

УДК 519.652

# АЛГОРИТМ ГЕНЕРАЦИИ ДВУМЕРНЫХ ТРАЕКТОРИЙ С ЗАДАНЫМИ КИНЕМАТИЧЕСКИМИ ХАРАКТЕРИСТИКАМИ ДЛЯ ФОРМИРОВАНИЯ СИНТЕТИЧЕСКИХ НАБОРОВ ДАННЫХ

**М.И. Костючек**

*Институт проблем управления им. В.А. Трапезникова РАН*

Россия, 117997, Москва, Профсоюзная ул., 65

E-mail: [max31@list.ru](mailto:max31@list.ru)

**А.В. Макаренко**

*Институт проблем управления им. В.А. Трапезникова РАН*

Россия, 117997, Москва, Профсоюзная ул., 65

E-mail: [avm.science@mail.ru](mailto:avm.science@mail.ru)

**Ключевые слова:** генерация траекторий, кинематика, синтетический набор данных, глубокое обучение.

**Аннотация:** В работе представлен подход к генерации синтетических траекторий в кинематическом приближении (контролируется местоположение, скорость, тангенциальное и центростремительное ускорения объекта, форма траектории в целом). Траектории формируются с помощью соединенных последовательно дуг окружностей. Приведены примеры работы алгоритма.

## 1. Введение

Для задач глубокого обучения, объем и качество наборов данных очень важны [1]. Ряд задач, связанных с подвижными объектами, требуют для своего решения массивов траекторных данных. Для снижения издержек (особенно на первых этапах) рекомендуется применять синтетические наборы. Многие известные методы генерации траекторий [2–4] обладают различными недостатками: либо они излишне сложны – используют динамическое моделирование, либо излишне примитивны – исходят из геометрического приближения (траектория представляется в виде набора ломанных линий). В настоящей работе предлагается простое и сбалансированное решение, позволяющее гибко управлять кинематическими характеристиками движения и формой траектории в целом.

## 2. Формальное описание подхода

Введём в рассмотрение траекторию  $\mathbf{r}(t) = [x(t), y(t)]$  на двумерной плоскости с определенной на ней декартовой системой координат,  $t \in T \subset \mathbb{R}$  – время. Определим условие по непрерывности скорости, тогда  $\mathbf{r}(T) \in C^1$ .

Будем формировать траекторию  $\mathbf{r}(t)$  в виде набора дуг окружностей соединенных последовательно так, что конец предыдущей дуги являлся началом следующей дуги. Для обеспечения условия по непрерывности скорости необходимо, чтобы в точке сшивки соседних  $n$ -й и  $(n+1)$ -й дуг окружностей угол наклона и направление их касательных векторов совпадали. Тогда такая траектория  $\mathbf{r}(t)$  легко параметризуется натуральным параметром  $l = l(t)$ , т.е. ее длиной. Причем  $n$ -я дуга будет задана на полуинтервале  $l \in L_n = (l_{n-1}, l_n]$ ,  $l_0 = 0$ , и  $L_n = l_n - l_{n-1}$  – есть длина  $n$ -й дуги. В общем случае, траектория  $\mathbf{r}(t)$  может формироваться из последовательности дуг окружностей двух типов:  $\gamma_n = -1$  – изображающая точка движется по часовой стрелке;  $\gamma_n = 1$  – изображающая точка движется против часовой стрелки. Тогда координаты дуги  $n$ -й окружности, возможно представить в виде:

$$(1) \quad x = x_{on} + R_n \cos B_n, \quad y = y_{on} + R_n \sin B_n, \quad B_n = \frac{\kappa_n + \gamma_n l}{R_n} + \theta_n,$$

где:  $(x_{on}, y_{on})$  – координаты центра окружности  $n$ -й дуги;  $\kappa_n$  – обеспечивает параметризацию окружности на полуинтервале  $L_n$ , чтобы траектория непрерывно зависела от параметра  $l$  по всей ее длине;  $\theta_n$  – обеспечивает требование равенства и однонаправленности касательных векторов двух соседних дуг.

Чтобы определить параметр  $\kappa_n$  рассмотрим вспомогательный случай  $l' \in (0, L_n]$ :

$$(2) \quad x = x_{on} + R_n \cos\left(\gamma_n \frac{l'}{R_n} + \theta_n\right), \quad y = y_{on} + R_n \sin\left(\gamma_n \frac{l'}{R_n} + \theta_n\right).$$

Из совместного рассмотрения (1) и (2) следует  $\kappa_n = -\gamma_n l_{n-1}$ .

