{\Delta}_k^i, & \mathbf{x}_{2k-1}^i = \widehat{\theta}_{2k-2}^i - \beta \widehat{\Delta}_k^i, \\ \widehat{\theta}_{2k-1}^i = \widehat{\theta}_{2k-2}^i, \\ \widehat{\theta}_{2k}^i = \widehat{\theta}_{2k-1}^i - \alpha_k \Psi \left[ \widehat{\Delta}_k^i \frac{y_{2k}^i - y_{2k-1}^i}{2\beta} + \right. \\ \quad \left. \gamma \sum_{j \in \bar{\mathcal{N}}_{2k-1}^i} b_{2k-1}^{i,j} (\widehat{\theta}_{2k-1}^i - \widehat{\theta}_{2k-1}^j) \right]. \end{cases} \quad (2.21)$$

Using the notations introduced earlier, algorithm (2.21) can be represented as follows:

$$\begin{aligned} \bar{\theta}_{2k} = \bar{\theta}_{2k-1} - \alpha_k \bar{\Psi} \left[ \bar{\Delta}_k \left( \frac{\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}}{2\beta} \otimes \mathbf{1}_{md} \right) + \right. \\ \left. \gamma (\mathcal{L}(B_{2k-1}) \otimes I_{md}) \bar{\theta}_{2k-1} \right]. \end{aligned} \quad (2.22)$$

The following theorem provides an asymptotically efficient upper bound on the covariance matrix of residuals obtained by algorithm (2.21).

**Theorem 4.** *If Assumptions 2–4 hold,  $\alpha_k = \frac{1}{k}$ , and  $-(\gamma \bar{\lambda}_2 + \frac{2}{m}) \bar{\Psi} + \frac{1}{2} I_{nmd}$  is stable (all its eigenvalues lie in the left half-plane), then the asymptotic convergence rate of algorithm (2.21) is characterized by the inequality*

$$\mathbb{E}[(\widehat{\theta}_t - \theta)(\widehat{\theta}_t - \theta)^T] \leq \frac{1}{n} S + o\left(\frac{1}{n}\right), \quad (2.23)$$

where the matrix  $S$  is the solution to the matrix equation

$$\begin{aligned} S \left( (\gamma \bar{\lambda}_2 + \frac{2}{m}) \bar{\Psi}^T - \frac{1}{2} I_{nmd} \right) + \\ \left( (\gamma \bar{\lambda}_2 + \frac{2}{m}) \bar{\Psi} - \frac{1}{2} I_{nmd} \right) S = 4nc_v^2 \bar{\Psi} \bar{\Psi}^T. \end{aligned} \quad (2.24)$$



*Proof.*

Let us denote  $\mathbf{d}_t^i = \widehat{\theta}_{2\lceil \frac{t-1}{2} \rceil}^i - \theta$ ,  $\bar{\mathbf{d}}_t = \text{col}\{\mathbf{d}_t^1, \dots, \mathbf{d}_t^n\}$ , where  $\lceil \cdot \rceil$  denotes the ceiling function,  $\nu_k = \bar{\mathbf{d}}_{2k}$ ,  $D_k = \nu_k \nu_k^\top$ ,  $\Sigma_k = \mathbb{E}[D_k]$ ,  $\bar{\mathbf{s}}_k = \frac{\alpha_k}{2\beta} \bar{\Delta}_k ((\bar{\mathbf{y}}_{2k} - \bar{\mathbf{y}}_{2k-1}) \otimes \mathbf{1}_{md})$ ,  $\bar{\mathbf{v}}_t = \text{col}\{\tilde{v}_t^1, \dots, \tilde{v}_t^n\}$ ,  $\bar{\mathbf{u}}_k = \text{col}\{\mathbf{u}_k^1, \dots, \mathbf{u}_k^n\}$ ,  $\bar{\Psi} = I_n \otimes \Psi$ .

Let  $\bar{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \bar{\mathbf{u}}_k, \bar{\Delta}_k\}$  be the sigma-algebra generated by the probability events generated by  $\mathcal{F}_{k-1}$ ,  $\bar{\mathbf{v}}_{2k-1}$ ,  $\bar{\mathbf{v}}_{2k}$ ,  $\bar{\Delta}_k$ ,  $\tilde{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}, \bar{\mathbf{u}}_k\}$ , и  $\hat{\mathcal{F}}_{k-1} = \sigma\{\mathcal{F}_{k-1}, \bar{\mathbf{v}}_{2k-1}, \bar{\mathbf{v}}_{2k}\}$ :  $\mathcal{F}_{k-1} \subset \hat{\mathcal{F}}_{k-1} \subset \tilde{\mathcal{F}}_{k-1} \subset \bar{\mathcal{F}}_{k-1} \subset \mathcal{F}_k$ .

Using that  $\bar{\theta}_{2k-1} = \bar{\theta}_{2k-2}$  and  $\mathcal{L}(B_{2k-2})\mathbf{1}_n = 0$ , we get

$$\begin{aligned} \nu_k &= \bar{\theta}_{2k} - \mathbf{1}_n \otimes \theta = \bar{\mathbf{g}}_k - \bar{\Psi}\bar{\mathbf{s}}_k - \\ &\alpha_k \gamma \bar{\Psi}[(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}] \nu_{k-1}, \end{aligned}$$

where  $\bar{\mathbf{g}}_k = [I_{nmd} - \alpha_k \gamma \bar{\Psi}(\mathcal{L}(B_{av}) \otimes I_{md})] \nu_{k-1}$ .

Then

$$\begin{aligned} D_k &= \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^\top - \bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^\top \bar{\Psi}^\top - \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^\top + \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^\top \bar{\Psi}^\top - \\ &\alpha_k \gamma (\bar{\mathbf{g}}_k - \bar{\Psi} \bar{\mathbf{s}}_k) \nu_{k-1}^\top [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]^\top \bar{\Psi}^\top - \\ &\alpha_k \gamma \bar{\Psi}[(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}] \nu_{k-1} (\bar{\mathbf{g}}_k^\top - \bar{\mathbf{s}}_k^\top \bar{\Psi}^\top) + \\ &\alpha_k^2 \gamma^2 \bar{\Psi}[(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}] D_{k-1} [(\mathcal{L}(B_{2k-2}) - \mathcal{L}(B_{av})) \otimes I_{md}]^\top \bar{\Psi}^\top. \end{aligned}$$

Now, let's take the conditional expectation over the sigma-algebra  $\bar{\mathcal{F}}_{k-1}$  and apply Assumption 3:

$$\begin{aligned} \mathbb{E}_{\bar{\mathcal{F}}_{k-1}}[D_k] &\leq \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^\top - \bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^\top \bar{\Psi}^\top - \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^\top + \bar{\Psi} \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^\top \bar{\Psi}^\top + \\ &\alpha_k^2 \gamma^2 b_{\max} \|D_{k-1}\|^2 \bar{\Psi} \bar{\Psi}^\top, \end{aligned} \quad (2.25)$$

where  $b_{\max}$  is the maximum element of  $Q_B$ .

Taking the conditional expectation consecutively over the sigma-algebra

$\tilde{\mathcal{F}}_{k-1}$ , we have:

$$\begin{aligned} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[D_k] &\leq \bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^\top - \bar{\mathbf{g}}_k \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k^\top] \bar{\Psi}^\top - \bar{\Psi} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k] \bar{\mathbf{g}}_k^\top + \\ &\quad \bar{\Psi} \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^\top] \bar{\Psi}^\top + \alpha_k^2 \gamma^2 b_{\max} \|D_{k-1}\|^2 \bar{\Psi} \bar{\Psi}^\top. \end{aligned} \quad (2.26)$$

Under Assumption 3, we have  $\bar{\lambda}_2 > 0$  (see [90]). Therefore, for the first term in (2.26), we obtain

$$\bar{\mathbf{g}}_k \bar{\mathbf{g}}_k^\top \leq D_{k-1} - \alpha_k \gamma \bar{\lambda}_2 (\bar{\Psi} D_{k-1} + D_{k-1} \bar{\Psi}^\top) + \alpha_k^2 \gamma^2 \bar{\lambda}_{\max}^2 \|D_{k-1}\|^2 \bar{\Psi} \bar{\Psi}^\top.$$

Let  $\mathbf{r}^{l(\mathbf{u}_k^i)} = \mathbf{e}_{l(\mathbf{u}_k^i)} \otimes [C^{\mathbf{u}_k^i}]^{-1} D^{\mathbf{u}_k^i}$ ,  $\hat{\mathbf{r}}_t^{l(\mathbf{u}_k^i)} = \text{diag}_{md}(\mathbf{e}_{l(\mathbf{u}_k^i)} \otimes I_d) \hat{\theta}_t^i$ ,  $\tilde{v}_k^i = v_{2k}^i - v_{2k-1}^i$ , then for any  $i \in \{1, \dots, n\}$ :

$$y_{2k}^i - y_{2k-1}^i = 4\beta (\hat{\Delta}_k^i)^\top (\hat{\mathbf{r}}_{2k-2}^{l(\mathbf{u}_k^i)} - \mathbf{r}^{l(\mathbf{u}_k^i)}) + \tilde{v}_k^i.$$

Under Assumption 4, we have  $\mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\tilde{v}_k^i \hat{\Delta}_k^i] = 0$ . Let us denote

$$\bar{R}_t = \text{diag}_{nmd}(\text{col}\{\mathbf{e}_{l(\mathbf{u}_{t \div 2}^1)} \otimes I_d, \dots, \mathbf{e}_{l(\mathbf{u}_{t \div 2}^n)} \otimes I_d\}).$$

According to Assumption 4, considering that  $\Delta_k^i$  is taken from a symmetric distribution, for the fourth term in (2.26) we have

$$\begin{aligned} &\mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^\top] \leq \\ &4\alpha_k^2 \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\Delta}_k (\bar{\Delta}_k)^\top \bar{R}_k D_{k-1} \bar{R}_k \bar{\Delta}_k^i (\bar{\Delta}_k)^\top] + \\ &\quad \frac{\alpha_k^2}{4\beta^2} \sum_{i \in \mathcal{N}} (\tilde{v}_k^i)^2 \mathbb{E}_{\tilde{\mathcal{F}}_{k-1}}[\bar{\Delta}_k^i (\bar{\Delta}_k)^\top]. \end{aligned}$$

Let  $Z_k = \mathbb{E}[\bar{\Delta}_k (\bar{\Delta}_k)^\top]$ . Summing up the second and third terms from (2.26) and taking the conditional expectation over the sigma-algebra  $\hat{\mathcal{F}}_{k-1}$ ,

we obtain:

$$\begin{aligned}
& -\mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{g}}_k \bar{\mathbf{s}}_k^T] \bar{\Psi}^T - \bar{\Psi} \mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{g}}_k^T] \leq \\
& -2 \frac{\alpha_k}{m} (D_{k-1} Z_k^T \bar{\Psi}^T + \bar{\Psi} Z_k D_{k-1}) + \\
& 2\alpha_k^2 \gamma \frac{1}{m} \bar{\lambda}_{\max} \bar{\Psi} (D_{k-1} Z_k^T + Z_k D_{k-1}) \bar{\Psi}^T.
\end{aligned}$$

Taking the conditional expectation over the sigma-algebra  $\hat{\mathcal{F}}_{k-1}$ , we have:

$$\begin{aligned}
\mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T] & \leq 4 \frac{\alpha_k^2}{m^2} \|D_{k-1}\| \mathbb{E}_{\hat{\mathcal{F}}_{k-1}}[\|\bar{\Delta}_k\|^4] + \\
& \frac{\alpha_k^2}{4\beta^2} \sum_{i \in \mathcal{N}} (\tilde{v}_k^i)^2 Z_k.
\end{aligned}$$

Finally, taking the conditional expectation over the sigma-algebra  $\mathcal{F}_{k-1}$  according to Assumption 2, for the fourth term in (2.26) we obtain

$$\begin{aligned}
\bar{\Psi} \mathbb{E}_{\mathcal{F}_{k-1}}[\bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T] \bar{\Psi}^T & \leq 4 \frac{\alpha_k^2}{m^2} \|D_{k-1}\| \bar{\Psi} \mathbb{E}_{\mathcal{F}_{k-1}}[\|\bar{\Delta}_k\|^4] \bar{\Psi}^T + \\
& \frac{\alpha_k^2}{4\beta^2} n c_v^2 \bar{\Psi} Z_k.
\end{aligned}$$

In the end, taking the unconditional expectation and considering that  $\|z_k^{-1} Z_k - I_{nmd}\| = \mathcal{O}(k^{-1})$ ,  $\alpha_k z_k = k^{-1}$ , we have the following from (2.26):

$$\begin{aligned}
\Sigma_k & \leq \Sigma_{k-1} - \left(\alpha_k \gamma \bar{\lambda}_2 + \frac{1}{k} \frac{2}{m}\right) (\Sigma_{k-1} \bar{\Psi}^T + \bar{\Psi} \Sigma_{k-1}) + \\
& 4\alpha_k^2 z_k n c_v^2 \bar{\Psi} \bar{\Psi}^T + \frac{1}{k} \kappa_k \mathcal{O}(\|\Sigma_{k-1}\|) + o(k^{-2}),
\end{aligned}$$

where  $\{\kappa_k\} : \kappa_k \rightarrow 0$  if  $k \rightarrow \infty$ .

