

# **Execution Monitoring for Long-Term Autonomous Plant Observation with a Mobile Robot**

---

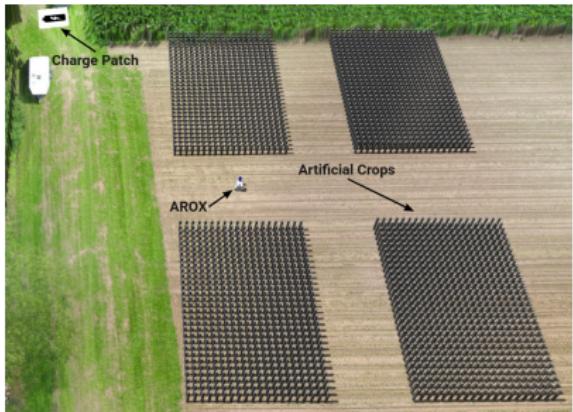
Tim Bohne

18. März 2022

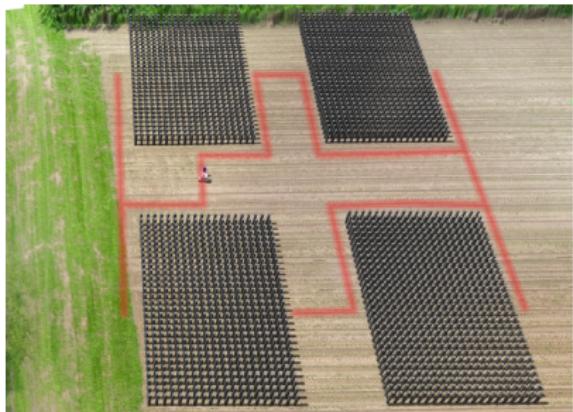
*AG Knowledge-Based Systems  
DFKI Plan-Based Robot Control Group*

- ① Szenario
- ② Langzeit-autonome mobile Roboter
- ③ Herausforderungen für Langzeitautonomie
- ④ Ausführungsüberwachung
- ⑤ Experimente / Evaluation
- ⑥ Fazit / Ausblick

# Szenario in der Simulation



PROTOTYPISCHES LAYOUT



BEISPIEL-SCAN-ROUTE

- 3D-Lidar-Scans / Hyperspektralaufnahmen von Pflanzen  
→ **Wachstum überwachen / Merkmale erkennen**
- **Dynamische, unstrukturierte Umgebung**
- 2.5D-Simulation der Testfeld-Umgebung in Gazebo inkl.  
AROX-Modell (URDF)

## Langfristigkeit



≈ JAHRE [4]



≈ MONATE [7]

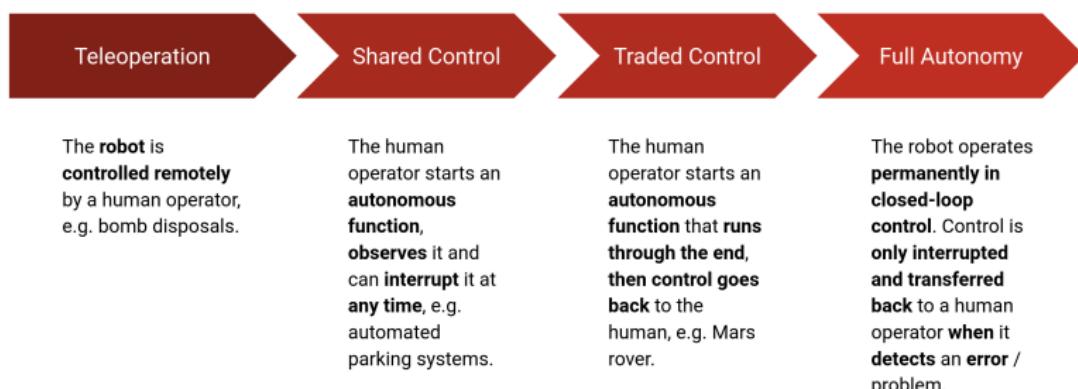


≈ WOCHEN [6]

