Децентрализованное управление строем

Хасан Хафизов

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

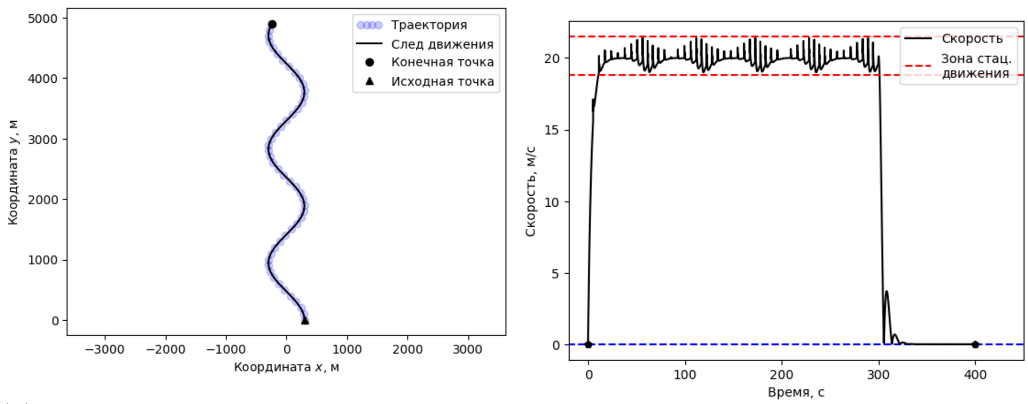
$$\vec{u}_{tr} = \vec{u}_{along} + \vec{u}_{across}$$

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 $\frac{\text{м}}{\text{с}}$

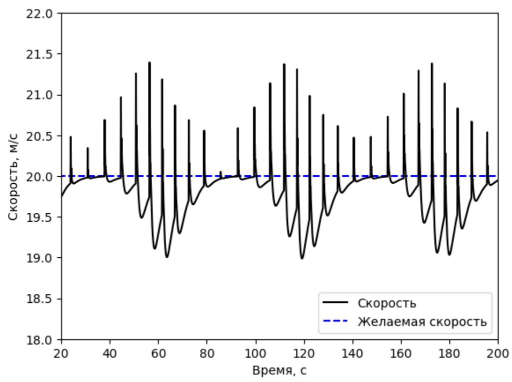
.

2.1.1 Оценка движения мастера по траектории

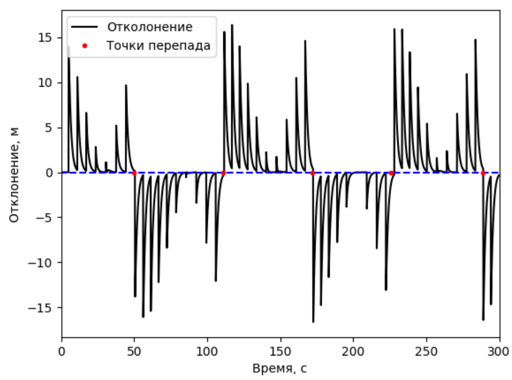
Агент преодолел заданную траекторию за $t = 331$с.
Проеденное расстояние:

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

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.

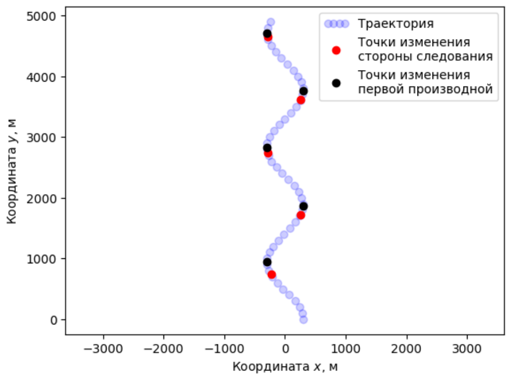


Рис. 3: Траектория с наложенными на неё точками изменения стороны следования и точками изменения знака первой производной.

(Тут будет исседование устойчивости алгоритма к случайным возмущениям)

2.1.2 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, во время движения он никак не использует информацию о траектории движения. Всё, что знает миньон i — это положение своего агента-лидера L. Относительно агента-лидера миньон выстраивает свой закон управления $\vec{u_m}$:

$$\vec{u_m} = \vec{u_m}(L)$$

Более подробно об агентах-лидерах будет сказано в гл. 3, 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 агента-лидера 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 обаразом:

$$\vec{u_m}(t) = S \cdot P, \text{ где } S = \left(\vec{L_1^v} - \vec{A_1}\right) \tag{1}$$

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.

2.2.1 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.

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

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

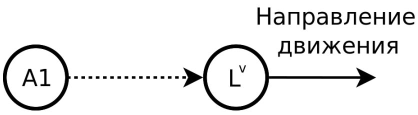


Рис. 4

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.

