

Санкт-Петербургский Государственный Университет

Кафедра системного программирования

Дятлов Кирилл Александрович

# Разработка программного обеспечения системы прогнозирования изменения поведения временных рядов

Отчет по учебной (проектно-технологической) практике

Научный руководитель:  
д. ф.-м. н., профессор Граничин О. Н.

Санкт-Петербург  
2022

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1. Обзор</b>	<b>5</b>
1.1. Исходные данные . . . . .	5
1.2. Алгоритм SPS . . . . .	6
1.3. Схема работы алгоритма SPS . . . . .	8
<b>2. Реализация</b>	<b>12</b>
2.1. Реализация алгоритма SPS . . . . .	12
2.2. Реализация пользовательского интерфейса . . . . .	14
<b>3. Результаты</b>	<b>16</b>
<b>Список литературы</b>	<b>17</b>

# Введение

В настоящее время системы мониторинга временных рядов все чаще применяются в мире. Прогнозирование изменения их поведения позволяет получить значительную выгоду в широких областях экономики, начиная от предсказания изменения цены акций на бирже и заканчивая мониторингом пульса пользователя в цифровых наручных часах. Не обошла эта тенденция и индустриальный сектор: для одних из наиболее важных элементов во вращающихся механизмах – подшипников необходим регулярный мониторинг и контроль, т.к. на их долю приходится почти 50% всех случаев поломок и сбоев в таких машинах.[7, 1] Нормальным механизмом разрушения радиального шарикового подшипника является стирание контактных шариков. Обычно у подшипников имеется конкретный срок службы, однако, по ряду причин, таким как дефекты сборки, недостаточные смазывания, наличие сторонних дефектов в машине, а также неблагоприятные условия окружающей среды этот срок может значительно сократиться. К счастью, процесс деградации подшипника происходит достаточно медленно, поэтому имеется возможность его отследить и запланировать замену подшипника. Для получения сведений о состоянии подшипника на практике чаще всего используют метод вибрационного мониторинга. Например, стандарт ISO 13373-3 предлагает использовать среднеквадратичное значение (RMS) сигнала ускорения для выявления состояния подшипника. Метрика RMS измеряется в каждый конкретный момент времени, а полученные значения объединяются и упорядочиваются согласно времени измерения. Из всего многообразия временных рядов данная работа будет посвящена предсказанию изменения поведения ряда среднеквадратичного значения шариковых подшипников.

## Цель работы

Целью данной работы является исследование методов предсказания изменения поведения временных рядов в зависимости от характера исходных данных для их последующей реализации. Ее результатом станет прототип ПО, позволяющий прогнозировать изменения поведения различных временных рядов.

Для достижения поставленной цели необходимо решить следующие задачи:

- Изучение существующих алгоритмов, решающих аналогичные проблемы прогнозирования изменения поведения различных временных рядов
- Разработка прототипа системы с реализацией интерфейса для наглядного представления пользователю информации о прогнозах
- Реализация алгоритма прогнозирования на примере предсказания выхода из строя шариковых подшипников
- Анализ результатов работы алгоритма с лабораторными и индустриальными данными

# 1. Обзор

## 1.1. Исходные данные

В индустриальной отрасли существует два подхода к измерению среднеквадратичной метрики вибрации RMS. Первый способ предполагает постоянное снятие измерений с работающей машины, что позволяет собрать обширную историю данных (рис. 1). Однако данный способ достаточно дорогой, т.к. для каждого механизма нужно собственное измеряющее устройство и обслуживающий персонал, поэтому его используют только на самых важных станках на производствах. К такому способу также относят данные, полученные лабораторным путем. Второй способ куда менее затратный и заключается в периодическом снятии метрик с длительными временными интервалами между ними (рис. 2), например, неделя, месяц, три месяца и т.п. Недостаток данного способа заключается в том, что получающаяся история измерений весьма разрежена, а с точки зрения прогнозирования дальнейших изменений именно история процесса наиболее важна. Множеству алгоритмов требуется сравнимо большое количество данных в стадии деградации подшипника для формирования разумного прогноза[3, 6], тогда как на практике, например, имеется только 6 измерений в последние два месяца.

