Таллиннский технический университет

Факультет инфотехнологии

Институт автоматики

Вячеслав Овдий

Метод фазовой плоскости при анализе динамических систем в MATLAB

Дипломная работа бакалавра

Руководитель: Алексей Тепляков

Институт автоматики,

Таллиннский технический университет

Таллинн 2014

Декларация: настоящим заявляю, что данная бакалаврская работа является результатом моей самостоятельной работы, представлена в Таллиннский технический университет для получения степени бакалавра, и ранее не была использована для соискания академической степени.

Deklareerin, et käesolev bakalaureusetöö, mis on minu iseseisva töö tulemus, on esitatud Tallinna Tehnikaülikooli bakalaureusekraadi taotlemiseks ja selle alusel ei ole varem taotletud akadeemilist kraadi.

Вячеслав Овдий

Дата .....

Подпись .....

Оглавление

Abstract

Phase plane analysis of dynamical systems in MATLAB

The purpose of this work is to describe PPLANE tool and its possibilities for use in nonlinear control theory. PPLANE is a tool for MATLAB, which allows to plot trajectories and phase portraits of autonomous second-order systems. Tool can be used in a research of different models and has many useful features such as plotting given solutions versus the independent variable and finding equilibrium points of the system, as well as displaying linearization around equilibrium points.

The main objectives of this bachelor's thesis are to review theoretical information, that can be used in analysis of the autonomous second order systems with phase plane method, and then to use this information for practical purposes (by resolving some examples from different courses).

Also author will compare two methods of analysis: by using standard MATLAB functions versus using PPLANE, and then conclusion will be drawn: which method is more intuitive and qualitative in terms of received information.

Kokkuvõte

Faasidiagrammil põhinev dünaamiliste süsteemide analüüs MATLAB keskkonnas

Käesoleva töö eesmärk on kirjeldada PPLANE tööriista ja selle kasutamise võimalusi mittelineaarsete juhtimissüsteemide analüüsimiseks. PPLANE on MATLABi vahend, mis võimaldab joonistada trajektooride graafikuid ja teist järku autonoomsete süsteemide faasportreesid. Tööriista saab kasutada erinevate mudelite uurimise jaoks, kuna sellel on sellised kasulikud funktsioonid nagu esitatud lahenduste versus sõltumatu muutuja joonestamine, süsteemi tasakaalu punkti leidmine, samuti linearisatsioon taasakaalupunktide ümber.

Bakalaureusetöö peamised eesmärgid on teoreetilise info töötlemine, et uurida teist järku autonoomsete süsteemide analüüsimise võimalusi faasi lennuki meetodiga ja seejärel kasutada saadud teavet konkreetsete praktiliste ülesannete sooritamiseks.

Töös võrreldakse kahte erinevat graafikute joonestamise meetodit: kasutades MATLABi standardseid funktsioonide ja PPLANE vahendit. Autor teeb järelduse, milline meetod on intuitiivsem ja annab paremaid tulemusi.

Резюме

Анализ динамических систем методом фазовой плоскости в MATLAB

Цель данной работы состоит в описании инструментария PPLANE и его возможностей для использования при анализе нелинейных систем. PPLANE — программа для MATLAB, позволяющая строить графики траекторий и фазовые портреты автономных систем второго порядка. Инструментарий, который может быть использован при исследовании совершенно различных моделей, имеет множество полезных функций: например, построение графиков зависимостей переменных и нахождение особых точек системы, а также линеаризация в окрестностях данных точек.

Основные задачи работы заключаются в рассмотрении теоретических вопросов по анализу автономных систем второго порядка с помощью метода фазовой плоскости, а затем применение полученной из теории информации на практике путём решения конкретных примеров из различных курсов. Также в ходе работы будет сравнены два метода построения графиков: с помощью стандартных функций MATLAB и с помощью PPLANE, а затем сделан вывод, какой из методов является более интуитивный и качественный в плане полученной информации.

Глава 1

Введение

В современном мире развитие технологий является одним из приоритетных задач всего человечества. Наряду с постоянным увеличением количества всевозможных систем создаются всё более изощрённые методы управления, позволяющие обеспечить быструю и качественную работу системы. Недостаточно лишь создать механизм: порой рассчитать наиболее оптимальную модель управления для данного устройства оказывается куда сложнее; и поэтому исследования в области анализа систем так важны в наши дни.

Основной целью данной работы является исследование методов анализа автономных систем второго порядка, так как поведение многих реальных систем в конкретный момент времени возможно представить в виде кривых на плоскости. Это позволяет наглядно продемонстрировать и облегчить понимание процессов, проходящих в данной системе.

В ходе работы будут рассмотрены возможности программы *PPLANE* — инструментария для построения траекторий и фазовых портретов автономных систем второго порядка. В возможности программы также входит изучение основных аспектов нелинейных систем второго порядка, с помощью которых возможно исследование поведения системы около точек равновесия (особых точек) и анализ систем с помощью метода фазовой плоскости.

PPLANE была создана профессором Джоном Полкингом, преподавателем университета Уильяма Марша Райса, почти 20 лет назад, но не утратила своей актуальности и по сей день. Данная работа является руководством к использованию *PPLANE*, в котором я постараюсь осветить все её возможно-

сти, все плюсы и минусы, а также разобрать на её основе конкретные примеры некоторых известных систем и популярные методы управления.

Структура работы

В Главе 2 приводится теоретическая информация о явлениях и терминах, которые возможно анализировать с помощью PPLANE8.

В Главе 3 осуществляется разбор самой программы для наиболее интуитивного понимания функциональности и возможностей программы.

В Главе 4 будут рассмотрены конкретные примеры с помощью теоретических знаний, полученных в Главе 2. Опытным путём будут проведены исследования работоспособности инструментария, и в заключении сделаны выводы о его достоинствах и недостатках.

Глава 2

Теоретическая часть

2.1 Анализ системы с помощью метода фазовой плоскости

Начать анализ автономной системы второго порядка возможно при наличии уравнений системы управления в нормальной форме: таким образом вектор состояния системы однозначно определяет её состояние. В пространстве состояний каждому состоянию системы будет соответствовать точка, называющаяся *изображающей точкой* [1]. Если состояние системы начинает изменяться, то изображающая точка описывает траекторию, называемую *фазовой траекторией*. Множество фазовых траекторий будет называться *фазовым портретом* . Фазовые траектории и фазовый портрет возможно проиллюстрировать с помощью двухмерного фазового пространства — *фазовой плоскости* — координатной плоскости, имеющей по осям координат две переменные (*фазовые координаты*), которые однозначно определяют состояние системы второго порядка. PPLANE же в свою очередь является отличным инструментарием для построения и анализирования фазовых портретов функций второго порядка с помощью *метода фазовой плоскости*.

Рассмотрим понятие автономной системы второго порядка [2], которая представляет собой систему дифференциальных уравнений вида

$$\begin{cases} \dot{x} &= f(t,x,y) \\ \dot{y} &= g(t,x,y) \end{cases}. \tag{2.1}$$

Переменная t в системе (2.1) обычно представляет собой время и называется *независимой переменной*. Чаще всего правые стороны дифференциальных уравнений не содержат в явном виде независимой переменной t, и могут быть записаны в виде

$$\begin{cases} \dot{x} &= f(x,y) \\ \dot{y} &= g(x,y) \end{cases}. \tag{2.2}$$

Такая система называется *автономной*.

Пример 2.1.1 В качестве примера простой автономной системы можно привести следующую систему уравнений (2.3)

$$\begin{cases} \dot{x} &= y \\ \dot{y} &= -x \end{cases}. \tag{2.3}$$

С помощью подстановки каждый может проверить и убедиться, что пара функций $x(t) = \cos(t)$ и $y(t) = -\sin(t)$ удовлетворяют обоим уравнениям, и, следовательно, являются решением.

Существует несколько способов представления функций в графическом виде. С помощью следующих команд мы можем построить в *MATLAB* графики зависимостей переменных x и y от t.

```
>> t = linspace(0,4*pi);
>> x = cos(t); y = -sin(t);
>> subplot(2,1,1)
>> plot(t,x)
>> title('x = cos t'), ylabel('x')
>> subplot(2,1,2)
>> plot(t,y)
>> title('y = -sin t')
>> xlabel('t'), ylabel('y')
```

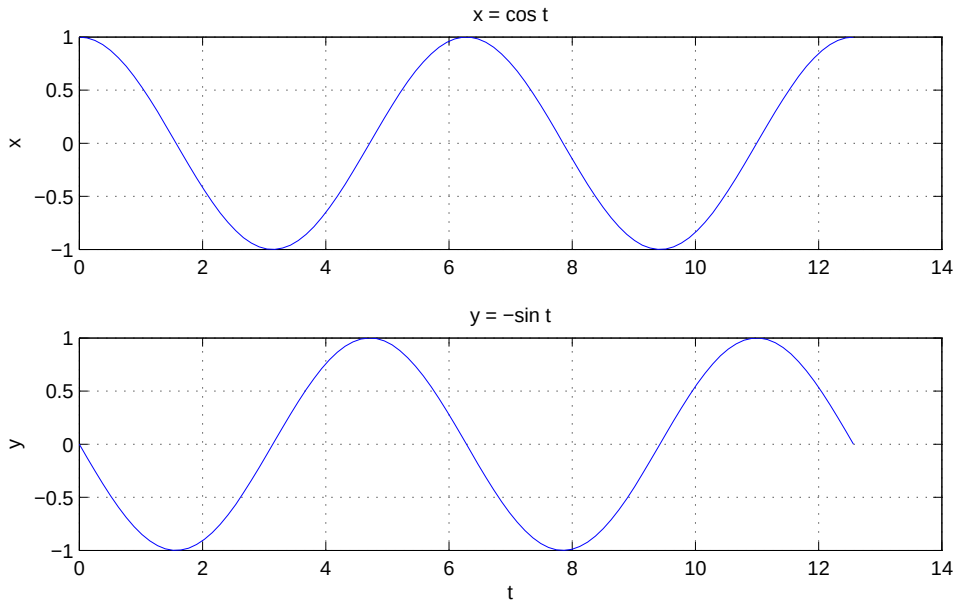

Рис. 2.1: Графики зависимости $x(t)$ и $y(t)$

Также возможно построить график зависимости функций x от y, называемый *фазовым портретом*, с помощью данных команд

```
>> plot(x,y)
>> axis equal
>> xlabel('x'), ylabel('y')
```

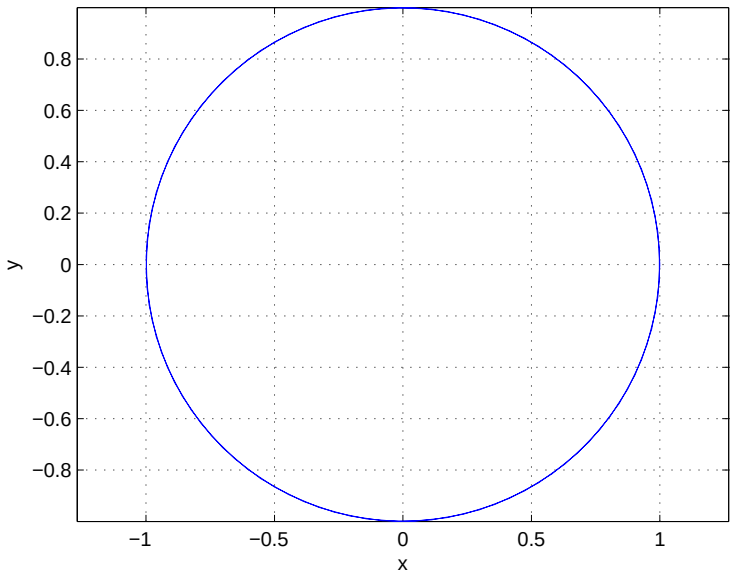

Рис. 2.2: Фазовый портрет функций x и y

Предположим, что $x = [x_1, x_2]^T$, тогда $\dot{x} = [\dot{x}_1, \dot{x}_2]^T$, и векторное уравнение принимает вид (2.4)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} f(t,x) \\ g(t,x) \end{bmatrix}. \tag{2.4}$$

Наконец если обозначить, что $F(t,x) = [f(t,x), g(t,x)]^T$, тогда уравнение (2.4) может быть записано как

$$\dot{x} = F(t,x). \tag{2.5}$$

Для плоской системы вида $\dot{x} = F(t,x)$ решением является вектор-функция $x(t) = [x_1(t), x_2(t)]^T$. Зависимости двух функций возможно представить как в виде графика (2.1), или изобразить её на фазовой плоскости (2.2). Так как решением дифференциального уравнения $\dot{x} = F(t,x)$ является функция $x(t)$, то в каждой точке $(t, x(t))$ кривая $t \to x(t)$, должна иметь касательный вектор $F(t, x(t))$.

