SetPoint Learning : Toward an approach to control them all

Joëlle FERZLY [*1] **Boubker TABITI** [*] **Fabien FURFARO** [*1]

Abstract

Deep learning has enabled significant advancements in the field of dynamic system control, particularly in robotics, video games, navigation, and energy management, especially for non-linear systems. However, this type of learning does not allow for generalization to all dynamic systems, and sometimes, controllers based on system modeling (MPC, PID, MOR) are more relevant than methods based on stochastic data, especially for cases where the system is linearizable, and the target setpoint is well-known, which corresponds to most cases in the electronic, mechanical, and thermal domains. We introduce a new learning method that is generalizable to all linear dynamic systems where the setpoint is known and adaptive. We show that when targeting a setpoint, neural networks adapt towards an objective and would be a promising approach for the use of generative diffusion models for direct interaction with multiple real-world environments in parallel. This study has led to the creation of a set of simulation environments that can serve as a basis for other reinforcement or data-based approaches.

Keywords : Reinforcement Learning, Adaptative control, Gym environment, Time invariant linear systems.

1 Introduction

The planning of decision-making in specific control environments is often a relatively simple problem to solve using data-driven approaches and reinforcement learning. In linearizable environments, planning can be reduced to a trajectory optimization to be achieved and can be done using controllers (https://doi.org/10.1016/S0005-1098(98)00178-2). In reinforcement learning, we observe results that surpass linear or data-based control. We can distinguish the DQN algorithms based on value and PPO based on policy.

The previous methods have worked well in specific situations, but they often require a lot of work to study the system and optimize the parameters for each case [Source]. This lack of ability to work in different situations has led to growing interest in approaches that can create agents that can adapt to different environments [Source]. In reinforcement learning, the ability of an agent to perform well in different environments is a big challenge. Agents can do well in specific training setups, but their performance drops a lot when faced with new situations.

For this, several alternatives have been proposed... [Aiding, Navigation with Human Feedloop, Planning diffusion, Google Genie, JEPA https://arxiv.org/pdf/2301.08243]. However, all of these latter techniques are complex and do not allow generalization to all systems.

In this work, we have developed a simple approach, which makes it possible to generalize the control of a linearizable system, as well as a set of control environments to adapt to any type of algorithm. learning. (...) the main hypothesis is that learning models always have the objective of achieving the best results, whereas here we assume that intermediate results where we aim for a lower score, allows us to expect better results as would a human with a trainer who lowers his level for his student. This approach is thus inspired in a certain way by the human learning approach used in educational psychology.

2 Background and contribution

Our approach is a analogue of past work in behavioral synthesis using trajectory optimization (MPC, Witkin Kass, 1988; Tassa et al., 2012). But we propose here an approach where the variables are always the same whatever the physical system and where the objective to be achieved is defined in the controller input. This approach is also compatible for the use of generative models.

∗ Equal contribution. 1 Capgemini Engineering, Meudon, France. Correspondence to: fabien.furfaro@capgemini.com.

Code and visualizations of the learned process are available at setpoint-learning.github.io.

Our method relies on modifying the states of the simulation environment through the use of a Wrapper or an environment we developed. This enables our approach to interact more dynamically with the system.

3 Problem Setting

Consider a system governed by discrete-time dynamics

$$S_{t+1} = f(S_t, a_t), \tag{1}$$

where S_t is the state of the system at time t, a_t is the control action at time t, and f represents the system dynamics function. For a linear system, the dynamics can be expressed by the following equation:

$$y(t) = s(t) * h(t), \tag{2}$$

where $y(t)$ is the output signal, $s(t)$ is the input action from the controlling system, $h(t)$ represents the impulse response of the system, and $*$ denotes the convolution operation. In this context, the function f in (1) is the convolution operation $f(S_t, a_t) = a_t * h(t)$. The objective is to achieve the optimal control input $u^*(t)$ that minimizes a given cost function $J(S)$:

$$u^*(t) = \arg\min J(S). \tag{3}$$

The main challenge lies in designing a model that can solve problem (1) regardless of the specific physical system.