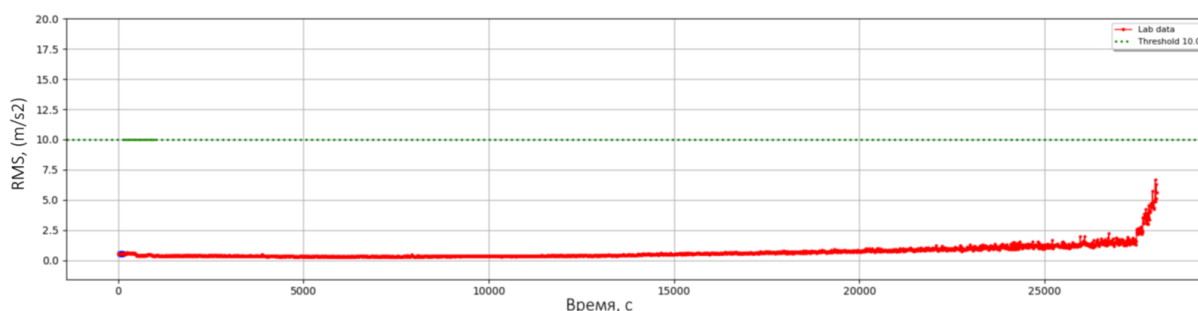


Рис. 1: Лабораторные данные

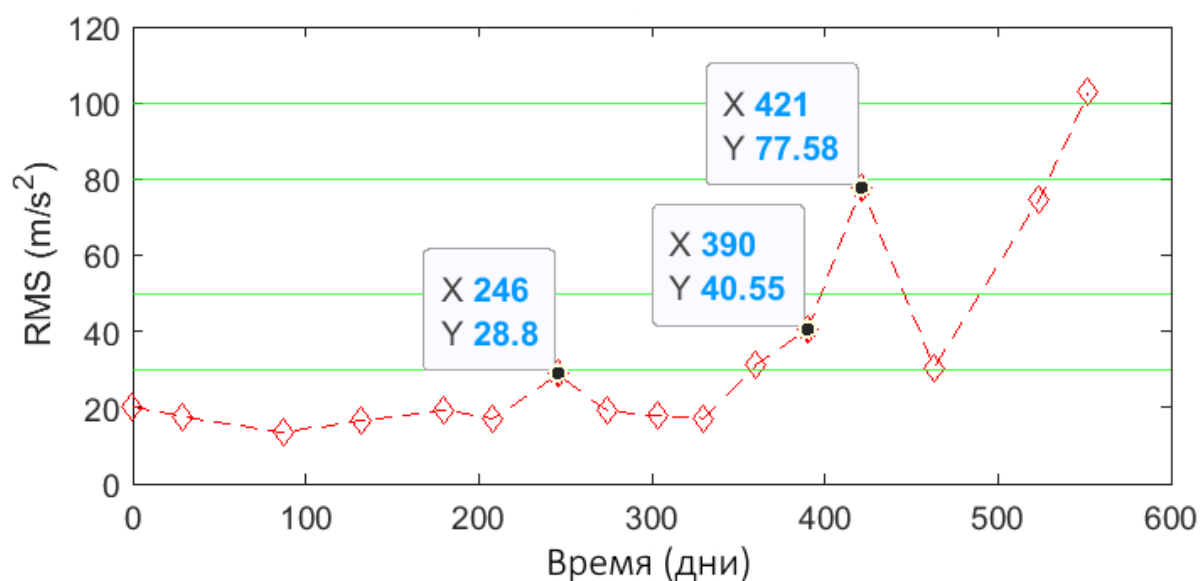


Рис. 2: Индустриальные данные

## 1.2. Алгоритм SPS

Один из алгоритмов, позволяющих справиться с такой задачей является алгоритм знакопеременных сумм (SPS)[2, 4], который дает возможность прогнозировать изменения тенденции временного ряда на основе малого числа исходных данных, в том числе изменения ряда RMS для подшипников. Алгоритм строит доверительное множество для заданной вероятности и включает в него подходящие для данной вероятности значения параметров модели. Алгоритм был успешно применен для оценки динамической прочности различных материалов[5, 8]. Рассмотрим его подробнее.