Если значение переменной t не изменяется, то вектор $F(t,x)$, находящийся на точке x, и представляющий из себя совокупность всех возможных касательных векторов к кривым для выбранного значения t. Естественно, что векторное поле изменяется при изменении переменной t, и анализ подобной визуализации процесса может быть затруднён. Но если система является автономной, то уравнение (2.5) может быть записано в виде

$$\dot{x} = F(x), \tag{2.6}$$

где векторное поле не изменяется со временем t и, соответственно, не зависит от t. Таким образом векторное поле содержит совокупность всех возможных касательных к кривым для любых значений t.

Рассмотрим фазовые портреты динамических систем более детально. Одним из важных понятий для анализа фазового портрета является особая точка системы — точка равновесия или покоя динамической системы. Следует отметить, что фазовые портреты нелинейных систем куда более разнообразны, чем в случае линейных систем, но типы особых точек обеих систем совпадают. Имеются в виду те особые точки, в окрестностях которых уравнения нелинейных систем допускают *линеаризацию*.

Линеаразиция — один из наиболее распространенных методов анализа нелинейных систем [3]. Идея линеаризации — использование линейной системы для аппроксимации поведения решений нелинейной системы в окрестности точки равновесия. Линеаризация позволяет выявить большинство качественных и особенно количественных свойств нелинейной системы.

Методы линеаризации имеют ограниченный характер, то есть эквивалентность исходной нелинейной системы и ее линейного приближения сохраняется лишь для ограниченных пространственных или временных масштабов системы, или для определенных процессов, причем, если система переходит из одного режима работы в другой, то следует изменить и ее линеаризованную модель.

2.2 Фазовые портреты и их свойства

Предположим, что имеем дело с линейной стационарной системой

$$\dot{x} = Ax, \tag{2.7}$$

где A — матрица 2x2 [4]. Решением для (2.7) в начальный момент времени x_0 является

$$x(t) = M\,\mathrm{e}^{J_r t}\,M^{-1}\,x_0, \tag{2.8}$$

где J_r — матрица Якоби (якобиана) матрицы A, а M — обратная матрица, такая, что $M^{-1}\,A\,M = J_r$. В зависимости от собственных значений матрицы A, якобиана может иметь 3 вида, в которых k принимает значения 0 или 1:

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \begin{bmatrix} \lambda & k \\ 0 & \lambda \end{bmatrix} \text{ или } \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}. \tag{2.9}$$

Первый вид имеет место, когда собственные значения λ_1 и λ_2 являются вещественными и не равны между собой. Второй — если собственные значения являются вещественными и равны. Если же собственные значения имеют

мнимую часть ($\lambda_{1,2}= \alpha \pm j\beta$), то они относятся к третьему случаю. Для более точного анализа стоит разделять все типы особых точек.

В случае, если оба корня характеристического уравнения вещественные и положительные ($\lambda_{1,2} > 0$), то имеем дело с *неустойчивым узлом*, а при вещественных отрицательных корнях ($\lambda_{1,2} < 0$) с *устойчивым узлом*. Для обоих случаев характерен один тип фазовых траекторий — *параболы* (Рис. 2.3).

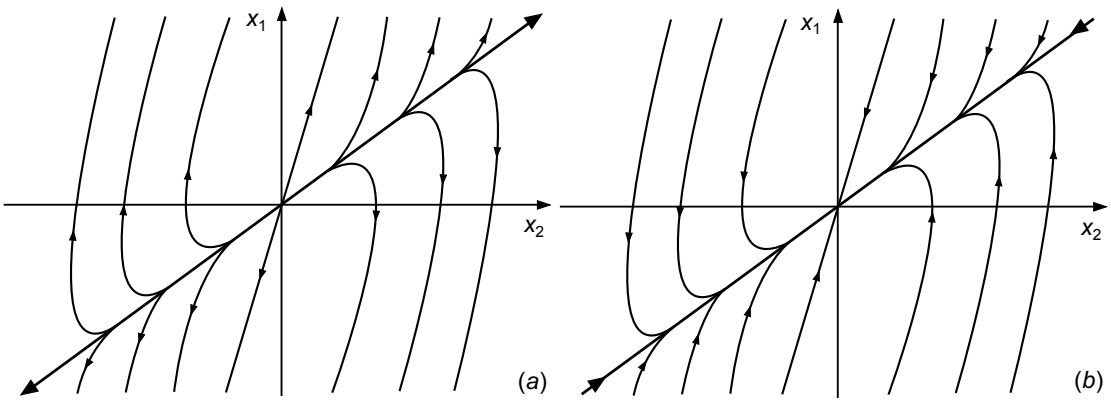

Рис. 2.3: Узлы: неустойчивый (a) и устойчивый (b)

Если же оба корня характеристического уравнения являются вещественными, но отличаются друг от друга знаками, то особая точка будет называться *седлом*. Фазовые траектории строятся на основании *гипербол* (Рис. 2.4).

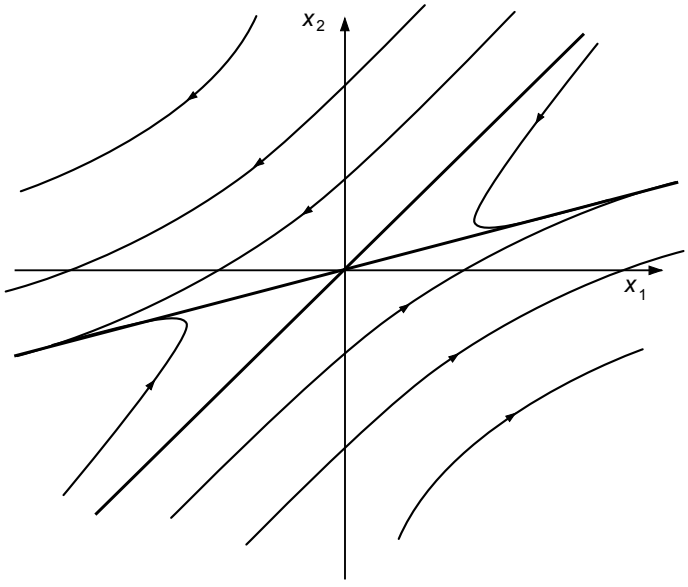

Рис. 2.4: Особая точка типа седло

Проанализируем случай, если корнями характеристического уравнения являются комплексные числа. Ниже приведены фазовые траектории для случаев, когда $\lambda_{1,2}= \alpha \pm j\beta$ (Рис. 2.5).

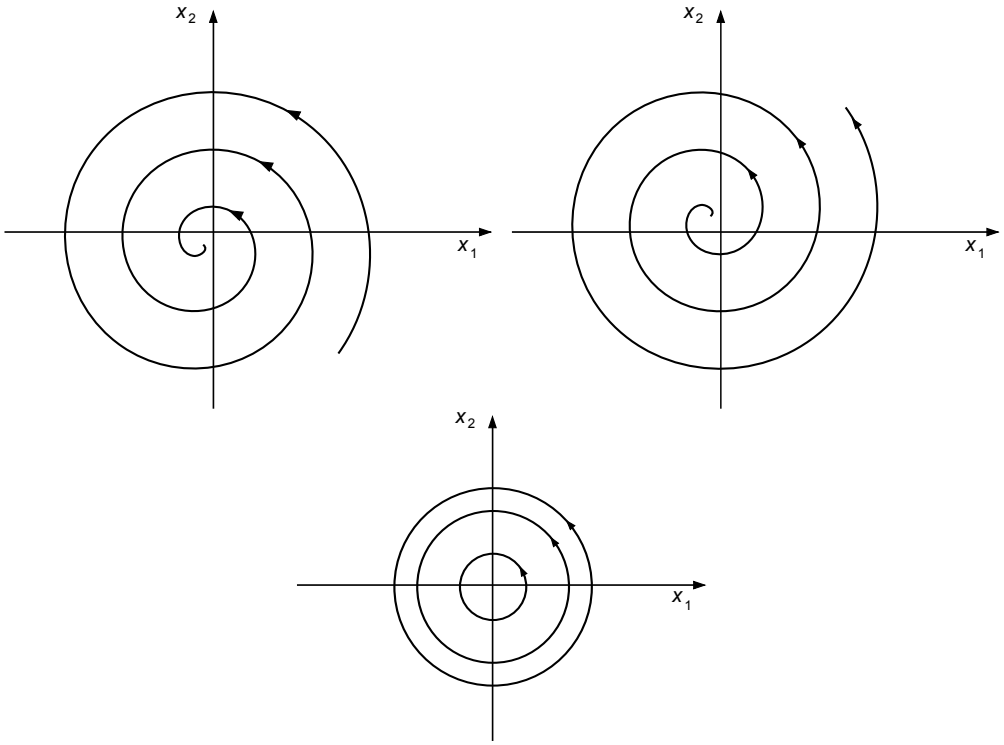

Рис. 2.5: Траектории в случае $\lambda_{1,2}= \alpha \pm j\beta$

Фазовые портреты же выглядят так (Рис. 2.6):

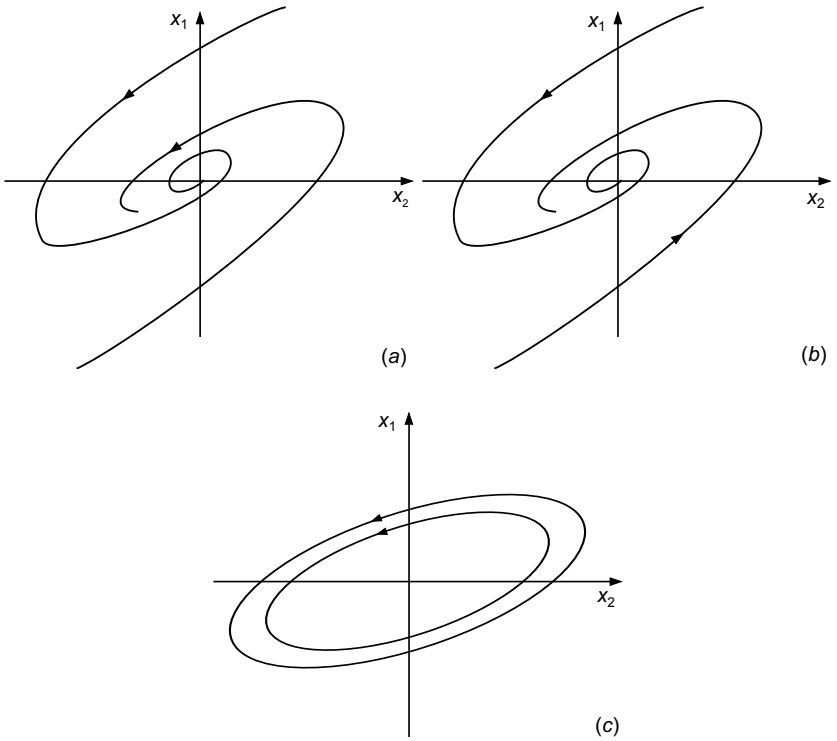

Рис. 2.6: Фазовые портреты в случаях: (a) устойчивый фокус, (b) неустойчивый фокус, (c) центр

Отдельно следует рассмотреть ситуации, когда характеристическое уравнение имеет один нулевой корень кратности 2, то есть $\lambda_1 = \lambda_2 = \lambda \neq 0$. В

данном случае положение равновесия называется *дикритическим узлом*, направление фазовых траекторий которого зависит от знака λ (Рис. 2.7).

