Controlling 'uncontrollable' stuff: playing with the H_2Arm

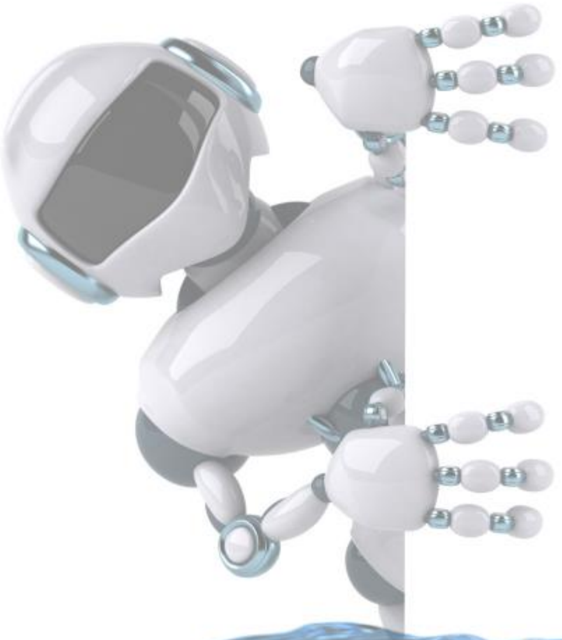
E. Zereik CNR-INM

ISTITUTO DI INGEGNERIA DEL MARE
INSTITUTE OF MARINE ENGINEERING

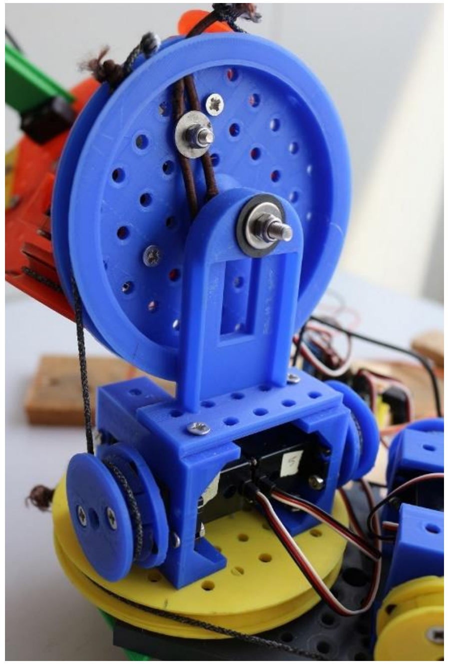
‘Uncontrollable’ stuff

- Uncertain and complex dynamics
- Uncertain model
- Large noise affecting control scheme
- Unstructured environment
- Soft robots

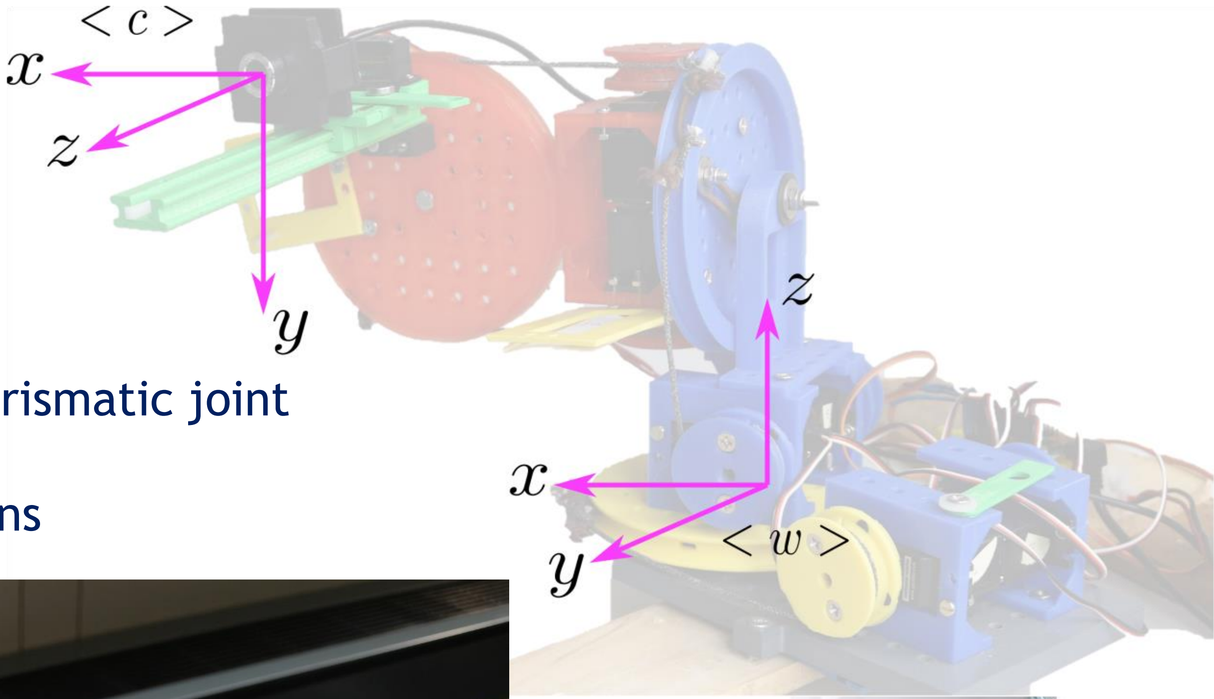
- Deterministic vs. stochastic algorithms
- Use of ML

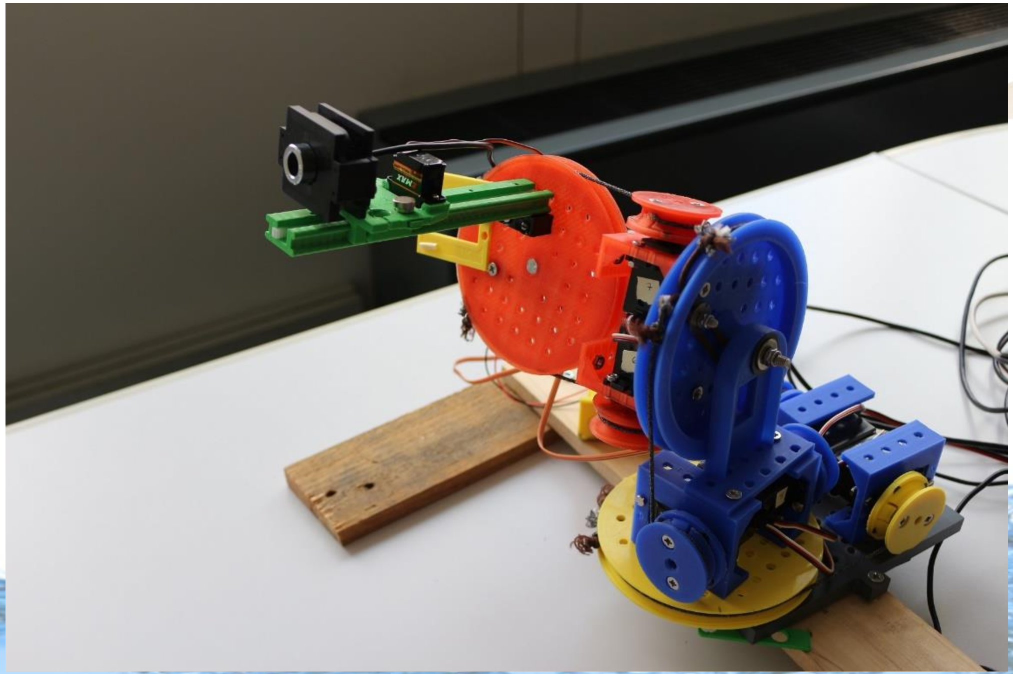


H_2Arm robotic setup

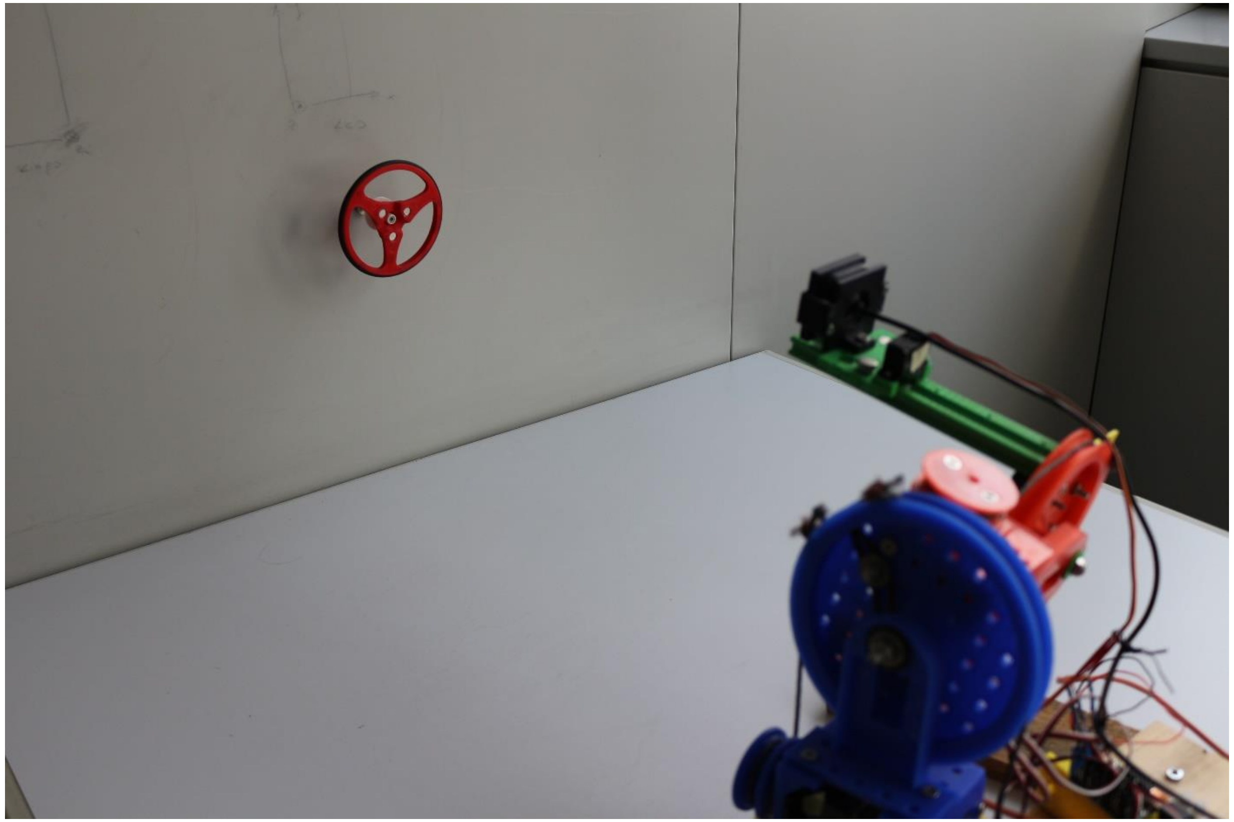


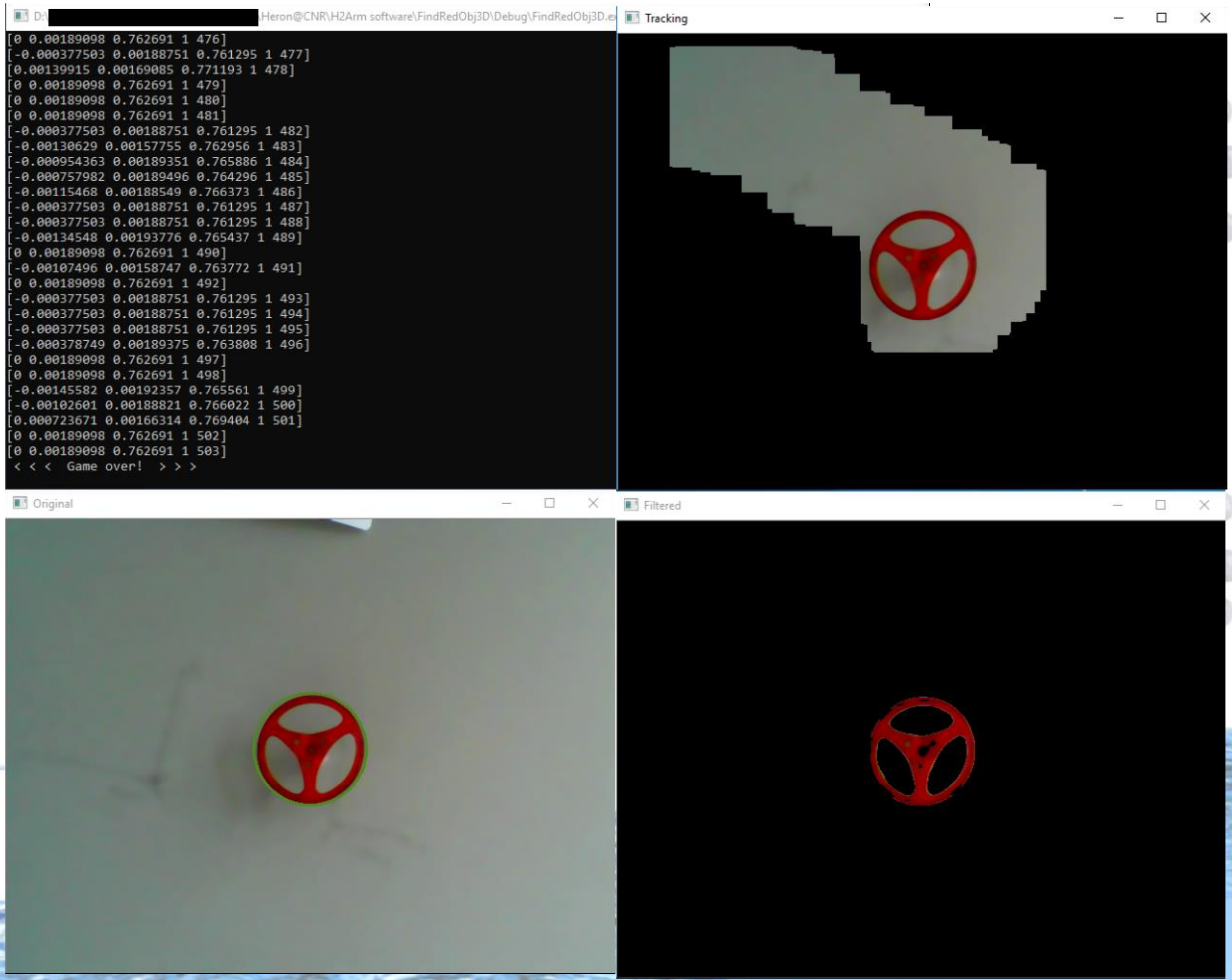
- 4 dofs, 3 rotational joints, 1 prismatic joint
- Webcam mounted on wrist
- Counteracting motors + tendons





Required task

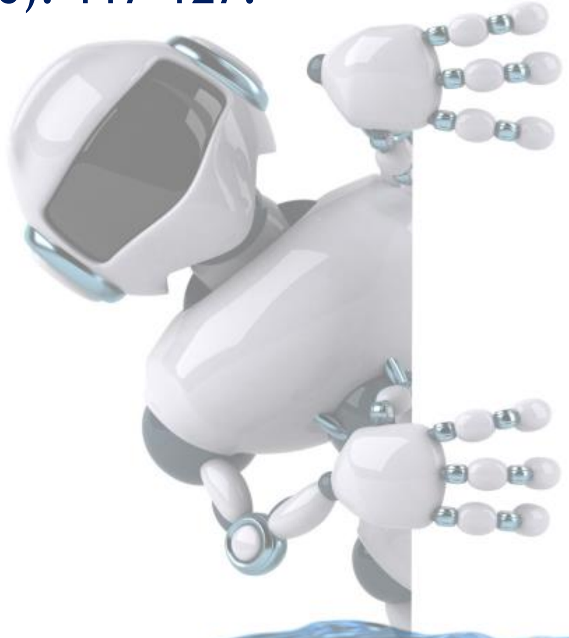




Reproducibility

Fabio Bonsignorio and Enrica Zereik. "A simple visual-servoing task on a low-accuracy, low-cost arm: an experimental comparison between belief space planning and proportional-integral-derivative controllers." IEEE Robotics & Automation Magazine 28.3 (2020): 117-127.

Deterministic
vs
Stochastic control



Belief Space Planning

- probability density function of the system state
- Gaussian distribution
- Uncertainties incorporated in the evolution of the belief state

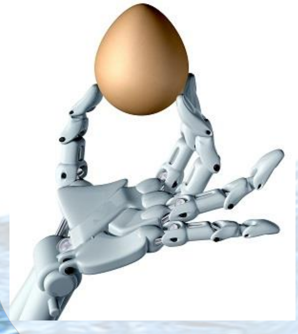
Robustness to measurement errors

Robustness to disturbances

Compensate for uncertain kinematics and dynamics

Concurrent reduction of

end-effector distance from the target



underlying uncertainty

BSP versus PID

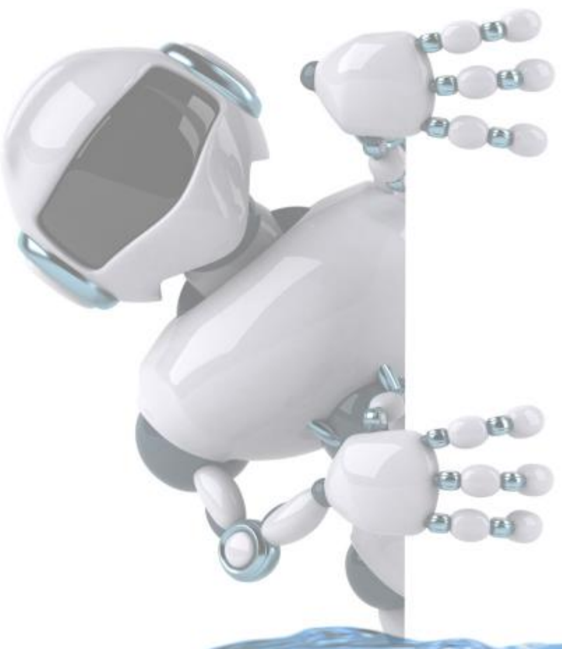
Input: m_0, m_{goal}
Output: $u_{1:s}$

```
initBSP ();
while ‖e_t‖ > thr_2 do
    (ū_{1:s}, m̄_{1:s}) ← CreatePlan(m_t, m_goal);
    for i = iCnt to s do
        t = GetSystemTime();
        u_t ← LQR(ū_t, m̄_t, m_t);
        z_t ← MeasureBlob3Dposition();
        e_t ← m_goal − z_t + ξ;
        m_{t+1} ← EKF(m_t, u_t, z_t);
        if ‖m̄_t − m_t‖ > thr_1 then
            rplCnt ← rplCnt + 1;
            if rplCnt%5 == 0 then
                iCnt ← 1;
            else
                iCnt ← i;
            break;
        DriveEE (u_t);
```

