Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра системного анализа

Отчёт по практикуму

«Построение множества достижимости»

Студент 315 группы
В. А. Сливинский

Руководитель практикума
к.ф.-м.н., доцент П. А. Точилин

Москва, 2018

Содержание

1 Постановка задачи

1.1 Формулировка задачи

Задано следующее обыкновенное дифференциальное уравнение:

$$\ddot{x} + x\dot{x} - \operatorname{arctg}(x^2) + x^2\cos(x^2) = u \tag{1.1}$$

Здесь, $x \in \mathbb{R}$, $u \in \mathbb{R}$. Кроме того, на управление u наложено дополнительное ограничение $u \in \mathfrak{U}$, $\mathfrak{U} = [-\alpha, \alpha]$, $\alpha > 0$. Задан начальный момент времени $t_0 = 0$ и начальная позиция $x(t_0) = \dot{x}(t_0) = 0$. Необходимо построить множество достижимости $\mathcal{X}(t, t_0, x(t_0), \dot{x}(t_0))$ (множество пар $(x(t), \dot{x}(t))$) в классе программных управлений в заданный момент времени $t \geqslant t_0$. Требуется:

1. Написать в среде Matlab функцию `reachset(alpha, t)`, которая по заданным значениям параметров $\alpha > 0$, $t \geqslant t_0$ рассчитывает приближенно множество достижимости $\mathcal{X}(t, t_0, x(t_0), \dot{x}(t_0))$. На выходе функции — два массива `X, Y` с упорядоченными координатами точек многоугольника, аппроксимирующего границу множества достижимости. Точки в этих массивах должны быть упорядочены так, чтобы их без дополнительной обработки можно было подавать на фход функции `plot`. Также, функция должна предусматривать режим работы, в котором, помимо границы множества достижимости, возвращаются также координаты линии переключения оптимального управления;

2. Реализовать функцию `reachsetdyn(alpha, t1, t2, N, filename)`, которая, используя функцию `reachset`, строит множества достижимости для моментов времени $\tau_i = t_1 + \frac{(t_2 - t_1)\cdot i}{N}$. Здесь, $t_2 \geqslant t_1 \geqslant t_0$, N — натуральное число. Для каждого момента времени τ_i функция должна отобразить многоугольник, аппроксимирующий границу множества достижимости. Результат работы функции должен быть сохранён в виде видеофайла `filename.avi`. Необходимо также предусмотреть вариант работы функции (при отсутствии параметра `filename`) без сохранения в файл, с выводом непосредственно на экран. Как частный случай, при $t_2 = t_1$ функция должна строить границу множества достижимости в один фиксированный момент времени.

1.2 Формализация задачи

Прежде всего, нормализуем систему (1.1):

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u - x_1^2 \cos(x_1^2) + \operatorname{arctg}(x_1^2) - x_1 \cdot x_2 \end{cases} \tag{1.2}$$

Здесь, $x_1 = x$, $x_2 = \dot{x}$. Управление u будем полагать *кусочно–непрерывной* функцией.

Определение 1.1. Для любого времени $t \geqslant t_0$ множество

$$\mathcal{X}[t] = \mathcal{X}(t, t_0, x_1(t_0), x_2(t_0)) =$$
$$= \Bigg\{ (x_1(t), x_2(t)) \in \mathbb{R}^2 : \ \exists u^* = u^*(t) : \ |u*(\tau)| \leqslant \alpha \ \dot{\forall} \tau \in [t_0, t],$$
$$(x_1(t), x_2(t)) = (x_1(t, t_0, x_1(t_0), x_2(t_0)), x_2(t, t_0, x_1(t_0), x_2(t_0)))\Big|_{u^*(t)} \Bigg\}$$

будем называть *множеством достижимости для системы* (1.2) *в момент времени* $t \geqslant t_0$.