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

Результаты численного моделирования с данным законом управления:

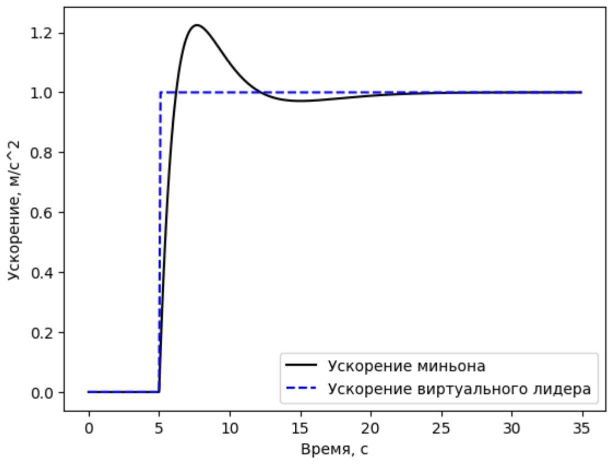


Рис. 5: Ускорение миньона и его виртуального лидера. Вирутальный лидер ускоряется по закону Хевисайда. Ускорение выходит на стационарный режим на 25с: через 20с. после воздействия.

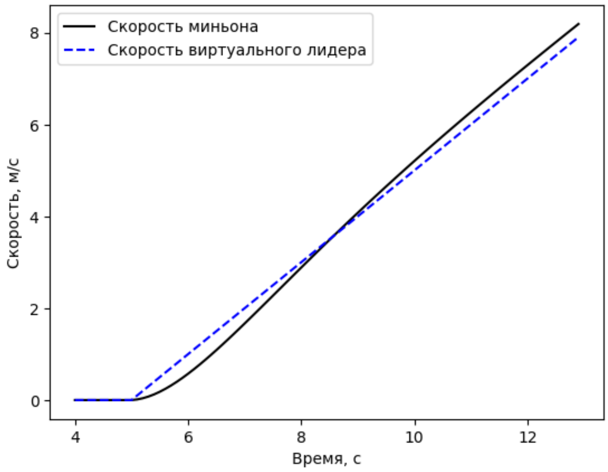


Рис. 6: Скорость миньона и виртуального лидера

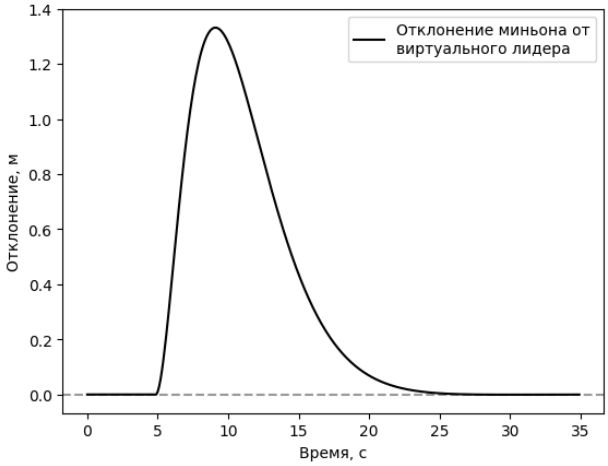
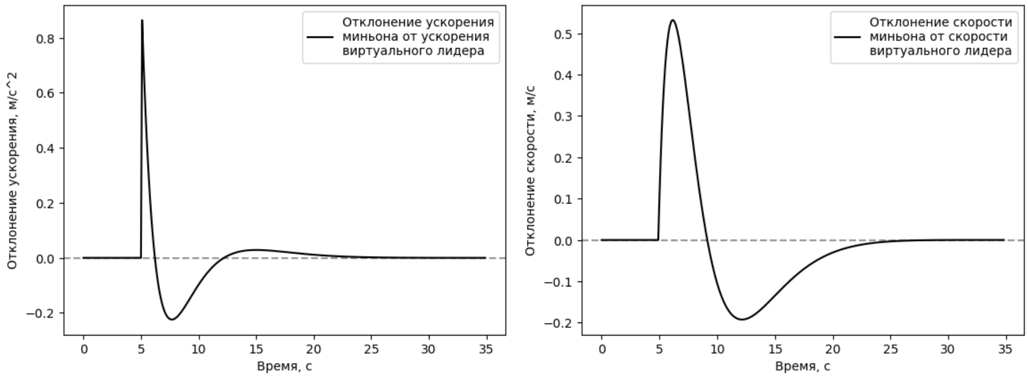


Рис. 7: Отклонение положения миньона от его виртуального агента



(a) Отклонение ускорения (b) Отклонение скорости

Рис. 8: Иллюстрация отклонений от закона движения виртуального лидера. Выше нуля — значение меньше, чем у лидера, ниже нуля — выше, чем у лидера.

2.2.2 Выводы

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

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.9.