Let

$$\begin{aligned}
\alpha_k & = \frac{1}{k}, \\
W_k & = \frac{1}{\frac{1}{k} \gamma \bar{\lambda}_2 + \frac{1}{k} \frac{2}{m}} \left(\Sigma_k - \frac{1}{k} S\right).
\end{aligned}$$

Consider the equation (2.24): if

$$-(\gamma\bar{\lambda}_2 + \frac{2}{m})\bar{\Psi} + \frac{1}{2}I_{nmd}$$

is stable, then there exists a positive definite matrix  $S$  that is a solution to this Lyapunov equation.

We estimate  $W_k$  from above:

$$\begin{aligned} W_k \leq & W_{k-1} - \frac{1}{k-1}W_{k-1} \left( (\gamma\bar{\lambda}_2 + \frac{2}{m})\bar{\Psi}^T - \frac{1}{2}I_{nmd} \right) - \\ & \frac{1}{k-1} \left( (\gamma\bar{\lambda}_2 + \frac{2}{m})\bar{\Psi} - \frac{1}{2}I_{nmd} \right) W_{k-1} + \\ & \frac{1}{k}\kappa_k \mathcal{O}(\|W_{k-1}\|) + o(k^{-1}). \end{aligned}$$

According to Lemma 9 in [12], we have  $W_k \xrightarrow[k \rightarrow \infty]{} 0$ . Therefore, the theorem is proven.  $\square$

## 2.5 Summary

In the second chapter, it is shown that due to the specifics of the problem, the distributed randomized stochastic approximation algorithm combined with the local voting protocol can be modified considering the developed sensor network management model. Three variants of such an algorithm are presented: an algorithm for estimating moving targets (2.10), a weighted version of the algorithm to account for the different nature of target motion (2.16), and an algorithm for estimating the locations of static objects (2.21). The applicability conditions of the distributed randomized stochastic approximation algorithm combined with the local voting protocol from [108] are also investigated for the network observation model. For the first algorithm, Theorem 1 is proven regarding convergence to the set of possible values. This theorem largely repeats the theorem from [108], but

the refined formulations allow for the derivation of a suboptimal step size for the algorithm (Theorem 2). For the second algorithm, Theorem 3 is proven regarding the asymptotic efficient boundary of the covariance matrix of the residuals. For the third algorithm, Theorem 4 is proven regarding the asymptotic convergence rate of the algorithm.

# Chapter 3. Object Motion Tracking with Uncertainties and Constraints on Agent Communications

## 3.1 Numerical Experiments

This section presents numerical experiments that illustrate the performance of the proposed algorithms (2.10), (2.16), (2.21).

Consider a distributed network of 10 planar ( $d = 2$ ) intelligent sensors (agents) that can collect data only from one of their neighbors. The sensors have 25 planar targets in their field of view, and the state vectors of these targets are to be estimated using the modified distributed randomized stochastic optimization algorithm combined with the local voting protocol for target tracking (2.10). At time  $t$ ,  $\mathbf{s}_t^i = [s_t^{i,1}, s_t^{i,2}]^T \in \mathbb{R}^2$  is the current state vector of sensor  $i, i \in \mathcal{N} = \{1, 2, \dots, 10\}$ ,  $\mathbf{r}_t^l = [r_t^{l,1}, r_t^{l,2}]^T \in \mathbb{R}^2$  is the state vector of target  $l, l \in \mathcal{M} = \{1, 2, \dots, 25\}$ , and  $\theta_t = \text{col}\{\mathbf{r}_t^1, \dots, \mathbf{r}_t^{25}\}$  is the common state vector of all targets.

Let's consider three types of disturbances: uniformly distributed random variables within the range  $[-1; 1]$ , unknown constants, and hybrid noise uniformly distributed around time-varying constants, for example,  $v_k^i = \pm 1 + 0.1 * \sin(k)$ , where the sign before 1 switches every 50th iteration. In the simulations presented below, the estimations typical for each type of

disturbance are shown.

Targets  $l = 1, \dots, m$  start their motion from positions consisting of randomly selected components from the interval  $[0; 120]$ . The dynamics of the target motion are as follows:  $\mathbf{r}_t^l = \mathbf{r}_{t-1}^l + \chi_{t-1}^l$ . Let  $\chi_{t-1}^l$  be a random vector uniformly distributed on a sphere with a radius of 0.25. The observers do not move, and their coordinates are random variables uniformly distributed in the interval  $[200; 250]$ .

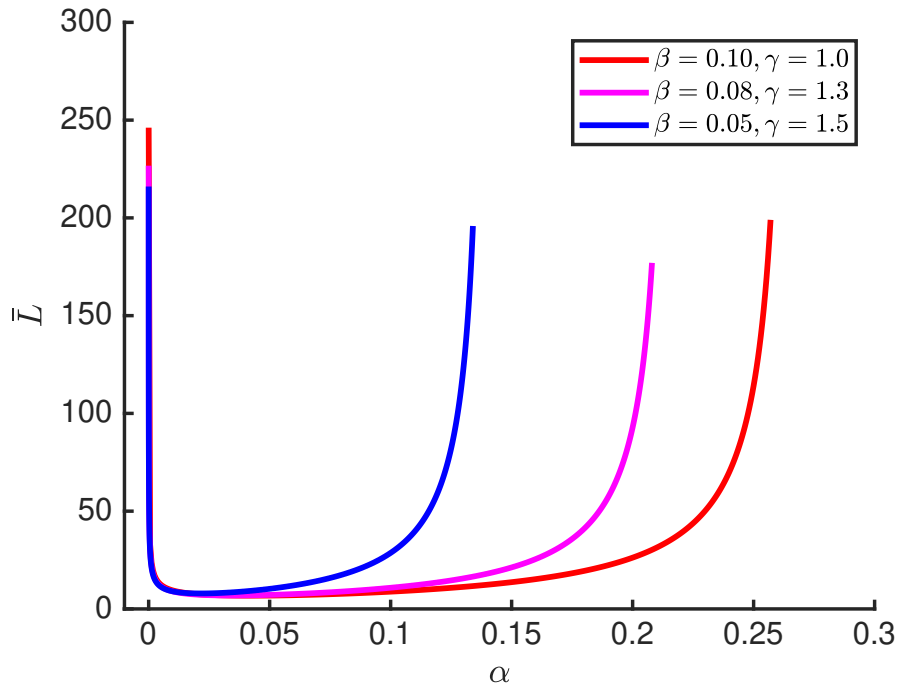


Figure 3.5: Dependency of the upper bound of mean squared errors on the step  $\alpha$  of the algorithm for different parameters  $\beta$  and  $\gamma$ .

Figure 3.5 shows the dependence of the upper bound of mean squared errors on different parameters  $\beta$  and  $\gamma$ . It can be observed that the choice of  $\beta$  and  $\gamma$  has almost no influence on the convergence of the algorithm when the step  $\alpha$  is chosen optimally. Thus, it is reasonable to adjust only one parameter of the algorithm, namely the step  $\alpha$ , which was implemented in Theorem 2.

The step size  $\alpha = 0.05$  was chosen based on Theorem 2 with  $\beta = 0.1$  and  $\gamma = 1.0$ . Figure 3.5 demonstrates that this  $\alpha$  is a suboptimal minimum of  $\bar{L}$ .

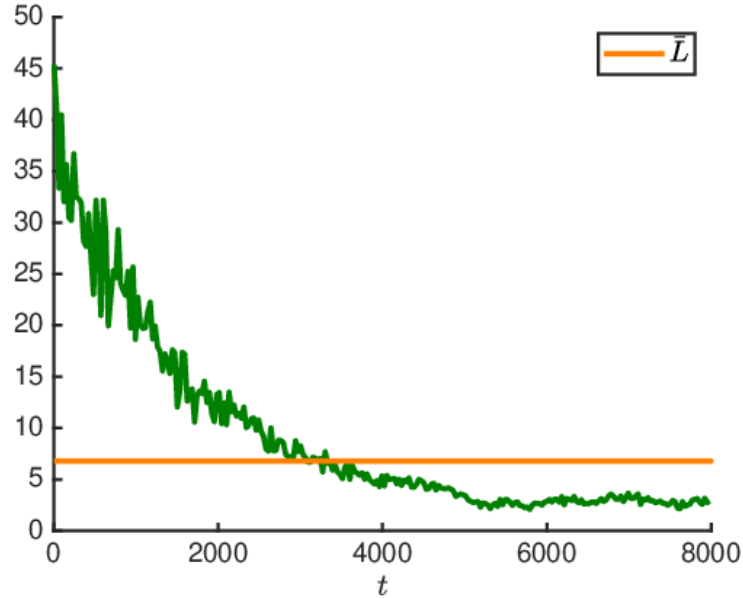


Figure 3.6: Residuals  $f_t^i(\mathbf{u}_t^i, \Pr_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$ , averaged over sensors and targets, and the upper bound  $\bar{L}$  of the mean squared tracking error

Figure 3.6 shows how the average value of the observations

$$f_t^i(\mathbf{u}_t^i, \Pr_{l(\mathbf{u}_t^i)}(\mathbf{x}_t^i))$$

changes over time. It can be seen that there is a moment in time  $t$  from which the estimates converge to the true value and fluctuate around it. Moreover, it is evident that the residuals are below the upper bound of the mean square tracking error  $\bar{L} = 6.8$  calculated according to Theorem 1.

Next, we consider numerical results obtained using the weighted version of the distributed randomized stochastic optimization algorithm combined with the local voting protocol (2.16).

Let's consider a distributed network of three planar ( $d = 2$ ) intelligent sensors (agents) that can collect data from two of their neighbors. There are 6 moving targets in the region of interest. The sensors need to estimate the locations of the targets in real-time.

Similar to the previous simulation, we consider three types of disturbances: uniformly distributed random variables within the range  $[-1; 1]$ ,



unknown constants, and hybrid noise uniformly distributed around time-varying constants, for example,  $v_k^i = \pm 1 + 0.1 * \sin(k)$ , where the sign before 1 switches every 50th iteration. The simulations presented below show estimations typical for each type of disturbance.

The algorithm (2.16), operating on each node (sensor), has the following parameters:  $\alpha = 0.05$ ,  $\beta = 0.1$ ,  $\gamma = 1.0$ . The targets start their motion from positions consisting of randomly selected components from the interval  $[0; 100]$ . The dynamics of the target motion are as follows:  $\mathbf{r}_t^l = \mathbf{r}_{t-1}^l + \chi_{t-1}^l$ . Let  $\chi_{t-1}^l$  be a random vector uniformly distributed on a sphere with a radius of 0.2 for targets with odd numbers and 0.6 for targets with even numbers. This means that some targets move faster than others. Let the sensors be stationary, and their coordinates be random variables uniformly distributed in the interval  $[100; 120]$ .

