Deploying Robots in Everyday Environments: Towards Dependable and Practical Robotic Systems

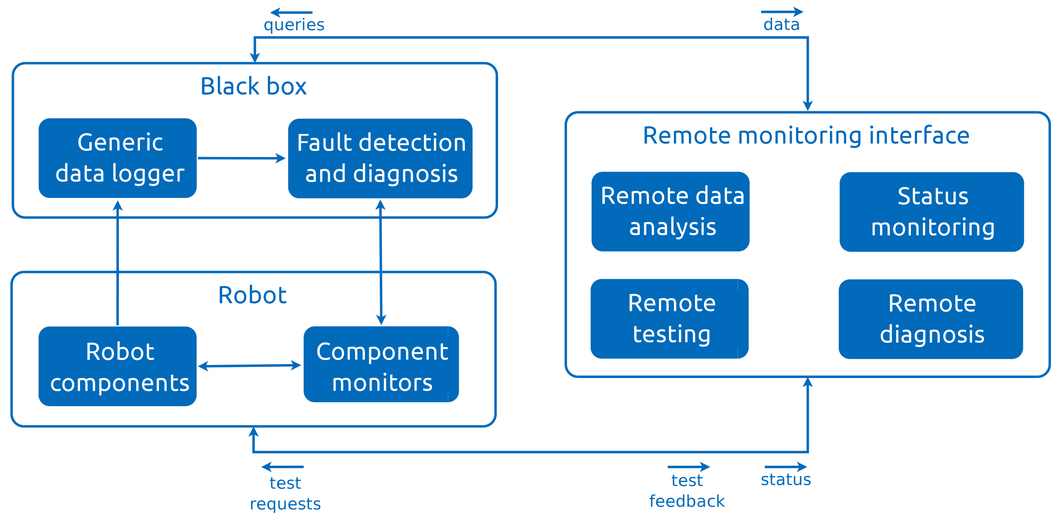
Alex Mitrevski, Santosh Thoduka, Argentina Ortega, Maximilian Schöbel, Patrick Nagel, Paul G. Plöger, Erwin Prassler

Motivation and Overall Idea

In ROPOD, a Horizon 2020 project, we are designing and developing wheeled robots to transport items such as trolleys and beds in a hospital. The deployment of robots in realistic dynamic environments is however challenging since they cannot always deal with failures in a reasonable manner. Manual diagnosis of failures is also a challenge since an on-site technician may not always be available. Strategies for automatic fault detection and diagnosis as well as remote operation monitoring are thus of utmost importance for smooth operation and error analysis.

Our work focuses on three main aspects related to this problem:

- **robotic black box**
- **remote monitoring**
- **fault detection and diagnosis**



Robotic Black Box

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

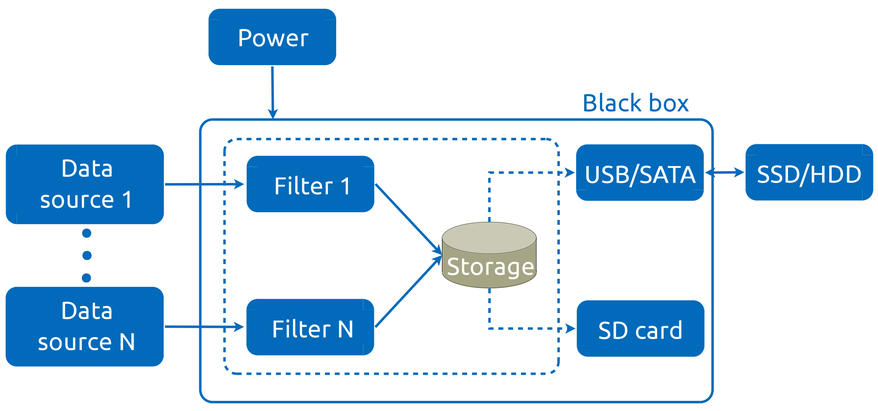
- **listens** to various data sources on a robot and
- **logs** data in a predefined format

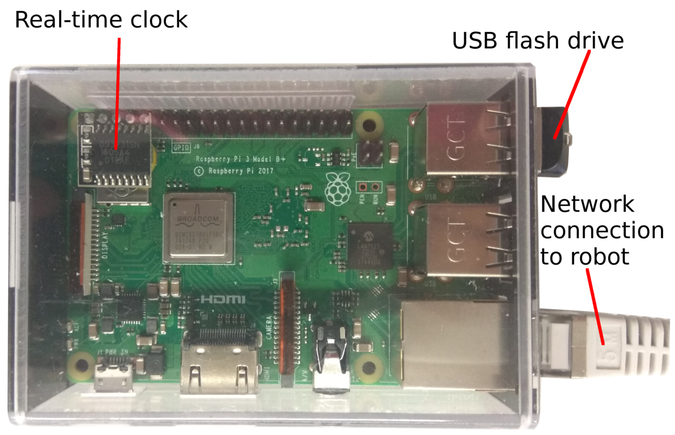
Black box design principles:

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

Current 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
- plug-and-play operation after initial setup





(Remote) Monitoring, Diagnosis, and Testing

Listing 1: Monitor configuration format

```
name: <string> | required
modes: <string>[] | required
dependencies: <string>[] | optional
```

Listing 2: Monitor mode configuration format

```
name: <string> | required
mappings:
    - mapping:
        inputs: <string>[] | required
        outputs: | required
            - output:
                name: <string> | required
                type: <string> | required
                expected: <bool> | <string> | <int> | <double> | optional
arguments: | optional
    - arg:
        name: <string> | required
        value: <bool> | <string> | <int> | <double> | required
```

Our fault diagnosis concept combines three aspects: **component monitoring**, **system-level diagnosis**, and **remote monitoring**.

Component monitors: Monitors are modelled as component-specific functions; a configuration file specifies the inputs and the outputs of the monitor, including arguments such as thresholds.

System-level diagnosis: A Failure Mode and Effects Analysis (FMEA) allows creating failure classes, rules, and models for rule-based diagnosis and structural analysis. Based on this, an expert system associates symptoms to a set of known effects on the components.

Remote monitoring and testing: Diagnosing problems without direct access to robots is of paramount importance for practical robot deployment. We are thus developing a remote monitoring interface that:

- **allows black box data access** (visualisation and offline download)
- **visualises the component monitor status**
- **provides remote experimentation and unit testing functionalities**, including testing procedures defined using the Business Process Model and Notation (BPMN) as it is targeted at non-skilled robot users

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N^{O}731848 ROPOD.

Contact

{aleksandar.mitrevski, santosh.thoduka, paul.ploeger, erwin.prassler}@h-brs.de

Grantham-Allee 20
53757 Sankt Augustin

Hochschule Bonn-Rhein-Sieg
University of Applied Sciences