Hochschule Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

RWTH AACHEN UNIVERSITY

On the Diagnosability of Actions Performed by Contemporary Robotic Systems

Alex Mitrevski, Ahmed Faisal Abdelrahman, Anirudh Narasimamurthy, and Paul G. Plöger

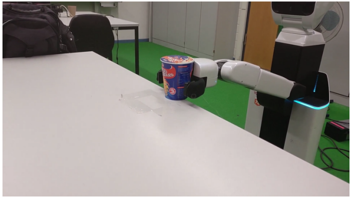
31th International Workshop on Principles of Diagnosis (DX)

DX WORKSHOP

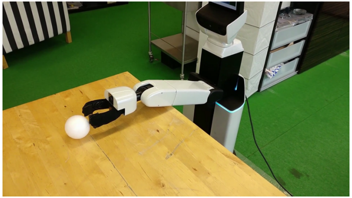
Problem of Interest: Robot Execution Failures

Object placed on the shelf edge

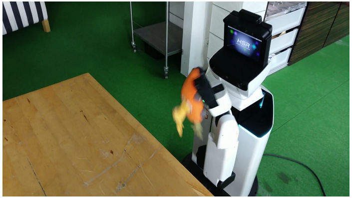
Missed drawer handle



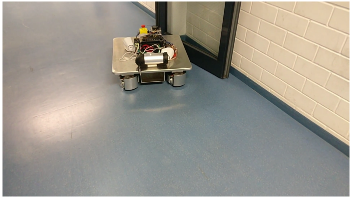
Object pulled too far



Ball pushed away while grasping



Toy slipped out



Door collisions

Robot Execution Failures

In all previous cases

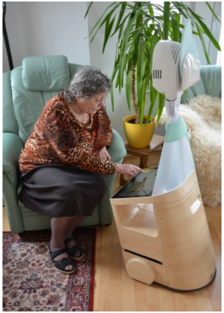
- the robots were performing some **action** (placing, grasping, pulling, driving)
- by following an (optimised) **execution policy**

Failures happened regardless

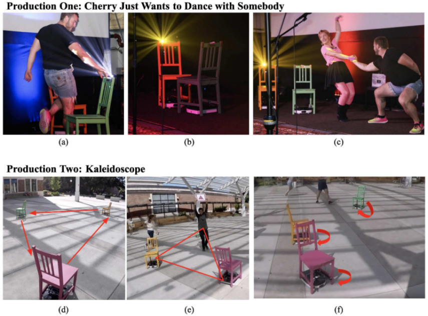
We would clearly want to answer the question **"Why?"**

Why Does Analysing Robot Faults Matter?

- Continuous system improvement
- Increased user trust[1,2]
- Complete autonomy and lifelong learning



Robot interacting with elderly person[1]



Entertainment robots on stage[2]

[1] H. Gross et al., "Living with a Mobile Companion Robot in your Own Apartment - Final Implementation and Results of a 20-Weeks Field Study with 20 Seniors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2253-2259, 2019.

[2] A. Fallatah, J. Urann, and H. Knight, "The Robot Show Must Go On: Effective Responses to Robot Failures," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 325-332, 2019.

How About Model-Based Diagnosis?

Diagnosis requires at least **a nominal model of execution**, but manually developing such models does not scale/generalise well

Contemporary robot action representations are thus - mostly - learning-based

We argue that **diagnosability exists on a modelling-learning continuum**

- More modelling knowledge $\implies$ more likely that the execution will be diagnosable
- Less modelling knowledge $\implies$ more flexible the execution is likely to be

General Topics of This Talk

- Types of action representation mechanisms in robotics
  - Focus on density-/symbolic-based and neural network-based
- Criteria for robot action diagnosability
  - We define some (minimal) criteria for being able to diagnose execution failures
- Where we stand in terms of action diagnosability
  - In general, we have a long way to go

What is a Robot Action?

In this paper, we adopt a planning definition of an action[3]:

- has preconditions and
- achieves some effects after being performed
- (the effects may be non-deterministic)

The execution process should maximise the probability that the desired effects will be achieved

PDDL definition of a grasping action[4]

```
(:action Pick
    :parameters (?Object - Object ?Plane - Plane ?Robot - Robot ?
           Waypoint - Waypoint ?Context - Context)
    :precondition (and
        (= ?Context pick_from_plane)
        (robotAt ?Robot ?Waypoint)
        (planeAt ?Plane ?Waypoint)
        (explored ?Plane)
        (on ?Object ?Plane)
        (emptyGripper ?Robot)
    )
    :effect (and
        (not (on ?Object ?Plane))
        (not (emptyGripper ?Robot))
        (holding ?Robot ?Object)
    )
)
```

[3] P. Zech et al., "Action representations in robotics: A taxonomy and systematic classification," *The International Journal of Robotics Research*, vol. 38, no. 5, pp. 518-562, 2019.