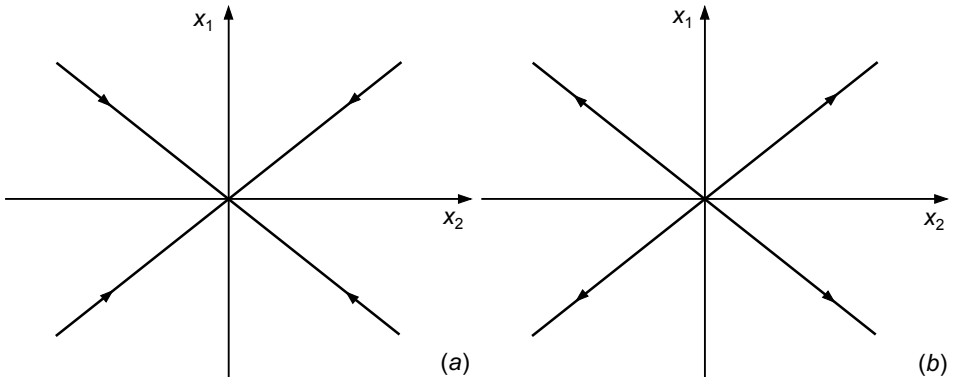

Рис. 2.7: Дикритические узлы: (a) устойчивый $\lambda_1 = \lambda_2 = \lambda < 0$, (b) неустойчивый $\lambda_1 = \lambda_2 = \lambda > 0$

В случае, если нулевой корень кратен 1 при сохранении равенства $\lambda_1 = \lambda_2 = \lambda \neq 0$, имеем дело с *вырожденными узлами*, направление траекторий которых также зависит от знака. Фазовые портреты вырожденных узлов изображены на Рис. 2.8.

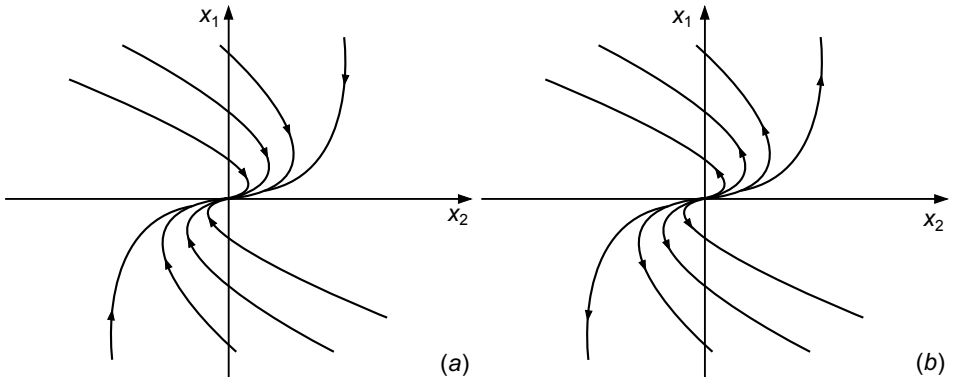

Рис. 2.8: Вырожденные узлы: (a) устойчивый $\lambda_1 = \lambda_2 = \lambda < 0$, (b) неустойчивый $\lambda_1 = \lambda_2 = \lambda > 0$

В редких случаях типом особой точки является *вырожденная матрица*, когда одно или оба собственных значений равны нулю $\lambda_1 = \lambda_2 = \lambda \neq 0$. В зависимости от знака λ движение при $t \to \infty$ происходит либо в направлении прямой V_1, либо от нее (Рис. 2.9).

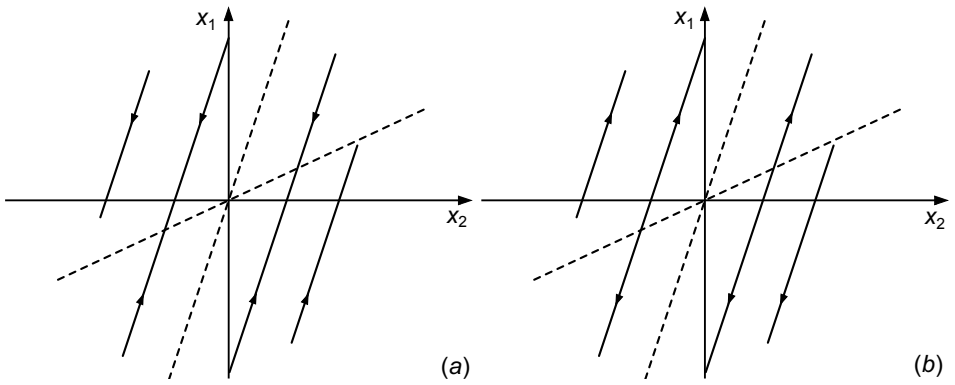

Рис. 2.9: Фазовые портреты вырожденных матриц: (a) $\lambda_1 = 0, \lambda_2 < 0$, (b) $\lambda_1 = 0, \lambda_2 > 0$

И последний рассматриваемый тип особой точки при $\lambda_1 = \lambda_2 = 0$ может иметь только 1 собственный вектор (матрица A — ненулевая). Все точки прямой, направленной вдоль собственного вектора V_1 являются неустойчивыми положениями равновесия, а фазовые траектории представляют из себя прямые, параллельные V_1 (Рис. 2.10).

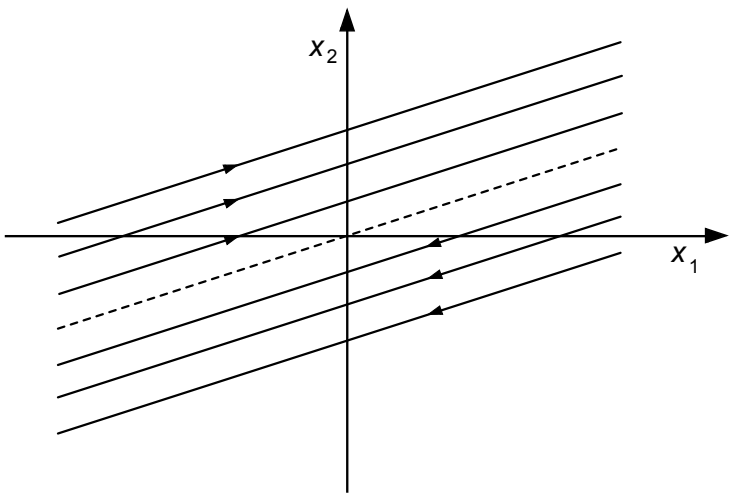

Рис. 2.10: Особая точка типа седло

Таким образом, определив тип особой точки с помощью фазового портрета системы, имеет смысл перейти к анализу конкретных систем в практической части работы. Стоит отметить, что программа MATLAB значительно упрощает процесс визуализации при анализе линейных и нелинейных систем. В следующей подглаве будет в деталях описан инструментарий для построения и исследования динамических систем второго порядка — PPLANE8.

Глава 3

Работа с инструментарием PPLANE8

3.1 Возможности PPLANE8

В качестве объекта исследования была выбрана последняя на текущий момент версия PPLANE8 для MATLAB, функционал которой практически идентичен Java-версии инструментария, представленной на сайте разработчика.

Для запуска инструментария в главном окне MATLAB следует прописать команду

>> **pplane8**

Загрузка инструментария может занять некоторое время в зависимости от конфигурации компьютера, после чего появится окно для ввода желаемой функции и её параметров. Рассмотрим его подробнее (Рис. 3.1).

3.1.1 Ввод уравнений и параметров

Окно содержит несколько полей для ввода системы и её параметров, а так же функций настройки отображения будущего графика. Меню данного окна будут рассмотрены немного позднее, а пока можно заметить, что в верхней части находятся поля для ввода функций автономной системы второго порядка, которая полностью соответствует формуле (2.2). По умолчанию в PPLANE уже введена стандартная система, но изменить её можно в любой момент,

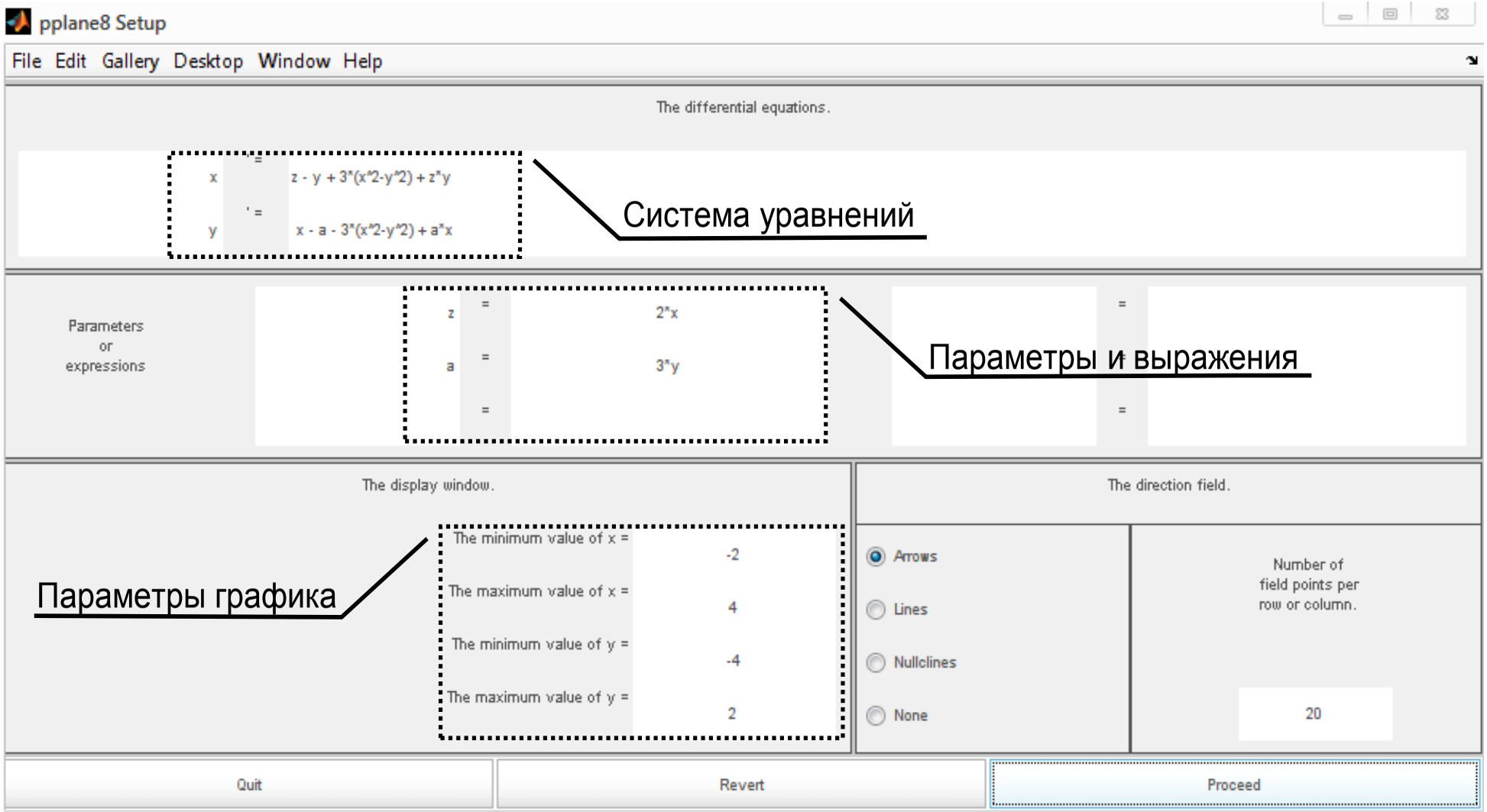

Рис. 3.1: Окно настройки параметров функций и графика

причём нет необходимости придерживаться предложенных переменных x и y — выбрать можно любые. В некоторых случаях при использовании нестандартных переменных программа может подсвечивать поля красным фоном, сообщая о ошибке, но их следует игнорировать. Ниже, в средней секции, находятся поля для ввода параметров и выражений, которые можно использовать в случае, если в системе имеются часто повторяющиеся выражения. Это может оказаться полезным так же в случае, когда параметры системы требуется периодически изменять. В нижней части окна возможно задать границы будущего графика (по x и y), а также вид отображения фазовых траекторий. Имеются также 3 кнопки: выход из программы, отмена изменений системы и, собственно, кнопка для построения графика. Нажатием на последнюю откроется окно визуализации для анализа заданной системы, изображенное на Рис. 3.2.