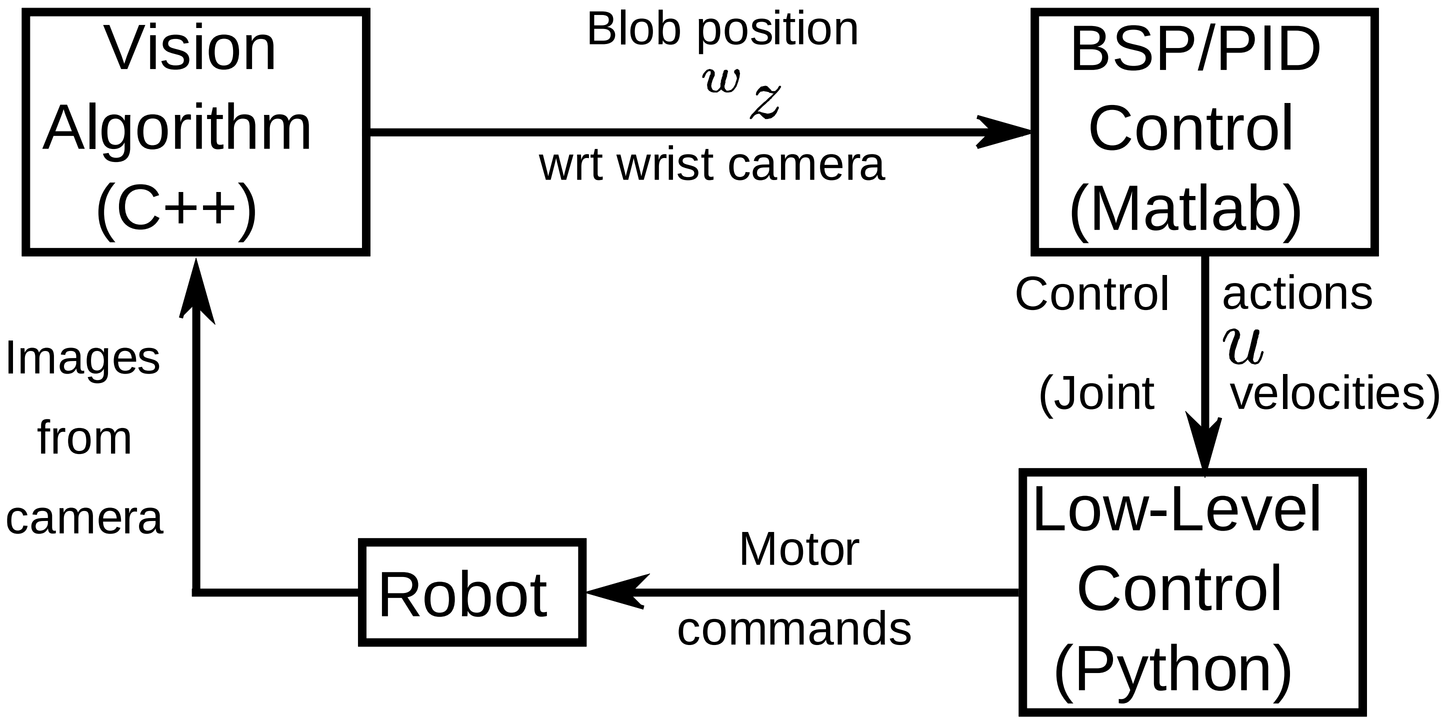
VS

Input: p_0, p_{goal}
Output: u_t

```
initPID ();
while ‖e_t‖ > thr_2 do
    t = GetSystemTime();
    dt = t − t';
    z_t ← MeasureBlob3Dposition();
    e_t ← p_goal − z_t + ξ;
    e_t∫ = e_t∫ + e_t dt;
    u_t = k_p e_t + k_i e_t∫ + k_d (e_t − e_t')/dt;
    t' = t;
DriveEE (u_t);
```



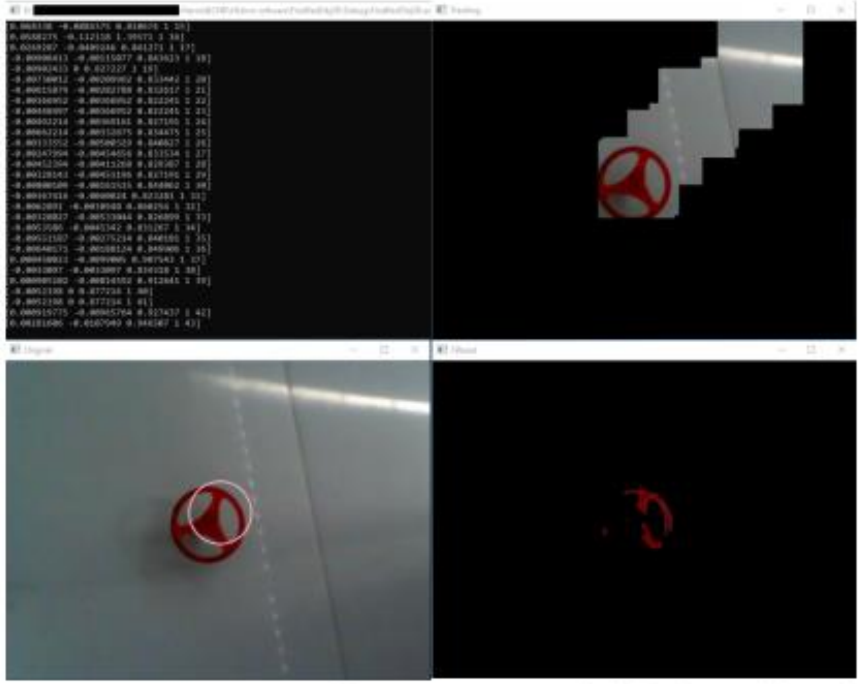
H2Arm software architecture

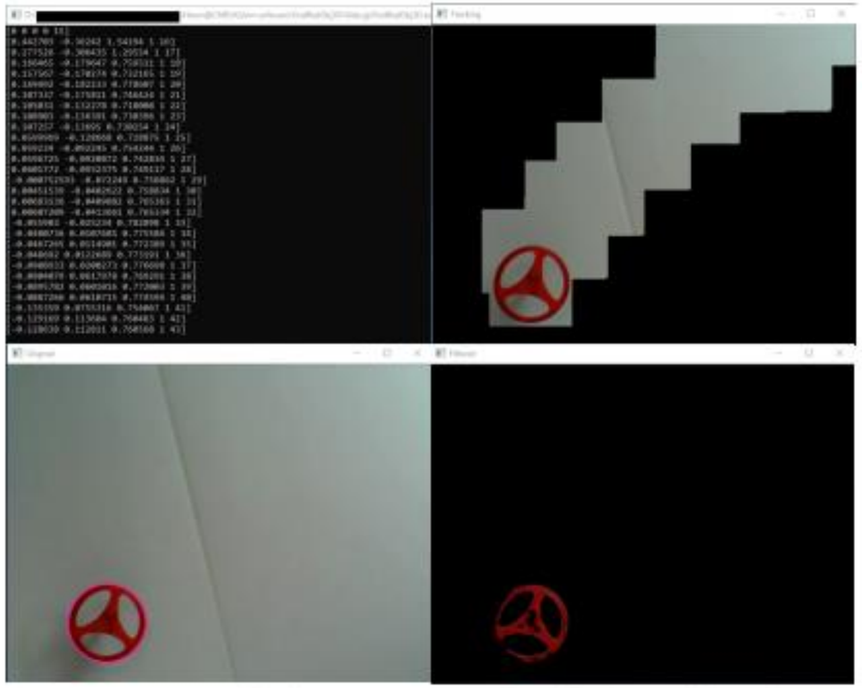


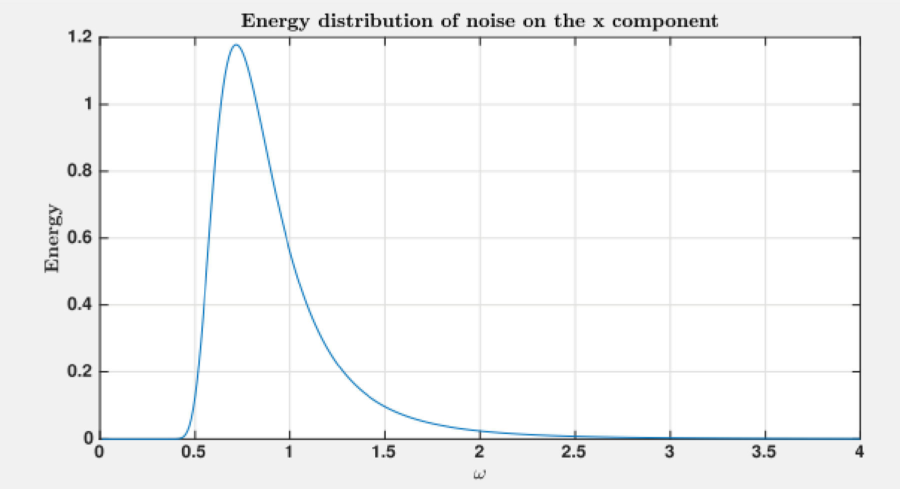
- Vision algorithm and low-level control are the same for each experimental run
- Blocks communicate through standard sockets

Additional Noise

Test Campaign	Calib	Blob	Noise	Cartesian axes wrt $< c >$	w
First	Yes	Fixed	No	–	0
		Moving	No	–	0
			Yes	$2,\ x-y$	0.05
			Yes	$3,\ x-y-z$	0.05
			Yes	$3,\ x-y-z$	0.025
Second	Yes	Fixed	No	–	0
		Moving	Yes	$3,\ x-y-z$	0.025
			Yes	$3,\ x-y-z$	0.035
Third	No	Fixed	No	–	0
		Moving	Yes	$3,\ x-y-z$	0.035


(a) *Estimation in a successful BSP experiment (fixed blob, no additional noise)*


(b) *Estimation in a failed PID experiment (fixed blob, no additional noise)*



- Additional noise distributed as the Pierson-Moskowitz spectrum
- Environmental noise (e.g. illumination)

Results at a glance

First test campaign

Fixed blob, no noise			Moving blob, no noise			Moving blob, 2ax-noise w = 0.05			Moving blob, 3ax-noise w = 0.025			Moving blob, 3ax-noise w = 0.05			TOTAL		
BSP control																	
Good	Tot	%	Good	Tot	%	Good	Tot	%	Good	Tot	%	Good	Tot	%	Good	Tot	%
8	11	**72.7**	5	5	**100**	10	10	**100**	15	15	**100**	21	22	**95.5**	59	63	**93.7**
PID control																	
Good	Tot	%	Good	Tot	%	Good	Tot	%	Good	Tot	%	Good	Tot	%	Good	Tot	%
4	12	**33.3**	5	6	**83.3**	13	30	**43.3**	3	11	**27.3**	1	12	**8.3**	26	71	**36.6**

Second test campaign

Fixed blob, no noise			Moving blob, 3ax-noise w = 0.025			Moving blob, 3ax-noise w = 0.035			TOTAL		
BSP control - optimized system											
Good	Tot	%	Good	Tot	%	Good	Tot	%	Good	Tot	%
17	19	**89.5**	6	6	**100.0**	6	6	**100.0**	29	31	**93.5**
PID control - optimized system											
Good	Tot	%	Good	Tot	%	Good	Tot	%	Good	Tot	%
12	15	**80.0**	2	6	**33.3**	3	6	**50.0**	17	27	**63.0**

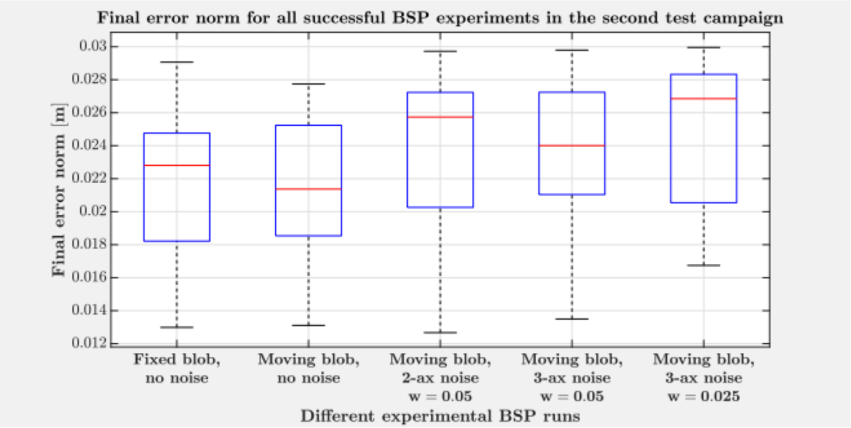
Third test campaign

Fixed blob, no noise			Moving blob, 3ax-noise w = 0.035			TOTAL		
BSP control - uncalibrated system								
Good	Tot	%	Good	Tot	%	Good	Tot	%
4	5	**80.0**	5	5	**100.0**	9	10	**90.0**
PID control - uncalibrated system								
Good	Tot	%	Good	Tot	%	Good	Tot	%
0	5	**0.0**	1	7	**14.3**	1	12	**8.3**

- BSP
  - *97* successful experiments out of *104* (*93.3%*)
  - *109.6* seconds average execution time
- PID
  - *44* successful experiments out of *110* (*40%*)
  - *31.5* seconds average execution time

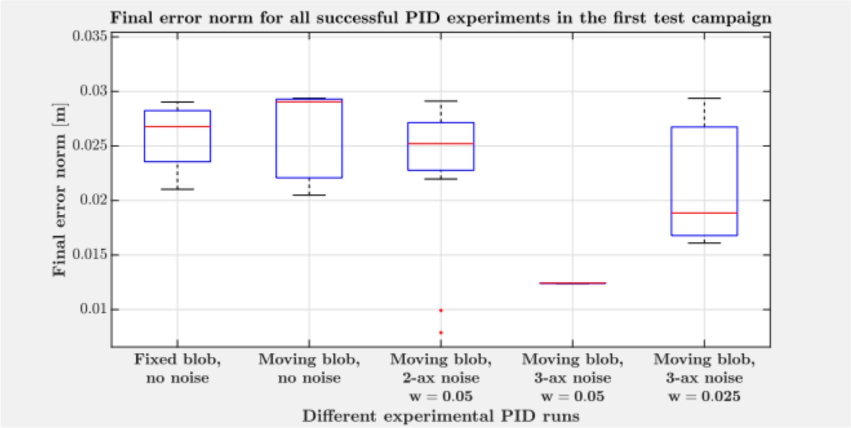
Results of the first test campaign

72.7% 100% 100% 100% 95.5%

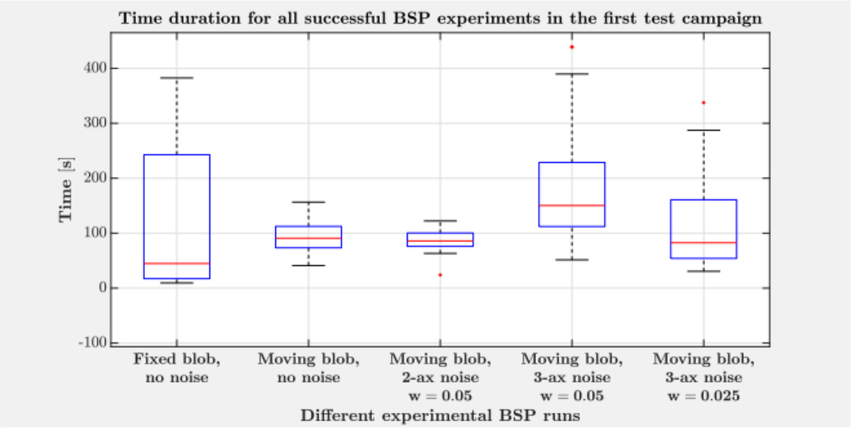


(a) *Final error norm for all BSP successful experiments*

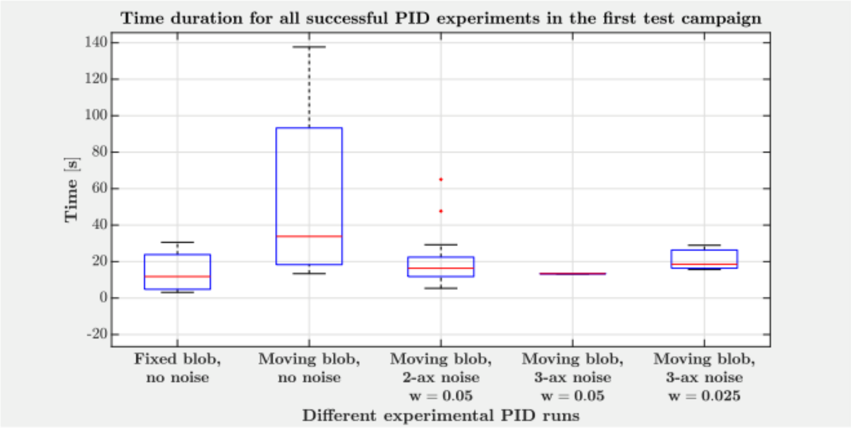
33.3% 83.3% 43.3% 27.3% 8.3%



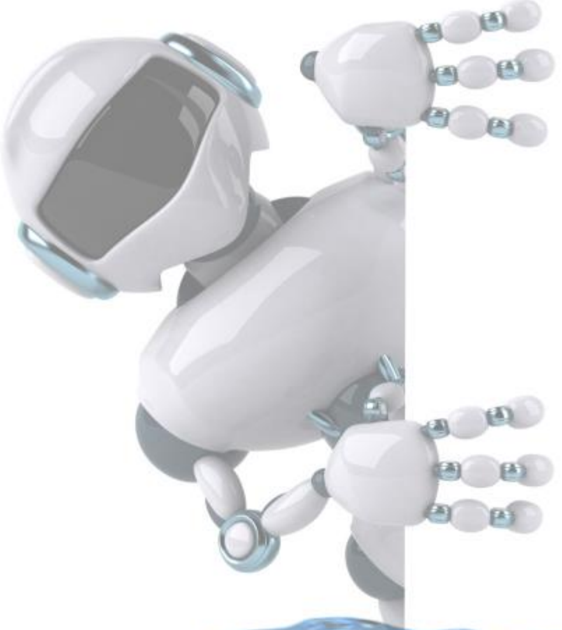
(b) *Final error norm for all PID successful experiments*



