Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

ropod

# ”Analysis mode!”: Robot (remote) monitoring, inspection, and diagnosis as prerequisites for deploying robots to everyday environments

## IROS18 workshop on robotics for logistics in warehouses and environments shared with humans

**Alex Mitrevski**, Santosh Thoduka,
Paul G. Plöger, Erwin Prassler
Autonomous Systems Group

October 05, 2018

# Motivation



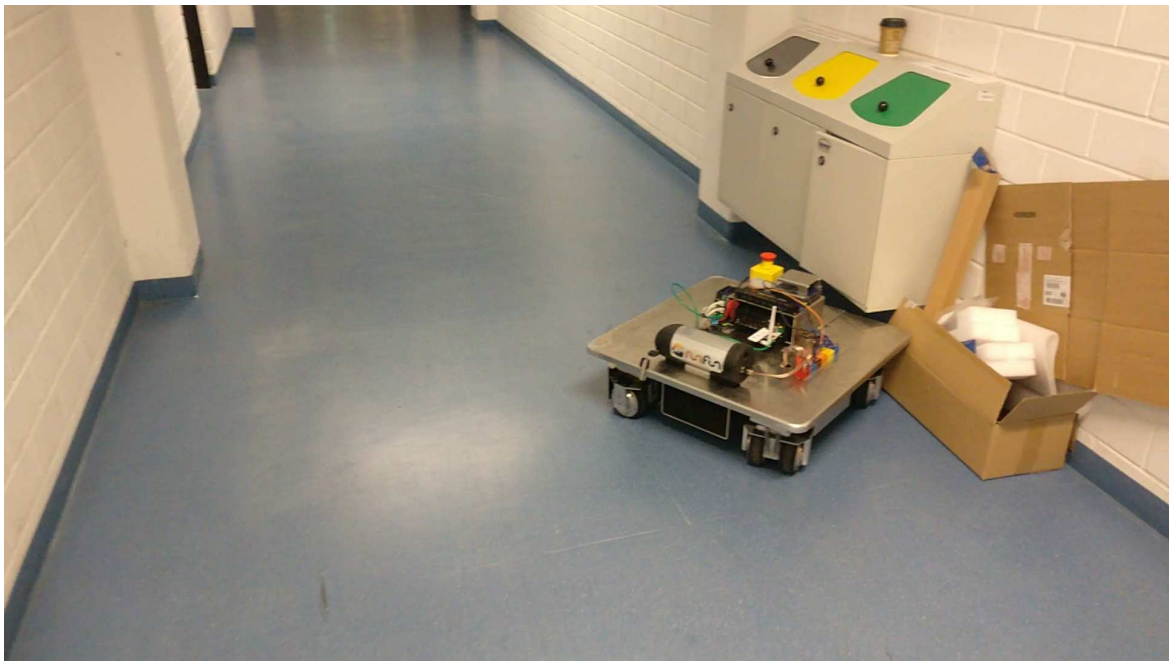
This talk is about the unfortunate fact that robots sometimes **deviate** from their **expected** behaviour

The ability to **detect** and **analyse** such deviations is essential for practical robot deployment

Detection and diagnosis require (i) component/system **observers** and (ii) some form of **introspection**

