

Explaining Data-Driven Control in Autonomous Systems: A Reinforcement Learning Case Study

Mengbang Zou, Adolfo Perrusquía and Weisi Guo

Abstract—Explaining what does a data-driven control algorithm learn plays a crucial role for safety critical control of autonomous platforms in transportation. This is more acute in reinforcement learning control algorithms, where the learned control policy depends on various factors that are hidden within the data. Explainable artificial intelligence methods have been used to explain the outcomes of machine learning methods by analysing input-output relations. However, data-driven control does not pose a simple input-output mapping and hence, the resulting explanations lack depth. To deal with this issue, this paper proposes a explainable data-driven control method that allows to understand what the data-driven method is learning from the data. The model is composed by a Q-learning algorithm enhanced by a dynamic mode decomposition with control (DMDc) algorithm for state-transition function estimation. Both the Q-learning and DMDc provides the elements that are learned from the data and allow the construction of counterfactual explanations. The proposed approach is robust and does not require hyperparameter tuning. Simulation experiments are conducted to observe the benefits and challenges of the method.

I. INTRODUCTION

The proliferation of autonomous platforms requires the design of trustworthy data-driven algorithmic tools that guarantee safety to people and national infrastructure. This is more acute in the transportation sector where the potential applications are highly diverse, e.g., fire detection, package delivery, and aids and supplies [1]–[4]. These applications require that the data-driven algorithm fulfils safety requirements, e.g., stability and robustness. Here, on-line data-driven control algorithms are the most challenging methods since the control learning and data collection are applied simultaneously, such as in reinforcement learning (RL) control.

Explainable artificial intelligence (XAI) tools have been adopted to explain the outcomes obtained by data-driven control algorithms given a combination of input features. XAI also aims to establish the mechanisms to obtain a desired behaviour [5] by modifying one or more input features. Most of the progress in XAI has been focused on supervised machine learning models [6]. These techniques can be classified in either gradient-based methods [7], [8] or perturbation methods [9], [10]. These methods aim to identify the importance weight [11], [12] of each input feature locally by perturbing the input features or observing

how the output of the model changes with respect to an input feature [13]. In terms of data-driven control methods such as reinforcement learning, the output policy is not a simple mapping from inputs to outputs as in the supervised learning case. In contrast, the learning process is dynamic and the inner learning factors are still hidden for traditional XAI methods. This in consequence makes the explanations lack of meaningful interpretations.

Other kind of explanations that has been adopted by responsible AI groups are known as counterfactual explanations [14]. These kind of explanations aim to establish clear common sense cause-effect relations in idealistic worlds [15]. In contrast to classic XAI methods, counterfactuals provide a larger picture of the factors involved in the learning process of data-driven and AI methods, and the mechanisms to obtain the desired outcome. To best of our knowledge, the design of counterfactual explanations applied in reinforcement learning agents is still an open problem and we are interested to contribute in filling this gap.

Inverse Reinforcement Learning (IRL) and imitation learning (IL) [16]–[18] have been adopted to explain RL models. Here, the optimisation of the cumulative reward determines the control policy that the autonomous system needs to perform. Hence, it is natural to use IRL/IL to explain RL models. However, there are some specific challenges of using IRL/IL: 1) since it is an inverse problem, then multiple reward functions can achieve the same behaviour which can lead to biased explanations, and 2) the goal of XAI is lost, that is, the reward function cannot explain what does the RL agent learns in order to design improvement mechanisms.

RL algorithms have several inner elements in the learning process which include: i) the policy that maps from states to control inputs [19], ii) the value function that determines how good is the followed policy [20], iii) a reward function that defines a cost signal that is optimised in an infinite horizon [21], iv) the state-transition function that determines how the behaviour of the drone is modified by applying a control policy in a certain state [22], [23], and v) an exploration signal which is used to ensure that the optimal control policy is obtained [24], [25]. Here, most of the RL algorithms focused on obtaining both the value function and subsequently the control policy in value or policy iteration algorithms. However, the learning of the state-transition function has been put aside despite of being the main factor that determines the optimal value function and thus, the optimal control policy. Here, the exploration signal plays a major role in the identification of the state-transition model which needs to be considered as an explanatory factor of the

This work was supported by the Engineering and Physical Sciences Research Council under Research Grants EP/V026763/1, EP/X040518/1 and EP/Y037421/1.

Mengbang Zou, Adolfo Perrusquía and Weisi Guo are with the School of Aerospace, Transport and Manufacturing, Cranfield University, MK43 0AL Bedford, UK.

adolfo.perrusquia-guzman@cranfield.ac.uk.

overall decision-making process.