4 Setpoint Learning method

In this work, we present a setpoint learning approach designed to improve setpoint tracking by incorporating a target guideline. This method mainly relies on selecting a set of primary input variables from initial observations, according to their relevance in understanding system dynamics and achieving predefined targets. Additionally, this approach uses a target setpoint as a reference to guide the learning process. The key variables include:

- a_{t-1} : the action taken in the preceding state,
- s_{t-1} : the state of the system in the previous timestep,
- s_t : the current state of the system
- $(s_{\text{sp}})_{t+1}$: the setpoint, or target state, to be achieved at the next timestep.

By incorporating the previous action a_{t-1}, we introduce a temporal dimension, allowing the models to learn from past actions and adapt to changing dynamics across different contexts. By directing actions towards the convergence of the current state s_t to the target state s_{sp} at the next timestep, this method ensures goal-oriented behavior while maintaining adaptability, as presented in Figure 1.

The setpoint learning the method relies on selecting $a_{(i,t-1)}$ from the action space A, the states $s_{(j,t-1)}$ and $s_{(j,t)}$ from the state space S, and the target setpoint $s_{\text{sp}(j,t+1)}$ to be achieved at the next timestep, resulting in the state space $\tilde{S}_t$ at time t, as describes in the following equation

$$\left\{a_{(i,t-1)}, s_{(j,t-1)}, s_{(j,t)}, s_{\text{sp}(j,t+1)}\right\} \longrightarrow \tilde{S}_t \tag{4}$$

Here, i represents the index of the action a within the set A, and j represents the index of the state s within the set S.

This work aims to demonstrate two key points: (i) a model trained on an environment using the variables defined in equation (4) can control any other linear environment that incorporates the same set of variables; and (ii) this model is capable of tracking any given setpoint.

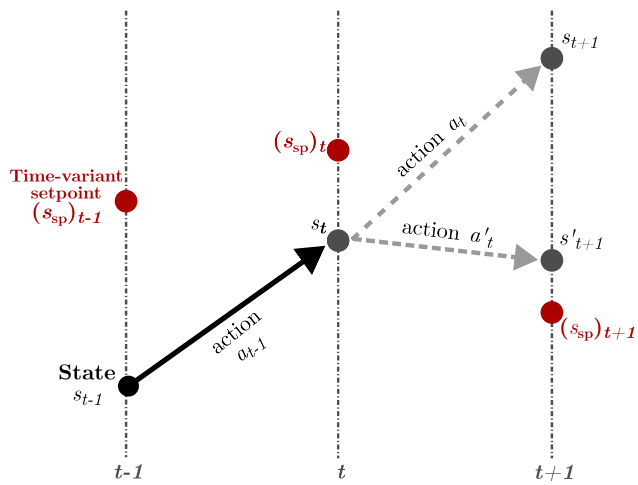

Figure 1. Representative diagram of the target setpoint concept. At time t, the agent in state s_t should choose the action in order to get as close as possible to the target state (setpoint s_{sp}) at time $t+1$.

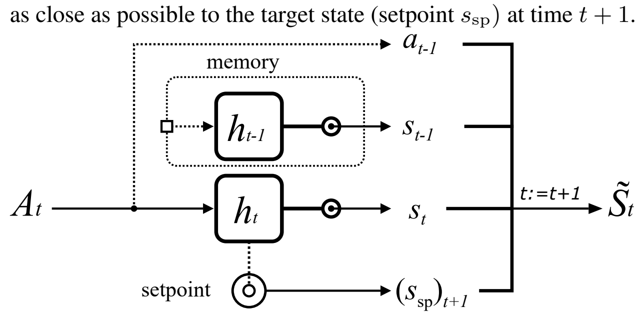

Figure 2. Representative scheme of the setpoint learning environment. Here, A_t represents the action space, h_{t-1} a linearized system or wrapped environment stored in memory and h_t the present system. At $t := t+1$, $\tilde{S}_t$ denotes the resulting state space.

This method is designed to be applied in two distinct frameworks: 1) reinforcement learning, presented in Section

5, where the goal is for the agent to learn to follow a setpoint and adapt to different environments, and 2) diffusion models, detailed in Section 6, where each image represents state-action spaces.