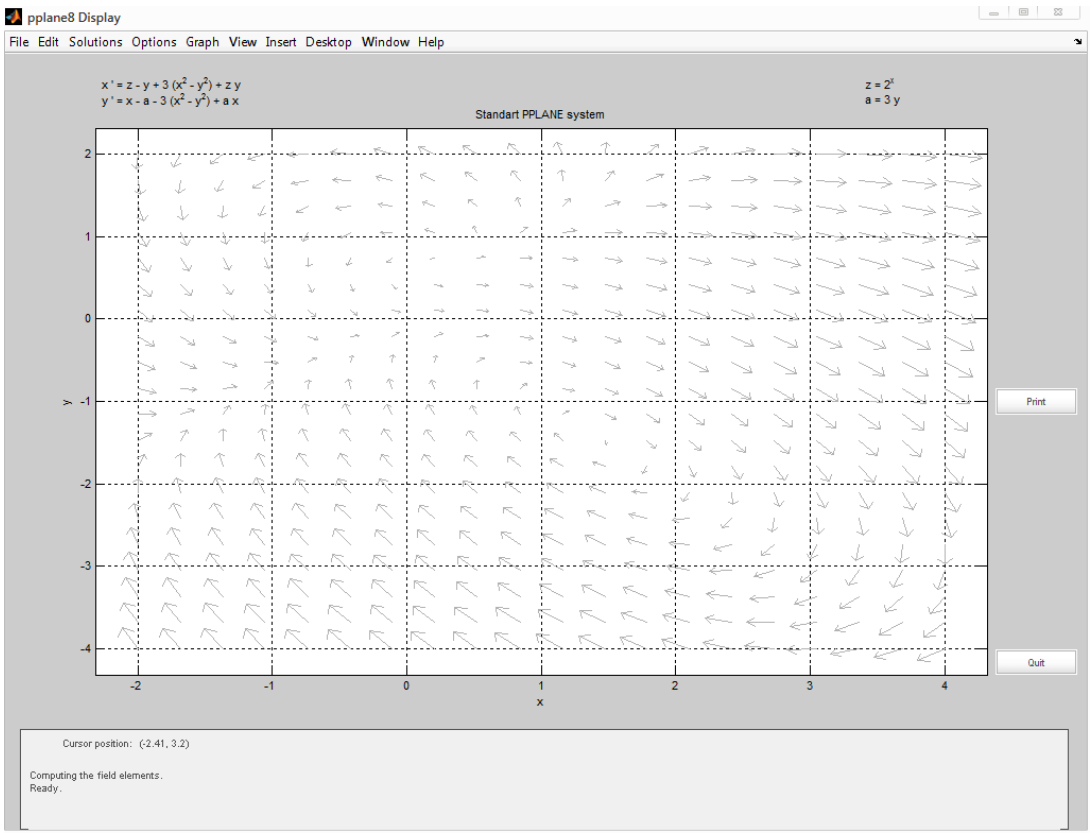

Рис. 3.2: Окно визуализации

3.1.2 Построение траекторий и фазового портрета

Для расчёта и построения фазовой траектории из выбранной пользователем точки следует навести курсор на данную точку и нажать левую кнопку

мыши. Траектория будет рассчитана и нанесена на график сначала в направлении, в котором значение независимой переменной увеличивается, а затем в обратном направлении. Немаловажным является тот факт, что в нижней части окна находится поле сообщений PPLANE, позволяющее почерпнуть важную информацию о свойствах данного графика. После построения ряда решений график будет выглядеть примерно следующим образом (Рис. 3.3):

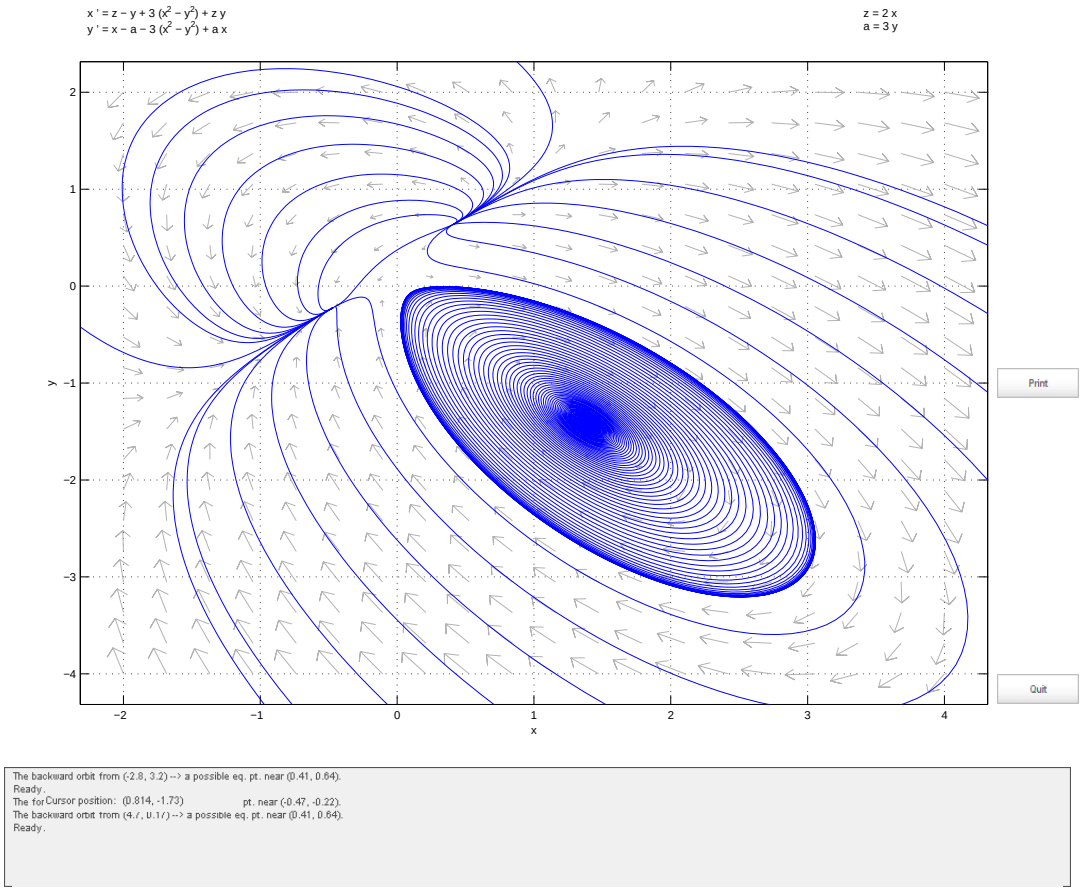

Рис. 3.3: Фазовый портрет системы

3.1.3 Анализ фазового портрета системы

Следует отметить, что все фазовые траектории начинаются и заканчиваются в одних и тех же двух точках, и окно сообщений подсказывает, что данные точки являются особыми точками данной системы. С помощью пункта меню *Options* возможно выбрать оптимальные настройки расчета и отображения фазовых траекторий для конкретной системы. Направление построения траекторий можно изменить с помощью пункта меню *Options* → *Solution Direction*. Также иногда может быть важен тот факт, из какой точки выхо-

дит та или иная фазовая траектория, что возможно отобразить с помощью пункта *Options* → *Mark initial points*. Выбрать приемлемый для данной ситуации численный метод решения дифференциальных уравнений позволит меню *Options* → *Solver*.

Важные с точки зрения анализа системы представлены в меню *Solutions*. Первый пункт *Solutions* → *Keyboard input* позволяет пользователю ввести координаты точки, через которую требуется провести фазовую траекторию. Данный пункт может быть очень полезен, если пользователю требуется визуализировать процесс автономной системы в конкретных точках. Функция *Solutions* → *Plot several solutions* позволяет с помощью курсора обозначить все точки, через которые PPLANE должен провести фазовые траектории, и сделано это будет только после нажатия пользователем клавиши *Enter*. Данная функция ускоряет процесс визуализации при работе с сложными системами.

Одной из важнейших функций PPLANE является быстрое нахождение особых точек. Выбрав *Solutions* → *Find an equilibrium point* и, нажав левой кнопкой мыши на любую точку на графике, возможно узнать, имеются ли в её окрестности особые точки. В случае нахождения таковых программа обведёт их красным на графике и предоставит пользователю координаты, якобианы, собственные значения и собственные векторы данной точки в окне сообщений. С помощью линеаризации легко можно узнать тип особой точки: например, для данного случая особая точка является *седлом* (Рис. 2.4).

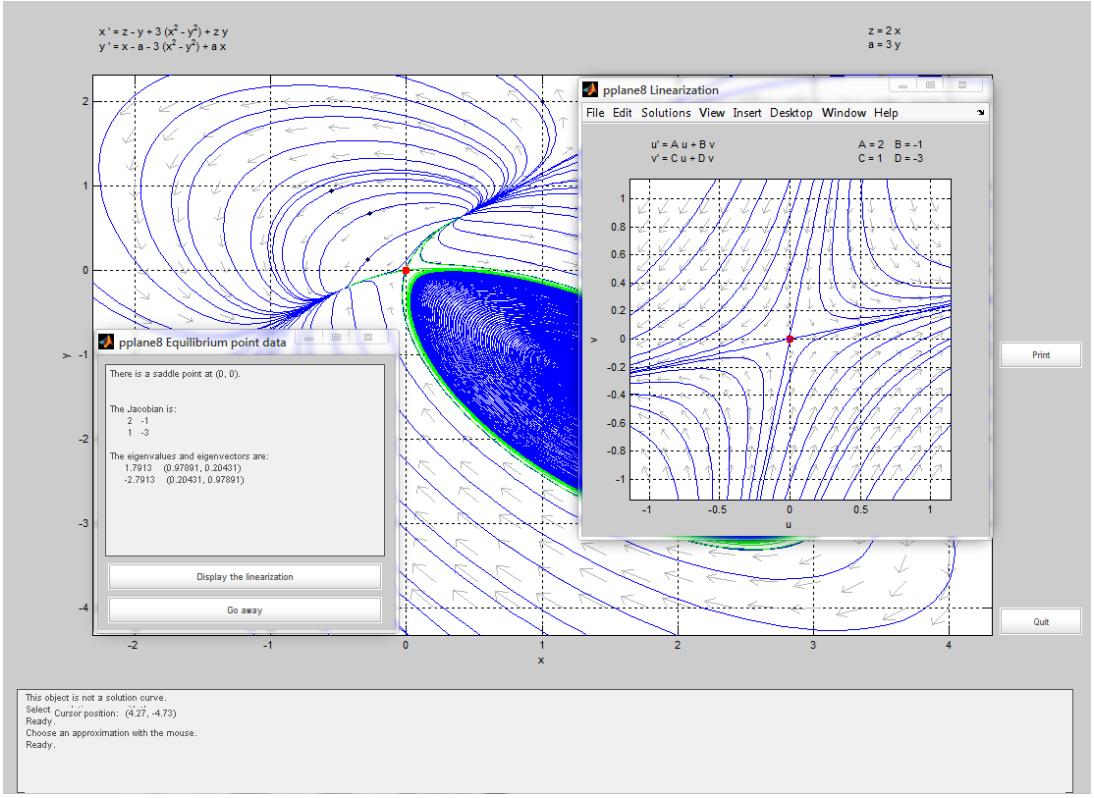

Рис. 3.4: Линеаризация особой точки

Подпункт меню *Solutions* → *Plot stable and unstable orbits* позволит с лёгкостью найти поблизости устойчивые и неустойчивые орбиты. Так выглядит фазовый портрет после нахождения точек равновесия и проведения нескольких фазовых траекторий и обрит (Рис. 3.5):

