Reinforcement Learning and Fault Diagnosis

How models and digital twins may support accelerator operation

Annika Eichler
With Oliver Stein, Jan Kaiser and Julien Branlard
8.10.2021

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

https://leaps-initiative.eu/

LEAPS

League of European Accelerator based-Photon Source

Goal

actively and constructively promote and ensure the quality and impact of fundamental, applied and industrial research carried out at their facilities

LEAPS Integrated Platform – LIP

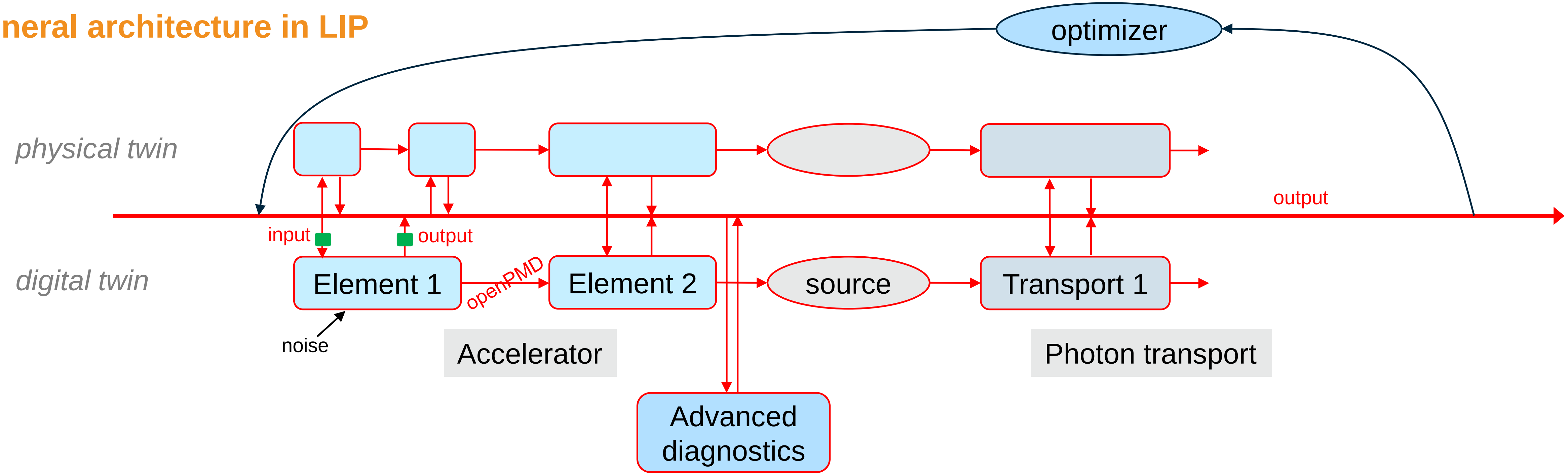
bring together experts in the field of **Digital Twinning, Machine Learning** and **Virtual Diagnostic**

- set up a detailed survey of the ongoing activities within LEAPS on DT, ML & VD
- draw up a summary document, which will constitute the cornerstone of the LIP project

SESAME LENS Canadian Light Source Centre canadien de rayonnement synchrotron ESUO EUROPEAN SYNCHROTRON AND FEL USER ORGANISATION

Digital Twin

General architecture in LIP



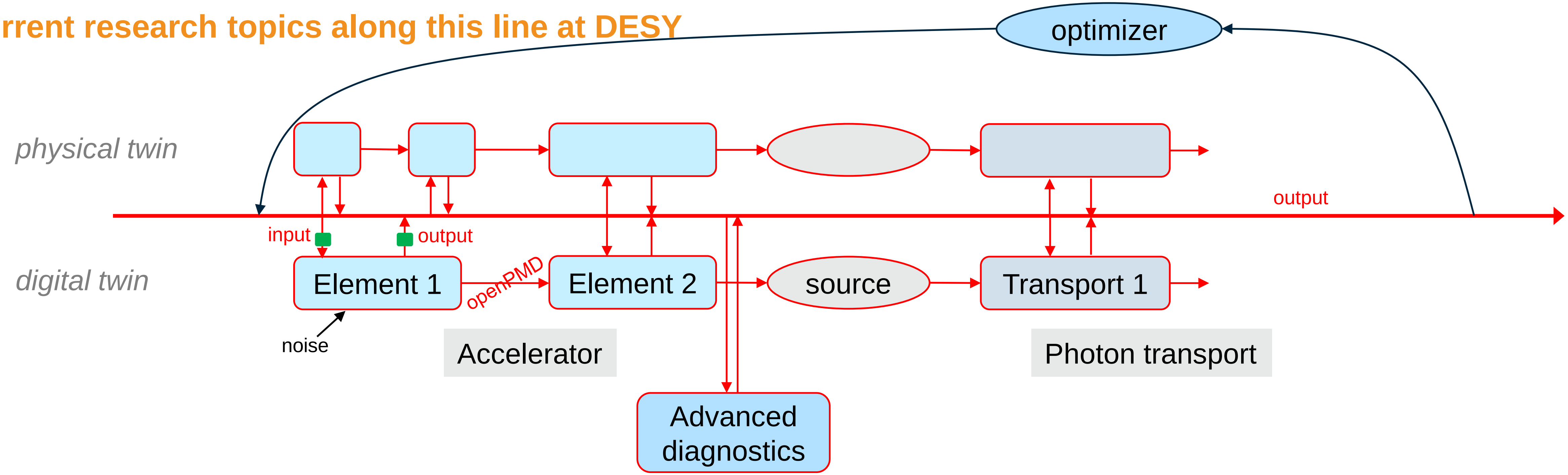
- experimentalists can get familiar with the facilities → more efficient experiments
- support the operators to increase the performance of the facility with respect to the experiments
- capability to test new algorithms for control and optimization also exploiting machine learning
- if real-time capable, usage for online control and diagnosis.

→ support on the way towards more automation / autonomy

Courtesy: Maxence Thevenet

Increasing autonomy

Current research topics along this line at DESY



- **Data acquisition and data analysis (pipelines)**
  - Get all relevant signals and provide understanding
  - Provide data infrastructure, interfaces, etc.
- **(Surrogate) modelling, simulations, digital twins**
  - Understanding physics
  - Requirement for predictions, development and control
- **Fault diagnosis and supervisory control**
  - Predict faults, prevent failures
  - Protect the system
- **Optimization and feedback control algorithms**
  - Push the way of operation
  - Optimize performance

Increasing autonomy

Current research topics along this line at DESY

- **Data acquisition and data analysis (pipelines)**
  - Get all relevant signals and provide understanding
  - Provide data infrastructure, interfaces, etc.

Provides the data for modeling identification and validation

Possibly provides the interface for online interaction

- **(Surrogate) modelling, simulations, digital twins**
  - Understanding physics

Requirement for diagnosis, predictions, and control

- **Fault diagnosis and supervisory control**
  - Predict faults, prevent failures
  - Protect the system
- **Optimization and feedback control algorithms**
  - Push the way of operation
  - Optimize performance

