Ссылка на статью:
// Ученые записки УлГУ. Сер. Математика и информационные технологии. 2023, № 2, с. 92-98.

Поступила: 23.11.2023
Окончательный вариант: 11.12.2023

УДК 004.94

Разработка компьютерной модели для оценки положения материальной точки при движении вдоль прямой средствами языка Python

Пацанков Д.Д., Рахимова Р.И.*

*rakhimova_ri@mail.ru
УлГУ, Ульяновск, Россия

В работе рассматривается задача оценивания положения материальной точки при движении вдоль прямой. Решение получено методами математического и компьютерного моделирования. Построена математическая дискретная линейная стохастическая модель движения и измерений в пространстве состояний. Оценки положения материальной точки по доступным измерениям получены с помощью алгоритма дискретного фильтра Калмана. Исследованы свойства дисперсии ошибки оценивания и получено аналитическое выражение для ее вычисления в установившемся режиме. Компьютерная модель разработана на языке Python. В работе рассмотрено понятие графического интерфейса и библиотек для его реализации. Практическим результатом работы является компьютерная модель в форме приложения с графическим интерфейсом, которое позволяет проводить численные эксперименты по оцениванию положения материальной точки при различных параметрах модели. Результаты компьютерного моделирования согласуются с полученным теоретическим выражением для дисперсии ошибки оценивания.

Ключевые слова: *математическое и компьютерное моделирование, дискретная стохастическая модель, фильтр Калмана, Python, графический интерфейс, GUI библиотеки*

Введение

Метод компьютерного моделирования заключается в построении модели, которая представляет собой компьютерную программу или комплекс программ, описывающий развитие процесса или поведение объекта исследования. В настоящее время компьютерное моделирование применяется во многих областях науки и техники, оно позволяет автоматизировать процессы управления, создавать различные имитаторы и тренажеры, визуализировать результаты моделирования и многое другое [1].

Как правило, компьютерная модель разрабатывается с помощью языка программирования высокого уровня. Язык программирования Python является самым простым и востребованным языком среди пользователей. Преимущества данного языка заключаются в универсальном и качественном синтаксисе. Программный код на языке Python читается легче в отличие от многих других языков программирования. Высокая скорость разработки, поддержка библиотек, переносимость программ, интеграция компонентов – это основные плюсы программного продукта Python [2].

С помощью языка Python можно создавать приложения для обработки больших данных, разрабатывать различные системные утилиты и мощные веб-приложения, проводить компьютерное моделирование с построением графиков, а также создавать компьютерные игры и мобильные приложения с графическим интерфейсом.

Целью данной работы является разработка компьютерной модели для решения задачи оценивания положения материальной точки вдоль оси OX средствами языка Python.

В разделе 1 построена математическая модель движения и измерений, записан алгоритм дискретного фильтра Калмана для данной модели, а также исследованы его свойства. В разделе 2 приведен краткий обзор библиотек языка Python и представлена разработанная компьютерная модель в форме приложения с графическим интерфейсом. В заключении приведены выводы по работе.

1. Постановка задачи и аналитическое решение

Рассмотрим задачу оценивания положения материальной точки при движении вдоль оси OX [3]. Пусть в любой момент времени t положение материальной точки вдоль оси OX, координата $x(t)$, есть сумма неизвестной (случайной) величины, равной $x(t_0) = x_0$, и процесса броуновского движения $\beta(t)$ с постоянной известной величиной диффузии σ_w^2. Измеряют положение $x(t)$ дискретно во времени с аддитивной случайной ошибкой, которая имеет нулевое среднее значение и дисперсию R. В момент измерения $t_i, i = 1,2,\dots$ ошибка измерения статистически не зависит от значений в другие моменты. Требуется разработать математическую и компьютерную модели для решения поставленной задачи.

Математическую модель будем строить в форме дискретной линейной стохастической модели процесса $x(t)$ и измерений $z(t_i)$.

Пусть $x(t) = x_0 + \beta(t)$, где $x(t)$ – координата положения материальной точки в момент времени t, $\beta(t)$ – винеровский процесс, т. е., $\beta(t) - \beta(s) \sim N\big(0,\ \sigma_w^2(t-s)\big)\ \forall 0 \le s < t < \infty$.

