Do you really think about consequences?

Bridging Classical Control and Reinforcement Learning for Delayed Outcome Optimisation

O. Mironova[1], L. Fischl[2], T. Gallien[3], S. Hirlaender[1]

1 IDA Lab, Paris Lodron University of Salzburg, Austria
2 MedAustron GmbH, Austria
3 Institute for Robotics and Flexible Production, Joanneum Research, Austria

Abstract

This study explores advanced strategies for optimal control in systems with delayed consequences, using beam steering in the AWAKE electron line at CERN as a benchmark. We formulate the task as a constrained optimization problem within a continuous, primarily linear Markov Decision Process (MDP), incorporating measured system parameters and realistic termination criteria. A wide range of approaches is implemented and compared, including classical response matrix inversion, control-theoretic methods, reinforcement learning, and structured model-based techniques.

While classical methods like matrix inversion offer accurate convergence, they fail to account for delayed effects and are sensitive to noise. Control-theoretic approaches, such as Model Predictive Control (MPC), leverage known dynamics and handle delays effectively when models are available. Data-driven methods, including Proximal Policy Optimization (PPO), adapt to uncertainty and non-linearities but require large amounts of data. Structured GP-MPC bridges both paradigms by learning system dynamics using Gaussian Processes while respecting the problem's causal structure, significantly improving robustness and sample efficiency.

Our experiments highlight key performance differences, particularly in how each method handles delayed outcomes, noise, and structural assumptions. We find that exploiting the causal structure of the problem provides a notable advantage, and that method choice ultimately involves trade-offs between adaptability, data efficiency, and computational cost. These findings offer guidance for applying advanced control strategies in high-dimensional, partially structured environments.

Problem definition

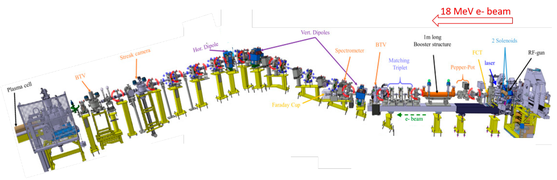
- **AWAKE electron line:** excellent environment to test optimisation and control algorithms with a high fidelity accurate simulation available
- **Goal:** steer the electron beam as fast as possible to a desired target beam
- Formulation as linear MDP with 10 dimensional continuous state and action space:
  - Episodic task:
    - Randomly initialised starting positions $\mathbf{S}_0$
    - Termination if:
      - RMS to target beam is smaller than a threshold
      - Beam hits the beam pipe (safety constraint)
    - Truncation if number of maximal interactions without termination
  - **Actions:** Ten dipole magnets limited
  - **States:** Ten beam position monitors
  - **Reward:** RMS value distance to target trajectory

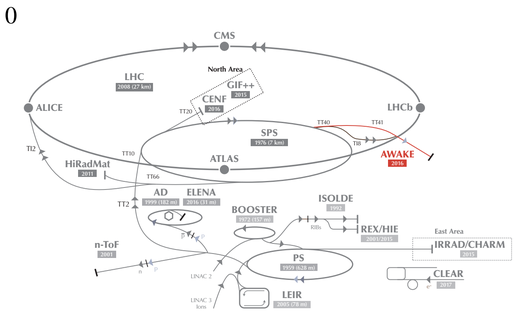
The system dynamics is given by: $\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{B}\mathbf{a}_t$, with $\mathbf{B} = \begin{pmatrix} b_{11} & 0 & \cdots & 0 \\ b_{21} & b_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix}$ (lower triangular matrix)