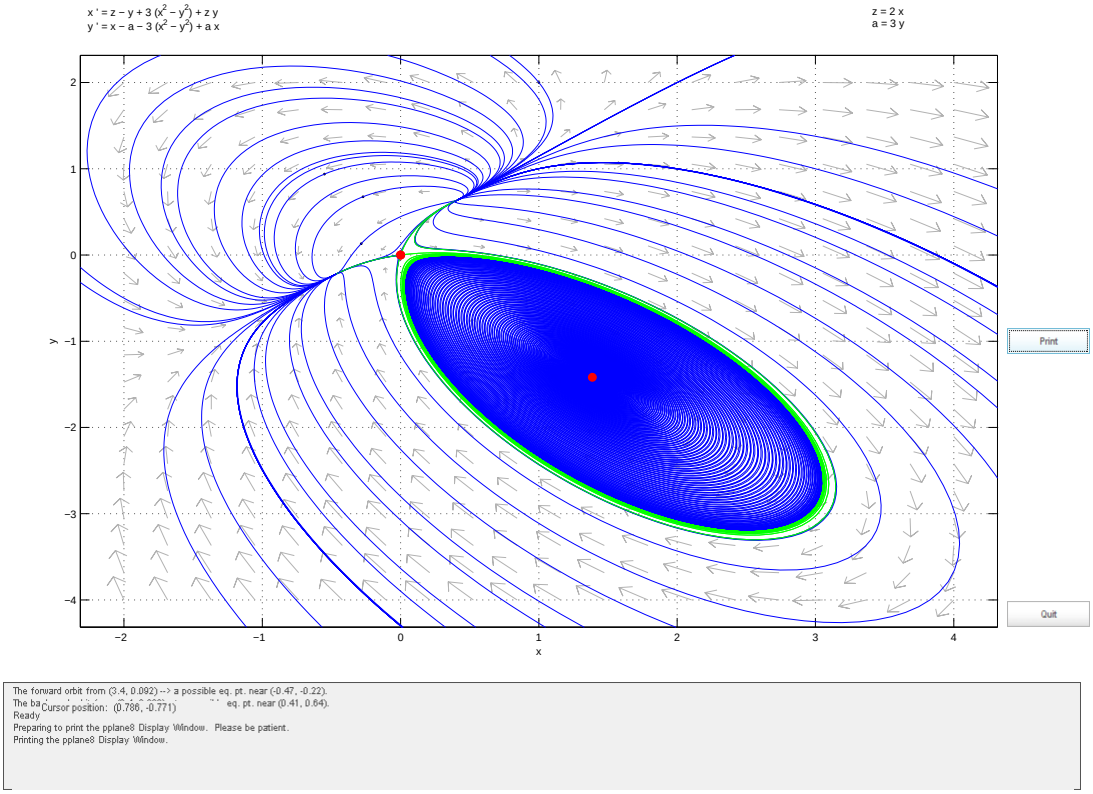

Рис. 3.5: Особые точки системы

3.1.4 Построение графиков зависимости функций

PPLANE предоставляет возможность не только изображать фазовые портреты, но и графики зависимости двух функций друг от друга или от независимой переменной t. Выбрав пункт меню *Graph* можно увидеть множество возможностей построения двух- и трёхмерных графиков. Ниже предоставлен график зависимости двух функций: x и y (Рис. 2.1):

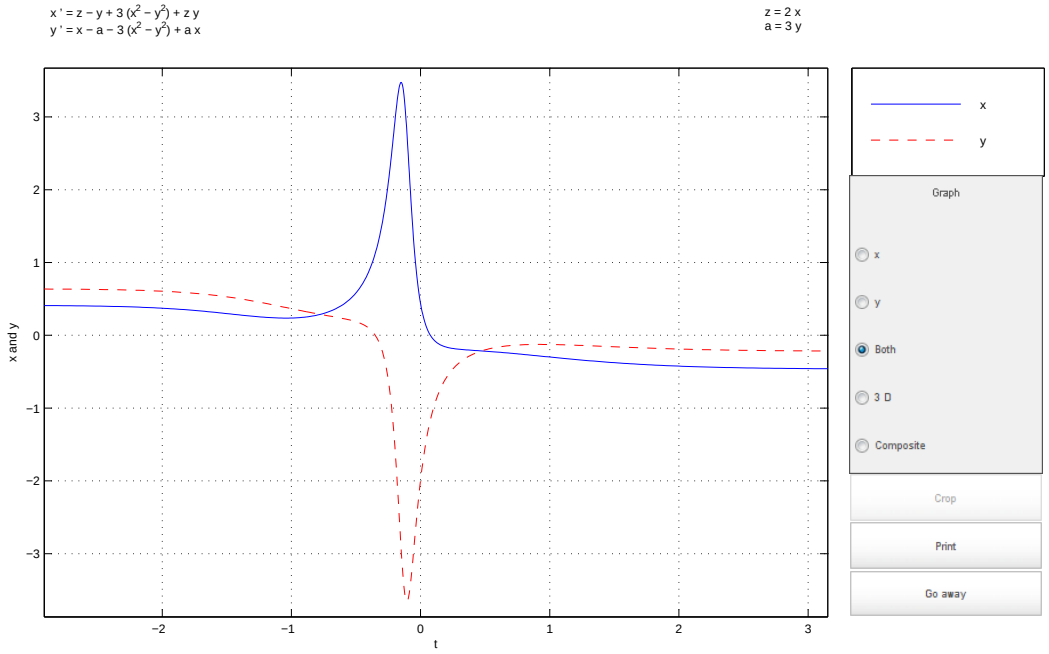

Рис. 3.6: График зависимости x от y

3.1.5 Стандартные функции

Рассмотрим пункт меню *Edit*. Помимо стандартных функций отмены действий, копирования, удаления и вставки объектов существует функция увеличения и уменьшения масштаба выбранной части фазового портрета. Также полезной является функция увеличения выделенной области изображения: *Edit* → *Zoom in square*. В любой момент пользователь с помощью пунктов меню *File* может сохранить или распечатать, импортировать уже существующий или экспортировать имеющийся график, что делает PPLANE отличной программой для использования в целях обучения.

Глава 4

Практическое использование PPLANE

4.1 Примеры исследования процессов с помощью PPLANE

В данной главе будут описаны способы практического применения инструментария. Предмет исследования успешно применяется также и в нашем университете, например, в рамках курса ISS0080 — Управление автоматизацией и процессами, преподаваемого в Таллиннском Техническом Университете. Разберём несколько из примеров данного курса.

Ещё два примера будут взяты из книги Хассана Халила “Нелинейные системы” и его курса “Введение в нелинейное управление”.

4.1.1 Трафальгарское сражение (1805)

27 кораблей под предводительством адмирала Горацио Нельсона сражаются против 33 кораблей флота Франции и командующего Пьера Вильнёва [5]. Скорость уничтожения кораблей во время сражения пропорциональна количеству вражеских судов, и в среднем любой из кораблей может быть уничтожен за полдня. Данное сражение может быть использовано как основа для составления модели динамической системы второго порядка, которая будет проанализирована ниже с помощью MATLAB и PPLANE. Для получения

графиков и информации с помощью MATLAB требуется написание специального скрипта, позволяющего проанализировать модель, а в PPLANE требуется лишь ввести уравнения модели и возможные параметры, если потребуется.

Уравнения для данной системы имеют вид

$$\begin{cases} \frac{\mathrm{d}N(t)}{\mathrm{d}t} &= -kV(t) \\ \frac{\mathrm{d}V(t)}{\mathrm{d}t} &= -kN(t) \end{cases}, \tag{4.1}$$

где переменная k и показывает, что один из кораблей уничтожается за полдня.

Используя стандартные функции MATLAB возможно получить следующий график зависимости потерь обеих сторон конфликта с течением времени (Рис. 4.1):

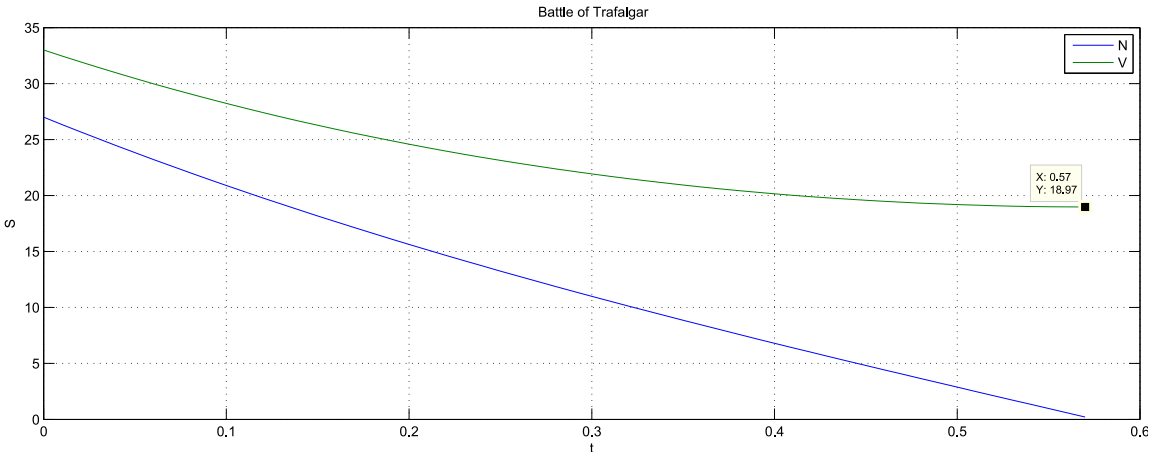

Рис. 4.1: График изменения количества кораблей сторон со временем (MATLAB)

Подобный график (Рис. 4.1) позволяет получить общие представления о поведении системы во времени, однако при использовании PPLANE возможно получить дополнительные сведения о системе. Ввод данных в PPLANE для анализа модели соответственно заданным уравнениям модели (Рис. 4.2):

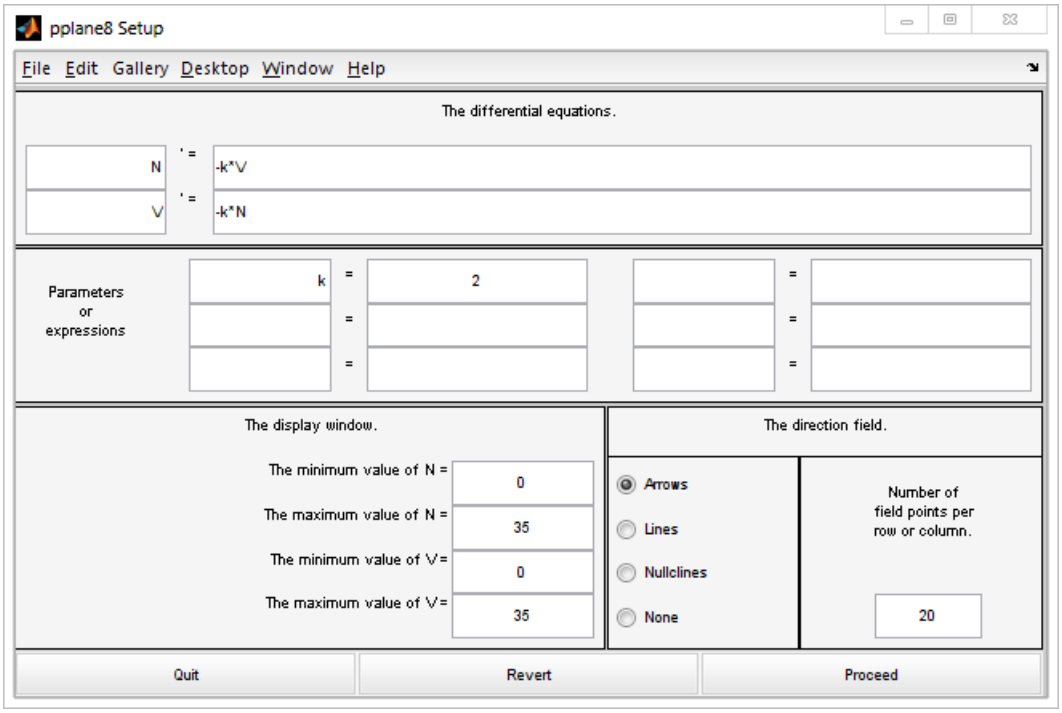

Рис. 4.2: Настройка параметров PPLANE для данной системы

Ниже представлены фазовый портрет и график зависимости, аналогичной полученному с помощью стандартных функций MATLAB (Рис. 4.3):