We hypothesize that to provide the complete picture of what does a RL agent is learning, we need to determine what state-transition model is inferred from the data. Therefore, this paper proposes a counterfactual-based approach to explain data-driven control outputs based on the quality of the data collected from the system's trajectories. This is done by incorporating an additional module in the reinforcement learning model which is in charge of estimating the state-transition model. This module consists in a dynamic mode decomposition with control (DMDc) algorithm that estimates the state-transition model in each episode of the RL algorithm. Parameter estimates convergence is achieved under the fulfilment of a persistent of excitation condition. Counterfactual explanations are then designed from the outputs of the RL agent and the DMDc module. Simulation experiments are conducted to verify the feasibility of the approach.

II. METHODOLOGY

The proposed methodology is depicted in Fig. 1. The diagram is a basic optimal/near optimal data-driven control using reinforcement learning composed by a discrete-time linear system controlled by a Q-learning algorithm. The Q-learning algorithm is enhanced by the incorporation of a dynamic mode decomposition with control (DMDc) algorithm that allows the inference of the state-transition matrices in each episode of the Q-learning algorithm. These additional elements are required to obtain the complete picture of what the data-driven control method is learning in order to design counterfactual explanations.

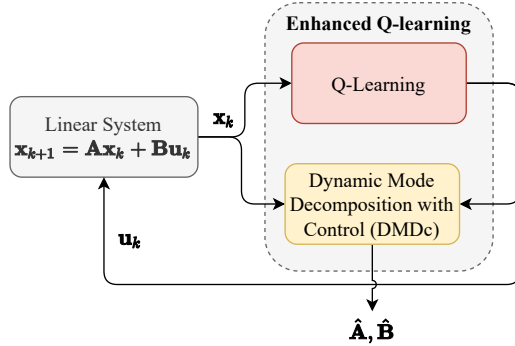


Fig. 1. Proposed methodology

Fig. 2 shows the elements of any reinforcement learning algorithm. In the top of the pyramid is the control policy, which is the final goal/outcome of any RL architecture. The second level is the Q -function which provides the information of how good is a policy and is used to update and derive the optimal control policy. The third level is associated to the inference of the state-transition matrices, which define all the dynamic properties of the system and are responsible of the construction of the Q -function. The fourth level is the reward design which injects a desired behaviour to the system. All of these levels define the design elements of the

RL that enables the design of counterfactual explanations proposed in this paper.

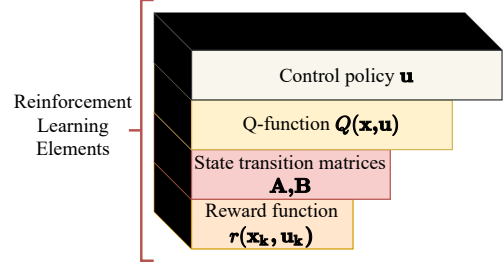


Fig. 2. Elements of any reinforcement learning architecture for counterfactual explanations design

III. OPTIMAL CONTROL PRELIMINARIES

Consider an autonomous platform model whose dynamics fulfil the following time invariant linear system,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (1)$$

where $\mathbf{x}_k \in \mathbb{R}^n$ defines the state vector of the system, $\mathbf{u}_k \in \mathbb{R}^m$ is the driven control input, $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$ define the state-transition matrices which are assumed to be unknown, and $\mathbf{x}_0 \in \mathbb{R}^n$ is the initial condition. Here, the pair $(\mathbf{A}, \mathbf{B})$ is assumed to be controllable.

In an optimal control context, the following cost index [26] is typically used

$$J = \sum_{i=k}^{\infty} r_i = \sum_{i=k}^{\infty} (\mathbf{x}_i^\top \mathbf{Q} \mathbf{x}_i + \mathbf{u}_i^\top \mathbf{R} \mathbf{u}_i), \quad (2)$$

where $\mathbf{Q} = \mathbf{Q}^\top \succeq 0 \in \mathbb{R}^{n \times n}$ and $\mathbf{R} = \mathbf{R}^\top \succ 0 \in \mathbb{R}^{m \times m}$ are weight matrices that shape the behaviour of the autonomous platform into a desired one. $r_k = \mathbf{x}_k^\top \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R} \mathbf{u}_k$ is the reward function that the RL agent aims to optimize. The following Q -function is defined in order to solve the optimal control problem,

$$Q(\mathbf{x}_k, \mathbf{u}_k) := \mathbf{z}_k^\top \mathbf{S} \mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}^\top \begin{bmatrix} \mathbf{S}_{xx} & \mathbf{S}_{xu} \\ \mathbf{S}_{xu}^\top & \mathbf{S}_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}, \quad (3)$$

where $\mathbf{z}_k = [\mathbf{x}_k^\top, \mathbf{u}_k^\top]^\top \in \mathbb{R}^{n+m}$ and $\mathbf{S} = \mathbf{S}^\top \succ 0 \in \mathbb{R}^{(n+m) \times (n+m)}$ with $\mathbf{S}_{xx} = \mathbf{A}^\top \mathbf{P} \mathbf{A} + \mathbf{Q}$, $\mathbf{S}_{xu} = \mathbf{A}^\top \mathbf{P} \mathbf{B}$ and $\mathbf{S}_{uu} = \mathbf{B}^\top \mathbf{P} \mathbf{B} + \mathbf{R}$. Here, $\mathbf{P} = \mathbf{P}^\top \succ 0 \in \mathbb{R}^{n \times n}$ is a kernel matrix that solves the following discrete algebraic Ricatti equation (DARE) [27], [28]