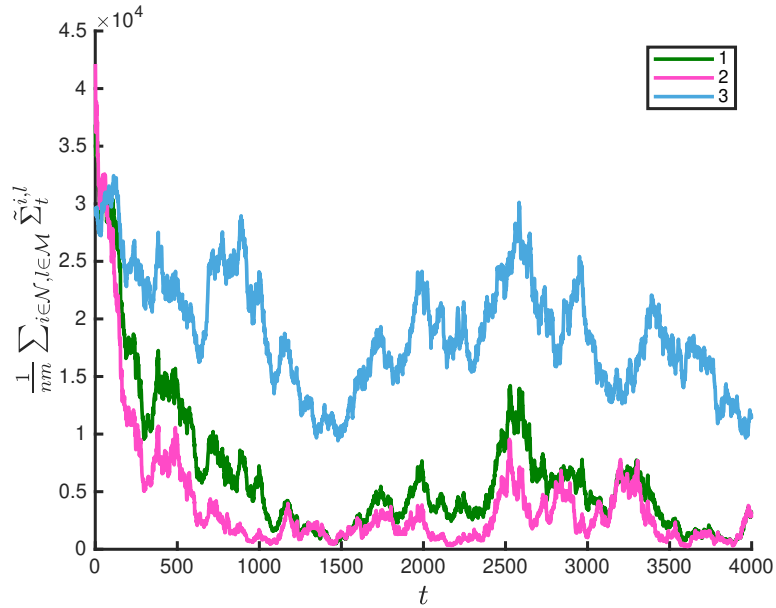


Figure 3.7: Average value of the first diagonal element of the covariance matrix of residuals  $\tilde{\Sigma}_t^{i,l}$ , averaged over all sensors and targets, where (1) corresponds to the previous algorithm (2.10) with  $\Psi = I_{md}$ , (2) and (3) correspond to the algorithm (2.16) with  $\Psi$  equal to  $\Psi^*$  and  $\Psi^{**}$ , respectively

For each target  $l$  and sensor  $i$  at each time  $t$ , let's consider the covariance matrix of residuals  $\tilde{\Sigma}_t^{i,l} \in \mathbb{R}^{d \times d}$ , which represents a part of the overall covariance matrix. In this simulation, the algorithm (2.16) is compared with the

algorithm (2.10). To illustrate how the matrix  $\Psi$  affects the convergence of the algorithm, we consider two different cases:

$$\Psi^* = \text{diag}_{12}(\text{col}\{2, 2, 10, 10, 2, 2, \dots, 10, 10\}),$$

$$\Psi^{**} = \text{diag}_{12}(\text{col}\{0.2, 0.2, 100, 100, 0.2, 0.2, \dots, 100, 100\}).$$

Figure 3.7 shows how the average first diagonal element of the covariance matrix of residuals  $\tilde{\Sigma}_t^{i,l}$  changes over time. It can be seen that the new algorithm with  $\Psi = \Psi^*$  converges faster than the previous algorithm. However, the convergence rate depends on  $\Psi$ .

Let's consider a numerical experiment illustrating the operation of the distributed randomized stochastic optimization algorithm combined with the local voting protocol for estimating the locations of static objects (2.21).

Consider a distributed network consisting of five sensors. Let  $\mathcal{N} = \{1, 2, 3, 4, 5\}$  be the set of sensors. Each sensor has at most two active communication channels at any given time. There are 10 stationary targets within the region of interest. The task of the sensors is to determine the locations of these targets. At time  $t$ ,  $\mathbf{s}^i = [s^{i,1}, s^{i,2}]^T \in \mathbb{R}^2$  represents the current state of sensor  $i \in \mathcal{N}$ ,  $\mathbf{r}^l = [r^{l,1}, r^{l,2}]^T \in \mathbb{R}^2$  represents the state of target  $l \in \mathcal{M} = \{1, 2, \dots, 10\}$ , and  $\theta = \text{col}\{\mathbf{r}^1, \dots, \mathbf{r}^{10}\}$  represents the overall state of all targets.

In this simulation, only hybrid noise is considered, uniformly distributed around time-varying constants, for example,  $v_k^i = \pm 1 + 0.1 * \sin(k)$ , where the sign before 1 switches every 50th iteration.

According to Theorem 4, the step size parameter  $\alpha$  should be set to  $\frac{1}{k}$ . However, the algorithm (2.21), operating on each node with a parameter  $\alpha_k = \frac{1}{k^{1-\rho}}$  for all  $\rho > 0$ , exhibits more consistent convergence. In this simulation, the following parameters were chosen:  $\alpha_k = \frac{1}{k^{3/5}}$ ,  $\beta = 0.1$ ,  $\gamma = 1.0$ . The coordinates of the targets and sensors are random variables uniformly distributed in the intervals  $[0; 100]$  and  $[100; 120]$ , respectively.

For each target  $l$  and sensor  $i$  at each time  $t$ , let's consider the covari-

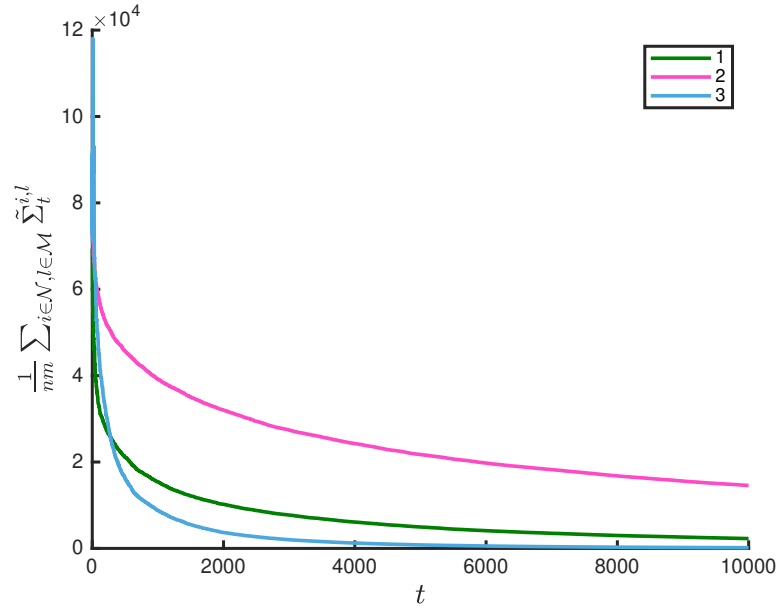


Figure 3.8: Average value of the first diagonal element of the covariance matrix of residuals  $\tilde{\Sigma}_t^{i,l}$ , averaged over all targets and sensors, where (1):  $\Psi_1 = I_{md}$ , (2):  $\Psi_1 = 0.5I_{md}$ , (3):  $\Psi_3 = 2I_{md}$

ance matrix of residuals  $\tilde{\Sigma}_t^{i,l} \in \mathbb{R}^{d \times d}$ , which represents a part of the overall covariance matrix. Figure 3.8 shows how the average value of the first diagonal element of the covariance matrix of residuals  $\tilde{\Sigma}_t^{i,l}$  changes over time for different matrices  $\Psi$ . It can be seen from Figure 3.8 that the new algorithm converges to a point.

## 3.2 Prototype of a System for Target Tracking by a Distributed Sensor Network

### 3.2.1 System Architecture

To implement the software product that implements the algorithms described in the previous chapter, software was developed<sup>1</sup> using the following

<sup>1</sup>Certificate of state registration of computer program No. 2022611354 “Prototype of a system for target tracking by a set of sensors under unknown-but-bounded noise” dated 24.01.2022 [22]

technologies:

- Programming language: MATLAB;
- Graphical user interface tool: MATLAB App Designer;
- Standalone application creation tool: MATLAB Application Compiler.

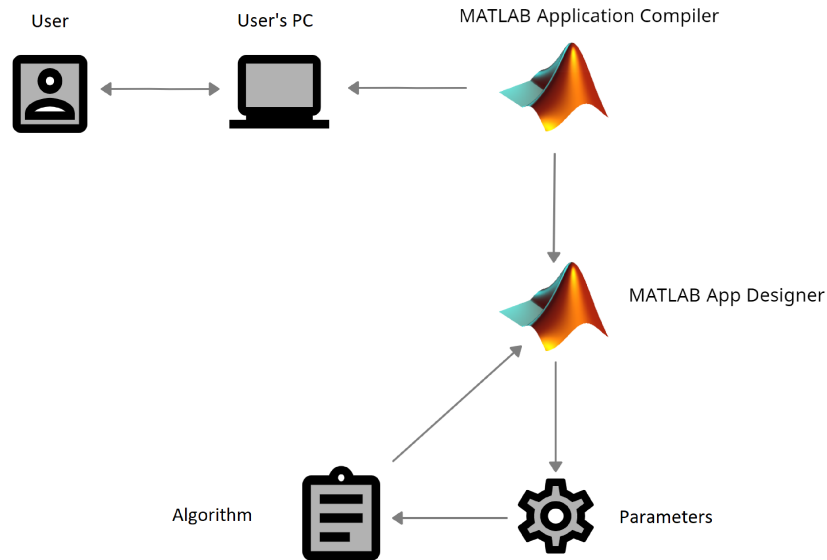


Figure 3.9: Diagram of the system architecture

The system is an independent user application, whose architecture is presented in Figure 3.9. The user enters parameters through the MATLAB App Designer graphical interface. The generated model is then input to the algorithm. After the computations, MATLAB App Designer provides the algorithm's results through the graphical interface, including a graph showing the variance dependency on the number of iterations, a graph showing the covariance matrix elements dependency on the number of iterations, a graph showing the movement of targets and sensors, and a graph showing the distribution of noise. By using MATLAB Application Compiler, all the necessary libraries can be packaged into a standalone application that does not require MATLAB installation to run on an end user's personal computer (PC).

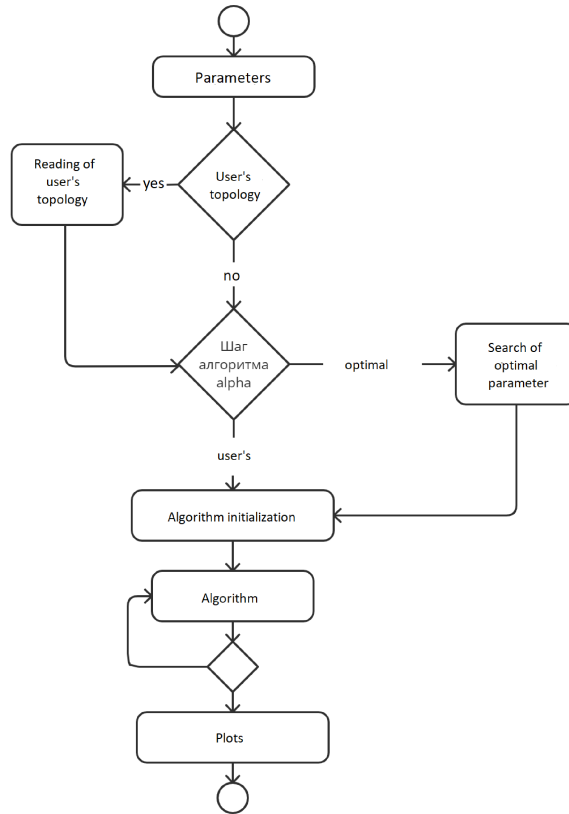


Figure 3.10: System activity diagram

The overall system workflow is depicted in Figure 3.10 as an activity diagram. The system receives input parameters specified in the graphical interface. Next, it checks whether the user has chosen to define an arbitrary sensor network topology. If yes, an interface for selecting such a topology is displayed and subsequently read. If not, a randomized topology is used. Then, it checks whether the user has chosen to calculate the optimal parameter (the algorithm's step)  $\alpha$ . If yes, the system performs a search for such a parameter. If not, the value entered by the user is read. The algorithm itself is then initialized, and the iterative computation of the minimum point estimate value is performed, repeating for the specified number of times (iterations). On each iteration, the root mean square error and error covariance matrix are computed. After the loop completes, the results are displayed as graphs of the algorithm's root mean square estimation errors and error covariance matrix elements.