Коэффициент  $\theta_n$  с учетом двух возможных ориентаций окружностей определяется через условие:

$$\theta_n = \begin{cases} \phi_{n-1}, & \gamma_{n-1} = \gamma_n \\ \phi_{n-1} + \pi, & \gamma_{n-1} = -\gamma_n \end{cases},$$

где  $\phi_{n-1}$  – так называемый нормализованный полярный угол конца  $(n-1)$ -й дуги, который определяется как обычный полярный угол при условии переноса координат центра окружности  $(n-1)$ -й дуги в точку  $(0, 0)$ .

При введенной параметризации  $l = l(t)$ , с учетом (1), скорость объекта  $v = v(t)$  по всей траектории будет иметь вид

$$v(t) = \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} = \left| \dot{l}(t) \right| = \dot{l}(t).$$

Предложенный способ построения траектории, в отличие от многих других (например от сплайнов), позволяет легко управлять кривизной траектории  $k$  в любой

ее точке через зависимости:

$$k = \frac{1}{R(l)}, \quad R(l) = \begin{cases} R_1, & l \in L_1 \\ R_n, & l \in L_n \\ \dots & \\ R_N, & l \in L_N \end{cases},$$

где  $N$  – число элементарных сегментов из которых состоит траектория. В купе с заданием зависимости натурального параметра от времени  $l = l(t)$ , и ограничений на эту зависимость, легко накладываются ограничения на кинематические характеристики движения: скорость  $v(t)$ , тангенциальное  $a_\tau(t)$  и центростремительное  $a_n(t)$  ускорения. Для ускорений справедливы формулы:

$$a_\tau(t) = \dot{v}(t) = \ddot{l}(t), \quad a_n(t) = \frac{v^2(t)}{R[l(t)]} = \frac{\dot{l}^2(t)}{R[l(t)]}.$$

Таким образом, через ограничения на первую и вторую производные  $l(t)$ , возможно непосредственно ограничить скорость и тангенциальное ускорение, а ограничениями на первую производную  $l(t)$  и радиусы окружностей  $R_n$  – ограничивается центростремительное ускорение.

### 3. Алгоритм построения траектории

Исходно задаем число дуг окружностей  $N$ , ограничения на радиус окружностей  $R_{\min} \leq R_n \leq R_{\max}$  и на угловые меры дуг  $\varphi_{\min} \leq \varphi_n \leq \varphi_{\max}$ , из которых будет состоять траектория, причём  $n = 1, N$ .

Далее случайным образом выбираем начальную точку дуги первой окружности  $(x_1, y_1)$ , которая будет начальной точкой траектории, и нормализованный полярный угол конца виртуальной нулевой дуги  $\phi_0$ . Зная эти параметры, мы можем построить нормаль к окружности на которой будет лежать координата центра окружности. Затем случайно выбираем абсциссу и ординату центра окружности с учетом ограничений на радиус (RND – функция случайного выбора,  $R'_{\min} = R_{\min} \cos \phi_0$ ,  $R'_{\max} = R_{\max} \cos \phi_0$ ):

1. Если  $\phi_0 = \pi/2$ , то  $x_{o1} = x_1$ ,  $y_{o1} = \text{RND}[y_1 - R_{\max}, y_1 - R_{\min}]$ .
2. Если  $\phi_0 = 3\pi/2$ , то  $x_{o1} = x_1$ ,  $y_{o1} = \text{RND}[y_1 + R_{\min}, y_1 + R_{\max}]$ .
3. Если  $\pi/2 < \phi_0 < 3\pi/2$ , то  $x_{o1} = \text{RND}[x_1 + R'_{\min}, x_1 + R'_{\max}]$ ,  
 $y_{o1} = (x_{o1} - x_1) \text{tg} \phi_0 + y_1$ .
4. Если  $\phi_0 < \pi/2$  или  $\phi_0 > 3\pi/2$ , то  $x_{o1} = \text{RND}[x_1 - R'_{\max}, x_1 - R'_{\min}]$ ,  
 $y_{o1} = (x_{o1} - x_1) \text{tg} \phi_0 + y_1$ .