5 Reinforcement learning case

The proposed approach can be employed in a reinforcement learning framework, leading to the Sepoint Reinforcement Learning approach (SPRL). It mainly enables agents to adapt to different setpoints across various environments without the need for specific trainings.

The environment integrating the SPRL approach is presented in Figure 3.

The reward system is redefined based on the deviation from the setpoint to precisely adjust the agent's objective. Thus, the trained agent can choose an action a_t based on the current state s_t, aiming to closely approach the target setpoint s_{sp} at time $t+1$, as illustrated in Figure 1.

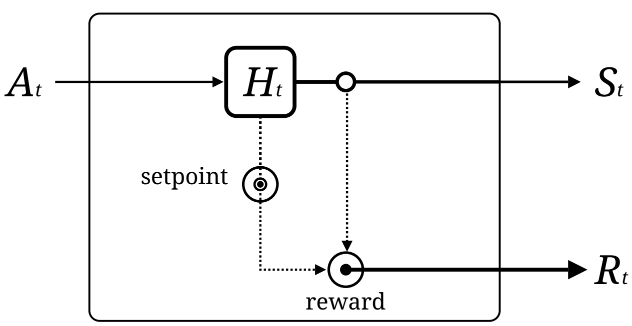

Figure 3. Scheme of the Setpoint Reinforcement Learning environment (SPRL). Here, A_t represents the action space, H_t a linearized system or wrapped environment. At $t := t+1$, R_t denotes the reward received (calculated based on the system's response and the setpoint), and S_t the resulting state space.

As already mentioned,

5.1 Application to the CartPole problem

To validate our approach on a simple model before tackling more complex problems, we first apply it to the inverted pendulum problem (CartPole). This environment is chosen due to its straightforward reward system and its number of independent variables, which are comparable to those in our proposed method.

5.1.1 Cartpole environment

We briefly recall the structure of the Cartpole environment and then present the modified environment. pourquoi modified env

Classical Cartpole environment

The classical CartPole environment as introduced in (Car) provides four observations that allow us to define the set of states S at time t as

$$S_t = \{x_t, v_t, \theta_t, \omega_t\}, \tag{5}$$

where x_t denotes the cart position, v_t the cart velocity, θ_t the pole angle, and ω_t the pole angular velocity. The reward function at time t is defined as follows:

$$R_t(\theta_t) = 1 \text{ if the episode is not terminated and } \theta_t \text{ is in the range } (-0.2095, 0.2095) \text{ rad.} \tag{6}$$

Modified Cartpole environment

In the design of our modified environment, the pole angle θ is selected as the primary variable due to its key role in maintaining equilibrium in the classical CartPole setup. Focusing on the angle provides the agent with relevant information to capture system dynamics and adjust its actions accordingly. The following table summarizes the observations provided by the modified environment.

Variable	Observation	min	max
a_{t-1}	Previous action	0	1
θ_{t-1}	Previous pole angle	-0.2095 rad	0.2095 rad
θ_t	Pole angle	-0.2095 rad	0.2095 rad
$(\theta_{\text{sp}})_{t+1}$	Target pole angle	-0.2095 rad	0.2095 rad

Table 1. Observations provided by the modified CartPole environment.

This leads us to define the set of states $\tilde{S}$ of the modified environment at time t as follows:

$$\tilde{S}_t = \{a_{t-1}, \theta_{t-1}, \theta_t, (\theta_{\text{sp}})_{t+1}\}, \tag{7}$$

where θ_{sp} is the target angle we aim to reach at time $t+1$.

Moreover, the introduction of a reward function based on the setpoint reinforces the understanding of the objective by concentrating high rewards around the setpoint, as shown in Figure 4. This reward function at time t is defined as follows, with $Z = \left(\theta_{\text{sp}} - \frac{\theta_\Omega}{2}, \theta_{\text{sp}} + \frac{\theta_\Omega}{2}\right)$ rad :

$$R_{\theta_{\text{sp}}}(\theta_t) = \begin{cases} 1 & \text{if } \theta_t \text{ is within the range } Z \\ 0.1 & \text{if } \theta_t \text{ is not in } Z \text{ and} \\ & \theta_t \text{ is within } \big(-(\theta_{\text{sp}})_{\max}, (\theta_{\text{sp}})_{\max}\big) \text{ rad.} \end{cases} \tag{8}$$