2 Некоторые необходимые теоретические выкладки

Для начала, выпишем для задачи (1.2) функцию Гамильтона–Понтрягина:

$$\mathcal{H}(\psi, x, u) = \langle \psi, f(x,u) \rangle = \psi_1 x_2 + \psi_2 \left(u - x_1^2 \cos(x_1^2) + \operatorname{arctg}(x_1^2) - x_1 \cdot x_2 \right)$$

Введём вспомогательную функцию

$$\mathcal{M}(\psi, x) = \sup_{u \in [-\alpha, \alpha]} \mathcal{H}(\psi, x, u).$$

Теперь, сформулируем принцип максимума Понтрягина для задачи достижимости

Принцип Максимума Понтрягина (в формулировке из [1])
Пусть $(x^*(\cdot), u^*(\cdot))$ *— оптимальная по быстродействию пара для задачи* (1.2) *при* $t_1 = t_1^*$ *(здесь* $t \in [0, t_1^*]$*). Тогда существует функция* $\psi^*(\cdot) : [0, t_1^*] \to \mathbb{R}^2$ *такая, что:*

$$\psi^* \not\equiv 0 \tag{2.1}$$

$$\dot{\psi^*}(t) = -\frac{\partial \mathcal{H}}{\partial x}\Big|_{\substack{\psi=\psi^*(t) \\ x=x^*(t) \\ u=u^*(t)}} \tag{2.2}$$

$$u^*(t) \in \operatorname{Argmax}_{u \in [-\alpha, \alpha]} \mathcal{H}(\psi^*(t), x^*(t), u) \ \dot\forall t \in [0, t_1^*] \tag{2.3}$$

$$\mathcal{M}(\psi^*(t), x^*(t)) \equiv \text{const} \geqslant 0 \tag{2.4}$$

Доказательство данной теоремы в общем случае приведено, например, в [3]. Систему (2.2) называют *сопряжённой системой*, условие (2.3) — условием максимума, а условие (2.1) — условием нетривиальности (из него следует, что $\psi^*(t) \neq 0$ для всех $t \in [0, t_1^*]$). Условие (2.3) позволит в явном виде выписать оптимальное управление.
Сформулируем также *теорему о нулях* ψ_2 *и* x_2[1]:

Теорема 2.1. *Пусть* $\tau_1 < \tau_2$ *и*

1. $\psi_2(\tau_1) = \psi_2(\tau_2) = 0$ *и* $x_2(\tau_1) = 0 \Rightarrow x_2(\tau_2) = 0$*;*
2. $x_2(\tau_1) = x_2(\tau_2) = 0$*,* $x_2(t) \neq 0$ *при* $t \in (\tau_1, \tau_2)$ *и* $\psi_2(\tau_1) = 0 \Rightarrow \psi_2(\tau_2) = 0$*.*
3. $\psi_2(\tau_1) = \psi_2(\tau_2) = 0$ *и* $x_2(\tau_1) \neq 0 \Rightarrow x_2(\tau_2) \neq 0$*, но* $\exists t' \in (\tau_1, \tau_2) : x_2(t') = 0$*.*
4. $x_2(\tau_1) = x_2(\tau_2) = 0$*,* $x_2(t) \neq 0$ *при* $t \in (\tau_1, \tau_2)$ *и* $\psi_2(\tau_1) \neq 0 \Rightarrow \psi_2(\tau_2) \neq 0$*, но* $\exists t'' \in (\tau_1, \tau_2) : \psi_2(t'') = 0$*.*

Обратим внимание, что функция $\psi_2(\cdot)$ имеет не более чем конечное число нулей на отрезке $[0, t_1^*]$ (в противном случае, получим противоречие с (2.1)).

3 Описание алгоритма

Для начала, выпишем сопряженную систему:

$$\begin{cases} \dot\psi_1 = \psi_2 \cdot \left(x_2 + 2x_1 \cos(x_1^2) + 2x_1^3 \sin(x_1^2) - \dfrac{4x_1 \operatorname{arctg}(x_1)}{x_1^4 + 1} \right) \\ \dot\psi_2 = \psi_2 x_1 - \psi_1 \end{cases} \tag{3.1}$$

[1] Её доказательство приведено в [1]

Из (2.3) получим:

$$u^*(t) = \begin{cases} \alpha, & \psi_2(t) > 0, \\ [-\alpha, \alpha], & \psi_2(t) = 0, \\ -\alpha, & \psi_2(t) < 0. \end{cases}$$

Обратим внимание, что особого режима не возникает, так как в противном случае получаем нулевой вектор ψ, что противоречит (2.1). Дополнительно, обозначим за S_+ систему (1.2) при $u = \alpha$, а за S_- систему (1.2) при $u = -\alpha$. Стало быть, в силу теоремы (2.1), для решения задачи применим следующий алгоритм:

- Решаем систему S_+ из начальной (нулевой) точки до момента $t_+ :\ x_2(t_+) = 0$;
- Строим разбиение отрезка $[0, t_+]$. Обозначим точку этого разбиения через t^i_+;
- Решаем систему S_-, присоединив к ней сопряжённую систему (3.1), с начальными условиями $(x_1(t^i_+), x_2(t^i_+), 1, 0)$ (в силу инвариантности решения системы (3.1)) относительно умножения на положительную константу и в силу того, что $\psi_1(t_+) > 0$, можно нормировать $\psi_1(t_+) = 1$) до момента $\psi_2(\hat{t}^i_+)$ (из теоремы (2.1) следует, что такой момент найдётся);
- Далее решаем систему S_+ и присоединённую к ней сопряжённую систему (3.1) с начальными условиями $(x_1(\hat{t}^i_+), x_2(\hat{t}^i_+), -1, 0)$ до следующего переключения. Повторяем последние два пункта до момента t, заданного пользователем;
- Аналогичные действия проводим для системы S_-;
- Соединяем концы полученных траекторий;
- Исследуем отрезки полученной ломаной. Если расстояние между каким-то соседними точками больше некоторого фиксированного значения (параметра функции), то производится дополнительное подразбиение отрезка $[t_i, t_{i+1}]$, где t_i и t_{i+1} — моменты первого переключения для этих точек. Для этого подразбиения рассчитываются новые траектории, концы которых включаются в новую аппроксимацию. Повторяем эти действия, пока каждая сторона многоугольника не станет меньше или равна этого заданного значения. Таким образом, будет получена кривая соответствующей гладкости и устранена (с некоторой точностью, зависящей от заданного значения) ошибка попадания неподвижных точек системы внуть множества достижимости;
- Удаляем самопересечения. Пройдём по точкам полученной ломаной, запоминая моменты пересечения отрезков. Затем, удалим участки, заключенные между точками пересечений, если до них было чётное (или ноль) число пересечений; это позволяет устранить «петли».

Отметим, что, во–первых, для вывода кривой переключений достаточно лишь запоминать моменты переключения при решении соответствующих систем, а, во–вторых, поскольку одна из фазовых переменных в системе при достаточно больших значениях t и α очень быстро растёт, дополнительно будем «обрезать» траектории, выходящие за некоторый радиус (параметр функции). Помимо этого, в функции также ищутся стационарные точки систем S_+ и S_-: для этого, поскольку в них $x_2 = 0$, достаточно при

помощи функции `fzero` отыскать нули функций $\mathrm{arctg}(x_1^2) - x_1^2\cos(x_1^2) \pm \alpha$ с начальными приближениями из равномерной сетки $[x_{min}, x_{max}]$, где $x_{min} = \min\limits_{\mathtt{X}} x_1,\ x_{max} = \max\limits_{\mathtt{X}} x_1$.[1]

[1] `X` — вектор, возвращаемый функцией `reachset`, состоящий из абсцисс границы множества достижимости