Пусть $x(t) = x_0 + \beta(t)$. Продифференцируем $x(t)$ по t:

$$\frac{dx(t)}{dt} = \frac{d\beta(t)}{dt} = w(t), \text{ где } w(t) - \text{белый гауссовский шум.}$$

$$\dot{x}(t) = w(t) = Fx(t) + Gw(t). \tag{1}$$

Таким образом, (1) является непрерывной моделью движения материальной точки с параметрами $F = 0$ и $G = 1$.

Перейдем к построению дискретной модели. Найдем переходную матрицу состояния $\Phi(t_{i+1}, t_i)$. Так как $F = 0 = const$, то $\Phi = \Phi(\Delta t) = \Phi(t_{i+1} - t_i)$.

$$\Phi(s) = (Is - F)^{-1} = s^{-1} = \frac{1}{s} \doteqdot 1 = \Phi.$$

Таким образом, $\Phi = 1$.

Найдем дисперсию дискретного шума $w_d(t_i)$ в уравнении состояния дискретной модели:

$$Q_d(t_i) = \int_{t_{i-1}}^{t_i} \Phi\, GQG^T \Phi^T d\tau = \int_{t_{i-1}}^{t_i} \sigma_w^2\, d\tau = \sigma_w^2 (t_i - t_{i-1}) = \delta_w^2 \Delta t.$$

Модель измерений запишем в следующем виде:

$$z(t_i) = x(t_i) + v(t_i), \qquad i = 1,2, \dots, K,$$

где $v(t_i) \sim N(0, R)$ – ошибка измерений, моделируемая гауссовской случайной величиной с нулевым математическим ожиданием и дисперсией R: $E\{v(t_i)\} = 0, E\{v(t_i)v^T(t_i)\} = R$.

Дискретная линейная стохастическая модель в пространстве состояний имеет вид:

$$\begin{cases} x(t_{i+1}) = \Phi(t_{i+1}, t_i)x(t_i) + G_d(t_i)w_d(t_i), \\ \quad z(t_i) = H(t_i)x(t_i) + v(t_i), \end{cases}$$

где $w_d(t_i) \sim N\big(0, Q_d(t_i)\big)$, $v(t_i) \sim N(0, R(t_i))$.

Запишем полученную дискретную модель процесса движения материальной точки вдоль оси *OX:*

$$\begin{cases} x(t_{i+1}) = x(t_i) + w_d(t_i), \\ \quad z(t_i) = x(t_i) + v(t_i), \end{cases} \tag{2}$$

где $Q_d = \sigma_w^2 \Delta t$, $x(t_0) = x_0$, $P(t_0) = P_0$.

Для оценки положения материальной точки по данным измерений $z(t_i)$, $i = 1,2, \dots, K$, будем использовать алгоритм дискретной фильтрации Калмана [4]. Учитывая уравнение (2), запишем алгоритм Калмана в следующем виде:

Алгоритм 1:

I. Начальные условия $\hat{x}(t_0^+) = x_0$, $P(t_0^+) = P_0$.

Для $i = 0,1,2, \dots, K-1$

II. Экстраполяция оценки положения материальной точки $x(t_{i+1})$ в момент t_{i+1} и дисперсия ошибки оценивания:

$$\hat{x}(t_{i+1}^-) = \hat{x}(t_i^+); \tag{3}$$

$$P(t_{i+1}^-) = P(t_i^+) + \sigma_w^2 \Delta t. \tag{4}$$

III. Обновление оценки $\hat{x}(t_{i+1})$ по измерению $z(t_{i+1})$ и дисперсии ошибки оценивания:

$$K(t_{i+1}) = \frac{P(t_{i+1}^-)}{P(t_{i+1}^-) + R}; \tag{5}$$

$$P(t_{i+1}^+) = \left(1 - K(t_{i+1})\right)P(t_{i+1}^-); \tag{6}$$

$$\hat{x}(t_{i+1}^+) = \hat{x}(t_{i+1}^-) + K(t_{i+1})[z(t_{i+1}) - \hat{x}(t_{i+1}^-)]. \tag{7}$$

Исследуем свойства Алгоритма 1.