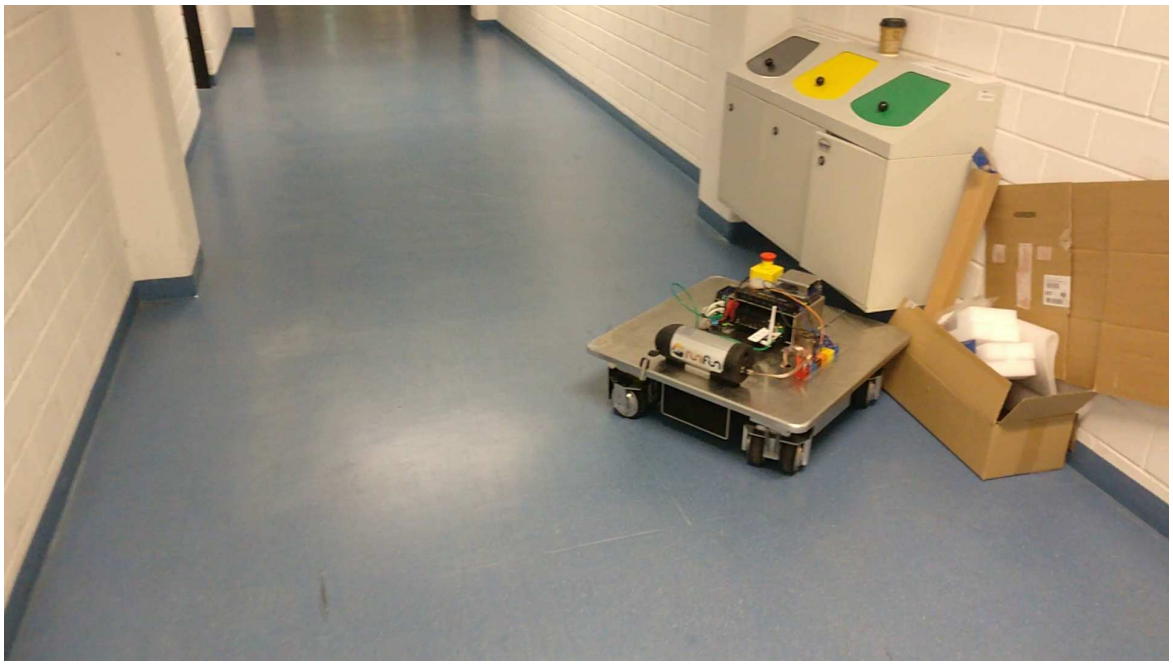
# Context





# Context

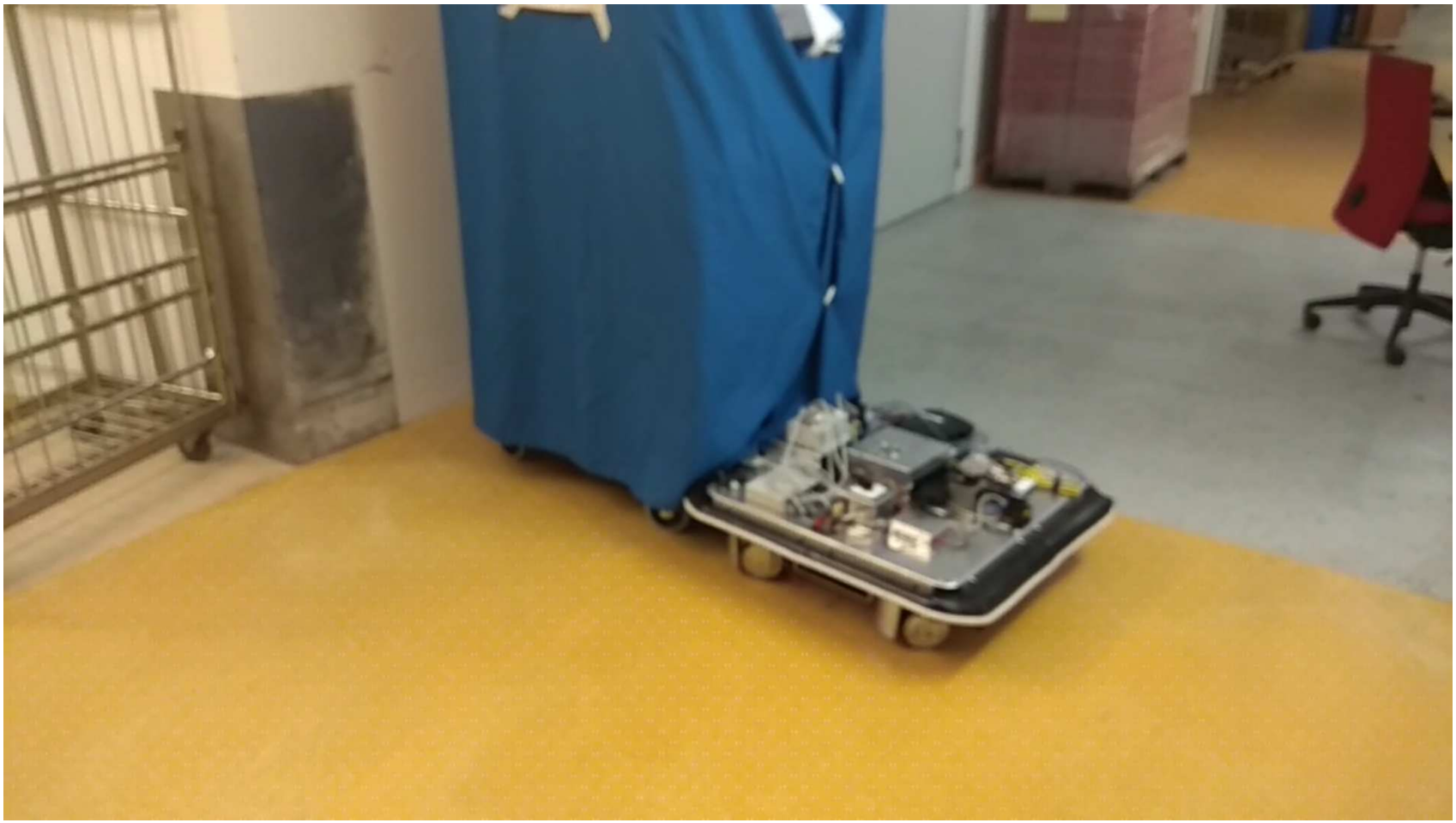






