

Управление движением группы мобильных роботов

Хасан Хафизов

25 июня 2017 г.

Содержание

1	Введение	2
1.1	Спектр задач	3
1.2	Техническое устройство	5
1.3	Алгоритмы управления одиночным роботом	6
1.4	Групповое управление	7
1.4.1	Централизованное управление	8
1.4.2	Децентрализованное управление	12
1.5	Управление строем	15
2	Постановка задачи	16
2.1	Описание постановки	16
2.2	Формальная постановка	18

3	Алгоритм управления	18
3.1	Мастер	19
3.1.1	Оценка движения мастера по траектории	20
3.1.2	Выводы	23
3.2	Миньон	24
3.2.1	Воздействие на агента-лидера функцией Хеви- сайда	25
3.2.2	Выводы	28
4	Строй	28
4.1	Алгоритм поддержания строя	30
4.2	Робастность алгоритма	32
4.2.1	Отказ мастера	33
4.2.2	Критически большое возмущение	37
	Список источников	40

1 Введение

Одной из основных целей мехатроники является создание автоматических устройств, которые могут заменить человека-оператора в опасных для жизни условиях. В связи с этим существенно возрастает роль беспилотных летательных аппаратов (БПЛА). Это связано с успешностью их внедрения для выполнения сложных технологических процессов и операций, для реализации которых необходимо управлять полетом. В настоящее время, в большинстве случаев, управление полетом осуществляется в полуавтоматическом режиме

по командам оператора с использованием навигации по опорным точкам или в дистанционном режиме с помощью пульта управления. Наряду с этим существенно возрастает роль программного управления БПЛА на базе интеллектуальных автопилотов. Это связано с сильным расширением спектра задач, которые сможет выполнить БПЛА при полном отсутствии ручного управления.

1.1 Спектр задач

Существует большое количество БПЛА различных видов, типоразмеров и функционального назначения. Каждый из этих типов обладает своими преимуществами и недостатками, из которых вытекают специфичные для каждого типа БПЛА области задач. В этой работе в качестве объекта управления будут рассмотрены квадрокоптеры — летательные аппараты, относящиеся к вертолётному типу, имеющие шесть степеней свободы, осуществляющие полет путем изменения скорости вращения роторов, работающих по парам. Это позволяет квадрокоптеру двигаться в трехмерном пространстве с помощью четырех режимов: нависание, крен, тангаж и рыскание. Реализация вышеупомянутых режимов осуществляется с помощью микро-ЭВМ, которая управляет механизмом генерирования подъемной силы роторов, регулирует положение квадрокоптера в соответствии с выбранным режимом полета и обеспечивает обмен навигационных данных с разными уровнями управления.

Простота в постройке и наладке, возможность серийного производства недорогих узлов для сборки, а так же появление микрокон-

троллеров, давших возможность производить все необходимые вычисления прямо на летательном аппарате сделали квадрокоптеры одним из приоритетных устройств для решения некоторых задач, ставящихся перед БПЛА. Квадрокоптеры стали использоваться в аэрофотосъёмке, при отладке взаимодействия между роботами, в спасательных операциях, метеорологии. Большим потенциалом обладает так же их использование в военных целях в качестве разведчиков. Квадрокоптер как нельзя лучше подходит для наблюдения и контроля зон, доступ к которым затруднён из-за естественных или техногенных катастроф, или находящихся в условиях непригодных для человека: зон с повышенным уровнем радиации или химически загрязнённых. Современные технологии и достижения в теории управления позволяют квадрокоптерам выполнять задачи во всех перечисленных условиях, практически при любых погодных условиях.

Спектр задач, ставящийся перед квадрокоптерами формируется на основе следующих технических характеристик:

- Простота изготовления, доступность материалов — приводит к возможности массового производства, низким экономическим потерям от выхода из строя даже большого количества устройств.
- Высокая стабильность полёта — позволяет осуществлять предсказуемое управление даже на предельно малых высотах
- Непрехотливость к условиям взлёта и посадки
- Невысокая скорость передвижения и ограниченный запас бата-

рей — требует дислокации близкой к месту выполнения задачи

1.2 Техническое устройство

Квадрокоптер изготавливается в виде креста с равными сторонами, на концах которых установлены пропеллеры с приводами. Все пропеллеры имеют фиксированную ось. Подъемная сила, возникающая в результате вращения пропеллера, перпендикулярна к платформе квадрокоптера. Следовательно, равнодействующая подъемная сила всегда перпендикулярна к платформе робота. Величина подъемной силы зависит от скорости вращения пропеллеров. Вертикальный полет квадрокоптера осуществляется за счет вертикальной составляющей равнодействующей подъемной силы, которая преодолевает силу тяжести. Горизонтальный полет осуществляется за счет горизонтальных составляющих подъемных сил, возникающих при наклоне квадрокоптера. Наклонить квадрокоптер можно с помощью регулирования скоростей вращения пропеллеров, создавая опрокидывающий момент относительно осей x или y (рис. 1). Для этого достаточно установить разную скорость вращения между пропеллерами, находящимися друг против друга по диагонали.

Если одна пара пропеллеров, расположенных друг против друга по диагонали, вращается по часовой стрелке, то другая пара вращается против часовой стрелки. Дело в том, что при вращении винта возникает момент, вращающий квадрокоптер вокруг собственной оси в обратную сторону. Если вращать пропеллеры попарно в разные стороны, то и момент, действующий на тело квадрокоптера и

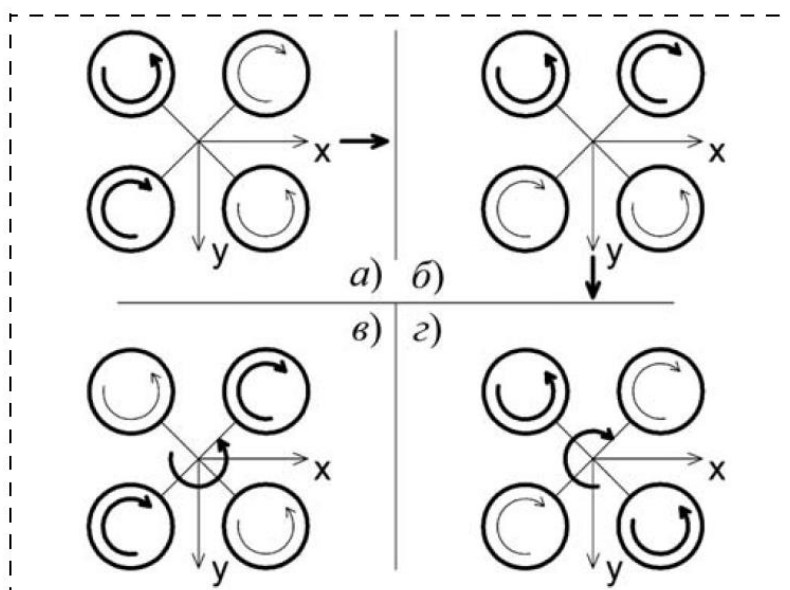


Рис. 1: Принцип полёта квадрокоптера

возникающий в результате вращения пары пропеллеров по часовой стрелке, будет компенсироваться моментом, возникающим в результате вращения пары пропеллеров против часовой стрелки. В вертолетах эту функцию выполняет хвостовой винт, компенсирующий момент, возникающий в результате вращения несущего винта вертолета. Вращение вокруг собственной оси осуществляется за счет разницы скоростей вращения между парами пропеллеров, вращающихся в разные стороны.

1.3 Алгоритмы управления одиночным роботом

Способ управления квадрокоптером чаще всего базируется на ПД или ПИД регуляторах, а также на линейно-квадратичных регуляторах (ЛКР), нечетких регуляторах и нейросетевых регуляторах.

ПИД-регулятор — в настоящее время наиболее часто встречающийся регулятор на технологическом производстве. Он применяется почти во всех задачах стабилизации, в частности его часто используют для стабилизации углов квадрокоптера в воздухе, для полёта по траектории и удержания позиции с помощью GPS, для удержания высоты по барометру, для стабилизации камеры в подвесе. Причиной для широкого использования служит простота построения и промышленного использования, ясность функционирования и пригодность для решения большинства практических задач.

