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

Отчёт по практикуму

«Линейная Задача Быстродействия»

Студент 315 группы
В. А. Сливинский

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

Москва, 2018

Содержание

1 Постановка задачи

1.1 Общая формулировка задачи

Задана линейная система ОДУ:

$$\dot{x} = Ax + u + f,\ t \in [t_0, +\infty) \tag{1.1}$$

Здесь, $x, f \in \mathbb{R}^2$, $A \in \mathbb{R}^{2\times 2}$, $u \in \mathbb{R}^2$. Кроме того, на управление u наложено дополнительное ограничение $u \in \mathcal{P}$. Пусть $\mathcal{X}_0$ — начальное множество значений фазового вектора, $\mathcal{X}_1$ — целевое множество значений фазового вектора. Для заданных множеств $\mathcal{X}_0$, $\mathcal{X}_1$, $\mathcal{P}$ необходимо решить задачу быстродействия, т.е. найти минимальное время $T > 0$, за которое траектория системы, выпущенная в момент времени t_0 из некоторой точки множества $\mathcal{X}_0$, может попасть в некоторую точку множества $\mathcal{X}_1$.

$$\mathcal{P} = p + \left\{(x_1, x_2) \in \mathbb{R}^2 : 9x_1^2 + 4x_2^2 \leqslant r\right\},\ p \in \mathbb{R}^2; \tag{1.2}$$

$$\mathcal{X}_0 = \{x_0\}; \tag{1.3}$$

$$\mathcal{X}_1 = \left\{x = (x_1, x_2) \in \mathbb{R}^2 : a(x_1 - x_{11})^2 + b|x_2 - x_{12}| \leqslant c\right\},\ a, b, c > 0. \tag{1.4}$$

Требуется:

1. Написать в среде Matlab программу с пользовательским интерфейсом, которая по заданным значениям параметров $A, f, t_0, r, p, x_0, a, b, c, x_{11}, x_{12}$ определяет, разрешима ли задача (1.1). Если задача разрешима, программа должна (приближённо) найти значение T и построить графики компонент оптимального управления, оптимальной траектории, сопряжённых переменных. Кроме того, программа должна допускать возможность улучшения решения, как локальным, так и глобальным методами.

2. Для различных значений параметров (в том числе, для различных собственных значений матрицы A) провести анализ системы (1.1), численно решить задачу и построить соответствующие графики.

1.2 Формальная постановка задачи

1. Провести необходимые исследования системы (1.1) и привести сопутствующие теоретические выкладки;

2. Разработать и описать численный метод решения задачи и возникающих подзадач;

3. Реализовать на языке MATLAB программу, удовлетворяющую условиям из 1.1 и реализующую численный метод из пункта 2. Для этого, реализовать:

   - Пользовательский интерфейс ввода исходных данных;
   - Алгоритм поиска управлений и траекторий, подозрительных на оптимальные, а также алгоритм отбора из них оптимальных (при наличии таковых);
   - Алгоритм и интерфейс построения требуемых графиков;
   - Алгоритм локального и глобального улучшения решения;
   - Алгоритм сохранения и загрузки промежуточных данных, значений параметров и полученных ответов.

4. Построить, используя написанную программу, графики для различных значений параметров

2 Некоторые необходимые теоретические выкладки

2.1 Принцип максимума

Прежде всего, установим принцип максимума Понтрягина в следующей формулировке:[1]

Принцип Максимума Понтрягина (В формулировке из [2])
Пусть $(u^*(\cdot), x^*(\cdot))$ *— оптимальная пара.*
Тогда существует $\psi(t) \in AC[t_0, t_1], \psi(t) \neq 0 \ \forall t \in [t_0, t_1]$*:*

$$\dot{\psi} = -A^T\psi \tag{2.1}$$
$$\langle Bu^*(t), \psi(t) \rangle = \rho(\psi(t)|B\mathcal{P}) \tag{2.2}$$
$$\langle \psi(t_0), x^*(t_0) \rangle = \rho(\psi(t_0)|\mathcal{X}_0) \tag{2.3}$$
$$\langle -\psi(t_1), x^*(t_1) \rangle = \rho(-\psi(t_1)|\mathcal{X}_1) \tag{2.4}$$

Систему (2.1) называют *сопряжённой системой*, её решение $\psi = \psi(t)$ — *сопряжёнными переменными*, а условия (2.3) и (2.4) — *условиями трансверсальности*. Условие (2.2) позволяет выделить из всех возможных управлений семейство «подозрительных» на оптимальные.

2.2 Исследование сопряжённой системы

Для того, чтобы однозначно определить решение системы (2.1), нам необходимо присовокупить к ней некоторые начальные условия. Мы, для удобства решения, будем рассматривать $\psi(t_0) = \psi_0$. В результате получим задачу Коши для сопряжённой системы:

$$\begin{cases} \dot{\psi}(t) = -A^T\psi(t), \ t \in [t_0, t_1] \\ \psi(t_0) = \psi_0 \end{cases} \tag{2.5}$$

Тогда, решение этой системы представимо в виде:

$$\psi(t) = e^{-A^T(t-t_0)}\psi_0 \tag{2.6}$$

2.3 Выделение управлений и траекторий, «подозрительных» на оптимальные

Условие (2.2), в силу свойств скалярного произведения, можно переписать в следующем виде:

$$\langle B^T\psi(t), u^*(t) \rangle = \rho(B^T\psi(t)|\mathcal{P})$$

В свою очередь, раскрыв определение опорной функции множества $\mathcal{P}$ в направлении $B^T\psi(t)$, окончательно получим:

$$\langle B^T\psi(t), u^*(t) \rangle = \sup_{u(t)\in\mathcal{P}} \langle B^T\psi(t), u(t) \rangle \tag{2.7}$$

