Interactive Tuning of Robot Program Parameters via Expected Divergence Maximization

Mattia Racca, Ville Kyrki, and Maya Cakmak

Aalto University
School of Electrical
Engineering

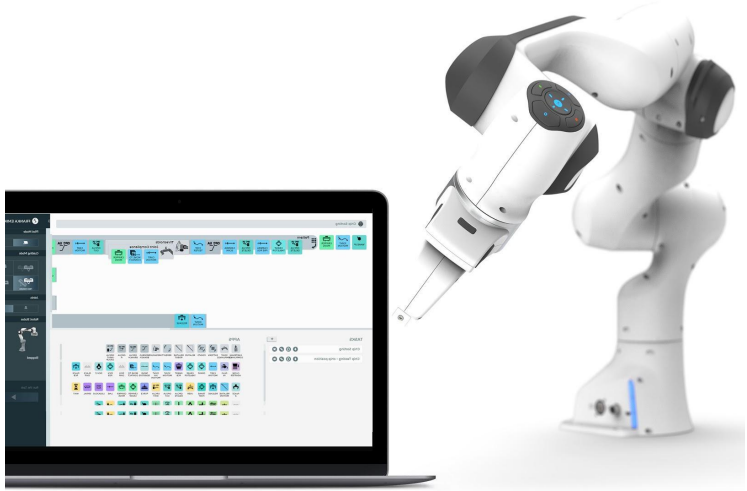
UNIVERSITY of
WASHINGTON



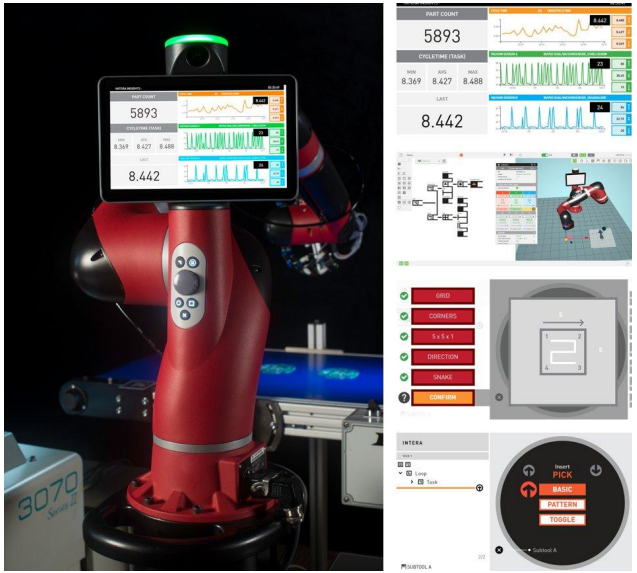
15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)

ROBOT END-USER PROGRAMMING

Main motivation: **End-User Programming makes robots accessible to novice users**



Desk Environment for FRANKA EMIKA Panda robot



Intera Environment for Rethink Robotics Sawyer

AIDING END-USER PROGRAMMING

Robot actions are the building blocks for EUP programs, with each varying number and complexity of parameters

PARAMETERS

- Goal Pose
- Translational Speed
- Collision Threshold (move until you sense a contact)

AIDING END-USER PROGRAMMING

Robot actions are the building blocks for EUP programs, with each varying number and complexity of parameters

PARAMETERS
- Goal Pose
- Translational Speed
- Collision Threshold (move until you sense a contact)

Challenges faced by end-users

- What robot actions to use to achieve the goal?
- **How to set the action parameters?**
- How to evaluate the program (debug and fix)?

PARAMETER TUNING FOR ROBOT ACTIONS

How are parameter values usually specified?

PARAMETERS

- Goal Pose → Kinesthetic Teaching
- Translational Speed
- Collision Threshold (move until you sense a certain amount of force)

→ GUI elements (e.g. 1-D sliders)

PARAMETER TUNING FOR ROBOT ACTIONS

How are parameter values usually specified?