### 3.2.2 Graphical Interface

The graphical interface is depicted in Fig. 3.11, 3.12, 3.13, 3.14. The graphical interface is designed as four tabs with a common fixed area. A detailed description of the system's graphical interface implementation is provided below.

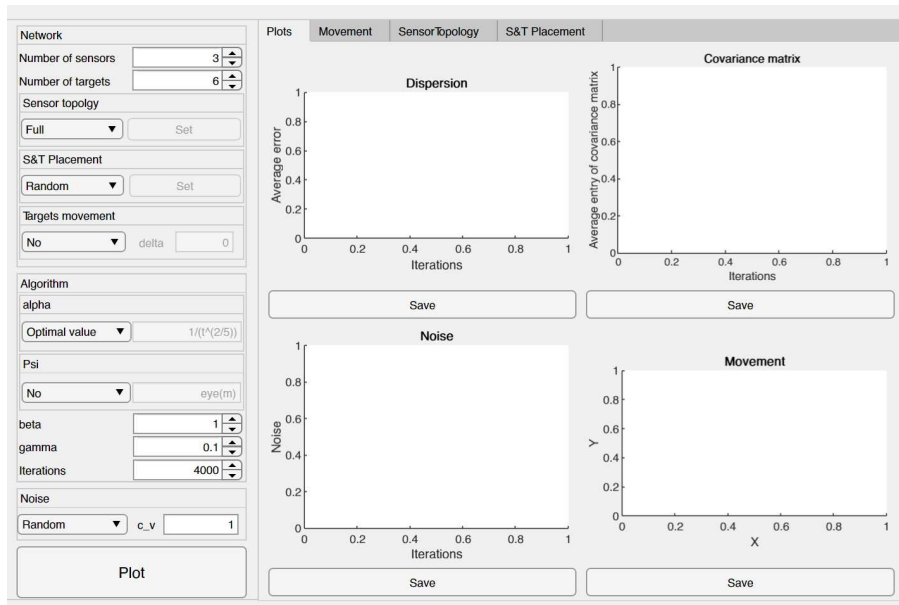


Figure 3.11: System graphical interface. Image of the first tab.

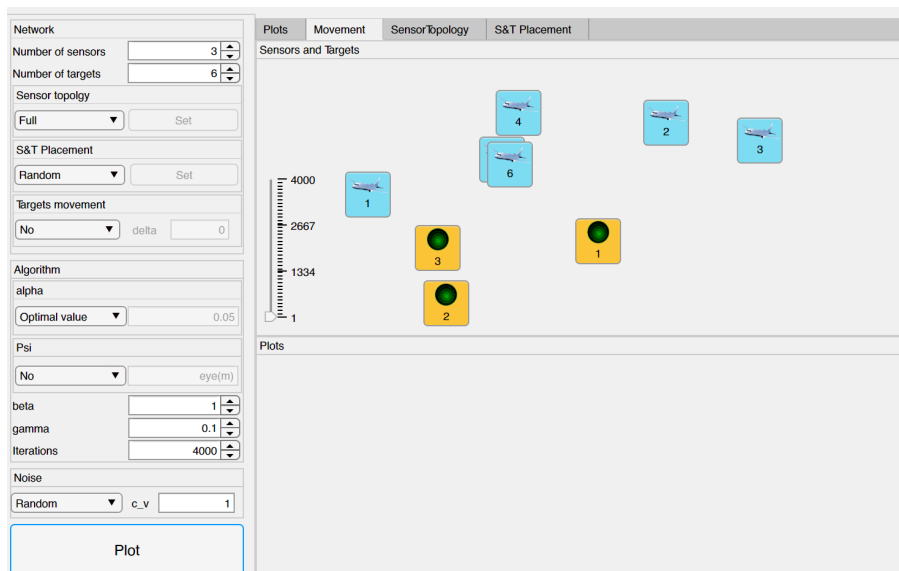


Figure 3.12: System graphical interface. Image of the second tab.

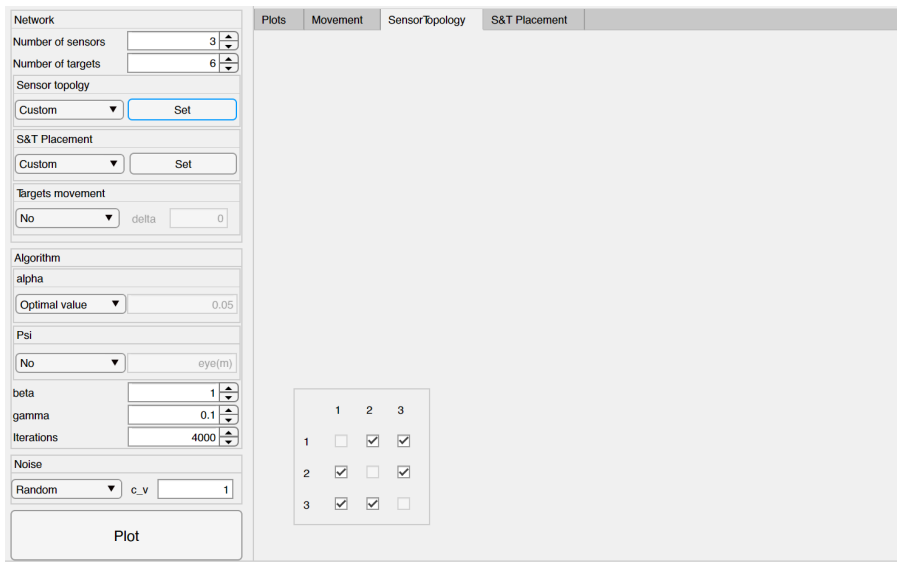


Figure 3.13: System graphical interface. Image of the third tab.

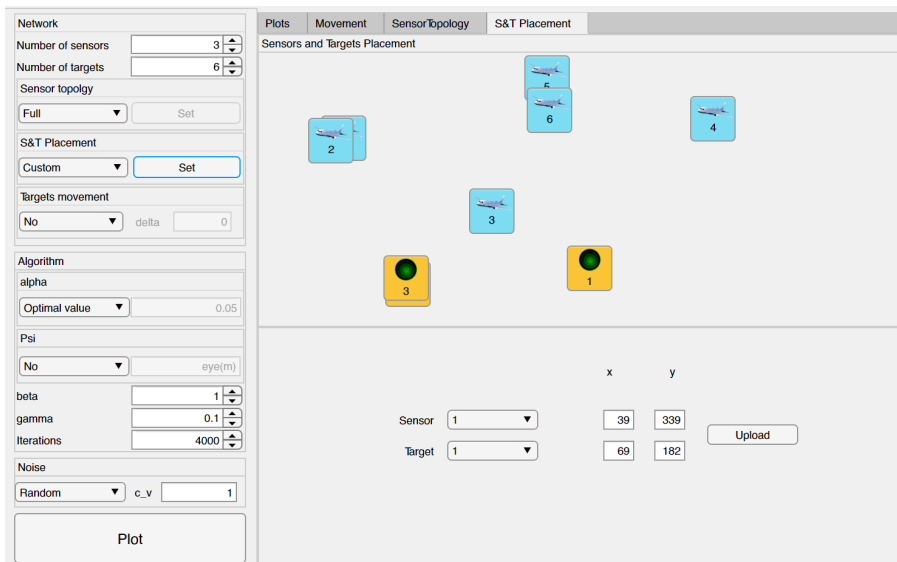


Figure 3.14: System graphical interface. Image of the fourth tab.

The overall area is divided into four blocks based on their purpose. The first block, “Network,” allows the user to specify the number of sensors in the system (“Number of sensors”) and the number of targets (“Number of targets”). In this block, the user can also select the sensor network topology, which can be either complete (“Full”) or custom (“Custom”). The “Set” button in the “Sensor topology” panel becomes active when the custom topology is chosen. Upon clicking it, the third tab, “Sensor Topology” (Fig. 3.13), displays a field for defining the topology. If  $\forall$  is present at the intersection

of the  $i$ -th column and  $j$ -th row, it means that sensor  $i$  can receive data from sensor  $j$ . It is worth noting that the number of neighbors for each sensor should be less than the space dimension ( $d = 2$ ), but it cannot be zero. Thus, if the selected number of neighbors becomes zero, the system displays an error. In the first block, the user can also specify the target motion models: no movement (“No”), drift with a given constraint  $\delta$  (“Drift”):  $r_t^l = r_{t-1}^l + \chi_{t-1}^l$ , where  $\chi_{t-1}^l$  is a random vector uniformly distributed on a sphere with radius  $\delta$ , or an arbitrary function depending on the number of targets  $m$  and the time moment  $t$  (“Custom”). Additionally, the user can either manually set the initial locations of sensors and targets or make them random. The “Set” button in the “S&T Placement” panel becomes active when the first option is chosen. After clicking it, the user can proceed to the fourth tab, “S&T Placement” (Fig. 3.14), where the interface for the current sensor and target locations will be displayed. By selecting a sensor and a target number, entering the values for the OX and OY axes in the respective fields, and clicking the “Upload” button, the user can upload the data into the system.

The algorithm parameters are set in the second block, “Algorithm.” The value of the parameter  $\alpha$  can be either optimal (“Optimal”) or custom (“Custom”), depending on the time moment  $t$ . For solving stationary problems, the optimal alpha is calculated as  $\alpha(t) = \frac{1}{t^{2/5}}$  [12, 47]. For solving non-stationary problems, alpha is computed according to Theorem 2. The calculated value is placed in the corresponding window. The matrix  $\Psi$  can be set as identity (“No”) or with a custom value (“Custom”) depending on the number of targets  $m$ . The user can also specify parameters such as  $\beta$ ,  $\gamma$ , and the number of iterations in this block.

The third block involves specifying the noise. There are three types of noise to choose from: random (“Random”), constant arbitrary value (“Constant”), and oscillations (“Oscillations”). It is also possible to define a custom type of noise (“Custom”) that depends on the time moment  $t$ . The noise will be limited by the number entered in the field “c\_v”.



The fourth block contains the “Plot” button, which starts the execution of the algorithm with the specified parameters.

The first tab (Fig. 3.11) presents the graphs that show the results of the algorithm. In the top left corner, there is a graph depicting the averaged root mean square error as a function of the number of iterations for all sensors and targets. In the top right corner, there is a graph showing the elements of the covariance matrix averaged over all sensors and targets as a function of the number of iterations. In the bottom left corner, there is a graph illustrating the noise as a function of the number of iterations. In the bottom right corner, there is a plot displaying the sensor locations (circles), target movements (dots), and target location estimates by different sensors (plus signs). Each of the mentioned graphs can be saved in EPS (Encapsulated PostScript) format by clicking the “Save” button.

The second tab (Fig. 3.12) consists of two parts. The upper part is used to visualize the locations of sensors and targets. The moving targets are represented by airplane images, and the sensors are represented by radar images. On the left, there is a slider that represents the number of iterations. It allows tracking the movement of targets by adjusting the slider to a specific time moment. Moreover, the system implements the following interactive mode: clicking on a specific target displays a graph in the lower part of the tab, showing the dependency of the root mean square error on the number of iterations for each sensor. A similar mode is also implemented for sensors, where selecting a specific sensor shows a graph depicting the dependency of the root mean square error on the number of iterations for each target.

### 3.2.3 Examples of System Operation

This subsection presents the results of the system operation on some examples.

