Повышение эффективности алгоритма KRRT* за счет использования внешних оценок множеств достижимости

студент 4 курса Н. Ю. Заварзин
научный руководитель — к.ф.-м.н., доцент П. А. Точилин

Кафедра системного анализа
факультета ВМК МГУ имени М.В. Ломоносова

3 июня 2024 г.

Постановка задачи

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + C, \\ x(t_0) = x_{init}, \ \ x(t_1) \in \mathcal{X}_{goal}. \end{cases} \tag{1}$$

$$x(t) \in \mathcal{X} \text{ — пространство состояний}, \ \ A, B, C \equiv \mathrm{const},$$

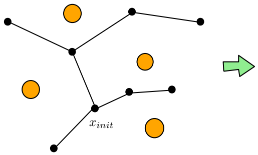
$$u(t) \in \mathcal{U} \text{ — мн-во допустимых управлений}, \mathcal{X}_{obs} \text{ — область препятствий.}$$

$$\mathcal{J}(u(\cdot)) = \int\limits_{t_0}^{t_1} \langle x(t), Qx(t) \rangle + \langle u(t), Ru(t) \rangle \, \mathrm{d}t,$$

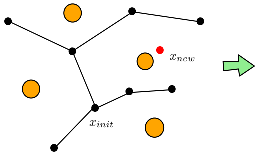
$$\text{где } Q \geqslant 0, \ \ R > 0.$$

Необходимо найти программное управление $u(\cdot) \in \mathcal{U}$, которое минимизирует функционал $\mathcal{J}(u(\cdot))$ на беспрепятственных траекториях системы (1). Беспрепятственными будем называть траектории $x(t) \in \mathcal{X} \setminus \mathcal{X}_{obs}$ $\forall t \in [t_0, t_1]$.

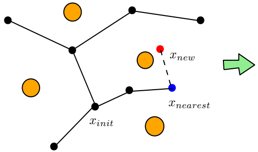
Принцип работы алгоритма RRT*



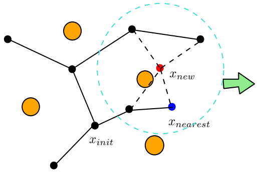
n-ая итерация



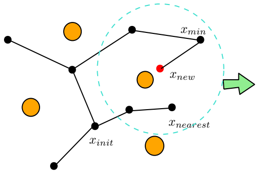
Генерация нового состояния



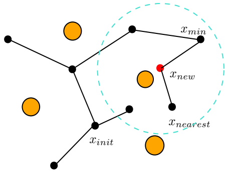
Поиск ближайшей (эвристика)



Поиск ближайшей (по функционалу)

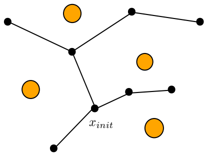


Соединение

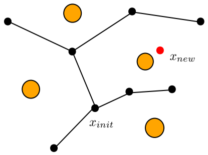


Перемонтирование

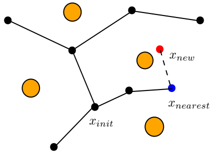
Принцип работы алгоритма KRRT*



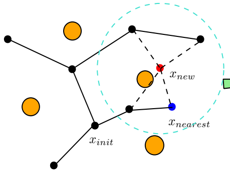
n-ая итерация



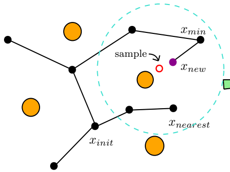
Генерация нового состояния



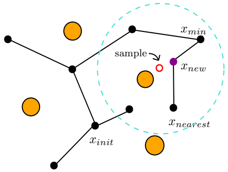
Поиск ближайшей (эвристика)



Поиск ближайшей (по функционалу)

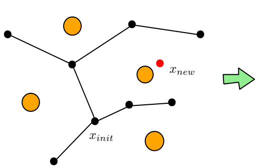


Соединение

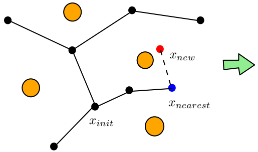


Перемонтирование

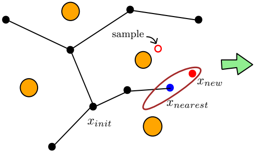
Принцип работы алгоритма MyKRRT*



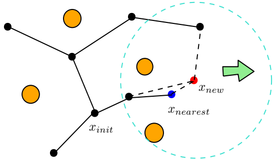
Генерация



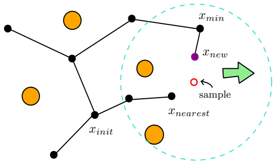
Поиск ближайшей (эвристика)



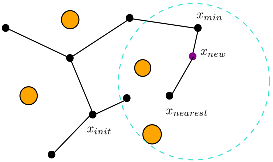
Построение внешней оценки



Поиск ближайшей (по функционалу)