PARAMETERS

- Goal Pose → Kinesthetic Teaching
- Translational Speed
- Collision Threshold (move until you sense a certain amount of force)

→ GUI elements (e.g. sliders)

Trial-and-error tuning strategy (**tedious** and **time consuming**):

- often effects of changes to parameter values are not immediate
- sometimes specifying a single value is not enough

AIDING 1-D PARAMETER TUNING

Idea: **what if the robot proposes the parameter values to try? instead of the user selecting them with sliders**

AIDING 1-D PARAMETER TUNING

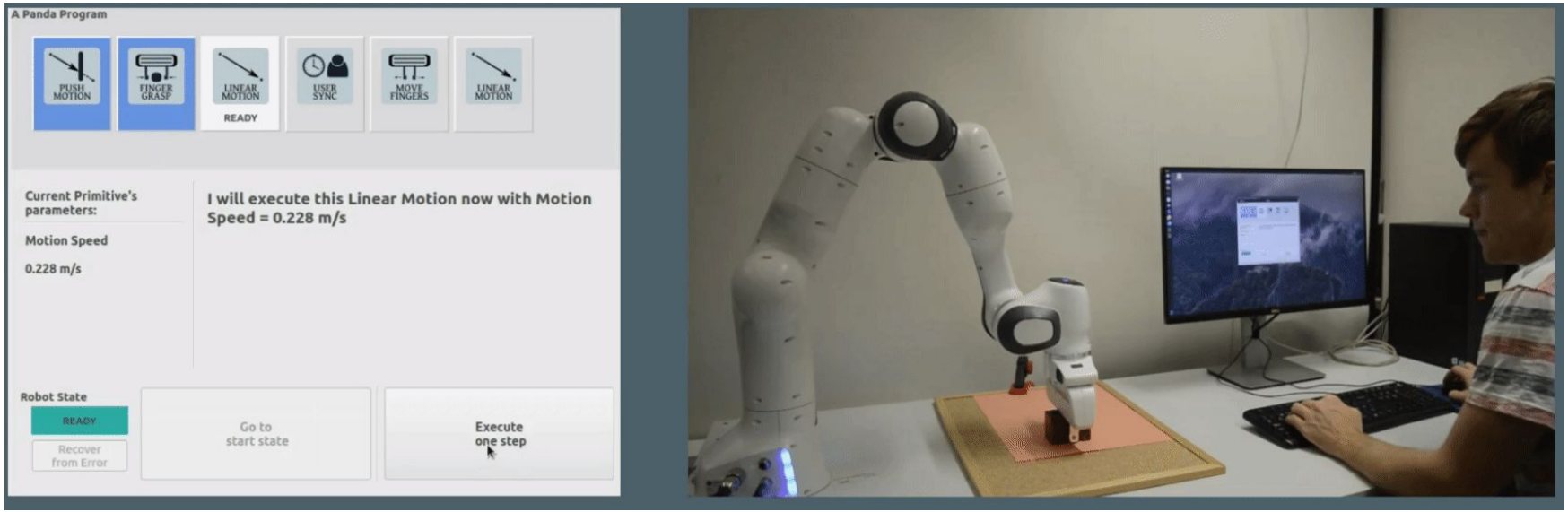
Idea: **what if the robot proposes the parameter values to try? instead of the user selecting them with sliders**

Formulation: **we formulate this as an Active Learning (AL) problem.**

AL agent iteratively:

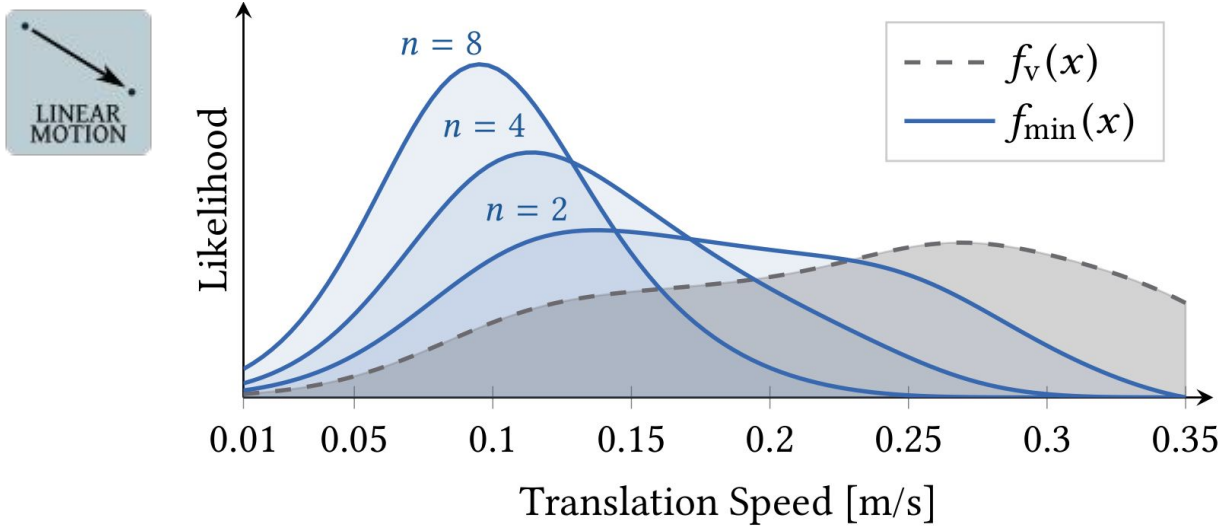
1. **selects informative parameter values to try** (*query selection*)
2. **action is reproduced with selected parameter** (*actual querying*)
3. **user gives feedback** (*answering*)
4. **parameter range estimation is updated** (*model update*)