- The observable is: $\mathbf{o}_t = \mathbf{s}_t + \varepsilon_t$, with $\varepsilon_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$
- Characterisation of the problem:
  - The system is linear and controllable. However, delayed consequences (i.e., the effects limited actions and a certain threshold for a successful termination) and partial observability add complexity to the control problem making it non-trivial.
- Mathematical formulation of the constrained optimisation problem:
  - With a finite horizon T (with $T \leq T_{\max}$), the optimisation problem is formulated as follows:

$$\min_{\{\mathbf{a}_t\}_{t=0}^{T-1}} \mathbb{E}_{\varepsilon_t}\left[\sum_{t=0}^{T-1} R_t(\mathbf{s}_t, \mathbf{a}_t, \varepsilon_t)\right]$$

$$\text{subject to} \quad \mathbf{s}_{t+1} = \mathbf{s}_t + B\,\mathbf{a}_t + \varepsilon_t, \qquad ,\forall t > 0$$
$$a_t^i \in [a_{\min}^i, a_{\max}^i], \quad i = 1,\ldots,10, \forall t > 0$$
$$\mathbf{s}_t \in S_{\text{safe}}, \forall t > 0$$
$$\mathbf{s}_0 = \mathbf{S}_0$$





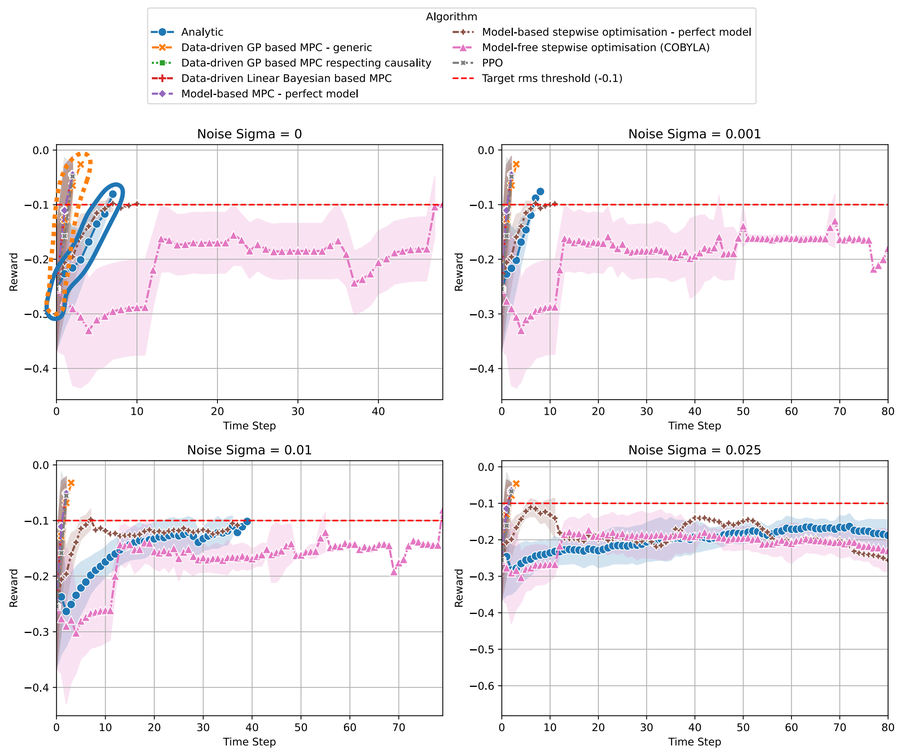
Motivation

- **Under what circumstances should we employ reinforcement learning versus traditional optimisation techniques?**
- **Even relatively simple setups, like the steering problem presented here, can exhibit delayed consequences.**
- **How can we systematically account for and respect these delayed consequences?**
- **Can we leverage any inherent structure in the problem to train faster and obtain more robust policies? Here, we exploit the causal structure.**