[1] Доказательство приведено, например, в [3]

Заметим, что множество $\mathcal{P}$ (см. (1.2)) есть эллипсоид $\mathcal{E}(p, P)$, где $P = \begin{pmatrix} \frac{r}{9} & 0 \\ 0 & \frac{r}{4} \end{pmatrix}$ — матрица конфигурации. Учтём также, что $B = E$; из [1] известно, что решение $u^*(t)$ уравнения (2.7) представимо в виде:

$$u^*(t) = p + \frac{P\psi(t)}{\sqrt{\langle \psi(t), P\psi(t) \rangle}} \tag{2.8}$$

Данное выражение корректно, так как $\psi(t) \neq 0$ для любого допустимого t, а $P \neq 0$. Кроме того, оно показывает, что для каждого ψ_0 существует единственное «подозрительное» на оптимальное управление.

С учётом того, что множество $\mathcal{X}_0$ состоит из одной точки (см. (1.3)) и первого условия трансверсальности (2.3), любая «подозрительная» на оптимальную траектория $x^*(t)$ выходит из точки x_0, т.е. $x^*(t_0) = x_0$. Тогда, подстановкой в (1.1) недостающих значений из (2.8), (2.6) и значения $x^*(t_0) = x_0$ окончательно получим систему:

$$\begin{cases} \dot{x}^*(t) = Ax^*(t) + u^*(t) + f, \; t \in [t_0, +\infty) \\ x^*(t_0) = x_0 \\ u^*(t) = p + \dfrac{P\psi(t)}{\sqrt{\langle \psi(t), P\psi(t) \rangle}} \\ \psi(t) = e^{-A^T(t-t_0)}\psi_0 \end{cases} \tag{2.9}$$

Решение данной системы $x^*(t)$ есть траектория, «подозрительная» на оптимальную, соответствующая управлению $u^*(t)$ из (2.8). Заметим, что и это управление $u^*(t)$, и соответствующая ему траектория $x^*(t)$ однозначно определяются (при фиксированных параметрах из 1.1) лишь значением $\psi(t_0) = \psi_0$.

2.4 Оценка погрешности

Поскольку численное решение задачи сопряжено с погрешностью, требуется ввести некоторую меру погрешности условия трансверсальности на правом конце (2.4): оно равносильно сонаправленности вектора $-\psi(t_1)$ и вектора внешней единичной нормали к границе множества $\mathcal{X}_1$ в точке $x^*(t_1)$. Множество $\mathcal{X}_1$ представляет собой область, заключённую внутри двух парабол; для удобства, перепишем (1.4) в следующем виде:

$$\mathcal{X}_1 = \left\{ x = (x_1, x_2) \in \mathbb{R}^2 : \frac{a}{b}(x_1 - x_{11})^2 - (\frac{c}{b} - x_{12}) \leqslant x_2 \leqslant x_{12} + \frac{c}{b} - \frac{a}{b}(x_1 - x_{11})^2 \right\}, \; a, b, c > 0.$$

Отсюда явно видно, что эти две параболы пересекаются в точках $\left(x_{11} \pm \sqrt{\frac{c}{a}},\, x_{12}\right)$

Уравнение касательной к верхней параболе в точке $\left(x_1^0, x_2^0\right)$[1]:

$$x_2 - x_2^0 = -\frac{2a}{b}(x_1^0 - x_{11})(x_1 - x_1^0)$$
$$\frac{2a}{b}(x_1^0 - x_{11})(x_1 - x_1^0) + (x_2 - x_2^0) = 0;$$

[1] Здесь и далее, по понятным причинам, предполагаем, что точка $\left(x_1^0, x_2^0\right)$ принадлежит границе множества $\mathcal{X}_1$

Отсюда, вектор нормали $\vec{n}_u$ и единичной нормали $\vec{\nu}_u$ к верхней параболе в точке $\left(x_1^0, x_2^0\right)$:

$$\vec{n}_u = \left(\frac{2a}{b}(x_1^0 - x_{11}),\ 1\right)$$
$$\vec{\nu}_u = \frac{\vec{n}}{\|n\|}$$

Аналогично, для нижней параболы имеем:

$$\vec{n}_l = \left(\frac{2a}{b}(x_1^0 - x_{11}),\ -1\right)$$
$$\vec{\nu}_l = \frac{\vec{n}}{\|n\|}$$

Таким образом, запишем общую формулу для внешней нормали к границе множества $\mathcal{X}_1$ в точке $\left(x_1^0, x_2^0\right)$, при условии, что $\left(x_1^0, x_2^0\right) \neq \left(x_{11} \pm \sqrt{\frac{c}{a}},\ x_{12}\right)$:

$$\vec{n} = \begin{cases} \left(\dfrac{2a}{b}(x_1^0 - x_{11}),\ 1\right) & \text{если } x_2^0 > x_{12} \\ \left(\dfrac{2a}{b}(x_1^0 - x_{11}),\ -1\right) & \text{если } x_2^0 < x_{12} \end{cases} \tag{2.10}$$

Отметим, что в точках пересечения парабол $\left(x_{11} \pm \sqrt{\frac{c}{a}},\ x_{12}\right)$ граница множества $\mathcal{X}_1$ не является гладкой, следовательно, существует целый сектор направлений, по которым выполняется условие (2.4).