TUNING PIPELINE IN ACTION



BAYESIAN PARAMETER ESTIMATION

Bayesian approach: **priors over parameter values** (e.g. from expert programs)



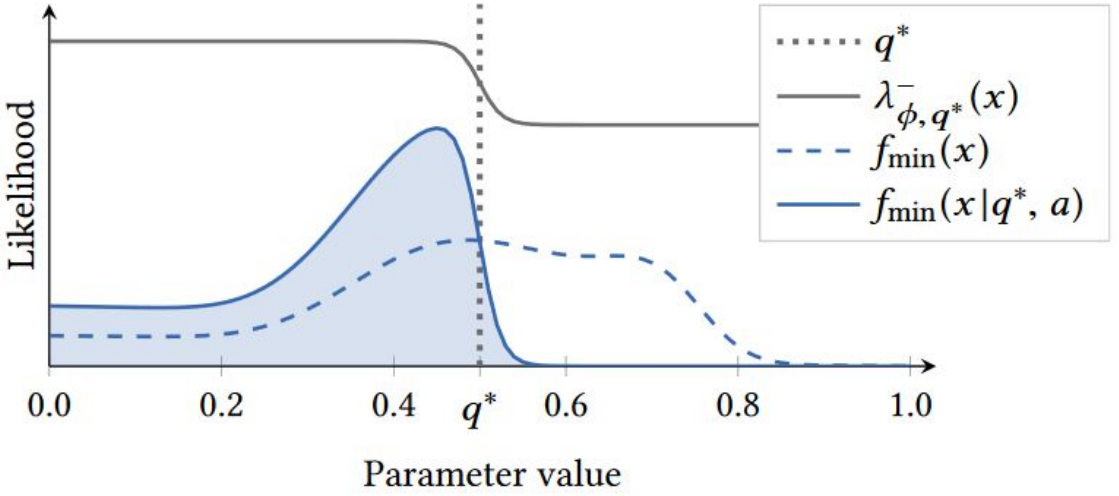
FEEDBACK AFTER THE ACTION

Directional answers: given the 1-D nature of the estimated parameters**, the user's feedback can be directional (higher, lower, fine)**

FEEDBACK AFTER THE ACTION

Directional answers: given the 1-D nature of the estimated parameters**, the user's feedback can be directional (higher, lower, fine)**

"The motion was too fast! I want it slower"



SELECTING THE VALUE TO PROPOSE

How to select the **parameter value**?

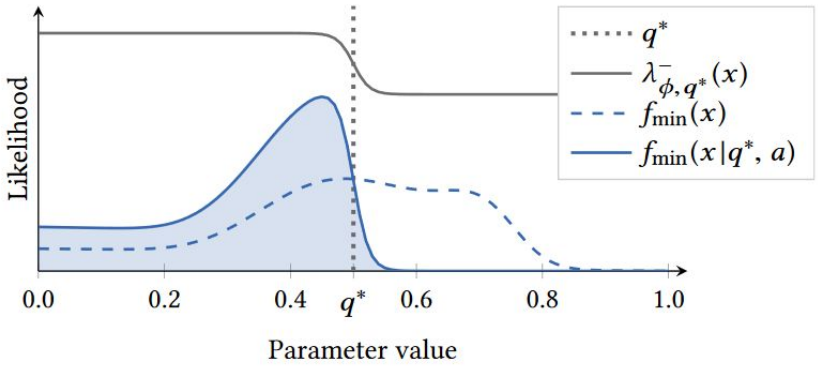
- **At random** (complexity O(1))
- **Uncertainty sampling** (O(k)) ~ basically a weighted binary search on the prior
- **Expected Divergence Maximization** ($O(k^2)$)

SELECTING THE VALUE TO PROPOSE

How to select the **parameter value**?