5.1.2 Experimental Setup

In order to evaluate the adaptability of the SPRL approach, we train agents using the SPRL approach with different target setpoint value θ_{sp} (zero, random constant X, and

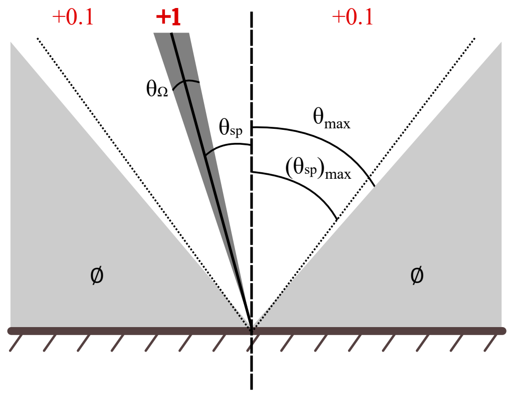

Figure 4. Distribution of the reward (8) in the modified CartPole environment. Around the setpoint, in the range Z (dark gray zone), the reward is equal to 1. Otherwise, the reward is 0.1 within the range $(-(\theta_{\mathrm{sp}})_{\mathrm{max}}, (\theta_{\mathrm{sp}})_{\mathrm{max}})$ rad.

variable random function $f(X)$). We also consider three different scenarios as references trainings, each distinguished by the space state (5) or (7) and the reward function (6) or (8). The configuration of the total six scenarios is resumed in Table 2.

We use the "CartPole-v1" version for agent training as well as for the tests. The constraint on the cart's position is removed, allowing the agent to learn to follow setpoints with more or less off-centered angles. The only constraint remains the pole angle. The episode is terminated when one of the stopping conditions is met, and truncated if the episode ends before the threshold defined by the version of the environment used. For more details, refer to (Car).

The training was realized over 1000 episodes using the Dual Double DQN algorithm, implemented in the RLlib library (RLl). During the training, we save the point at which the moving average of the last 100 scores is maximal, enabling us to compare the performance of the different models as shown in Figure 5.

Training model	Environment	Space state	Reward	Setpoint θ_{sp}	Goal
$\mathrm{T}_{\tilde{S},R_{\theta_0}}$	SPRL	$\tilde{S}$ (7)	$R_{\theta_{\text{sp}}}$ (8)	0	Train SPRL approach with three different setpoints
$\mathrm{T}_{\tilde{S},R_{\theta_X}}$	SPRL	$\tilde{S}$ (7)	$R_{\theta_{\text{sp}}}$ (8)	random variable, constant per episode	
$\mathrm{T}_{\tilde{S},R_{\theta_{f(X)}}}$	SPRL	$\tilde{S}$ (7)	$R_{\theta_{\text{sp}}}$ (8)	random, time-varying function	
$\mathrm{T}_{S,R}$	classical	S (5)	R (6)		Reference training
$\mathrm{T}_{\tilde{S},R}$	SPRL	$\tilde{S}$ (7)	R (6)	0	
$\mathrm{T}_{S,R_{\theta_0}}$	classical	S (5)	$R_{\theta_{\text{sp}}}$ (8)		

Table 2. Overview of the six training models, each defined by the approach used (classical or SPRL), the reward function (R or $R_{\theta_{\text{sp}}}$), and the associated setpoint.

5.1.3 RESULTS

We begin by presenting training results for both the SPRL and classic CartPole environments. Figure 5 illustrates agent performance, displaying rewards trajectories across episodes for each model detailed in Table 2.