В качестве меры погрешности в случае, если $x^*(t_1) = \left(x_{11} \pm \sqrt{\frac{c}{a}},\ x_{12}\right)$ возьмём модуль синуса угла между $-\psi(t_1)$ и вектором внешней нормали к границе (2.10), а если $x^*(t_1) = \left(x_{11} \pm \sqrt{\frac{c}{a}},\ x_{12}\right)$[1] — модуль синуса угла отклонения от соответстующего сектора «нормалей» (или 0, если $-\psi(t_1)$ лежит в этом секторе). Для нахождения модуля синуса угла между векторами воспользуемся основным тригонометрическим тождеством, и тем фактом, что для векторов $-\vec{\psi}(t_1)$ и $\vec{n}$ косинус угла φ между ними равен:

$$\cos\varphi = \frac{\langle -\vec{\psi}(t_1), \vec{n}\,\rangle}{\|-\psi(t_1)\|\|n\|},$$

откуда

$$\sin\varphi = \sqrt{1 - \left(\frac{\langle -\vec{\psi}(t_1), \vec{n}\,\rangle}{\|-\psi(t_1)\|\|n\|}\right)^2}.$$

[1] При решении в числах, это равенство равносильно тому, что точка $x^*(t_1)$ лежит в некоторой достаточно малой окрестности точки пересечения парабол

3 Описание численного метода

1. Задаём все параметры системы (загрузкой или вводом с клавиатуры), длину временного интервала T, инициализируем конфигурационную матрицу эллипса $\mathcal{P}$, соответствующие функции для множества $\mathcal{X}_1$, функцию попадания во множество $\mathcal{X}_1$;

2. Проверяем, не находится ли точка x_0 во множестве $\mathcal{X}_1$ изначально;

3. Для решения сопряжённой системы используется функция `solveConj` со следующей спецификацией:

```
function [ tMaj, tRet, xMaj, psi0Maj, alphaMaj, xSusp ] = solveConj( tMaj, ...
A, f, p, P, x0, t0, T, alphaSpace, opts )
```

4. В отдельной функции `solveConj` на единичной сфере перебираем всевозможные значения $\psi_0 = (\sin(\alpha),\ \cos(\alpha))^T$, рассматривая α на сетке `alphaSpace`. Для каждого значения ψ_0 решаем сопряженную систему при помощи функции `ode45`, реализующей метод Рунге-Кутты четвёртого порядка с параметром `events`, соответствующим функции попадания во множество $\mathcal{X}_1$;

5. В случае попадания во множество $\mathcal{X}_1$ за отведённое время, фиксируем данное время как оптимальное, если оно меньше предыдущего минимума (в начале алгоритма полагаем это время равным переданному в `solveConj` значению параметра `tMaj`, при первом запуске — $+\infty$), сохраняем соответствующие этому времени значения ψ_0, $x^*(t)$, α^*;

6. Таким образом, функция `solveConj` либо возвращает в качестве оптимального времени $+\infty$, что соответствует случаю отсутствия решения (на данной сетке), либо возвращает набор, состоящий из оптимального времени t^*, оптимальной траектории $x^*(t)$, оптимального начального условия для сопряжённой системы ψ_0 и соответствующего ему угла α^*, а также коллекцию тестовых («подозрительных») траекторий `xSusp`; по этим данным однозначно восстанавливается оптимальное управление $u^*(t)$ и значение сопряжённых переменных $\psi(t_1)$;

7. Ошибка второго условия трансверсальности рассчитывается согласно пункту 2.4 в функции `calcError`;

8. Производится построение графиков согласно требованиям;

9. При необходимости, повторно вызвав функцию `solveConj` с параметром `tMaj` равным оптимальному времени t^*, полученному на предыдущем шаге, можно осуществить локальное или глобальное улучшение решения. Для локального улучшения подаём на вход функции (параметр `alphaSpace`) разбиение некоторой окрестности α^*, а для глобального — сетку большего размера

4 Примеры

4.1 Локальное и глобальное улучшение

$A = \begin{bmatrix} -0.2 & 0.2 \\ 0.8 & 0.4 \end{bmatrix}$, $f = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $a = 1,\ b = 1,\ c = 2,\ x_{11} = 2,\ x_{12} = 4,\ x_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $r = 2,$
$p = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}$, $t_0 = 0,\ T = 2,\ \texttt{gridsize} = 50$
Собственные значения матрицы A: $\lambda_1 = -0.4,\ \lambda_2 = 0.6$
Результат работы программы:

```
Optimal time: 1.7809
Error: 0.17255
```