(c) *Time duration for all successful BSP experiments*

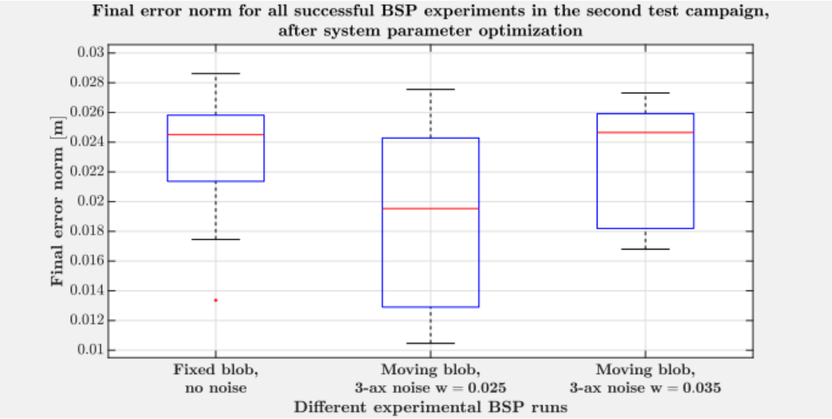


(d) *Time duration for all successful PID experiments*



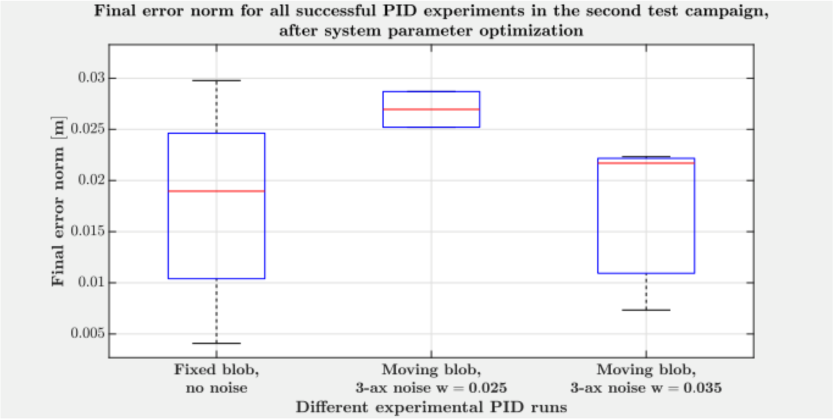
Results of the second test campaign

89.5% 100% 100%

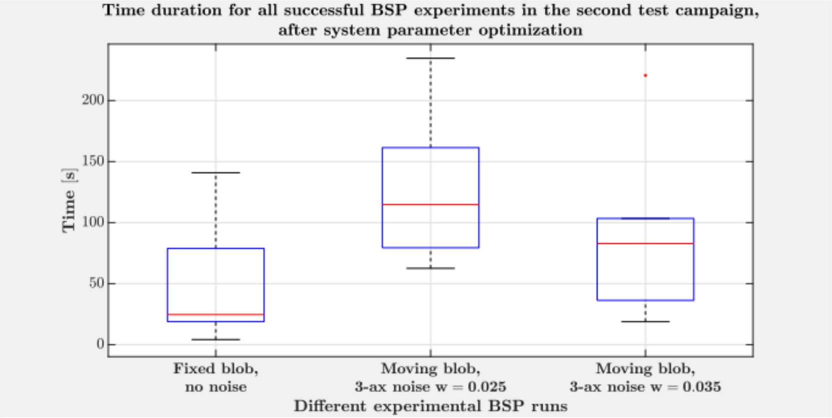


(a) *Final error norm for all BSP successful experiments*

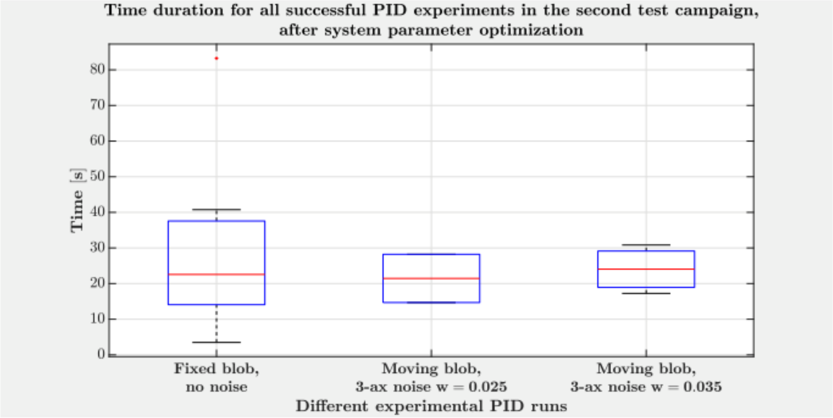
80% 33.3% 50%



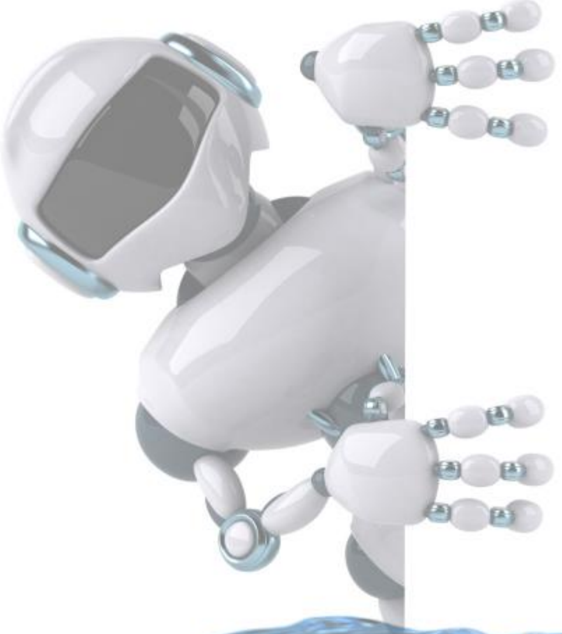
(b) *Final error norm for all PID successful experiments*



(c) *Time duration for all successful BSP experiments*

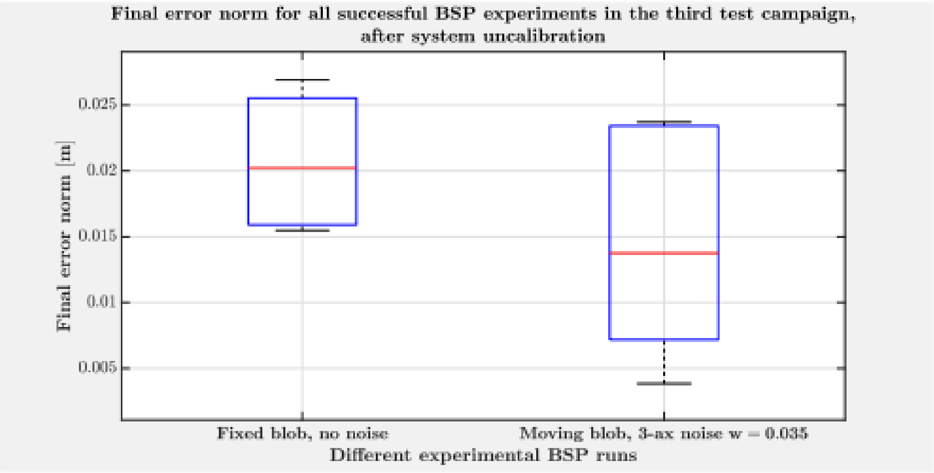


(d) *Time duration for all successful PID experiments*

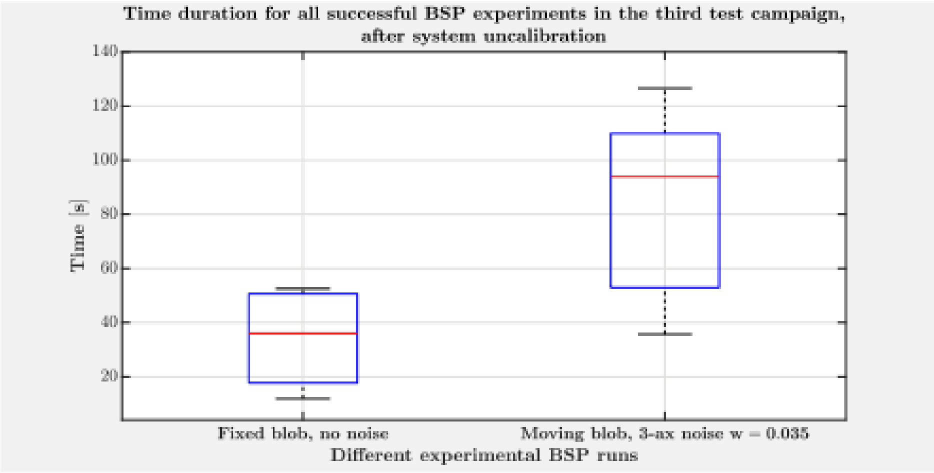


Results of the third test campaign

80% 100%

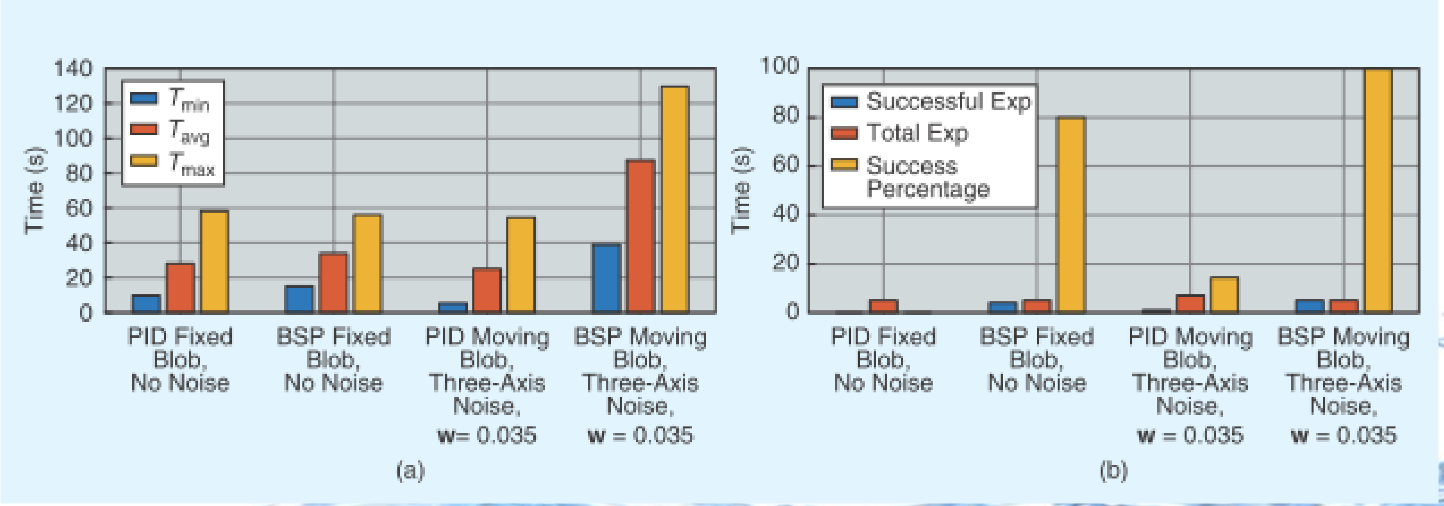


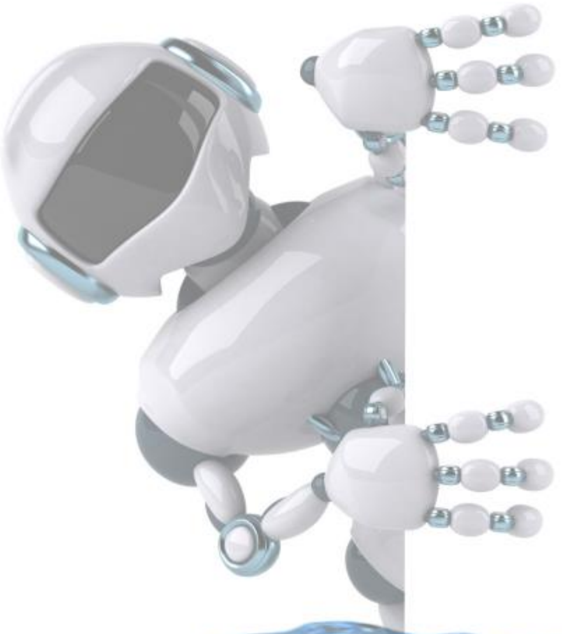
(a) *Final error norm for all BSP successful experiments*



(b) *Time duration for all successful BSP experiments*

No plot on final error norm for PID because just one experiment for each PID category was successful



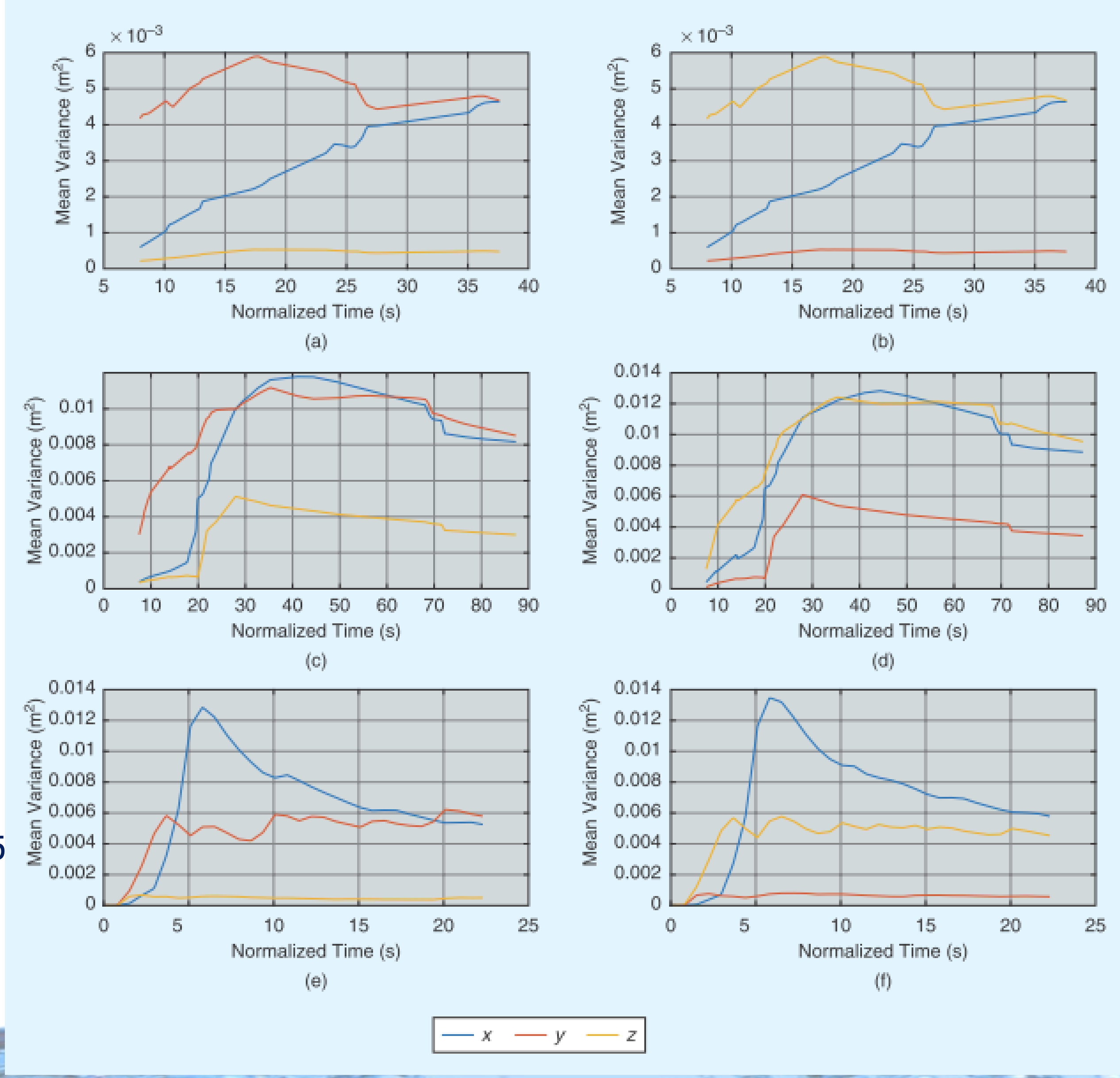


Results of the third test campaign – successful runs

BSP fixed blob no noise

BSP moving blob w=0.035

PID moving blob w=0.035

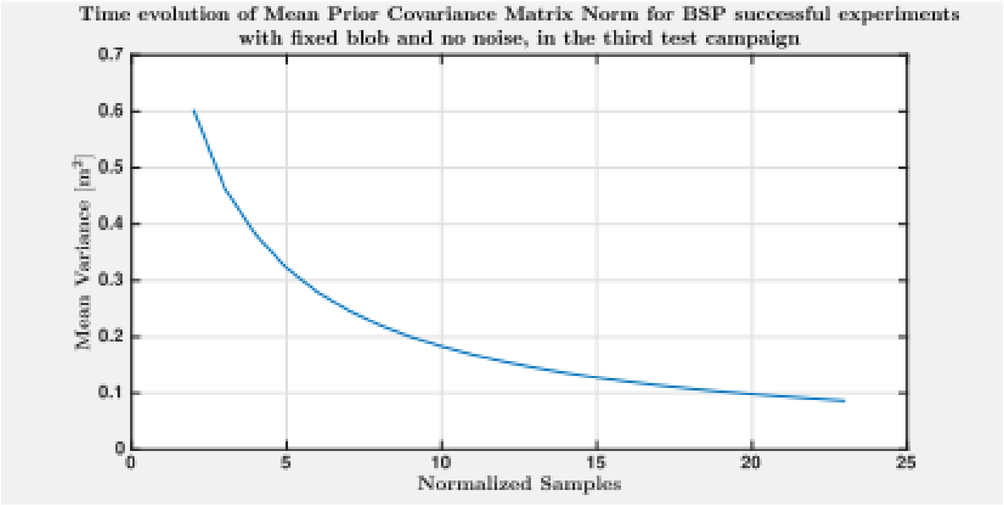


The time evolution of mean posterior variance of the blob position measurement (left column) and of the end-effector Cartesian error (right column) throughout successful experiments in the different categories of the third test campaign. No PID run in the "fixed blob, no noise" case was successful. As expected, the graphs in the right column are very similar to those in the left column.

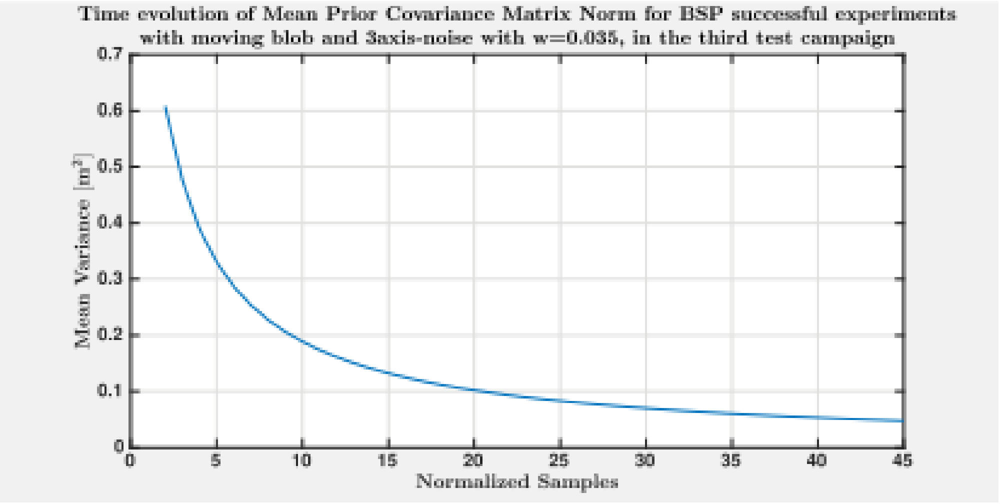
(a) and (b) BSP with fixed blob, each with no noise. (c) and (d) BSP moving blob, each with three-axis noise, w = 0.035 (e) and (f)

PIDs with moving blobs, each with three-axis noise, [Just one experiment in each of (e) and (f) was successful.]