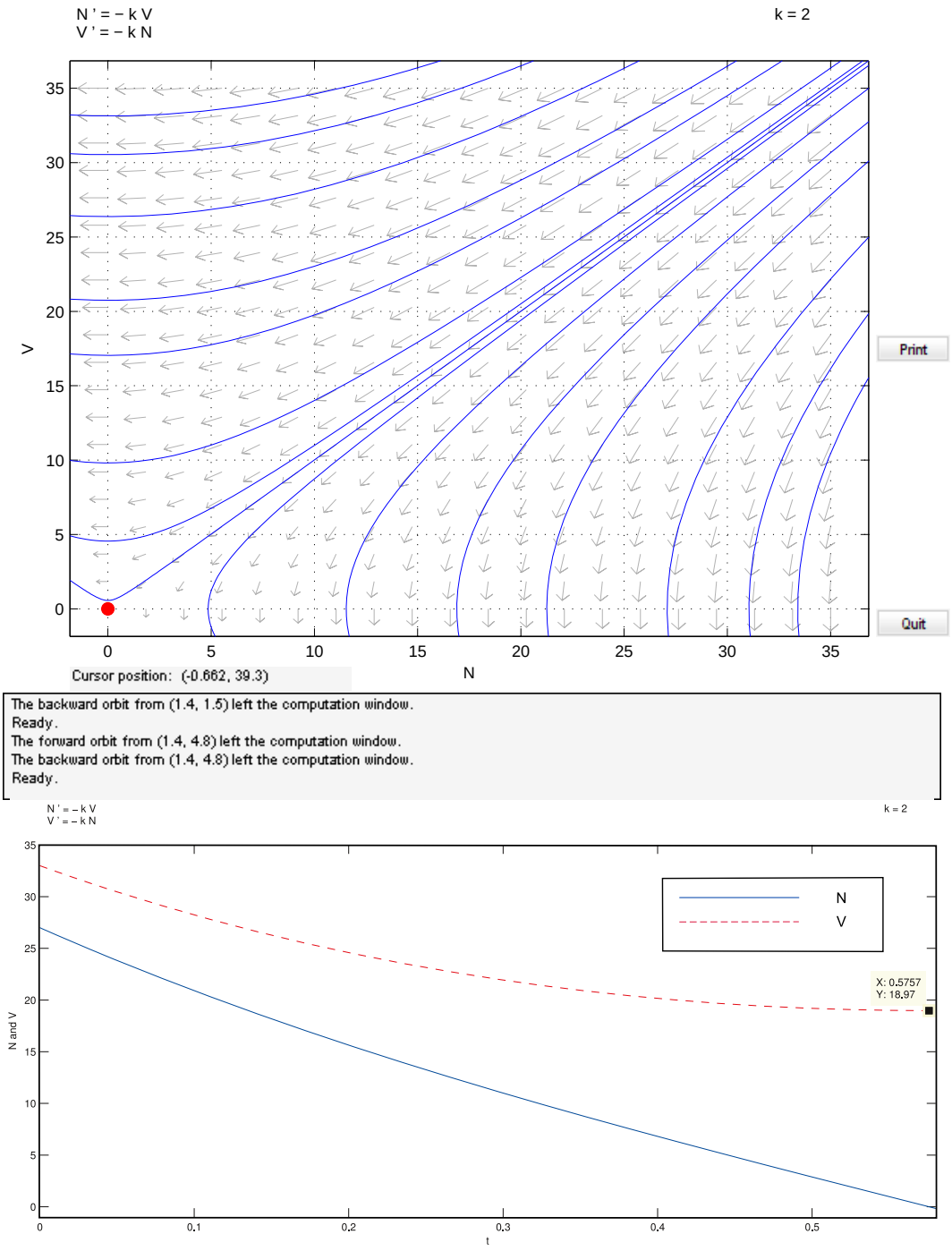

Рис. 4.3: График изменения количества кораблей сторон со временем (PPLANE)

Как можно заметить, график, предоставленный MATLAB с помощью специального скрипта, использующегося в рамках вышеуказанного курса, полностью совпадает с графиками, полученными с помощью PPLANE, однако с помощью последнего получено гораздо больше информации. В процессе исследования данных графиков возможно ответить на следующие вопросы, являющиеся частью практического задания предмета ISS0080:

- победу одержат силы флота Франции за 14 часов, при чём после полно-

го уничтожения всей флотилии противника на воде останутся 19 судов;

- при увеличении скорости уничтожения судов, бой закончится победой, соответственно, раньше;
- точка равновесия типа седло находится в начале координат $(0, 0)$;
- процесс является неустойчивым, собственными значениями матрицы A являются $\lambda_1 = -2$, $\lambda_2 = 2$.

Таким образом, проанализировав систему, можно сделать вывод, что для победы в данной битве королевский военно-морской флот Великобритании должен выделить адмиралу Нельсону как минимум на один корабль больше, чем имеется у его противников, иначе его ждёт бесславное поражение.

4.1.2 Битва за Атлантику (Вторая Мировая Война)

Разберём другой пример из истории — Вторую битву за Атлантику. Бой между немецкими подводными лодками (U) в количестве 210 штук и 130 британскими эсминцами (D). Скорость уничтожения плавательных средств во время баталии пропорциональна количеству вражеских судов, и в среднем одно судно уничтожает 0.25 судов противника в неделю. Германия производит две подлодки в неделю. В данном случае также возможно проанализировать происходящее с помощью построенной модели и узнать, кто же победит в данном сражении.

Уравнения для данной системы имеют вид

$$\begin{cases} \frac{\mathrm{d}U(t)}{dt} &= -kD(t) + Un \\ \frac{\mathrm{d}D(t)}{dt} &= -kU(t) + Dn \end{cases}, \tag{4.2}$$

где Un и Dn — количество судов, производимых Германией и Англией в неделю. В случае, если Великобритания так же производит по два эсминца в неделю, график зависимостей двух функций в MATLAB будет выглядеть следующим образом (Рис. 4.4):

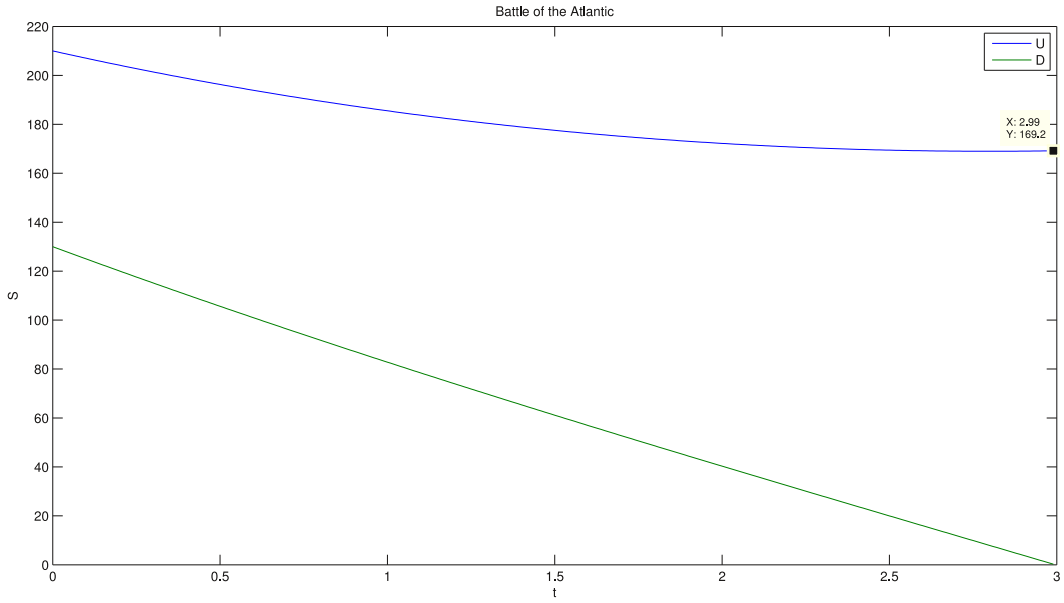

Рис. 4.4: График изменения количества судов сторон со временем (MATLAB)

В PPLANE он выглядит идентично (Рис. 4.5):

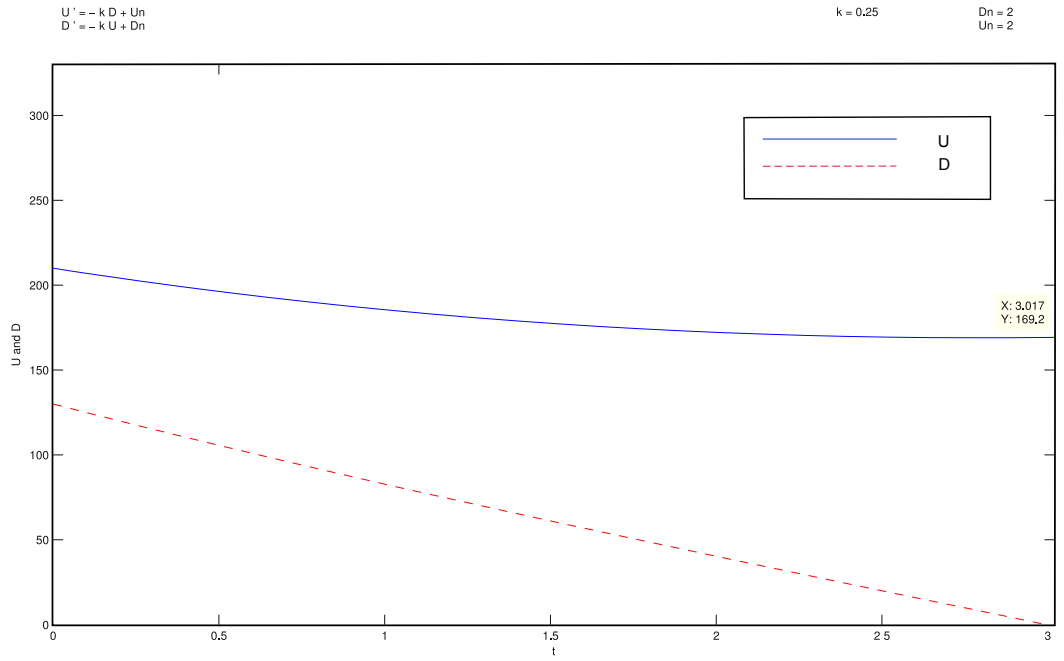

Рис. 4.5: График изменения количества судов сторон со временем (PPLANE)

Фазовый портрет будет выглядеть следующим образом (Рис. 4.5):

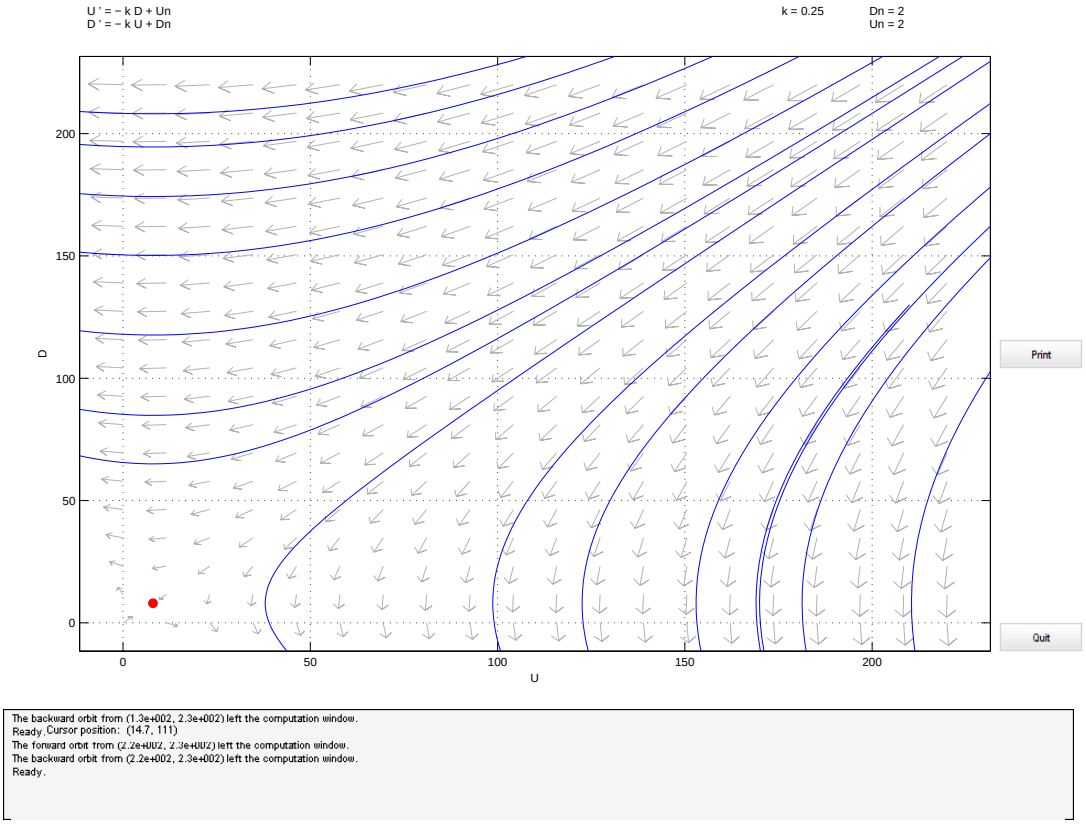

Рис. 4.6: Фазовый портрет системы (PPLANE)