1) Покажем, что дисперсия ошибки фильтрации $P(t_i^+) \in [0; R]$. Из (4), (6) запишем

$$P(t_i^+) = \left(1 - K(t_i)\right)P(t_i^-) = \frac{RP(t_i^-)}{P(t_i^-)+R}.$$

Функция $P(t_i^+)$ является монотонно возрастающей по $P(t_i^-)$ и $P(t_i^+) \geq 0$ по определению. Если $P(t_i^-) = 0 =>$ и $P(t_i^+) = 0$.

Если $P(t_i^-) \to \infty => \lim_{P(t_i^-)\to\infty} \frac{RP(t_i^-)}{P(t_i^-)+R} = \lim_{P(t_i^-)\to\infty} \frac{R}{1+P(t_i^-)} = R$. Таким образом, $P(t_i^+) \in [0; R]$.

2) Покажем, что $K(t_i) \in [0; 1]$. Из (5) $K(t_i) = \frac{P(t_i^-)}{P(t_i^-)+R}$ и $P(t_i^-) \geq 0$. Если $P(t_i^-) = 0 =>$ $K(t_i) = 0$. Функция $K(t_i)$ монотонно возрастает по $P(t_i^-)$. Если $P(t_i^-) \to \infty$, то $\lim_{P(t_i^-)\to\infty} K(t_i) = \lim\limits_{P(t_i^-)\to\infty} \frac{P(t_i^-)}{P(t_i^-)+R} = \lim_{P(t_i^-)\to\infty} \frac{R}{1+\frac{R}{P(t_i^-)}} = 1$. Следовательно, $K(t_i) \in [0; 1]$.

3) Найдем значения дисперсии ошибки фильтрации в установившемся режиме. Из уравнений (4), (5), (6) получаем $P(t_{i+1}^+) = \frac{RP(t_{i+1}^-)}{P(t_{i+1}^-)+R}$.

С учетом (4),

$$P(t_{i+1}^+) = \frac{R(P(t_i^+)+\sigma_w^2\Delta t)}{P(t_i^+)+\sigma_w^2\Delta t+R}. \tag{8}$$

Обозначим $\bar{P} = P(t_{i+1}^+) = P(t_i^+)$. Тогда из (8) получим уравнение

$$\bar{P} = \frac{R(\bar{P}+\sigma_w^2\,\Delta t)}{\bar{P}+\sigma_w^2\,\Delta t+R}. \tag{9}$$

Преобразуем (9) и запишем квадратное уравнение относительно $\bar{P}$:

$$\bar{P}^2 + \sigma_w^2\Delta t * \bar{P} - R\sigma_w^2\Delta t = 0. \tag{10}$$

Найдем дискриминант $D = (\sigma_w^2\Delta t)^2 + 4R\sigma_w^2\Delta t > 0$, поскольку $\sigma_w^2 > 0, \Delta t > 0$ и $R > 0$.

Тогда $\bar{P}_{1.2} = \frac{-\sigma_w^2\,\Delta t \pm \sqrt{D}}{2}$. Так как $\bar{P} > 0$ по определению, то выбираем положительный корень $\bar{P} = \frac{-\sigma_w^2\,\Delta t + \sqrt{D}}{2}$. Окончательно преобразуем $\bar{P}$ к виду:

$$\bar{P} = \frac{\sigma_w^2\,\Delta t\left(-1+\sqrt{1+\frac{4R}{\sigma_w^2\,\Delta t}}\right)}{2}. \tag{11}$$

Таким образом, алгоритм 1 позволяет вычислить оценку положения материальной точки при движении вдоль оси OX с конечной дисперсией ошибки, равной $\bar{P}$, которая зависит от ковариаций шумов σ_w^2 и R, а также от интервала дискретизации Δt.

2. Построение компьютерной модели на языке Python

Компьютерную модель разработаем на языке программирования Python. Оптимальным решением представляется приложение с графическим интерфейсом, поскольку он является дружественным для пользователя с возможностью удобного ввода и изменения параметров компьютерной модели, а также наглядной визуализацией результатов моделирования.

Графический интерфейс пользователя – это оболочка программы, состоящая из различных окон с расположенными на них элементами управления, которые используются для взаимодействия пользователя с программой. В современных операционных системах GUI (Graphical User Interface) является стандартным интерфейсом для пользователей. Большинство разработчиков для создания программного продукта предпочитают использовать графический интерфейс на Python, так как удобный функционал и хороший выбор библиотек делают программу привлекательной и интуитивно понятной [2].