Для дальнейшего исследования введем ряд предположений:

- Зависимость вибрации от времени имеет вид кусочно-линейной функции, в которой каждая ступень соответствует своему режиму работы подшипника, т.е. стабильный режим, контролируемый износ и интенсивный износ.
- Случайные помехи в измерениях имеют нулевое среднее, несмотря на их неизвестную природу.
- В целях уменьшения влияния предыдущей стадии износа подшип-

ника на результаты работы алгоритма, будем обрабатывать только ряд последних измерений.

- Алгоритм SPS применяется для оценки двух параметров линейной модели: углового коэффициента  $k$  и вертикального смещения  $b$ . Результатом работы алгоритма является  $2D$  набор, состоящий из пар значений  $(k^*, b^*)$ , удовлетворяющие линейной модели с заданной вероятностью.

Поставленная проблема может быть сформулирована следующим образом. Дано  $N$  точек  $(y_i, t_i)$ , необходимо найти оптимальные параметры  $(k^*, b^*)$ , удовлетворяющие уравнению линейной модели:

$$y(t) = \phi(t) = kt + b$$

При этом модель наблюдения имеет вид:

$$y_i = \phi(t_i) + v_i, \quad i = 1, 2, \dots, N,$$

где  $v_i$  – симметричные случайные помехи.

Главная задача: вычислить при помощи метода SPS доверительный набор  $T$  для вектора  $(k, b)$ .

Для этого необходимо осуществить следующие шаги:

1. Построить  $2D$  сеть пробных значений целевых параметров  $(k^*, b^*)$  и далее провести каждую пару значений  $(k, b)$  через процедуру SPS-индикатора.
2. Выбрать значения параметров  $M$  и  $q$  таким образом, чтобы найденные величины  $(k^*, b^*)$  лежали в границах набора  $T$  (рис. 3) с вероятностью

$$Prob(k^*, b^*) \in T = 1 - \frac{q}{M}$$

3. Сгенерировать матрицу размера  $N \times (M - 1)$  случайных значений  $\beta_{i,j} = \pm 1$  с вероятностями

$$Prob(\beta_{i,j} = 1) = Prob(\beta_{i,j} = -1) = \frac{1}{2}$$

Следующие шаги алгоритма описывают принцип работы процедуры SPS-индикатора:

4. Для пробных значений  $(k, b)$  необходимо вычислить невязки:

$$\delta_i(k) = (y_i - \phi((k, b)t_i)), \quad i = 1, 2, \dots, N$$

5. Для каждого набора величин  $\beta_j$  вычислить знакопеременные суммы  $H_j$  и невозмущенную сумму  $H_0$ :

$$H_0((k, b)) = \sum_{i=1}^N \delta_i(k)$$

$$H_j((k, b)) = \sum_{i=1}^N \beta_{i,j} \delta_i(k), \quad j = 1, 2, \dots, M - 1$$

6. Упорядочить скалярные значения  $|H_j(k, b)|$  в порядке возрастания.
7. Вычислить ранг  $R((k, b))$ , являющийся порядковым номером  $|H_0(k, b)|$  в упорядоченном наборе  $|H_j(k, b)|$ , например  $R((k, b)) = 1$ , если  $|H_0(k, b)|$  наименьший,  $R((k, b)) = 2$ , если  $|H_0(k, b)|$  второй по величине и т.д.
8. Если  $R((k, b)) \leq M - q$  для пробных  $(k, b)$ , то эти значения включаются в результирующий доверительный набор  $T$ , в противном случае нет.

### 1.3. Схема работы алгоритма SPS

На рисунке 4 показана ожидаемая схема работы метода SPS для вероятности  $P$ . Каждая точка набора  $T$  может быть интерпретирована как искомое значение параметра  $(k^*, b^*)$  и, следовательно, может быть сопоставлена с соответствующей прямой на плоскости (на рисунке серые линии пунктиром), которые описывают искомое поведение модели.