RL for accelerator operation

Reinforcement Learning

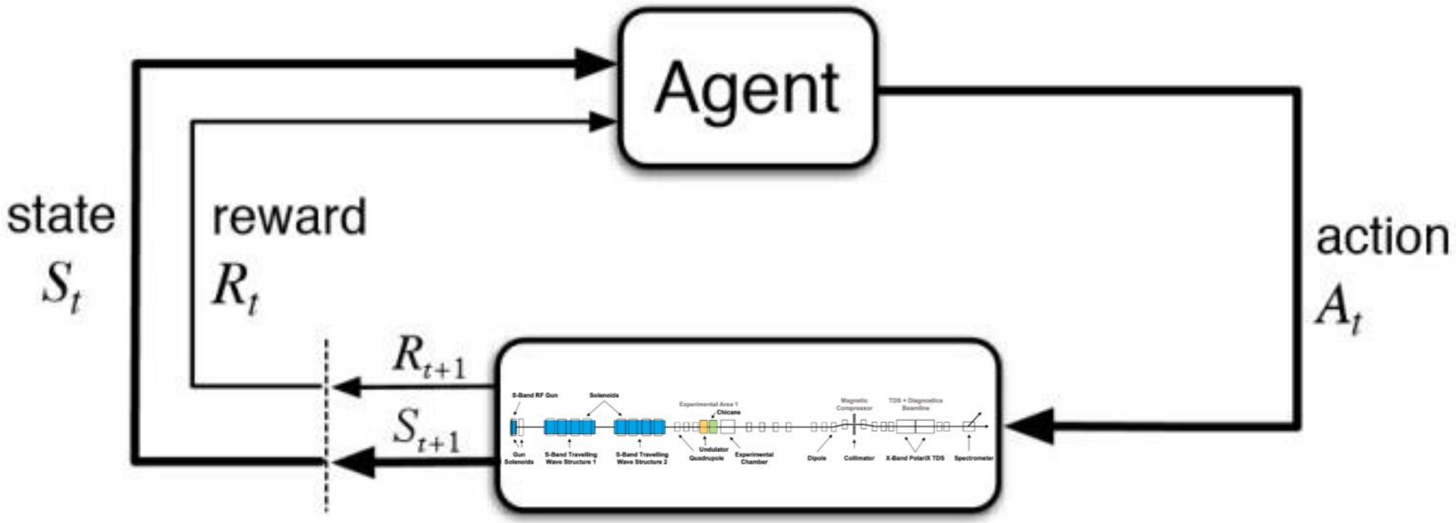
Artificial intelligence as an enabler for autonomy

Reinforcement Learning

- machine learning algorithm with an
  - Agent interacts with environment (simulation or real world),
  - Take action based on observation
  - Receiving reward
  - → Plan ahead

Examples and applications

- Games (AlphaGo)
- Personalized recommendations
- Robotics
- Traffic light control
- …



Sutton: Reinforcement Learning, an introduction

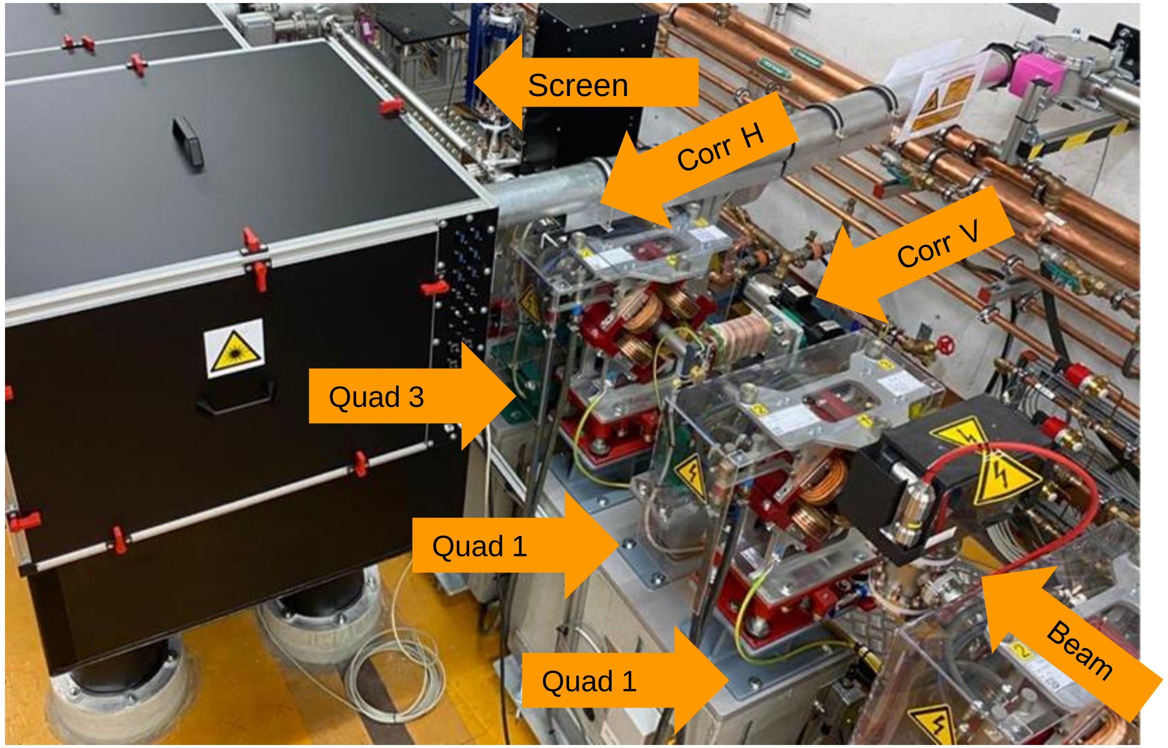
Goal: Apply reinforcement learning (RL) to accelerator operation

- Collaboration between KIT & DESY 2 years
- 2 years project funded by Helmholtz AI: Initiative and Networking Fund by the Helmholtz Association
- Research facilities:
  - ARES (DESY)
  - FLUTE (KIT)

Transferability?

Beam Focusing

Proof of concept example at ARES

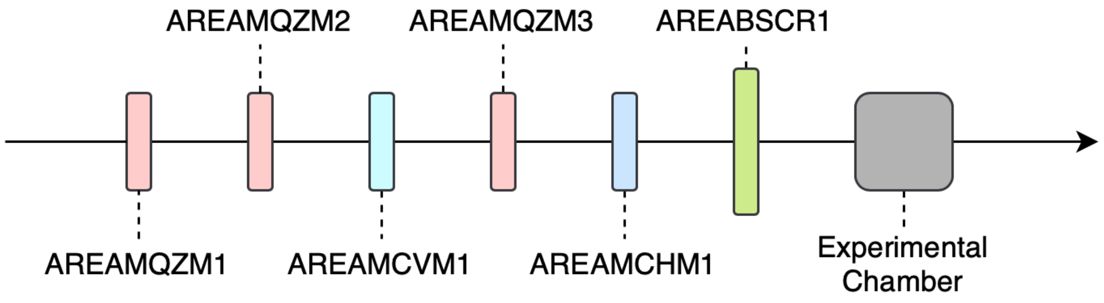


Task

- Position and focus the electron beam on a diagnostic screen in the ARES Experimental Area

Motivation

- Recurring problem from ARES operation.
- Simple enough to still understand what agent does, yet complex enough to be interesting.
- ARES as an easily accessible testbed to eventually map experiences to larger machines like European XFEL



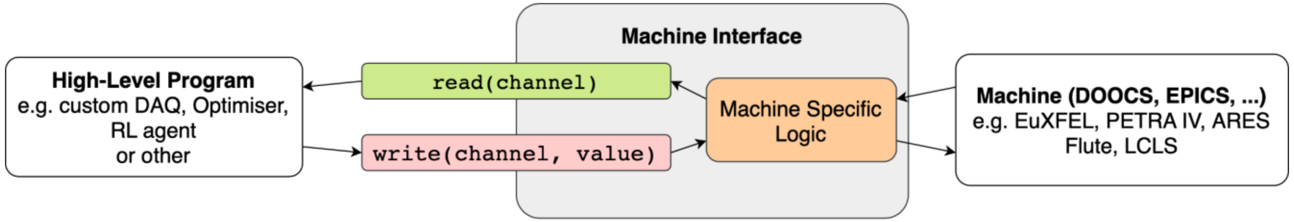
Courtesy: Oliver Stein & Jan Kaiser

Gym Environments for Particle Accelerator Applications

„Digital Twin“ with standardised simulation and machine interfaces

Accelerator-Environments Project

- **Train with fast iteration in simulation, apply on machine**
- Collection of environments implemented around OpenAI Gym
  - Very simple to move from simulation to machine, various backends are provided
  - Interface for reinforcement learning and optimisation algorithms
- Compatible with popular RL and optimisation libraries such as Stable Baselines3 and SciPy Optimize