1. Stationary problem without considering disturbances.

The first example, shown in Fig. 3.15, demonstrates the result of the sys-

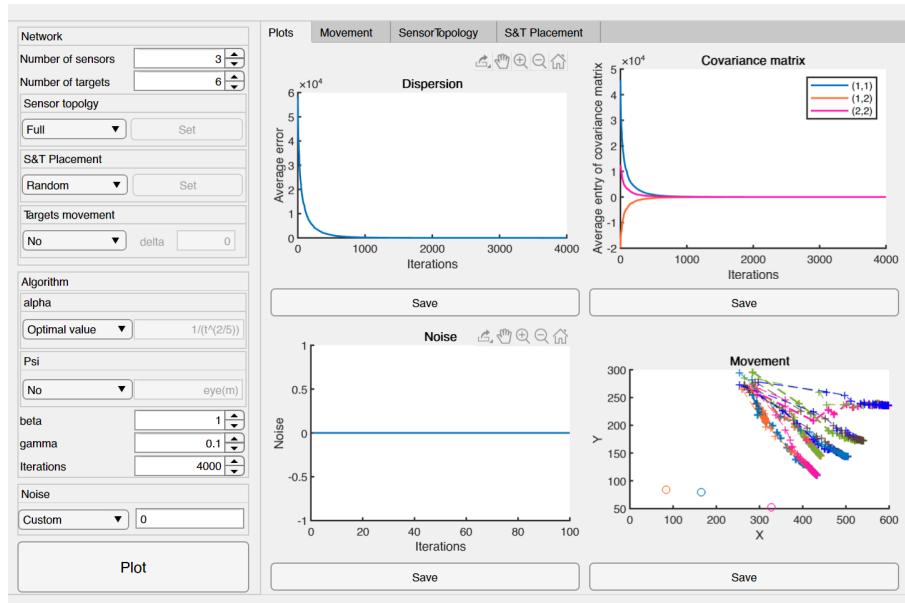


Figure 3.15: Example of system operation for solving a stationary problem without considering disturbances. Image of the first tab

tem operation in the absence of target motion and any disturbances obtained by the sensors observing the targets. In this simulation, three sensors estimate the locations of six stationary targets. The sensor topology is chosen as complete, and the locations of sensors and targets are randomly selected. The algorithm step size  $\alpha$  is chosen as optimal and calculated according to Theorem 2. Other algorithm parameters are set as follows:  $\beta = 1$ ,  $\gamma = 0.1$ , and the number of iterations is 4000. Since this is a case of stationary deterministic optimization, the graph of the mean square error dependence shows that the algorithm converges to zero, which means that the sensors accurately determine the locations of the targets.

## 2. Stationary problem with disturbances in the form of oscillations.

Fig. 3.16 shows the result of the system operation for solving a stationary problem. In contrast to the previous example, in this case, the sensors receive observations of targets with disturbances in the form of periodic oscillations, limited in magnitude by  $c_v = 1$ . The distribution of such disturbances is shown in the lower left graph in Fig. 3.16. The algorithm parameters coincide with those from the previous example. From the graph of the mean square error dependence, it can also be seen that the algorithm

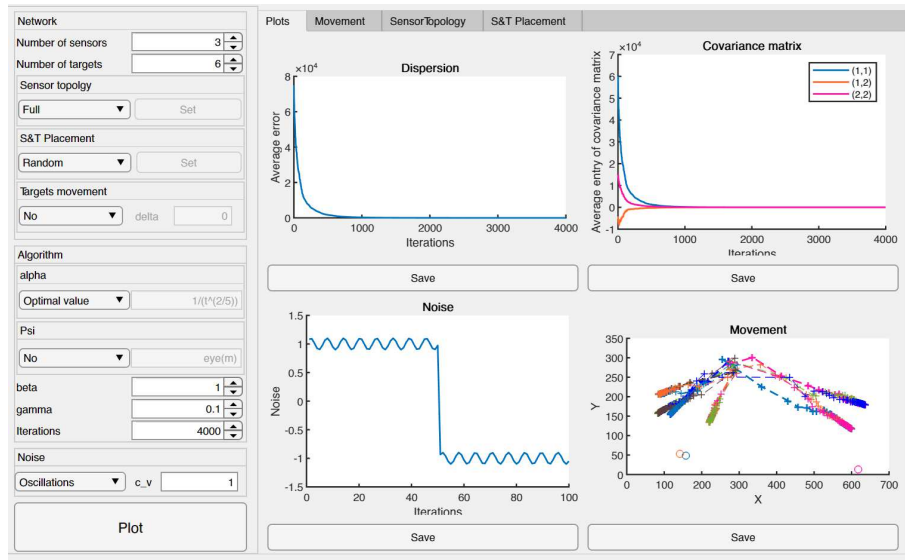


Figure 3.16: Example of system operation for solving a stationary problem with disturbances in the form of oscillations. Image of the first tab

converges even in the presence of such disturbances.

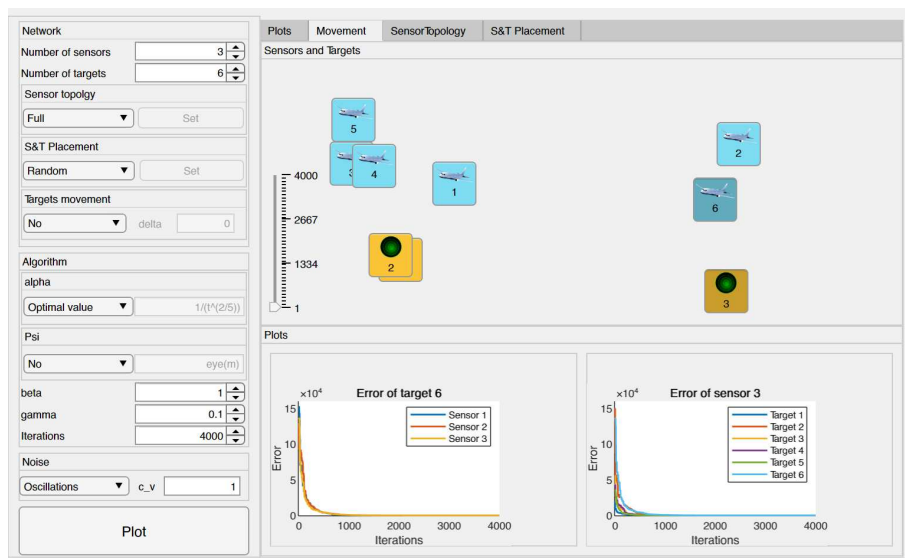


Figure 3.17: Example of system operation for solving a stationary problem with disturbances in the form of oscillations. Image of the second tab

The appearance of the second tab is shown in Fig. 3.17. On the upper panel, sensors and targets are located according to their actual randomly

chosen positions. In this example, buttons corresponding to the third sensor and the sixth target are pressed, and the corresponding graphs of the mean square error for the selected objects are drawn. As described earlier, adding such an interactive mode is necessary for further algorithm exploration, namely to understand which sensor is better to monitor which target.

. 3. Non-stationary problem with specifying a user-defined sensor topology and random disturbances.

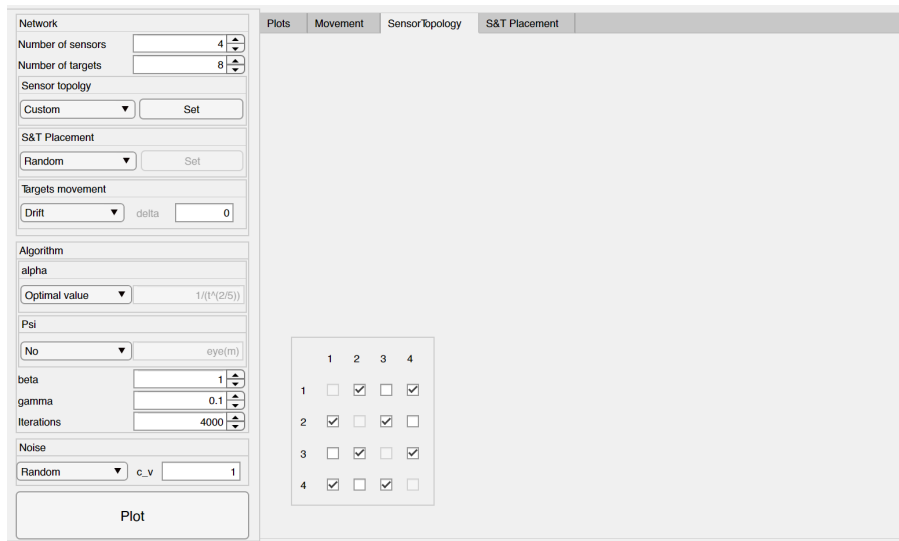


Figure 3.18: Example of system operation for solving a non-stationary problem with specifying a user-defined sensor topology and random disturbances. Image of the third tab

In this example, four sensors track the movement of eight targets. The sensor network topology is specified according to Fig. 3.18. At each time step, the sensors randomly select at most one neighbor to communicate with based on this topology. The target movement model is defined as a drift with  $\delta = 2$ . The sensors receive observations of target locations with random disturbances. From the plots shown in Fig. 3.19, it can be seen that the algorithm converges.

4. Non-stationary problem with specifying initial sensor and target locations and constant disturbances.

In the next example, five sensors track ten targets. The sensors receive observations with disturbances in the form of a constant, bounded by  $c_v = 3$ .

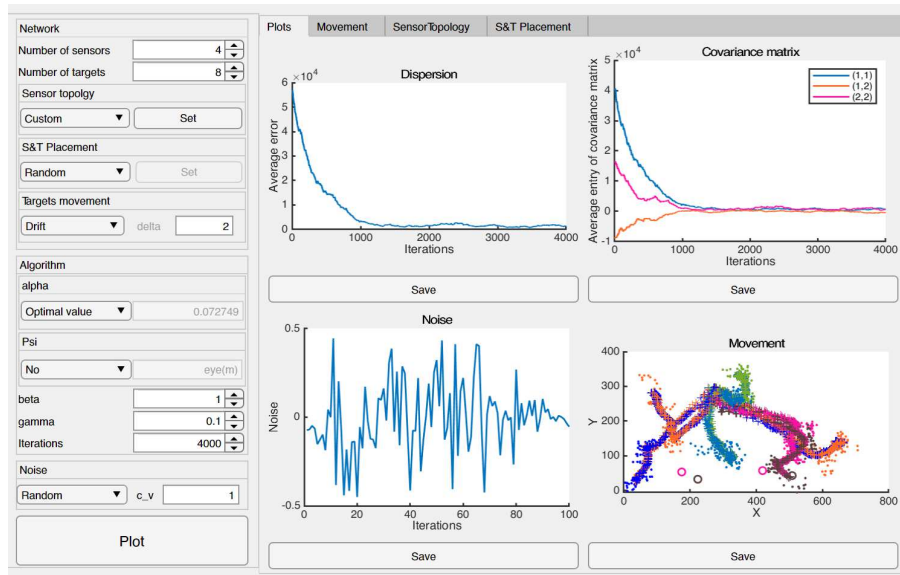


Figure 3.19: Example of system operation for solving a non-stationary problem with specifying a user-defined sensor topology and random disturbances. Image of the first tab

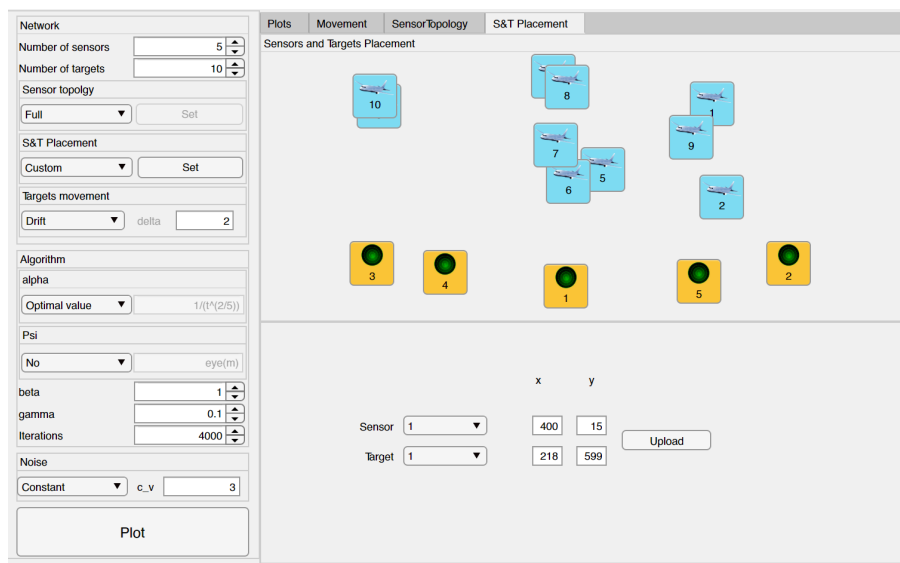


Figure 3.20: Example of system operation for solving a non-stationary problem with specifying initial sensor and target locations and constant disturbances. Image of the fourth tab