- **At random** (complexity O(1))
- **Uncertainty sampling** (O(k)) ~ basically a weighted binary search on the prior
- **Expected Divergence Maximization** (O(k^2))

$$S_v^{\cdot} = \mathop{\mathbb{E}}_{a}[\mathbb{JS}(\overbrace{f_{\cdot}(x|v,a)}^{\text{post query}}, \overbrace{f_{\cdot}(x)}^{\text{pre query}})]$$
$$= \sum_{a} p(a|v, f_{\cdot}(x))\mathbb{JS}(f_{\cdot}(x|v,a), f_{\cdot}(x))$$



EXPERIMENTS

1. synthetic priors and simulated oracle users
2. priors from expert programs (8 experts) and simulated oracle users
3. **usability study: 8 novice users, using expert priors**

EXPERIMENTS

1. synthetic priors and simulated oracle users
2. priors from expert programs (8 experts) and simulated oracle users
3. **usability study: 8 novice users, using expert priors**

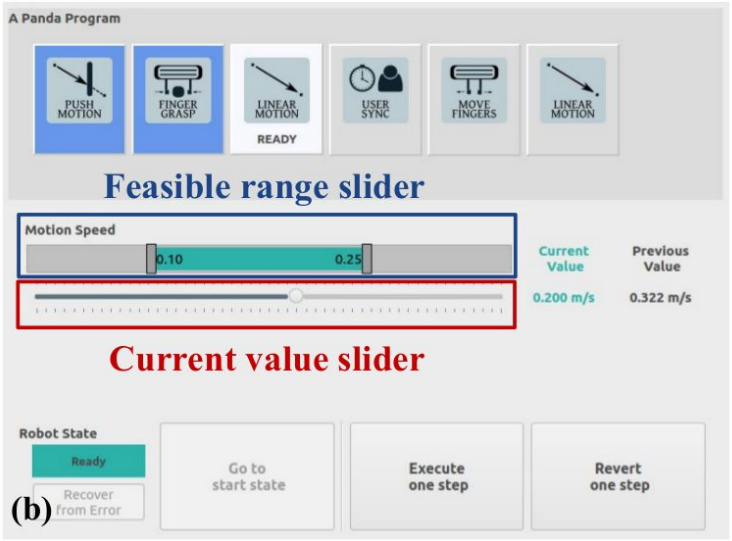
Domain Specific Language (DSL) -- 5 actions each with 1 or 2 parameters

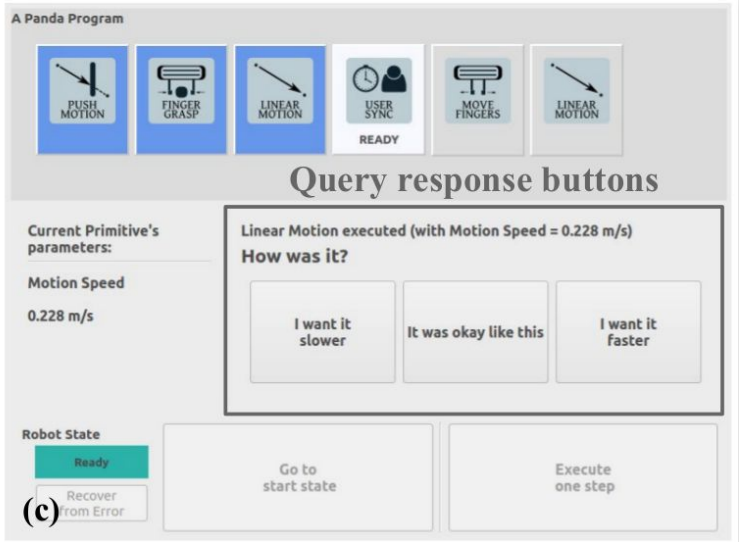
Tasks for priors: handover, 2 pushing tasks, and a pick and place