Далее случайным образом выбираем угловую меру первой дуги  $\varphi_1$ , вычисляем радиус первой окружности  $R_1 = \sqrt{(x_{o1} - x_1)^2 + (y_{o1} - y_1)^2}$  и вычисляем длину траектории на данном шаге  $l_1 = R_1 \varphi_1$ . Затем случайно выбираем тип первой окружности через задание параметра  $\gamma_1$  и определяем коэффициенты  $\kappa_1$  и  $\theta_1$ . Полученные величины подставляем в (1) и получаем формулы для описания координат точек

дуги первой окружности. В конце шага алгоритма вычисляем нормализованный полярный угол конца первой дуги:

$$\phi_1 = \begin{cases} \phi_0 - \varphi_1, & \phi_0 - \varphi_1 \geq 0 \text{ и } \gamma_1 = -1, \\ \phi_0 - \varphi_1 + 2\pi, & \phi_0 - \varphi_1 < 0 \text{ и } \gamma_1 = -1, \\ \phi_0 + \varphi_1, & \phi_0 + \varphi_1 < 2\pi \text{ и } \gamma_1 = 1, \\ \phi_0 + \varphi_1 - 2\pi, & \phi_0 + \varphi_1 \geq 2\pi \text{ и } \gamma_1 = 1. \end{cases}$$

В цикле переходим к генерации  $n$ -й окружности, где  $n = \overline{2, N}$ .

Через (1) вычисляем начальную точку дуги  $n$ -й окружности  $(x_n, y_n)$ , которая является концом  $(n-1)$ -й дуги. Затем случайным образом выбираем тип  $n$ -й окружности, через задание параметра  $\gamma_n$ , и координаты ее центра  $x_{on}, y_{on}$ , с учетом ограничений на радиус ( $R'_{\min} = R_{\min} \cos \phi_{n-1}$ ,  $R'_{\max} = R_{\max} \cos \phi_{n-1}$ ):

Если  $\gamma_{n-1} = \gamma_n$ :

1. Если  $\phi_{n-1} = \pi/2$ , то  $x_{on} = x_n, y_{on} = \text{RND}[y_n - R_{\max}, y_n - R_{\min}]$ .
2. Если  $\phi_{n-1} = 3\pi/2$ , то  $x_{on} = x_n, y_{on} = \text{RND}[y_n + R_{\min}; y_n + R_{\max}]$ .
3. Если  $\pi/2 < \phi_{n-1} < 3\pi/2$ , то  $x_{on} = \text{RND}[x_n - R'_{\min}, x_n - R'_{\max}]$ ,  
 $y_{on} = (x_{on} - x_n) \text{tg } \phi_{n-1} + y_n$ .
4. Если  $\phi_{n-1} < \pi/2$  или  $\phi_{n-1} > 3\pi/2$ , то  $x_{on} = \text{RND}[x_n - R'_{\max}, x_n - R'_{\min}]$ ,  
 $y_{on} = (x_{on} - x_n) \text{tg } \phi_{n-1} + y_n$ .

Если  $\gamma_{n-1} \neq \gamma_n$ :

1. Если  $\phi_{n-1} = \pi/2$ , то  $x_{on} = x_n, y_{on} = \text{RND}[y_n + R_{\min}, y_n + R_{\max}]$ .
2. Если  $\phi_{n-1} = 3\pi/2$ , то  $x_{on} = x_n, y_{on} = \text{RND}[y_n - R_{\max}, y_n - R_{\min}]$ .
3. Если  $\pi/2 < \phi_{n-1} < 3\pi/2$ , то  $x_{on} = \text{RND}[x_n + R'_{\max}, x_n + R'_{\min}]$ ,  
 $y_{on} = (x_{on} - x_n) \text{tg } \phi_{n-1} + y_n$ .
4. Если  $\phi_{n-1} < \pi/2$  или  $\phi_{n-1} > 3\pi/2$ , то  $x_{on} = \text{RND}[x_n + R'_{\min}, x_n + R'_{\max}]$ ,  
 $y_{on} = (x_{on} - x_n) \text{tg } \phi_{n-1} + y_n$ .

Далее для  $n$ -й дуги случайным образом выбираем угловую меру дуги  $\varphi_n$ , вычисляем радиус окружности  $R_n = \sqrt{(x_{on} - x_n)^2 + (y_{on} - y_n)^2}$  и рассчитываем длину траектории на данном шаге  $l_n = R_n \varphi_n + l_{n-1}$ . Определяем коэффициенты  $\kappa_n$  и  $\theta_n$ . Полученные величины подставляем в (1) и получаем формулы для описания координат точек дуги  $n$ -й окружности. Вычисляем нормализованный полярный угол конца  $n$ -й дуги ( $\phi_- = \phi_{n-1} - \varphi_n$ ,  $\phi_+ = \phi_{n-1} + \varphi_n$ ):

Если  $\gamma_{n-1} = -1$  и  $\gamma_n = -1$ ,  $\gamma_n = 1$ , соответственно:

$$\phi_n = \begin{cases} \phi_-, & \phi_- \geq 0, \\ \phi_- + 2\pi, & \phi_- < 0. \end{cases}, \quad \phi_n = \begin{cases} \phi_- + \pi, & \phi_+ + \pi < 2\pi, \\ \phi_+ - \pi, & \phi_+ + \pi \geq 2\pi. \end{cases}$$

Если  $\gamma_{n-1} = 1$  и  $\gamma_n = -1$ ,  $\gamma_n = 1$ , соответственно:

$$\phi_n = \begin{cases} \phi_- + \pi, & \phi_+ + \pi \geq 0, \\ \phi_- + 3\pi, & \phi_+ + \pi < 0. \end{cases}, \quad \phi_n = \begin{cases} \phi_+, & \phi_- < 2\pi, \\ \phi_+ - 2\pi, & \phi_- \geq 2\pi. \end{cases}$$

В конце шага алгоритма проверяем условие выхода:  $n = N$ .