Как любой может заметить, в этом случае Великобритания проиграет. Требуется экспериментально вычислить то количество эсминцев, которое нужно выпускать военно-морскому флоту Великобритании, чтобы выиграть в битве (при условии, что Германия по-прежнему выпускает по 2 подлодки в неделю). График зависимостей в данном случае (Рис. 4.7):

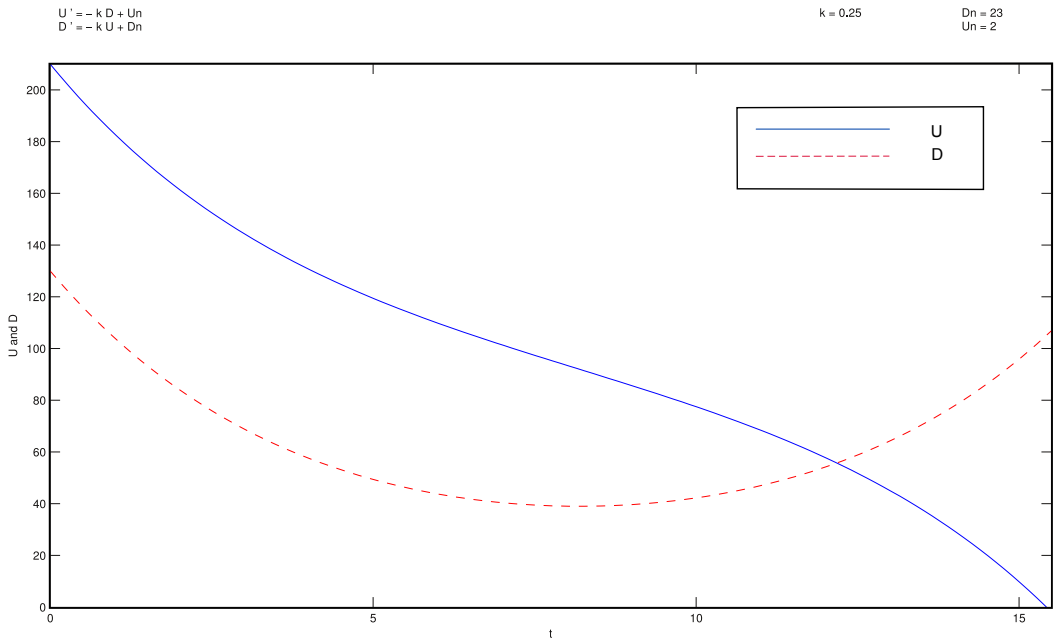

Рис. 4.7: График зависимости количества судов сторон при условии, что Великобритания победит (PPLANE)

Итак, Великобритания сможет одержать победу только в случае, если будет выпускать по 23 эсминца в неделю: только это сможет перевернуть ход битвы. Результаты анализа начальной модели:

- в исходном варианте Германии понадобится три недели для победы над противником, потеряв всего 40 подлодок;
- координаты точки равновесия типа седло $(8, 8)$;
- процесс является неустойчивым, а собственные значения матрицы A равны $[0.25, -0.25]$.

4.1.3 Задача курса “Введение в нелинейное управление”

Следующим будет рассмотрен пример из курса “Введение в нелинейное управление”, преподаваемого Хассаном Халилом в 2012 году. Дана следующая функция:

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1 - 2\tan^{-1}(x_1 + x_2) \end{cases}. \tag{4.3}$$

Требуется найти все точки равновесия данной системы и определить их типы, а также построить фазовый портрет системы и проанализировать её поведение.

Так как система является нелинейной, её входы равны нулю: $\dot{x}_1 = \dot{x}_2 = 0$, отсюда $0 = x_1 - 2\tan^{-1}(x_1 + 0)$. Таким образом можно получить функцию, используемую для построения фазового портрета (4.4):

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 2\tan^{-1}(x_1) \end{cases}. \tag{4.4}$$

Фазовый портрет функции выглядит следующим образом (Рис. 4.8):

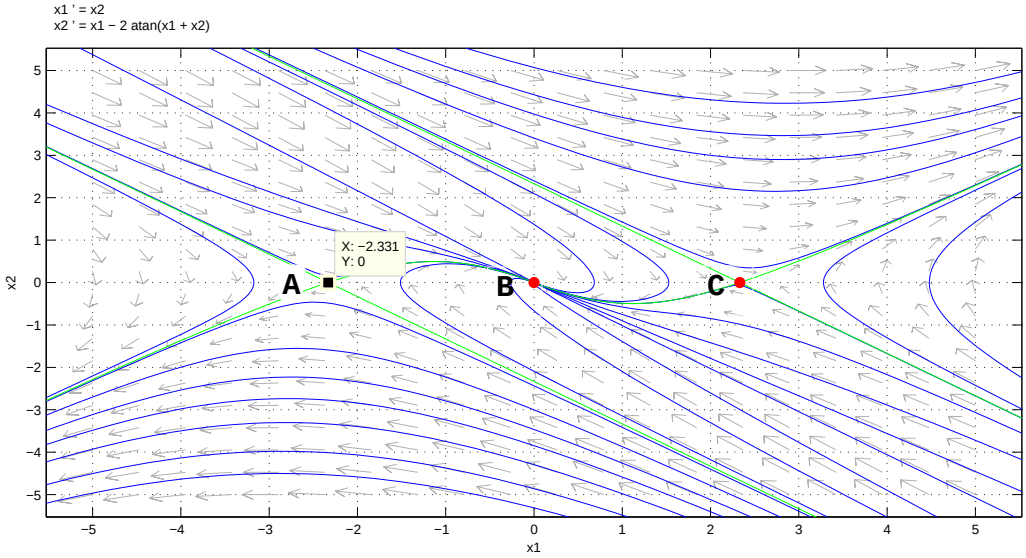

Рис. 4.8: Фазовый портрет предоставленной функции

Как можно заметить, данная система имеет три точки равновесия: одна из них лежит в центре координатной плоскости и имеет координаты $(0, 0)$, а две другие находятся на разных полюсах и имеют координаты $(-2.331, 0)$ и $(2.331, 0)$ соответственно. Рассмотрим их подробнее:

- $B_{(0,0)} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}, \lambda_1 = \lambda_2 = -1$

Из полученных данных можно сделать вывод, что в точка B является устойчивым узлом. С помощью PPLANE с лёгкостью можно это перепроверить на практике (Рис. 4.9).

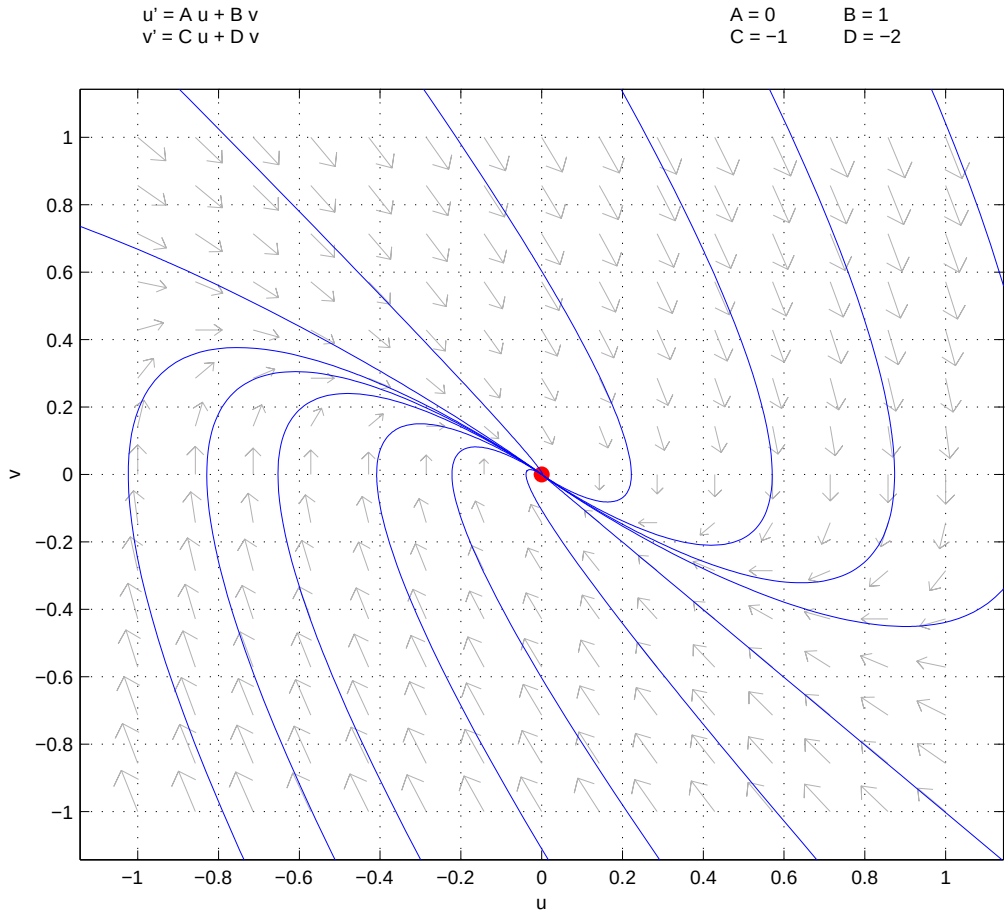

Рис. 4.9: Особая точка B типа устойчивый узел

Действительно, портрет полностью идентичен представленному в теоретической части портрету устойчивого узла. Рассмотрим две другие точки равновесия системы:

- $A_{(-2.331,0)} = \begin{bmatrix} 0 & 1 \\ 0.689 & -0.311 \end{bmatrix}, \lambda_1 = 0.689, \lambda_2 = -1,$
- $C_{(2.331,0)} = \begin{bmatrix} 0 & 1 \\ 0.689 & -0.311 \end{bmatrix}, \lambda_1 = 0.689, \lambda_2 = -1.$

В данном случае точки одинаковы, но лежат на разных полюсах, что, естественно, не изменяет их тип — седло. Линеаризация в районе точки A выглядит так (Рис. 4.10):

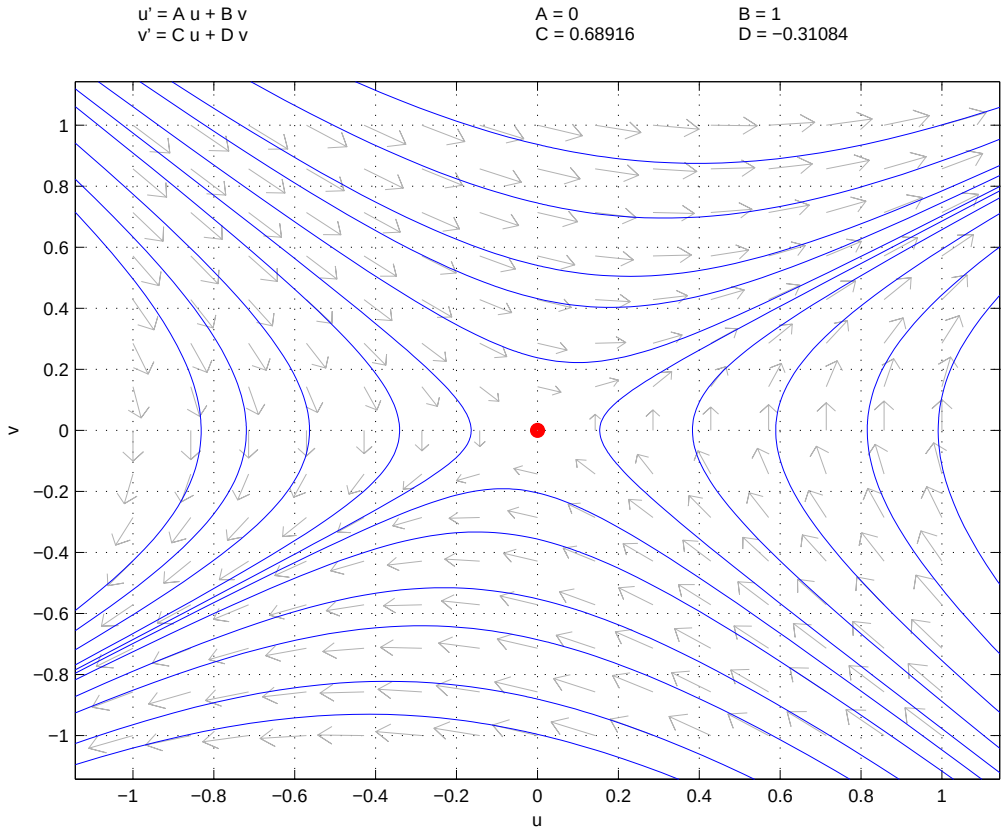

Рис. 4.10: Особая точка A типа седло

Исходя из соответствия данных, полученных с помощью анализа фазового портрета и рассмотрения самих особых точек, можно сделать вывод, что задание решено верно.