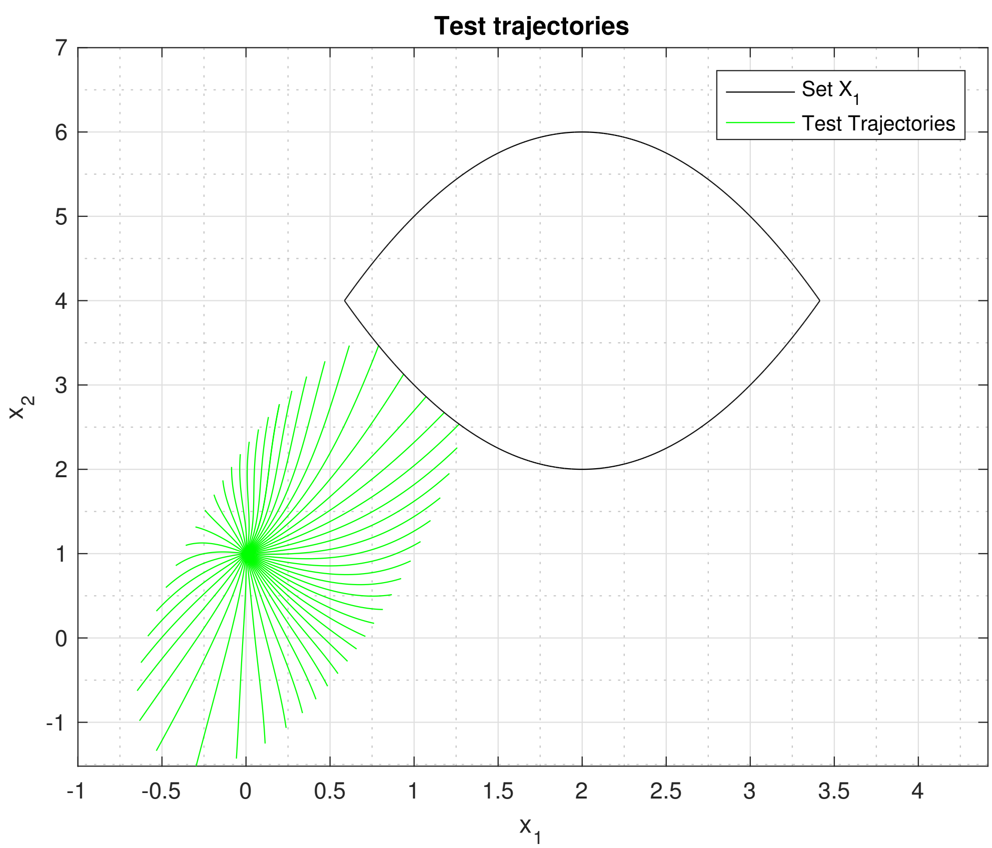


Рис. 1: Тестовые траектории

Таким образом, до улучшения $t_1^* = 1.7809$, ошибка имеет порядок $1.7 \cdot 10^{-1}$.
Проведём локальное улучшение: в окрестности угла α^* построим сетку вдвое меньшего размера $\texttt{gridsize}/2$.
Новый результат работы программы:

```
Improved optimal time: 1.7747
Error: 0.021635
```

После локального улучшения, t_2^* изменилось на величину порядка $5 \cdot 10^{-3}$, а вот ошибка уменьшилась на порядок, до величины $2 \cdot 10^{-2}$.

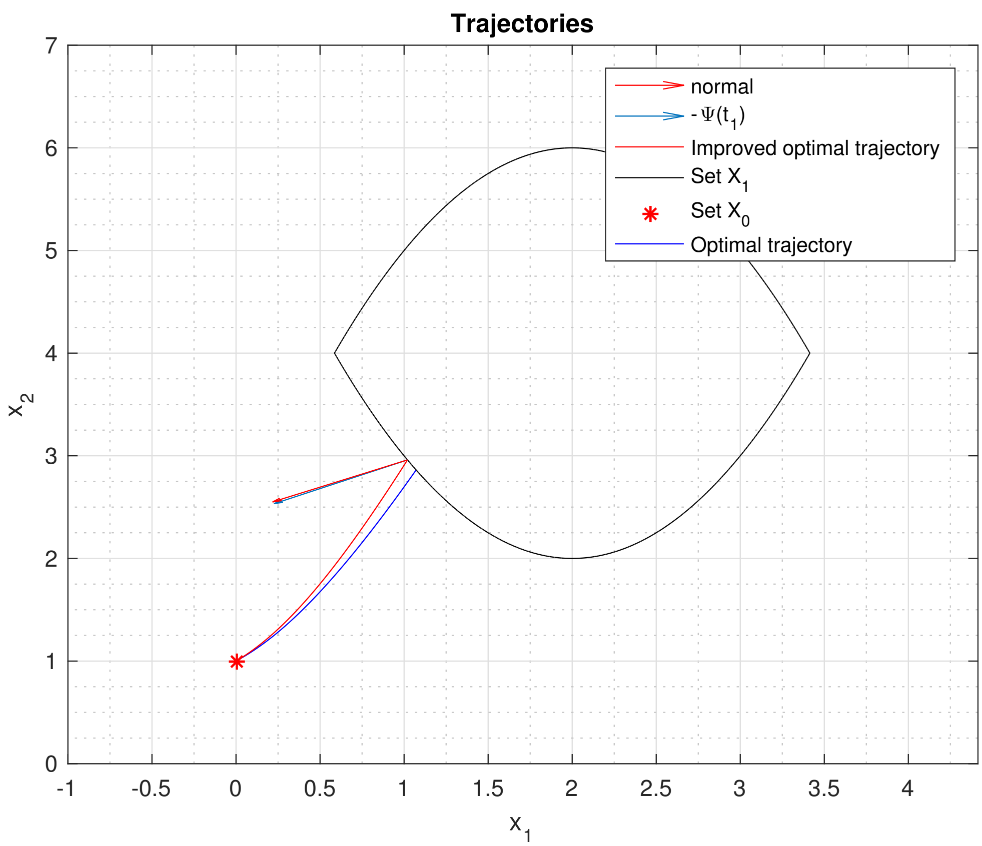


Рис. 2: Траектории после локального улучшения

Теперь, проведём глобальное улучшение независимо от локального: запустим программу на вдвое большей сетке, но с мажорантой $t_1^* = 1.7809$, полученной про первоначальном запуске. Результат работы:

```
Improved optimal time: 1.7766
Error: 0.11483
```

