Децентрализованное управление строем

Хасан Хафизов

13 июня 2017 г.

1 Постановка задачи

2 Механическая модель агента

Моделью агента является материальная точка с массой m. Закон движения:

$$\begin{cases} m\ddot{x} = F_x \\ m\ddot{y} = F_y \end{cases}$$

$\vec{F}$ — сила, действующая на агента, может включать в себя:

$$\vec{F} = \vec{u} + \vec{W} + \vec{F_{\text{тр}}}$$

Где $\vec{u}$ — управляющее воздействие, $\vec{W}$ — случайные помехи, $\vec{F_{\text{тр}}}$ — сила трения.

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

- интеллектуальный (мастер)
- управляемый (миньон)

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

2.1 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 $\vec{tr}(t)$:

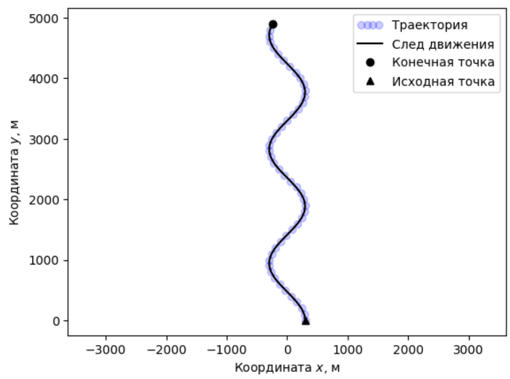
$$\vec{u}_{tr} = \vec{u}_{along} + \vec{u}_{across}$$

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.

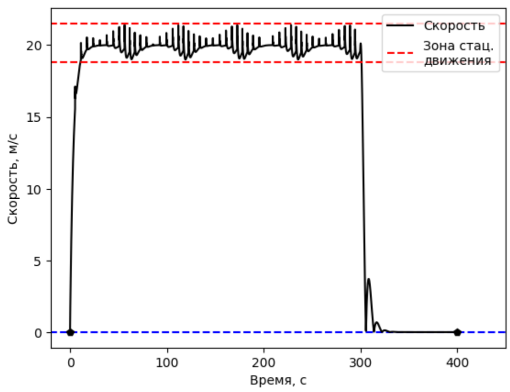
(Тут будет более подробно о том, как вычисляется следующая точка траектории, о алгоритмах управления на PD регуляторах, которые используются как в $\vec{u}_{along}$, так и в $\vec{u}_{across}$)

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

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



(a) Исходная траектория и след от движения



(b) Скорость движения мастера

Рис. 1: 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 $\frac{\text{м}}{\text{с}}$

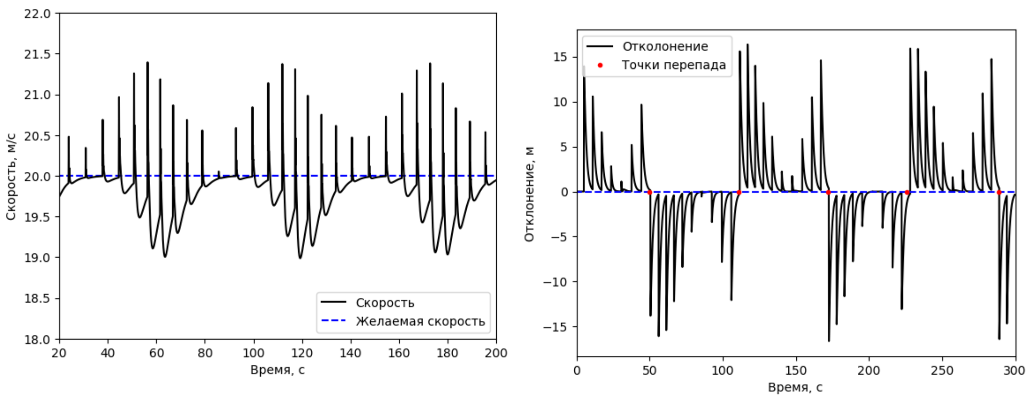
.

2.1.1 Оценка движения мастера по траектории

Агент преодолел заданную траекторию за $t = 331$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 $18.8\frac{\text{м}}{\text{с}}$ и $21.5\frac{\text{м}}{\text{с}}$. Более подробно скорость мастера в стационарном режиме можно увидеть на рис. 2a



(a) Скорость мастера в стационарном режиме

(b) Отклонения мастера от траектории

Рис. 2: Иллюстрация отклонений от желаемого закона движения

.

Отклонения агента от заданной траектории представлены на рис. 2b. 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 является уравнением

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

Равенство нулю первой производной:

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

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

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

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

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

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

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

То есть, изменение стороны относительно линии траектории по которой движется агент изменяется незадолго до того, как будет изменён знак первой производной траектории. Эта закономерность так же наблюдается на рис. 3.