Как было упомянуто выше помимо ПИД-регуляторов существует несколько линейных регуляторов. В их числе: П-регуляторы, ПИ-регуляторы, ПД-регуляторы, И-регуляторы, Д-регулятор.

Одним из минусов ПИД-регулятора является необходимость в настройке коэффициентов, которая требует детального описания моделируемой системы. Однако, достаточно полно смоделировать такую сложную систему как квадрокоптер — не всегда возможно. По этой причине для задачи вычисления коэффициентов ПИД-регулятора используют нейросети. Совокупность этих двух подходов называют нейросетевым регулятором.

1.4 Групповое управление

Группой роботов называется совокупность некоторого числа однотипных или разнотипных роботов, объединенных общей целевой задачей. Для решения задачи, поставленной перед группой роботов или её частью, в недетерминированной среде каждый из роботов вы-

полняет ряд действий, направленных на решение общей задачи. Эти действия должны быть определенным образом скоординированы, согласованы по времени, а часто и в пространстве. Таким образом, возникает задача управления группой роботов, которая заключается либо в реализации заранее найденной последовательности действий роботов группы, либо в оперативном формировании последовательности действий самими и роботами одновременной или рациональной последующей реализации этой последовательности. Имея в виду данную формулировку задачи группового управления, предполагается, что рассматриваемые далее группы состоят из интеллектуальных роботов, то есть снабжены достаточно мощным вычислительным комплексом.

Управление роботами группы осуществляется системой группового управления роботами которая может основываться на одной из следующих стратегий:

- Централизованная стратегия управления
- Децентрализованная стратегия управления
- Гибридная стратегия управления

1.4.1 Централизованное управление

При реализации методов централизованного управления группа роботов имеет центральное управляющее устройство, вычислительный комплекс которого является технической базой системы управления. Управляющее устройство формирует и распределяет действия ро-

ботов группы по информационным каналам. При такой стратегии роботы-исполнители, фактически, решают лишь локальные задачи управления, поэтому могут иметь простые вычислительные комплексы, а значит и небольшие размеры. Однако подход требует наличия надёжного канала связи между управляющим устройством роботами, при потере которого система полностью выходит из строя, равно как и при поломке управляющего устройства.

Централизованные стратегии разделяют на:

- Единоначальное управление
- Иерархическое управление

Под единоначальным управлением подразумевается как расположение центрального устройства управления на пульте управления оператора (рис. 2), так и использование в качестве центрального устройства управления одного из роботов группы (рис. 3). Подход с использованием одного из агентов в качестве управляющего устройства позволяет группе дистанцироваться от исходной точки на большее расстояние, так как связи с "базой" не требуется. Однако, грузоподъёмность агента-лидера ограничивает его максимальную вычислительную мощность, а значит и сложность выполняемой задачи.

Так же недостатком единоначальной стратегии управления можно назвать ограниченную масштабируемость группы: одно вычислительное устройство не может справиться со сколь угодно большим количеством вычислений, необходимых для управления произвольным числом агентов-исполнителей.

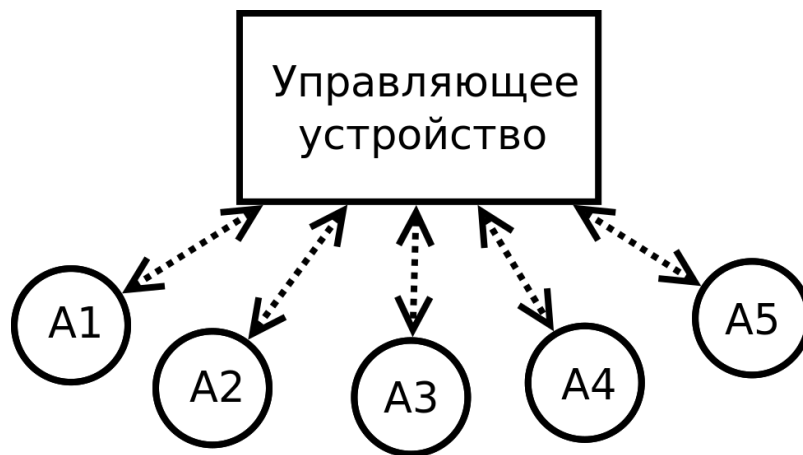


Рис. 2: Схема взаимодействия роботов-исполнителей $A_1 \dots A_5$ и центрального управляющего устройства при единноначальной стратегии управления

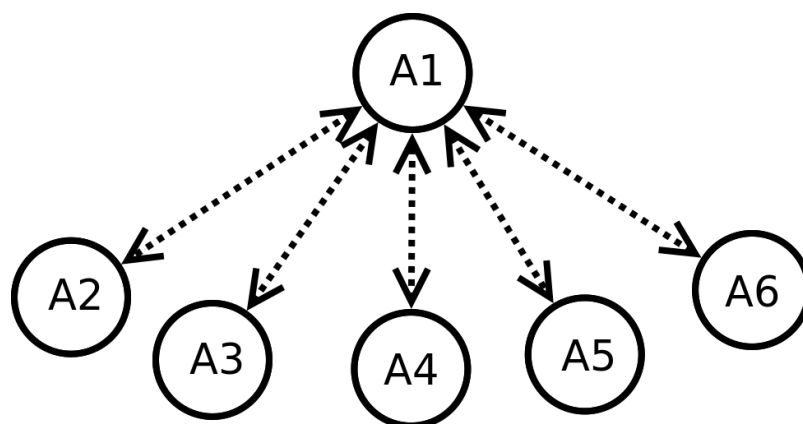


Рис. 3: Схема взаимодействия роботов-исполнителей $A_2 \dots A_6$ и робота-лидера A_1 при единноначальной стратегии управления

Одним из выходов является применение иерархических стратегий управления (рис 4), при которых группа роботов делится на подгруппы, каждая из которых имеет своего лидера, а лидеры, в свою очередь, управляются центральным устройством управления (или роботом с устройством управления на борту).

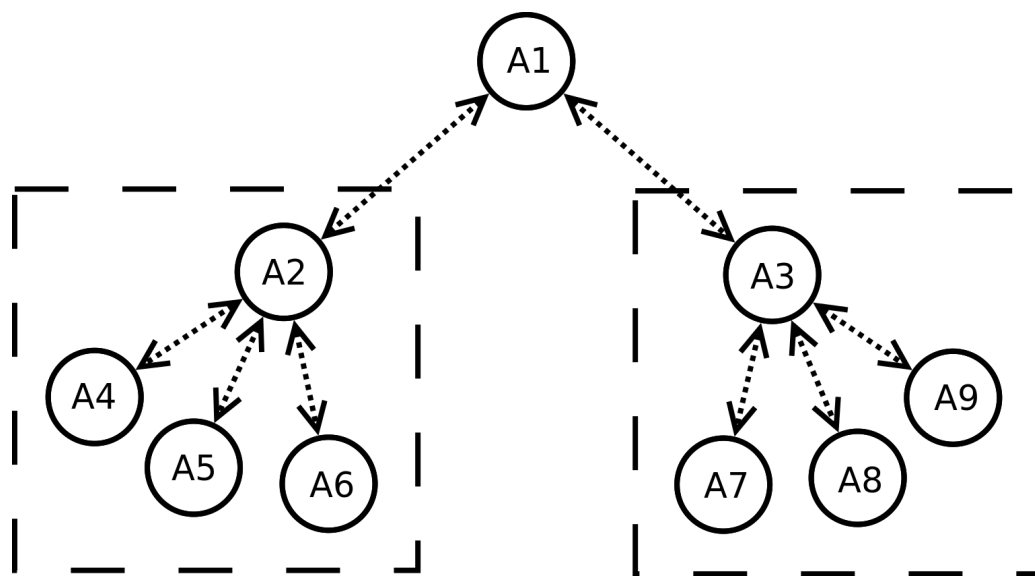


Рис. 4: Схема взаимодействия лидера A_1 , лидеров подгрупп A_2 и A_3 и роботов-исполнителей $A_4 \dots A_9$ при иерархической стратегии управления

Однако, и при таком формировании стратегии управления надёжность по-прежнему низка. Отказ лидера-подгруппы приведёт к отказу всех подконтрольных роботов, а выход из строя лидера — к срыву выполняемой задачи.