- Keine strikte Definition → **kontextabhängig**
- Hier:
  - **Ladezyklus** - zeitlicher Handlungsradius des Roboters
  - **Missionszyklus** - repetitiver Charakter der Mission
- **LB**: Prozesse, die eine Reihe solcher Zyklen erfordern
- **UB**: Wartungsintervalle

## Autonomie

- Ebenfalls nicht scharf definiert und **kontextabhängig**
- **Kooperation** nicht prinzipiell ausgeschlossen
- Hier: **Vollautonom** - Inanspruchnahme menschlicher Hilfe ausschließlich im Fehlerfall

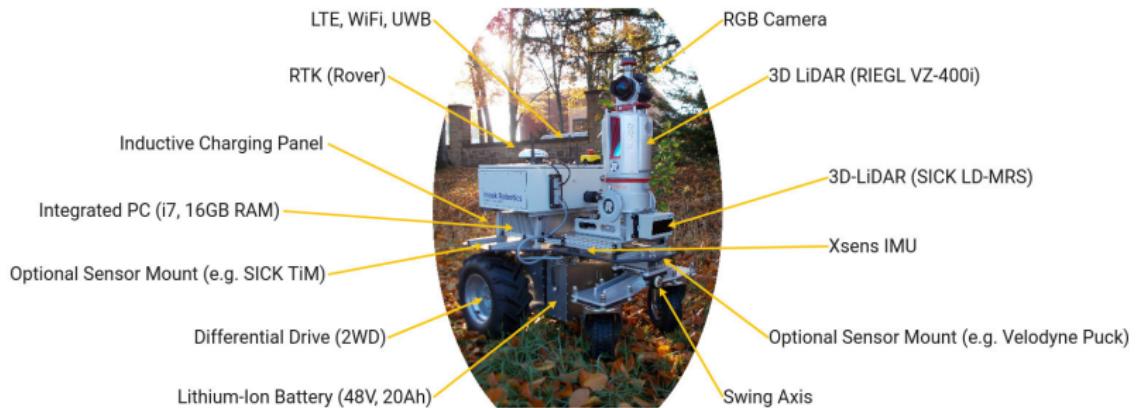


SPEKTRUM DER AUTONOMIE (BASIEREND AUF [1])

## Mobilität

System, das in der Lage ist, sich innerhalb bestimmter Grenzen frei in einer Umgebung zu bewegen [2]

# AROX - Autonomous Robotic Experimentation Platform



**Lokalisierung:** `robot_localization` [3]

→ Sensorfusion von IMU-, Odometrie- und RTK-GNSS-Daten

**Navigation:** `move_base_flex` [5]

# Plangenerierung und -bereitstellung

**ROS-Service** parst manuell erstellte Pläne (CSV)

**Aktionsmenge:**

```
{ drive_to(lat,lng,θ) , return_to_base , charge , scan }
```

**Inkl. Container-Infrastruktur:** dock , undock

```
drive_to,52.32043739026998,8.153532860027937,0
scan
drive_to,52.32063437120376,8.153602612483242,0
scan
drive_to,52.32055680647346,8.153450831140496,0
scan
drive_to,52.32053057955027,8.153441902826218,0
scan
return_to_base
charge
```

BEISPIELPLAN

## Bedingungen:

- ① Kann im betrachteten Szenario **praktisch auftreten**
- ② Kann das reibungslose **Funktionieren des Systems** bzw. die **Qualität** der Ergebnisse **beeinträchtigen**
- ③ Kann **durch Monitoring-Methoden detektiert** (gelöst / kommuniziert) werden

- Energiemanagement
- Fehlgeschlagener Ladevorgang
- Extreme Wetterbedingungen
- Sensorausfall (Wahrnehmung)
- Datenmanagement
- Verbindungsabbruch
- Navigationsfehler
- Fehlerhafte bzw. ungenaue Lokalisierung
- Planverteilungsfehler