The initial sensor and target locations are specified using the graphical interface (Fig. 3.20). The results of the system operation are shown in Fig. 3.21. As in previous cases, the plots confirm the convergence of the algorithm. It should be noted that the bottom-right plot “Movement” becomes less in-

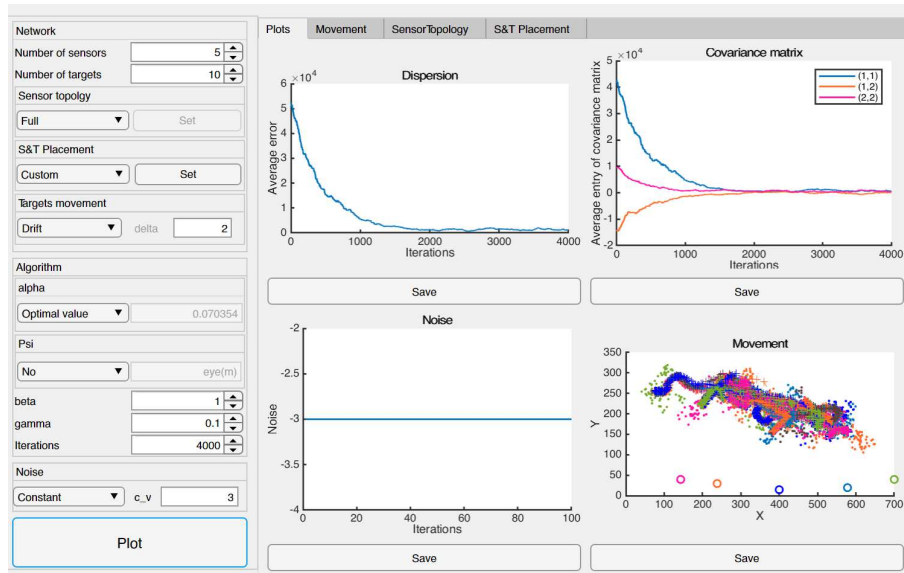


Figure 3.21: Example of system operation for solving a non-stationary problem with specifying initial sensor and target locations and constant disturbances. Image of the first tab

formative when the number of targets and sensors exceeds four and eight, respectively. For a more detailed study of target movement and estimation of their locations, the user is advised to use the second tab “Movement”.

5. Non-stationary problem with specifying arbitrary target movement function and the matrix  $\Psi$  considering random disturbances.

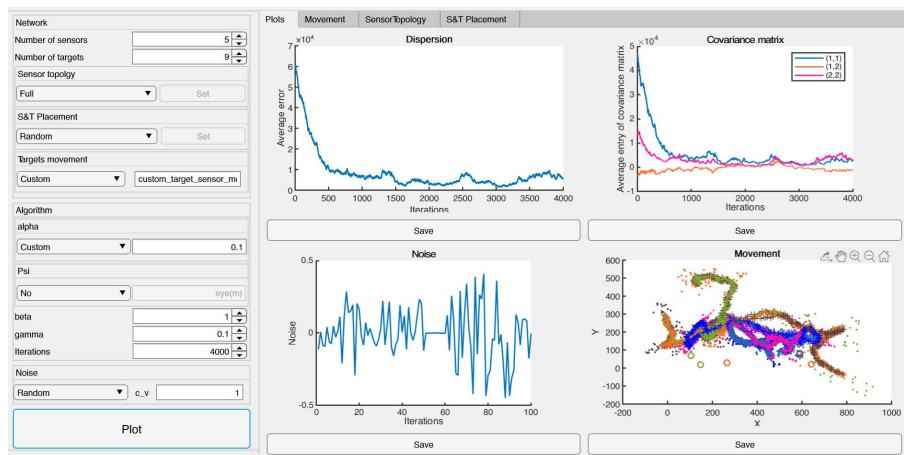


Figure 3.22: Example of system operation for solving a non-stationary problem with specifying an arbitrary target movement function and the matrix  $\Psi$ . In this simulation, the matrix  $\Psi$  is an identity matrix

In this example, five sensors observe nine targets. They receive observa-

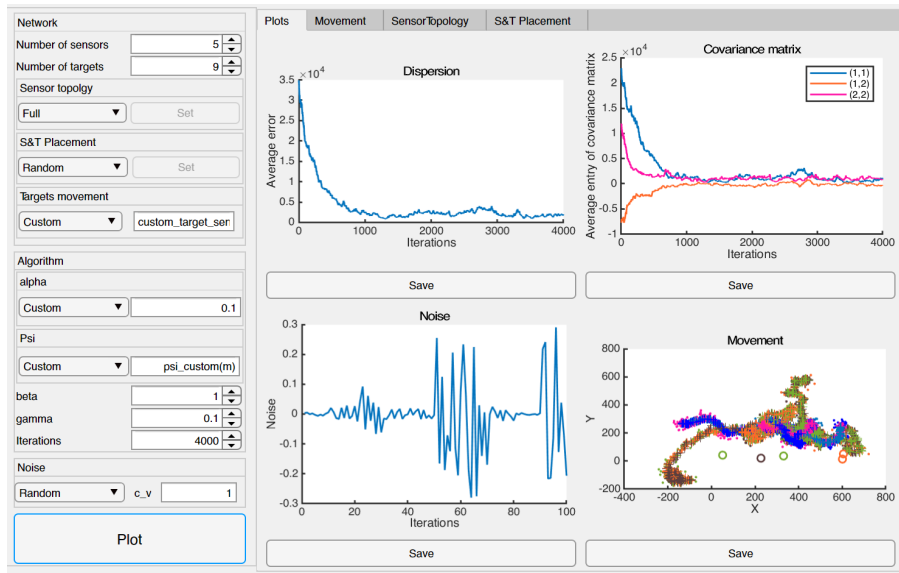


Figure 3.23: Example of system operation for solving a non-stationary problem with specifying an arbitrary target movement function and the matrix  $\Psi$

tions with random disturbances bounded by  $c_v = 1$ . In [121], it is described that the algorithm with the addition of the matrix  $\Psi$  will converge faster if the targets have different movement characteristics. To verify this assumption, the target movement is defined as the function

$$\text{custom\_target\_sensor\_movement}(m, \text{Time}),$$

in such a way that the first four targets have a drift model with  $\delta = 3$ , while the other five have  $\delta = 5$ .

Fig. 3.22 shows the results of the system operation with the identity matrix  $\Psi$ , and Fig. 3.23 shows the results with a user-defined matrix  $\Psi$ . As seen from the plots of mean square errors and covariance matrix elements, the algorithm with the addition of the matrix  $\Psi$  indeed converges faster.

### 3.3 Practical Application of Randomized Stochastic Approximation Algorithm Combined with Local Voting Protocol

One practical application of the randomized stochastic approximation algorithm combined with a local voting protocol is in the field of tracking, specifically the tracking of aircraft movement. Ensuring the safety of air traffic requires monitoring all aircraft in the airspace. In the past, primary radar systems were used to track aircraft, which could only provide distance information to the aircraft fuselage. Secondary radar systems have now become the primary means of tracking aircraft. However, these systems have their drawbacks, including high costs and the need for fixed radar installations, which can often exceed the cost of the radar equipment itself. Therefore, deploying an adequate number of secondary radars to address various tracking requirements is not always feasible. Additionally, these radar systems are not mobile due to their large size and weight.

In recent years, multi-position surveillance systems (MPSS) have been actively implemented in Russian airports [21]. An MPSS is a distributed system that uses the time of arrival of signals from aircraft at reception stations within its coverage area to determine their locations [13]. MPSS offers several key advantages [70]: the coverage area of an MPSS is continuous compared to the cone-shaped coverage area of a secondary radar; the individual positions of an MPSS are relatively inexpensive, enabling the deployment of systems for surveillance, including at low and very low altitudes; and MPSS equipment is lightweight (5-8 kg). As a result, these systems are theoretically mobile.

Considering the increasing number of aircraft and the volume of data transmitted by each aircraft, such as the aircraft's call sign, indicated airspeed, Mach number, etc., particularly in the case of commercial airliners, the data transmission over communication channels becomes heavily loaded.



Moreover, due to terrain features and other interference factors, some stations may fail to accurately determine the position of an aircraft and transmit false information to the central processing facility. In cases where data from multiple sensors do not match, the aircraft's position estimation process starts over.

The developed system for target tracking using a sensor network and a distributed randomized stochastic optimization algorithm combined with a local voting protocol can significantly improve the accuracy of aircraft location estimation. Furthermore, by utilizing a distributed sensor network, the amount of data transmitted over communication channels is reduced. All these factors can lead to faster and more accurate tracking of aircraft [31].

Moreover, most machine learning algorithms and systems for estimating (identifying) unknown parameters and control strategies in networked systems involve optimization methods. The current development in this field aims to address several challenges, including complex types of objective functions (non-convex, non-smooth, non-convex integer and combinatorial optimization), large-scale data (high-dimensional parameter vectors), real-time optimization, and dealing with uncertainties. The proposed algorithm can be used to tackle such problems.

### 3.4 Summary

The third chapter presents the results of simulation modeling that illustrate the performance of different versions of the modified randomized stochastic approximation algorithm combined with the local voting protocol. It also describes a prototype system for distributed tracking, for which a certificate of registration of the computer program has been obtained. The prototype system is implemented in the MATLAB programming language.

While the primary application of the randomized stochastic approximation algorithm combined with the local voting protocol is the tracking of

aircraft movement, the algorithm can also be used in machine learning, identification of unknown parameters, and other areas.

## Conclusion

Main scientific results of the dissertation research achieved within the framework of the set tasks:

- 1) a sensor network control model based on randomized and multi-agent approaches was developed (Chapter 1);
- 2) the distributed randomized optimization algorithm combined with a local voting protocol was modified, and the properties of its estimations for the tracking problem (tracking parameter changes) were investigated using the developed sensor network control model (Chapter 2);
- 3) the applicability conditions of the modified distributed randomized optimization algorithm combined with a local voting protocol were investigated for the network observation model. The results were tested in the problem of observing moving objects in the presence of uncertainties and constraints on the number of connections between sensors (Chapters 2, 3).

# Bibliography

- [1] Agaev R.P., Chebotarev P.Yu. Matrix of maximum out-forests of a directed graph and its applications // Automation and Remote Control. — 2000. — № 9. — C. 15–43.
- [2] Andrievskiy B.R., Matveev A.S., Fradkov A.L. Control and estimation under information constraints: towards a unified theory of control, computation, and communication // Automation and Remote Control. — 2010. — T. 2010, № 4. — C. 34–99.
- [3] Chebotarev P.Yu., Agaev R.P. Coordination of characteristics in multi-agent systems and spectra of Laplacian matrices of oriented graphs // Automation and Remote Control. — 2009. — № 3. — C. 136–151.
- [4] Derevitskiy D.P., Fradkov A.L. Two Models for the Analysis of Adaptation Algorithm Dynamics // Automation and Remote Control. — 1974. — № 1. — C. 59–67.
- [5] Derevitskiy D.P., Fradkov A.L. Applied Theory of Discrete Adaptive Control Systems. — Moscow: Nauka, 1981. — C. 216.
- [6] Ermakov S.M. Monte Carlo Method and Related Issues. — Moscow: Nauka, 1971. — C. 471.
- [7] Erofeeva V.A., Sergeenko A.N., Tursunova M.B. Accelerated Randomized Stochastic Approximation Algorithm for the Tracking Problem // Navigation and Motion Control. Proceedings of the XXIV Conference of Young Scientists with International Participation. — 2022. — C. 143–145.
- [8] Gorodetsky V.I., Granichin O.N., Skobelev P.O. Decentralization, Self-Organization, and Emergent Intelligence - Digital Explosion of Smart Technologies // MKPU-2022, St. Petersburg. — 2022.