1.4.2 Децентрализованное управление

Децентрализованная стратегия –подразумевает отсутствие в системе единого управляющего центра формирования координационных команд, а каждый агент независимо принимает решение о своих действиях, стремясь принести пользу для достижения групповой цели. Такая стратегия управления группой масштабируема и имеет высокую надежность, но сложна в алгоритмизации.

Децентрализованное управление по характеру взаимодействия между агентами обычно делят на следующие типы:

- Коллективное управление
- Стайное управление
- Роевое управление

При коллективной стратегии управления (рис. 5) каждый робот группы передаёт собранные им данные об окружающей среде, о своём состоянии, о его представлении обо всех остальных роботах группы в общий канал связи таким образом, что эта информация доступна всем остальным роботам группы. С помощью информации в канале каждый робот самостоятельно оценивает обстановку и принимает решение о своих дальнейших действиях.

Группа сохраняет работоспособность в случае выхода из строя одного из роботов, но требует, чтобы все агенты находились друг от друга на расстоянии не выше, чем максимальная дальность работы

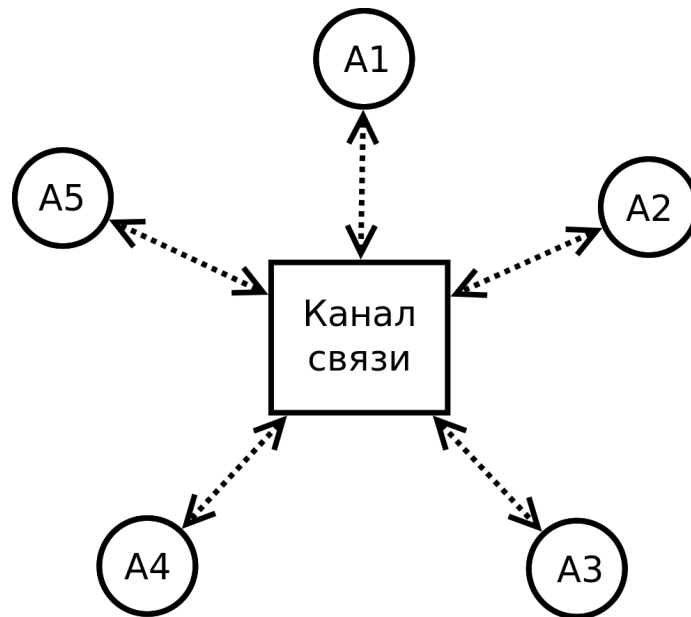


Рис. 5: Схема взаимодействия равнозначных агентов $A_1...A_5$ через общий канал связи

приборов связи, для поддержки канала. Если между двумя агентами будет превышено это расстояние, то они начнут работать в условиях недостаточности информации, и если эту ситуацию не учесть заранее, то выполнение групповой задачи может быть нарушено. Аппаратные требования к агентам достаточно высоки, однако на сегодняшний день вычислительная техника среднего ценового сегмента имеет достаточную мощность, чтобы справиться с предполагаемыми нагрузками.

Нагрузка на канал связи между роботами возрастает прямо пропорционально увеличению численности группы роботов. Также возрастает нагрузка на бортовые вычислительные устройства роботов, так как им необходимо обрабатывать принятую информацию. Хо-

тя верхний предел допустимой численности группы у коллективных методов управления в среднем выше, чем у централизованных, масштабируемость этих методов оставляет желать лучшего.

При стайных стратегиях управления (рис. 6) выделенный канал связи для обмена информацией между роботами отсутствует, каждый робот собирает информацию об окружающей среде и местоположениях остальных роботов самостоятельно и так же самостоятельно принимает решение о своих последующих действиях с тем расчетом, чтобы внести свой вклад в выполнение групповой задачи. Коммуникация между роботами, фактически отсутствует, что ограничивает спектр приложений до задач, которые могут легко распараллеливаться на независимые несвязные подзадачи.

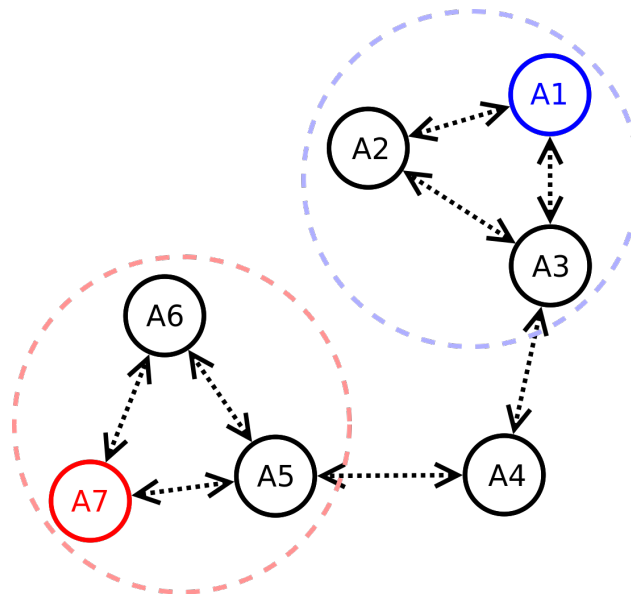


Рис. 6: Группа агентов $A_1...A_7$ с изображенными для агентов A_1 и A_7 зонами радиовидимости

При роевой стратегии управления каждый робот группы взаимодействует лишь с некоторыми соседними роботами, попадающими в зону видимости, ограниченную дальностью действия его телекоммуникационных приборов (либо ограниченную искусственно). Каждый робот самостоятельно принимает решение о дальнейших действиях, опираясь на простые локальные правила (simple rules, local rules). Роботу доступна та информация об окружающей среде, которую он собрал самостоятельно, а также та информация об окружающей среде и о состоянии некоторых роботов группы, которую ему передали соседние роботы. Собранную информацию об окружающей среде, а также о собственном состоянии робот передает в канал связи. Эта информация становится доступна тем роботам, в зону видимости которых попадает этот робот (в случае одинаковых по радиусу зон видимости — это те же соседние роботы). Благодаря такому подходу роботы получают больше информации об окружающей среде, чем при стайных стратегиях управления, при этом доступная им информация касается окружающей их области, т. е. наиболее актуальна. При этом сохраняется масштабируемость — увеличение численности группы не приводит к увеличению нагрузки на бортовые вычислительные устройства.

1.5 Управление строем

Частным случаем группового управления группой является строевая задача. В ней, помимо достижения поставленной цели от группы агентов требуется поддержание упорядоченного рисунка в течение

всего времени выполнения задачи.

Строй БПЛА может применяться для видеомониторинга местности, формирования фазированных антенных решеток и ряда других задач. Так же при задании специфического рисунка строя можно добиться снижения энергозатрат на движение некоторых роботов. Например, можно добиться этого сформировав группу агентов в клин, аналогичный клину, который создаёт косяк перелётных птиц.

2 Постановка задачи

2.1 Описание постановки

В этой работе будет разработан алгоритм роевого управления группой роботов, объединённых в строй с произвольным порядком (рисунком). Каждый робот в строю является интеллектуальным агентом, движущимся в соответствии со своим законом управления, который направлен, во-первых на то, чтобы выполнить общую задачу, а во-вторых чтобы сохранить исходную структуру строя. В качестве задачи, которая будет выполняться строем была выбрана задача движения по произвольной траектории с задаваемой скоростью.

Для взаимозаменяемости требования с аппаратной точки зрения ко всем агентам одинаковые, однако программно в каждый момент времени агент будет принадлежать одному из двух классов:

- Мастер — агент, задача которого отрабатывать движение по траектории и быть главным ориентиром для всех остальных агентов.

- Миньон — агент, задача которого сохранять своё местоположение относительно других участников группы, тем самым сохраняя структуру строя

Моделирование в работе будет проводиться в горизонтальной 2D плоскости. Моделями агентов будут материальные точки с массой.