Experiments

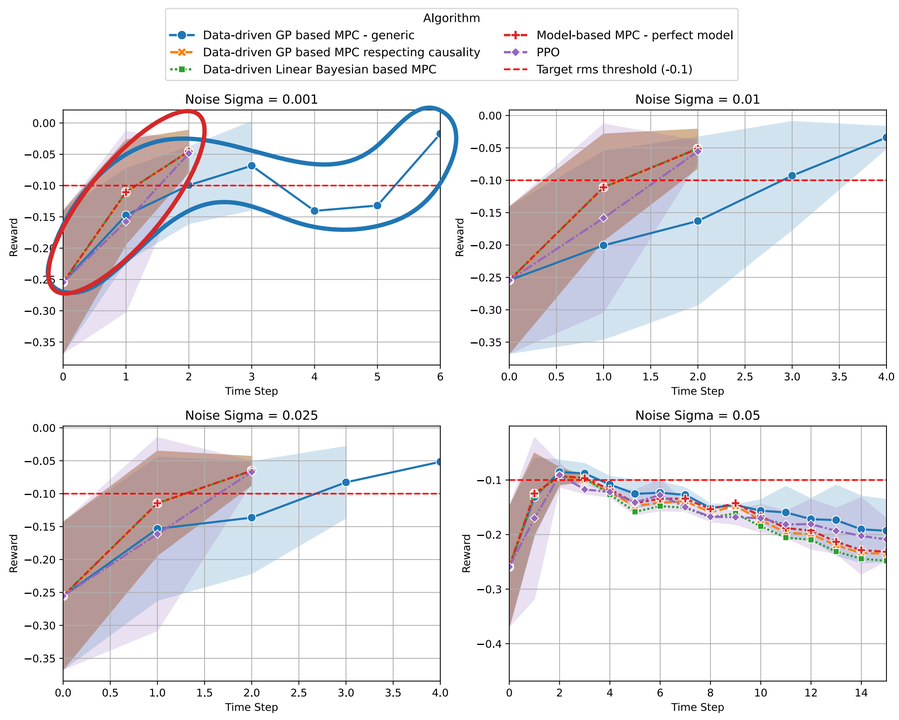
Delayed consequences:

- We evaluate trained policies using both data-driven methods and model-based approaches derived from the known system model.
- A clear distinction emerges between algorithms that can manage delayed consequences and those that rely on greedy strategies.
- Our experiments suggest that algorithms lacking mechanisms to account for delayed consequences exhibit also an increased sensitivity to noise.



Structure of the problem:

- Methods that leverage the problem's causal structure produce more robust and effective final policies than higher-capacity algorithms that lack structural insight.
- These approaches also show markedly improved sample efficiency.
- Notably, the structured GP-MPC method reduces online optimisation time by an order of magnitude.



Methods

Method	Needs model	Handle delayed consequences	Computational complexity	Sample Efficiency	Robustness / Stability	Ease of Tuning / Implementation Complexity	Real-Time Applicability
Classical output feedback controller	Yes	No	Low	N/A (perfect model used)	High (for well-modeled linear systems)	Easy	High
Model-predictive control with the perfect model	Yes	Yes	High	N/A (perfect model used)	High if the model is perfect	Moderate	Low/Moderate
Greedy optimisation with the perfect model	Yes	No	Medium	N/A (perfect model used)	Low (cannot handle delayed effects)	Easy	High
Data-driven MPC based on Bayesian linear regression	No	Yes	Low	High	Moderate	Moderate	High
Data-driven MPC using a structured model based on Gaussian processes	No	Yes	High	High	Moderate to High	Difficult	Low/Moderate
Data-driven MPC with a generic model based on Gaussian processes	No	Yes	High	Medium	High	Difficult	Low
Proximal policy optimisation	Yes (via simulator/ environment)	Yes	Low	Low	High	Moderate	High
Model-free step wise optimisation	No	No	Low	Low	Low	Easy	High

Conclusion

- We implemented a broad array of methods to derive an optimal policy.
- A clear performance disparity emerges between algorithms that account for delayed consequences and those that do not.
- Classical response matrix inversion converges reliably on average, but its disregard for delayed effects limits both speed and robustness.
- When a system model is available, control-theoretic methods can match the performance of reinforcement learning while offering greater interpretability.
- Exploiting the causal structure of the problem significantly boosts robustness and sample efficiency in data-driven approaches.
- Ultimately, each method involves trade-offs, and no single approach dominates across all criteria (see table).

PARIS LODRON UNIVERSITÄT SALZBURG

olga.mironova@plus.ac.at
simon.hirlaender@plus.ac.at

Code available at:
S. Hirlaender, O. Mironova, L. Fischl, T. Gallien (2025). Do you really think about consequences? Bridging Classical Control and Reinforcement Learning for Delayed Outcome Optimization:
https://github.com/MathPhysSim/Do_you_really_think_about_consequences.git