Следует отметить, что приведенный алгоритм позволяет гибко управлять формой синтезируемых траекторий. Это достигается посредством формирования соответствующих последовательностей из коэффициентов  $\gamma_n$  – тип окружности и  $\varphi_n$  – угловая мера дуги.

## 4. Примеры работы алгоритма

Различные режимы генерации траекторий представлены на рисунках: **1a** – псевдослучайный равновероятный выбор типа окружностей; **1b** – вероятность смены типа окружности определяется функциональной зависимостью  $P(k) = 2/\pi \operatorname{arctg}(k/80)$ , где  $k$  – число окружностей одного типа, идущих подряд. Тип окружности меняется при выполнении условия  $\xi_n < P(k)$ , здесь  $\xi$  – равномерно распределенное на интервале  $[0, 1]$  псевдослучайное число. После смены типа окружности принимается  $k = 1$ . Следует отметить, что все прочие параметры одинаковы для обоих вариантов:  $\varphi_{\max} = 0.5$  рад,  $R_{\min} = 0.025$ ,  $R_{\max} = 0.25$ .

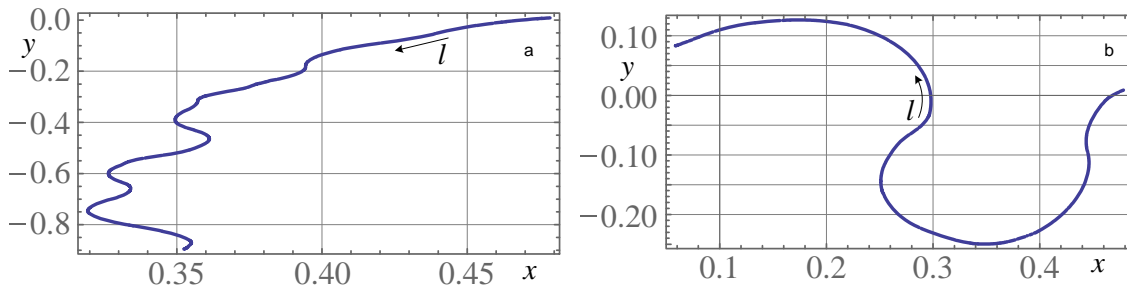


Рис. 1. Примеры траекторий с различными правилами псевдослучайной смены типа окружностей: а – равновероятный выбор; б – вероятность смены повышается с числом окружностей одного типа идущих подряд.

## 5. Заключение

Представлен алгоритм для формирования двумерных гладких непрерывных траекторий, который позволяет накладывать ограничения как на вид траектории, так и на кинематические характеристики движения. Траектории, полученные данным алгоритмом, легко параметризуются от натурального параметра и времени. Алгоритм позволяет гибко управлять формой синтезируемых траекторий.

В заключение отметим, что предложенное решение реализовано в виде подключаемого модуля на языке программирования Wolfram Language. Оно было применено при решении, с помощью глубоких нейронных сетей, задачи классификации подвижных объектов на основе кинематических характеристик их движения.

## Список литературы

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. London: MIT Press, 2016. 778 p.

2. Giannotti F., Nanni M., Pedreschi D., Pinelli F. Trajectory Pattern Mining // 13th ACM SIGKDD international conference on Knowledge discovery and data mining. California, 2007. P. 330-339.
3. Hervieu A., Bouthemy P., Le Cadre J-P. A HMM-Based Method for Recognizing Dynamic Video Contents from Trajectories // 2007 IEEE International Conference on Image Processing. San Antonio, 2007. Vol. 4. P. 533-536.
4. Pelekis N., Ntrigkogias C., Tampakis P., Sideridis S., Theodoridis Y. Hermoupolis: A Trajectory Generator for Simulating Generalized Mobility Patterns // Blockeel H., Kersting K., Nijssen S., Zelezny F. (Eds.) Machine Learning and Knowledge Discovery in Databases. Berlin, Heidelberg: Springer, 2013. Vol. 3. P. 659-662.