Results of the third test campaign – successful runs

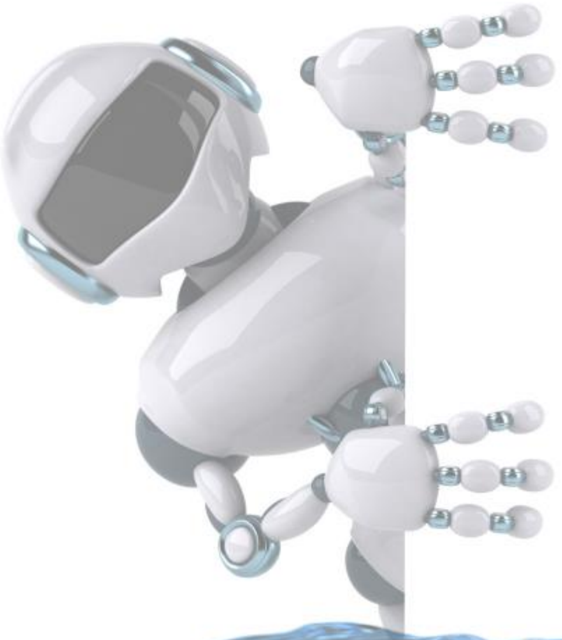


(a) *BSP fixed blob no noise*

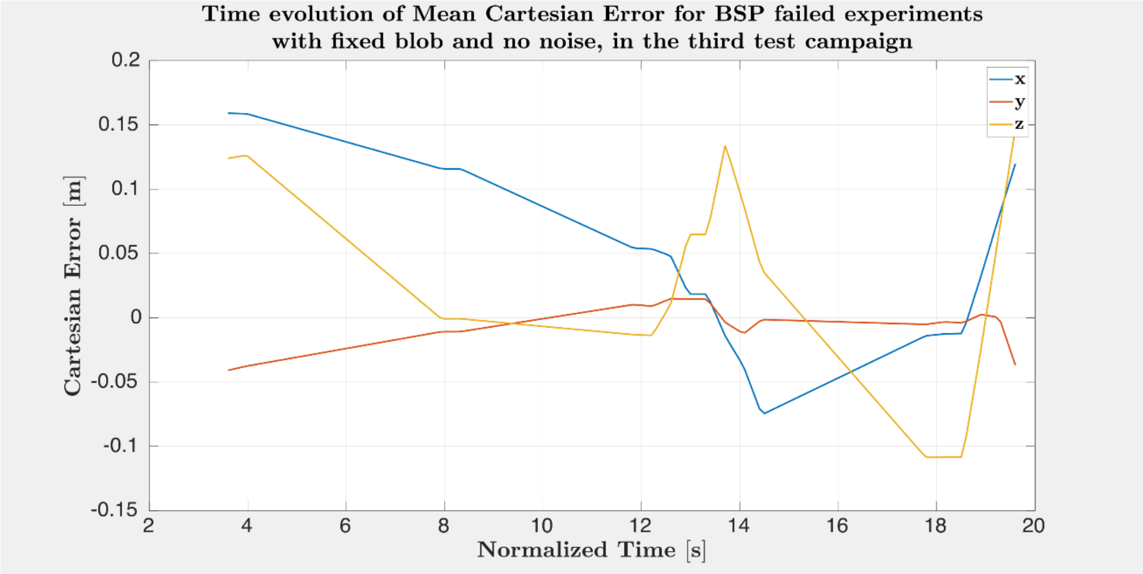


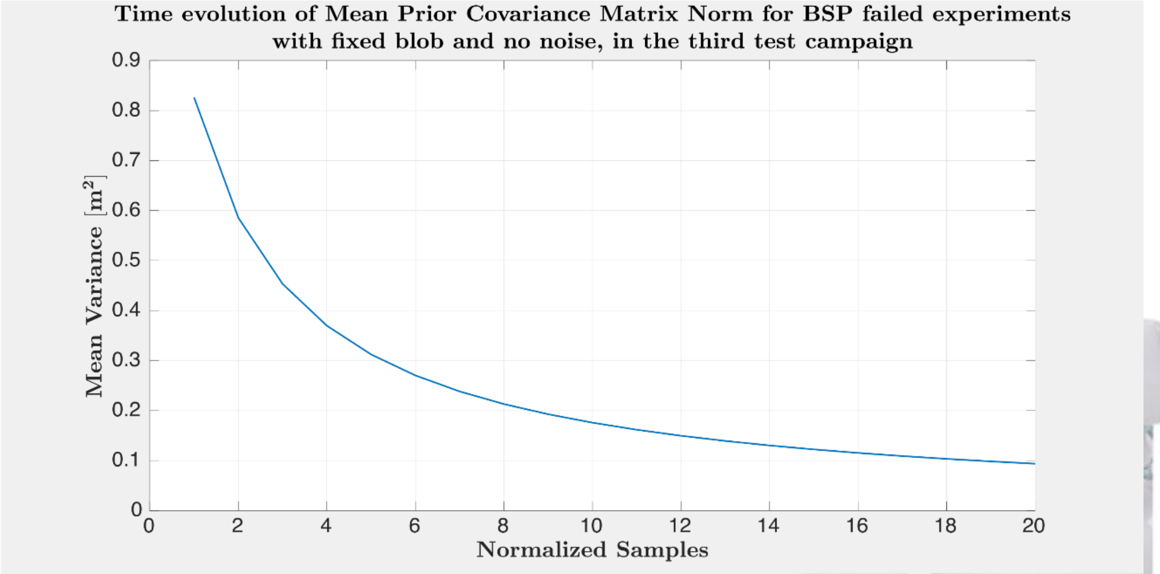
(b) *BSP moving blob 3-ax noise*, $\mathbf{w} = 0.035$.

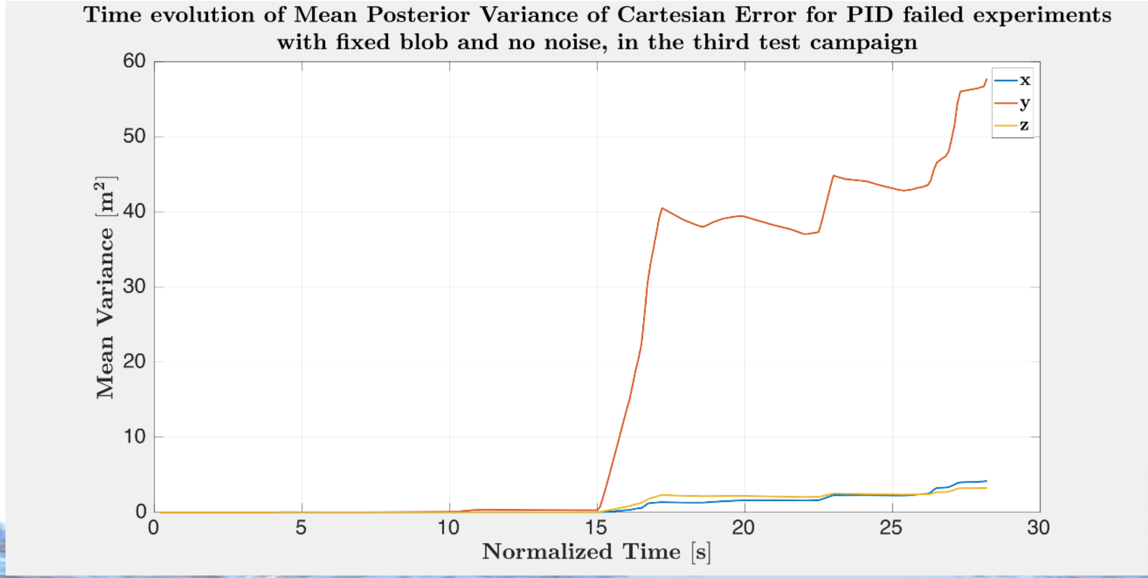
Time evolution of mean prior covariance matrix norm throughout successful BSP experiments in the different categories of the third test campaign.

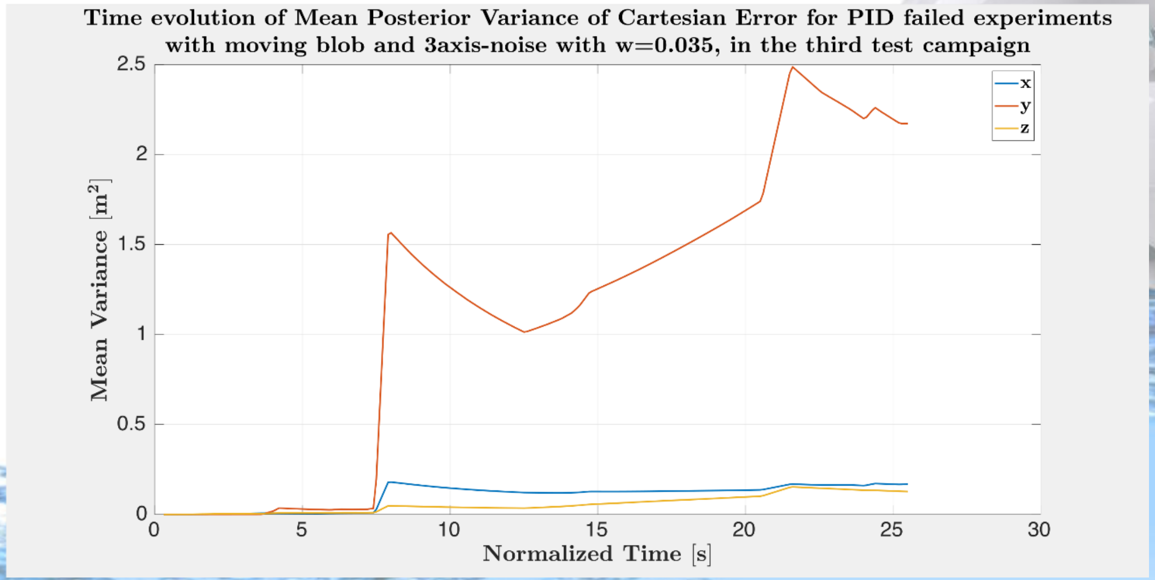


Results of the third test campaign – failed runs





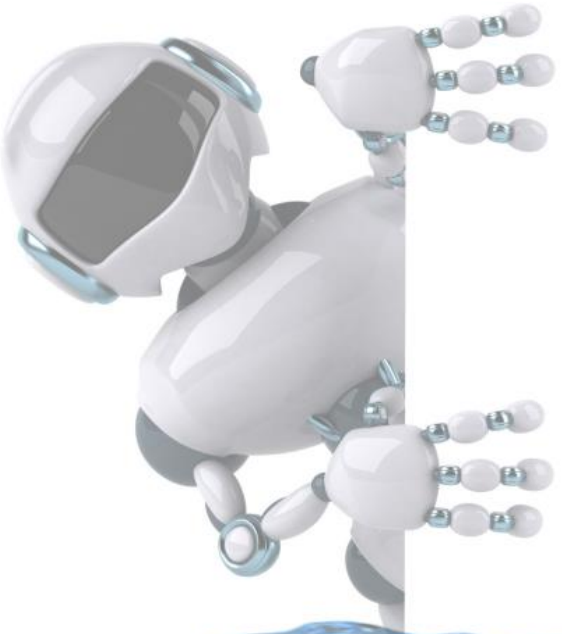




Discussion of results

Thanks to the experiment reproducibility it was possible to fairly compare BSP and PID:

- **Effects of noise:**
  - benefit from randomization
  - low precision of the vision system
- **Effects of LLC uncalibration:**
  - BSP: 9 successful runs over 10
  - PID: 1 successful run over 12
- **Effects of Measurement Variance on Experiments**
  - vision-based measurements intrinsically affected by noise
  - BSP enhances the measurement process with its smoother motion
  - BSP mean measure variance more homogeneous wrt PID
  - BSP control strategy limits unnecessary vibrations and large spikes on the motors



Lesson learned

- PID surely works for **heavy**, **rigid** and **very accurate** robotic structures, but it is not suitable for soft or lightweight and non-reapeatable robots
- BSP, on the contrary, performs well on systems affected by high noise and uncertainty
- PID is faster in task execution than BSP
- BSP is computationally heavy
- Reproducibility allows to highlight significant behaviours of the system, and to fully investigate the trigger causes

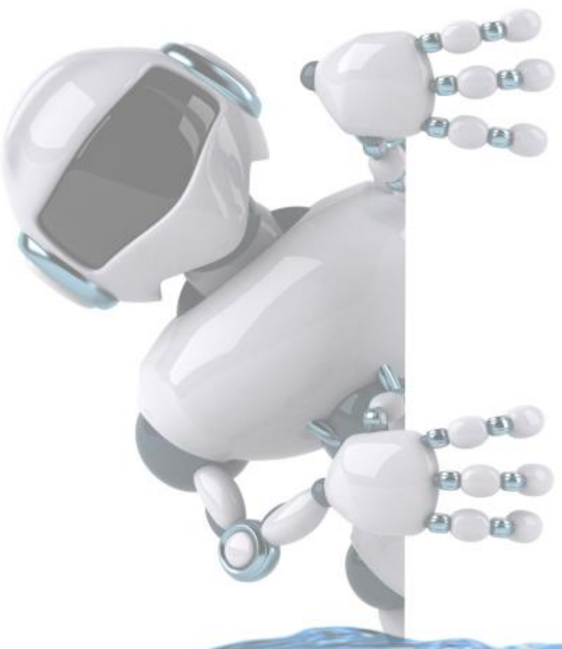
Reproducible articles

Three main components:

1. **Main article**:
   General idea, motivations, high-level description of the work, results in brief
2. **Supplemental material**:
   Methodology, full results, discussion, experiment user guide
3. **Data and code repository**
   Implemented code, logged datasets, user guide to experiment, eventually hardware specs

H2Arm - BSP vs PID experiments – Supplemental Material

- *Introduction* and *Problem statement*
- *Materials and Methods* (system setup, software components, control strategies)
- *Results* (experiments, disturbance and noise modeling, description of test campaigns)
- *Discussion* (BSPvsPID comparison, effects of noise, effects of motor uncalibration, effects of measurements variance)
- *User guide* (hardware and software setup, material organization on DataPort, changing experiment parameters, launching procedure)



Supplemental information

Model A	Param set							
	1		2		3		4	
	τ_j	K_j	τ_j	K_j	τ_j	K_j	τ_j	K_j
J_1	0.001	4	0.001	4	0.001	4	0.001	4
J_2	0.001	2	0.001	2	0.001	3	0.001	4
J_3	0.01	1	0.01	1	0.001	1	0.001	1
J_4	0.01	8	0.01	10	0.001	10	0.001	10

Model B	Param set							
	1				2			
	$\tau_{j,1}$	$\tau_{j,2}$	working area	K_j	$\tau_{j,1}$	$\tau_{j,2}$	working area	K_j
J_1	0.0001	0.001	$\dot{q}_1 < \tau_{1,1}$	0	0.0001	0.001	$\dot{q}_1 < \tau_{1,1}$	0
			$\tau_{1,1} < \dot{q}_1 < \tau_{1,2}$	4			$\tau_{1,1} < \dot{q}_1 < \tau_{1,2}$	4
			$\dot{q}_1 > \tau_{1,2}$	3			$\dot{q}_1 > \tau_{1,2}$	3
J_2	0.0001	0.001	$\dot{q}_2 < \tau_{2,1}$	0	0.0001	0.001	$\dot{q}_2 < \tau_{2,1}$	0
			$\tau_{2,1} < \dot{q}_2 < \tau_{2,2}$	4			$\tau_{2,1} < \dot{q}_2 < \tau_{2,2}$	5
			$\dot{q}_2 > \tau_{2,2}$	3			$\dot{q}_2 > \tau_{2,2}$	4
	τ_j		K_j		τ_j		K_j	
J_3	0.01		1		0.01		1	
J_4	0.01		10		0.01		10	

Model C	Param set									
	1					2				
	$\tau_{j,1}$	$\tau_{j,2}$	working area	K_j		$\tau_{j,1}$	$\tau_{j,2}$	working area	K_j	
				$\dot{q}_j > 0$	$\dot{q}_j < 0$				$\dot{q}_j > 0$	$\dot{q}_j < 0$
J_1	0.0001	0.001	$\dot{q}_1 < \tau_{1,1}$	0		0.0001	0.001	$\dot{q}_1 < \tau_{1,1}$	0	
			$\tau_{1,1} < \dot{q}_1 < \tau_{1,2}$	2	5			$\tau_{1,1} < \dot{q}_1 < \tau_{1,2}$	1	4
			$\dot{q}_1 > \tau_{1,2}$	1	4			$\dot{q}_1 > \tau_{1,2}$	1	3
J_2	0.0001	0.001	$\dot{q}_2 < \tau_{2,1}$	0		0.0001	0.001	$\dot{q}_2 < \tau_{2,1}$	0	
			$\tau_{2,1} < \dot{q}_2 < \tau_{2,2}$	4	3			$\tau_{2,1} < \dot{q}_2 < \tau_{2,2}$	6	5
			$\dot{q}_2 > \tau_{2,2}$	3	2			$\dot{q}_2 > \tau_{2,2}$	5	4
J_3	τ_j		K_j			τ_j		K_j		
	0.01		1			0.01		1		
J_4	τ_j		K_j			τ_j		K_j		
	0.01		10			0.01		6		

Model D	Param set									
	1					2				
	$\tau_{j,1}$	$\tau_{j,2}$	working area	K_j		$\tau_{j,1}$	$\tau_{j,2}$	working area	K_j	
				$\dot{q}_j > 0$	$\dot{q}_j < 0$				$\dot{q}_j > 0$	$\dot{q}_j < 0$
J_1	0.0001	0.001	$\dot{q}_1 < \tau_{1,1}$	0		0.0001	0.001	$\dot{q}_1 < \tau_{1,1}$	0	
			$\tau_{1,1} < \dot{q}_1 < \tau_{1,2}$	2	5			$\tau_{1,1} < \dot{q}_1 < \tau_{1,2}$	2	5
			$\dot{q}_1 > \tau_{1,2}$	1	4			$\dot{q}_1 > \tau_{1,2}$	1	4
J_2	0.0001	0.001	$\dot{q}_2 < \tau_{2,1}$	0		0.0001	0.001	$\dot{q}_2 < \tau_{2,1}$	0	
			$\tau_{2,1} < \dot{q}_2 < \tau_{2,2}$	4				$\tau_{2,1} < \dot{q}_2 < \tau_{2,2}$	3	
			$\dot{q}_2 > \tau_{2,2}$	3				$\dot{q}_2 > \tau_{2,2}$	2	
J_3	τ_j		K_j			τ_j		K_j		
	0.01		1			0.01		1		
J_4	τ_j		K_j			τ_j		K_j		
	0.01		10			0.01		10		

Supplemental information

BSP

Test Campaign	Category	# succ runs	# tot runs	t_{VIS} [ms]	$t_{LLC_{stop}}$ [ms]	$t_{LLC_{go}}$ [ms]	LLC model	param set
First	Fixed blob, no noise	3	6	500	500	150	A	1
		3	3	500	500	150	A	1
		2	2	500	5	150	A	2
	Moving blob, no noise	5	5	500	500	150	A	2
	Moving blob, 2ax-noise, $\mathbf{w} = 0.05$	10	10	250	250	150	A	2
	Moving blob 3ax-noise, $\mathbf{w} = 0.05$	5	5	250	250	150	A	2
		1	1	250	250	150	A	3
		1	1	250	250	150	A	4
		1	1	250	250	150	B	2
		13	14	250	250	150	B	1
	Moving blob, 3ax-noise, $\mathbf{w} = 0.025$	15	15	250	250	150	B	1
Second	Fixed blob, no noise, optimized system	10	12	250	250	150	B	1
		1	1	250	250	150	D	1
		1	1	250	250	150	D	2
		5	5	250	250	150	C	1
	Moving blob, 3ax-noise, $\mathbf{w} = 0.025$, optimized system	6	6	250	250	150	C	1
	Moving blob, 3ax-noise, $\mathbf{w} = 0.035$, optimized system	6	6	250	250	150	C	1
Third	Fixed blob, no noise, uncalibrated system	4	5	250	250	200	C	2
	Moving blob, 3ax-noise, $\mathbf{w} = 0.035$, uncalibrated system	5	5	250	250	200	C	2

PID

Test Campaign	Category	# succ runs	# tot runs	k_p	k_i	k_d	t_{VIS} [ms]	$t_{LLC_{stop}}$ [ms]	$t_{LLC_{go}}$ [ms]	LLC model	param set
First	Fixed blob, no noise	1	1	$\frac{4}{7}$	$\frac{10}{1000}$	$\frac{1}{700}$	500	500	150	A	1
		2	5	$\frac{1}{7}$	$\frac{10}{1000}$	$\frac{1}{700}$	500	500	150	A	1
		1	6	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	500	500	150	A	1
	Moving blob, no noise	5	6	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	500	500	150	A	2
	Moving blob, 2ax-noise, $\mathbf{w} = 0.05$	1	10	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	A	2
		4	4	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	A	2
		5	11	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	A	2
		3	5	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	A	2
	Moving blob 3ax-noise, $\mathbf{w} = 0.05$	0	1	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	B	1
		1	6	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	B	1
		0	5	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	B	1
	Moving blob, 3ax-noise, $\mathbf{w} = 0.025$	0	5	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	B	1
		3	6	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	B	1
Second	Fixed blob, no noise, optimized system	7	10	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	B	1
		5	5	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	C	1
	Moving blob, 3ax-noise, $\mathbf{w} = 0.025$, optimized system	1	2	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	C	1
		1	4	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	C	1
	Moving blob, 3ax-noise, $\mathbf{w} = 0.035$, optimized system	3	6	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	150	C	1
Third	Fixed blob, no noise, uncalibrated system	0	5	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	200	C	2
	Moving blob, 3ax-noise, $\mathbf{w} = 0.035$, uncalibrated system	1	7	$\frac{1}{7}$	$\frac{5}{1000}$	$\frac{1}{700}$	250	250	200	C	2