4 Примеры работы программы

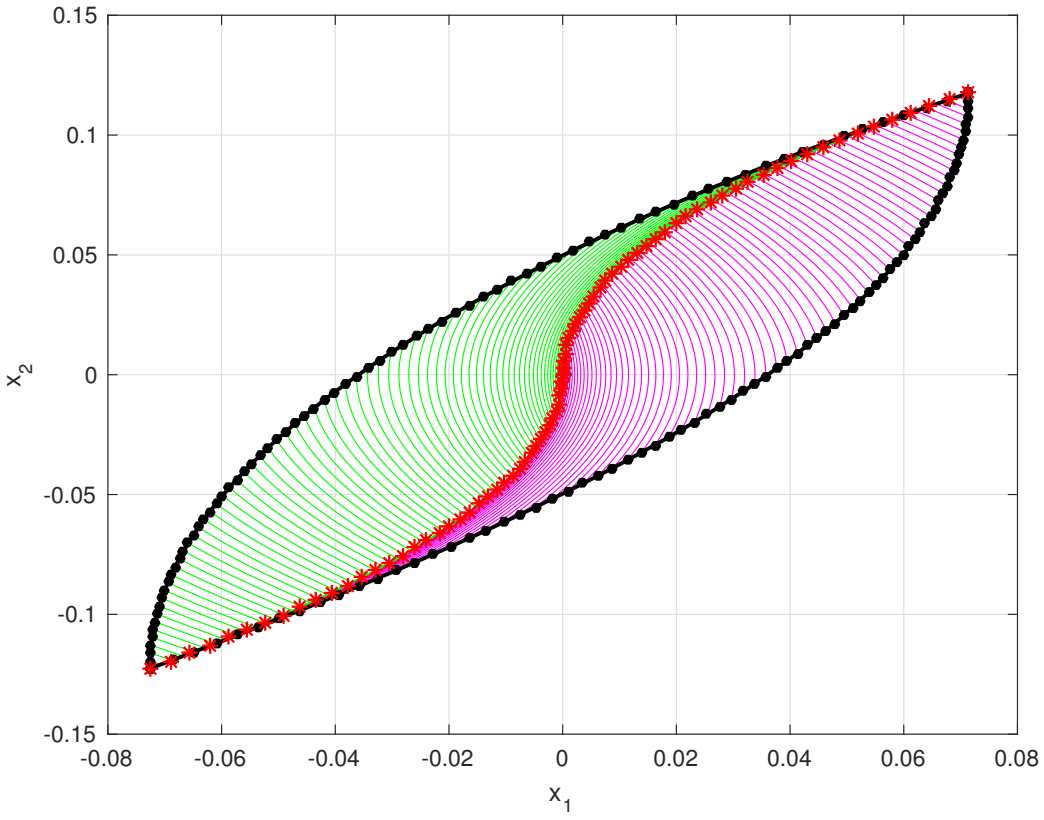

Рис. 1: Множество достижимости при $\alpha = 0.1$, $t = 1.2$, с трассировкой траекторий, красным отмечена кривая переключений, чёрным — граница множества достижимости