Train RL agent

```
import accelerator_environments
import gym
from stable_baselines3 import TD3

model = TD3("MlpPolicy", "ARESEA-JOSS-v0").learn(1000)
```

Run on machine

```
env = gym.make("ARESEA-Pydoocs-v0")

observation = env.reset()
done = False
while not done:
    action = model.predict(observation)
    observation, _, done, _ = env.step(action)
```

Optimisation

```
from scipy import optimize

env = gym.make("ARESEA-Pydoocs-v0")

env.reset()
bounds = optimize.Bounds(env.optimisation_space.low,
                         env.optimisation_space.high)
x = optimize.minimize(fun=env.objective_function,
                      x0=env.initial_actuators,
                      bounds=bounds)
```

Defining an Optimisation Problem

Proof of concept example at ARES

Objective

Natural logarithm of the weighted sum of parameter differences

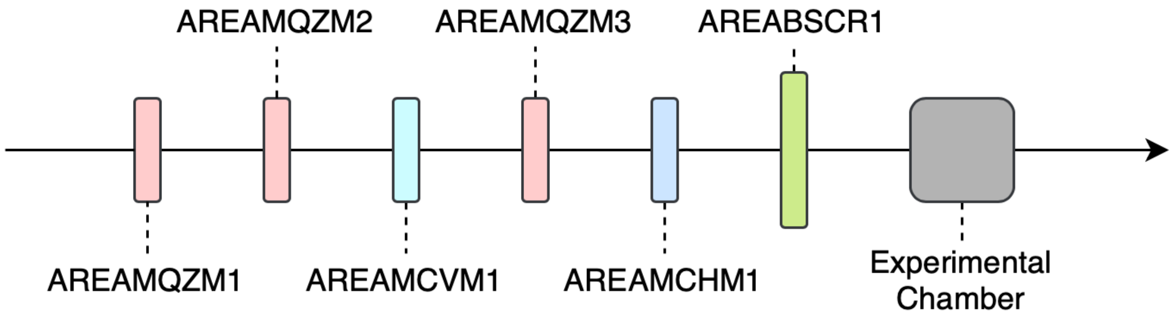
$$O(\boldsymbol{x}) = \ln \sum_{p \in \boldsymbol{b}_S, p' \in \boldsymbol{b}'_S} \alpha_p \left| p - p' \right|$$

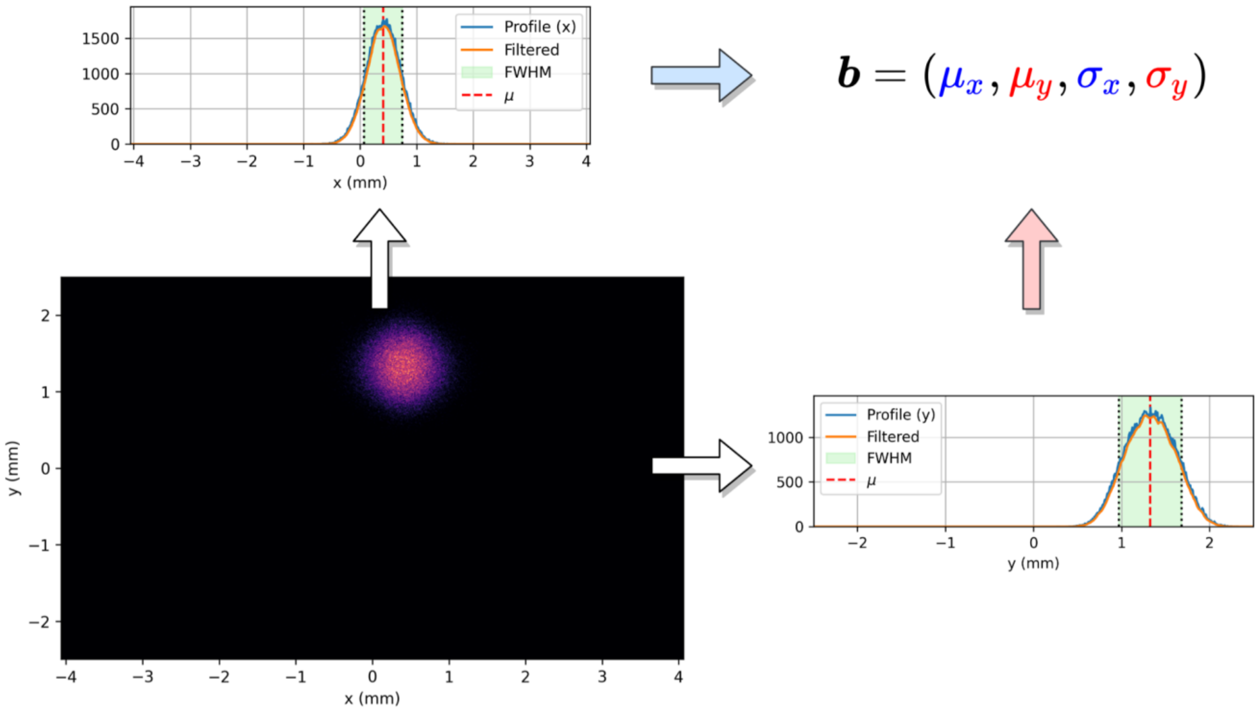
Actuators $\boldsymbol{x} = (k_{Q_1}, k_{Q_2}, k_{Q_3}, \alpha_{C_v}, \alpha_{C_h})$

Beam Parameters $\boldsymbol{b}_S = (\mu_x, \mu_y, \sigma_x, \sigma_y)$

Reading Beam Parameters

1. Sum over pixels in x, y → filter to clean up
2. Find widest interval edges > 0.5x the maximum, call it FWHM
3. Beam position = centre between interval edges
4. Beam size = FWHM / 2.355





$$\boldsymbol{b} = (\mu_x, \mu_y, \sigma_x, \sigma_y)$$

Translation to a Reinforcement Learning Task

Proof of concept example at ARES

Reward

Improvement in objective (difference of current and last step‘ s objective)

$$\hat{r}\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) = O\left(\boldsymbol{x}_t\right) - O\left(\boldsymbol{x}_{t+1}\right)$$

$$r\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) = \begin{cases} \hat{r} & \text{if } \hat{r} \geq 0 \\ 2\hat{r} & \text{otherwise} \end{cases}$$

Observation

Beam parameters (of this and previous time step)