4.1.4 Обратный маятник

В качестве заключительного примера автор выбрал обратный маятник — отлично подходящее задание для исследований подобного рода [4]. Как и в предыдущем случае, имеем дело с динамической системой (4.5):

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l}\sin(x_1) - \frac{k}{m}x_2 \end{cases}, \tag{4.5}$$

где g — ускорение, l — радиус круга, который описывает подвес маятника, k — коэффициент трения и m — масса подвеса.

При построении фазового портрета (4.11) используется начальная функция

при условии, что $l = k = m = 1$ — так обеспечивается наглядность. Входы системы приравнены к нулю: $\dot{x}_1 = \dot{x}_2 = 0$.

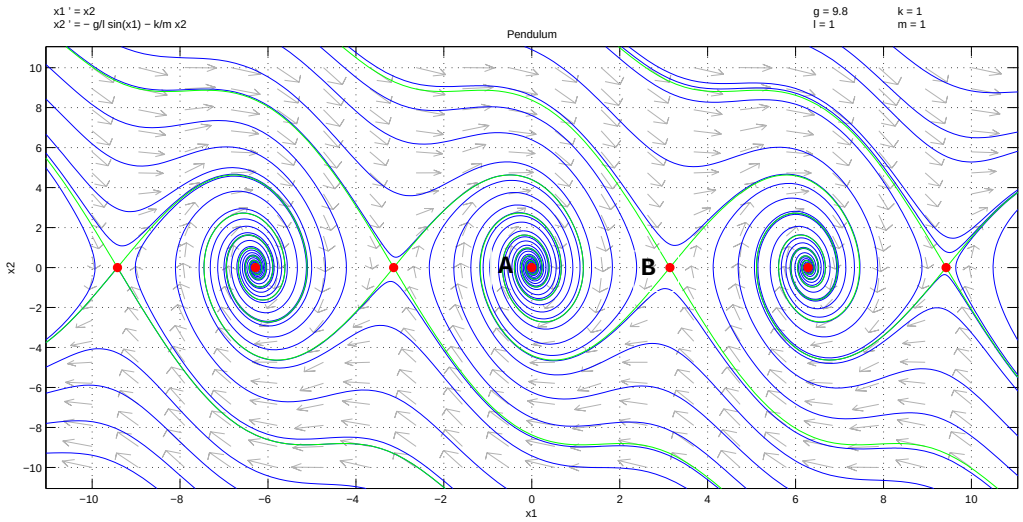

Рис. 4.11: Фазовый портрет системы обратного маятника

Как можно заметить, все точки равновесия системы располагаются на $(n\pi, 0)$ при $n = 0, \pm 1, \pm 2, ...$ Однако только две из них являются точками положения равновесия данной функции — это $(0, 0)$ и $(\pi, 0)$; остальные же являются их копиями, количество которых зависит от того, сколько полных оборотов совершил маятник, пока находился в положении равновесия. Ниже проанализированы точки, в которых система находится в положении равновесия:

- $A_{(0,0)} = \begin{bmatrix} 0 & 1 \\ -9.8 & -1 \end{bmatrix}, \lambda_1 = -0.5 + 3i, \lambda_2 = -0.5 - 3i.$

В соответствии с теоретической частью, полученные собственные значения указывают на то, что типом особой точки является устойчивый фокус. Сверим результат с реальной картиной на Рис. 4.12:

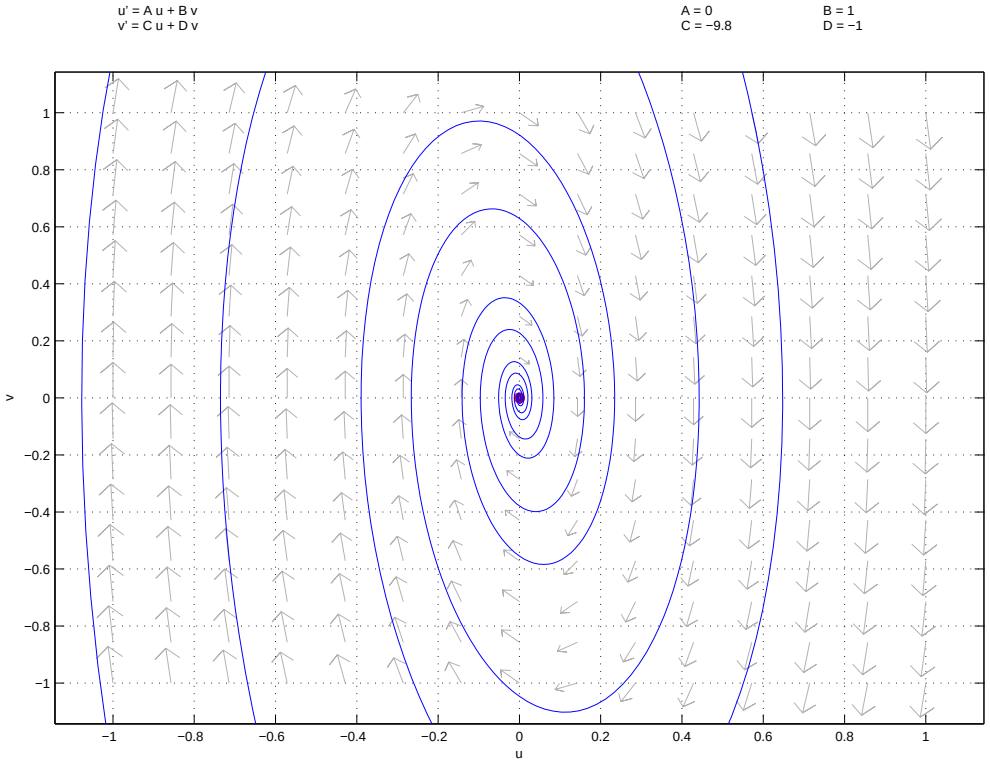

Рис. 4.12: Особая точка A типа устойчивый фокус

Вторая точка равновесия находится в координатах $(\pi, 0)$:

- $B_{(\pi,0)} = \begin{bmatrix} 0 & 1 \\ -9.8 & -1 \end{bmatrix}, \lambda_1 = 2.67, \lambda_2 = -3.67$

Оба корня являются вещественными, но различаются знаками: имеем дело с особой точкой типа седло. График на Рис. 4.13 наглядно демонстрирует истинность данного вывода.

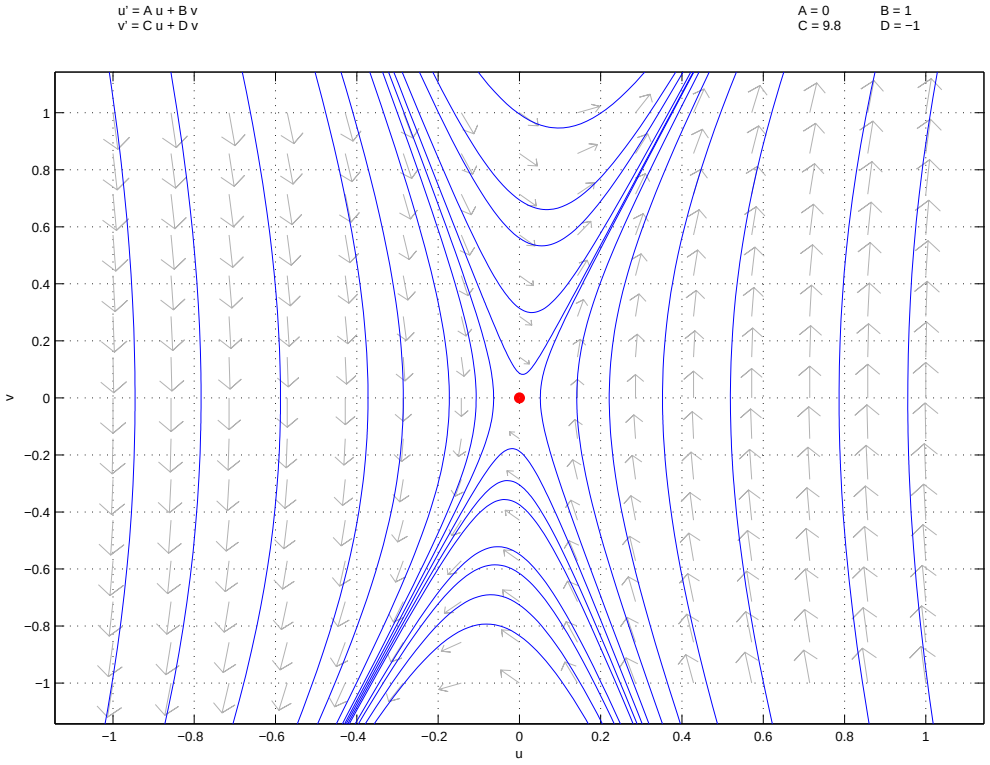

Рис. 4.13: Особая точка B типа седло

Таким образом, PPLANE8 качественно выполняет все вычисления и задачи по визуализации процессов, поставленные перед инструментарием в данной главе. Стоит отметить тот факт, что PPLANE может использоваться не только при анализе простых систем, но и незаменим в процессе исследования более сложных систем второго порядка.

Заключение

В ходе исследования была проведена обширная работа с теоретическим материалом, что позволило автору достаточно подробно разобрать все основные нюансы при анализе нелинейных систем второго порядка методом фазовой плоскости. Детально описан смысл самого метода и тех функций, что используются во время анализа. Были изучены и описаны все возможные типы особых точек, а также возможности по их определению.

Затем автор во всех подробностях описал возможности самого инструментария PPLANE8 и его интерфейс, что может существенно снизить время ознакомления с программой человека, никогда не использовавшего её ранее. Этапы анализа описаны один за одним, что в свою очередь позволяет добиться понимания своих действий читателем, даже если программа используется им в первый и последний раз. Все значимые функции были разобраны и отражены в работе.

В ходе практического использования инструментария для анализа различных систем возникли некоторые сложности с экспортированием графиков в векторный формат, однако, со всеми своими основными обязанностями PPLANE8 справился на отлично. Оказалось, что с его помощью можно получить куда больший объём информации о системе в более краткие сроки по сравнению с обычными инструментами MATLAB. Интуитивный интерфейс также способствует получению быстрого и качественного результата. Успешно были прорешены некоторые примеры из курса магистратуры Таллиннского технического университета, а также примеры, предоставленные Хассаном Халилом.

Все результаты, полученные с помощью PPLANE8, полностью совпали с теоретическими и теми, что были добыты с помощью стандартных функций

MATLAB. Все те задачи, что ставились перед автором работы, были выполнены.

Можно смело сделать вывод, что PPLANE8 является незаменимым инструментом при анализе динамических систем второго порядка и должен быть использован в каждом учебном курсе, где подобные проблемы ставятся. В Таллиннском техническом университете данный инструментарий пока не пользуется значительной популярностью, однако, автор надеется, что данная работа будет полезна в качестве учебного материала и поможет как студентам, так и преподавателям учебных заведений тратить меньше времени с большей продуктивностью.

Литература

[1] Ким Д. П., *Теория автоматического управления. Т. 2. Многомерные, нелинейные, оптимальные и адаптивные системы.* ФИЗМАТЛИТ, 2004.

[2] J. Polking, *Ordinary Differential Equations Using MATLAB*, 3rd ed. Pearson, 2004.

[3] В. Иванов, "Суть и способы линеаризации нелинейных динамических систем," [Онлайн; доступен 27.05.2014] http://homehelper.in.ua/modelirovanie/sut-i-sposoby-linearizacii-nelineynyh-dinamicheskih-sistem.html, 2011.

[4] H. K. Khalil, *Nonlinear Systems Third Edition*, M. Horton, Ed. Prentice Hall, 2002.

[5] A. Tepljakov, *ISS0080 Automation and Process Control*, TUT, 2014.