Соединение



Перемонтирование

Построение эллипсоидальных оценок

Множество достижимости рассматриваемой системы задаётся как

$$X[t] = F(t,t_0)x_0 + \int_{t_0}^{t} F(t,\tau)(B\mathcal{E}(q(\tau),P(\tau)) + C)\,\mathrm{d}\tau =$$

$$= \left[F(t,t_0)x_0 + \int_{t_0}^{t} F(t,\tau)C\,\mathrm{d}\tau\right] + \int_{t_0}^{t} F(t,\tau)B\mathcal{E}(q(\tau),P(\tau))\,\mathrm{d}\tau =$$

$$= v(t) + \int_{t_0}^{t} \mathcal{E}(F(t,\tau)Bq(\tau), F(t,\tau)BP(\tau)B^T F^T(t,\tau))\,\mathrm{d}\tau.$$

Тогда

$$q_+(t) = v(t) + \int_{t_0}^{t} F(t,\tau)Bq(\tau)\,\mathrm{d}\tau,$$

$$X_+[t] = \left(\int_{t_0}^{t} p(\tau)\,\mathrm{d}\tau\right)\left(\int_{t_0}^{t} p^{-1}(\tau)F(t,\tau)BP(\tau)B^T F^T(t,\tau)\,\mathrm{d}\tau\right),$$

где

$$p(\tau) = \left\langle l_0, F(t_0,\tau)BP(\tau)B^T F^T(t_0,\tau)l_0\right\rangle^{\frac{1}{2}}.$$

Генерация новых состояний

Из равенства опорных функций:

$$\mathcal{E}(q, Q) = q + Q^{\frac{1}{2}}\mathcal{E}(0, I) = q + Q^{\frac{1}{2}}S(0, 1),$$

то есть гиперэллипсоид является линейным преобразованием единичной гиперсферы. Покажем, как свести задачу равномерного сэмплирования в гиперэллипсоиде к таковой для единичной гиперсферы:

$$p_{ball}(x) = \begin{cases} \frac{1}{\zeta_n}, & \forall x \in X_{ball}, \\ 0, & \text{иначе}, \end{cases}$$

$$x_{ellipse} = g(x_{ball}) = Q^{\frac{1}{2}}x_{ball} + q,$$

$$p_{ellipse}(x) = p_{ball}(g^{-1}(x))\left| \det\left\{ \left.\frac{\mathrm{d}g^{-1}}{x_{ellipse}}\right|_x \right\} \right|,$$

$$p_{ellipse}(x) = \begin{cases} \frac{1}{\zeta_n}\left| det\left\{ \left(Q^{\frac{1}{2}}\right)^{-1} \right\}\right|, & \forall x \in X_{ellipse}, \\ 0, & \text{иначе}. \end{cases}$$

Функция рулевого управления

Перед нами стоит задача найти программное управление, минимизирующее исходный функционал на

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + C, \\ x(t_0) = q_1, \; x(T) = q_2. \end{cases} \tag{2}$$

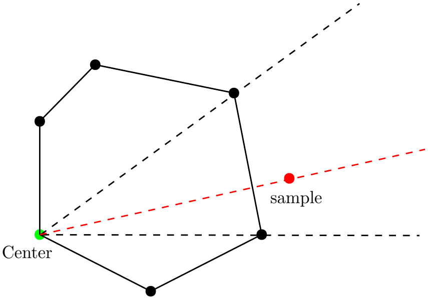
Откажемся от жёсткого условия на правом конце в пользу дополнительного слагаемого в функционале с $S \geqslant 0$:

$$\mathcal{J}(u(\cdot)) = \int\limits_{t_0}^{t_1} \langle x(t), Qx(t) \rangle + \langle u(t), Ru(t) \rangle \, \mathrm{d}t + \langle x(T) - q_2, S(x(T) - q_2) \rangle \, .$$

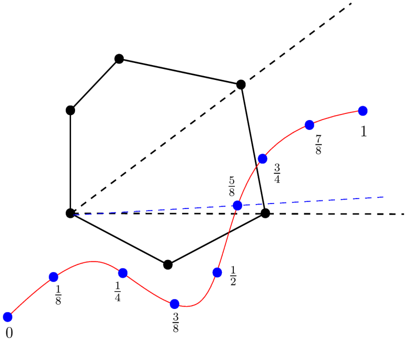
Тогда через уравнение Беллмана задача сводится к матричному уравнению Риккати:

$$\begin{cases} \dot{P}(t) + P(t)A + A^T P(t) - P(t)BR^{-1}B^T P(t) + Q = 0, \\ P(T) = S. \end{cases}$$

Проверка на столкновение



Проверка конфигурации



Проверка пути

- При правильно подобранных параметрах предлагаемая модификация алгоритма исследует пространство также широко, как и базовая реализация $\mathtt{KRRT}^*$, но при этом делает это с большей степенью дискретизации.
- Использование эвристики сэмплирования снижает число холостых итераций. При этом в среднем алгоритм ускоряется в 1.1 раз, а в отдельных случаях в два и более раз.