→ 48 konkret behandelte Problemfälle

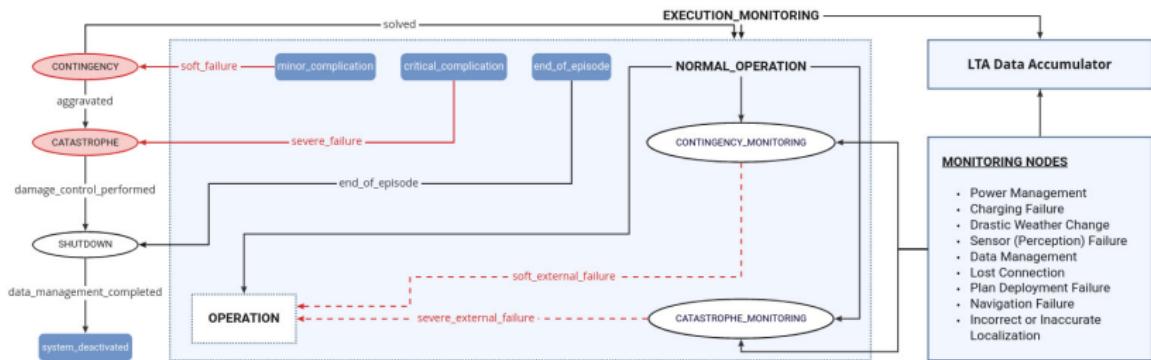
→ Viele weitere implizit abgedeckt

## Klassifikation:

- ① Roboter detektiert ein Problem und ist in der Lage, es selbstständig zu lösen
  - ② Roboter detektiert ein Problem, ist jedoch nicht in der Lage, es zu lösen → informiert (menschlichen) Operator
  - ③ Roboter hat Störung / Problem, detektiert es jedoch nicht und kann es entsprechend weder lösen noch kommunizieren
- **Initial wird jedes potenzielle Problem als Typ (3) eingestuft**
- Ziel:** Sämtliche Probleme in Kategorie (1) / (2) verschieben

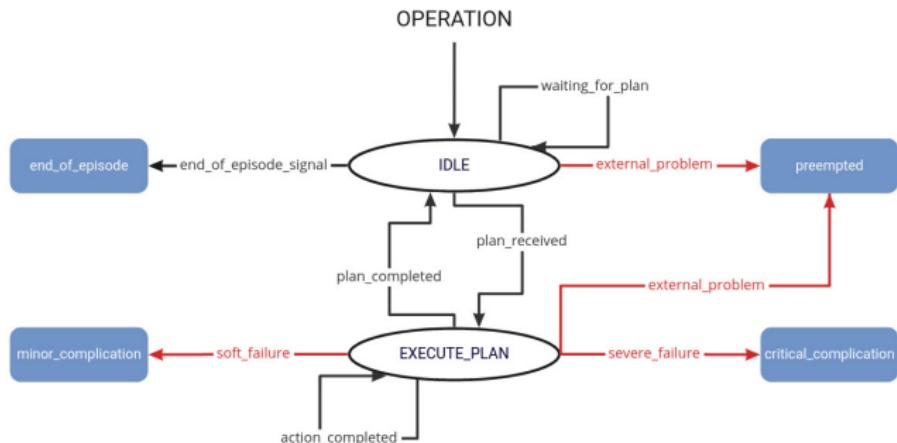
# Execution Monitoring State Machine

## High-Level Architektur der hierarchischen „State Machine“ (ROS-SMACH)



# Execution Monitoring State Machine

## Architektur der eingebetteten „State Machine“



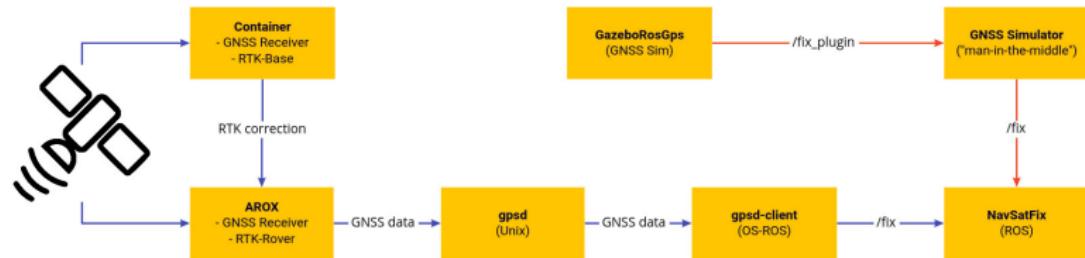
## Zwei Fälle:

- ① IDLE → trivial
- ② EXECUTE\_PLAN → Plan  $\pi$  im userdata-Feld des Zustands gespeichert
  - Aktion  $a \in \pi$  erfolgreich ausgeführt  
 $\rightarrow \pi := \pi \setminus \{a\}$  Input der nächsten Iteration
  - Problem → Unterbrechung der Planausführung
  - Problem gelöst → IDLE
    - $\pi \in \text{userdata.input\_plan}, \pi \neq \emptyset \rightarrow \text{Plan } \pi \text{ fortsetzen}$
    - $\text{userdata.input\_plan} = \emptyset \rightarrow \text{Neuen Plan anfordern}$

## Planausführung fortsetzen

- Geplante Ladestopps nicht notwendigerweise weiterhin zulässig
- **Unterbrechung mit Rückkehr zur Basis-Station**  
→ Planausführung **zulässig**
- **Unterbrechung ohne Rückkehr zur Basis-Station**  
→ Roboter verbraucht ungeplante Ressourcen, sodass a priori Ladestopp-Planung **unzulässig** wird → Batterie-Monitoring

# Exemplarisch: GNSS-Verbindungsprobleme



## Praxis / Simulation

**Resultat:** NavSatFix -Daten auf `/fix` -Topic

# Exemplarisch: GNSS-Verbindungsprobleme

File: `sensor_msgs/NavSatFix.msg`

```
uint8 COVARIANCE_TYPE_UNKNOWN=0
uint8 COVARIANCE_TYPE_APPROXIMATED=1
uint8 COVARIANCE_TYPE_DIAGONAL_KNOWN=2
uint8 COVARIANCE_TYPE_KNOWN=3
std_msgs/Header header
sensor_msgs/NavSatStatus status
float64 latitude
float64 longitude
float64 altitude
float64[9] position_covariance
uint8 position_covariance_type
```

File: `sensor_msgs/NavSatStatus.msg`

```
int8 STATUS_NO_FIX=-1
int8 STATUS_FIX=0
int8 STATUS_SBAS_FIX=1
int8 STATUS_GBAS_FIX=2
uint16 SERVICE_GPS=1
uint16 SERVICE_GLONASS=2
uint16 SERVICE_COMPASS=4
uint16 SERVICE GALILEO=8
int8 status
uint16 service
```

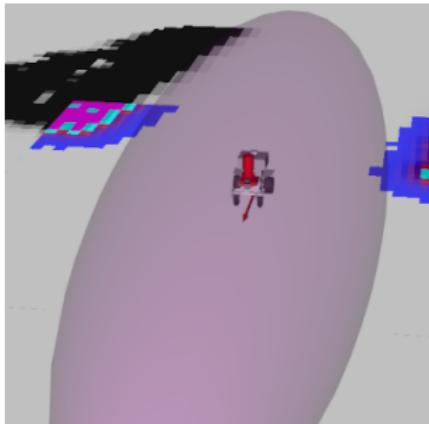
**Kovarianzmatrix:**  $c_{ij} \in C_{3 \times 3}$ ,  $i, j \in \{E, N, U\}$