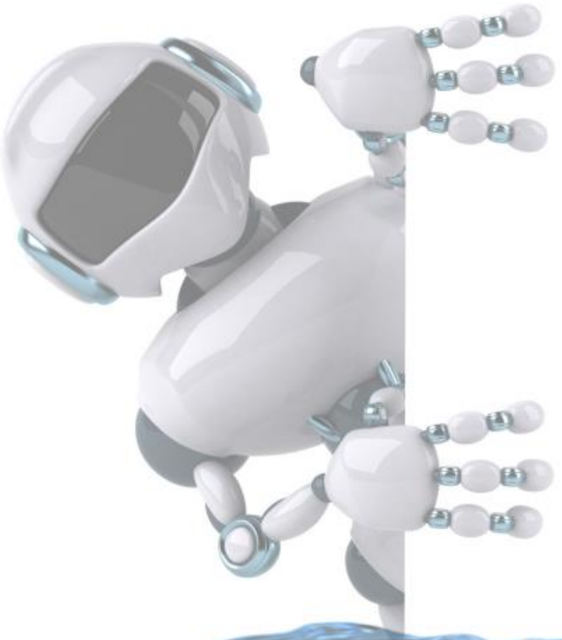
Reproduced results can be easily compared

Code repository

Code, *logged data*, *instruction* and *all details* needed to reproduce experiment should be shared

Repository on

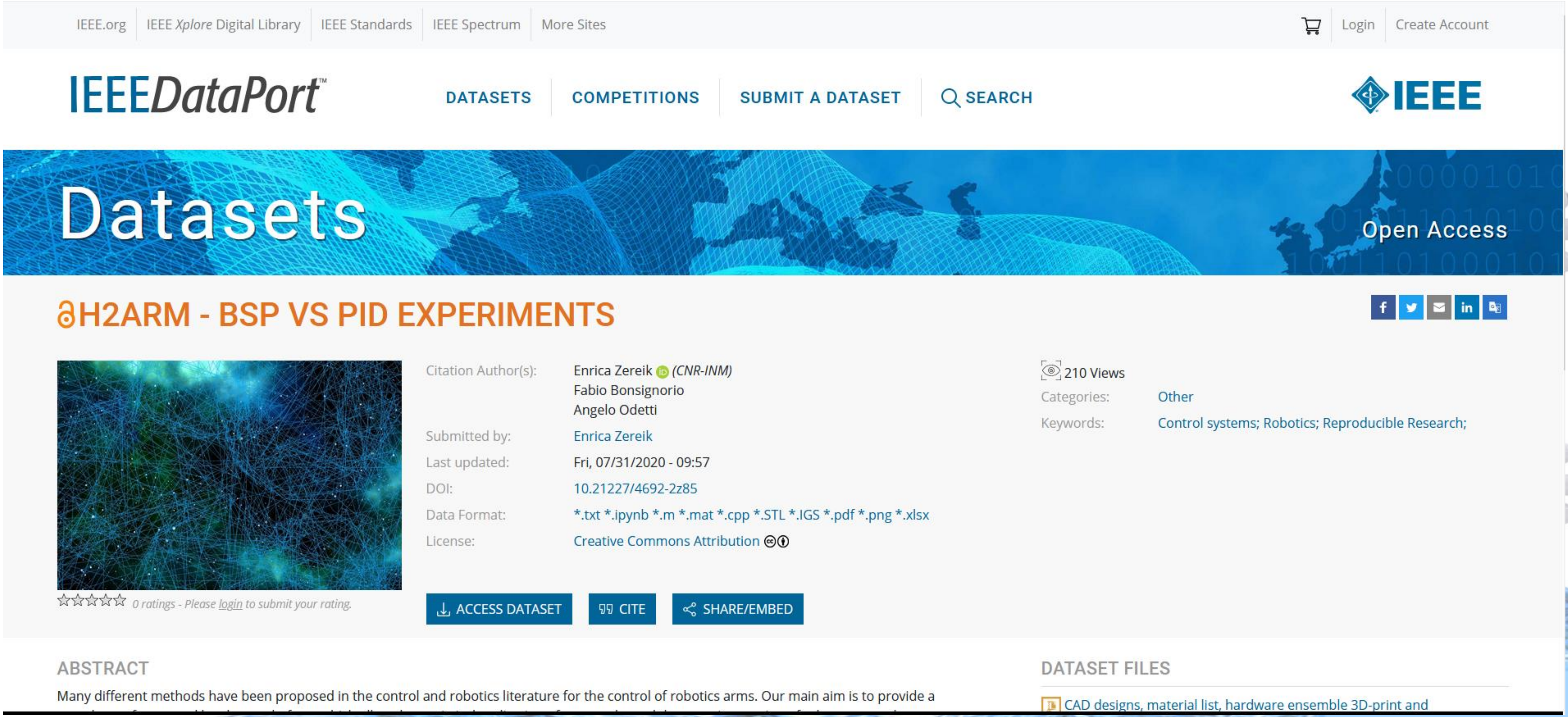
- **CodeOcean**: suitable for simulation, run on remote machine
- **IEEE DataPort**: store many kinds of files, suitable to share logged data, implementation details, instruction…



H2Arm - BSP vs PID experiments

https://ieee-dataport.org/open-access/h2arm-bsp-vs-pid-experiments

IEEE DataPort



IEEE DataPort

Data Format: *.txt *.ipynb *.m *.mat *.cpp *.STL *.IGS *.pdf *.png *.xlsx

License: Creative Commons Attribution

☆☆☆☆☆ *0 ratings - Please login to submit your rating.*

ACCESS DATASET | CITE | SHARE/EMBED

ABSTRACT

Many different methods have been proposed in the control and robotics literature for the control of robotics arms. Our main aim is to provide a complete software and hardware platform which allow the statistical replication of our results and the experimentation of other more or less sophisticated control strategies and algorithms.

In our work, in order to validate our reproducible research platform and provide a template methodology for its usage, we have thoroughly compared in a reproducible way the performance of simple BSP and PID controls when applied to a light-weight, low accuracy and compliant open source robot arm (H2Arm). BSP significantly outperforms PID on this platform, but not w.r.t. to all metrics.The findings are interesting by themselves. They also show how easily statistically weak results can lead to qualitatively wrong conclusions, if you cherry-pick results.

The present dataset contains: i) CAD design to 3D-print the H2Arm; ii) experimental data of comparison BSP vs PID; iii) all the code used to perform the experiments (both BSP and PID).

All the data in this repository are linked (and thoroughly documented in the paper "An Experimental Comparison of BSP and PID Controllers for A Simple Visual-servoing Task on a Low-Accuracy Low-Cost Arm" by Fabio Bonsignorio and Enrica Zereik, accepted for publication in IEEE Robotics and Automation Magazine).

DATASET FILES

- CAD designs, material list, hardware ensemble 3D-print and elements.zip (2.56 MB)
- Data logged in all the experiments (BSP and PID), subdivided by test campaign and by experiment condition (see instruction) LoggedData (.txt and .mat).zip (6.89 MB)
- All code employed in the experiments (as described in the instructions) code.zip (51.46 kB)

LOGIN TO ACCESS DATASET FILES

DOCUMENTATION

Instruction and User Guide (652.81 KB)

QUESTIONS?

✉ Login to Send Author a Private Message

⚠ Report a problem with this Dataset

Find Datasets

Looking for datasets? Search and browse datasets and data competitions. Standard datasets are available to IEEE DataPort subscribers. Open Access datasets are available to all users.

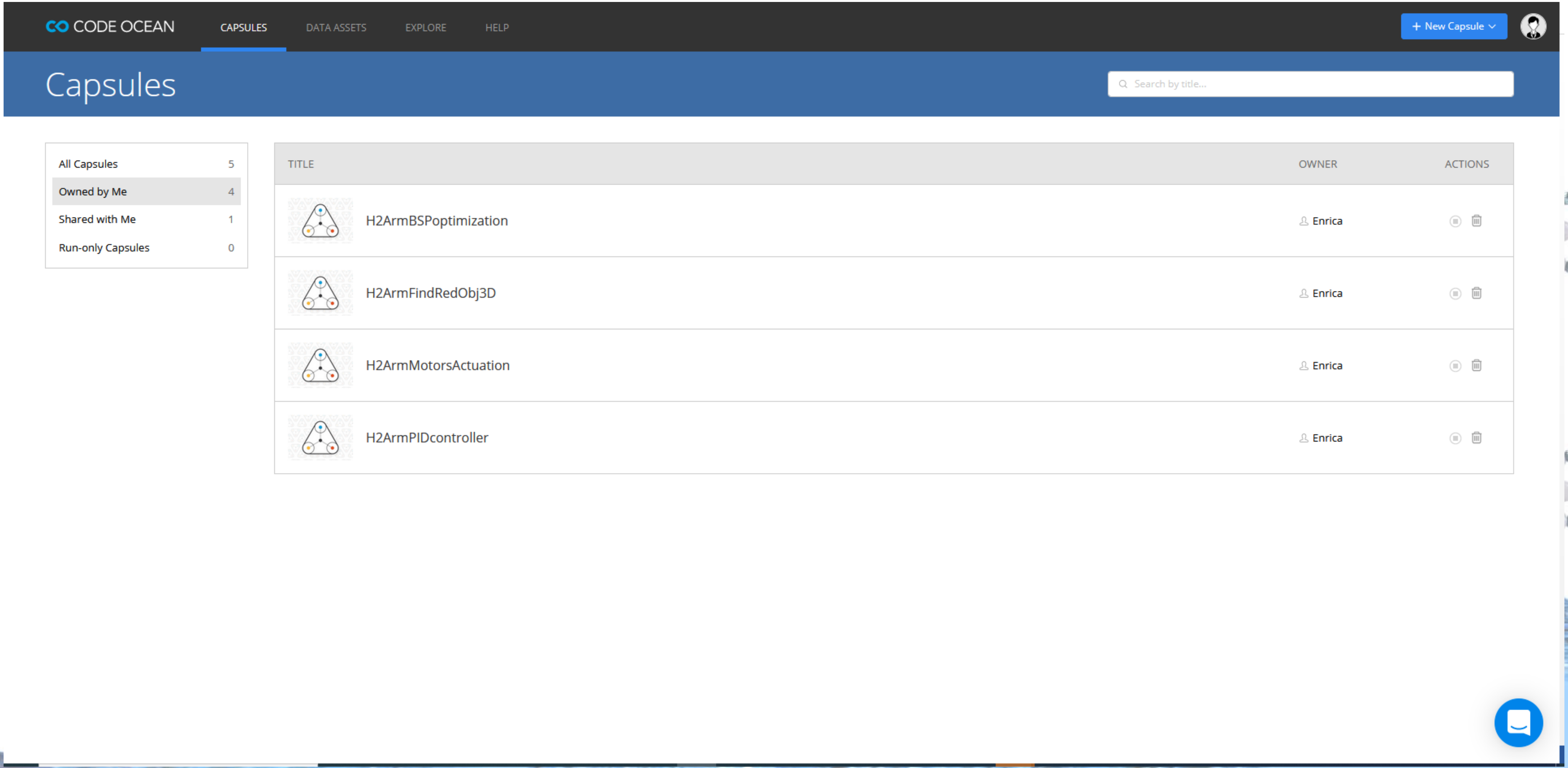
Submit a Dataset

All users may submit a standard dataset up to 2TB free of charge. Submit an Open Access dataset to allow free access to all users, or create a data competition and manage access and submissions.

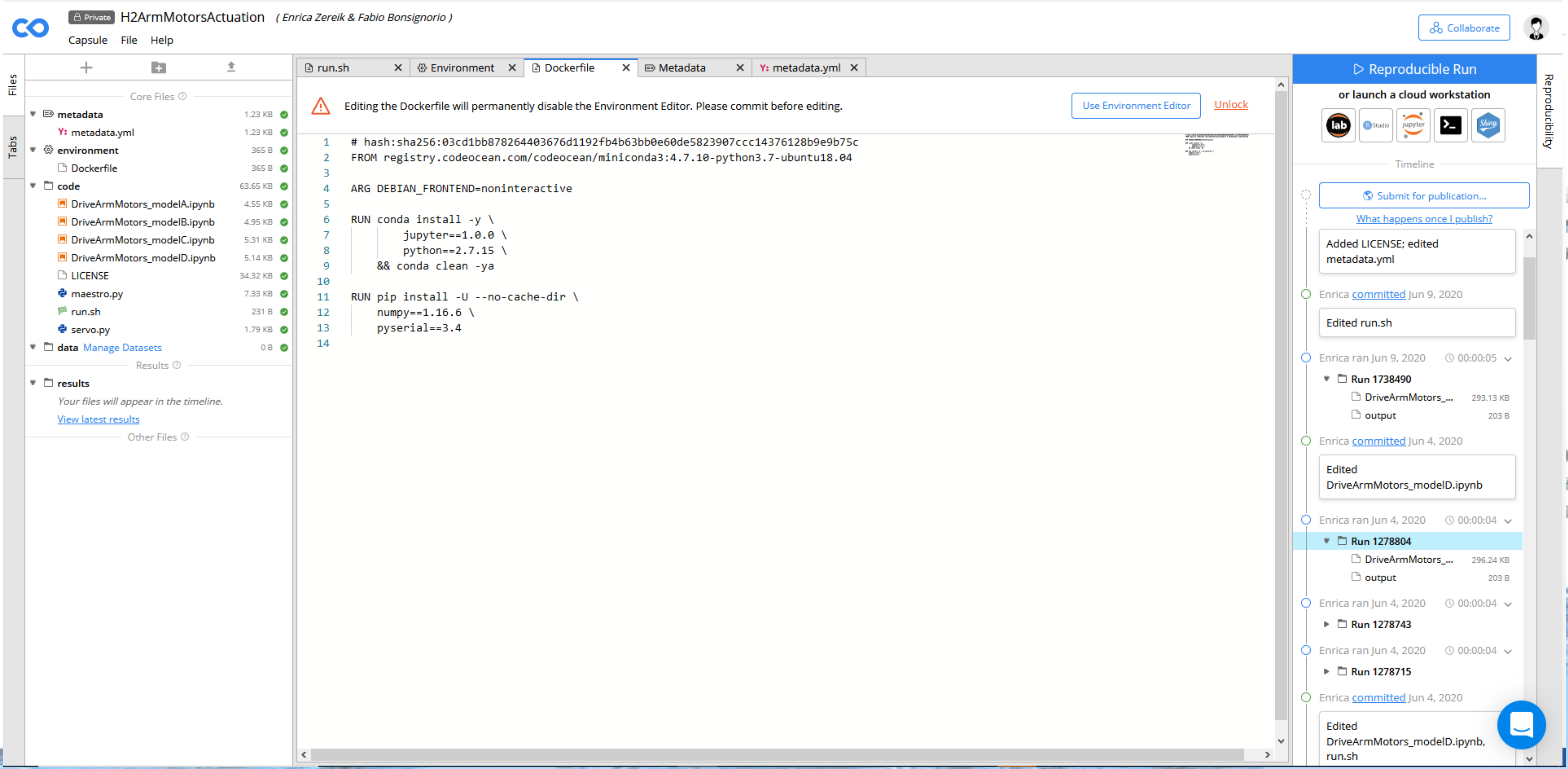
Subscribe to IEEE DataPort

IEEE DataPort Subscribers may download all our datasets or access them directly on AWS. Limited time offer: use coupon code DATAPORT1 for free access. If you use the DATAPORT1 coupon to obtain your free IEEE DataPort

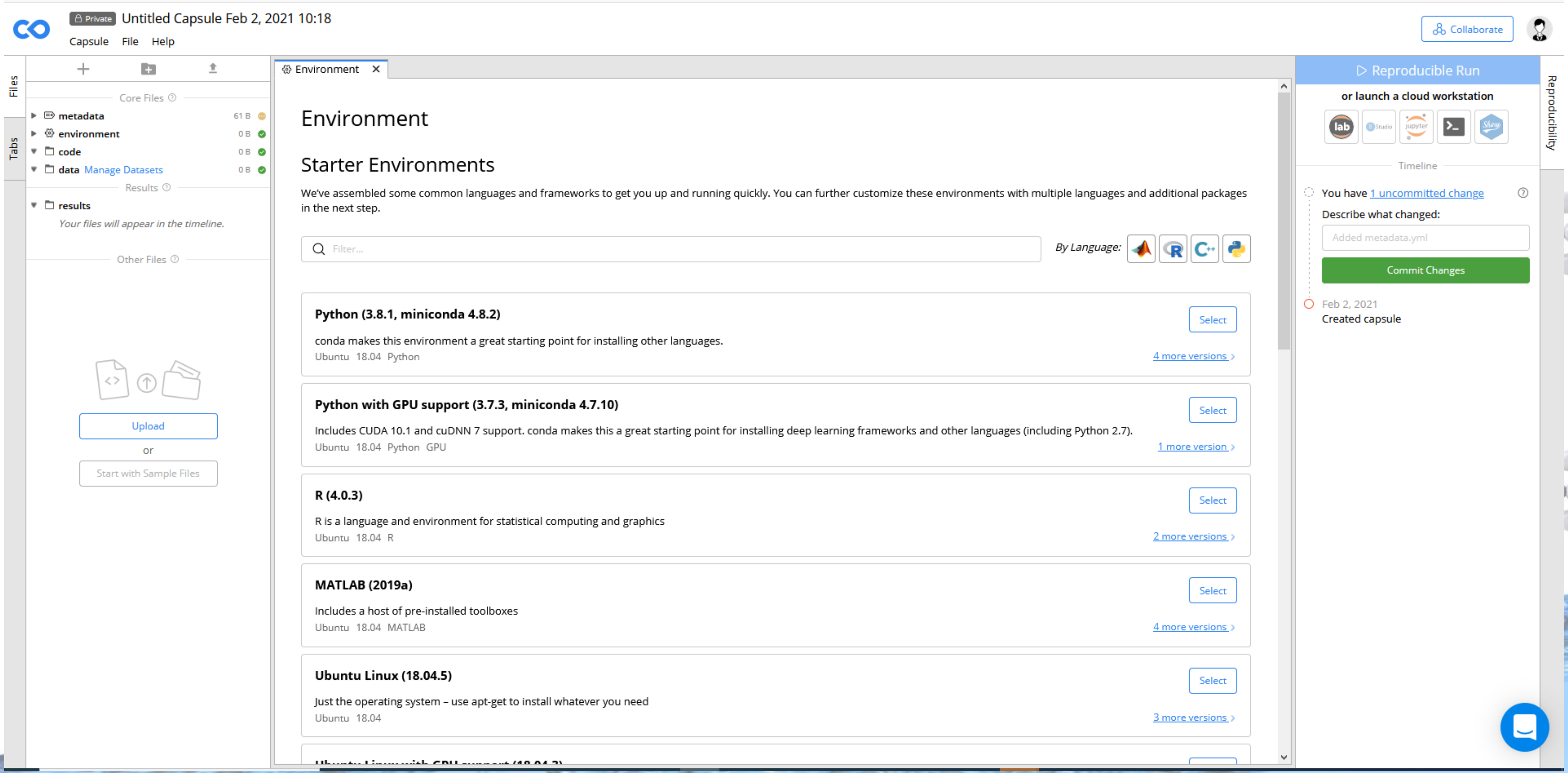
CodeOcean


CODE OCEAN
CAPSULES
DATA ASSETS
EXPLORE
HELP
+ New Capsule
Capsules
Search by title...
All Capsules 5
Owned by Me 4
Shared with Me 1
Run-only Capsules 0
TITLE
OWNER
ACTIONS
H2ArmBSPoptimization
Enrica
H2ArmFindRedObj3D
Enrica
H2ArmMotorsActuation
Enrica
H2ArmPIDcontroller
Enrica