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

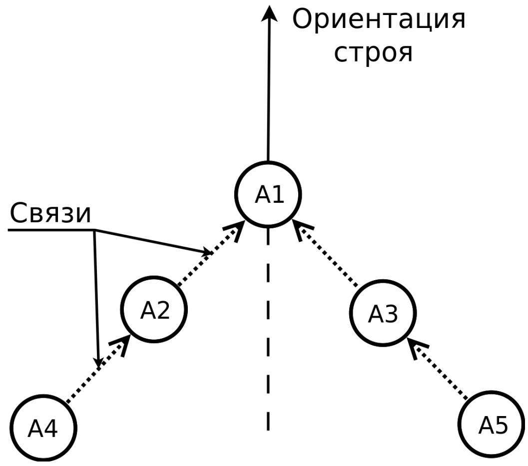


Рис. 9: Пример клиновидного строя из 5-ти агентов

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 счёт чего далее пересчитывается позиция, в которой должен находиться агент после поворота. Эта позиция будет именоваться виртуальным лидером. Виртуальный лидер — точка пространства, в которую стремиться попасть миньон для того, чтобы поддерживать фигуру строя.

Например, на рис. 10 изображён поворот строя на угол b. Для того, чтобы пересчитать позицию, в которой должен находиться миньон после поворота (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 расчитан виртуальный лидер A'_5 для агента A_5 (см. рис. 10), для которого определён вектор связи $J_{53} = (r_{53};\ \varphi_{53})$ (вектор связи с A_3). Для начала повернём вектор изначальной ориентации O_0:

$$\vec{O_1} = \begin{pmatrix} \cos{(b+\varphi_{53})} & \sin{(b+\varphi_{53})} \\ \sin{(b+\varphi_{53})} & \cos{(b+\varphi_{53})} \end{pmatrix} \vec{O_0}$$

Теперь вдоль полученного вектора $\vec{O_1}$ необходимо отложить от позиции

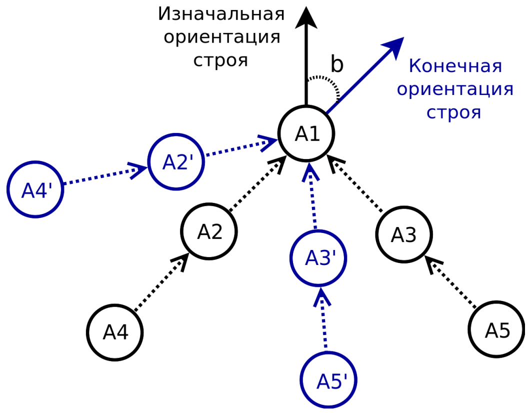


Рис. 10: Поворот строя на угол b

агента A_3 расстояние равное r_{53}:

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

В приведённых выше расчётах неявно предполагалось, что агент A_3 на момент пересчёта виртуального лидера для A_5 уже находится в точке A'_3. До тех пор, пока не будет выполнено это условие получаемое A'_5 не будет совпадать с тем, что приведено на рисунке. Однако, это ожидаемая ситуация, т.к. A_5 не знает ни о каких агентах кроме A_3, то он и полагается только на данные, получаемые от A_3, а если A_3 ещё не достиг точки виртуального лидера, то и в координтах виртуального лидера A'_5 будут содержаться ошибки.

Описанную выше проблему несложно решить, если строить виртуальных лидеров на основе позиции не агента-лидера, а на основе точки виртуального лидера агента-лидера. Однако, после поворота у миньонов находящихся дальше от центра поворота (центр поворота — мастер) расстояние до их виртуальных лидеров окажется больше, чем у миньонов ближе к центру поворота (см. A_3 — A'_3 в сравнении с A_5 — A'_5 на рис. 10), что приведёт к тому, что пропорциональная составляющая PID регулятора вырабатывающего управление для следования миньонов в точку виртуального лидера выдаст команду создать большее управляющее воздействие, следовательно АКБ агентов будут разряжаться более неравномерно, чем при более инертном рассчёте виртуальных лидеров от позциии агентов-лидеров.

3.1 Моделирование движения строя

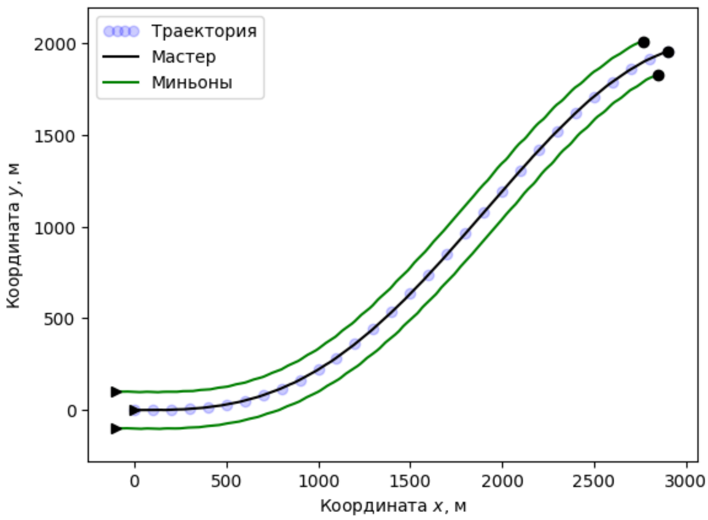


Рис. 11: Движение строем мастера и двух миньонов