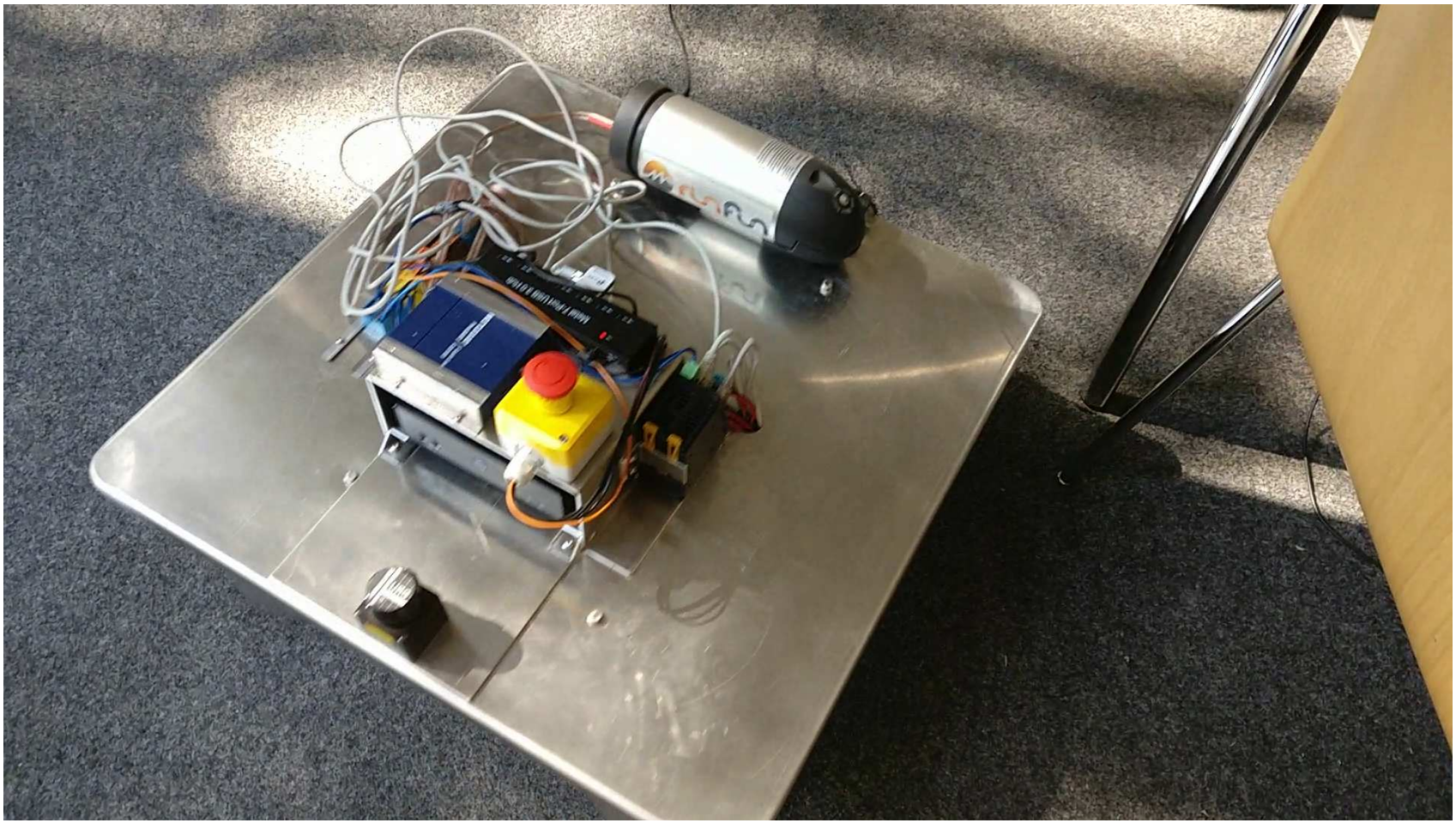
# Have I Parked Properly?