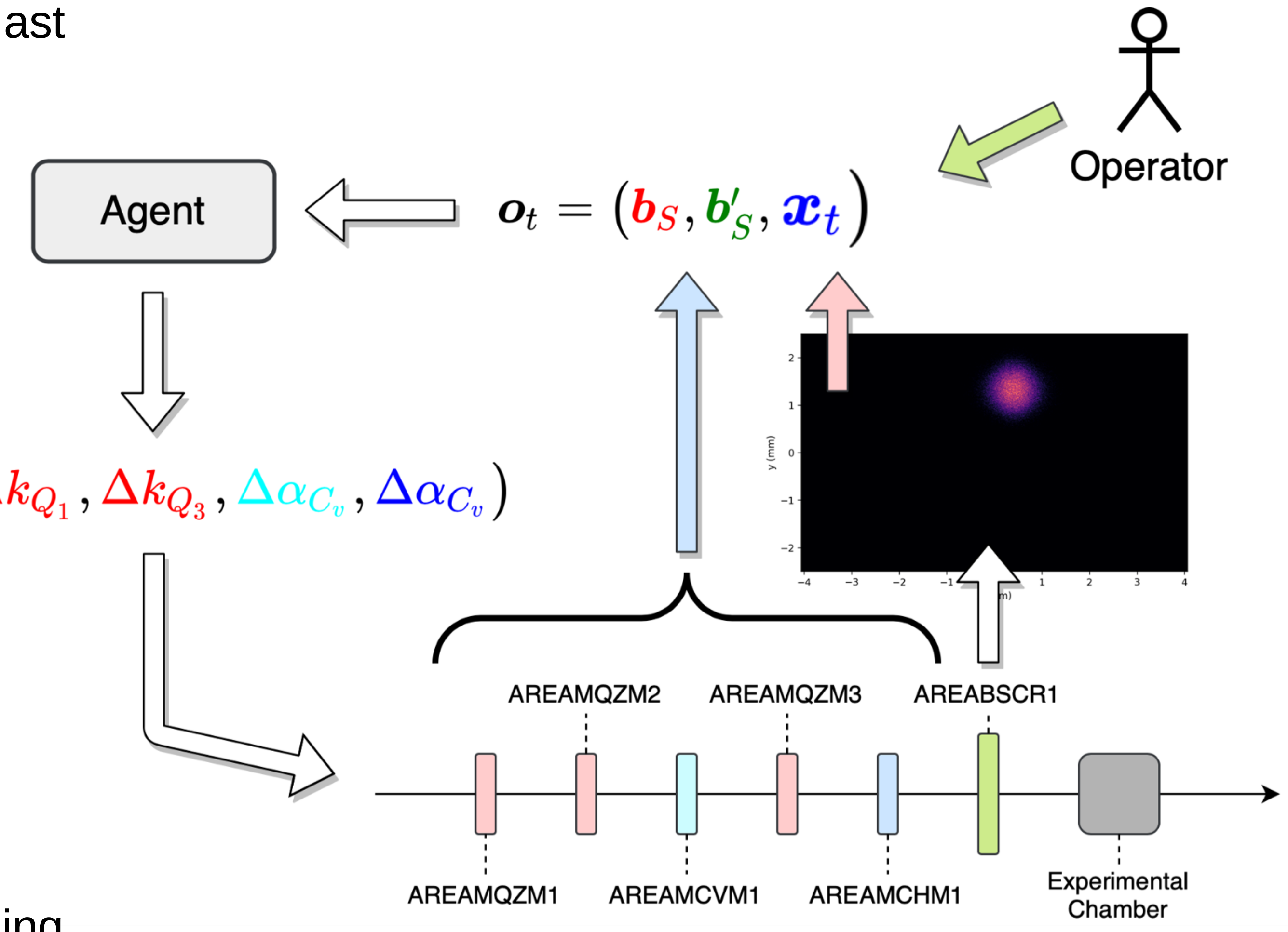
Actuator values

$$\boldsymbol{o}_t = \left(\boldsymbol{b}_S, \boldsymbol{b}'_S, \boldsymbol{x}_t\right)$$

Action

Delta of actuators values

$$\boldsymbol{a}_t = \left(\Delta k_{Q_1}, \Delta k_{Q_1}, \Delta k_{Q_3}, \Delta \alpha_{C_v}, \Delta \alpha_{C_v}\right)$$



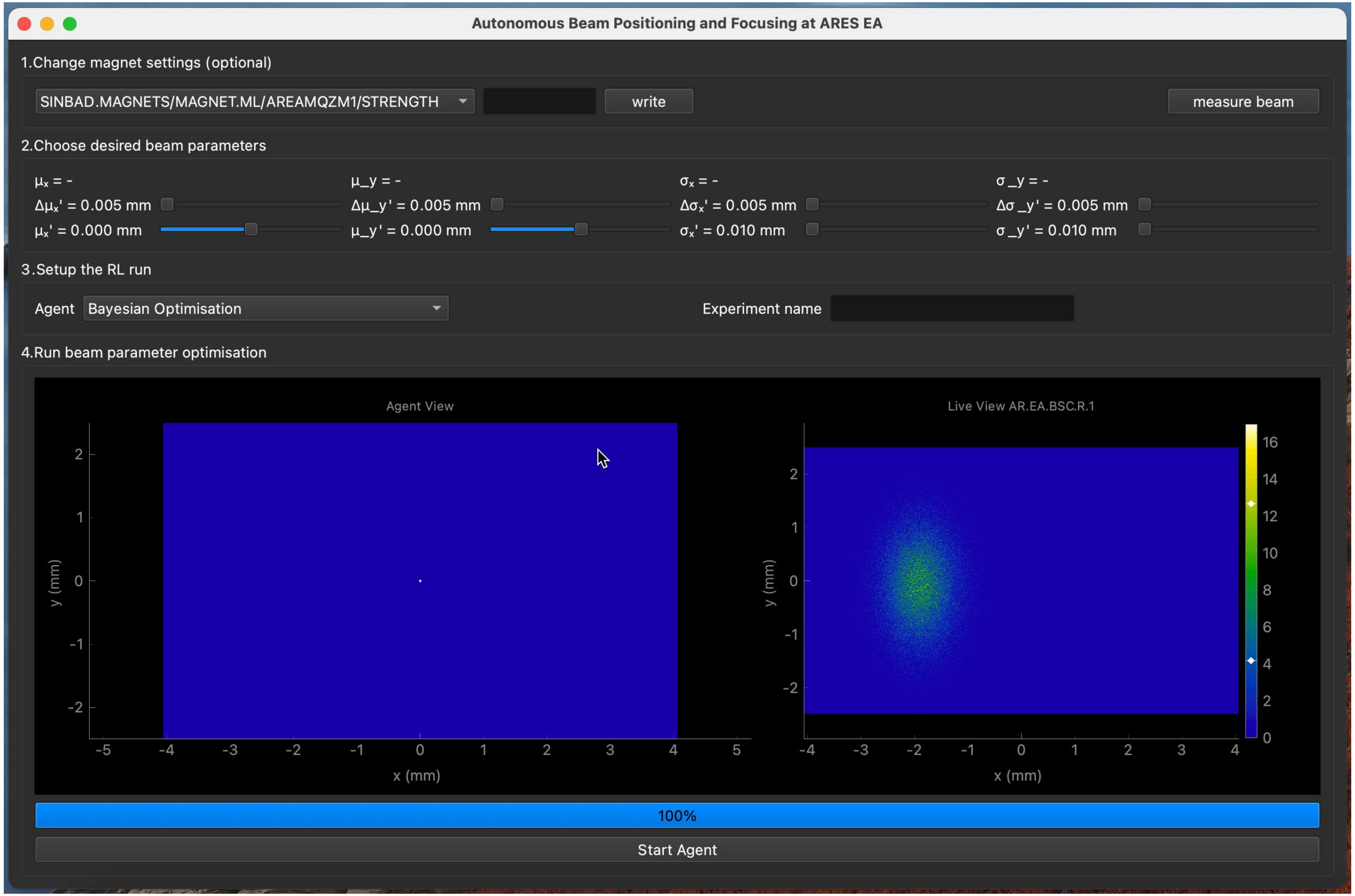
Training

- Twin Delayed DDPG (TD3)
  - Benchmarked *Stable Baselines3* implementation
- Open AI *Gym* environment implementation
- Different beams and objectives are seen in training

Courtesy: Oliver Stein & Jan Kaiser

Control Room Application

Prototype



Summary and Outlook

What's next and what are the challenges

	Simulation (train)	Experiment (act)
Time per step	0.03 seconds	10-30 seconds
Number of steps	600 000	5-10

At ARES

- Transfer from simulation to machine has successfully demonstrated
  - Otherwise training would be impossible (600 000 steps in simulation ~ 5 hours, 33 steps/seconds, 0.03 seconds/step)
- For training in simulation a fast and accurate model is required
- Test agent on FLUTE (simulation and real accelerator) for transferability
- How to train and act more efficiently? Training 600 000 steps, actions 5-10 steps needed
- Include errors in simulation (quadrupole offset, screen offset, etc.)