Ограничения:

- У каждого агента есть ограничение видимости. Если два участника строя в данный момент времени находятся вне зоны видимости друг-друга, то они не могут напрямую обмениваться информацией
- Максимальное по модулю управляющее воздействие ограничено
- Агенты передают друг-другу информацию только о своих координатах

Основные требования к решению:

- Без возмущений строй должен отрабатывать заданную траекторию с малыми погрешностями
- При малых возмущениях каждый отдельно взятый агент не должен существенно отклоняться от своей позиции в строю. Строй в целом должен по прежнему выполнять задачу, с погрешностями соразмерными возмущениям

- При краткосрочных больших возмущениях, даже если строй потерял целостность, он должен самовосстанавливаться и продолжать выполнение задачи
- При выходе из строя одного или нескольких агентов строй должен переформироваться так, чтобы сохранить исходный рисунок. Строй должен быть устойчив как к отказу мастера, так и к отказу любого из миньонов

2.2 Формальная постановка

Вектор состояния каждого агента в горизонтальной плоскости xu :

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix},$$

где $x_1 = x$, $x_2 = \dot{x}$, $x_3 = y$, $x_4 = \dot{y}$. Тогда закон движения:

$$\dot{X} = AX + BU$$

3 Алгоритм управления

В предлагаемом мной алгоритме управления агентов можно разделить на два класса:

- мастер — агент, который отрабатывает движение по траектории

- миньон — поддерживает своё местоположение в строю

Закон управления для этих двух типов агентов задаётся по-разному.

3.1 Мастер

Мастером является агент, для которого желаемый закон движения S_d задаётся оператором извне: это может быть записанная в память агента траектория, целевая позиция или скорость.

Фактически, этот агент ничего не знает о существовании других агентов в строю (миньонов). Его задача — выполнение поставленного закона движения, поэтому закон управления зависит только от закона движения:

$$\vec{u} = \vec{u}(S_d)$$

Рассмотрим конкретный закон управления \vec{u}_{tr} : движение по некоторой траектории $\vec{tr}(t)$:

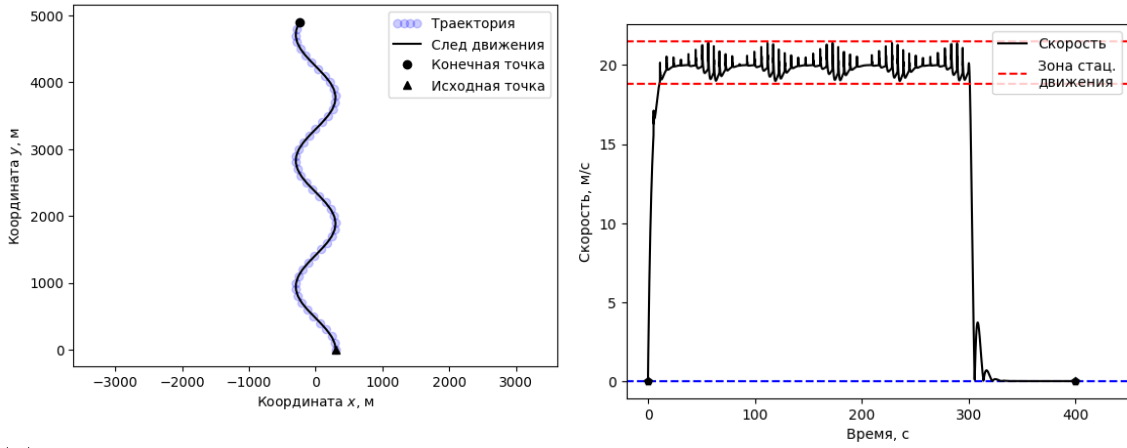
$$\vec{u}_{tr} = \vec{u}_{along} + \vec{u}_{across}$$

Закон управления состоит из двух частей. Первая \vec{u}_{along} отвечает за усилие вдоль траектории, вторая \vec{u}_{across} — поперёк. Направлением для \vec{u}_{along} служит направление вектора между текущим положением агента и следующей точкой траектории.

(Тут будет более подробно о том, как вычисляется следующая точка траектории, о алгоритмах управления на PD регуляторах, которые используются как в \vec{u}_{along} , так и в \vec{u}_{across})

Пример движения мастера по траектории, задаваемой параметрическим уравнением, где s — параметр: $s \in [0, 5000]$.

$$x(s) = 300 \cdot \cos\left(\frac{s}{300}\right); \quad y(s) = s$$



(a) Исходная траектория и след от движения

(b) Скорость движения мастера

Рис. 7: Движение мастера по заданной траектории с заданной скоростью $v_{desired} = 20 \frac{\text{м}}{\text{с}}$. Расстояния на рисунках задаются в метрах, время в секундах, скорость в $\frac{\text{м}}{\text{с}}$.

3.1.1 Оценка движения мастера по траектории

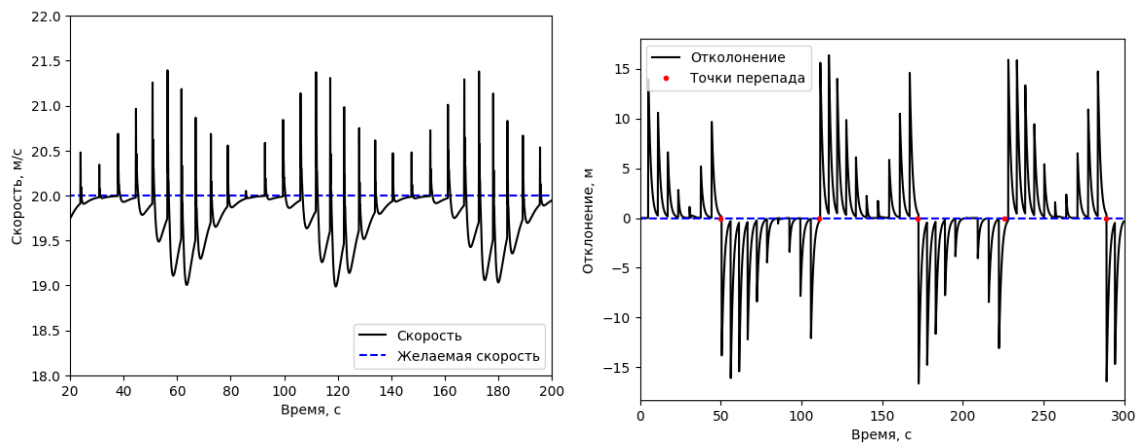
Агент преодолел заданную траекторию за $t = 331\text{с}$.

Проеденное расстояние:

$$S = \int_0^{5000} \sqrt{x'^2(s) + y'^2(s)} \, ds \approx 6051\text{м}$$

Средняя скорость $v_{av} = 18.3 \frac{\text{м}}{\text{с}}$.

Стационарным режимом движения можно назвать режим, при котором скорость агента колеблется в пределах между $18.8 \frac{м}{с}$ и $21.5 \frac{м}{с}$ (рис. 7b). Более подробно скорость мастера в стационарном режиме можно увидеть на рис. 8a



(а) Скорость мастера в стационарном режиме (б) Отклонения мастера от траектории

Рис. 8: Иллюстрация отклонений от желаемого закона движения

Отклонения агента от заданной траектории представлены на рис. 8b. Выше нуля — отклонения от траектории вправо, ниже нуля — влево. Максимальное отклонение составляет примерно 15 м.

Резкие перепады на графике, обозначенные как точки перепада объясняются тем, что в этих точках у траектории изменяется знак первой производной, а агент, всегда остающийся на внутренней части траектории резко оказывается на внешней. Проверим это утверждение. Заданное выше параметрическое уравнение фактически явля-

ется уравнением

$$x = 300 \cdot \cos\left(\frac{y}{300}\right)$$

Равенство нулю первой производной:

$$\sin\left(\frac{y}{300}\right) = 0 \Rightarrow y = 300\pi n$$

Найдём первые 5 точек траектории, в которой происходит смена знака второй производной:

$$M = \left\{(-300; 942), (300; 1885), (-300; 2827), (300; 3770), (-300; 4712)\right\}$$

Найдём моменты времени, в которые эти точки будут достигнуты агентом:

$$\hat{L}_t = \{61c, 118c, 177c, 234c, 292c\}$$

Реальные же моменты времени, в которые происходит резкое изменение величины отклонения от траектории:

$$L_t = \{50c, 110c, 172c, 226c, 289c\}$$

То есть, изменение стороны относительно линии траектории по которой движется агент изменяется незадолго до того, как будет изменён знак первой производной траектории. Эта закономерность также наблюдается на рис. 9.