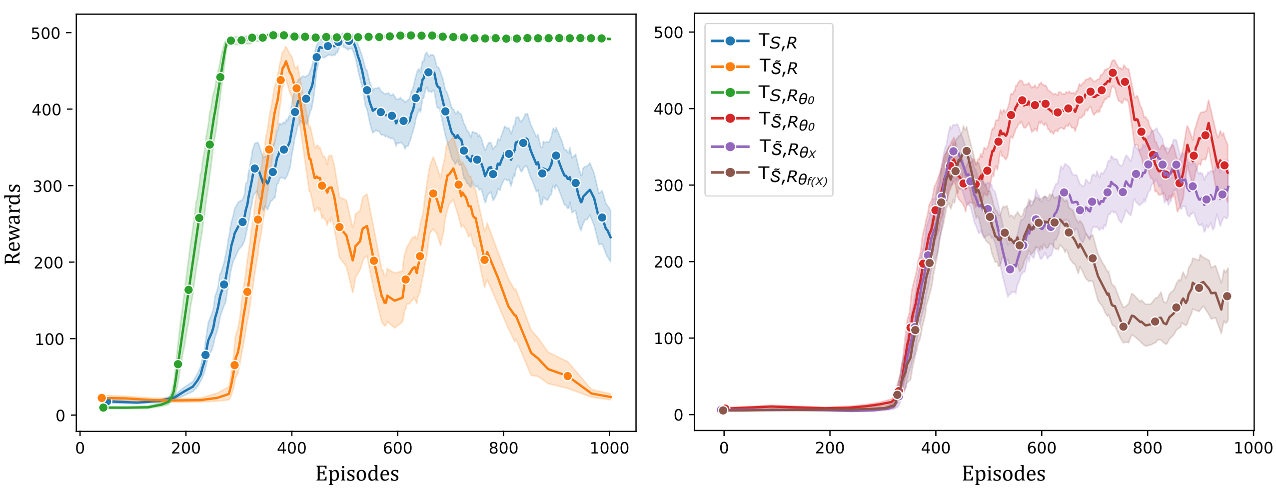

Figure 5. Performance of the six trained agents following the models described in Table 2.

Tests \ Trainings	$\mathrm{T}_{S,R}$	$\mathrm{T}_{S,R_{\theta_0}}$	$\mathrm{T}_{\tilde{S},R}$	$\mathrm{T}_{\tilde{S},R_{\theta_0}}$	$\mathrm{T}_{\tilde{S},R_{\theta_X}}$	$\mathrm{T}_{\tilde{S},R_{\theta_{f(X)}}}$
R_{θ_0}	0.910542	0.994389	0.993675	**0.997996**	0.758116	0.894289
R_{θ_X}	0.151506	0.100000	0.109036	0.100000	0.737048	**0.869880**
$R_{\theta_{f(X)}}$	0.134337	0.100000	0.403614	0.437952	0.682530	**0.859036**

Table 3. Adaptability test results for agents trained in three distinct test environments (null setpoint, random constant per episode, and random time-varying function), expressed by median over 100 iterations.

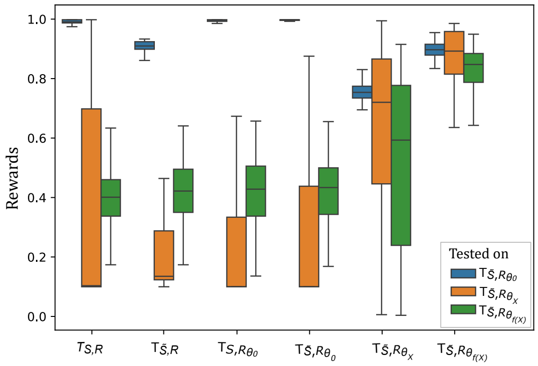

Figure 6. Box plot of the six models tested on the three scenarios respectively integrating a zero setpoint (blue), random constant (orange), and random function (green).

The results of this synthesis, as depicted in table 3, highlight the remarkable adaptability of the $\mathrm{T}_{\tilde{S},R_{\theta_{f(X)}}}$ model. Evaluating $\mathrm{T}_{\tilde{S},R_{\theta_{f(X)}}}$ across three tests shows good performance in following the setpoint, whether constant or varying over time. This performance even surpassed that of the $\mathrm{T}_{\tilde{S},R_{\theta_X}}$ model, specifically trained to follow a constant setpoint X.

5.2 Linear Time-invariant system

In order to improve generalization performance of the approach and evaluate its adaptability, we apply it to linear time-invariant (LTI) systems, which are widely used in engineering and physical sciences.