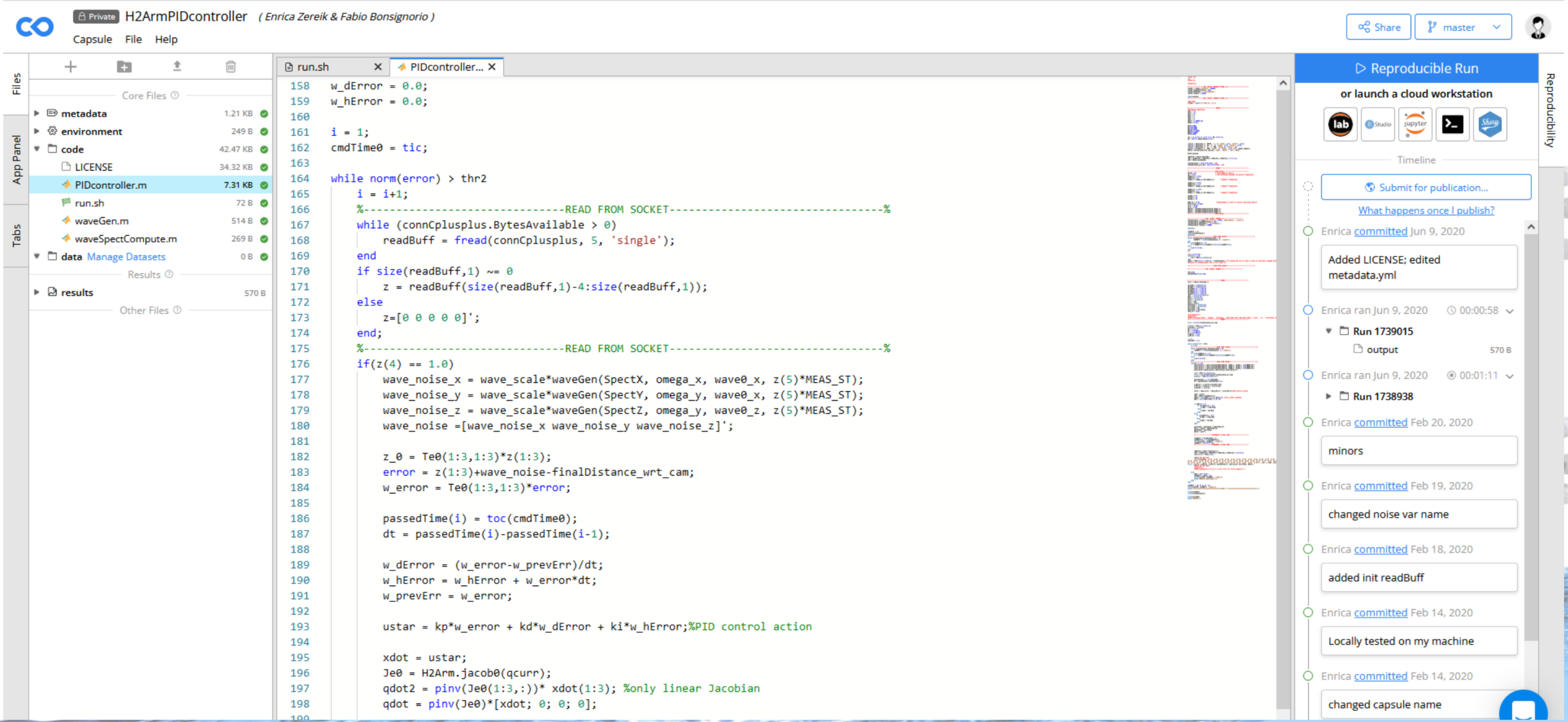
CodeOcean


Private H2ArmMotorsActuation (Enrica Zereik & Fabio Bonsignorio)
Capsule File Help
Collaborate
Files
Tabs
Core Files
metadata 1.23 KB
metadata.yml 1.23 KB
environment 365 B
Dockerfile 365 B
code 63.65 KB
DriveArmMotors_modelA.ipynb 4.55 KB
DriveArmMotors_modelB.ipynb 4.95 KB
DriveArmMotors_modelC.ipynb 5.31 KB
DriveArmMotors_modelD.ipynb 5.14 KB
LICENSE 34.32 KB
maestro.py 7.33 KB
run.sh 231 B
servo.py 1.79 KB
data Manage Datasets 0 B
Results
results
Your files will appear in the timeline.
View latest results
Other Files
run.sh
Environment
Dockerfile
Metadata
metadata.yml
Editing the Dockerfile will permanently disable the Environment Editor. Please commit before editing.
Use Environment Editor
Unlock
hash:sha256:03cd1bb878264403676d1192fb4b63bb0e60de5823907ccc14376128b9e9b75c
FROM registry.codeocean.com/codeocean/miniconda3:4.7.10-python3.7-ubuntu18.04
ARG DEBIAN_FRONTEND=noninteractive
RUN conda install -y \
jupyter==1.0.0 \
python==2.7.15 \
&& conda clean -ya
RUN pip install -U --no-cache-dir \
numpy==1.16.6 \
pyserial==3.4
Reproducible Run
or launch a cloud workstation
lab
jupyter
Timeline
Submit for publication...
What happens once I publish?
Added LICENSE; edited metadata.yml
Enrica committed Jun 9, 2020
Edited run.sh
Enrica ran Jun 9, 2020 00:00:05
Run 1738490
DriveArmMotors_... 293.13 KB
output 203 B
Enrica committed Jun 4, 2020
Edited DriveArmMotors_modelD.ipynb
Enrica ran Jun 4, 2020 00:00:04
Run 1278804
DriveArmMotors_... 296.24 KB
output 203 B
Enrica ran Jun 4, 2020 00:00:04
Run 1278743
Enrica ran Jun 4, 2020 00:00:04
Run 1278715
Enrica committed Jun 4, 2020
Edited DriveArmMotors_modelD.ipynb, run.sh
Reproducibility

CodeOcean


Private
Untitled Capsule Feb 2, 2021 10:18
Capsule
File
Help
Collaborate
Files
Tabs
Core Files
metadata 61 B
environment 0 B
code 0 B
data Manage Datasets 0 B
Results
results
Your files will appear in the timeline.
Other Files
Upload
or
Start with Sample Files
Environment
Environment
Starter Environments
We've assembled some common languages and frameworks to get you up and running quickly. You can further customize these environments with multiple languages and additional packages in the next step.
Filter...
By Language:
Python (3.8.1, miniconda 4.8.2)
conda makes this environment a great starting point for installing other languages.
Ubuntu 18.04 Python
4 more versions
Select
Python with GPU support (3.7.3, miniconda 4.7.10)
Includes CUDA 10.1 and cuDNN 7 support. conda makes this a great starting point for installing deep learning frameworks and other languages (including Python 2.7).
Ubuntu 18.04 Python GPU
1 more version
Select
R (4.0.3)
R is a language and environment for statistical computing and graphics
Ubuntu 18.04 R
2 more versions
Select
MATLAB (2019a)
Includes a host of pre-installed toolboxes
Ubuntu 18.04 MATLAB
4 more versions
Select
Ubuntu Linux (18.04.5)
Just the operating system – use apt-get to install whatever you need
Ubuntu 18.04
3 more versions
Select
Reproducible Run
or launch a cloud workstation
lab
Timeline
You have 1 uncommitted change
Describe what changed:
Added metadata.yml
Commit Changes
Feb 2, 2021
Created capsule
Reproducibility

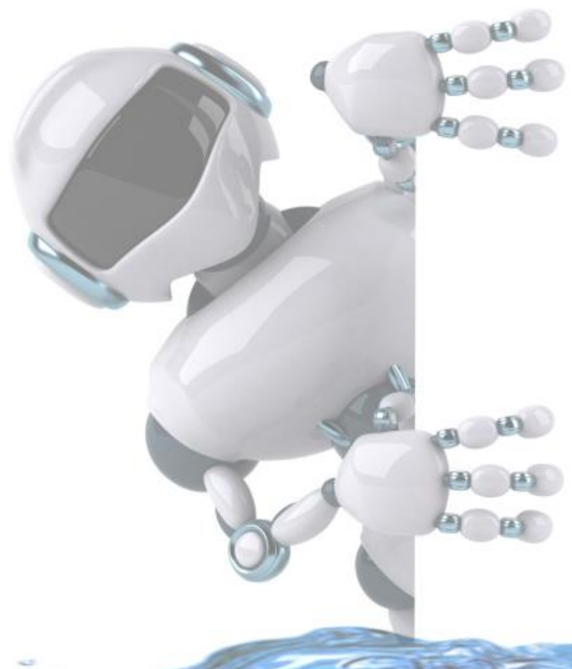
CodeOcean



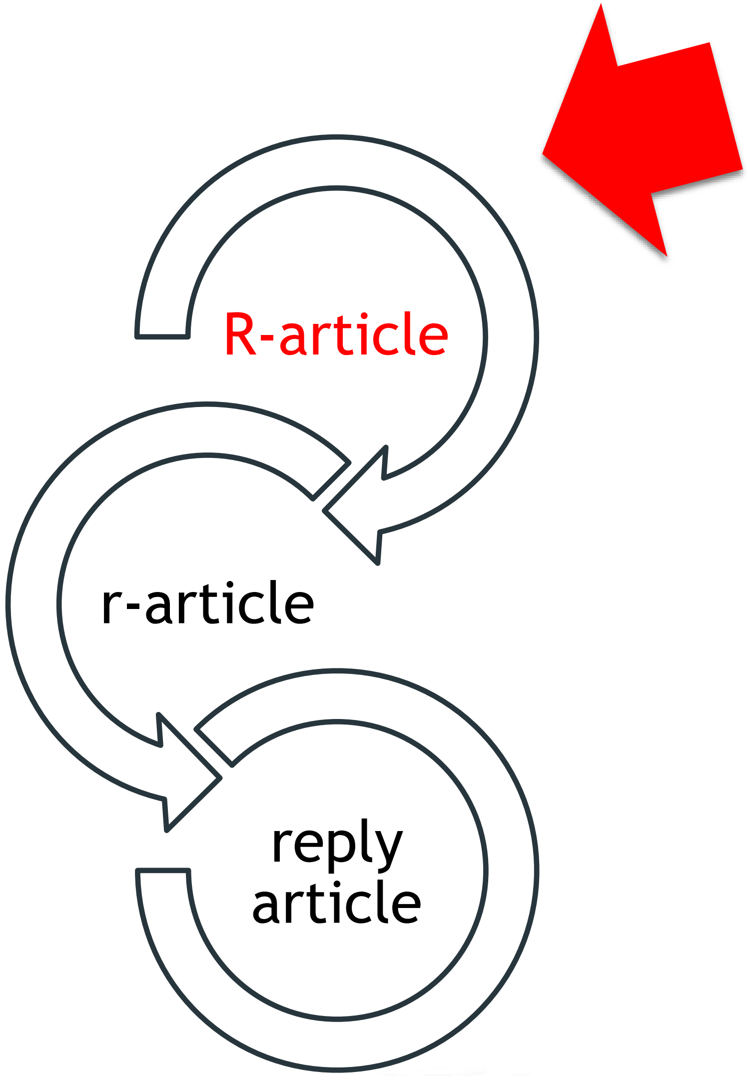
Private
H2ArmPIDcontroller
(Enrica Zereik & Fabio Bonsignorio)
Capsule
File
Help
Share
master
Files
App Panel
Tabs
Core Files
metadata 1.21 KB
environment 249 B
code 42.47 KB
LICENSE 34.32 KB
PIDcontroller.m 7.31 KB
run.sh 72 B
waveGen.m 514 B
waveSpectCompute.m 269 B
data Manage Datasets 0 B
Results
results 570 B
Other Files
run.sh
PIDcontroller...
w_dError = 0.0;
w_hError = 0.0;
i = 1;
cmdTime0 = tic;
while norm(error) > thr2
i = i+1;
%----------------------------READ FROM SOCKET--------------------------------%
while (connCplusplus.BytesAvailable > 0)
readBuff = fread(connCplusplus, 5, 'single');
end
if size(readBuff,1) ~= 0
z = readBuff(size(readBuff,1)-4:size(readBuff,1));
else
z=[0 0 0 0 0]';
end;
%----------------------------READ FROM SOCKET--------------------------------%
if(z(4) == 1.0)
wave_noise_x = wave_scale*waveGen(SpectX, omega_x, wave0_x, z(5)*MEAS_ST);
wave_noise_y = wave_scale*waveGen(SpectY, omega_y, wave0_x, z(5)*MEAS_ST);
wave_noise_z = wave_scale*waveGen(SpectZ, omega_y, wave0_z, z(5)*MEAS_ST);
wave_noise =[wave_noise_x wave_noise_y wave_noise_z]';
z_0 = Te0(1:3,1:3)*z(1:3);
error = z(1:3)+wave_noise-finalDistance_wrt_cam;
w_error = Te0(1:3,1:3)*error;
passedTime(i) = toc(cmdTime0);
dt = passedTime(i)-passedTime(i-1);
w_dError = (w_error-w_prevErr)/dt;
w_hError = w_hError + w_error*dt;
w_prevErr = w_error;
ustar = kp*w_error + kd*w_dError + ki*w_hError;%PID control action
xdot = ustar;
Je0 = H2Arm.jacob0(qcurr);
qdot2 = pinv(Je0(1:3,:))* xdot(1:3); %only linear Jacobian
qdot = pinv(Je0)*[xdot; 0; 0; 0];
Reproducible Run
or launch a cloud workstation
Reproducibility
Timeline
Submit for publication...
What happens once I publish?
Enrica committed Jun 9, 2020
Added LICENSE; edited metadata.yml
Enrica ran Jun 9, 2020 00:00:58
Run 1739015
output 570 B
Enrica ran Jun 9, 2020 00:01:11
Run 1738938
Enrica committed Feb 20, 2020
minors
Enrica committed Feb 19, 2020
changed noise var name
Enrica committed Feb 18, 2020
added init readBuff
Enrica committed Feb 14, 2020
Locally tested on my machine
Enrica committed Feb 14, 2020
changed capsule name

Main points of a reproducible work

1. Experimental paper	3. Evaluation Criteria	5. How Methods and measurements match the criteria	7. Fair and realistic picture of the system being studied
Yes. Claims are based on experiments	a. Percentage of successful task execution b. Average time to execute the task c. Smoothness in task execution	4.a is used to calculate 3.b 4.b samples are used to calculate 3.b 4.c samples are used to calculate variance and covariance on the trajectories and are used to quantify smoothness	We report and thoroughly discuss both successful and unsuccessful test data in the scope of the experiment we have designed.
2. Hypotheses and Assumptions	**4. What is measured and how**	**6. Information to reproduce the work**	**8. Conclusions precise and valid**
Task: Reaching of an object of interest a. H2Arm platform b. No measure filtering c. Basic BSP d. Basic PID e. Simple Video camera f. Naturally varying daylight in the lab g. Controlled disturbances: manual displacement of the target, Gaussian modelling of noise, injection of additional Pierson-Moskowitz noise in the measures	a. Number of successful tasks, number of experimental tasks b. Time to execute the task c. End point trajectory estimated by the video camera relative to the target blob	It is given in the Supplemental information. Data and code can be found at: 1) **IEEE Dataport** , at *https://ieee-dataport.org/open-access/h2arm-bsp-vs-pid-experiments* You need a free IEEE Account to access it. 2) **CodeOcean** capsules *H2ArmFindRedObj3D*, *H2ArmMotorsActuation*, *H2ArmBSPoptimization* and *H2ArmPIDcontroller* (*https://codeocean.com*). It is necessary to register to the system.	Our conclusions are wrt to Criteria in 3 evaluated with measures as in 4 under hypotheses and assumption spelled out in 2.



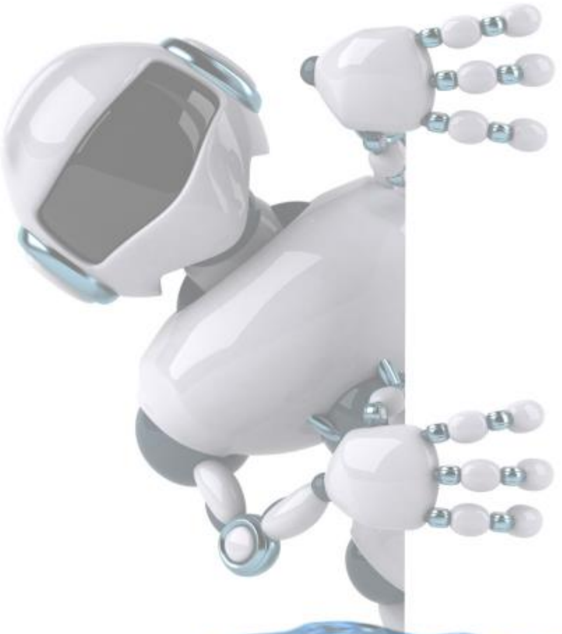
R-article Life Cycle



We are here

It will be possible to publish a short article about the results replication of the R-article. Such articles will be peer reviewed like any other RAM article and will undergo a data and code consistency check.

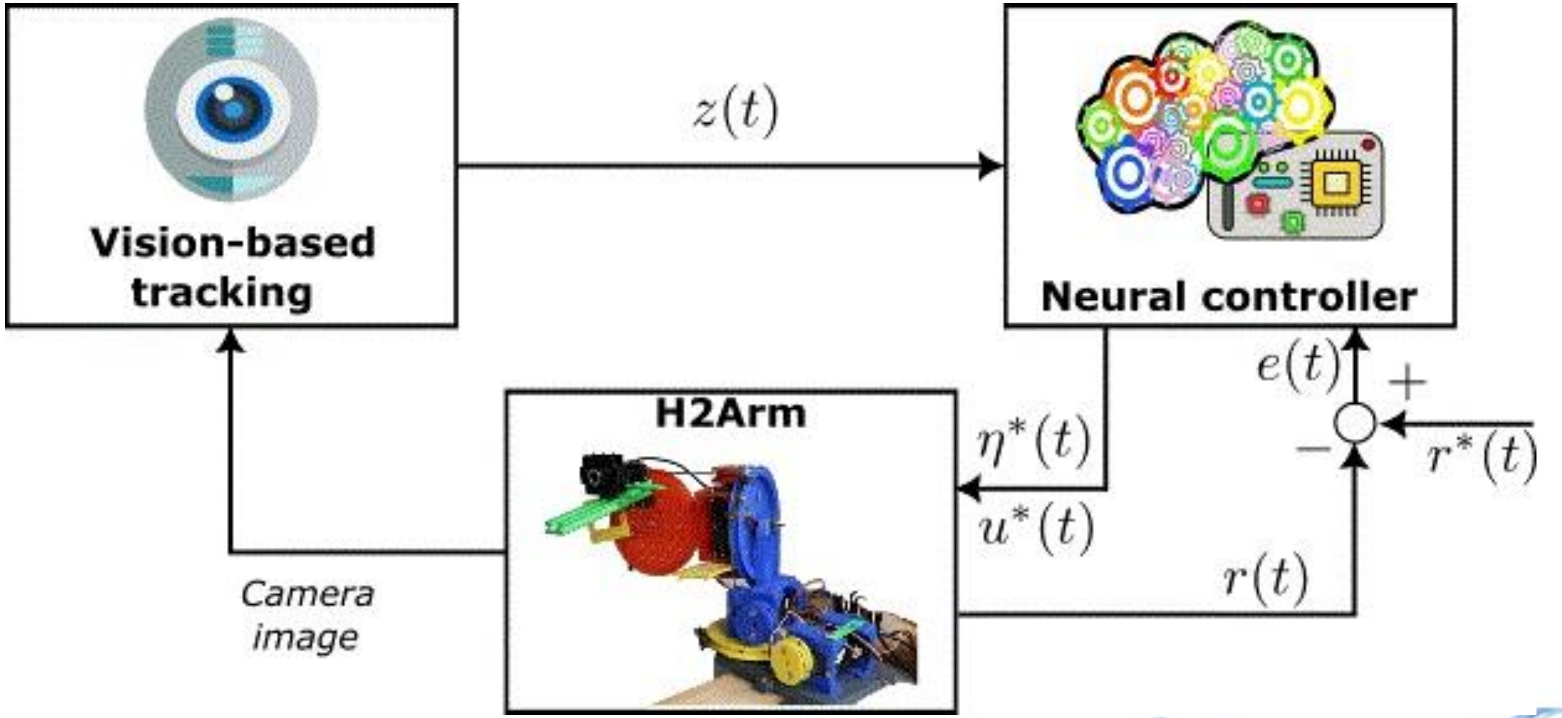
Similarly, we, the authors of the original R-article, will be able to submit, again, in the form of a short peer-reviewed article, a reply to the authors of the r-article, again, with a data and code consistency check.