(Тут будет исследование устойчивости алгоритма к случайным возмущениям)

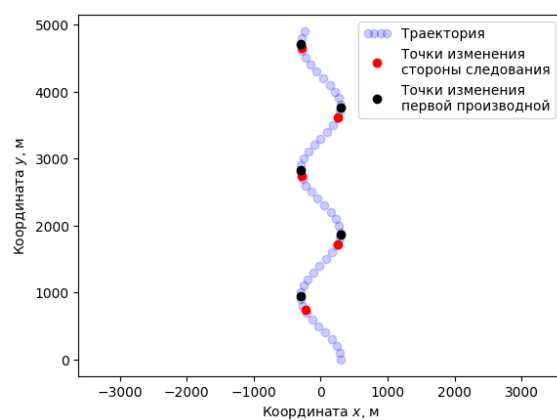


Рис. 9: Траектория с наложенными на неё точками изменения стороны следования и точками изменения знака первой производной.

3.1.2 Выводы

Движение мастера по траектории является точным и предсказуемым. Данный алгоритм управления по траектории с задаваемой скоростью является пригодным для применения.

3.2 Миньон

Миньон является ведомым агентом, во время движения он никак не использует информацию о траектории движения. Всё, что использует миньон i — это положения доступных для него агентов лидеров L_k . Относительно них миньон выстраивает свой закон управления \vec{u}_m :

$$\vec{u}_m = \vec{u}_m(L_k)$$

Более подробно об агентах-лидерах будет сказано в гл. 4, а пока о них можно думать просто как о постоянно обновляемой точке, относительно которой агент строит некую другую точку, в которую стремится попасть. Эта точка будет называться виртуальным лидером L^v . Точка виртуального лидера своя у каждого миньона и зависит только от реального положения агентов-лидеров L .

Пусть, есть миньон с координатами \vec{A}_1 и виртуальным лидером в точке \vec{L}_1^v . Первый закон управления, который приходит на ум может выглядеть следующим образом:

$$\vec{u}_m(t) = S \cdot P, \text{ где } S = (\vec{L}_1^v - \vec{A}_1) \quad (1)$$

Суть такого закона управления в том, что воздействие осуществляется в направлении виртуального лидера с усилием пропорциональным расстоянию между текущей координатой и координатой виртуального лидера. Коэффициент P — пропорциональная составляющая PID регулятора.

Однако, такой закон управления не выдерживает некоторых испытаний.

3.2.1 Воздействие на агента-лидера функцией Хевисайда

Этот численный эксперимент заключается в следующем: есть пара агентов — миньон и его агент-лидер. На агента-лидера воздействует сила по закону Хевисайда, то есть, фактически, после некоторого момента времени \hat{t} появляется постоянное воздействие на агента-лидера.

Для облегчения понимания ситуации рассмотрим воздействие на примере.

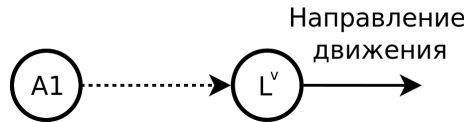


Рис. 10

На рис. 4 изображён миньон A_1 и его виртуальный лидер L^v . На L^v действует сила (распределённая как функция Хевисайда):

$$F(t) = \begin{cases} 1, & \text{если } t \geq 5\text{с} \\ 0, & \text{в остальных случаях} \end{cases}$$

Если использовать закон управления, записанный уравнением (1), то возникнет ситуация, при которой расстояние между миньоном и его виртуальным агентом будет постоянно расти. Ускорение лидера будет расти быстрее, чем ускорение миньона из-за того, что агент строит свой закон управления только на основании положения лидера, ничего не зная о его ускорении и скорости. Однако передавать эти данные от лидера к миньону может оказаться слишком накладно, т.к. придётся поддерживать более широкий канал связи.

В качестве альтернативы миньон может использовать уже имеющиеся данные о положениях виртуального лидера и увеличивать управляющее воздействие, если расстояние между ним и лидером растёт \dot{S} . Так же для устранения статических ошибок добавляется I составляющая PID регулятора.

$$\vec{u}_m(t) = P \cdot S + I \cdot \int_0^t S dt + D \cdot \dot{S} \quad (2)$$

Результаты численного моделирования с данным законом управления представлены на рисунках 11, 12, 13, 14.

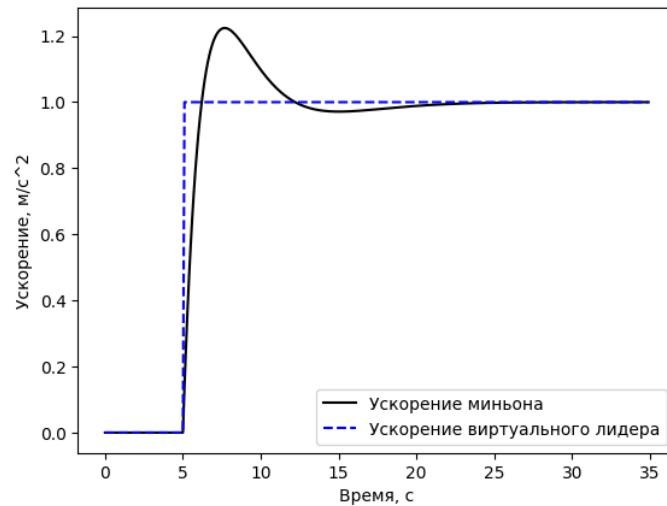


Рис. 11: Ускорение миньона и его виртуального лидера. Виртуальный лидер ускоряется по закону Хевисайда. Ускорение выходит на стационарный режим на 25с: через 20с. после воздействия.