$$\mathbf{A}^\top \mathbf{P} \mathbf{A} + \mathbf{Q} - \mathbf{A}^\top \mathbf{P} \mathbf{B} (\mathbf{R} + \mathbf{B}^\top \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{P} \mathbf{A} = \mathbf{P}. \quad (4)$$

The optimal Q -function satisfies

$$Q^*(\mathbf{x}_k, \mathbf{u}_k) = \min_{\mathbf{u}_k \in \mathbb{R}^m} Q(\mathbf{x}_k, \mathbf{u}_k), \quad (5)$$

and hence the optimal control policy $\mathbf{u}_k^*$ can be obtained by

$$\begin{aligned} \mathbf{u}_k^* &= \arg \min_{\mathbf{u}_k \in \mathbb{R}^m} Q(\mathbf{x}_k, \mathbf{u}_k) \\ &= -\mathbf{S}_{uu}^{-1} \mathbf{S}_{xu}^\top \mathbf{x}_k := -\mathbf{K} \mathbf{x}_k. \end{aligned} \quad (6)$$

The control policy (6) is a model-based policy that can be computed using the well-known linear quadratic regulator (LQR) [29] under known state-transition matrices $\mathbf{A}$ and $\mathbf{B}$.

IV. ENHANCED Q-LEARNING

Since the state-transition model is unknown, then a Q-learning algorithm [30] is developed to find a model-free controller from the data. To this end, the elements of matrix S are estimated using the following Q -function approximation,

$$\begin{aligned}\widehat{Q}(\mathbf{x}_k, \mathbf{u}_k) &= \mathbf{z}_k^\top \widehat{S} \mathbf{z}_k, \\ \widehat{Q}(\mathbf{x}_k, \mathbf{u}_k) &= (\mathbf{z}_k^\top \otimes \mathbf{z}_k^\top) \text{vech}(\widehat{S}) := \phi_k^\top \widehat{\boldsymbol{\theta}}^i,\end{aligned}\quad (7)$$

where $\widehat{S} = \widehat{S}^\top \succ 0 \in \mathbb{R}^{(n+m) \times (n+m)}$ is a matrix estimate of S , $\phi_k = (\mathbf{z}_k \otimes \mathbf{z}_k)$ defines a complete set of basis functions, and $\widehat{\boldsymbol{\theta}}^i \in \mathbb{R}^{\frac{1}{2}(n+m)(n+m+1)}$ is a vector of parameters whose elements correspond to the upper triangular terms of the matrix estimate $\widehat{S}$. The Q-learning algorithm rewrites the cost index (2) in its recursive form to reformulate the optimal control problem in terms of the Q -function

$$\begin{aligned}\widehat{Q}(\mathbf{x}_k, \mathbf{u}_k) &= r_k + \widehat{Q}(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) \\ (\phi_k - \phi_{k+1})^\top \widehat{\boldsymbol{\theta}}^i &= r_k.\end{aligned}\quad (8)$$

A batch least-squares rule is used to estimate the vector of parameters $\widehat{\boldsymbol{\theta}}^i$ in each episode of the Q-learning algorithm. Here, $\kappa \geq w = \frac{1}{2}(n+m)(n+m+1)$ samples of $(\phi_k - \phi_{k+1})$ and r_k are collected to construct a system of equations and compute the vector of parameters $\widehat{\boldsymbol{\theta}}^i$ as follows

$$\widehat{\boldsymbol{\theta}}^i = (\Phi \Phi^\top)^{-1} \Phi \mathbf{r}, \quad (9)$$

where

$$\Phi = \begin{bmatrix} \phi_i - \phi_{i+1} \\ \vdots \\ \phi_i - \phi_{i+1} \end{bmatrix} \in \mathbb{R}^{w \times \kappa}, \quad \mathbf{r} = \begin{bmatrix} r_i \\ \vdots \\ r_i \end{bmatrix} \in \mathbb{R}^\kappa.$$

The terms of $\widehat{S}$ are recovered by applying the symmetric matricization as follows $\widehat{S} = \text{mat}(\widehat{\boldsymbol{\theta}}^i)$. Then, the optimal control policy can be easily computed as

$$\widehat{\mathbf{u}}_k^* = -\widehat{S}_{uu}^{-1} \widehat{S}_{xu} \mathbf{x}_k = -\widehat{K} \mathbf{x}_k. \quad (10)$$

A. Dynamic Mode Decomposition with Control (DMDc)

The proposed enhanced Q-learning algorithm incorporates an additional module to estimate from the data the state-transition matrices. To this end, the number of data samples are slightly increased to $\iota = n(n+m) > \kappa$ and construct the following matrices $\mathbf{X} = [\mathbf{x}_0 \ \cdots \ \mathbf{x}_{\iota-1}] \in \mathbb{R}^{n \times \iota}$, $\mathbf{X}' = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_\iota] \in \mathbb{R}^{n \times \iota}$, and $\Upsilon = [\mathbf{u}_1 \ \cdots \ \mathbf{u}_{\iota-1}] \in \mathbb{R}^{m \times \iota}$.

