

Towards Natural Language-driven Autonomous Particle Accelerator Tuning

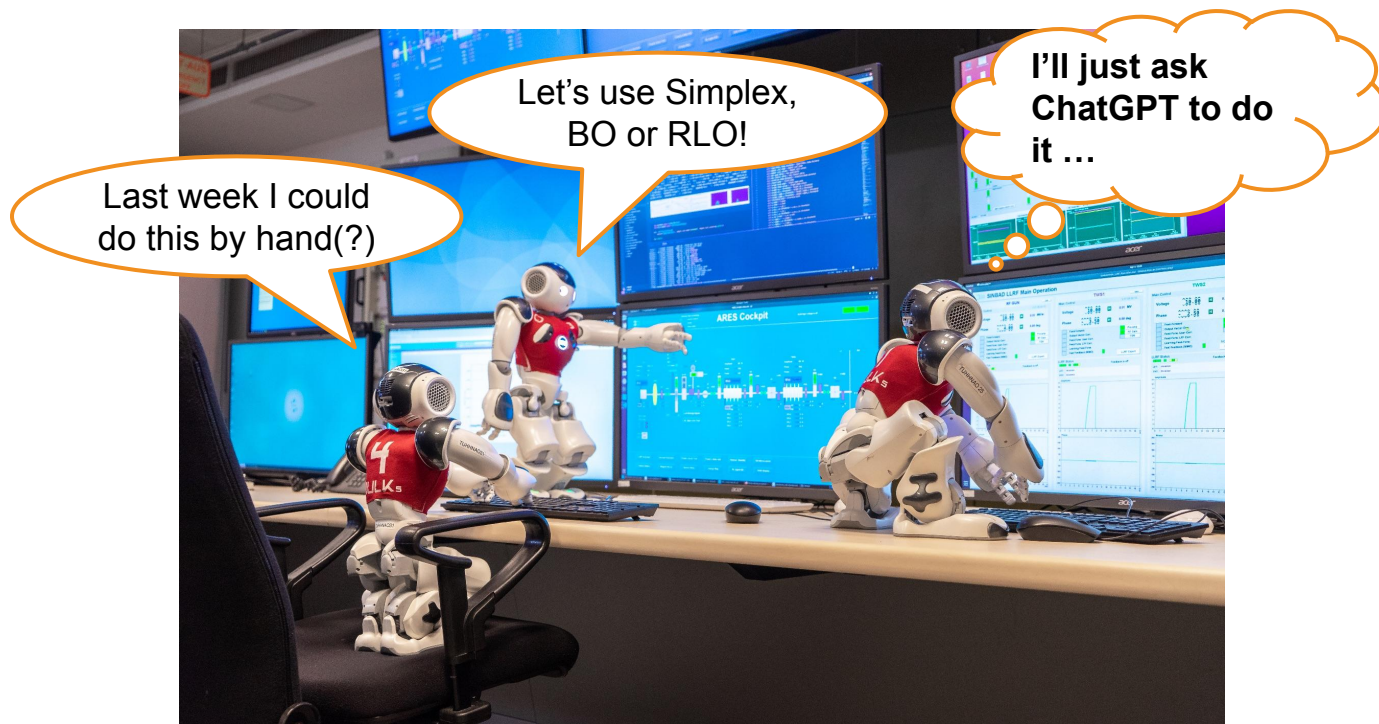
MT ARD ST3 Meeting



Jan Kaiser, Annika Eichler and Anne Lauscher
Darmstadt, 5 July 2024

An Oversimplified History of Autonomous Accelerator Tuning

From human intelligence over optimisation to artificial intelligence



Let's Ask ChatGPT to Do It ...

Questions

large language models (LLMs)

Can ~~ChatGPT~~ tune a particle accelerator?

How would that be implemented?