EVALUATION WITH NOVICE USERS

Task: **tune the parameters of a provided handover program**

Conditions: **Baseline (GUI sliders)** *vs* **Active tuning**





RESULTS and OBSERVATIONS

- **SUS score: baseline 73.7** *vs* **Active tuning 73.1 (good usability)**
- **With Active tuning**, novice users produced parameter ranges **closer** to the expert ones
- **With Active tuning**, **faster tuning** (**8 min** *vs* **13 min**)

RESULTS and OBSERVATIONS

- **SUS score: baseline 73.7** *vs* **Active tuning 73.1 (good usability)**
- **With Active tuning**, novice users produced parameter ranges **closer** to the expert ones
- **With Active tuning**, **faster tuning** (**8 min** *vs* **13 min**)

From participants' feedback:

- **Active tuning** helped at the beginning **but had strict control over the process**
- Participants reduced tuning attempts over time with the baseline
  Active tuning did not → perceived as slower and less efficient!

INTERESTING FUTURE DIRECTIONS

Interaction side:

- **Control issue: more discreet ways of suggesting parameter values to try out**
  e.g. overlaying information on the GUI sliders or Active tuning only on demand
- **Time consuming: can (some) executions be handled in simulation/visualization tools?** can users still express feedback?

INTERESTING FUTURE DIRECTIONS

Interaction side:

- **Control issue: more discreet ways of suggesting parameter values to try out**
  e.g. overlaying information on the GUI sliders or Active tuning only on demand
- **Time consuming: can (some) executions be handled in simulation/visualization tools?** can users still express feedback?

Learning side:

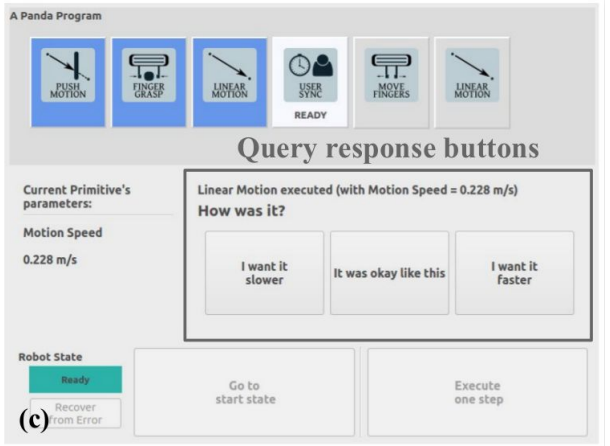
- **Different querying schemes**: if action has 2 or more parameter, let AL agent **pick which parameter to tune**
- **Tune single or multiple parameters at a time?** can users still express feedback?

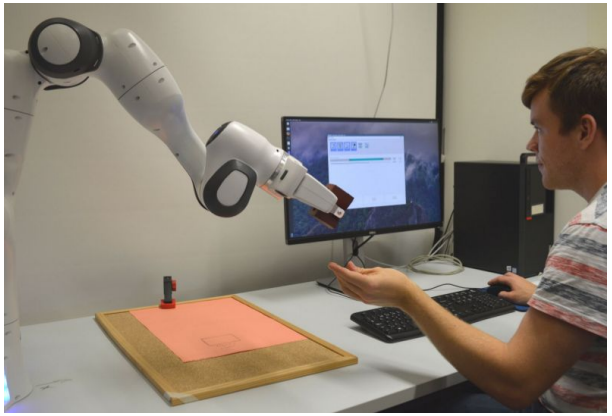
CONCLUSIONS

We framed the tuning of parameters of robot actions as an Active Learning problem and proposed a novel interactive tuning method.

We validated the tuning approach both in simulation and in a real robot scenario.

Experiments showed the usability of the method with novice users, and allowed us to identify several promising future directions.





Interactive Tuning of Robot Program Parameters via Expected Divergence Maximization

Code available at github.com/mattiaracca/eupanda
Showcase video at vimeo.com/mattiaracca/hri20
Paper available on ACM Digital Library!

Mattia Racca, Ville Kyrki, and Maya Cakmak

mattia.racca@aalto.fi

Aalto University
School of Electrical
Engineering

UNIVERSITY of
WASHINGTON

15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)