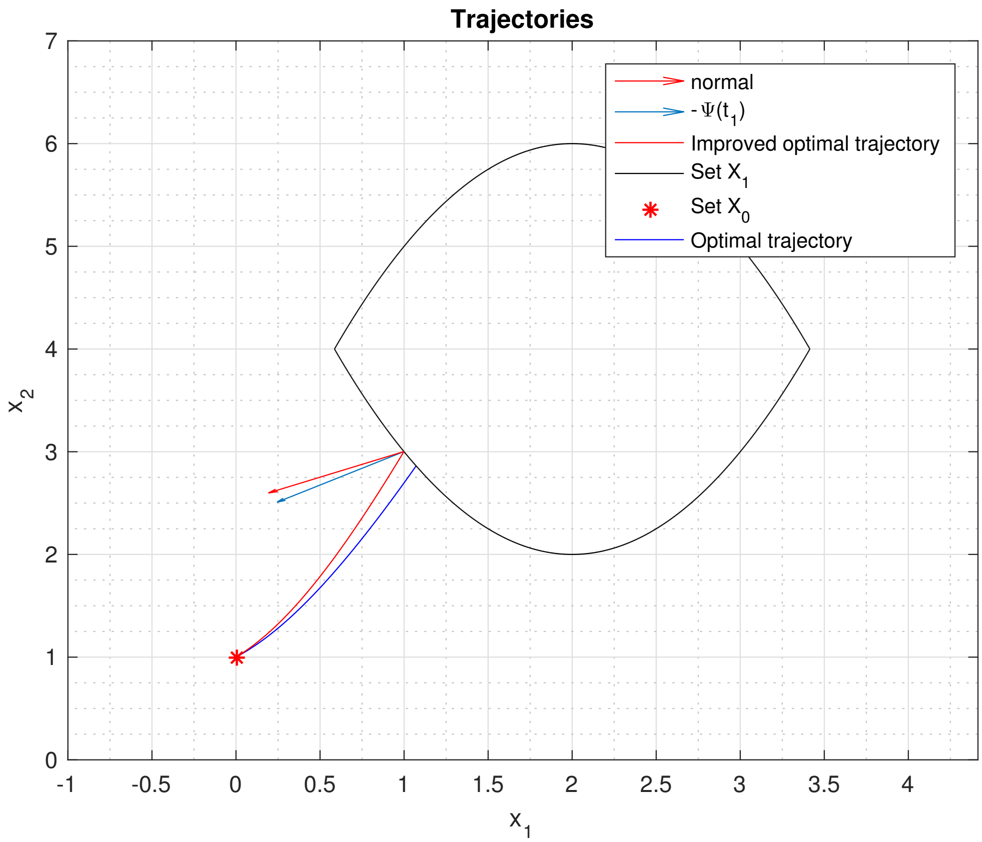


Рис. 3: Траектории после глобального улучшения

В данном случае, локальное улучшение эффективнее, чем глобальное, так как даёт меньшее время и меньшую ошибку.

4.2 Неоднородная система

$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $f = \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix}$, $a = 1$, $b = 1$, $c = 1$, $x_{11} = -1.5$, $x_{12} = 2$, $x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $r = 3$,
$p = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $t_0 = 0$, $T = 1$, $\texttt{gridsize} = 72$