Move to more complex tasks:

- ARES start-up
- RL applications at FLASH and the European XFEL → SASE tuning at FLASH
  - Problem: good and fast simulation required

Fault diagnosis for SRF cavities

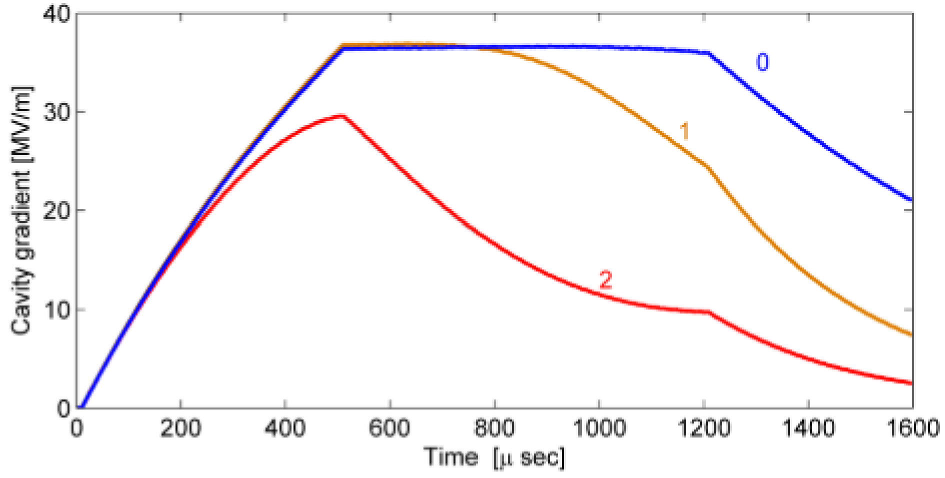
Quench detection system

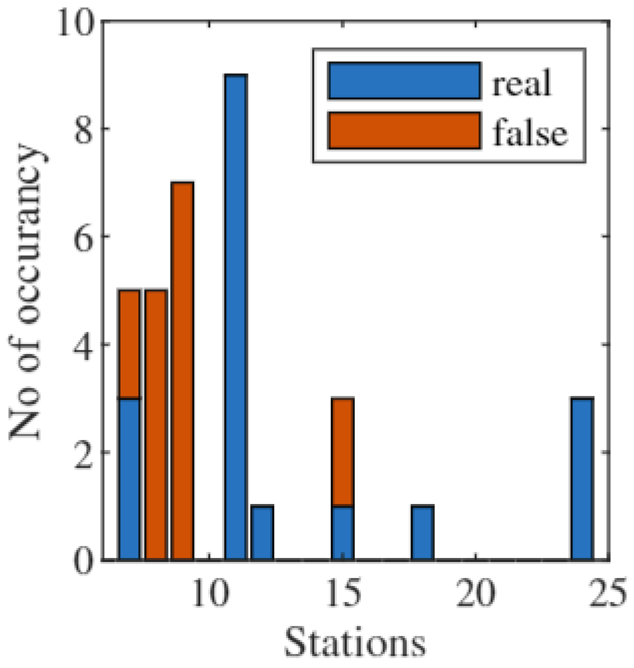
In operation at European XFEL

- **Quench**
  - Severe cavity fault
  - Loss of superconductivity of the cavity walls

- **Quench detection system**
  - Works very well
  - In total: not so many quenches
  - Based on the determination of the loaded quality factor (in decay)
  - Soft quenches are not so easy to detect
  - False positive
    - 07/08/2020 till 11/18/2020, 34 snap shots were saved triggered by the quench detection (thanks to Nicholas Walker)
    - 18/34 were real quenches

J. Branlard et. al., “Superconducting cavity quench detection and prevention for the European XFEL,” 16th International Conference on RF Superconductivity, 2013.

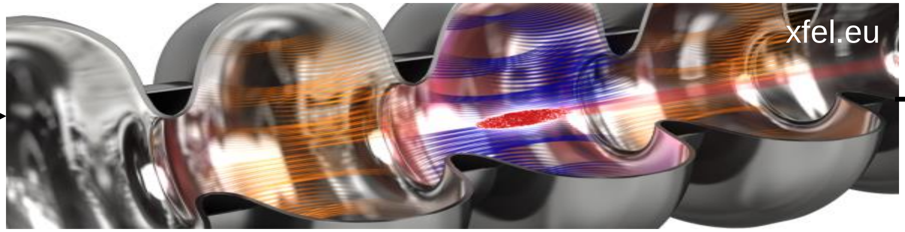




Anomaly detection for SRF cavities

Modeling approach

$V_{F,I}$
$V_{F,Q}$

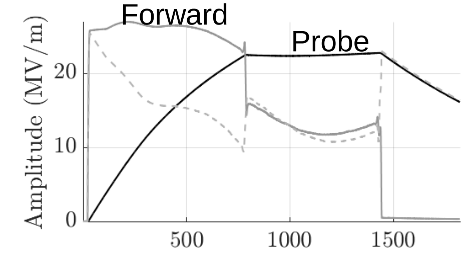


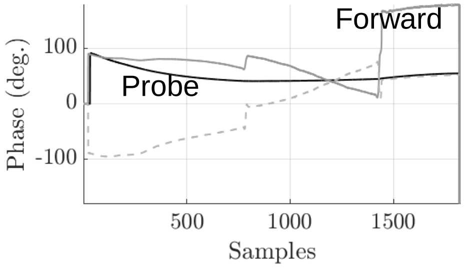
$V_{P,I}$
$V_{P,Q}$

$V_{F,I}$
$V_{F,Q}$

Nonlinear Model (electromagnetic + mechanical model)

$V_{P,I}$
$V_{P,Q}$





Eletromagnetic oscillation

$$\begin{bmatrix}\dot{V}_{P,I}(t)\\ \dot{V}_{P,Q}(t)\end{bmatrix} = \begin{bmatrix}-\omega_{1/2} & -\Delta\omega(t)\\ \Delta\omega(t) & -\omega_{1/2}\end{bmatrix}\begin{bmatrix}V_{P,I}(t)\\ V_{P,Q}(t)\end{bmatrix} + 2\omega_{1/2}\begin{bmatrix}V_{F,I}(t)\\ V_{F,Q}(t)\end{bmatrix} - V_B(t)\begin{bmatrix}cos(\phi_P)\\ cos(\phi_P + \frac{\pi}{2})\end{bmatrix}$$

Mechanical deformation

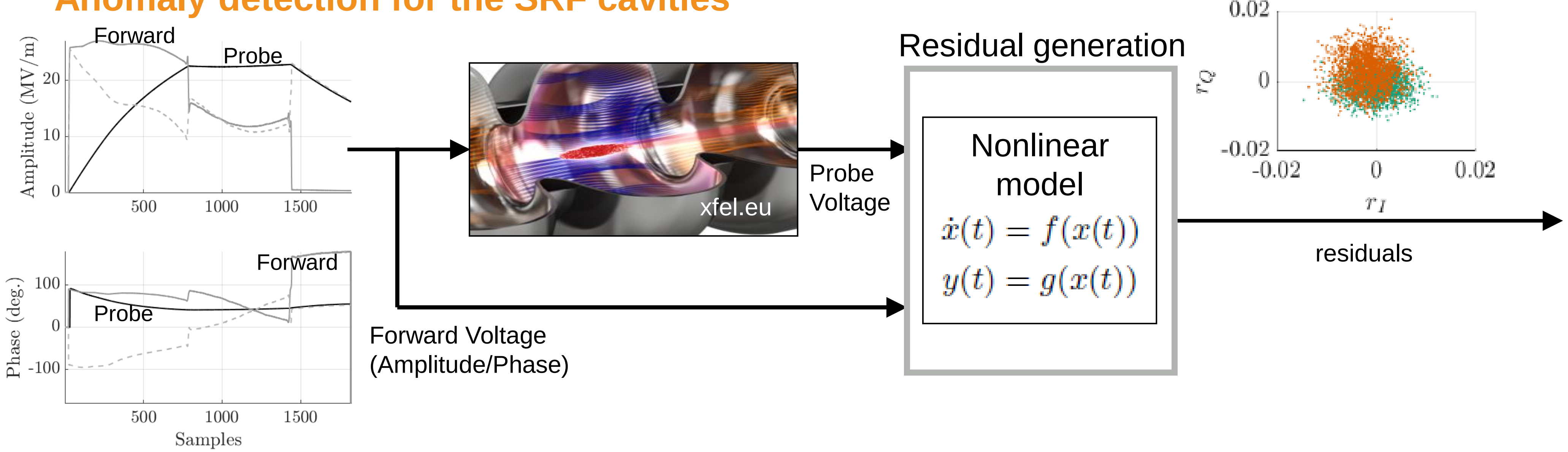
$$\Delta\dot{\omega}_n(t) = -\frac{1}{\tau_n}\Delta\omega_n(t) + K_n\left(V_{P,I}^2(t) + V_{P,Q}^2(t)\right)$$