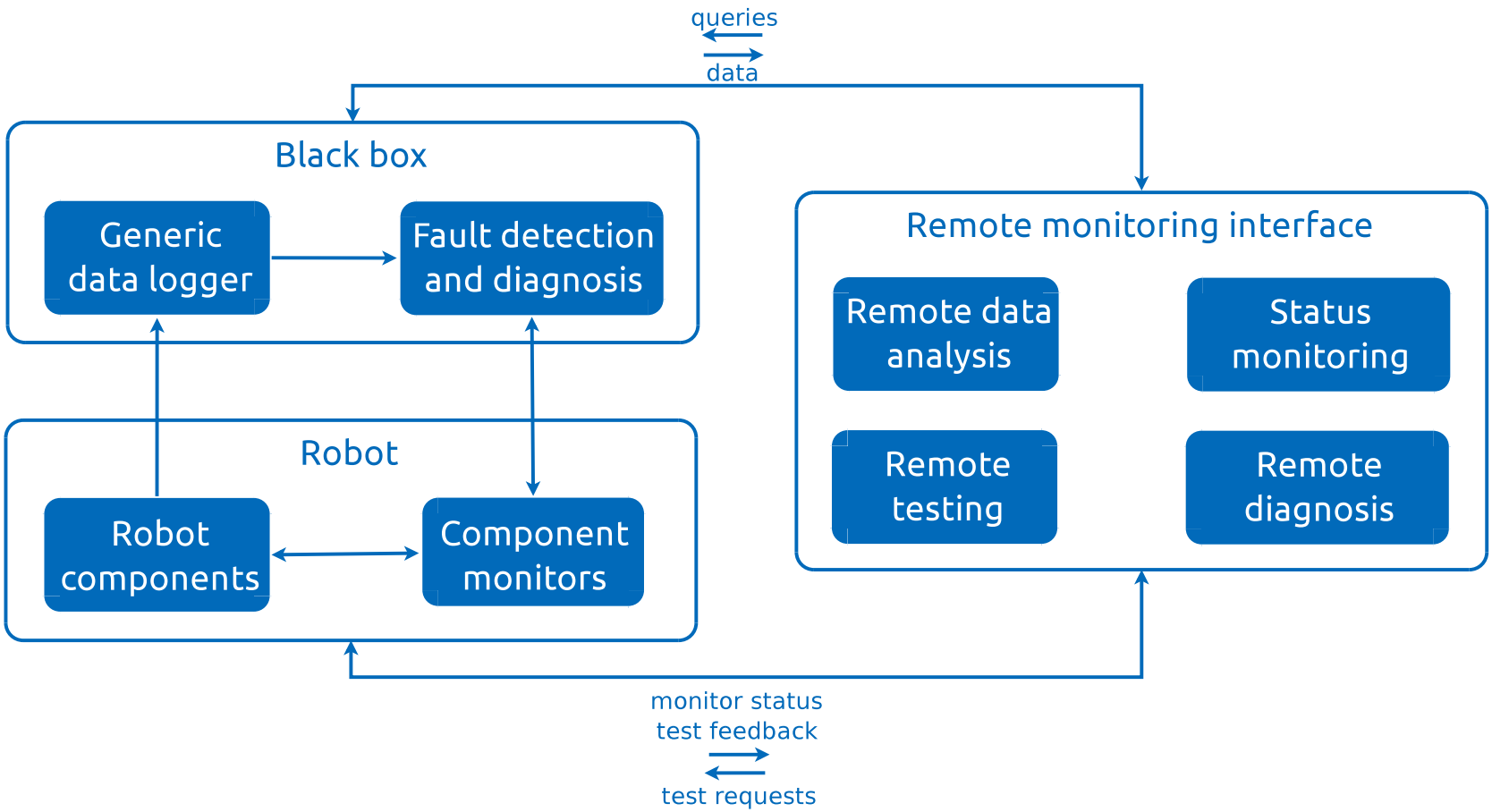


# Is My Motion Correct?





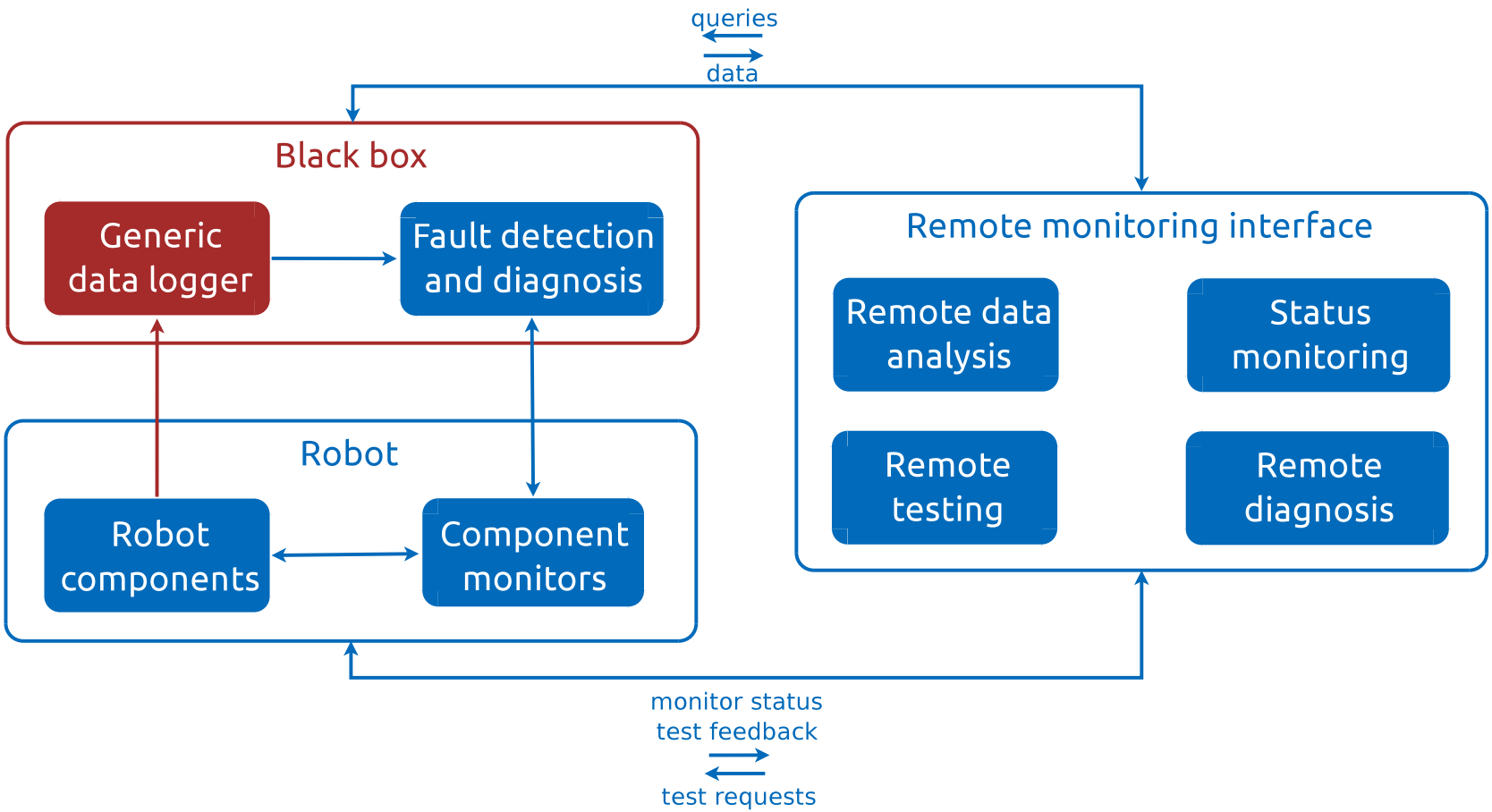
# Our Approach



# Generic Data Logger for Robots

Hochschule Bonn-Rhein-Sieg
University of Applied Sciences

ropod



# Robotic Black Box



The robotic black box operates as an **independent embedded component** that:

- **listens** to various data sources on a robot and
- **logs** data in a predefined format

# Robotic Black Box



The robotic black box operates as an **independent embedded component** that:

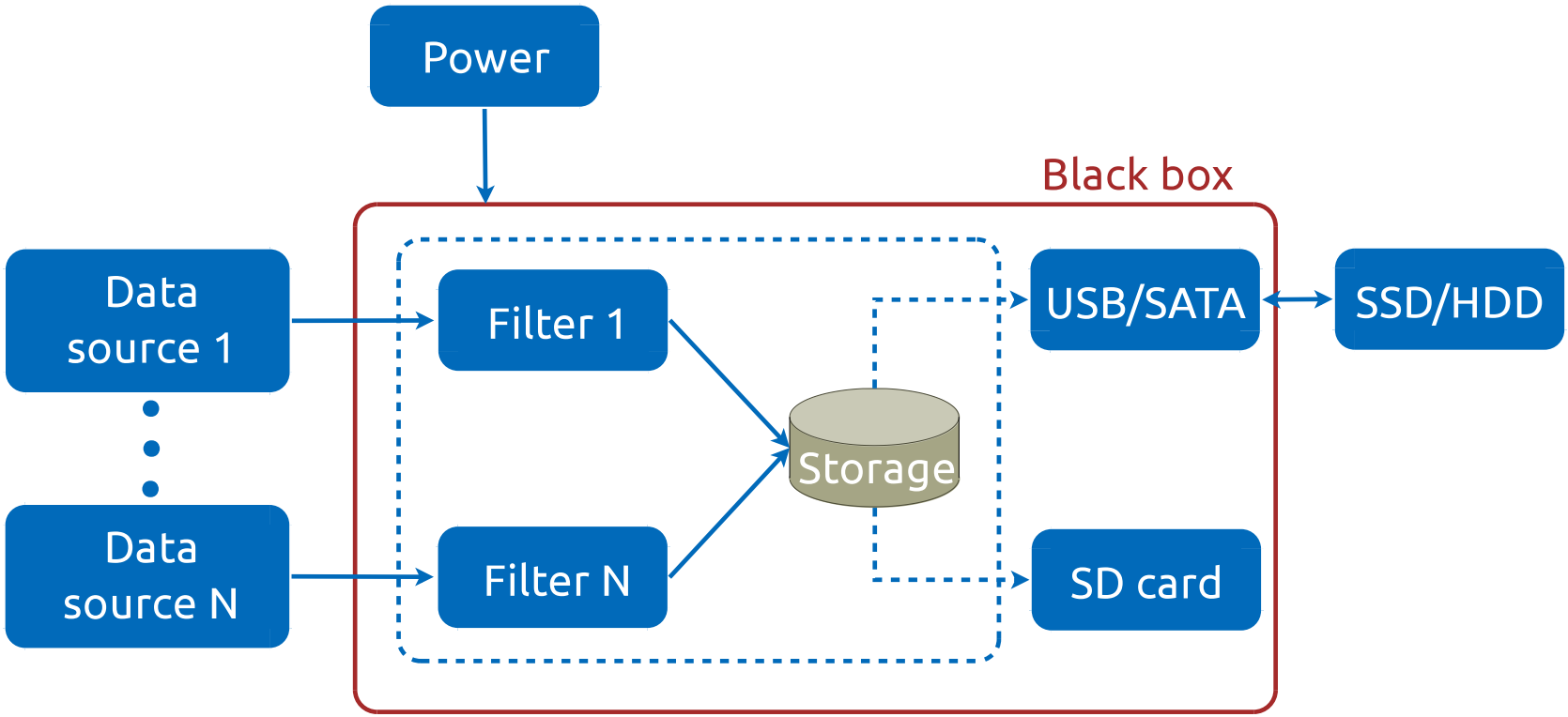
- **listens** to various data sources on a robot and
- **logs** data in a predefined format

Black box design principles:

- **data filtering**
- converting the data into a **standard format**
- **reconfigurability** for different communication interfaces and formats