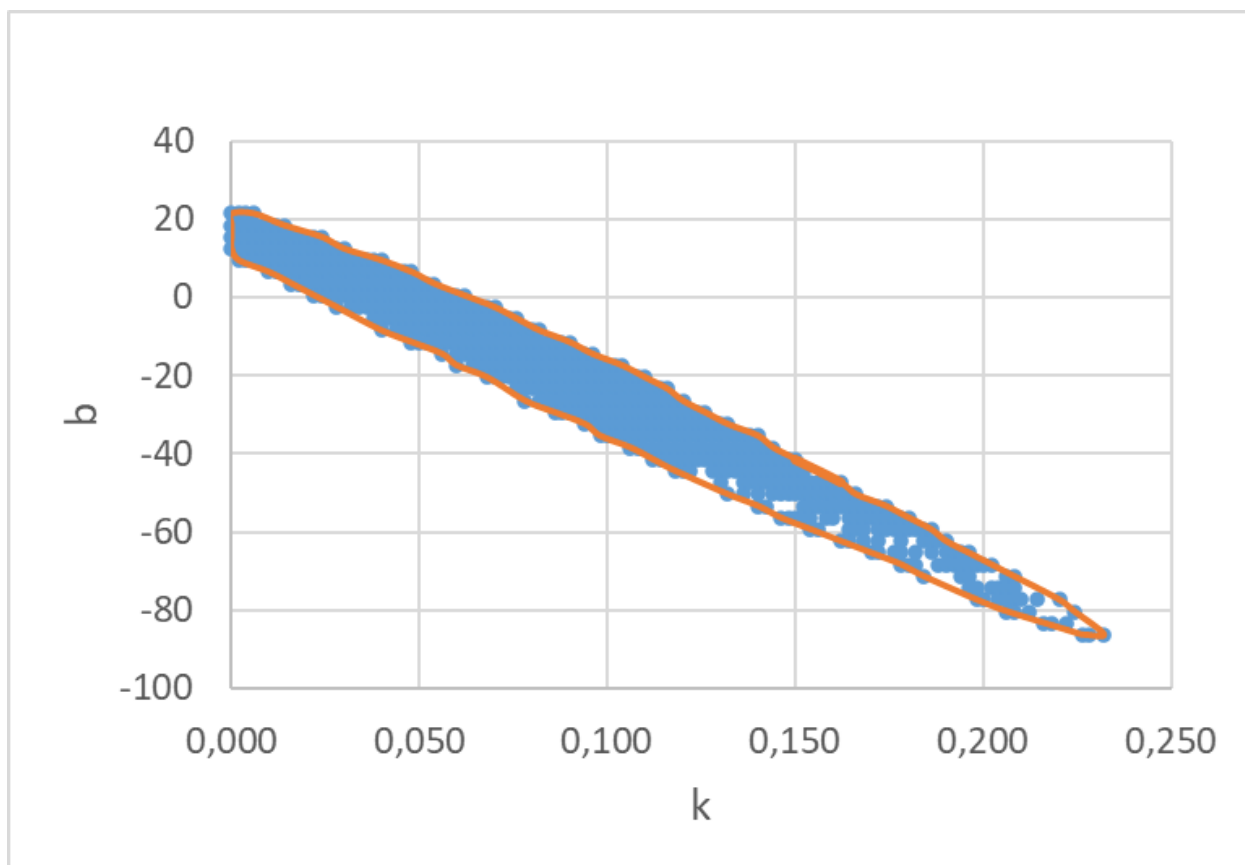


Рис. 3: Доверительный набор  $T$  с вероятностью 84%

Исходя из этого, точка пересечения данной прямой с линией порогового значения предсказывает момент времени выхода из строя подшипника. Более того, мы можем получить множество таких точек пересечения, используя все пары  $(k, b)$  из доверительного набора  $T$ . Данное множество можно назвать проекцией  $T$  на прямую порогового значения, чьи границы определяют минимальные и максимальные значения моментов времени выхода из строя подшипника. Зачастую максимальное значение  $t_{max}$  стремится к бесконечности и поэтому не дает полезных результатов. Самым важным значением является  $t_{min}$  (на рисунке ромбовидные маркеры), т.к. оно предсказывает момент времени после которого поломка подшипника может произойти в любое время. Другими словами, левая граница проекции множества  $T$  определяет возможный временной отрезок эксплуатации подшипника в нормальном режиме. Возьмем данное время как случайную величину, предсказывающую оставшийся период использования подшипника до его поломки. Соответствующая функция распределения (CDF) может быть численно рассчитана путем изменения доверительного уровня алгоритма SPS. Например, варьирование доверительной вероятности приводит к тому, что в результате своей работы метод SPS дает разные множества, и, следовательно, значения  $t_{min}$ .

В целях проверки ожидаемых результатов на практике, алгоритм SPS был реализован в программном коде.