In this case, the number of parameters are increased by at most $n+m$ parameters. Then, we can build the following system of equations

$$\begin{aligned}\mathbf{X}' &= \widehat{A} \mathbf{X} + \widehat{B} \Upsilon, \\ \mathbf{X}' &= \widehat{\Theta} \Omega,\end{aligned}\quad (11)$$

where $\widehat{A} \in \mathbb{R}^{n \times n}$ and $\widehat{B} \in \mathbb{R}^{n \times m}$ are estimates of the transition function matrices, $\widehat{\Theta} = [\widehat{A} \ \widehat{B}] \in \mathbb{R}^{n \times (n+m)}$, and $\Omega = [\mathbf{X}^\top, \Upsilon^\top]^\top \in \mathbb{R}^{(n+m) \times \iota}$ is a matrix constructed from the collected data. So, the optimisation problem is equivalent

to find the state-transition matrices in $\widehat{\Theta}$ that minimizes the following error function

$$E = \|\mathbf{X}' - \widehat{\Theta} \Omega\|_{\mathcal{F}},$$

where $\|\cdot\|_{\mathcal{F}}$ denotes the Frobenius norm. The solution of this optimization problem is given by a least-squares solution of the form

$$\widehat{\Theta} = \mathbf{X}' \Omega^\dagger, \quad (12)$$

where $\dagger$ refers to the Moore-Penrose pseudoinverse. DMD applies low-rank decomposition to facilitate the computation of the pseudoinverse by writing Ω as follows

$$\Omega = \mathbf{U} \Sigma \mathbf{V}^*, \quad (13)$$

where $\mathbf{U} \in \mathbb{C}^{(n+m) \times (n+m)}$ denotes the left singular vectors, $\Sigma \in \mathbb{R}^{(n+m) \times \iota}$ are the singular values, and $\mathbf{V}^* \in \mathbb{C}^{\iota \times \iota}$ are the complex conjugate of the right singular vectors. Then, the state transition matrices are computed as,

$$\widehat{\Theta} = \mathbf{X}' \mathbf{V} \Sigma^{-1} \mathbf{U}^*. \quad (14)$$

The state-transition matrices estimates can be used to compute a model-based version of the optimal control problem. However, we are more interested in understanding what the data-driven models are learning from the data and the mechanisms to modify it into a desired control policy that drives to a desired behaviour.

Remark 1: Parameter estimates convergence of both the Q-learning algorithm and the DMDc is ensured if the collected data in Φ and Ω fulfil a persistent of excitation (PE) condition [20].

B. Counterfactual Explanations

After we uncover all the elements learned from the data using data-driven control models, then we are able to design counterfactual explanations. The key idea of counterfactuals is to provide simple causal explanations of what should happen if something would have been different. In other words, counterfactual gives explanations of idealistic scenarios that allows to understand what should we need to do in order to obtain the desired outcome.

In the context of data-driven control models, we can formulate counterfactuals with the following structure

- ✓ “If the matrices Φ and Ω had been full rank in each episode i , then the Q-learning and DMDc methods would remain stable”.
- ✓ “If the number of steps k had been $\alpha \iota$, then the Q-learning and DMDc estimates would be close to their real values”. Here $\alpha \in \mathbb{N}$ is a scaling factor.
- ✓ “If the matrix estimate $\widehat{S}$ had been S , then the Q-learning algorithm would accurately estimate the matrices $\widehat{A}$ and $\widehat{B}$ ”.

These simple *if-then* explanations provide the information of what we need to modify to obtain a desired outcome. For example, the first counterfactual indicates that we need to ensure that the matrices constructed from the data to be full rank and hence, we need to incorporate a probing noise in the control input to ensure that the samples are linearly independent.