Собственные значения матрицы A: $\lambda_{1,2} = 1$

Результат работы программы:

```
Optimal time: 0.78886
Error: 0.054337
```

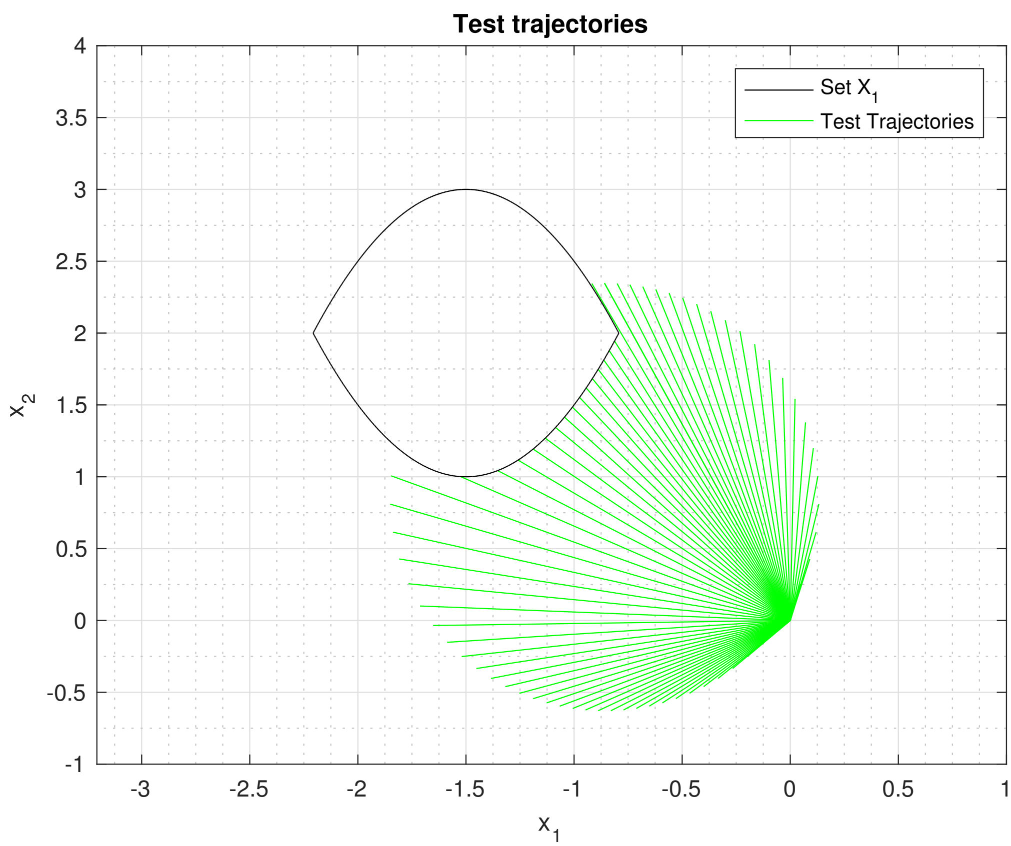


Рис. 4: Тестовые траектории

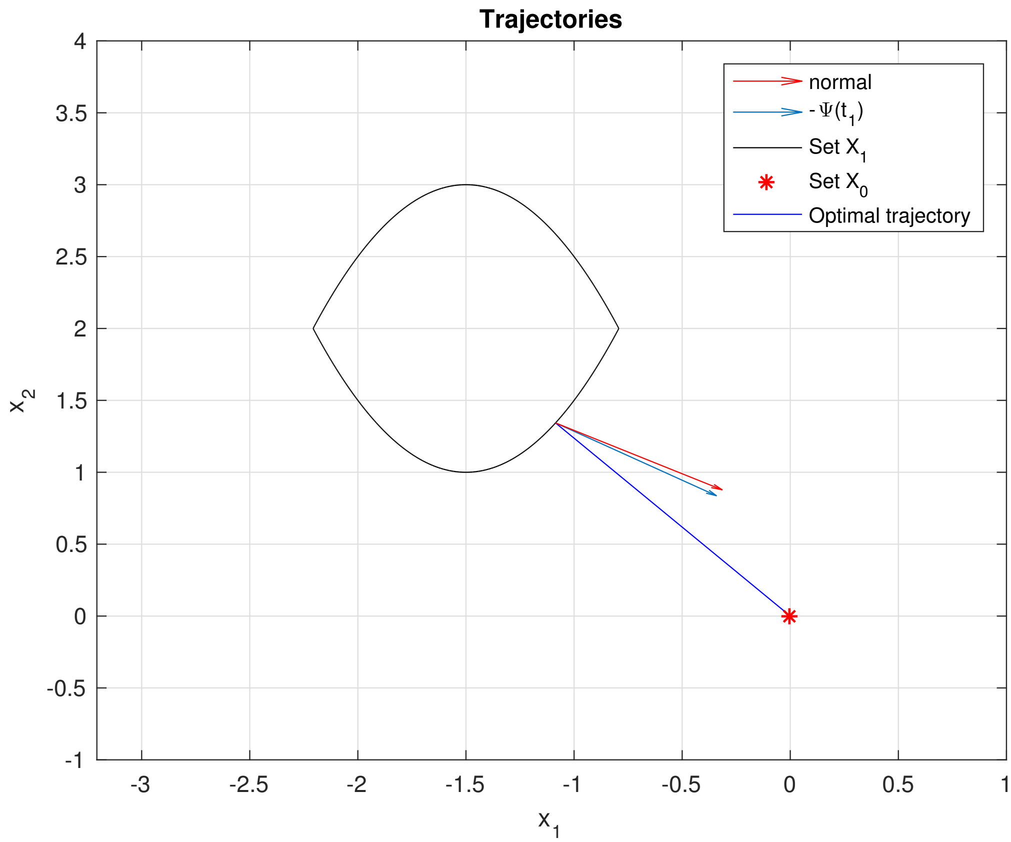
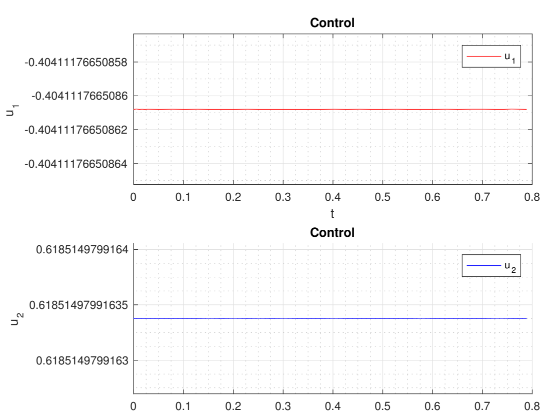
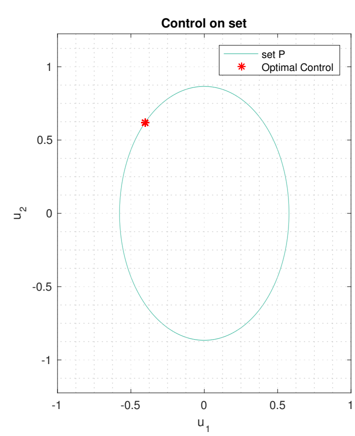