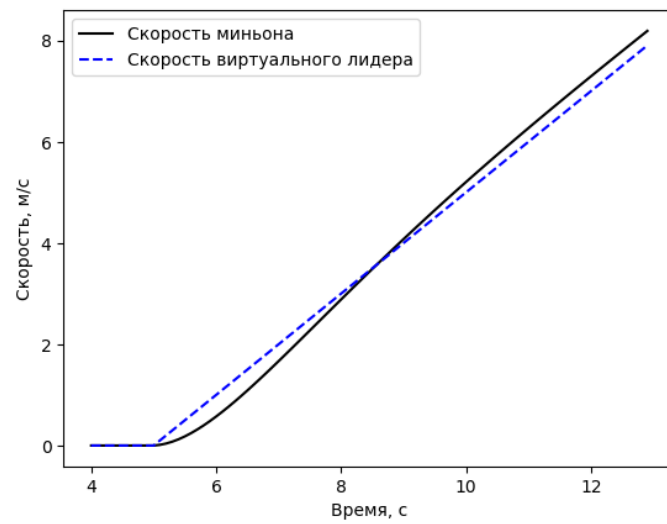


Рис. 12: Скорость миньона и виртуального лидера

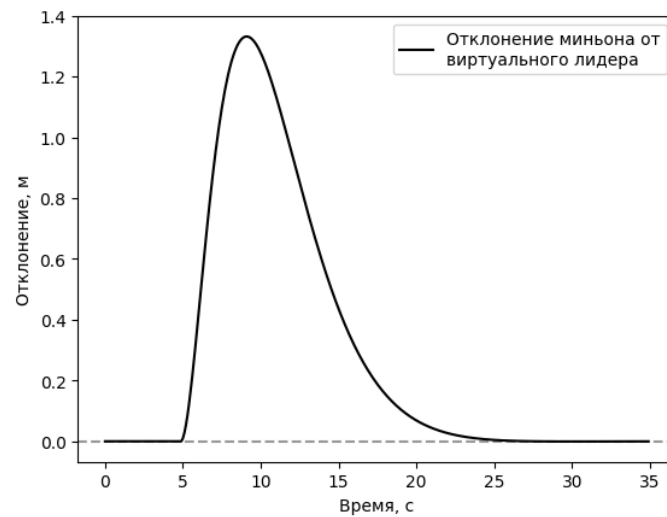


Рис. 13: Отклонение положения миньона от его виртуального агента

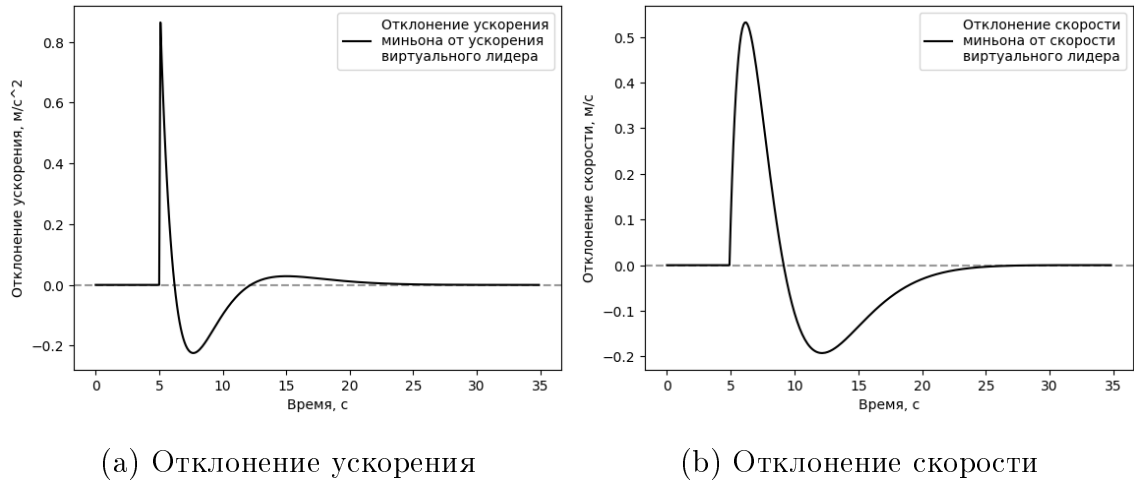


Рис. 14: Иллюстрация отклонений от закона движения виртуального лидера. Выше нуля — значение меньше, чем у лидера, ниже нуля — выше, чем у лидера.

3.2.2 Выводы

Описанный закон управления справляется с задачей приведения миньона в точку виртуального лидера и будет использован в дальнейшем для управления строем.

4 Строй

Строй представляет из себя множество агентов соединённых связями, подобно графу. Для каждого агента, не являющегося мастером (миньона) определён набор связей: каждый агент с порядковым номером i связан со всеми агентами с порядковыми номерами $j : i > j$, называемыми лидерами данного миньона. Каждый агент — вершина графа, каждая связь миньон-лидер — ребро графа. Пример строя из

5-ти агентов изображён на рис.15.

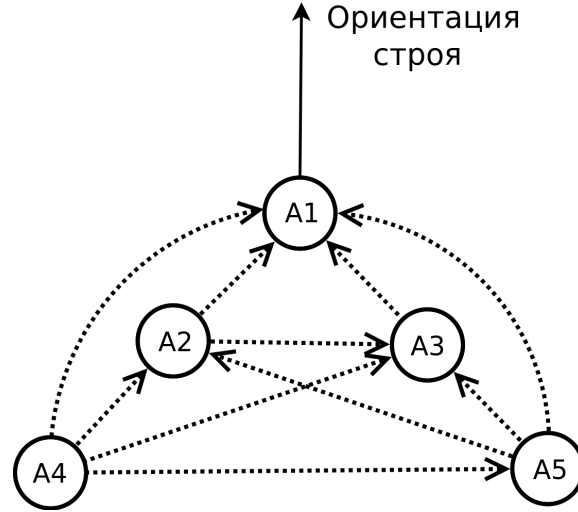


Рис. 15: Пример клиновидного строя из 5-ти агентов

Каждая связь J_{ij} — это вектор в полярных координатах, у которого радиус — это расстояние от миньона i до миньона лидера j , а угол — это угол между изначальной ориентацией строя и вектором, соединяющим миньона с его лидером. К примеру, если расстояния между каждой парой агентов равно 1м, то связи:

- J_{1j} — не определены ни для каких j : A_1 является мастером
- $J_{21} = (r = 1\text{м}; \varphi = 225^\circ)$ — для миньона A_2 агентом лидером является мастер A_1 . Агенты находятся на расстоянии 1м, угол между соединяющим их вектором и вектором изначальной ориентации составляет 225°
- J_{31}, J_{42}, J_{53} и т.д. — определяются аналогично

4.1 Алгоритм поддержания строя

В каждый момент времени положение миньона определяется положением его лидеров. Точка, в которую стремится попасть миньон называется точкой виртуального лидера. Виртуальный лидер — точка пространства, в которую стремится попасть миньон для того, чтобы поддерживать фигуру строя. Для того, чтобы рассчитать эту точку необходимо отложить вектор связи от каждого лидера этого миньона и всять центр масс от полученных точек.

При повороте строя местоположение каждого миньона должно измениться в соответствии с повёрнутой структурой. При повороте строя поворачивается и вектор ориентации строя, за счёт чего далее пересчитывается позиция, в которой должен находиться агент после поворота.

Например, на рис. 16 изображён поворот строя на угол b . Для того, чтобы пересчитать позицию, в которой должен находиться миньон после поворота (виртуальный лидер) необходимо для каждого доступного лидера отложить повёрнутый вектор изначальной ориентации O_0 на угол $b + \varphi$, где φ — угол из вектора связи J_{ij} миньона i и выбранного лидера. Длина откладываемого вектора равна расстоянию r из вектора связи.

Рассмотрим пример того, как будет рассчитан виртуальный лидер A'_5 для агента A_5 (см. рис. 16), для которого единственным агентом-лидером в зоне видимости является A_3 . Вектор связи между ними $J_{53} = (r_{53}; \varphi_{53})$. Для начала повернём вектор изначальной ориентации O_0 :

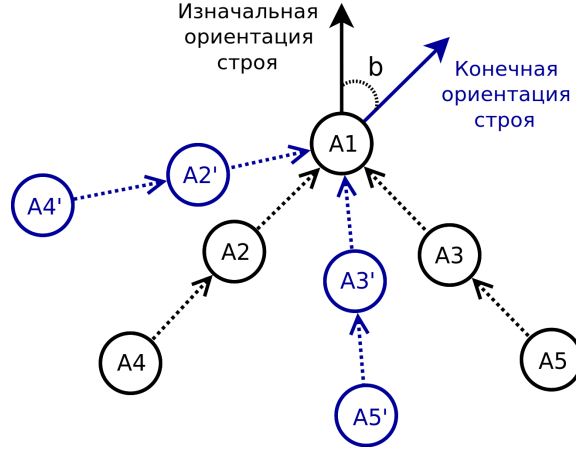


Рис. 16: Поворот строя на угол b

$$\vec{O}_1 = \begin{pmatrix} \cos(b + \varphi_{53}) & \sin(b + \varphi_{53}) \\ \sin(b + \varphi_{53}) & \cos(b + \varphi_{53}) \end{pmatrix} \vec{O}_0$$

Теперь вдоль полученного вектора \vec{O}_1 необходимо отложить от позиции агента A_3 расстояние равное r_{53} :

$$\vec{A}'_5 = \vec{A}_3 + \frac{\vec{O}_1}{|\vec{O}_1|} r_{53}$$

Если для миньона доступно больше агентов-лидеров, то виртуальный лидер будет, как уже говорилось выше, центром масс получаемых точек. Аналогичным образом каждый миньон рассчитывает своего виртуального лидера и за счёт стремления в эти точки и сохраняется структура строя.

Резюмируя, алгоритм строевого движения заключается в том, что агенты связаны между собой. Мастер, двигаясь по траектории заставляет двигаться и всех миньонов, для которых он является лидером, а перемещение этих миньонов, приводит в движение следую-

щий слой миньонов, для которых они являются лидерами.

Результат симуляции движения строя по траектории без внешних возмущений изображен на рис. 17.