V. RESULTS

In this section, we test the proposed model using a continuous-time quadcopter model. The states are defined by the Cartesian positions and velocities, whilst the control inputs are given by the total thrust, and roll and pitch angles. The plant is discretized using a sample time of $t_s = 0.02$ seconds. A Gaussian distributed probing noise $\eta \sim \mathcal{N}(0, 4)$ with mean zero and variance 4 is used to ensure parameter estimates convergence of the proposed enhanced Q-learning algorithm. The following discrete-time linear drone model is obtained after discretizing the plant,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & t_s & 0 & 0 \\ 0 & 1 & 0 & 0 & t_s & 0 \\ 0 & 0 & 1 & 0 & 0 & t_s \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & -\frac{t_s}{10} & 0 \\ \frac{t_s}{10} & 0 & 0.0004 \\ 0 & -gt_s & 0 \\ gt_s & 0 & 0 \\ 0 & 0 & 0.0411 \end{bmatrix}.$$

The quadratic reward function is designed with the following weight matrices $\mathbf{Q} = \text{diag}\{1, 1, 1, 2, 2, 2\}$ and $\mathbf{R} = 2\mathbf{I}_3$. The control gain $\mathbf{K}$ and the Q-function matrix $\mathbf{S}$ of the model-based optimal control problem are given by

$$\mathbf{K} \approx \begin{bmatrix} 0 & 0.637 & 0 & 0 & 0.970 & 0 \\ -0.637 & 0 & 0 & -0.97 & 0 & 0 \\ 0 & 0 & 0.688 & 0 & 0 & 1.272 \end{bmatrix}$$

$$\mathbf{S} \approx \begin{bmatrix} 77.16 & 0 & 0 & 8.77 & 0 & 0 & 0 & -1.57 & 0 \\ 0 & 77.16 & 0 & 0 & 8.77 & 0 & 1.57 & 0 & 0 \\ 0 & 0 & 93.39 & 0 & 0 & 36.29 & 0 & 0 & 1.45 \\ 8.77 & 0 & 0 & 14.28 & 0 & 0 & 0 & -2.39 & 0 \\ 0 & 8.77 & 0 & 0 & 14.28 & 0 & 2.39 & 0 & 0 \\ 0 & 0 & 36.29 & 0 & 0 & 67.72 & 0 & 0 & 2.68 \\ 0 & 1.57 & 0 & 0 & 2.39 & 0 & 2.47 & 0 & 0 \\ -1.57 & 0 & 0 & -2.39 & 0 & 0 & 0 & 2.47 & 0 \\ 0 & 0 & 1.45 & 0 & 0 & 2.68 & 0 & 0 & 2.11 \end{bmatrix}.$$

An episode is completed after $n(n+m)$ samples are collected to ensure that the DMD algorithms contains enough data samples to compute the state-transition matrices. The sample trajectories are shown in Fig. 3, where the RL policy is capable to stabilize the drone dynamics at the origin.

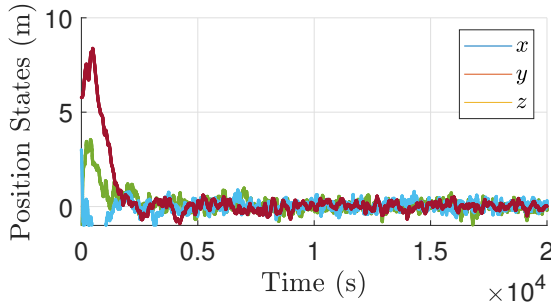


Fig. 3. Evolution of the drone states

The enhanced Q-learning algorithm converges to the following control gain and Q-function matrix values after 100,000 iterations (approximately 1850 episodes)

$$\hat{\mathbf{K}} \approx \begin{bmatrix} 0 & 0.637 & 0 & 0 & 0.97 & 0 \\ -0.637 & 0 & 0 & -0.97 & 0 & 0 \\ 0 & 0 & 0.688 & 0 & 0 & 1.272 \end{bmatrix}$$

$$\hat{\mathbf{S}} \approx \begin{bmatrix} 77.16 & 0 & 0 & 8.77 & 0 & 0 & 0 & -1.57 & 0 \\ 0 & 77.16 & 0 & 0 & 8.77 & 0 & 1.57 & 0 & 0 \\ 0 & 0 & 93.39 & 0 & 0 & 36.29 & 0 & 0 & 1.45 \\ 8.77 & 0 & 0 & 14.28 & 0 & 0 & 0 & -2.39 & 0 \\ 0 & 8.77 & 0 & 0 & 14.28 & 0 & 2.39 & 0 & 0 \\ 0 & 0 & 36.29 & 0 & 0 & 67.72 & 0 & 0 & 2.68 \\ 0 & 1.57 & 0 & 0 & 2.39 & 0 & 2.47 & 0 & 0 \\ -1.57 & 0 & 0 & -2.39 & 0 & 0 & 0 & 2.47 & 0 \\ 0 & 0 & 1.45 & 0 & 0 & 2.68 & 0 & 0 & 2.11 \end{bmatrix}.$$

From the above results we can numerically conclude that the Q-learning part of the proposed approach effectively estimates the control gain and Q-function matrix. To evidence this more precisely, the spectral norm error is used to show the evolution of the control gain and Q-function estimates as the number of iteration increases. The spectral error norm results are shown in Fig. 4.