Рис. 5: Оптимальная траектория



(a) Компоненты оптимального управления



(b) Оптимальное управление на множестве $\mathcal{P}$

Рис. 6: Оптимальное управление

4.3 Разрыв по времени

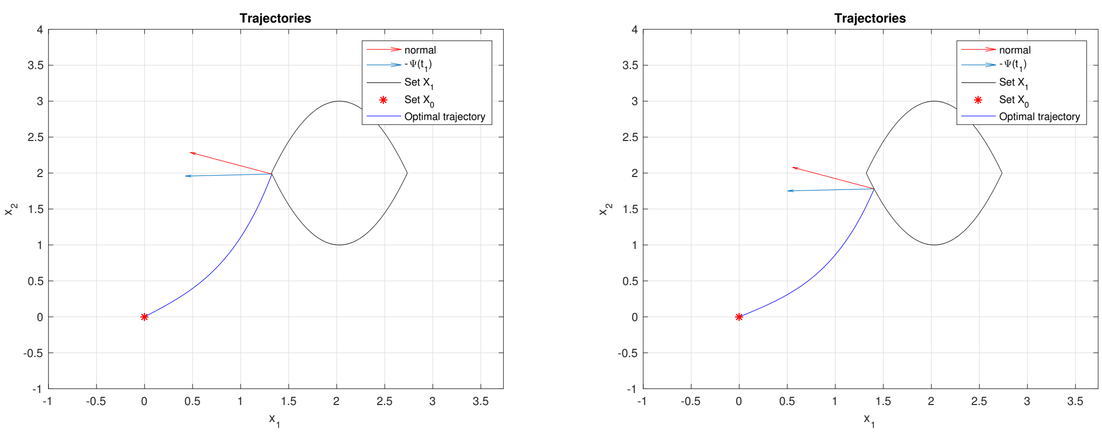
$A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, $f = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $a = 2,\ b = 1,\ c = 1,\ x_{11} = 2.028,\ x_{12} = 2,\ x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $r = 3$,
$p = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $t_0 = 0,\ T = 3,\ \texttt{gridsize} = 100$
Собственные значения матрицы A: $\lambda_1 = 0,\ \lambda_2 = 1$
Результат работы программы:

```
Optimal time: 2.3461
Error: 0
```

Теперь, сдвинем множество $\mathcal{X}_1$: положим $x_{11} = 2.029$. Снова запустим программу:

```
Optimal time: 2.4638
Error: 0.35233
```

Как можно заметить, при сдвиге множества на величину порядка 10^{-3}, оптимальное время изменилось на величину большего порядка (10^{-1}). Кроме того, возросла и ошибка. Приведём, в заключение, графики оптимальных траекторий в обоих случаях:



(a) $x_{11} = 2.028$

(b) $x_{11} = 2.029$

Рис. 7: Оптимальная траектория

В первом случае, траектория попадает в точку границы множества $\mathcal{X}_1$, в которой нарушается гладкость; при этом, вектор сопряженных переменных $-\vec{\psi}(t_1)$ попадает в сектор нормалей, вследствие чего удовлетворяет условию трансверсальности (2.4).

4.4 Система с вырожденной матрицей

$A = \begin{bmatrix} -1 & -1 \\ 0 & 0 \end{bmatrix}$, $f = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $a = 2,\ b = 1,\ c = 1, : x_{11} = -1.5,\ x_{12} = 2,\ x_0 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$, $r = 5$,
$p = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $t_0 = 1,\ T = 2.5,\ \texttt{gridsize} = 70$

Собственные значения матрицы A: $\lambda_1 = -1,\ \lambda_2 = 0$

Запустим программу и проведём одно локальное улучшение. Результат работы программы:

```
Optimal time: 2.2033
Error: 0.029728
Improved optimal time: 2.2029
Error: 0.00038871
```

Отметим, что выдаваемое программой время это конечное значение времени t_1, а не величина $t_1 - t_0$, характеризующая длительность перехода. В предыдущих примерах, так как $t_0 = 0$, эти две величины совпадали.

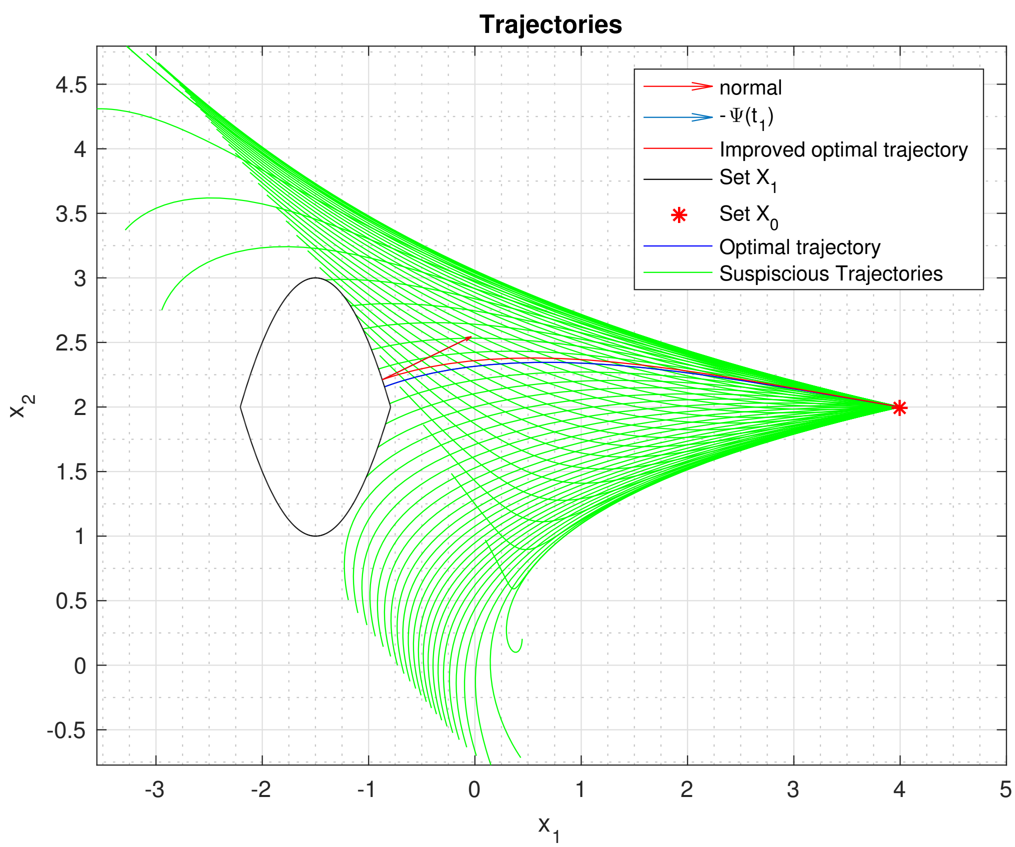


Рис. 8: Траектории системы (тестовые, оптимальная и улучшенная оптимальная)