Python GUI имеет широкий выбор библиотек: Tkinter, PyQt, Kivy и др. Они обладают различными функциональными возможностями и поддерживают разнообразие стилей по оформлению интерфейса. GUI библиотеки в Python значительно упрощают и ускоряют процесс разработки, позволяют создать простой и быстрый графический интерфейс программы. Выбор библиотек зависит от требований к проекту и опыта разработчика. Современные программисты предпочитают использование Python GUI для создания приложений с графическим интерфейсом. Использование библиотек с готовым кодом позволяют разработчикам сосредоточиться над самим процессом создания проекта, что повышает качество работы программы, делая ее более быстродействующей и функциональной.

PyQt – кроссплатформенная библиотека на основе Qt для создания мощных графических приложений. Эта библиотека является самой популярной в кругу сообщества Python. Поддержка данного инструмента может осуществляться на любой операционной системе. Библиотека PyQt основана на языке C++, что обеспечивает высокую скорость работы и многообразие возможностей для пользовательского интерфейса. Главной особенностью является реализация паттерна проектирования модель-представления-контроллер (MVC), что позволяет разделять данные приложения и их отображение. PyQt имеет обширную документацию и большой выбор виджетов, благодаря которым можно создавать множество полезных функций, таких как автоматическое управление памятью, поддержка потоков, соединение сигнал/слот и многое другое [5].

Библиотека Tkinter больше всего подходит для создания несложных приложений, которые не требуют высокой производительности и дополнительных функций. PyQt – это уже полноценная кроссплатформенная библиотека, которая отлично подойдет для создания проектов, как для профессионалов, так и для начинающих программистов. Библиотека Kivy используется для создания мобильных приложений. Она отлично отображает пользовательские интерфейсы и их функциональные возможности. Больше подойдет для опытных разработчиков.

Таким образом, можно сделать вывод, что наиболее универсальной GUI библиотекой является PyQt. Большой выбор инструментария, поддержка обновлений, богатый выбор виджетов и библиотек позволяют создать простой и интуитивно понятный графический интерфейс для приложения. Используем данную библиотеку для разработки приложения.

Разработанная компьютерная модель представляет собой приложение с графическим интерфейсом, представленным на рис. 1.