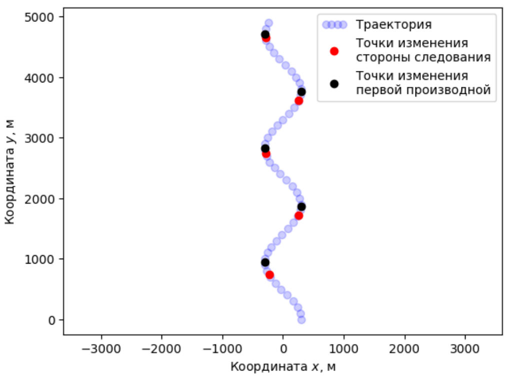


Рис. 3: Траектория с наложенными на неё точками изменения стороны следования и точками изменения знака первой производной.

(Тут будет исседование устойчивости алгоритма к случайным возмущениям)

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

2.2 Миньон

Миньон является ведомым агентом, он не имеет информации о траектории движения.

Расчёт местоположения виртуального лидера просходит следующим образом:

1. Получение текущих координат от агента, являющегося

3 Строй

Строй представляет из себя множество агентов соединённых связями, подобно графу. Для каждого агента, не являющегося мастером должно быть определено не менее одного агента-лидера. Лидером для агента может быть как мастер, так и любой миньон. Каждый агент — вершина графа, каждая связь миньон-лидер — ребро графа. Пример строя из 5-ти агентов изображён на рис.4.

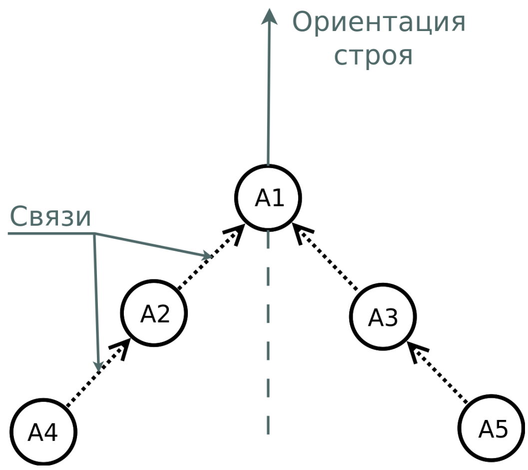


Рис. 4: Пример клиновидного строя из 5-ти агентов

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
- $J_{21} = (r = 1\text{м}; \varphi = 225°)$ — 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} — определяются аналогично

При повороте строя поворачивается и вектор ориентации строя, за чёт чего далее пересчитывается позиция, в которой должен находиться агент после поворота.

Например, на рис. 5 изображён поворот строя на угол b. Для того, чтобы пересчитать позицию, в которой должен находиться миньон после поворота (далее виртуальный лидер) необходимо повернуть вектор изначальной ориентации O_0 на угол $b + \varphi$, где φ — угол из любого вектора связи J_{ij} миньона i, далее от положения выбранного миньона-лидера необходимо отложить расстояние равное r из вектора связи в направлении повёрнутого вектора изначальной ориентации O_0.

Рассмотрим пример того, как будет пересчитан

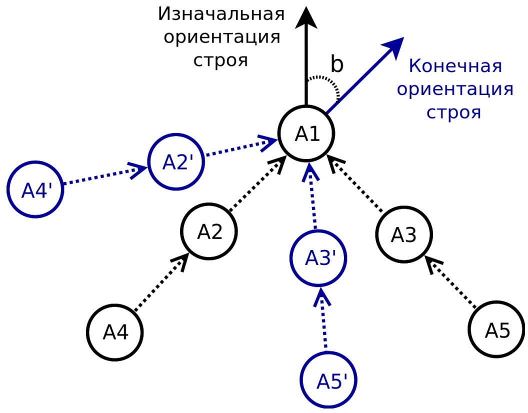


Рис. 5: Поворот строя на угол b

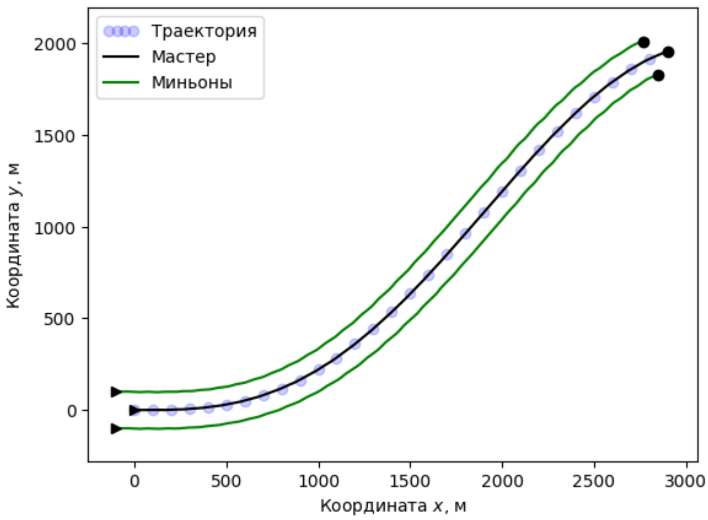


Рис. 6: Движение строем мастера и двух миньонов