$$\Delta\omega(t) = \sum_{n=1}^{N}\Delta\omega_n(t)\ ,\ \forall n = 1,\ldots,N\ .$$

Courtesy: Ayla Nawaz

Advanced Fault Diagnosis

Anomaly detection for the SRF cavities



Parity space

- Solve both electromagnetic equations for detuning Δω
- Residual is the difference (small if model fits well, large otherwise)

\+ Little calculation effort

\- Sensitive to noise

Unscented Kalman filter

- Kalman filter for nonlinear systems
- Predict and update steps (weighting model and new measurements)

\- Calculation intensive

\+ Optimal filtering (if Gaussian noise)

Parameter estimation

- Calculate detuning Δω and half bandwidth ω1/2 from forward and probe signals

\+ Little calculation effort

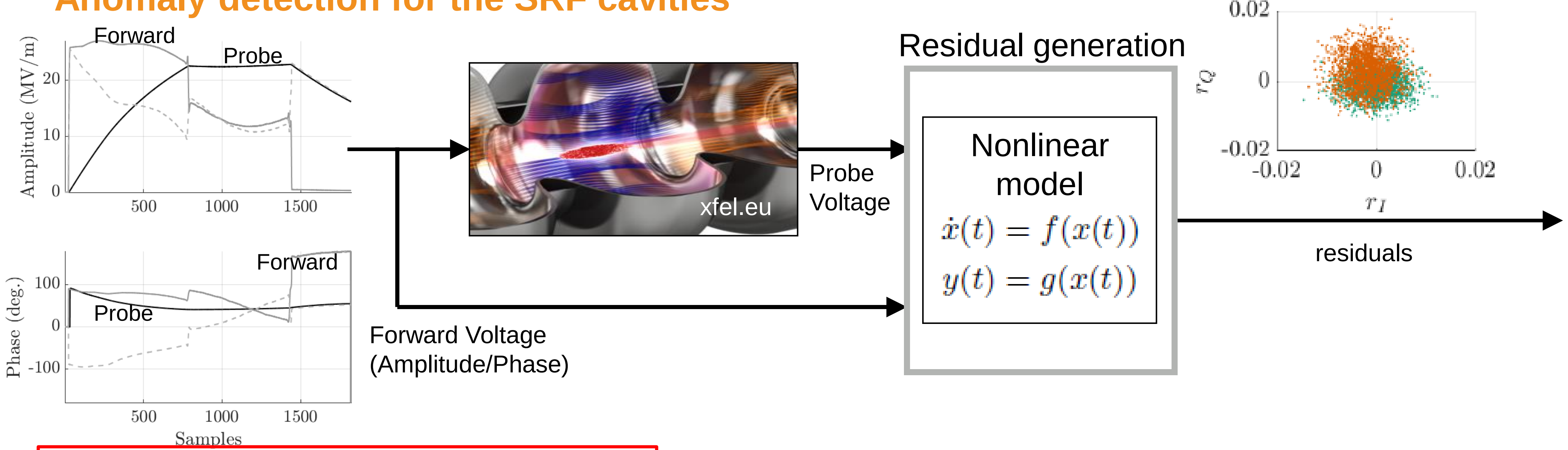
\- Sensitive to noise

\+ Good physical interpretability

Courtesy: Ayla Nawaz

Advanced Fault Diagnosis

Anomaly detection for the SRF cavities



Parity space

- Solve both electromagnetic equations for detuning Δω
- Residual is the difference (small if model fits well, large otherwise)

\+ Little calculation effort
\- Sensitive to noise

Unscented Kalman filter

- Kalman filter for nonlinear systems
- Predict and update steps (weighting model and new measurements)

\- Calculation intensive
\+ Optimal filtering (if Gaussian noise)

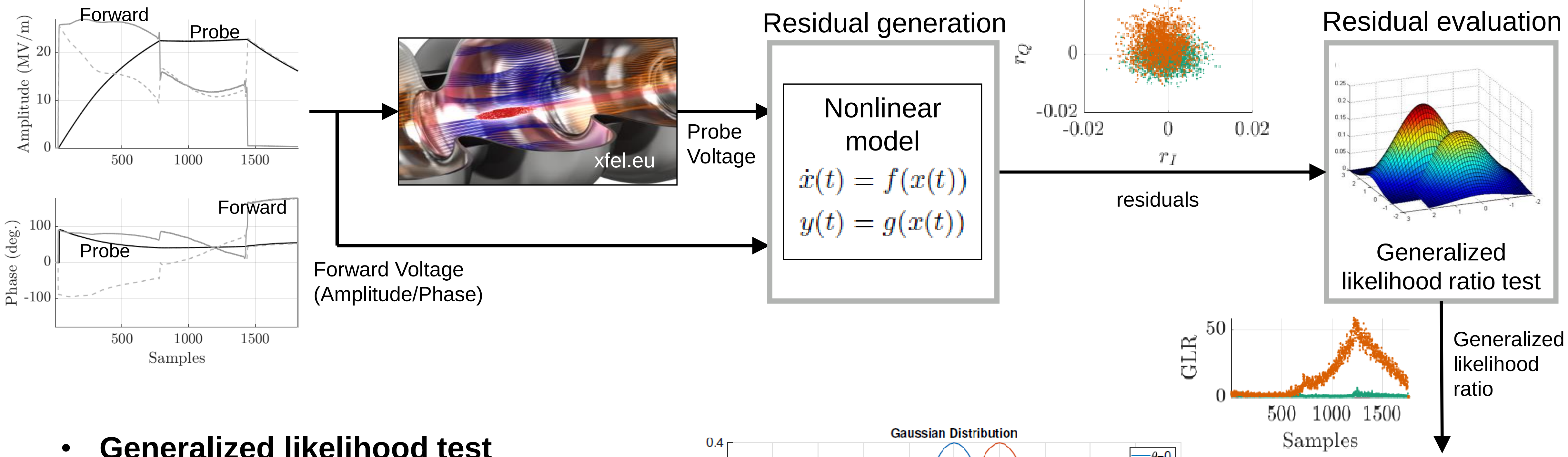
Parameter estimation

- Calculate detuning Δω and half bandwidth ω1/2 from forward and probe signals

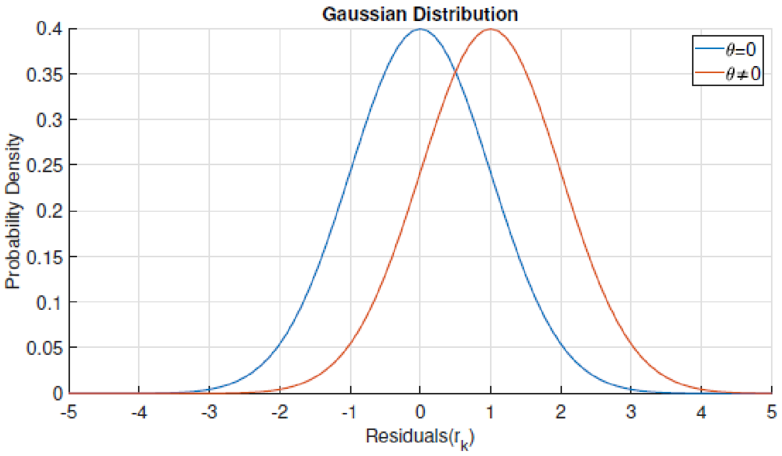
\+ Little calculation effort
\- Sensitive to noise
\+ Good physical interpretability