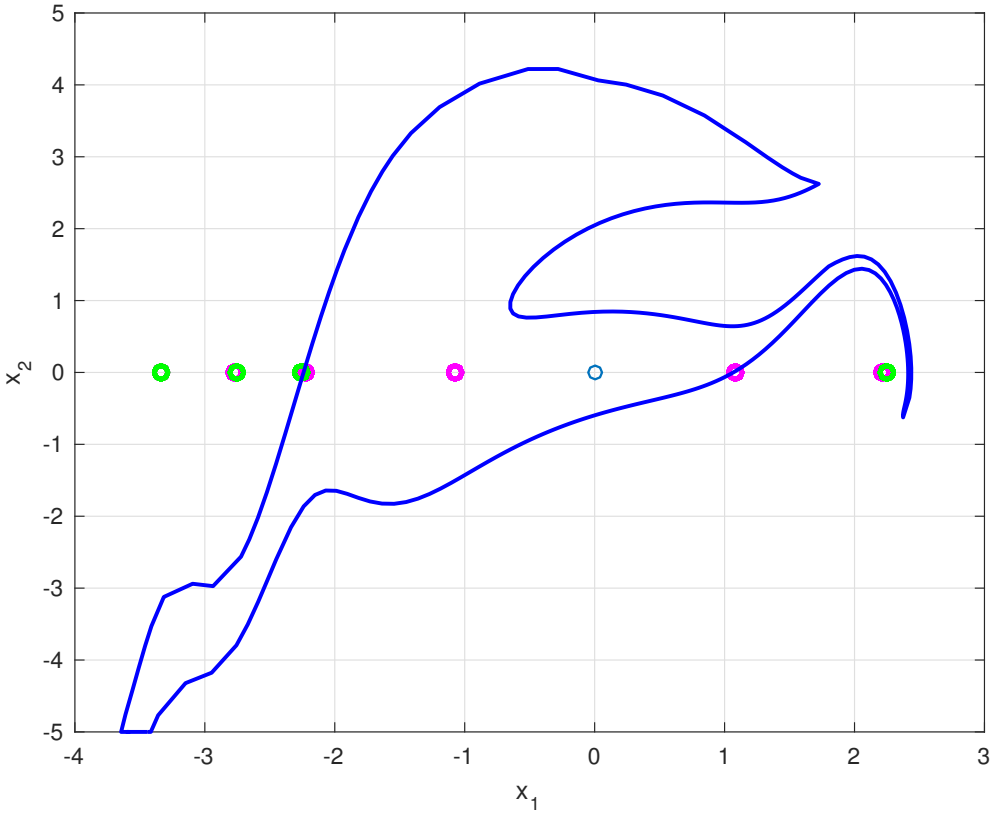

Рис. 2: Множество достижимости при $\alpha = 0.4$, $t = 4.25$, синим отмечена граница множества достижимости, дополнительно отмечены стационарные точки систем S_+ и S_-; здесь установлен сравнительно небольшой радиус ограничения траекторий `maxRadius` $= 5$

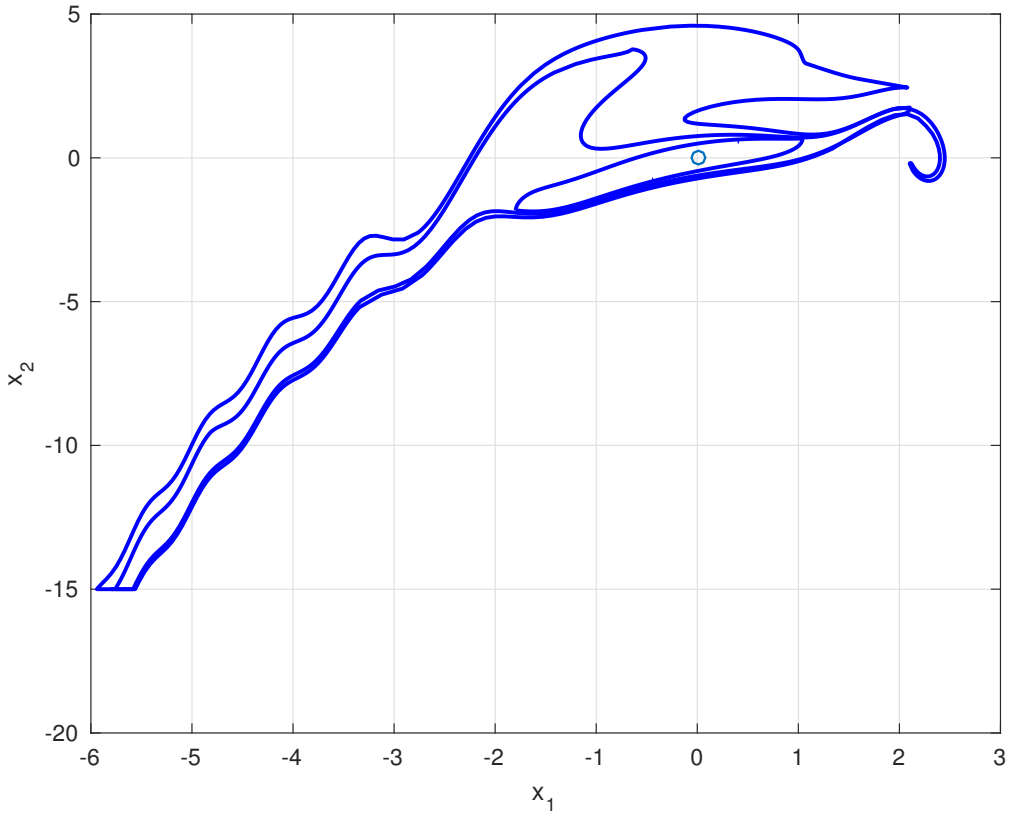

Рис. 3: Эволюция множества достижимости при $\alpha = 0.5$, $t = 1.5 \ldots 4.5$, синим отмечена граница множества достижимости

Список литературы

[1] И. В. Рублёв. *Лекционный курс Оптимальное Управление (Нелинейные Системы)*, кафедра Системного Анализа, Факультет Вычислительной Математики и Кибернетики, МГУ им. М. В. Ломоносова, 2018

[2] Точилин П. А. *Лекционный курс Программирование на языке* ***MATLAB***, кафедра Системного Анализа, Факультет Вычислительной Математики и Кибернетики, МГУ им. М. В. Ломоносова, 2017 – 2018

[3] Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. *Математическая теория оптимальных процессов*, — М.: Наука, 1976.

[4] Справочные средства языка `MATLAB`