5.2.1 Developed environment

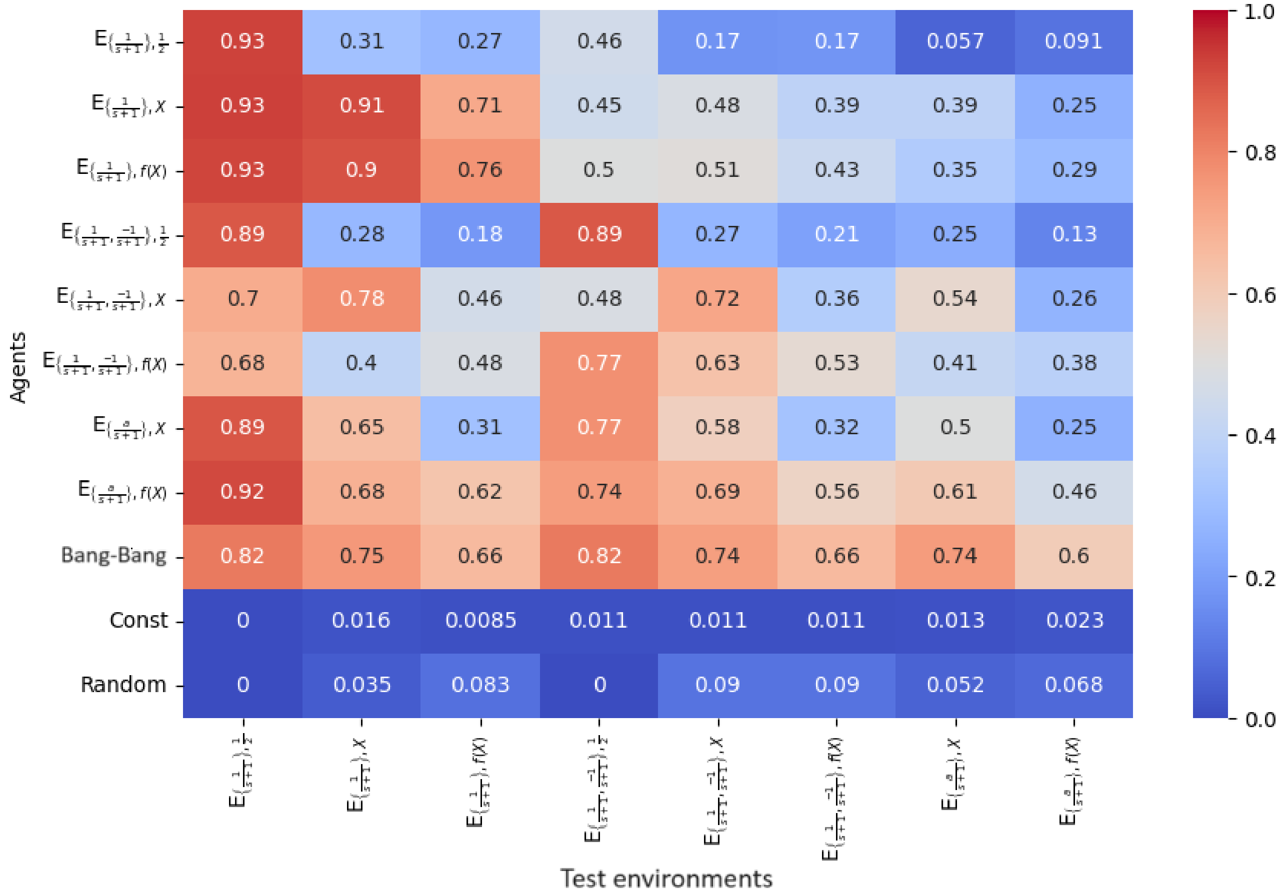

Figure 7. Comparison of agent performance as a function of evaluation environment over 100 episodes, expressed by normalized means.

6 Diffusion learning case

Numerous applications demonstrate that reinforcement learning enables agents to control manageable physical environments (Mnih et al., 2015). However, when agents must control multiple different environments in parallel, reinforcement learning has shown several limitations in terms of generalization (**?**). Currently, one of the approaches that allows generalization across multiple environments is the diffusion method (**?**). To address this, we have developed a parallel environment generator where we test a diffusive approach based on our setpoint method.