- [9] Gorodetsky V.I., Skobelev P.O. Multi-agent technologies for industrial applications: reality and prospects // Proceedings of SPIIRAS. — 2017. — T. 55. — C. 11–45.
- [10] Granichin O.N. Stochastic approximation procedure with input perturbation // Automation and Remote Control. — 1992. — T. 2. — C. 97–104.
- [11] Granichin O.N., Fomin V.N. Adaptive control using probing signals in the feedback channel // Automation and Remote Control. — 1986. — T. 2. — C. 100–112.
- [12] Granichin O.N., Polyak B.T. Randomized Estimation and Optimization Algorithms in the Presence of Arbitrary Noise. — russian Federation : Publishing House “Nauka”, 2003.
- [13] International Civil Aviation Organization. — Manual on Aviation Surveillance, Second Edition, 2017.
- [14] Ivanskiy Yu.V., Sergeenko A.N. Distributed Algorithms for Tracking the Trajectories of a Set of Objects using an Array of Mobile Sensors // Navigation and Motion Control. Abstracts of the International Seminar. Edited by I.V. Belokonov. — 2020. — C. 22.
- [15] Karpov M.I., Petrosyan L.A. Cooperative solutions in communication networks // Bulletin of St. Petersburg University. Series 10: Applied Mathematics, Informatics, Control Processes. — 2012. — C. 37–45.
- [16] Polyak B.T. Convergence and convergence rate of iterative stochastic algorithms. II. Linear case // Automat. Remote Control. — 1977. — T. 38, № 4. — C. 537–542.
- [17] Polyak B.T, Tsibakov A.B. Optimal accuracy orders of stochastic optimization search algorithms // Problems of Information Transmission. — 1990. — T. 26, № 2. — C. 45–53.
- [18] Protocol of Cooperative Self-Organization for Group Behavior of Robots / K.S. Amelin, V.A. Erofeeva, O.N. Granichin и др. // XIV All-russian Multiconference on Control Problems MKPU-2021. Proceedings of the XIV Multiconference. — 2021. — C. 94–96.
- [19] Randomization of Connections in Multi-Agent Systems to Satisfy Cost Constraints / V.A. Erofeeva, O.N. Granichin, Yu.V. Ivanskiy,

- A.N. Sergeenko // XIV All-russian Multiconference on Control Problems MKPU-2021. Proceedings of the XIV Multiconference. — 2021. — C. 48–50.
- [20] Rastrigin L.A. Statistical Search Methods. — Moscow: Nauka, 1968. — C. 376.
- [21] Scientific and Production Enterprise "Digital Radiotechnical Systems". — URL: [npp-crts.ru](http://npp-crts.ru).
- [22] Sergeenko A.N. Prototype System for Tracking Targets with a Set of Sensors under Unknown but Bounded Interference. — Certificate of Computer Program Registration No. 2022611354 dated 24.01.22.
- [23] Sergeenko A.N. Comparing the Complexity of the Brute-Force Method and the DNA Computing-Based Algorithm in Solving the Traveling Salesman Problem // Science Week at SPbPU. Proceedings of the International Scientific Conference. Best Reports. — 2018. — C. 167–169.
- [24] Sergeenko A.N. The Traveling Salesman Problem: Comparing the Complexity of the Brute-Force Method with an Algorithm Based on DNA Computing // Science Week at SPbPU. Proceedings of the International Scientific Conference. Institute of Physics, Nanotechnology, and Telecommunications. — 2018. — C. 376–378.
- [25] Sergeenko A.N. Application of DNA Computing-Based Algorithm for Solving an Optimization Problem // Navigation and Motion Control. Proceedings of the XI Conference of Young Scientists with International Participation. Edited by V.G. Peshkhonov. — 2019. — C. 274–276.
- [26] Sergeenko A.N. DNA Computing for the Hamiltonian Path Problem // Self-organization and Artificial Intelligence in Groups of Autonomous Robots: Methodology, Theory, Practice. Collective Monograph. Edited by O.N. Granichin, S.F. Sergeev. — 2020. — C. 43–50.
- [27] Sergeenko A.N. DNA Computing for the Hamiltonian Path Problem // Stochastic Optimization in Computer Science. — 2020. — T. 16, № 1. — C. 40–47.
- [28] Sergeenko A.N. Distributed Tracking of a Large Number of Aircraft in Uncertain Conditions // Navigation and Motion Control. Proceedings of the XII Conference of Young Scientists with International Participation. — 2020. — C. 319–321.

- [29] Sergeenko A.N. Trajectory Tracking Algorithm of a Set of Objects by a Network of Mobile Sensors // Mathematical Control Theory and its Applications (MTUIP-2020). Conference Proceedings. State Scientific Center of the Russian Federation AO “Concern “CSRI “Elektropribor”. — 2020. — C. 130–132.
- [30] Sergeenko A.N. Upper Bounds of Estimates Obtained by a Combined Randomized Stochastic Approximation Algorithm and Local Voting Protocol for the Tracking Problem // Navigation and Motion Control. Proceedings of the XIII Conference of Young Scientists with International Participation. — 2021. — C. 255–257.
- [31] Sergeenko A.N. Development of Technology for Tracking Aircraft under Uncertainty // Young Scientist. — 2022. — T. 35, № 325. — C. 3–6.
- [32] Sergeenko A.N., Granichin O.N. DNA Computing as a Solution for the Traveling Salesman Problem // Eighteenth National Conference on Artificial Intelligence with International Participation KII-2020. Conference Proceedings. Edited by V.V. Borisov, O.P. Kuznetsov. — 2020. — C. 137–144.
- [33] Sergeenko A.N., Granichin O.N. Sensor Network Control Problem Based on Randomized and Multi-Agent Approaches and Its Applications // Computer Tools in Education. — 2022. — C. 94–107.
- [34] Accelerated Simultaneous Perturbation Stochastic Approximation for Tracking Under Unknown-but-Bounded Disturbances / V. Erofeeva, O. Granichin, Munira Tursunova et al. // 2022 American Control Conference (ACC). — 2022. — P. 1582–1587.
- [35] Adleman L.M. Molecular computation of solutions to combinatorial Problems // Science, New Series. — 1994. — Vol. 266, no. 5187. — P. 1021–1024.
- [36] Amelin K.S., Granichin O.N. Multiagent network control for a group of light UAVs // Neurocomputers: Design & Application. — 2011. — Vol. 2011, no. 6. — P. 64–72.
- [37] Amelina N., Fradkov A. Approximate consensus in the dynamic stochastic network with incomplete information and measurement delays // Automation and Remote Control. — 2012. — Vol. 73, no. 11. — P. 1765–1783.

- [38] Approximate consensus in stochastic networks with application to load balancing / N. Amelina, A. Fradkov, Y. Jiang, D.J. Vergados // IEEE Transactions on Information Theory. — 2015. — Vol. 61, no. 4. — P. 1739–1752.
- [39] Barricelli N. A. Esempi numerici di processi di evoluzione // Methodos. — 1954. — P. 45–68.
- [40] Barricelli N. A. Numerical testing of evolution theories. Part II. Preliminary tests of performance, symbiogenesis and terrestrial life // Acta Biotheoretica. — 1963. — P. 99–126.
- [41] Bhattacharya K., Vicsek T. Collective decision making in cohesive flocks // New Journal of Physics. — 2010. — Vol. 12, no. 9. — P. 093019.
- [42] Blackman S. S. Multiple hypothesis tracking for multiple target tracking // IEEE Aerospace and Electronic Systems Magazine. — 2004. — Vol. 19, no. 1. — P. 5–18.
- [43] Borkar Vivek S. Stochastic Approximation: A Dynamical Systems Viewpoint. — Cambridge : Cambridge University Press, 2008.
- [44] Bullo F. Lectures on Network Systems.— published online at <http://motion.me.ucsb.edu/book-lns>, 2016. — With contributions by J. Cortes, F. Dorfler, and S. Martinez.
- [45] Bullo F., Cortés J., Martínez S. Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms. — 2009. — 07. — P. 333.
- [46] Consensus-based Distributed Algorithm for Multisensor-Multitarget Tracking under Unknown-but-Bounded Disturbances / N. Amelina, V. Erofeeva, O. Granichin et al. // IFAC-PapersOnLine. — 2020. — Vol. 53, no. 2. — P. 3589–3595. — 21th IFAC World Congress.
- [47] Convergence Analysis of Weighted SPSA-based Consensus Algorithm in Distributed Parameter Estimation Problem / A. Sergeenko, V. Erofeeva, O. Granichin et al. // Proc. of the 19th IFAC SYSID: learning models for decision and control. — Vol. 54. — 2021. — P. 126–131.
- [48] Cooperative Control of Multi-agent Systems: Optimal and Adaptive Design Approaches / F. L. Lewis, H. Zhang, K. Hengster-Movric, A. Das. — Springer Science & Business Media, 2013.



- [49] DeGroot M. H. Reaching a Consensus // Journal of the American Statistical Association. — 1974. — Vol. 69. — P. 118–121.
- [50] Del Moral P., Doucet A. Particle Motions in Absorbing Medium with Hard and Soft Obstacles // Stochastic Analysis and Applications. — 2004. — Vol. 22, no. 5. — P. 1175–1207.
- [51] Differentiated consensus in decentralized load balancing problem with randomized topology, noise, and delays / N. Amelina, O. Granichin, O. Granichina, Y. Jiang // 53rd IEEE Conference on Decision and Control / IEEE. — 2014. — P. 6969–6974.
- [52] Distributed optimization and statistical learning via the alternating direction method of multipliers / S. Boyd, N. Parikh, E. Chu et al. // Foundations and Trends® in Machine learning. — 2011. — Vol. 3, no. 1. — P. 1–122.
- [53] Dorigo M., Maniezzo V., Colorni A. Ant system: optimization by a colony of cooperating agents // IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). — 1996. — Vol. 26, no. 1. — P. 29–41.
- [54] Dynamic Voltage-Frequency Optimization using Simultaneous Perturbation Stochastic Approximation / E. Bogdanov, A. Bozhnyuk, D. Bykov et al. // 2021 60th IEEE Conference on Decision and Control (CDC). — 2021. — P. 3774–3779.
- [55] Emergent intelligence viaself-organization in group of robotics devices / K. Amelin, O. Granichin, A. Sergeenko, Z.V. Volkovich // Mathematics. — 2021. — Vol. 9, no. 12. — P. 1314.
- [56] Epidemic Algorithms for Replicated Database Maintenance / A. Demers, D. Greene, C. Hauser et al. // Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing. — PODC '87. — New York, NY, USA : Association for Computing Machinery, 1987. — P. 1–12.
- [57] Equation of State Calculations by Fast Computing Machines / N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth et al. // The Journal of Chemical Physics. — 1953. — Vol. 21, no. 6. — P. 1087–1092.
- [58] Erofeeva V., Granichin O., Granichina O. Multi-sensor task assignment using linear matrix inequalities in the multiple target tracking

- problem // IFAC-PapersOnLine. — 2018. — Vol. 51, no. 15. — P. 880–885.
- [59] Erofeeva V., Granichin O., Sergeenko A. Distributed Stochastic Optimization with Heavy-Ball Momentum Term for Parameter Estimation // 2021 5th Scientific School Dynamics of Complex Networks and their Applications (DCNA). — 2021. — P. 69–72.
- [60] Erofeeva V., Sergeenko A., Granichin O. Accelerated Online Distributed Optimization for Parameter Estimation Under Uncertainties // 2022 6th Scientific School Dynamics of Complex Networks and their Applications (DCNA). — 2022. — P. 244–247.
- [61] Fax J.A., Murray R.M. Information flow and cooperative control of vehicle formations // IEEE Transactions on Automatic Control. — 2004. — Vol. 49, no. 9. — P. 1465–1476.
- [62] Glover F. Future Paths for Integer Programming and Links to Artificial Intelligence // Computers and Operations Research. — 1986. — Vol. 13, no. 5.
- [63] Granichin O. Linear regression and filtering under nonstandard assumptions (arbitrary noise) // IEEE Transactions on Automatic Control. — 2004. — Vol. 49, no. 10. — P. 1830–1837.
- [64] Granichin O., Amelina N. Simultaneous perturbation stochastic approximation for tracking under unknown but bounded disturbances // IEEE Transactions on Automatic Control. — 2015. — Vol. 60, no. 6. — P. 1653–1658.
- [65] Granichin O., Volkovich V., Toledano-Kitai D. Randomized algorithms in automatic control and data mining. — Springer, 2015. — Vol. 67.
- [66] Grant M., Boyd S. CVX: Matlab software for disciplined convex programming, version 2.1. — 2014.
- [67] Hlinka O., Hlawatsch F., Djuric P. M. Distributed particle filtering in agent networks: A survey, classification, and comparison // IEEE Signal Processing Magazine. — 2013. — Vol. 30, no. 1. — P. 61–81.
- [68] Huang M. Stochastic approximation for consensus: a new approach via ergodic backward products // IEEE Transactions on Automatic Control. — 2012. — Vol. 57, no. 12. — P. 2994–3008.