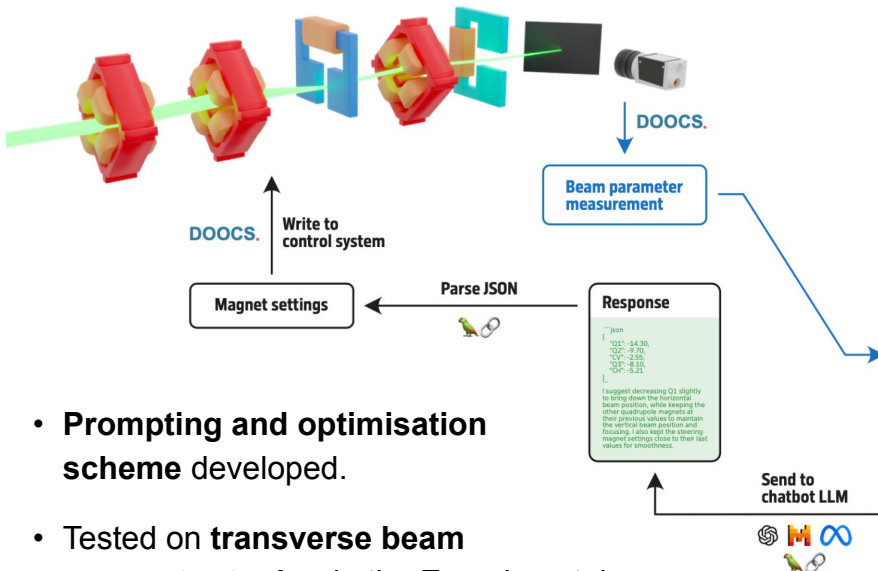
Teaser

Yes and no ...

(a) Successful episodes

	Tuning	Explained	CoT	Optimisation
Gemma 2B	7	5	6	5
Gemma 7B	-	8	-	2
GPT 3.5 Turbo	-	2	-	9
GPT 4	-	4	-	9
GPT 4 Turbo	2	5	6	9
Llama 2 7B	-	4	-	1
Llama 2 13B	-	1	-	8
Llama 2 70B	-	3	-	8
Orca 2 7B	-	3	-	6
Orca 2 13B	-	2	-	2
Vicuna 7B 16K	-	2	-	3
Mistral 7B	-	1	-	0
Mixtral 8x7B	8	7	3	5
Starling LM 7B	-	6	-	1

Reinforcement learning	9
Bayesian optimisation	9
Extremum seeking	9
Random search	0
Do nothing	0



- Prompting and optimisation scheme developed.
- Tested on **transverse beam parameter tuning** in the Experimental Area section at **ARES accelerator**.
- **Yes!** It works, if you choose the right model and prompt. A neural network trained for text completion can tune an accelerator and perform optimisation.

Prompt (explained tuning prompt)

Human: Now you will help me optimise the horizontal and vertical position and size of an electron beam on a diagnostic screen in a particle accelerator.

You are able to control five magnets in the beam line. The magnets are called Q1, Q2, C1, Q3 and C2.

Q1, Q2 and Q3 are quadrupole magnets. When their k1 strength is increased, the beam becomes more focused in the horizontal plane and more defocused in the vertical plane. When their k1 strength is decreased, the beam becomes more focused in the vertical plane and more defocused in the horizontal plane. When their k1 strength is zero, the beam is not focused in either plane. Quadrupole magnets might also tilt the beam in the horizontal or vertical plane depending on their k2 strength, when the beam does not travel through the centre of the magnet. The range of the k1 strength is -200 to 200 m^-2.

C1 is vertical steering magnet. When its deflection angle is increased, the beam is steered upwards. When its deflection angle is decreased, the beam is steered downwards. The range of the deflection angle is -4.0 to 4.0 mrad.

C2 is horizontal steering magnet. When its deflection angle is increased, the beam is steered to the right. When its deflection angle is decreased, the beam is steered to the left. The range of the deflection angle is -4.0 to 4.0 mrad.

You are optimising four beam parameters: m_x , σ_{m_x} , m_y , σ_{m_y} . The beam parameters are measured in millimetres [mm]. The target beam parameters are:

```

{
  "m_x": 1.20,
  "sigma_m_x": 0.11,
  "m_y": 1.20,
  "sigma_m_y": 0.06
}

```

Task description

Below are previously measured pairs of magnet settings and the corresponding observed beam parameters.

Magnet settings:

```

{
  "Q1": -14.30,
  "Q2": -9.10,
  "Q3": -4.05,
  "C1": 8.21,
  "C2": -9.21
}

```

Beam parameters:

```

{
  "m_x": -1038.65,
  "sigma_m_x": 205.55,
  "m_y": -292.77,
  "sigma_m_y": 226.84
}

```

Previous samples

Give me new magnet settings that are different from all pairs above. The magnet settings you should provide should lead to beam parameters closer the target or if you do not have enough information yet, maximise information gain for finding new beam parameters. Do not set any magnet setting to zero. Smooth changes relative to the last magnet settings are preferred.