**Standardabweichung ( $m$ ):**  $\sqrt{c_{ii}} \forall i \in \{E, N, U\}$

# Exemplarisch: GNSS-Verbindungsprobleme



(a) RELATIV GERINGE UNSICHERHEIT



(b) HOHE UNSICHERHEIT BZGL.  $E$

$$(a): (\sqrt{6}m, \sqrt{6}m, \sqrt{6}m)$$

$$(b): (\sqrt{101}m, \sqrt{6}m, \sqrt{6}m)$$

**Beide nicht ideal - kein RTK!**

## Qualitative Bewertung einer prinzipiell funktionierenden GNSS-Verbindung

- **Hoch:**

- STATUS\_GBAS\_FIX
- $\geq$  COVARIANCE\_TYPE\_DIAGONAL\_KNOWN
- $\sqrt{c_{ii}} \leq d_{max} \forall i \in \{E, N, U\}, d_{max} \in \mathbb{R}_0^+$

- **Passabel:**

- $\geq$  STATUS\_FIX
- $\geq$  COVARIANCE\_TYPE\_APPROXIMATED
- Standardabweichungen  $\leq d_{max}$

- **Gering:**

- $\geq$  STATUS\_FIX
- Kovarianztyp unbekannt

Information via /robot\_info

## Problemfall: Nachricht auf /contingency\_preemption

- Verbindungsabbruch (Topic-Check)
- Status unzulässig bzw. STATUS\_NO\_FIX
- Längen- und Breitengrad-Informationen nicht vorhanden bzw. unzulässig
- Unsicherheit - Standardabweichungen
  - Absolute Werte ( $\leq d_{max}$ )
  - Entwicklung über die Zeit

**Lösung:** Kurze Auszeit → Erneuter Verbindungstest  
→ Kommunikation des Problems bei wiederholtem Misserfolg

**Kompatibel mit sämtlichen Systemen, die NavSatFix verwenden!**

## (Problem-Simulation →) Monitoring → Lösung

- **Simulation:** Auslösbar durch Nachricht auf Topic, z.B.  
`rostopic pub -1 /toggle_snow_sim std_msgs/String fail`
- **Monitoring:** Detektiert Problem, unterricht  
`NORMAL_OPERATION`, initiiert geeignete Lösung
- **Lösung:**
  - Problem gelöst → Übergang zu `NORMAL_OPERATION`
  - Problem nicht gelöst → Übergang zu `CATASTROPHE`

**S1:** Zeigen, dass die Mission **ohne Simulation eines der identifizierten Probleme** über einen längeren Zeitraum fehlerfrei läuft

- **Verifikation des Beispielplans**
- **Demonstration der grundsätzlichen Funktionalität**

### Def. gescheiterte Mission:

- ① > 900s keine Nachricht auf `arox/ongoing_operation` trotz offener Tasks
- ② `severe_failure` in `OPERATION` → CATASTROPHE

### **3 Durchläufe à 5 Stunden:**

→ Ø105 Tasks, keine gescheiterte Mission

## S2: Klassifikation der Problemkategorien in Bezug auf ihren potenziellen (worst-case) Schaden

problem	contingency & catastrophe (with monitoring)		catastrophe (without monitoring)		potential to render mission worthless	
	sim	prac	sim	prac	sim	prac
power_management	✓	✓	✓	✓	✓	✓
charging_failure	✓	✓	✓	✓	✓	✓
drastic_weather_change	✓	✗	(✓)	✗	✗	✓
sensor_failure	✓	✗	✗	✗	✓	✓
data_management	✓	✗	✗	✗	✓	✓
lost_connection	✓	(✓)	(✓)	(✓)	✓	✓
plan_deployment_failure	✓	✓	✓	✓	✓	✓
navigation_failure	✓	(✓)	(✓)	(✓)	✓	✓
incorrect_localization	✓	(✓)	(✓)	(✓)	✓	✓

„catastrophe“-Spalte (ohne Monitoring):

- ✓ → Scheitern der Mission sicher
  - (✓) → Scheitern der Mission möglich
- Sämtliche Kategorien sind missionskritisch