[4] https://github.com/b-it-bots/mas_domestic_robotics/blob/devel/mdr_planning/mdr_rosplan_interface/config/default_domain.pddl

Action Execution Policies

The execution of a robot action is governed by an **execution policy**

A common representation of robot behaviour is using a Markov Decision Process (MDP)

$$\mathcal{M} = (S, A, \mathcal{T}, \mathcal{R}, s_0)$$

An execution policy $\pi : S \rightarrow A$ is then the model of the robot's behaviour, and an optimal policy π^* is often found using reinforcement learning

But MDPs are not the only existing model of behaviour

Learning-Based Action Execution Models

In the paper, we contrast two models of execution:

- Density-based and symbolic action models
- Neural network-based execution policies

Please see the paper for a brief overview of the state-of-the-art of these two paradigms

Density-Based Execution Models

An execution model is often represented by a density function for selecting action parameters

$$p(\mathbf{x}|\mathbf{s}) \sim p(\mathbf{s}|\mathbf{x})p(\mathbf{s})$$

Density-based models often encode modelling knowledge about the relation between the observed state and the appropriate action

Models usually tailored to a particular use case

Network-Based Execution Models

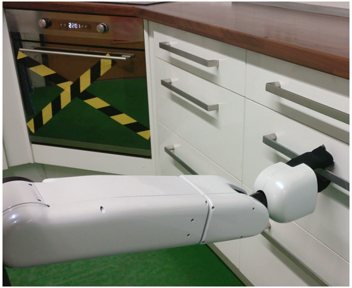
An execution policy is parameterised by the network weights: $\pi(\mathbf{s}, \mathbf{w})$

Quite flexible and can be used with high-dimensional inputs (e.g. visual input)

In robotics, often trained in simulation and/or using prior knowledge (e.g. human demonstrations)

Running Example: Handle Grasping Action

```
(:action GraspHandle
    :parameters (?Handle - Handle ?Robot - Robot ?Waypoint -
        Waypoint)
    :precondition (and
        (robotAt ?Robot ?Waypoint)
        (handleAt ?Handle ?Waypoint)
        (emptyGripper ?Robot)
    )
    :effect (and
        (not (emptyGripper ?Robot))
        (holding ?Robot ?Handle)
    )
)
```



(Minimal) Criteria for Diagnosability

- Abstractability
- Predictability
- Composability

Abstractability

Abstractability

Complexity to find an abstraction $\mathcal{A}(a)$ for each action a

An action is diagnosable if we can create an abstraction of it and reason about it

The abstraction can be, for example:

- an identity mapping (if the action model is already suitable for reasoning)
- created alongside a parametric execution model

Predictability

Predictability

Small changes in the input should cause small changes in the output: $a(\mathbf{x}) \approx a(\mathbf{x} + \epsilon)$

Diagnosability also depends on actions being predictable

Predictability imposes a smoothness criterion on the parameter space of the action

Important because diagnosis may require exploring alternative action parameterisations

Composability

Composability

Given a pair of actions a_i and a_j, their composition is defined as $(a_j \circ a_i)(\mathbf{x})$

For diagnosis, action composability is also an important criterion

Blame assignment can be done more accurately for composable actions

Composability at a very granular level can be detrimental to diagnosis

Diagnosability Levels

1. **Designer level**
   - System designers are improving the system, so they should be able to (easily) diagnose failures
   - Can be achieved by making the execution state explicit and extensive data logging

2. **User level**
   - If/When robots are deployed in everyday environments, users should also be able to understand the reasons for a failure
   - Related to abstractability

3. **Self-diagnosis**
   - Necessary for full robot autonomy
   - Requires an ability to discover relations between action parameters and action outcomes

Handle Grasping: Density-Based Model

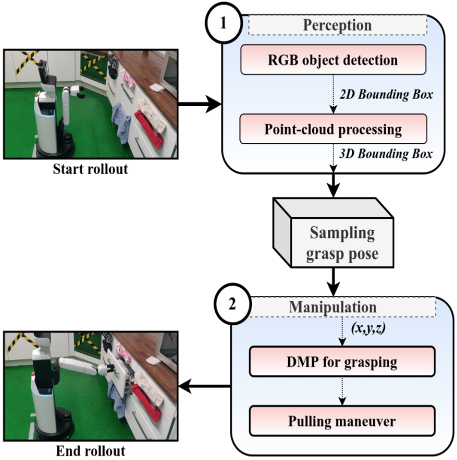
Input: 3D bounding box of a detected handle

Output: Grasping position relative to the center of the handle's bounding box

Mapping:

$$\Delta \mathbf{p} \sim \mathcal{N}(\Delta \mathbf{p} | \mu, \Sigma)$$

where μ and Σ are learned from nominal executions



Handle Grasping: Network-Based Policy

Input: RGB image I

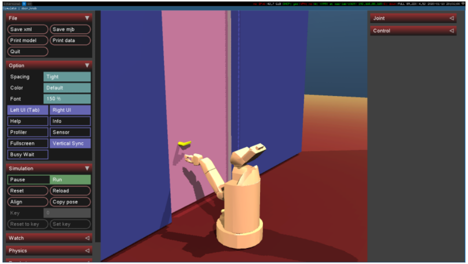
Output: Arm joint torques

Mapping:

$$\mathbf{x} = f(I, \mathbf{w})$$

where

- the output represents the mean of a diagnonal Gaussian distribution over arm joint torques
- the network is trained in a Mujoco simulation[5]using Proximal Policy Optimization[6]



[5] Y. Urakami et al., "DoorGym: A Scalable Door Opening Environment And Baseline Agent," *CoRR, abs/1908.01887*, 2019.