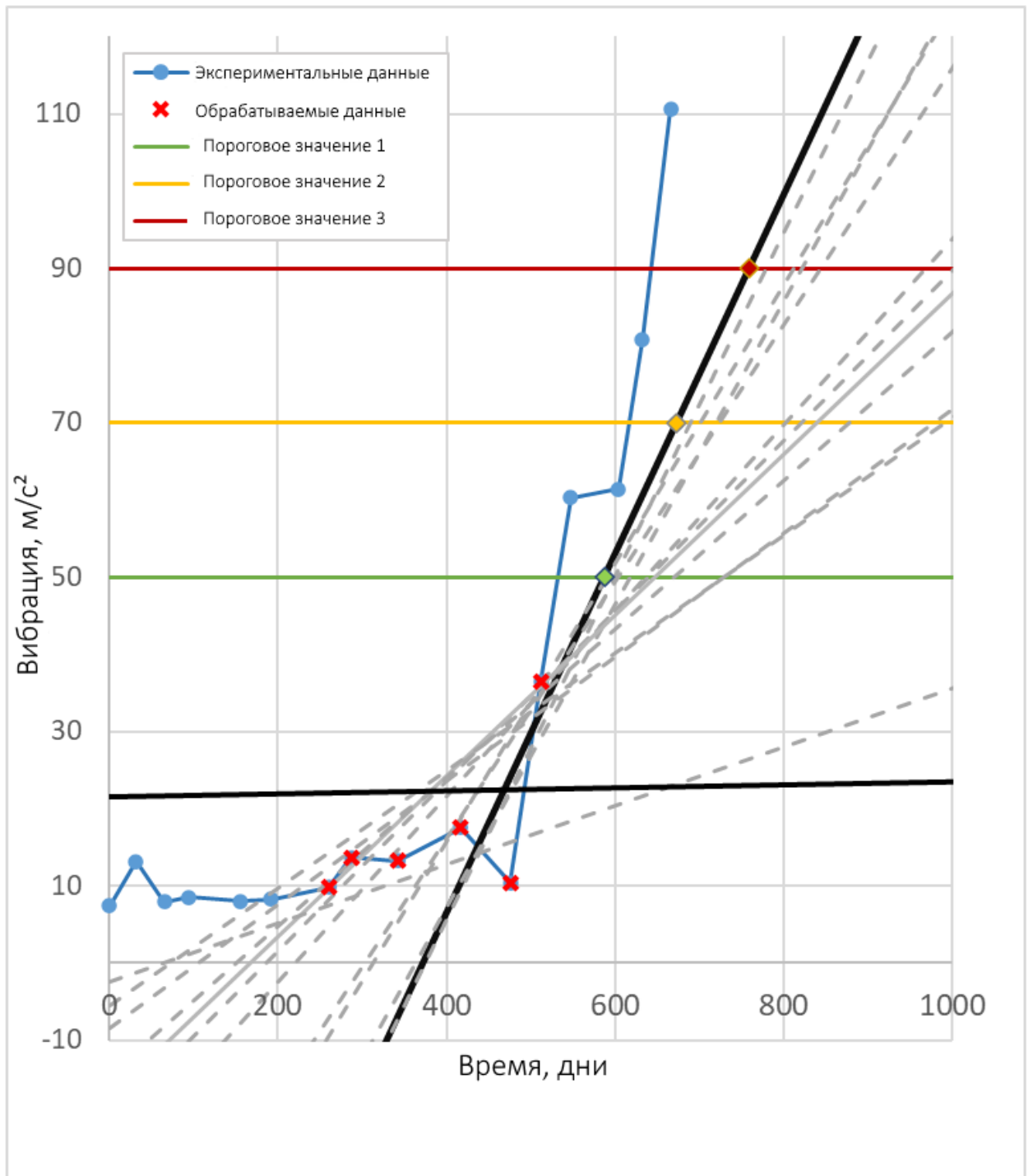


Рис. 4: Схема работы метода SPS

## 2. Реализация

### 2.1. Реализация алгоритма SPS

Для реализации алгоритма SPS был выбран язык Python, ввиду его удобства в создании методов решения для часто корректирующихся исследовательских задач, а также наличия простых в использовании, но содержательных пакетов для работы с математическими и алгоритмическими проблемами.

Алгоритм в своем математическом представлении изначально предполагает неограниченное количество пробных значений  $(k, b)$ . Разумеется, вычислительные машины обладают конечным ресурсом как по памяти, так и по вычислительным мощностям, поэтому алгоритм был в первую очередь адаптирован под конечное число пробных значений. Следующим этапом реализации алгоритма стала проверка соответствия практических результатов теоретическим. Для этого была произведена реализация алгоритма на чистом Python, ввиду простоты отладки и отслеживания процесса работы метода SPS. Был получен результат, совпадающий с теоретическими расчетами, однако длительность исполнения программы ожидаемо была неудовлетворительной, что вынудило искать пути оптимизации. В целях оптимизации адаптированный под вычислительные задачи алгоритм был сначала изменен с использованием матричных операций, а затем полностью переписан при помощи библиотек `numpy` и `scipy`, что позволило многократно повысить быстродействие прототипа ПО. Ключевым этапом работы алгоритма SPS является подсчет знакопеременных сумм и вычисление их ранга:

```
outer_prod = np.einsum('ij,jk->ikj', phi, phi.T)
R = 1 / self._num_points_sps * outer_prod.sum(axis=-1)
R_sqrt = np.real(scipy.linalg.sqrtm(R))
R_sqrt_inv = np.linalg.inv(R_sqrt)

predict = phi.T @ theta
epsilon = self._y_data_slice.T - predict
eps_shape = (self._num_points_sps, theta.shape[0], self._num_random_sets, self.
    ↪ _num_k_samples * self._num_b_samples)
```

```

epsilon = np.broadcast_to(epsilon[:, np.newaxis, np.newaxis, :], shape=
    ↪ eps_shape)
epsilon = np.swapaxes(epsilon, 0, 1)
broadcasted_mult = self._Beta[np.newaxis, :, :, np.newaxis] * phi[:, :, np.
    ↪ newaxis, np.newaxis]
residual = (1 / self._num_points_sps * (broadcasted_mult * epsilon).sum(axis=1)
    ↪ )
residual = np.swapaxes(residual, 0, 1)

sps_sums = R_sqrt_inv @ residual
normalized_sps_sums = np.linalg.norm(sps_sums, axis=1) ** 2

zero_sum = normalized_sps_sums[0, :]
rank = (normalized_sps_sums < zero_sum).sum(axis=-2)
equal_sums = (normalized_sps_sums == zero_sum).sum(axis=-2)
normalized_bias = np.trunc(np.random.uniform(0, equal_sums + 1))
rank = rank + normalized_bias

```

Более того, данная модификация обобщила реализацию, добавив возможность устанавливать в качестве предсказательной модели как линейную, так и другие функции. Таким образом была получена реализация алгоритма SPS для заданных параметров  $q$  и  $M$ , т.е. для заданной доверительной вероятности.