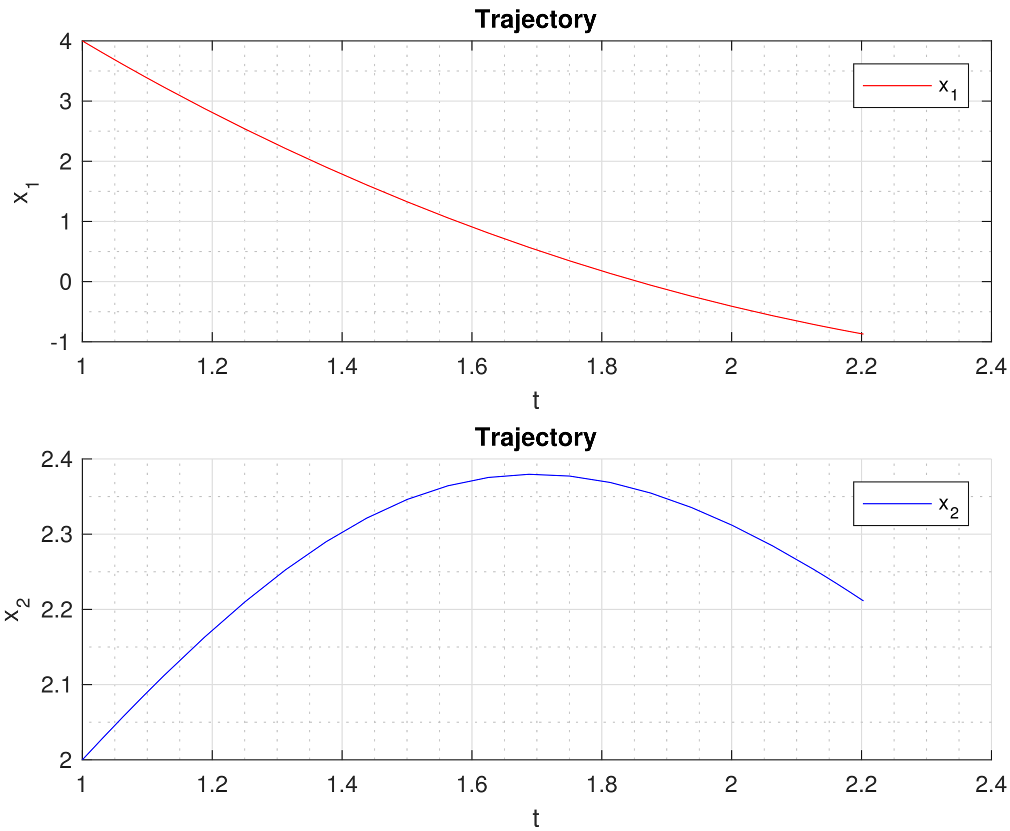
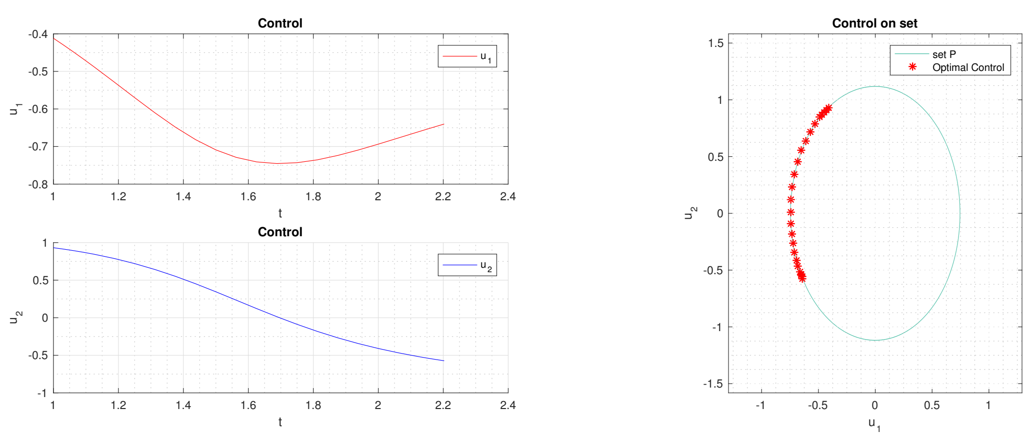


Рис. 9: Компоненты улучшенной оптимальной траектории



(a) Компоненты оптимального управления (b) Оптимальное управление на множестве $\mathcal{P}$

Рис. 10: Оптимальное управление

Список литературы

[1] И. В. Рублёв. *Лекционный курс Оптимальное Управление (Линейные Системы)*, кафедра Системного Анализа, Факультет Вычислительной Математики и Кибернетики, МГУ им. М. В. Ломоносова, 2017

[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`