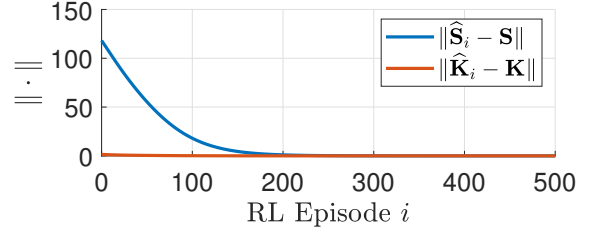


Fig. 4. Convergence of the control gain and Q-function matrix $\mathbf{S}$

We can observe from the above results that the Q-learning algorithm converges in approximately 180 episodes. Here, we compute more episodes to obtain accurate estimates. For the DMD part of the proposed enhanced Q-learning, the following estimates of the state-transition matrices are obtained

$$\hat{\mathbf{A}} \approx \begin{bmatrix} 1 & 0 & 0 & 0.02 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.02 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.02 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \hat{\mathbf{B}} \approx \begin{bmatrix} 0 & -0.002 & 0 \\ 0.002 & 0 & 0 \\ 0 & 0 & 0.0004 \\ 0 & -0.1962 & 0 \\ 0.1962 & 0 & 0 \\ 0 & 0 & 0.0411 \end{bmatrix},$$

which are equivalent to the real state-transition matrix. In contrast to Q-learning, the DMD part finds the state-transition function almost in the first iteration due to the richness of the collected data. This fact is observed in the spectral error norm results in Fig. 5. Notice that the scale of the spectral error norm is quite small and the observed results are associated to numerical issues.

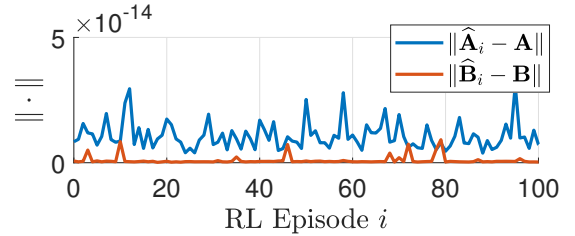


Fig. 5. Convergence of the state-transition matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$

In this particular scenario we obtain satisfactory results such that the counterfactuals explanations can be formulated in the sense of what we do not want to observe, e.g.,

- ✓ “If the probing noise had not been added in the control u_k , then the Q-learning and DMDc algorithms would diverge fast”.
- ✓ “If a flawed reward function had been designed, then the control policy would lack of robustness”.

To verify the above counterfactuals, consider a 1-DOF crane system with the following state-transition function matrices

$$\mathbf{A} = \begin{bmatrix} 0.9961 & 0.02 \\ -0.3919 & 0.9961 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.0016 \\ 0.1641 \end{bmatrix}.$$

Consider flawed reward weights $\mathbf{Q} = \text{diag}\{0.1, 0\}$ and $R = 1000$ that produces the following optimal control gain $\mathbf{K}$ and $\mathbf{S}$ matrix are

$$\mathbf{K} = \begin{bmatrix} -0.0004 & 0.0023 \end{bmatrix}, \quad \mathbf{S} \approx \begin{bmatrix} 256.9 & 0.13 & -0.4 \\ 0.13 & 13.18 & 2.16 \\ -0.4 & 2.16 & 1000 \end{bmatrix}.$$

Notice that the control gain $\mathbf{K}$ is too small, such that the contribution to the states update is minimal such that it will require many steps to stabilize the system. Furthermore, small perturbations can destabilize the closed-loop trajectories since the controller is not capable to compensate them.

The enhanced Q-learning obtains the following results after approximately 1,600 episodes

$$\hat{\mathbf{K}} \approx \begin{bmatrix} -0.0004 & 0.0006 \end{bmatrix}, \quad \hat{\mathbf{S}} \approx \begin{bmatrix} 80.5 & -1.8 & -0.4 \\ -1.8 & 4 & 0.7 \\ -0.4 & 0.7 & 1000 \end{bmatrix}$$

$$\hat{\mathbf{A}} \approx \begin{bmatrix} 0.9961 & 0.02 \\ -0.3919 & 0.9961 \end{bmatrix}, \quad \hat{\mathbf{B}} \approx \begin{bmatrix} 0.0016 \\ 0.1641 \end{bmatrix}.$$

Here, the results indicate that the Q-learning is highly affected by the reward design, whilst the DMDc achieves accurate results. This means that the data quality is good but, due to the flawed design of the reward function requires more episodes to converge or modify the reward function as stated in the previous counterfactual explanations. This is observed in Figs. 6-8.