[6] J. Schulman et al., "Proximal Policy Optimization Algorithms," *CoRR, abs/1707.06347*, 2017.

Thought Experiment: Handle Grasping Action Diagnosability

Qualitative comparison of the action representations

	Density-based model	Network-based model
Abstractability	* Doesn't include a direct abstraction, but an abstraction can be created alongside the model (see next slide)	* Doesn't include a direct abstraction; creating a mapping may include a process similar to that of learning the density-based model
Predictability		
Composability		

Thought Experiment: Handle Grasping Action Diagnosability

Qualitative comparison of the action representations

	Density-based model	Network-based model
Abstractability	* Doesn't include a direct abstraction, but an abstraction can be created alongside the model (see next slide)	* Doesn't include a direct abstraction; creating a mapping may include a process similar to that of learning the density-based model
Predictability	* Smoothness of output guaranteed by Gaussian model * Output affected by image noise	* Smoothness of output guaranteed by Gaussian model * Output affected by image noise * Policy trained purely in simulation, so affected by domain shift
Composability		

Thought Experiment: Handle Grasping Action Diagnosability

Qualitative comparison of the action representations

	Density-based model	Network-based model
Abstractability	* Doesn't include a direct abstraction, but an abstraction can be created alongside the model (see next slide)	* Doesn't include a direct abstraction; creating a mapping may include a process similar to that of learning the density-based model
Predictability	* Smoothness of output guaranteed by Gaussian model * Output affected by image noise	* Smoothness of output guaranteed by Gaussian model * Output affected by image noise * Policy trained purely in simulation, so affected by domain shift
Composability	Execution split into different subactions	Monolithic policy, compositional at a low level of abstraction (since output represents joint motions)

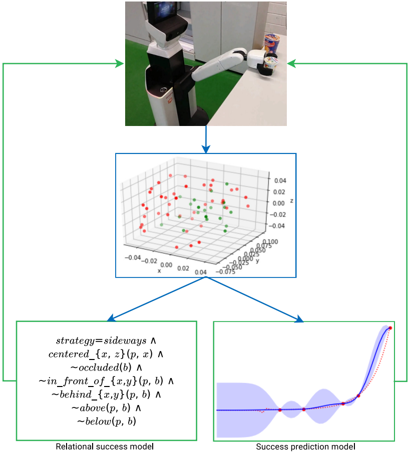
Hybrid Action Representation Framework[7]

In our current work, we are developing a hybrid representation of action execution

This representation:

- includes an abstraction (in terms of qualitative relations)
- is predictable (mapping action parameters to predicted execution success)
- is inherently composable

Ongoing work looks at using the representation for diagnosing failures as violations of the learned relational model



[7] A. Mitrevski, P. G. Plöger, and G. Lakemeyer, "Representation and Experience-Based Learning of Explainable Models for Robot Action Execution," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. To appear.

Challenges in Diagnosing Contemporary Robots

- **Unclear what is a good level of abstraction** for meaningful diagnosis
- **Hierarchical structure** inherent in robot actions **difficult to use without sacrificing the flexibility** of the representation
- **Prior knowledge essential** for learning execution policies effectively, **but too much prior knowledge may reduce flexibility and generalisability**
- Development of diagnosis methods complicated by the **existence of conceptually distinct representations**
- Currently, **little focus on understanding causal relations** while acting

Promising Avenues for Improving Robot Action Diagnosability

- Learning execution abstractions[8]
- Learning composable structures[9,10] and predictive models[11]
- Testing robots as extensively as possible, particularly when visuomotor policies are used[12]

[8] C. Mueller, J. Venicx, and B. Hayes, "Robust Robot Learning from Demonstration and Skill Repair Using Conceptual Constraints," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6029-6036, 2018.

[9] J. Tremblay et al., "Synthetically Trained Neural Networks for Learning Human-Readable Plans from Real-World Demonstrations," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5659-5666, 2018.

[10] B. Ames, A. Thackston, and G. Konidaris, "Learning Symbolic Representations for Planning with Parameterized Skills," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 526-533, 2018.

[11] J. Stüber, M. Kopicki, and C. Zito, "Feature-Based Transfer Learning for Robotic Push Manipulation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5643-5650, 2018.

[12] M. Zhang, et al. "DeepRoad: GAN-Based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, pages 132-142, 2018.