В ходе анализа результатов работы полученной реализации было выдвинуто предложение объединить полученные результаты для набора доверительных вероятностей в единую схему. Наглядным решением данной задачи стал цветной индикатор, для удобства представленный в виде перехода из зеленого в красный в соответствии с  $t_{min}$ , полученными в результате работы метода SPS на всем наборе доверительных вероятностей. Также для получения точных прогнозируемых значений моментов времени выхода подшипника из строя была построена функция распределения оставшегося времени  $t_{min}$ :

```

spl = scipy.interpolate.PchipInterpolator(t_min_unique,
    ↪ inv_confidence_levels)
cdf_x = np.arange(t_min_unique.min(), t_min_unique.max())
cdf_y = spl(cdf_x)
pdf = spl.derivative()
pdf_y = -pdf(cdf_x)
pdf_peaks, _ = scipy.signal.find_peaks(pdf_y)

```

Завершающим этапом реализации в данной работе стала необходимость

ввода и вывода множественного числа данных, необходимых для комфортного использования прототипа ПО. В целях повышения удобства работы и наглядности был разработан пользовательский интерфейс.

## 2.2. Реализация пользовательского интерфейса

Пользовательский интерфейс был разработан с использованием фреймворка Qt, в частности использовались библиотеки PyQt, позволяющие обращаться к функциональности фреймворка напрямую из языка Python. Данный фреймворк был выбран в первую очередь ввиду его кроссплатформенности, что делает возможным сборку прототипа ПО как под Windows, так и под Linux, без изменения кодовой базы. Другой причиной выбора данного фреймворка является его богатый функционал, позволивший реализовать удобное представление графиков и другие необходимые составляющие интерфейса прототипа ПО. На рис. 5 представлен скриншот реализованного пользовательского интерфейса прототипа ПО.

Интерфейс позволяет:

- Загружать .mat файлы с исходными данными ускорения вибрации подшипников,
- Задавать количество точек из исходных данных, используемых для расчета,
- Выбирать точку, после которой необходимо сделать прогноз оставшегося времени исправной работы подшипника,
- Задавать пороговые значения ускорения вибрации подшипника,
- Вводить интересующую пользователя временную точку и получать спрогнозированную вероятность ее достижения моделью при заданном пороговом значении,
- Выводить спрогнозированные временные точки, соответствующие вероятностям в 50% и 90%, для заданных пороговых значений,

- Выводить графики с предсказанными значениями оставшегося времени исправной работы подшипника в виде цветного индикатора на первом графике, функции распределения на втором, и плотности распределения на третьем.