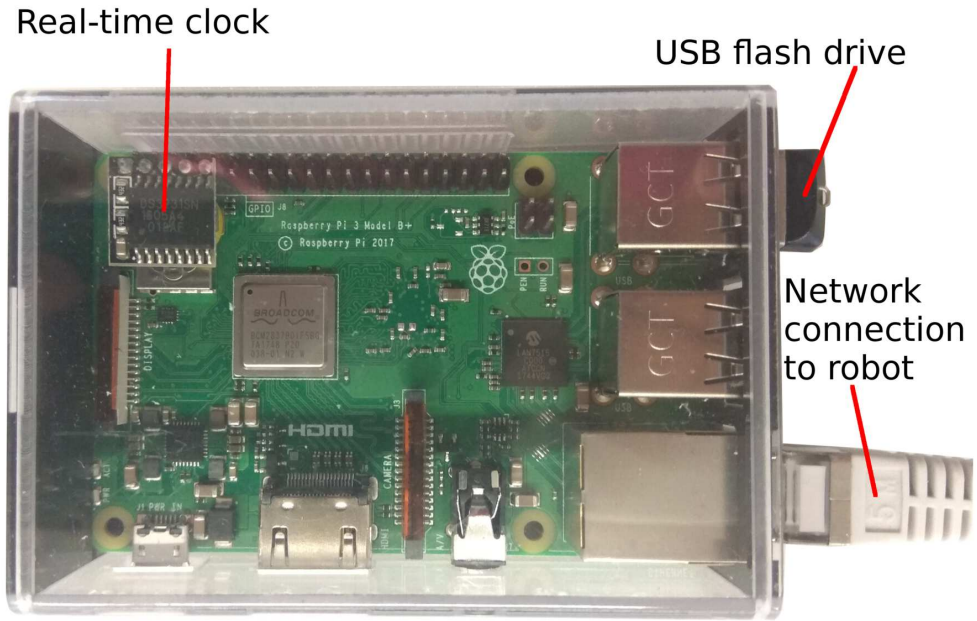
# Black Box Concept





# Black Box Prototype






- Raspberry Pi 3B+ with 64-bit Arch Linux Arm
- 128 GB flash drive
- A real-time clock module
- A MongoDB database for storing the data in JSON format
- Plug-and-play operation after initial setup
- Source code freely available at
  `https://github.com/ropod-project/black-box`

# The Black Box Toolkit



Black box data are not so useful without appropriate processing and analysis tools

Tools are needed for:

- Easy data access
- Data plotting
- Component and system monitoring
- Fault diagnosis

# Black Box Code Generation

Hochschule Bonn-Rhein-Sieg
University of Applied Sciences

ropod

```
{
    "_id" : "0",
    "angular" :
    {
        "x" : 0.0,
        "y" : 0.0,
        "z" : 0.0
    },
    "linear" :
    {
        "x" : 1.0,
        "y" : 0.0,
        "z" : 0.0
    },
    "timestamp" :
         1538251490.494
}
```

```
# This file has been autogenerated by black_box_orm
import pymongo as pm

class RosCmdVel(object):
    db_name = "logs"
    collection_name = "ros_cmd_vel"

    def __init__(self):
        self.variables = ['timestamp', '_id']
        self._id = "5bafdae2805b912676139ab2"
        self.variable_mappings = {'timestamp': '["
            timestamp"]', '_id': '["_id"]'}
        self.objects = ['linear', 'angular']
        self.timestamp = 1538251490.494
        self.linear = Linear()
        self.angular = Angular()
        self.mongo_client = pm.MongoClient()

    @staticmethod
    def get_all():
        client = pm.MongoClient()
        db = client[RosCmdVel.db_name]
        collection = db[RosCmdVel.collection_name]
        cursor = collection.find()
        data = list()
        for doc in cursor:
            obj = RosCmdVel()
            obj._id = doc["_id"]
            obj.timestamp = doc["timestamp"]
```

# Black Box Code Generation

Hochschule Bonn-Rhein-Sieg
University of Applied Sciences

ropod

```
{
    "_id" : "0",
    "angular" :
    {
        "x" : 0.0,
        "y" : 0.0,
        "z" : 0.0
    },
    "linear" :
    {
        "x" : 1.0,
        "y" : 0.0,
        "z" : 0.0
    },
    "timestamp" :
         1538251490.494
}
```

```
            obj.linear.from_doc(doc)
            obj.angular.from_doc(doc)
            data.append(obj)
        return data

    def from_doc(self, doc):
        self._id = doc["_id"]
        self.timestamp = doc["timestamp"]
        self.linear.from_doc(doc)
        self.angular.from_doc(doc)

definitions continue...
```

# Smart Wheel

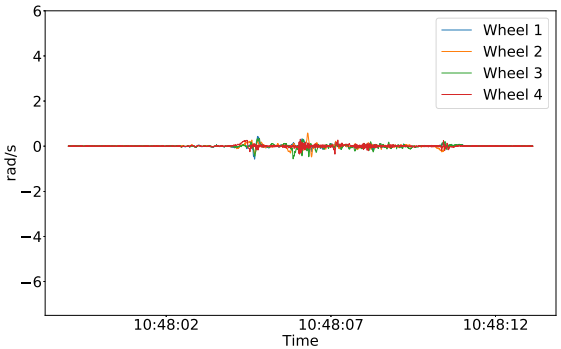


# Using Redundancies to Detect Faults

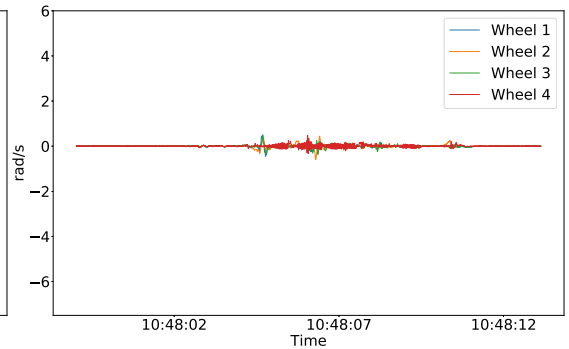


- ROPOD's smart wheel gives us access to **40+ state variables** - plenty of opportunities for exploring redundancy relations
  - three encoder measurements
  - IMU
  - voltage and current measurements
  - temperature
- Component monitors can exploit these redundancies for consistency checking
- Robot fleets: an additional level of redundancy!