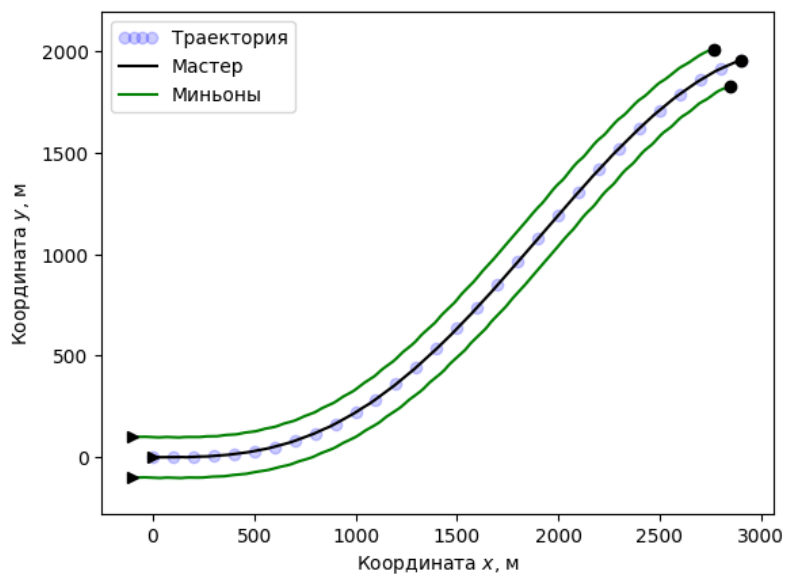


Рис. 17: Движение строем мастера и двух миньонов

Однако, более интересным является тестирование алгоритма на критические воздействия.

4.2 Робастность алгоритма

В этой главе будут рассмотрены некоторые сценарии воздействия на строй, дополненные описанием используемых возможностей алгоритма, не упомянутых ранее.

4.2.1 Отказ мастера

При отказе мастера строй не должен потерять работоспособность. Для этого в алгоритме предусмотрена возможность смены мастера на любого из агентов в строю. Чтобы объяснить механизм обнаружения выхода из строя мастера необходимо описать некоторые, до этого момента не освещённые функциональные возможности алгоритма.

Перед вылетом строя каждому агенту присваивается порядковый номер, который будет определять то, на каком месте в строю будет двигаться каждый из агентов. Позднее этот параметр будет определять то, на какую позицию перейдёт каждый агент, если какой-то другой агент вышел из строя. Для того, чтобы отслеживать полностью строй, между агентами создаётся общий канал связи, в который каждый из них отправляет сообщение, чтобы оповестить остальных о своей работоспособности. Если какой-то агент долгое время не отправляет данных о себе в этот канал, то остальные считают его оказавшим и строй переформируется так, чтобы заполнить вышедший элемент.

Из-за того, что не каждая пара агентов может общаться напрямую (из-за ограниченной зоны радиосвязи) канал связи формируется следующим образом: каждый агент отправляет в широкополосный канал информацию не только о себе самом, но и собственную версию жизнеспособности всех агентов. Таким образом удаётся достичь того, что если между парой агентов есть хотя бы косвенная связь, то они будут знать о существовании друг друга.

На рис. 18 изображен сценарий поочерёдного выхода из строя

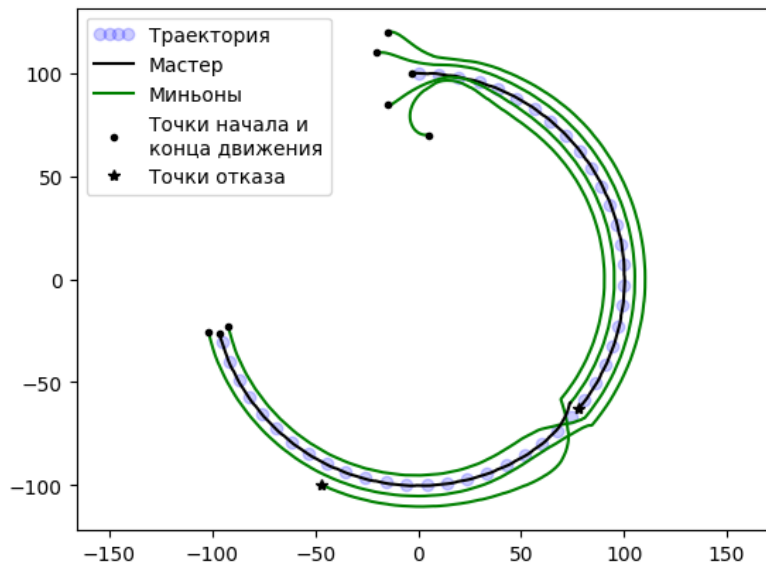


Рис. 18: Движение агентов строем с отказом двух агентов. Чёрной линией изображён мастер. После отказа мастера происходит его смена и чёрной линией изображается сменивший отказавшего мастер.

двух агентов. Сначала мастера, затем одного из миньонов. Так же на рисунке присутствует отклонение по начальным условиям: агенты формируют строй изначально находясь в произвольных точках. Подробнее важные области изображены на рисунках 19, 20, 21.

Вывод: выход мастера и одного из миньонов не оказал критического влияния на строй в целом.

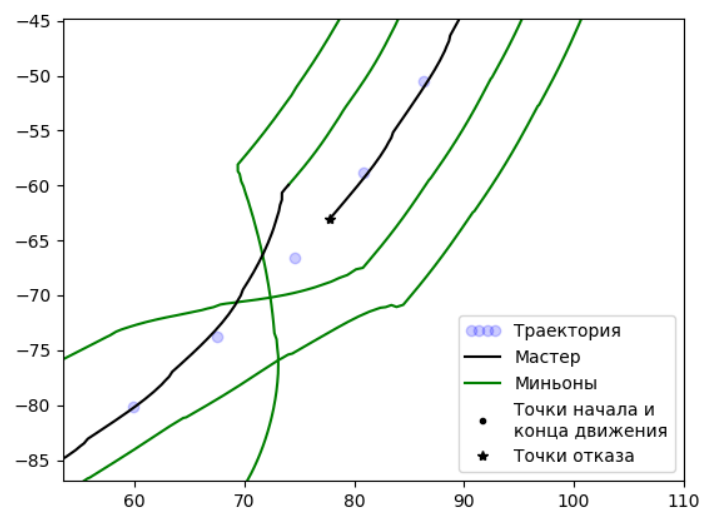


Рис. 19: Область отказа мастера. На позицию мастера встаёт другой агент.

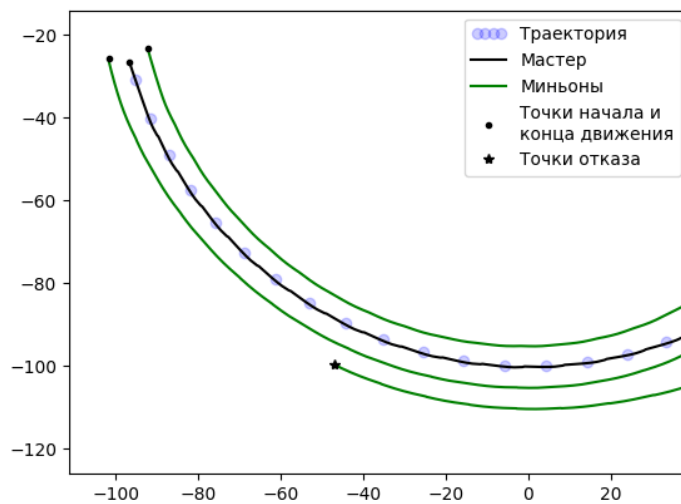


Рис. 20: Область отказа миньона. На его позицию не встаёт никакой агент, т.к. эта позиция в строю имеет приоритет ниже, чем остальные.