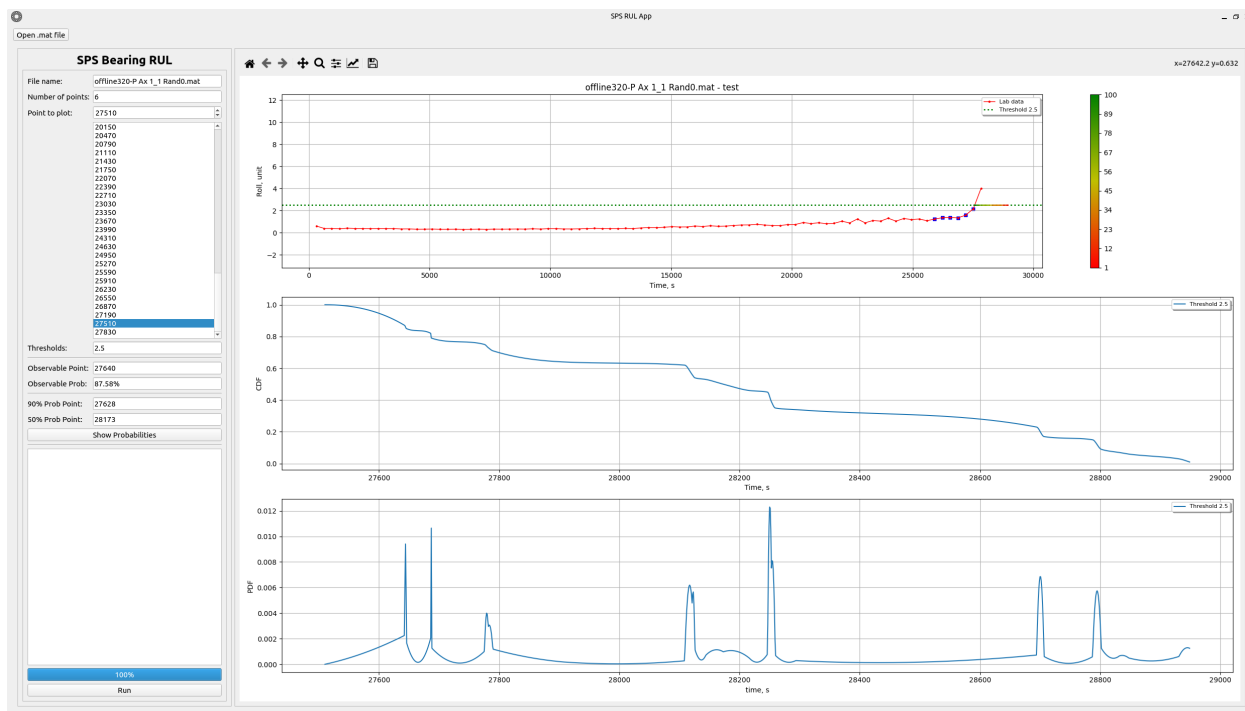


Рис. 5: Скриншот пользовательского интерфейса прототипа ПО

### 3. Результаты

В ходе данной работы были получены следующие результаты:

- Разработан прототип системы с реализацией интерфейса для наглядного представления пользователю информации о прогнозах
- Реализован алгоритм прогнозирования SPS на примере предсказания выхода из строя шариковых подшипников
- Получены результаты работы алгоритма с лабораторными и промышленными данными



## Список литературы

- [1] Behzad Mehdi, Arghand Hesam Addin, and Rohani Bastami Abbas. Remaining useful life prediction of ball-bearings based on high-frequency vibration features // Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science. — 2018. — Vol. 232, no. 18. — P. 3224–3234.
- [2] Csáji Balázs Csanád, Campi Marco Claudio, and Weyer Erik. Sign-perturbed sums: A new system identification approach for constructing exact non-asymptotic confidence regions in linear regression models // IEEE Transactions on Signal Processing. — 2014. — Vol. 63, no. 1. — P. 169–181.
- [3] Lei Yaguo, Li Naipeng, Guo Liang, Li Ningbo, Yan Tao, and Lin Jing. Machinery health prognostics: A systematic review from data acquisition to RUL prediction // Mechanical systems and signal processing. — 2018. — Vol. 104. — P. 799–834.
- [4] Volkova MV, Granichin ON, Volkov GA, and Petrov Yu V. On the possibility of using the method of sign-perturbed sums for the processing of dynamic test data // Vestnik St. Petersburg University, Mathematics. — 2018. — Vol. 51, no. 1. — P. 23–30.
- [5] Granichin Nikolay, Volkov Grigory, Petrov Yuri, and Volkova Marina. Randomized approach to determine dynamic strength of ice // Cybernetics and Physics. — 2021. — Vol. 10, no. 3. — P. 122–126.
- [6] Rezaeianjouybari Behnoush and Shang Yi. Deep learning for prognostics and health management: State of the art, challenges, and opportunities // Measurement. — 2020. — Vol. 163. — P. 107929.
- [7] Saon Sharifah, Hiyama Takashi, et al. Predicting remaining useful life of rotating machinery based artificial neural network // Computers & Mathematics with Applications. — 2010. — Vol. 60, no. 4. — P. 1078–1087.

- [8] Volkov Grigory and Smirnov Ivan. A probabilistic approach to evaluate dynamic and static strength of quasi-brittle materials through high-rate testing // International Journal of Mechanical Sciences. — 2022. — Vol. 216. — P. 106960.