- [69] Huang M., Manton J. H. Coordination and Consensus of Networked Agents with Noisy Measurements: Stochastic Algorithms and Asymptotic Behavior // *SIAM Journal on Control and Optimization*. — 2009. — Vol. 48, no. 1. — P. 134–161.
- [70] ICAO. — Multilateration (MLAT) Concept of use, Edition 1.0 edition, 2007.
- [71] Isaksson A. J., Gustafsson F. Comparison of some Kalman filter based methods for manoeuvre tracking and detection // *IEEE Conference on Decision and Control*. — Vol. 2. — 1995. — P. 1525–1531.
- [72] Ivanskiy Y., Sergeenko A. Distributed algorithms for tracking the trajectories of many objects by the set of mobile sensors // *IOP Conference Series: Materials Science and Engineering*. — 2020. — Vol. 984, no. 1. — P. 012003.
- [73] Jadbabaie A., Lin Jie, Morse A.S. Coordination of groups of mobile autonomous agents using nearest neighbor rules // *IEEE Transactions on Automatic Control*. — 2003. — Vol. 48, no. 6. — P. 988–1001.
- [74] Kirkpatrick S., Gelatt Jr. C.D., Vecchi M.P. Optimization by simulated annealing // *Science*. — 1983. — Vol. 220, no. 4598. — P. 671–680.
- [75] Krylatov A. Yu. Network flow assignment as a fixed point problem // *Journal of Applied and Industrial Mathematics*. — 2016. — Vol. 10, no. 2. — P. 243–256.
- [76] Kushner H. J., Yin G. *Stochastic Approximation and Recursive Algorithms and Applications*. — New York, Springer–Verlag, 2003.
- [77] Leonard M. R., Zoubir A. M. Multi-target tracking in distributed sensor networks using particle PHD filters // *Signal Processing*. — 2019. — Vol. 159. — P. 130–146.
- [78] Li X. R., Bar-Shalom Y. Design of an interacting multiple model algorithm for air traffic control tracking // *IEEE Transactions on Control Systems Technology*. — 1993. — Vol. 1, no. 3. — P. 186–194.
- [79] Lynch N. A. *Distributed algorithms*. — Elsevier, 1996.
- [80] Maritime Surveillance Using Multiple High-Frequency Surface-Wave Radars / S. Maresca, P. Braca, J. Horstmann, R. Grasso // *IEEE*

- Transactions on Geoscience and Remote Sensing. — 2014. — Vol. 52, no. 8. — P. 5056–5071.
- [81] Matviychuk O. State estimation for bilinear impulsive control systems under uncertainties // Cybernetics and Physics. — 2018. — Vol. 7, no. 10. — P. 35–40.
- [82] Metropolis N., Ulam S. The Monte Carlo Method // Journal of the American Statistical Association. — 1949. — Vol. 44, no. 247. — P. 335–341.
- [83] Multi-target Tracking in Multiple Non-overlapping Cameras Using Fast-Constrained Dominant Sets / Y.T. Tesfaye, E. Zemene, A. Prati et al. // Int. J. Computer Vision. — 2019. — Vol. 127. — P. 1303–1320.
- [84] Nedic A., Olshevsky A. Distributed Optimization Over Time-Varying Directed Graphs // IEEE Transactions on Automatic Control. — 2015. — Vol. 60, no. 3. — P. 601–615.
- [85] Nedic A., Ozdaglar A. Distributed subgradient methods for multi-agent optimization // IEEE Transactions on Automatic Control. — 2009. — Vol. 54, no. 1. — P. 48.
- [86] Nedic A., Ozdaglar A.E., Parrilo P.A. Constrained Consensus and Optimization in Multi-Agent Networks // IEEE Transactions on Automatic Control. — 2010. — Vol. 55, no. 4. — P. 922–938.
- [87] Network science on belief system dynamics under logic constraints / N.E. Friedkin, A.V. Proskurnikov, R. Tempo, S.E. Parsegov // Science. — 2016. — Vol. 354, no. 6310. — P. 321–326.
- [88] Novel Type of Phase Transition in a System of Self-Driven Particles / T. Vicsek, A. Czirók, E. Ben-Jacob et al. // Phys. Rev. Lett. — 1995. — Aug. — Vol. 75. — P. 1226–1229.
- [89] Olfati-Saber R. Distributed Kalman filter with embedded consensus filters // Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on / IEEE. — 2005. — P. 8179–8184.
- [90] Olfati-Saber R., Murray R. M. Consensus problems in networks of agents with switching topology and time-delays // IEEE Transactions on automatic control. — 2004. — Vol. 49, no. 9. — P. 1520–1533.

- [91] Polyak B. T., Khlebnikov M. V., Shcherbakov P. S. Sparse feedback in linear control systems // Automation and Remote Control. — 2014. — Vol. 75, no. 12. — P. 2099–2111.
- [92] Proskurnikov A., Granichin O. Evolution of clusters in large-scale dynamical networks // Cybernetics and Physics. — 2018. — 11.
- [93] Proskurnikov A.V., Tempo R. A tutorial on modeling and analysis of dynamic social networks. Part I // Annual Reviews in Control. — 2017. — Vol. 43. — P. 65–79.
- [94] Proskurnikov A. V. Average consensus in networks with nonlinearly delayed couplings and switching topology // Automatica. — 2013. — Vol. 49, no. 9. — P. 2928–2932.
- [95] Proskurnikov A. V., Matveev A. S., Cao M. Opinion Dynamics in Social Networks With Hostile Camps: Consensus vs. Polarization // IEEE Transactions on Automatic Control. — 2016. — Vol. 61, no. 6. — P. 1524–1536.
- [96] Rajagopal R., Wainwright M. J. Network-based consensus averaging with general noisy channels // IEEE Transactions on Signal Processing. — 2011. — Vol. 59, no. 1. — P. 373–385.
- [97] Ren W., Beard R. W. Consensus seeking in multiagent systems under dynamically changing interaction topologies // IEEE Transactions on Automatic Control. — 2005. — Vol. 50, no. 5. — P. 655–661.
- [98] Ren W., Beard R. W., Atkins E. M. Information consensus in multivehicle cooperative control // IEEE Control Systems Magazine. — 2007. — Vol. 27, no. 2. — P. 71–82.
- [99] Rutherford S., Bassler B. Bacterial quorum sensing: its role in virulence and possibilities for its control // Cold Spring Harbor Perspectives in Medicine. — 2012. — 11. — Vol. 2.
- [100] Rzevski G., Skobelev P. Emergent Intelligence in Large Scale Multi-Agent Systems // Int. J. Educ. Inf. Technol. — 2007. — Vol. 1. — P. 64–71.
- [101] Rzevski G., Skobelev P. Managing Complexity. — WIT Press, 2014. — P. 217.

- [102] Sensor Selection under Unknown but Bounded Disturbances in Multi-Target Tracking Problem / V. Erofeeva, O. Granichin, O. Granichina et al. // Proceedings of the 2019 27th Mediterranean Conference on Control and Automation (MED). — 2019. — P. 215–220.
- [103] Sergeenko A., Granichin O. Sensor network control based on randomized and multi-agent approaches // Cybernetics and Physics. — 2022. — Vol. 11, no. 2. — P. 94–105.
- [104] Sergeenko A., Granichin O., Proskurnikov A.V. Advanced SPSA-based Algorithm for Multi-Target Tracking in Distributed Sensor Networks // 2020 59th IEEE Conference on Decision and Control (CDC). — 2020. — P. 2424–2429.
- [105] Sergeenko A.N., Granichin O.N., Yakunina M.V. Hamiltonian path problem: the performance comparison deoxyribonucleic acid computing and the branch-and-bound method // Journal of Physics: Conference Series. “International Workshop Navigation and Motion Control”. — Vol. 1536. — 2020. — P. 012003.
- [106] Sergeenko A., Yakunina M., Granichin O. Hamiltonian path problem solution using DNA computing // Cybernetics and Physics. — 2020. — Vol. 9, no. 1. — P. 69–74.
- [107] Shchegryaev A. N., Zakharov V. V. Multi-period cooperative vehicle routing games // Contributions to Game Theory and Management. — 2014. — Vol. 7, no. 1. — P. 349–359.
- [108] Simultaneous Perturbation Stochastic Approximation-Based Consensus for Tracking Under Unknown-But-Bounded Disturbances / O. Granichin, V. Erofeeva, Y. Ivanskiy, Y. Jiang // IEEE Transactions on Automatic Control. — 2021. — Vol. 66, no. 8. — P. 3710–3717.
- [109] Simultaneous perturbation stochastic approximation in decentralized load balancing problem / N. Amelina, V. Erofeeva, O. Granichin, N. Malkovskii // IFAC-PapersOnLine. — 2015. — Vol. 48, no. 11. — P. 936–941.
- [110] Smyth H. D. Atomic Energy for Military Purposes (The Smyth Report). — 1945. — Online; accessed 03 May 2022.
- [111] Spall J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation // IEEE Transactions on Automatic Control. — 1992. — Vol. 37, no. 3. — P. 332–341.

- [112] Spall J. C. Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. — John Wiley & Sons, 2005. — P. 618.
- [113] Spall J. C. Identification for Systems With Binary Subsystems // IEEE Transactions on Automatic Control. — 2014. — Vol. 59, no. 1. — P. 3–17.
- [114] Survey on Blockchain Technology: Evolution, Architecture and Security / M. Nasir, M. Bhutta, A. K. Khwaja, et al. // IEEE Access. — 2021. — Vol. 10. — P. 61048–61073.
- [115] Toward Optimal Distributed Node Scheduling in a Multihop Wireless Network Through Local Voting / D. J. Vergados, N. Amelina, Y. Jiang et al. // IEEE Transactions on Wireless Communications. — 2018. — Vol. 17, no. 1. — P. 400–414.
- [116] Tsitsiklis J. N. Problems in Decentralized Decision Making and Computation : Ph.D. thesis / J. N Tsitsiklis ; PhD. dissertation, MIT. — 1984.
- [117] Tsitsiklis J. N., Bertsekas D., Athans M. Distributed asynchronous deterministic and stochastic gradient optimization algorithms // IEEE Transactions on Automatic Control. — 1986. — Vol. 31, no. 9. — P. 803–812.
- [118] Tsyganov A. V., Tsyganova Y. V., Golubkov A. V. Decentralized algorithm for detecting changes in the motion mode of an object based on multisensor data // 2022 VIII International Conference on Information Technology and Nanotechnology (ITNT) / IEEE. — 2022. — P. 1–5.
- [119] Vicsek T., Zafeiris A. Collective motion // Physics Reports. — 2012. — Vol. 517, no. 3. — P. 71–140.
- [120] Watson J.D., Crick F.H.C. A structure for deoxyribose nucleic acid // Nature. — 1953. — Vol. 171, no. 346. — P. 737–738.
- [121] Weighted SPSA-based Consensus Algorithm for Distributed Cooperative Target Tracking / V. Erofeeva, O. Granichin, A. Proskurnikov, A. Sergeenko // Proc. of the 2021 European Control Conference. — 2021. — P. 1074–1079.
- [122] Zhu H., Spall J. C. Tracking performance of stochastic gradient algorithms with constant step sizes // 2016 IEEE 55th Conference on Decision and Control (CDC) / IEEE. — 2016. — P. 5310–5315.