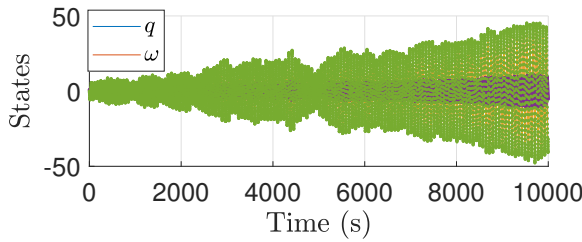


Fig. 6. States evolution of the crane system

The results verify our hypothesis. Fig.6 shows that the low contribution of the control in the states-transition causes a slow convergence time, such that the spectral error norm in Fig. 7 requires more episodes to converge. The main factors that can be used to explain this behaviour are: i) the number of episodes is too short or ii) the reward design is inappropriate. The first factor is the highest level of explanations that we can provide due to the observed behaviour. However, the

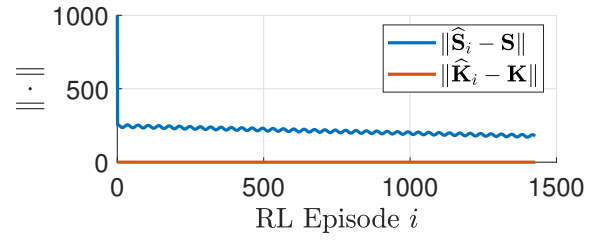


Fig. 7. Spectral norm error of the control gain and Q-function matrix $\hat{\mathbf{S}}$

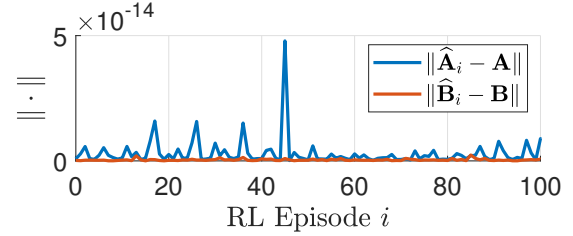


Fig. 8. Convergence of the state-transition matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$

second factor provides more information of why we observe this behaviour and how we can modify it. We change the reward weights to $\mathbf{Q} = \text{diag}\{10, 1\}$ and $R = 1$ whose optimal control gain and $\mathbf{S}$ matrix are

$$\mathbf{K} = \begin{bmatrix} 1.2099 & 1.0817 \end{bmatrix}, \quad \mathbf{S} \approx \begin{bmatrix} 240.04 & 11.25 & 1.47 \\ 11.25 & 9.11 & 1.31 \\ 1.47 & 1.31 & 1.21 \end{bmatrix}.$$

The enhanced Q-learning results are given in Fig. 9-11. The estimates of the control gain $\hat{\mathbf{K}}$ and $\hat{\mathbf{S}}$ converge to the following values,

$$\hat{\mathbf{K}} = \begin{bmatrix} 1.2099 & 1.0817 \end{bmatrix}, \quad \hat{\mathbf{S}} \approx \begin{bmatrix} 240.04 & 11.25 & 1.47 \\ 11.25 & 9.11 & 1.31 \\ 1.47 & 1.31 & 1.21 \end{bmatrix}.$$

The results indicate that the main issue of the previous implementation was the reward design. By modifying the reward function, the control contribution is improved and subsequently the quality of the data. Therefore, the proposed counterfactuals can provide a good notion of what is learning the data-driven model and what we need to modify to achieve the desired behaviour.

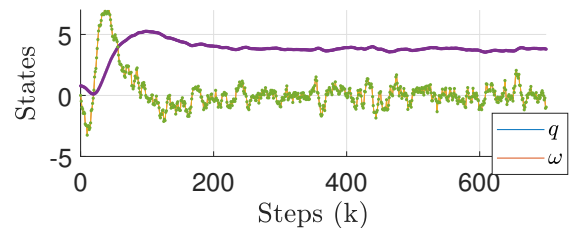


Fig. 9. States evolution of the crane system

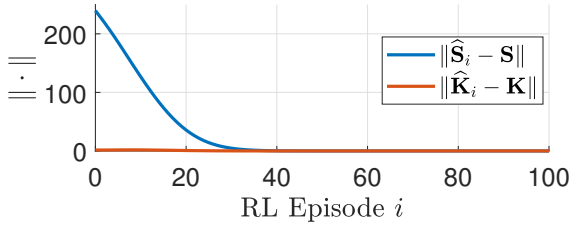


Fig. 10. Spectral norm error of the control gain and Q-function matrix $\hat{S}$

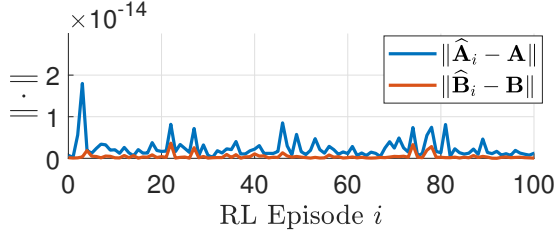


Fig. 11. Convergence of the state-transition matrices $\hat{A}$ and $\hat{B}$

VI. CONCLUSIONS

This paper reports a methodology to explain data-driven control models of autonomous systems. We focus on reinforcement learning algorithms and establish the main learning elements that allows the construction of counterfactual explanations. A enhanced Q-learning algorithm, composed of a traditional Q-learning algorithm and dynamic mode decomposition with control (DMDc) model, is used to uncover each leaning element. Simulation experiments are conducted to show how the proposed model can explain data-driven control outcomes and the mechanisms to modify them into desired ones.