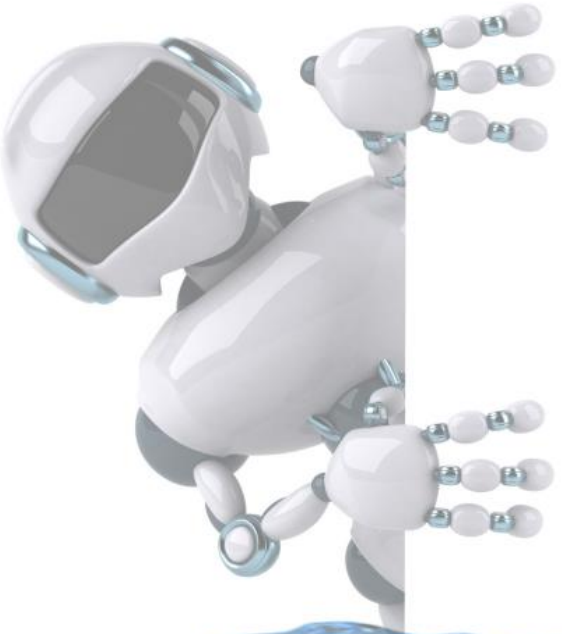


Imitation Learning on H_2Arm

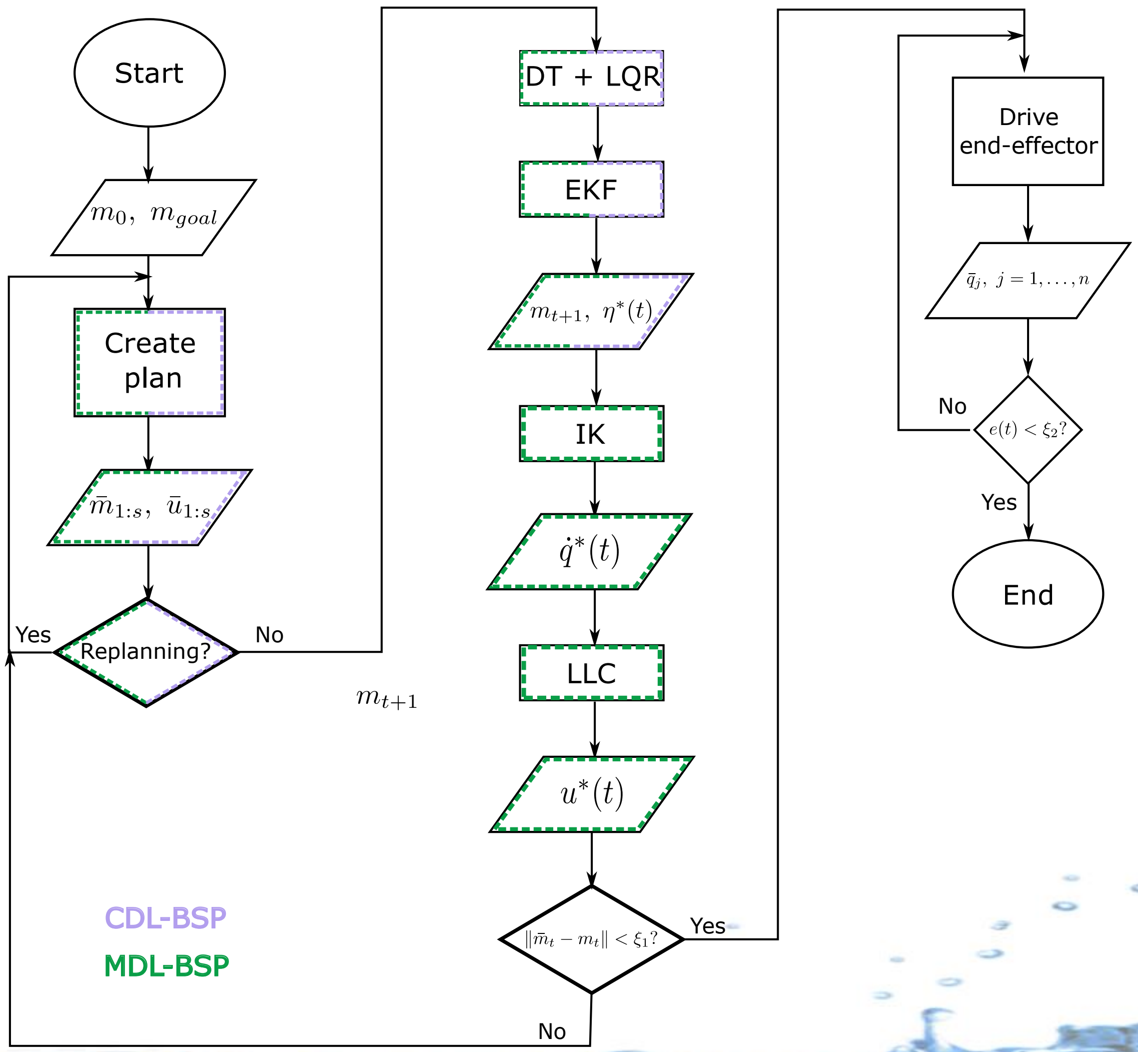
Use of ML

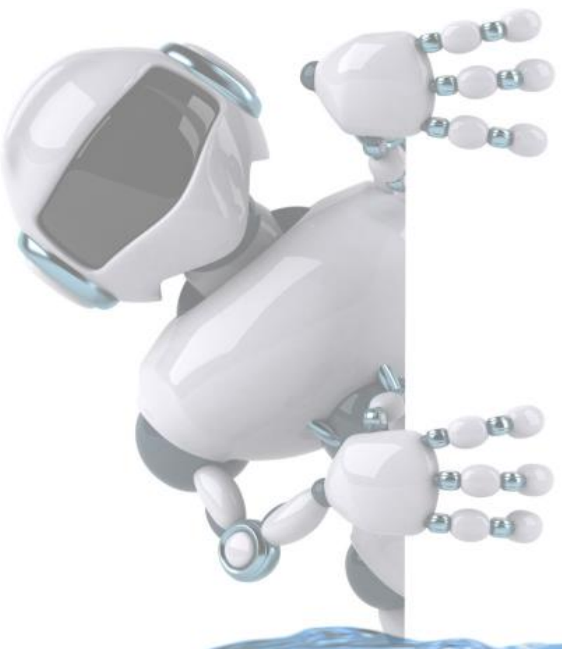
Fabio Bonsignorio, Cristiano Cervellera, Danilo Macciò and Enrica Zereik. "An imitation learning approach for the control of a low-cost low-accuracy robotic arm for unstructured environments." Springer International Journal of Intelligent Robotics and Applications. *In press*.



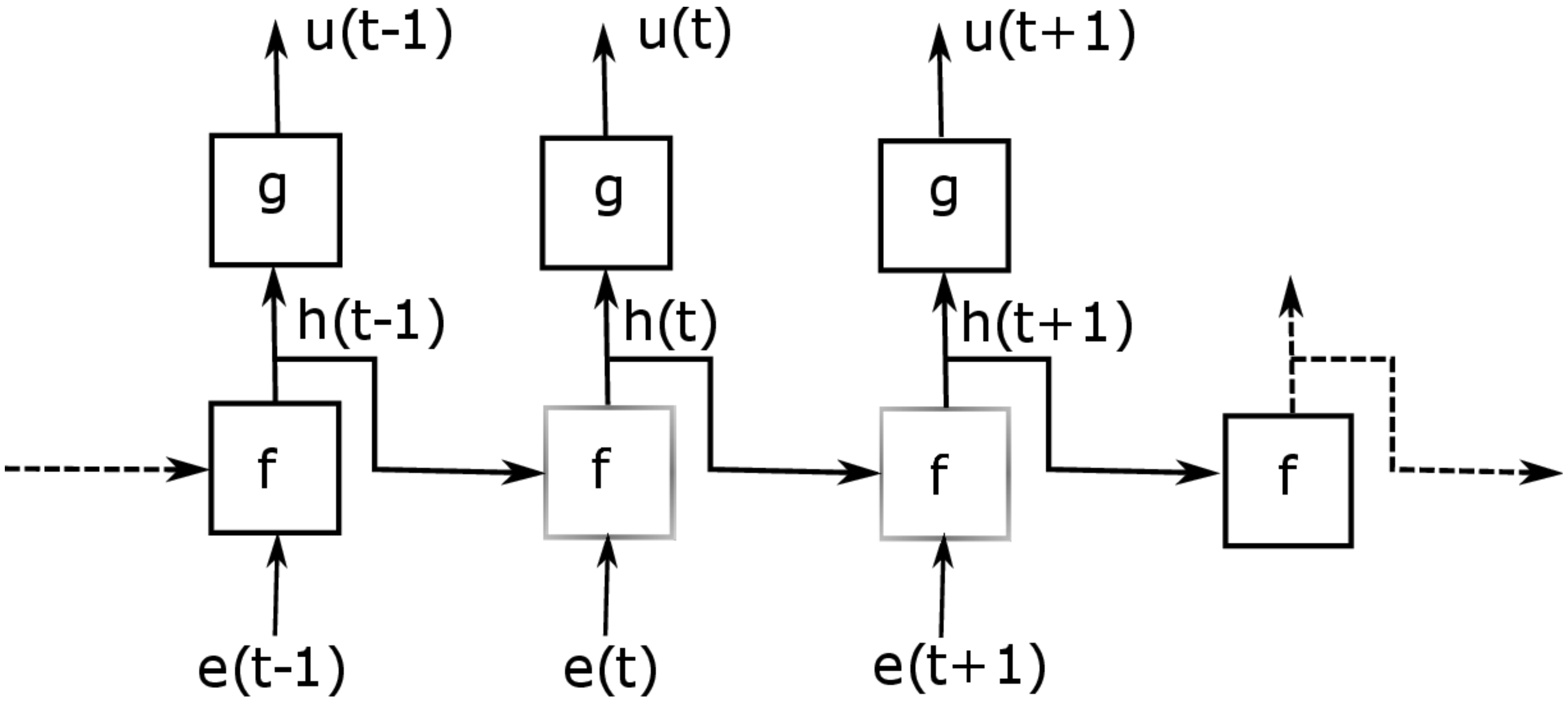


Motor (MDL) vs Cartesian (CDL) schemes





RNN structure



Results at a glance

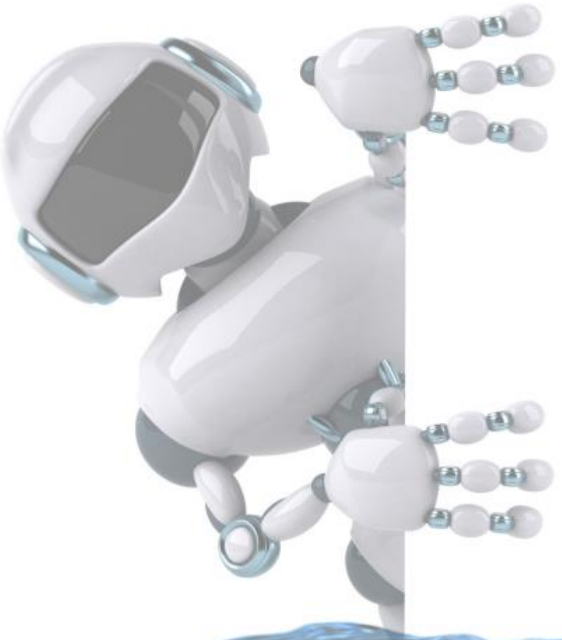
Test trajectories

Neural Controller	Noise	Average steps	Average time [s]
CDL	No	18.25	6.98
MDL	No	19.45	7.53
CDL	Yes	37.1	14.66
MDL	Yes	31.7	12.33

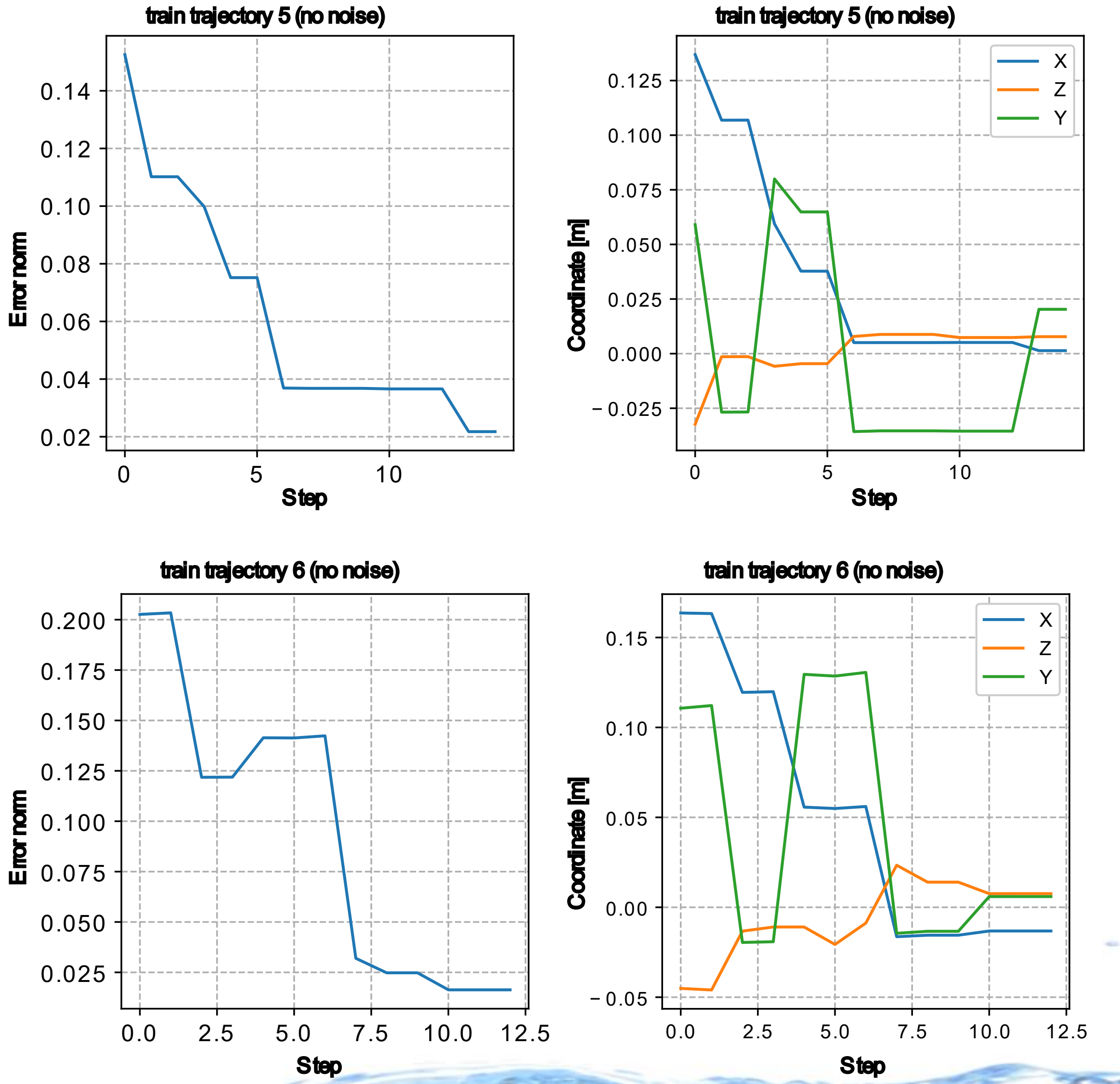
Training trajectories

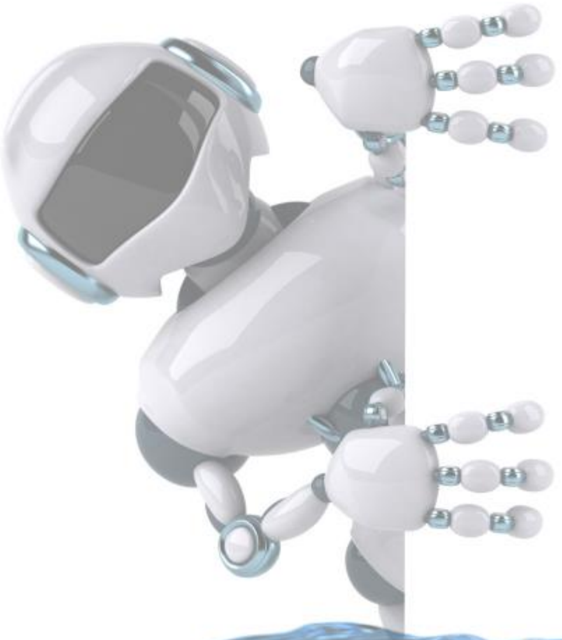
Neural Controller	Noise	Average steps	Average time [s]
CDL *MDL*	No	15.74	73.83
CDL *MDL*	Yes	13.91	91.08

Almost 10 times faster!