## S3: Evaluation der Monitoring-Ansätze

10 Durchläufe à 5h

Zufälliger Fehlerfall alle  $\approx 20\text{min}$

### Betrachtete Aspekte:

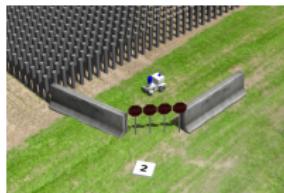
- Erfolgreich abgeschlossene Mission
- **Erwartete Reaktionen** auf Fehlerfälle
  - Korrekte CONTINGENCY -Situationen
  - Korrekte „ $\neg$  CONTINGENCY -Situationen“
- **Unerwartete Reaktionen**
  - Falsch-positiv / falsch-negativ
  - Falsche Reaktion (Problemidentifikation)
- Erfolgreich abgeschlossene Tasks
- Lade- und Missionszyklen
- Zurückgelegte Distanz

# Korrekte „ $\neg$ CONTINGENCY-Situationen“

## Simulation statischer Hindernisse (mit Ausweg)



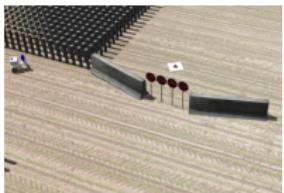
(a) SZENARIO 1



(b) SZENARIO 2

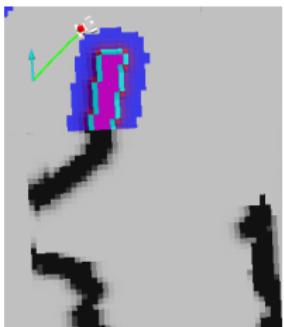
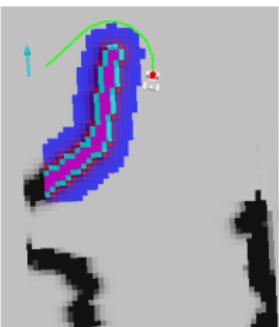
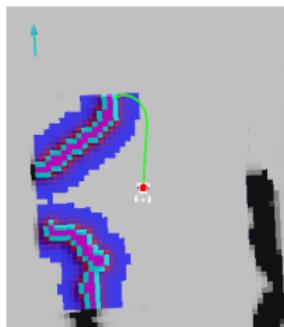
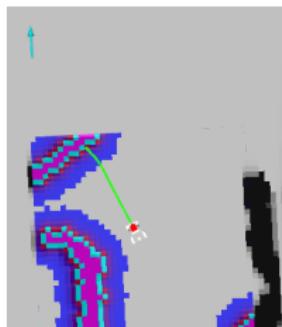


(c) SZENARIO 3



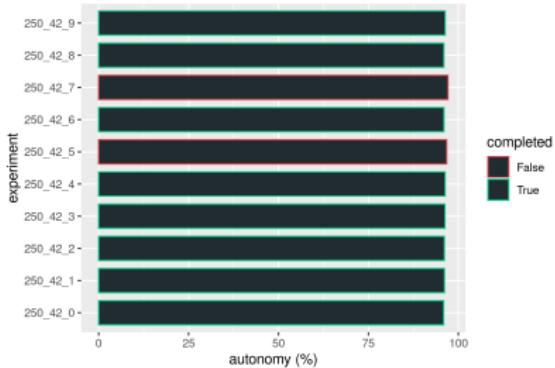
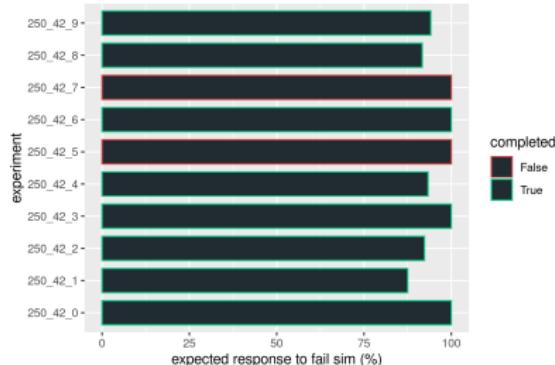
(d) SZENARIO 4