REFERENCES

- [1] B. Fraser, A. Perrusquía, D. Panagiotakopoulos, and W. Guo, "A deep mixture of experts network for drone trajectory intent classification and prediction using non-cooperative radar data," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2023, pp. 1–6.
- [2] G. Singh, A. Perrusquía, and W. Guo, "A two-stages unsupervised/supervised statistical learning approach for drone behaviour prediction," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2023, pp. 1–6.
- [3] M. El Debeiki, S. Al-Rubaye, A. Perrusquía, C. Conrad, and J. A. Flores-Campos, "An advanced path planning and uav relay system: Enhancing connectivity in rural environments," *Future Internet*, vol. 16, no. 3, p. 89, 2024.
- [4] B. Fraser, A. Perrusquía, D. Panagiotakopoulos, and W. Guo, "Hybrid deep neural networks for drone high level intent classification using non-cooperative radar data," in *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. IEEE, 2023, pp. 1–6.
- [5] Y. Bathaee, "The artificial intelligence black box and the failure of intent and causation," *Harv. JL & Tech.*, vol. 31, p. 889, 2017.
- [6] H. Yau, C. Russell, and S. Hadfield, "What did you think would happen? explaining agent behaviour through intended outcomes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 375–18 386, 2020.
- [7] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2660–2673, 2016.

- [8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.
- [9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.
- [10] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.
- [11] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.
- [12] K. KN, D. Ignatyev, A. Tsourdos, and A. Perrusquia, "Advancing fault diagnosis in aircraft landing gear: An innovative two-tier machine learning approach with intelligent sensor data management," in *AIAA SCITECH 2024 Forum*, 2024, p. 0759.
- [13] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.
- [14] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.
- [15] H. Andreas, M. Armgardt, and M. Gunther, "Counterfactuals for causal responsibility in legal contexts," *Artificial intelligence and law*, vol. 31, no. 1, pp. 115–132, 2023.
- [16] J. Ramírez, W. Yu, and A. Perrusquía, "Model-free reinforcement learning from expert demonstrations: a survey," *Artificial Intelligence Review*, vol. 55, pp. 3213–3241, 2022.
- [17] J. Sun, L. Yu, P. Dong, B. Lu, and B. Zhou, "Adversarial inverse reinforcement learning with self-attention dynamics model," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1880–1886, 2021.
- [18] A. Perrusquía, W. Guo, B. Fraser, and Z. Wei, "Uncovering drone intentions using control physics informed machine learning," *Communications Engineering*, vol. 3, no. 1, p. 36, 2024.
- [19] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, 2013.
- [20] A. Perrusquía and W. Yu, "Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview," *Neurocomputing*, vol. 438, pp. 145–154, 2021.
- [21] W. Yu and A. Perrusquia, *Human-Robot Interaction Control Using Reinforcement Learning*. John Wiley & Sons, 2021.
- [22] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. Nathan Kutz, and S. L. Brunton, "Physics-informed dynamic mode decomposition," *Proceedings of the Royal Society A*, vol. 479, no. 2271, 2023.
- [23] A. Perrusquia and W. Guo, "Physics informed trajectory inference of a class of nonlinear systems using a closed-loop output error technique," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
- [24] H. Tang, *Towards Informed Exploration for Deep Reinforcement Learning*. University of California, Berkeley, 2019.
- [25] J. A. Flores-Campos and A. Perrusquía, "Robust control of linear systems: A min-max reinforcement learning formulation," in *2023 20th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. IEEE, 2023, pp. 1–6.
- [26] A. Perrusquia and W. Guo, "Drone's objective inference using policy error inverse reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [27] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, vol. 9, no. 3, pp. 32–50, 2009.
- [28] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2042–2062, 2017.
- [29] A. Perrusquía, "Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning," *Information Sciences*, vol. 595, pp. 364–377, 2022.
- [30] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters*, vol. 100, pp. 14–20, 2017.



DSPACE

<https://dspace.org/>

Swarm decoys deployment for missile deceive using multi-agent reinforcement learning

Bildik, Enver; Tsourdos, Antonios; Perrusquía, Adolfo; Inalhan, Gokhan

2024-06-04

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/ICUAS60882.2024.10556889>

Bildik E, Tsourdos A, Perrusquía A, Inalhan G. (2024), Swarm decoys deployment for missile deceive using multi-agent reinforcement learning. In: 2024 International Conference on Unmanned Aircraft Systems (ICUAS), 4-7 June 2024, Crete, Greece, pp. 256-263