"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.
Have I Parked Properly?
Is My Motion Correct?
Our Approach

Generic data logger → Fault detection and diagnosis → Black box

Robot components → Component monitors → Robot

queries

Remote monitoring interface
- Remote data analysis
- Remote testing
- Status monitoring
- Remote diagnosis

monitor status

test feedback

test requests
Generic Data Logger for Robots

Diagram:
- Black box
  - Generic data logger
  - Fault detection and diagnosis
- Robot
  - Robot components
  - Component monitors
- Remote monitoring interface
  - Remote data analysis
  - Status monitoring
  - Remote testing
  - Remote diagnosis

Data flow:
- Queries to data
- Monitor status test feedback
- Test requests
The robotic black box operates as an \textit{independent embedded component} that:

- \textbf{listens} to various data sources on a robot and
- \textbf{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 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
```python
# 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()

    @classmethod
    def get_all(self):
        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']
```
```python
{
    "_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
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 Data Analysis

ROPOD Remote Monitoring

Ropod info

Mitrevski, Thoduka, Plöger, Prassler  Robot (remote) monitoring, inspection, and diagnosis
Additional Diagnosis Aids

- Models of behaviour\(^1\)
- Correlations between sensor measurements\(^2\)
- Event tagging
- Status LEDs
- Intent communication


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