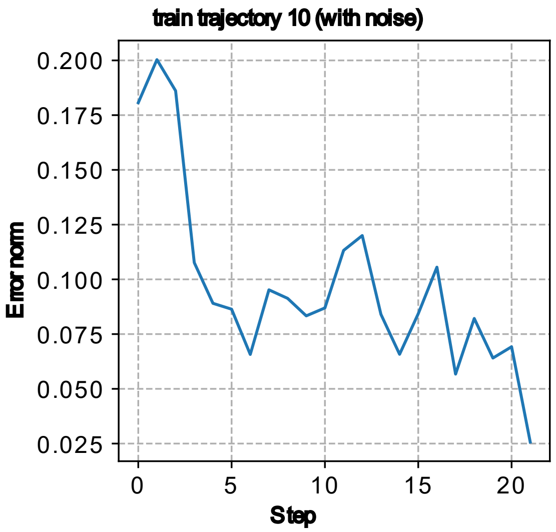


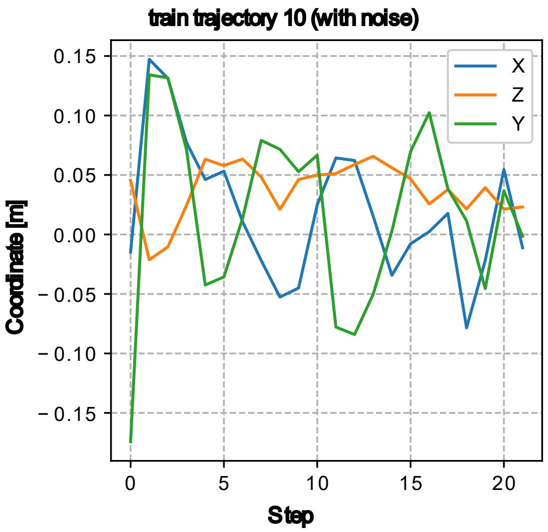
Train trajectories without noise

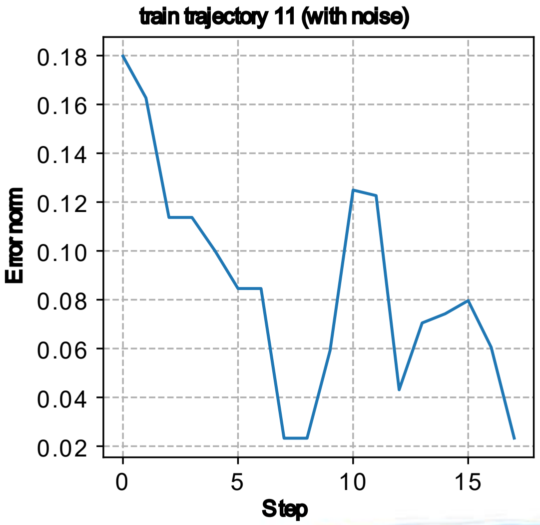


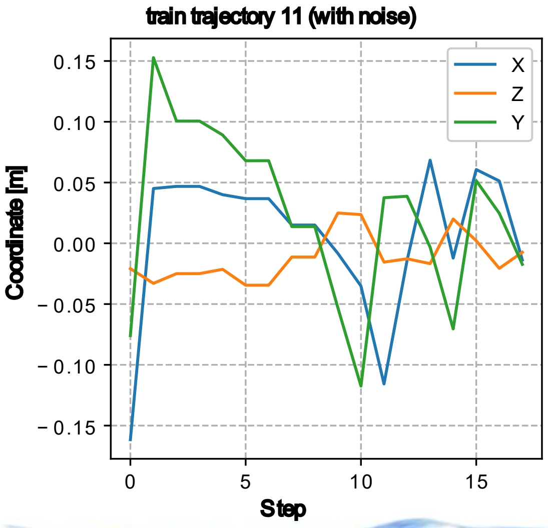


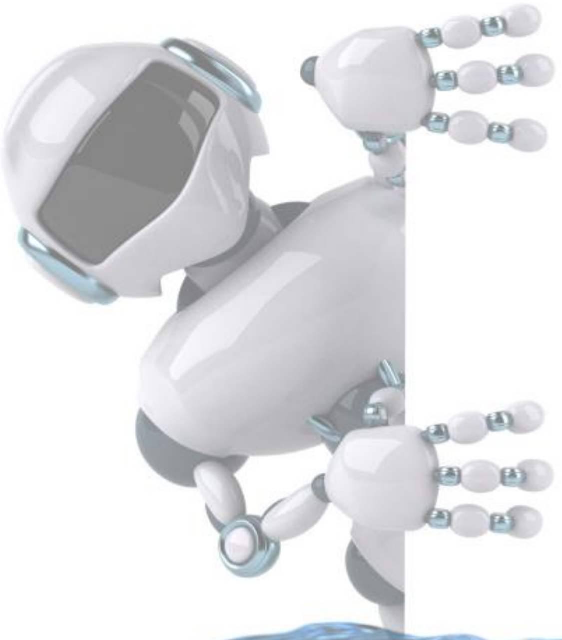
Train trajectories with noise



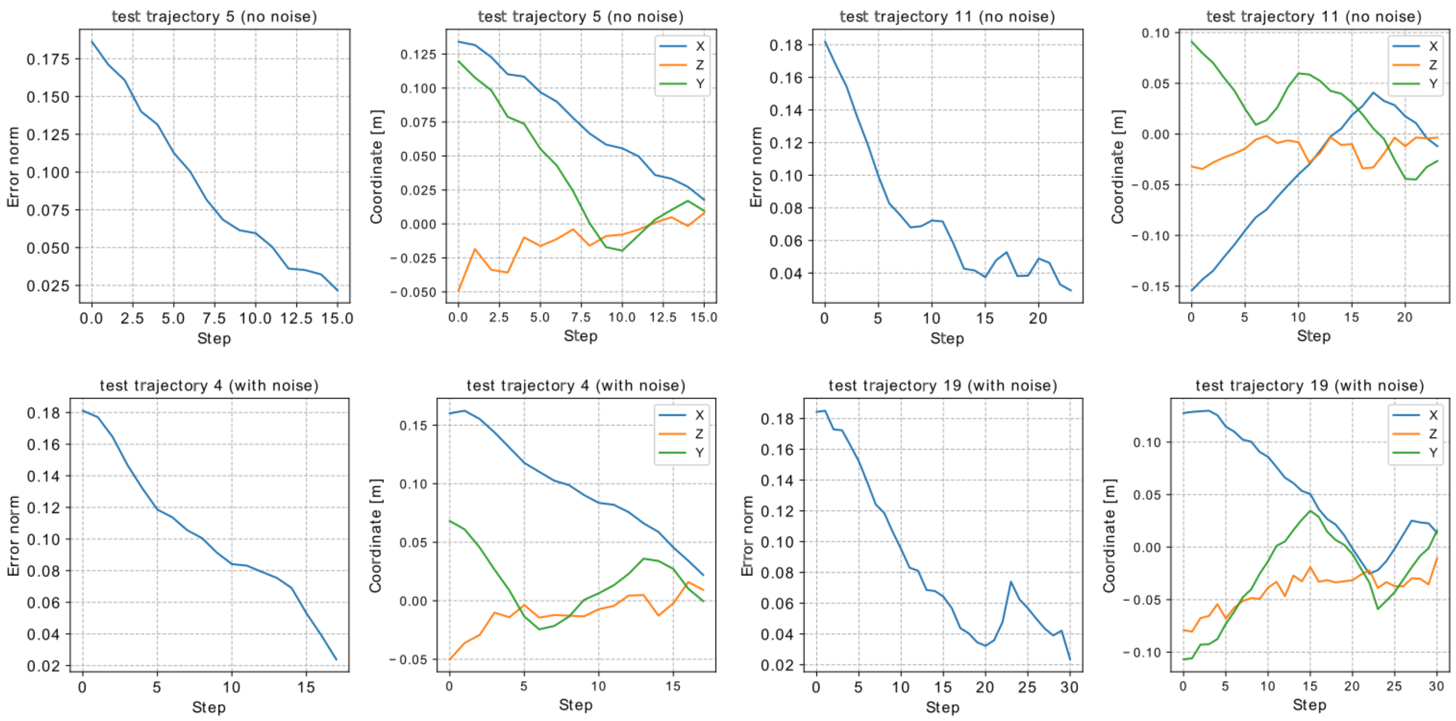


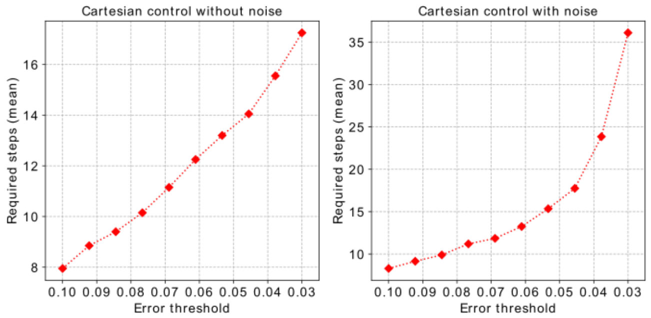




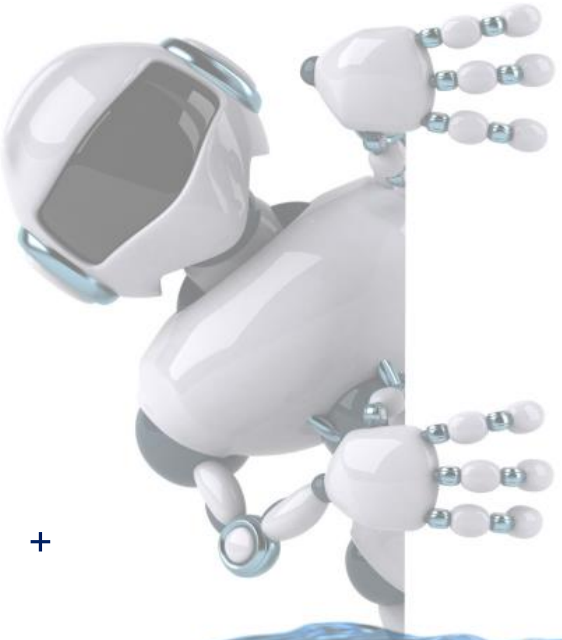


CDL trajectories

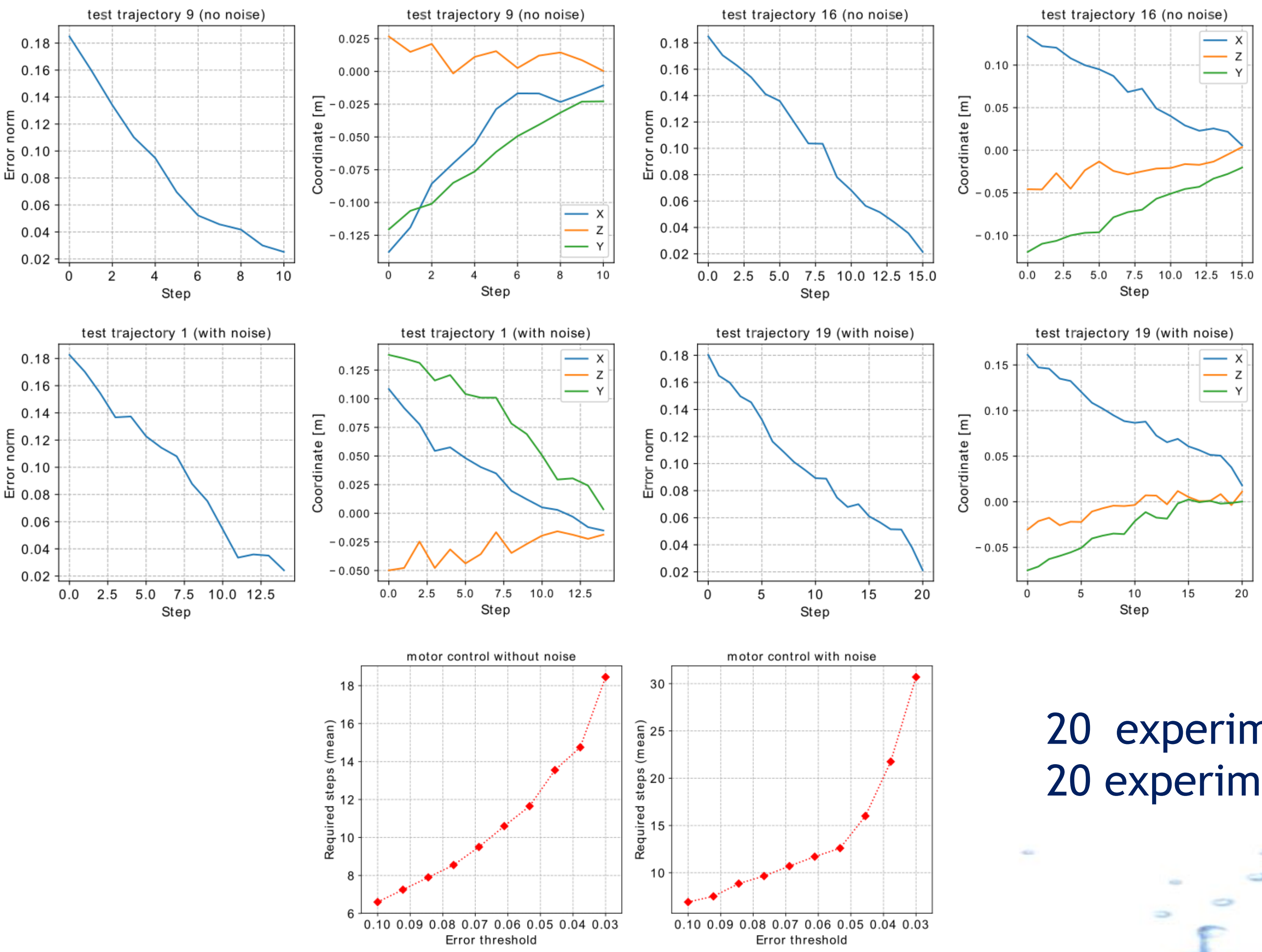




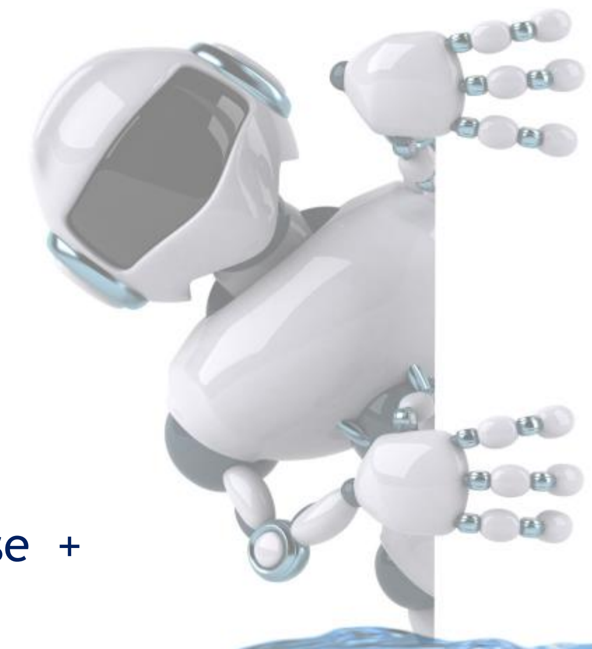
20 experiments without noise +
20 experiments with noise



MDL trajectories

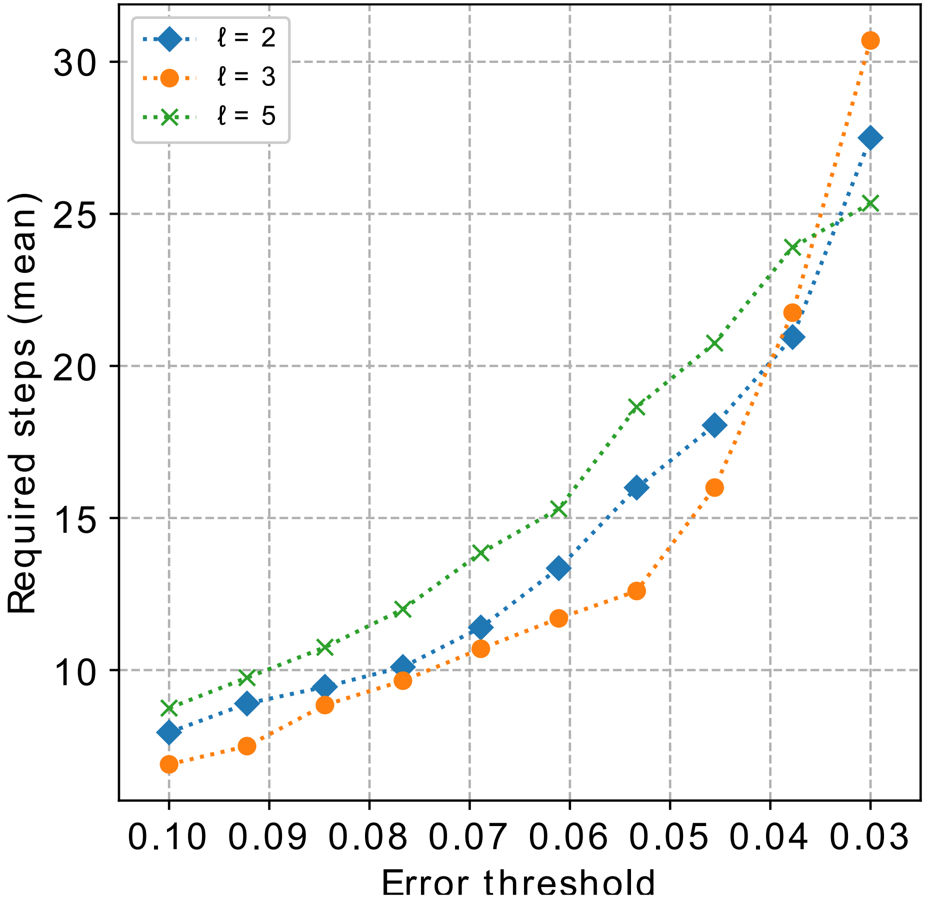


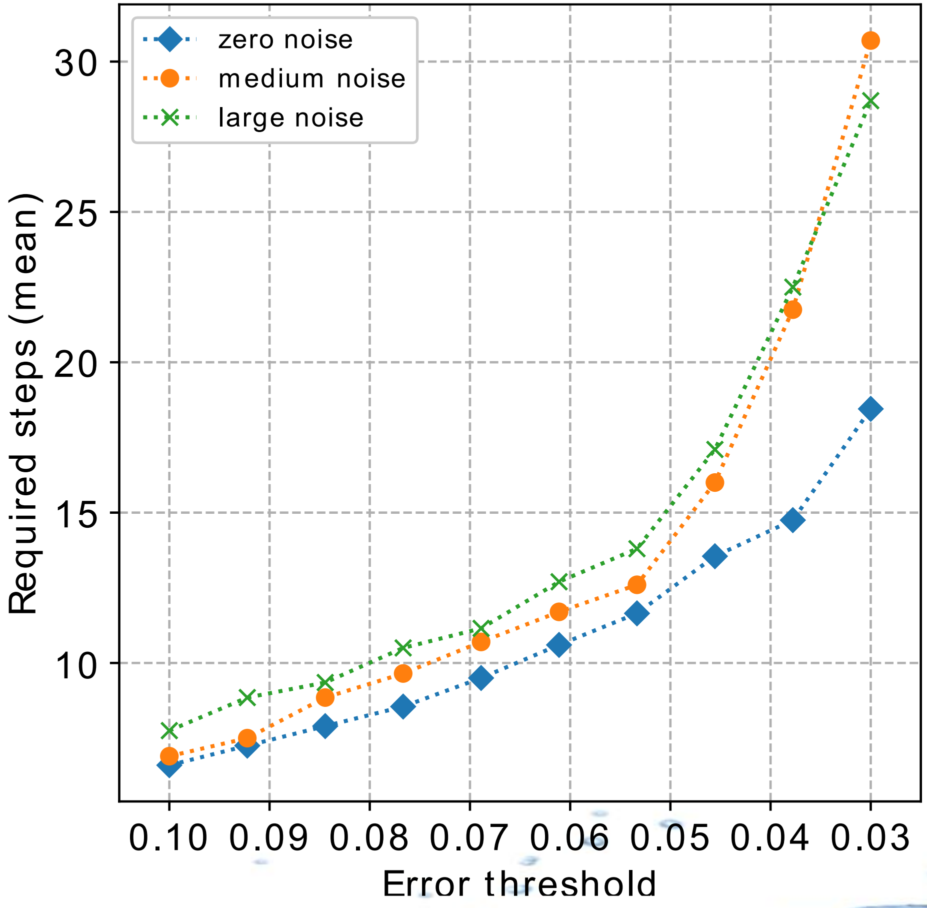
20 experiments without noise +
20 experiments with noise

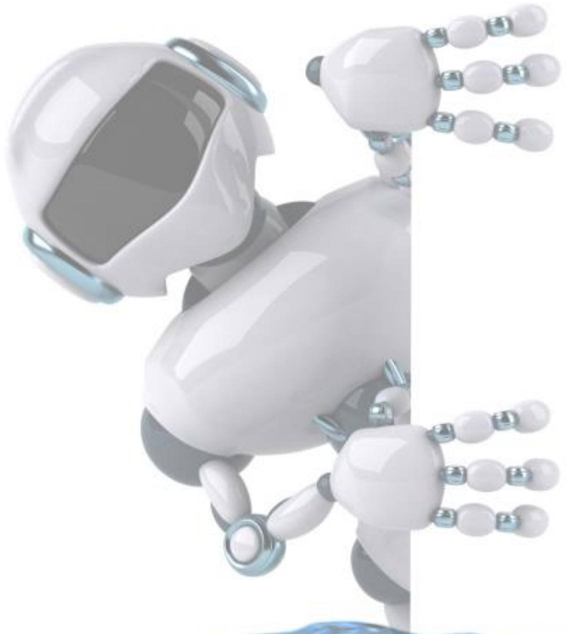


Results in more details

20 experiments without noise + 20 experiments with noise for both CDL and MDL



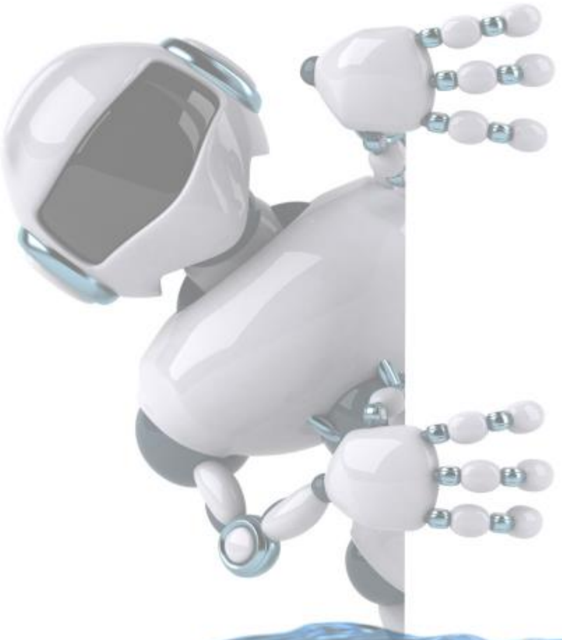




Thank you!

enrica.zereik@cnr.it

ISTITUTO DI INGEGNERIA DEL MARE
INSTITUTE OF MARINE ENGINEERING



And now…. **DEMO**!