Advanced Fault Diagnosis

Anomaly detection for the SRF cavities



- **Generalized likelihood test**
  - Anomaly is significant change in otherwise white Gaussian process
  - GLR = Generalized likelihood ratio
  - GLR follows chi-square
  - Choose a desired false positive rate

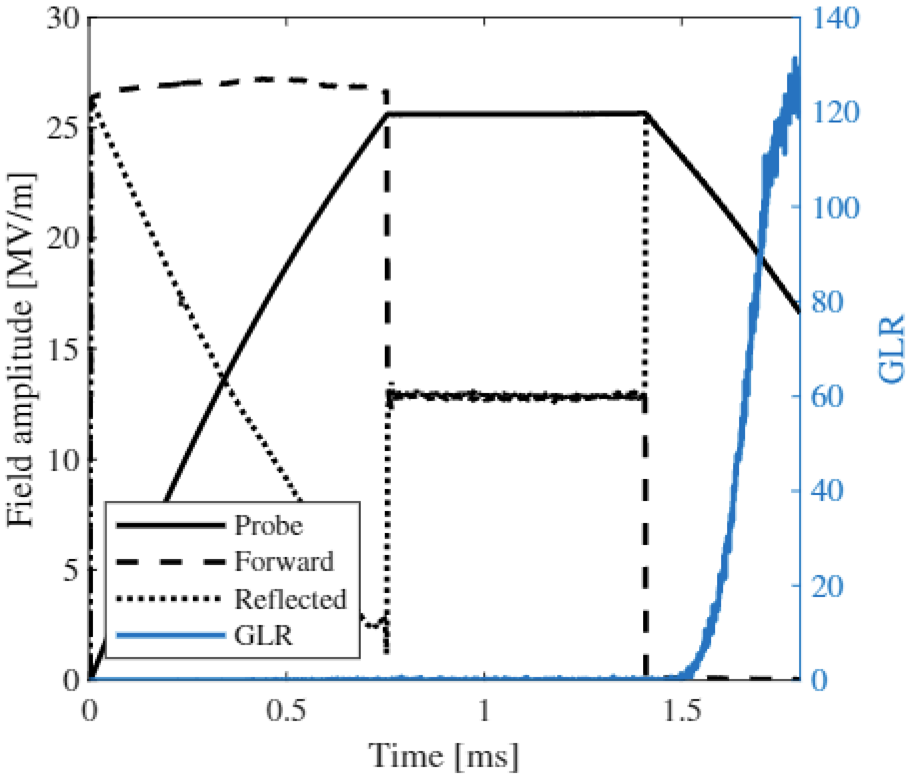


Residuals

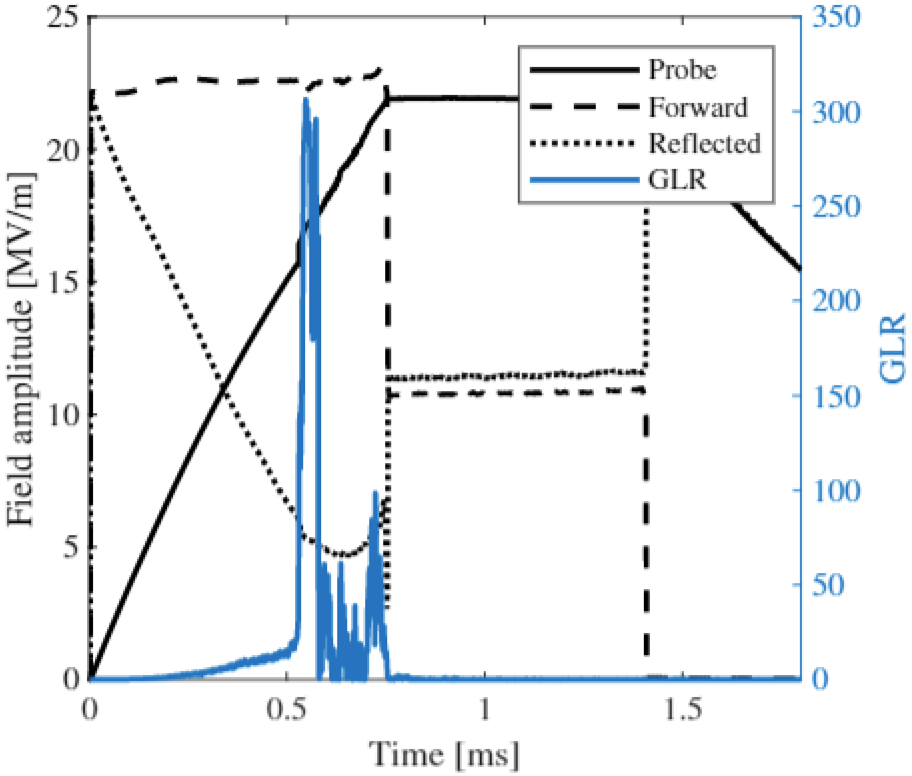
Generalized likelihood ratio

- Amplification of small anomalies
- Clear distinction between different kind of faults