`/spawn_static_obstacles` mit msg  $scene_n$ ,  $n \in \{1, 2, 3, 4\}$



ERFOLGREICHE BEWÄLTIGUNG VON SZENARIO 1

# Ergebnisse



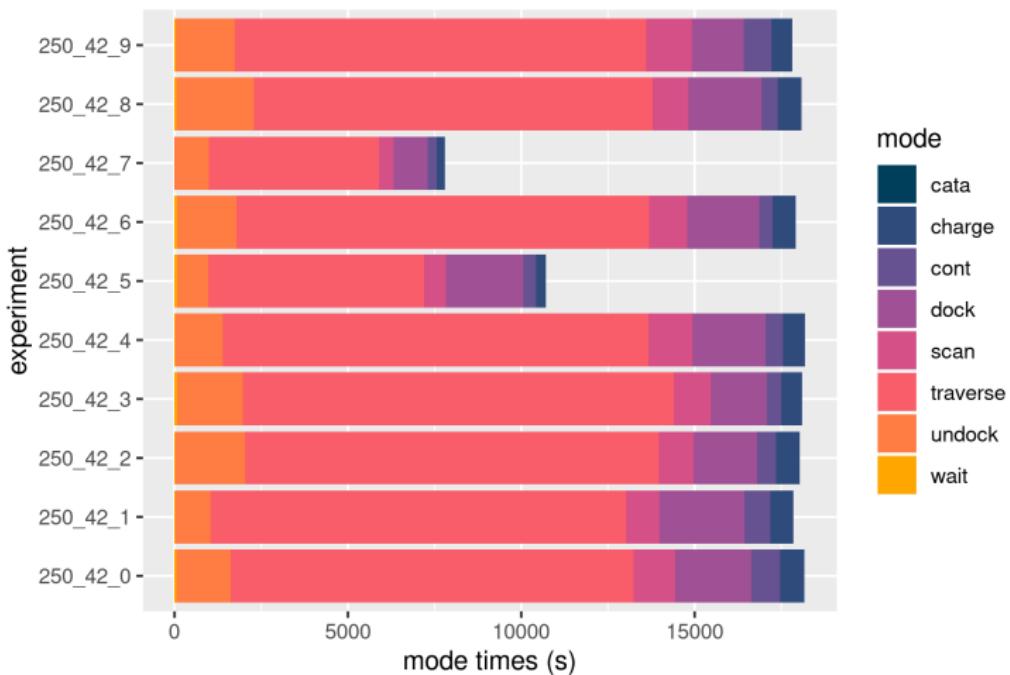
Laufzeit:  $\varnothing 5.06h$  / Ausgeführte Tasks:  $\varnothing 88.75$

Ladezyklen:  $\varnothing 10.38$  / Missionszyklen:  $\varnothing 3$

Zurückgelegte Distanz:  $\varnothing 967.1m$

Sim. Problemfälle:  $\varnothing 13.75$  / Erwartete Reaktion:  $\varnothing 95.89\%$

# Aktivitätsverteilung



- **Funktionalität des integrierten Systems demonstriert**
  - Praktisch relevante **Herausforderungen identifiziert, klassifiziert und in Simulation implementiert**
  - **Generisches Monitoring(-Resolution)-Framework entwickelt und bzgl. Zuverlässigkeit evaluiert**
  - **Sämtliche Problemtypen in Kategorie (1) / (2) verschoben**
- Erheblich **verbesserte Robustheit** bzgl. der identifizierten Herausforderungen