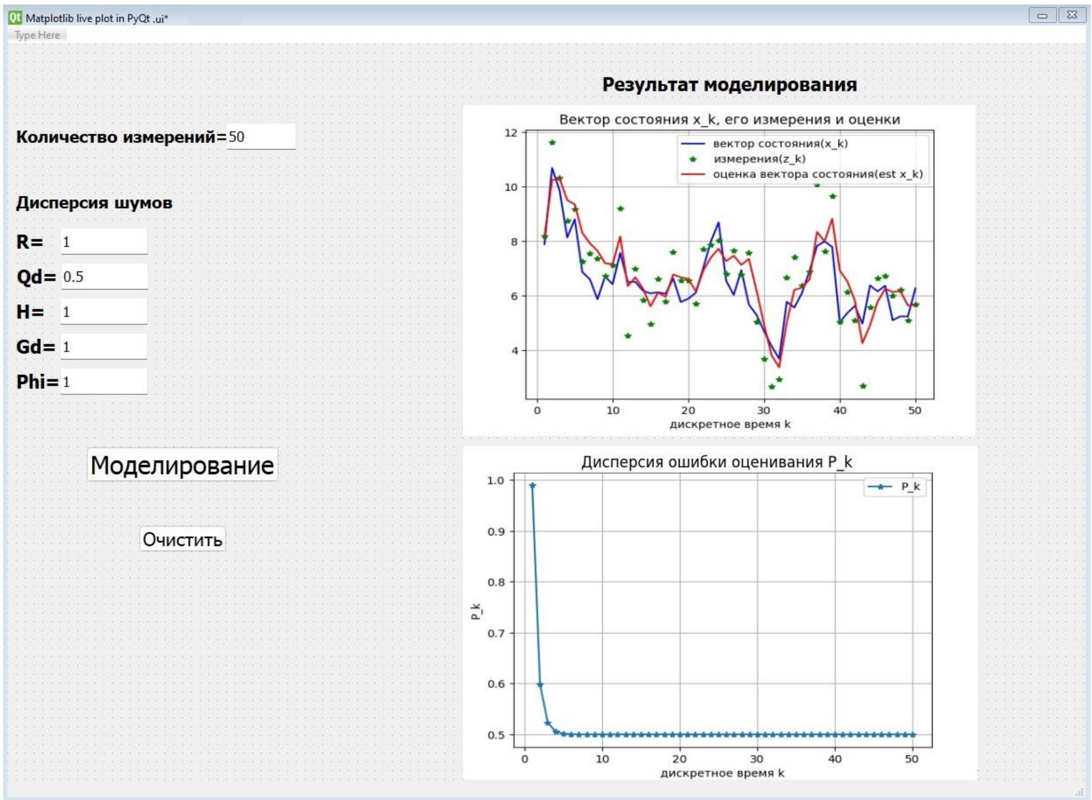

Рис. 1. Компьютерное моделирование движения материальной точки вдоль оси OX

На главной форме слева расположены окна для ввода значений параметров модели. Кнопка "Моделирование" запускает процесс компьютерного моделирования. Справа расположены окна для визуализации результатов. Справа вверху отображаются графики изменения координаты материальной точки, зашумленных измерений и оценок координат на заданном интервале моделирования. Справа снизу отображается график изменения дисперсии ошибки оценивания P. По результатам моделирования видно, что установившееся значение дисперсии ошибки, равное 0.5 при заданных значениях параметров модели (см. рис. 1), согласуется с полученным аналитическим выражением (11).

Заключение

В работе получено решение задачи оценивания положения материальной точки при движении вдоль оси OX методами математического и компьютерного моделирования. Построена дискретная линейная стохастическая модель движения и измерений в пространстве состояний. Для вычисления оценки положения материальной точки по доступным измерениям применялся алгоритм дискретного фильтра Калмана. Исследованы свойства дисперсии ошибки оценивания и получено аналитическое выражение для ее вычисления в установившемся режиме. Компьютерная модель в форме приложения с графическим интерфейсом разработана на языке Python. Программа позволяет проводить численные эксперименты по оцениванию положения материальной точки при различных параметрах модели. Результаты численных экспериментов показали, что значения дисперсии ошибки, вычисленные при работе алгоритма Калмана согласуются с полученным теоретическим выражением.

Список литературы

1. Королев А.Л., Паршукова Н.Б. *Компьютерное моделирование объектов, процессов и систем:* учебное пособие. Челябинск: Изд-во Южно-Урал. гос. гуманитар.–пед. ун-та, 2020. – 329 с.
2. Лутц М. *Изучаем Python, 3-е издание.* Пер. с англ. СПб.: Символ-Плюс, 2009. 848 с.
3. *Стохастические модели и оценки. Лабораторный практикум по курсу "Теория оптимального управления"* / Сост.: И. В. Семушин, Ю. В. Цыганова. Ульяновск, 2001. 42 с.
4. Grewal M. S., Andrews A. P. *Kalman Filtering: Theory and Practice Using Matlab.* 4th Edition. New Jersey: Prentice Hall, 2014. 640 p.
5. *Лучшие GUI библиотек Python 2023: обзор и сравнение.* Режим доступа: https://apipython.ru/top-10-luchshih-gui-bibliotek-python-2023-obzor-i-sravnenie/ (дата обращения: 27.10.2023).

Development of a computer model for estimating the position of a material point moving along a straight line using the Python language

Patsankov, D.D., Rakhimova, R.I.*

*rakhimova_ri@mail.ru

Ulyanovsk State University, Russia

The paper considers the problem of estimating the position of a material point when moving along a straight line. The solution was obtained using mathematical and computer modeling methods. A mathematical discrete-time linear stochastic state-space model of motion and measurements is constructed. Estimates of the position of a material point based on available measurements are obtained using the discrete-time Kalman filter algorithm. The properties of the estimation error variance were studied and an analytical expression for its calculation in a steady state was obtained. The computer model was developed in Python. The paper discusses the concept of a graphical interface and libraries for its implementation. The practical result of the work is a computer model in the form of an application with a graphical interface, which allows you to conduct numerical experiments to estimate the position of a material point for various model parameters. The computer simulation results are consistent with the obtained theoretical expression for the variance of the estimation error.

Key words: *mathematical modeling, simulation, discrete-time stochastic model, Kalman filtering, graphic interface, GUI libraries*