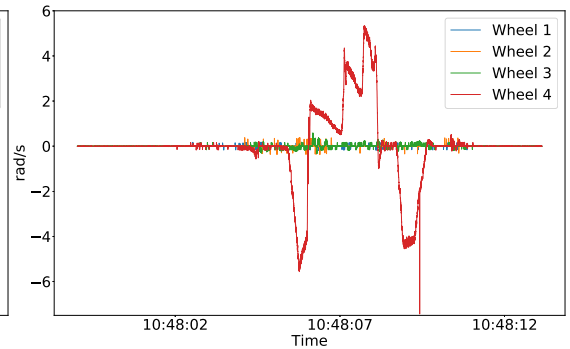
# Exploiting Redundancies



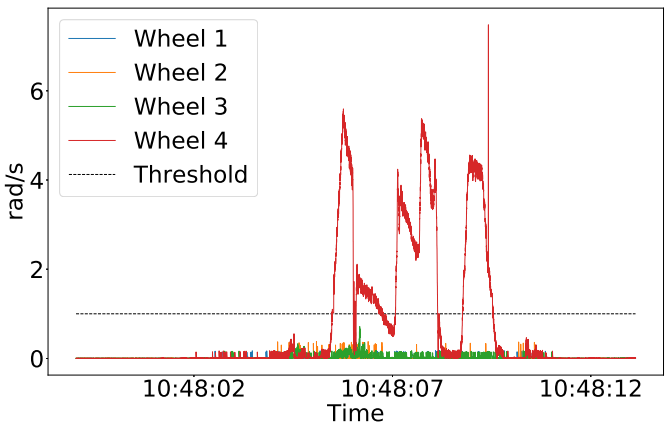

(a) Encoder 1 velocity



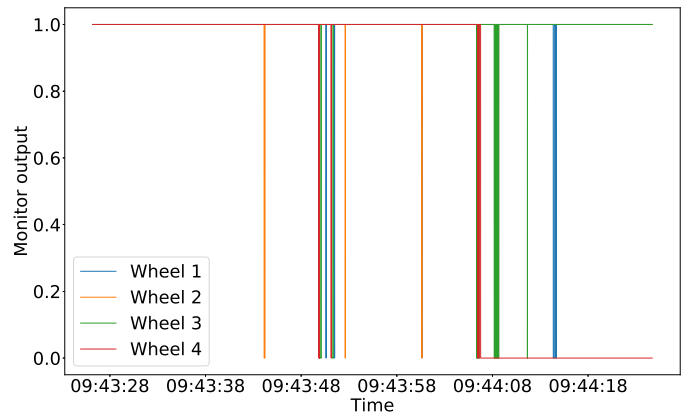

(b) Encoder 2 velocity




(c) Pivot encoder velocity



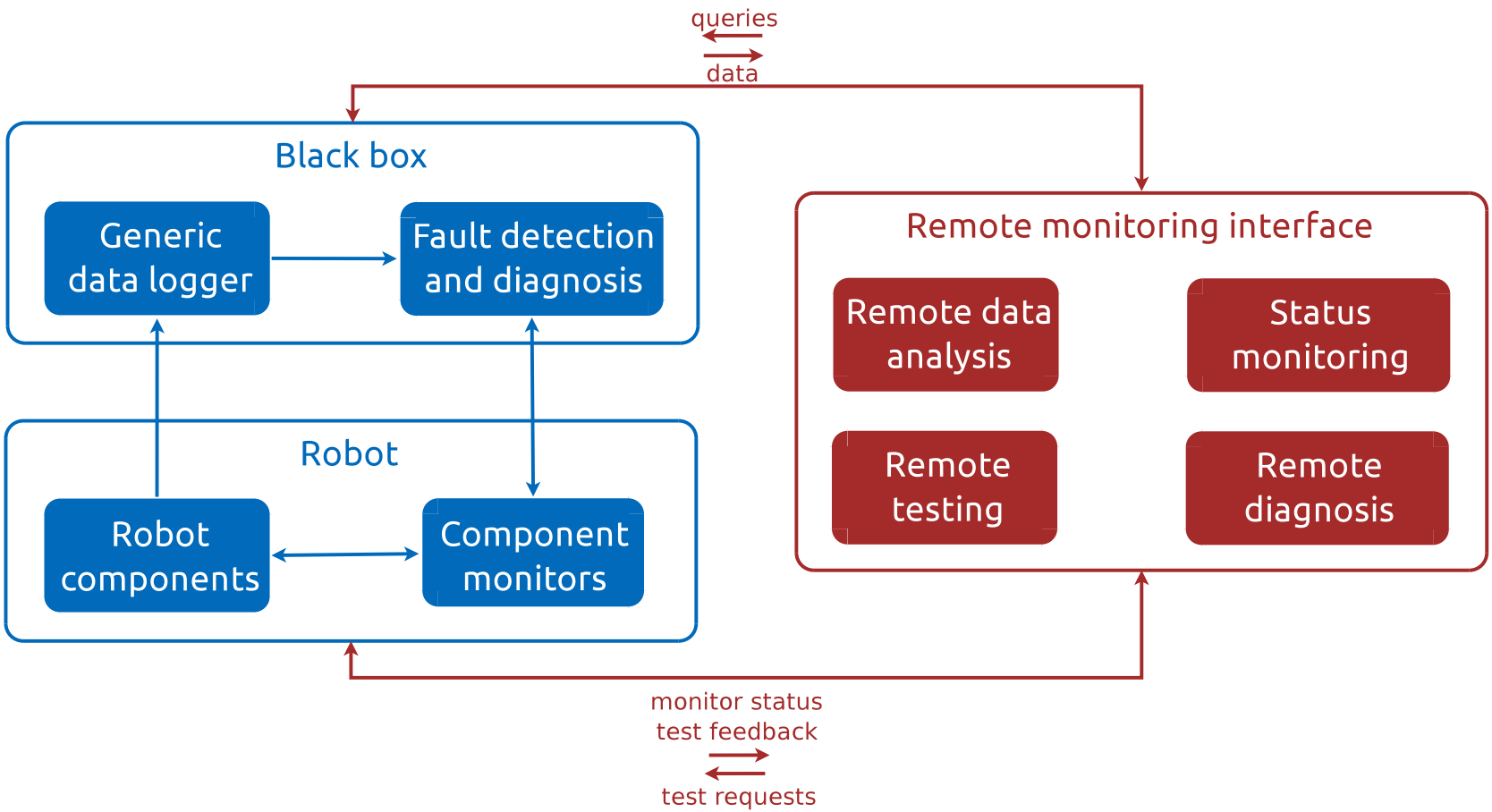

(d) Residual




(e) Monitor output

# Remote Fault Diagnosis





# Remote Data Analysis

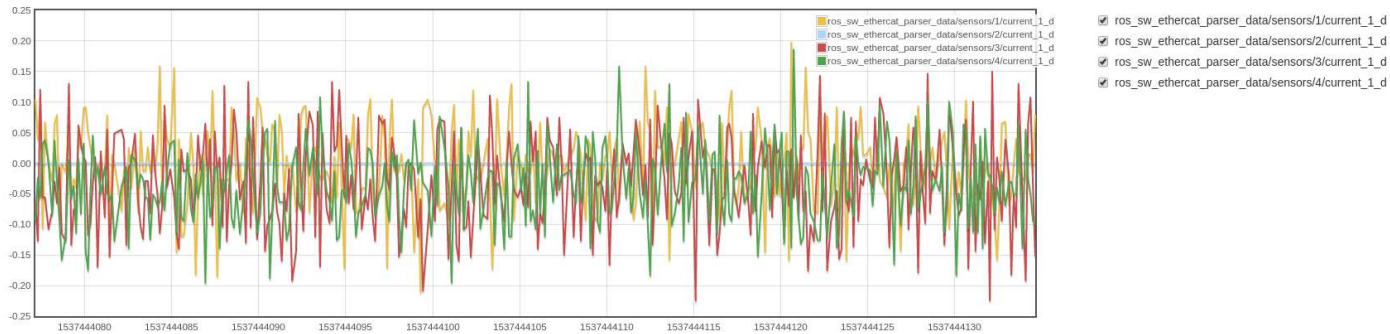
Hochschule Bonn-Rhein-Sieg
University of Applied Sciences

ropod

## ROPOD Remote Monitoring

black_box_0 | 2018/09/16 09:22 | 2018/09/24 09:22 | Display data | Download data

+ ros_cmd_vel
+ ros_sw_ethercat_parser_data




---

## Ropod info

| Ropod | Status | Status time | Detailed status |
|---|---|---|---|
| component_monitor | ● | 2018-09-22T09:57:43.437415 | Get detailed status |

| Status of component_monitor | Expand all | Collapse all | Hide detailed status |
|---|---|---|---|
| **Wireless network** | | | ● |
| Network quality and strength | | | ● |
| **Laser scanner** | | | ● |
| Laser devices recognised by OS | | | ● |
| **Encoder** | | | ● |
| Matching between the measured and differentiated encoder velocities | | | ● |
| Matching between the expected and measured pivot angular velocity | | | ● |
| **ROS** | | | ● |
| ROS master running | | | ● |
| Topics published | | | ● |

# Additional Diagnosis Aids



- Models of behaviour[1]
- Correlations between sensor measurements[2]
- Event tagging
- Status LEDs
- Intent communication

---

[1] M. Fox, J. Gough, and D. Long, ”Detecting Execution Failures Using Learned Action Models,” in *Proc. 22nd Nat. Conf. Artificial Intelligence - Vol. 2*, pp. 968-973, 2007.

[2] E. Khalastchi and M. Kalech, ”A sensor-based approach for fault detection and diagnosis for robotic systems,” *Autonomous Robots*, vol. 42, no. 6, pp. 1231-1248, Dec. 2017.

# Lessons Learned



- The black box has proven useful for debugging and performing diagnostics, but tools for easy data access are essential
- Communication is a significant challenge for performing remote monitoring and requires various practical considerations (e.g. network connectivity and data volume)
- User profiling is quite important for creating effective (remote) monitoring and diagnosis tools
- Often overlooked: Failure tolerance is just as important for software as it is for hardware

# Thanks for Your Attention