

## **СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОГНОЗИРОВАНИЯ ПОЛОМОК ДВИГАТЕЛЯ ЛОКОМОТИВА**

**Котов В.А.**, студент

**Романенко В.В.**, старший преподаватель

УО «Белорусский государственный университет транспорта»,  
г. Гомель, Беларусь

**Аннотация.** В данной статье описывается возможность применения математических методов для эффективной диагностики и управления локомотивов в транспортной железнодорожной инфраструктуре.

**Ключевые слова:** подвижной состав, поломки двигателя, программное обеспечение.

В настоящее время на железнодорожном транспорте значительно возрастает внимание к обеспечению высокой эксплуатационной надежности подвижного состава и безопасности движения поездов. Это является ключевым аспектом для повышения эффективности и качества работы железнодорожного сектора.

В то же время наблюдается устойчивый рост стремления к максимально рациональному использованию ресурсов железнодорожного транспорта. Это, в свою очередь, приводит к увеличению длинных участков безостановочного движения поездов, росту скорости и увеличению нагрузки на ось. В связи с этим особенно актуальным становится сбор информации о состоянии вагонных букс, в частности о перегреваемых буксовых подшипниках. Их перегрев может привести к повреждению шейки оси колесной пары, а также к возгоранию вагонов, грузов и объектов инфраструктуры [1].

Одним из важных свойств сложных технических объектов является надежность компонентов системы, включая датчики, оборудование и программное обеспечение, должны работать стабильно и предсказуемо.

Автоматизация системы сбора, анализа и мониторинга данных работы двигателя локомотива, требуют комплексного подхода, включающего разработку программного обеспечения. Вот пример кода, который может быть использован для автоматизации такой системы на языке Python:

Код представляющий базовую структуру обработки данных:

```
# Импорт необходимых библиотек

import sensor_data_processor

import predictive_maintenance

import communication_module

# Класс для обработки данных с датчиков

class SensorDataHandler:

    def __init__(self):
```

```
self.data = []
```

```
def collect_data(self):
```

```
# Сбор данных с датчиков
```

```
self.data = sensor_data_processor.collect()
```

```
def analyze_data(self):
```

```
# Анализ данных на предмет аномалий
```

```
anomalies = sensor_data_processor.analyze(self.data)
```

```
return anomalies
```

```
# Класс для предиктивного обслуживания
```

```
class MaintenancePredictor:
```

```
def __init__(self):
```

```
self.predictions = []
```

```
def predict_failures(self, data):
```

```
# Предсказание потенциальных неисправностей
```

```
self.predictions = predictive_maintenance.predict(data)
```

```
def schedule_maintenance(self):
```

```
# Планирование технического обслуживания
```

```
    maintenance_schedule =  
predictive_maintenance.schedule(self.predictions)
```

```
return maintenance_schedule
```

```
# Основной цикл системы мониторинга
```

```
def main_monitoring_loop():
```

```
sensor_handler = SensorDataHandler()
```

```
maintenance_predictor = MaintenancePredictor()

while True:

    sensor_handler.collect_data()

    anomalies = sensor_handler.analyze_data()

    if anomalies:

        communication_module.send_alert(anomalies)

        maintenance_predictions =
maintenance_predictor.predict_failures(sensor_handler.data)

        maintenance_schedule =
maintenance_predictor.schedule_maintenance()

        if maintenance_schedule:

            communication_module.send_maintenance_schedule(maintenance_s
chedule)

# Запуск системы
```

```
if __name__ == "__main__":
```

```
    main_monitoring_loop()
```

С помощью этого кода можно создать систему, которая может быть адаптирована под конкретные датчики и условия эксплуатации двигателя локомотива, например:

- выявлять и предотвращать потенциальные неисправности: анализ данных с датчиков может помочь определить элементы двигателя, которые могут выйти из строя;
- планировать техническое обслуживание: системы могут анализировать данные о работе двигателя и рекомендовать оптимальные сроки для проведения технического обслуживания, чтобы минимизировать риск внезапных отказов;
- оптимизировать использование ресурсов: предиктивное обслуживание помогает определить, когда и какие запасные части потребуются, ресурс включает в себя сбор данных с датчиков.

Система представляет собой комплексное решение, направленное на повышение эффективности, безопасности и надежности двигателя локомотива. Интеграция современных технологий, таких как сбор данных с датчиков, их анализ, предиктивное обслуживание позволяет оперативно реагировать на возможные неисправности двигателя локомотива и оптимизировать расписание технического обслуживания.

### Литература

1. Бурченков В. В. Автоматизированные системы контроля подвижного состава : Учеб. пособие. Гомель : БелГУТ, 2020 – 226 с.