# Aus Erfahrung lernen..

## mongodb\_store

```
[{"_id": ObjectId("622d487847f53faf6023d9e2"),  
 "data": "docking failure - explicit docking smach failure",  
 "_meta": {  
     "name": "contingency",  
     "inserted_at": "2022-03-13 02:27",  
     "timestamp": NumberLong(8453495000000),  
     "stored_type": "std_msgs/String",  
     "published_at": "2022-03-13 02:27",  
     "inserted_by": "/data_accumulator",  
     "stored_class": "std_msgs.msg._String.String"  
 },  
  
,  
 {"_id": ObjectId("622d487847f53faf6023d9e3"),  
 "data": "{\"charge\": 67.301, \"completed_tasks\": 123, \"charge_cycle\": 11, \"robot_pose\": {\"lat: 52.320353496, lng: 8.15367125982\", \"operation_time\": \"21523.377303s\"}}",  
 "_meta": {  
     "name": "failure_circumstances",  
     "inserted_at": "2022-03-13 02:27",  
     "timestamp": NumberLong(8453505000000),  
     "stored_type": "std_msgs/String",  
     "published_at": "2022-03-13 02:27",  
     "inserted_by": "/data_accumulator",  
     "stored_class": "std_msgs.msg._String.String"  
 },  
  
,  
 {"_id": ObjectId("622d489e47f53faf6023d9e8"),  
 "data": "launch charging failure resolver -- type of charging failure: docking failure - explicit docking smach failure",  
 "_meta": {  
     "name": "resolution",  
     "inserted_at": "2022-03-13 02:27",  
     "timestamp": NumberLong(8461502000000),  
     "stored_type": "std_msgs/String",  
     "published_at": "2022-03-13 02:27",  
     "inserted_by": "/data_accumulator",  
     "stored_class": "std_msgs.msg._String.String"  
 }]
```

- Ausführliche **Protokollierung der LTA-Missionen**  
(insbesondere Fehlerkontext)
- Gesammelte **Daten semantisch gruppiert**

- **Praxistest** mit dem AROX-System
- Weitere **Experimente** (Seed, Laufzeit, Fehlerfrequenz, etc.)
- Datenbankeinträge nutzen: **Aus LTA-Missionen lernen**
- **Elaboriertere Lösungsmethoden** (Fokus hier Detektion)
- **Web-Interface**: Remote-Zugriff auf Sensoren / Kontrolle
- **Usability**: Einbindung / Rekonfiguration des Frameworks

-  Joachim Hertzberg.  
**Technische Gestaltungsoptionen für autonom agierende Komponenten und Systeme.**  
In *Das Recht vor den Herausforderungen der modernen Technik. Beiträge der 1. Würzburger Tagung zu Technikrecht*, pages 63–74, 2015.
-  Joachim Hertzberg, Kai Lingemann, and Andreas Nüchter.  
**Mobile Roboter: Eine Einführung aus Sicht der Informatik.**  
eXamen.press. Springer Vieweg, Berlin, 2012.
-  Thomas Moore and Daniel Stouch.  
**A Generalized Extended Kalman Filter Implementation for the Robot Operating System.**  
In *Intelligent Autonomous Systems - Proceedings of the 13th International Conference IAS-13, Padova, Italy*, volume 302 of *Advances in Intelligent Systems and Computing*, pages 335–348. Springer, 2014.

- Maas Digital LLC NASA/JPL/Cornell University.  
**NASA Mars Rover.**  
[https://commons.wikimedia.org/wiki/File:NASA\\_Mars\\_Rover.jpg](https://commons.wikimedia.org/wiki/File:NASA_Mars_Rover.jpg).  
Accessed: 2022-03-02.
- Sebastian Pütz, Jorge Santos Simón, and Joachim Hertzberg.  
**Move Base Flex: A Highly Flexible Navigation Framework for Mobile Robots.**  
In *International Conference on Intelligent Robots and Systems, IROS, Madrid, Spain*, pages 3416–3421. IEEE, 2018.
- MetraLabs GmbH | Neue Technologien und Systeme.  
**SCITOS A5.**  
<https://www.metralabs.com/en/service-robot-scitos-a5/>.  
Accessed: 2022-03-02.



Oregon State University.

**Seafloor Geodesy and the Wave Glider.**

[https://blogs.oregonstate.edu/pictures/2016/11/25/  
seafloor-geodesy-wave-glider/](https://blogs.oregonstate.edu/pictures/2016/11/25/seafloor-geodesy-wave-glider/).

Accessed: 2022-03-02.