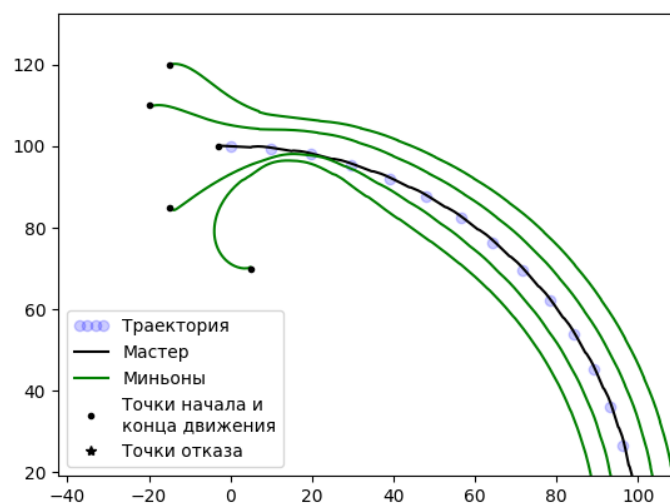


Рис. 21: Область изначального формирования строя агентами.

4.2.2 Критически большое возмущение

Под критически большими возмущениями имеются в виду возмущения, способные разбить строй на части. Фактически, разбить строй на части означает развести агентов в пространстве так, что одна часть не будет радиодоступна другой даже косвенно, то есть между любой парой агентов из разных частей расстояние больше чем радиус радиодоступности R_a .

Симуляция такого сценария преведена на рис 22. Описывается происходящее следующим образом: на двух агентов из строя в некоторый момент времени действует сильное кратковременное возмущение (рис. 23), выводящее этих агентов из зоны видимости остального строя. Далее эти два отделившихся агента формируют собственный строй (рис. 25) и двигаются по заданной траектории так, будто главного строя не существует. Когда отделившийся строй попадает в зону видимости другого они сливаются в один рис. (24).

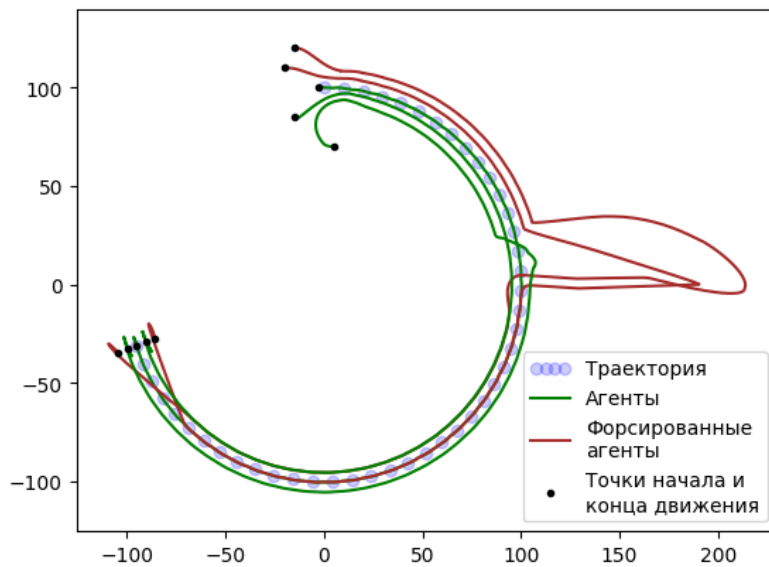


Рис. 22: На двух агентов из строя кратковременно действует такая сила, что они отделяются от основного строя

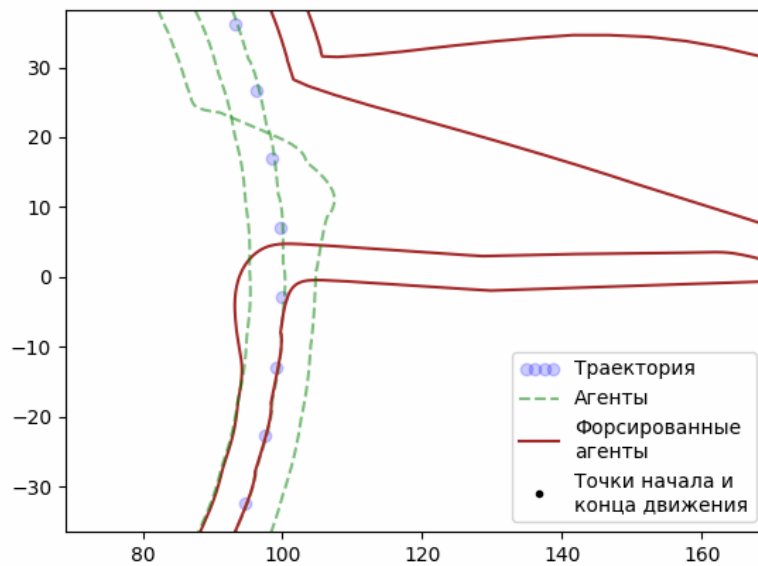


Рис. 23: Отделившиеся агенты формируют собственный строй из мастера и одного миньона и двигаются по траектории независимо от другой части изначального строя

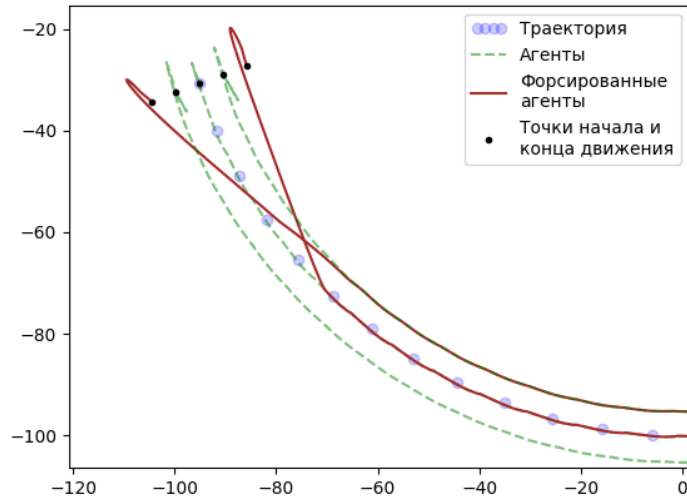


Рис. 24: При достижении отделившимся строем границы зоны видимости первого строя они сливаются в единый строй

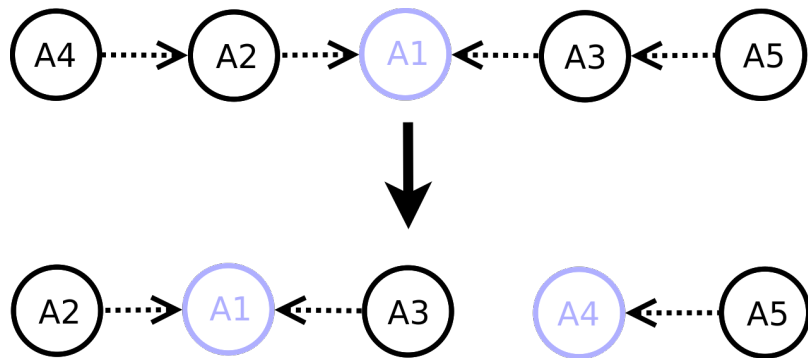


Рис. 25: Строй разбивается на две группы. В одной 3 агента, в другой 2 агента. Голубым цветом помечены мастера

Список литературы

1. Пшихопова В.Х. Групповое управление подвижными объектами в неопределённых средах — 2015
2. Иванов Д.Я. Методы построения пространственных формаций в группах БПЛА типа квадрокоптеров
3. Петров В.Ф., Барунин А.А., Терентьев А.И. Модель системы автоматического управления БПЛА
4. Иванов Д.Я. Решение строевой задачи в группе беспилотных квадрокоптеров
5. Иванов Д.Я. Построение формаций в группах квадрокоптеров с использованием виртуального строя
6. Каляев И.А. Самоорганизация в мультиагентных системах
7. Галустян Н.К. Децентрализованное управление группой квадрокоптеров
8. Амелин К.С., Антал Е.И., Васильев В.И. Адаптивное управление автономной группой летательных аппаратов
9. Иванов Д.Я. Методы роевого интеллекта для управления группами малоразмерных беспилотных летательных аппаратов
10. Иванов Д.Я. Использование принципов роевого интеллекта для управления целенаправленным поведением массово применяемых микророботов в экстремальных условиях

11. Бесекерский В.А., Попов Е.П. Теория систем автоматического управления — 2007
12. Морозова Н.С. Управление движением строя в мультиагентных системах — 2015
13. Гурьянов А.Е. Моделирование управления квадрокоптером — 2014