The output should be a markdown code block formatted in the following schema, including the leading and trailing "```" and "```":

```

```json
{
 "Q1": float // k1 strength of the first quadrupole magnet
 "Q2": float // k1 strength of the second quadrupole magnet
 "Q3": float // k1 strength of the third quadrupole magnet
 "C1": float // k2 strength of the first steering magnet
 "C2": float // k2 strength of the second steering magnet
}
```

```

Do not add comments to the output JSON.

Output instructions

Come to the poster!

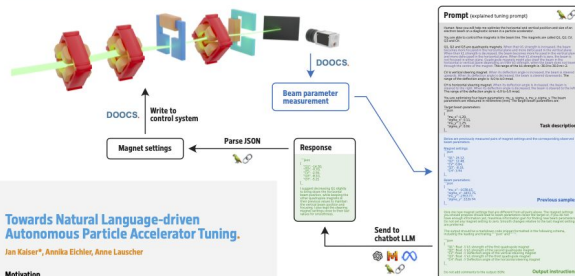
Contact

DESY. Deutsches
Elektronen-Synchrotron

www.desy.de

Jan Kaiser
Machine Beam Controls (MSK)
jan.kaiser@desy.de

Large language models (LLMs) like GPT-4, which are really **only trained to do text completion**, can be (ab)used as optimisers to directly and **autonomously tune a particle accelerator**.



Towards Natural Language-driven Autonomous Particle Accelerator Tuning.

Jan Kaiser*, Anika Eichler, Anne Leusser

Motivation

- Large language models (LLMs) have most recently demonstrated incredible capabilities beyond simple text completion, such as programming and problem solving, with **emergent behaviours** appearing as models become larger and better trained.
- This raises the question: **Can LLMs tune a particle accelerator?**
- Ultimately, this could be a step towards accelerator operators defining goals for accelerator operation through natural language, with LLMs figuring out how to get there.

Implementation

- We engineered a prompt made up of three main components:
 - In the **task description**, we explain the task and define a goal.
 - We then list a history of **previously collected samples** of inputs and resulting outputs.
 - At last, we provide **instructions for the model output**, where the model is asked to output the next magnet settings and providing formal instructions for later parsing.
- We test four different **prompting strategies** following the above scheme:
 - **Tuning prompt:** Frames the task in terms of accelerator tuning, where the model is asked to find magnet settings that result in specific target beam parameters.
 - **Explained tuning prompt:** We explain in simple terms how quantities magnet etc. work and how they influence the beam parameters. Otherwise like the tuning prompt.
 - **Chain-of-thought prompt:** Ask the model to explain its reasoning before providing the next magnet settings. Otherwise like explained tuning prompt.
 - **Optimization prompt:** Frames the task as a simple optimization, where the output is an objective value. The model assumes it's tuning an accelerator.
- Implementation of prompt creation with LangChain. LLMs are run through the OpenAI API or with Cohere on a100 GPUs on the DESY Maxwell cluster.

Results

- Success depends on model and prompting scheme:
 - For example, **GPT-4 Turbo with optimization prompt is reliably successful**.
 - Some models, like GPT-4 Turbo, fail because the prompt is too constrained or providing too many magnet settings. Others, like Mistral 7B, fail to show reasonable behaviour.
- Performance **does not match state of the art**, such as RL and BO.
- Intensive resource requirements: Up to 5 USD, 2.5 L water and 36 g of CO₂ (same as modern fridge in 1.1 h) per tuning run.

Outlook

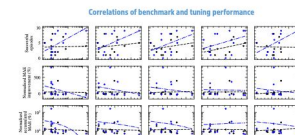
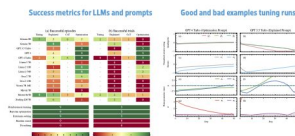
- Test potentially better suited prompting schemes like **ReAct**.
- Integrate with RLO/BO, lookbooks, etc. towards **accelerator copilot** (e.g. GAIA).

*Corresponding author: jan.kaiser@desy.de

Deutsches Elektronen-Synchrotron DESY
A Research Centre of the Helmholtz Association



Scan me to download the full paper!



SINBAD HIRX Universität Hamburg