Quench

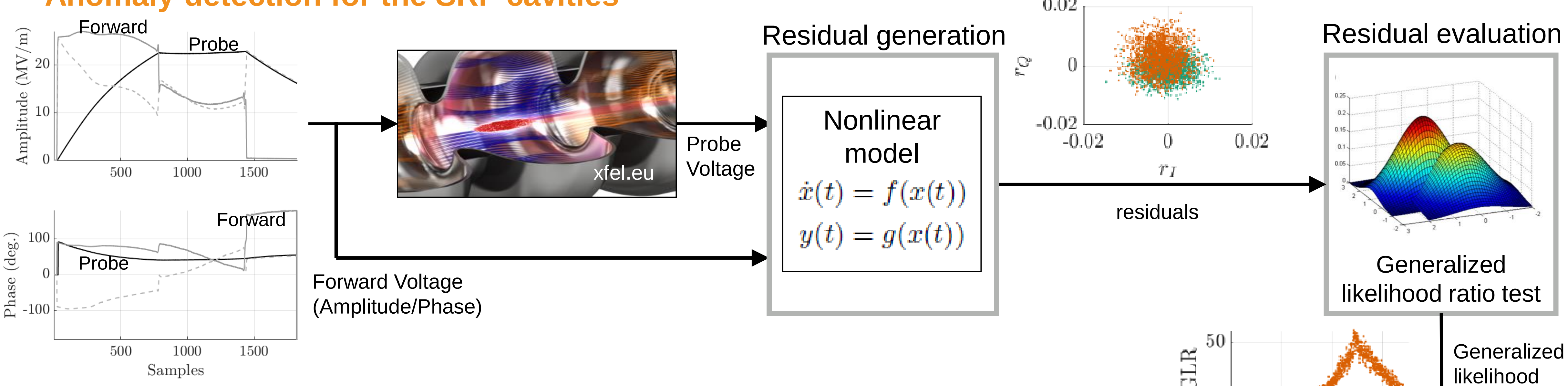


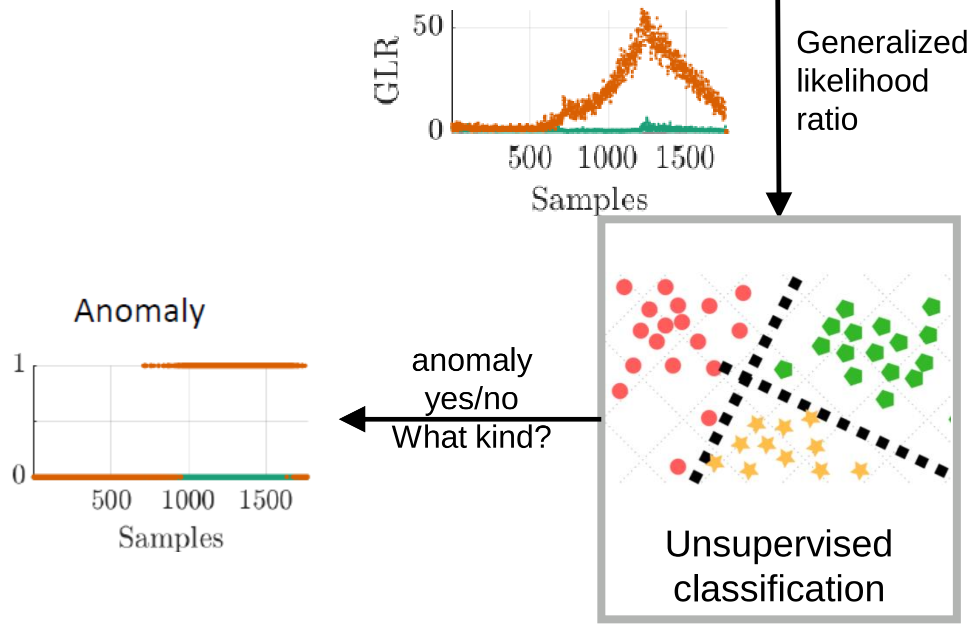
Field emitter



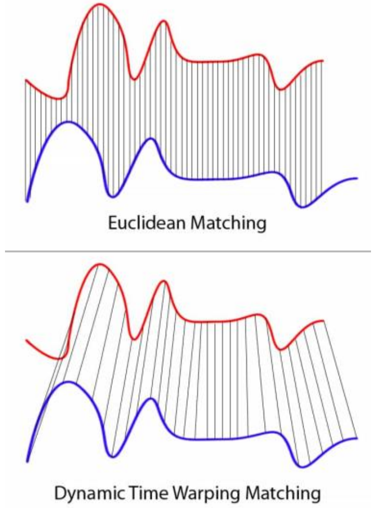
Advanced Fault Diagnosis

Anomaly detection for the SRF cavities





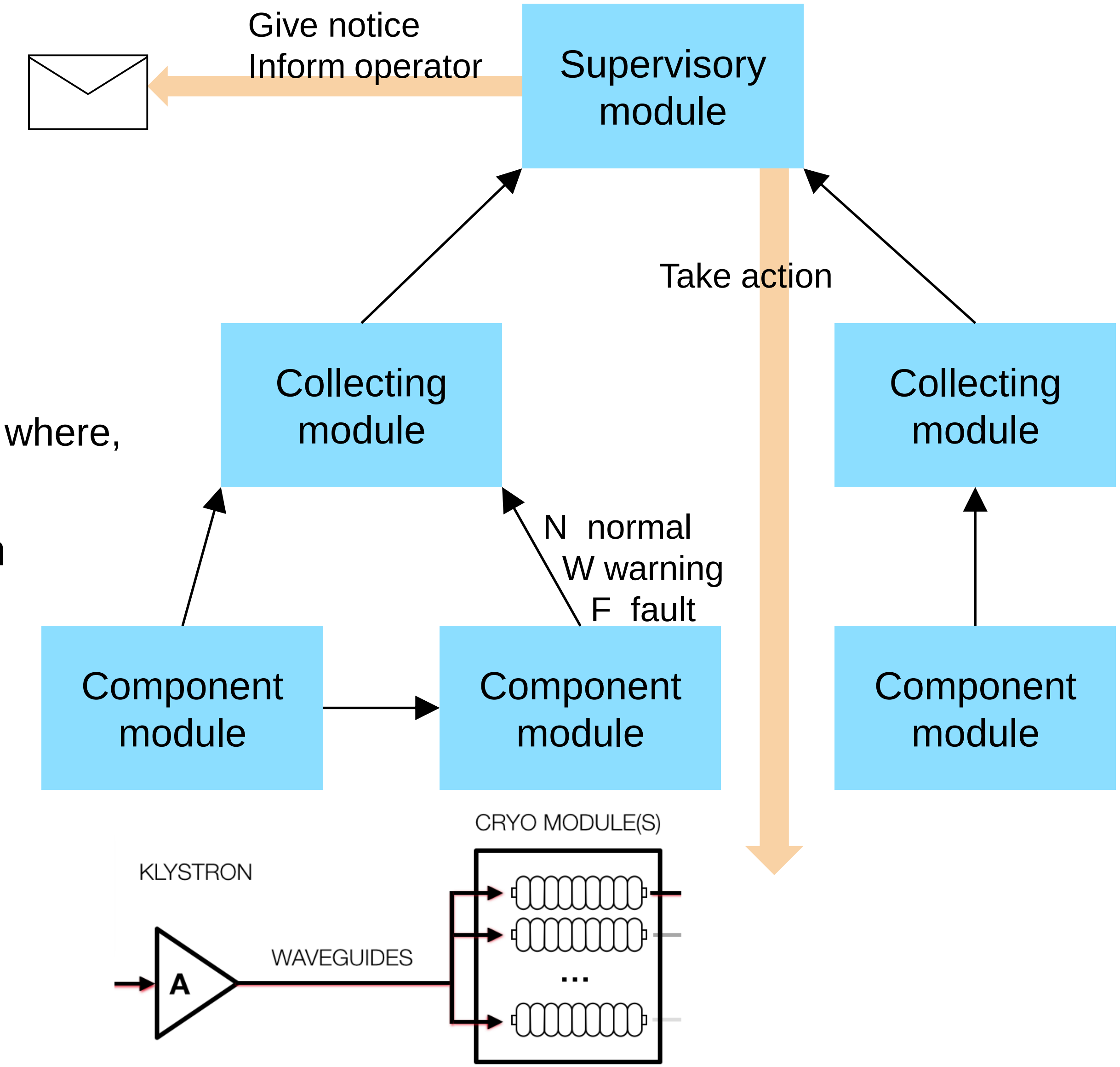
- **Unsupervised classification**
  - K-means clustering
  - Dynamic time warping
  - Or Euclidian norm but on a subset of the time signal



Summary and Conclusion

What's next and what are the challenges

- Fast model is needed to calculate the residual
- Goal: Online fault detection on all 808 cavities with
  - Allows for instantaneous automatic reaction
  - Direct feedback to operators
  - Online root cause analysis (what kind of failure occurred, where, etc.)
- In principle: calculation is real-time capable (has been demonstrated for a few cavities)
  - BUT: Bandwidth limitations! All signals need to be collected fast enough
- Infrastructure has been built up (Trip Event Logger):
  - Modular (other modules can easily be added
  - Can deal with different control systems
  - Can switch between online and offline data



Conclusion

Models can support operations

- Models / digital twins help
- If it is only to test infrastructure
- But also for algorithm training and for model-based algorithms
- Requirements:
  - Speed: Online capable to fast “enough”
  - Robustness: Need to deal with or include errors (quadrupole offsets, different field energy, etc.)
  - Performance: Accurate enough (really depends on the application)

Thank you

Contact

DESY. Deutsches Elektronen-Synchrotron

www.desy.de

Annika Eichler
MSK
annika.eichler@desy.de
+49 40 8998 4041