6.1 Multiple LTI system data generation

In our multiple LTI environment, we explore four distinct configurations, labeled (a), (b), (c), and (d) (see Figure 8). It's important to note that in this context, A_t yields S_{t+1p}, representing the output state. Configurations (b), (c), and (d) corresponds to the division of configuration (a) into $N \times N$ transfer functions. For that, we use interconnected system techniques in python-control.

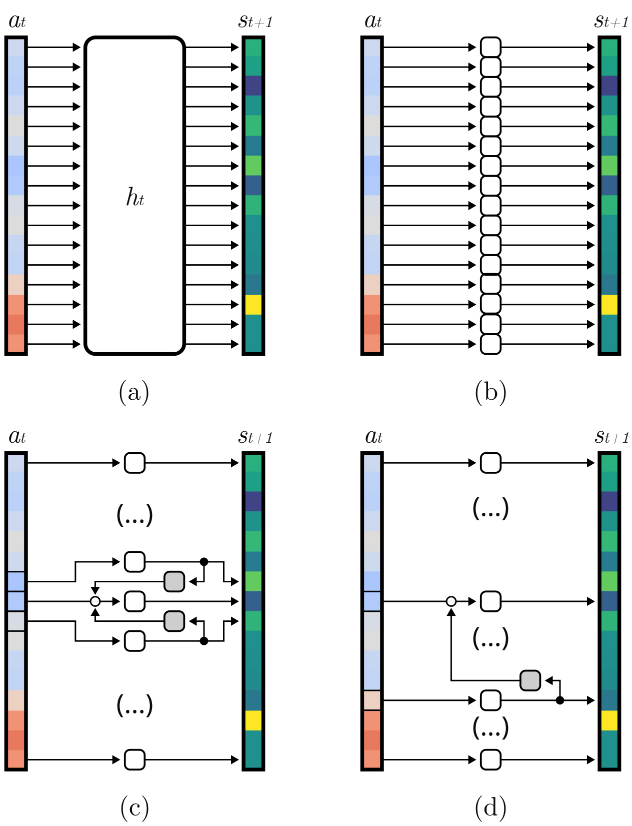

Figure 8. Scheme of multiple LTI system data generation. (a) One transfert function with $N \times N$ input and output, (b), (c), and (d) $N \times N$ transfert function with $N \times N$ input and output. (b) Independant transfert function, (c) 2D local diffusion interconnection, (d) Randomized link beetwen transfert function.

The initial inputs and state variables can also be divided independently of the configurations. This flexibility allows for a more adaptable approach to various environmental complexities. These benefits align with recent trends in reinforcement learning that focus on efficient scaling and generalization across multiple environments(**?**). The ability to independently divide initial inputs and state variables offers:

1. Greater control over the granularity of the input space
2. Flexibility in handling environments of varying complexities
3. Potential for hierarchical learning approaches

To implement our approach with multiple environments, we generated a set of images with dimensions $(1, 5, 32, 32)$ to incorporate into a diffusive model. This constructed database contains, respectivly for 5 channel images:

$$\tilde{S}_{\mathrm{D},t} = \left\{ s_{t-1}, a_{t-1}, s_t, a_t, s_{t+1} \right\} \tag{9}$$

Where s_{t-1}, s_t and s_{t+1} are the system states at times $t-1$, t and $t+1$ respectively, and a_{t-1} represents the previous action before getting the next state s_t. This structure ensures consistency with our reinforcement learning approach while being adaptable for the diffusive approach. The use of image-based representations allows us to leverage the power of convolutional neural networks in processing spatial information, which has shown great success in various learning tasks (Mnih et al., 2015).

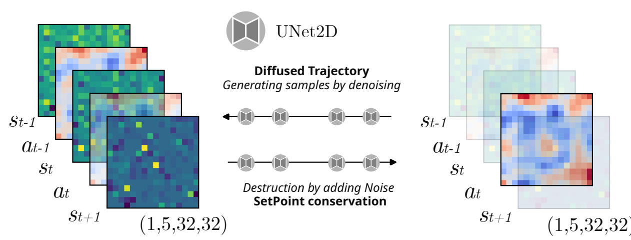

Figure 9. Algorithm scheme of DDPMS-Approach. Je separerais les figures ? pour bien seperarer la partie "génération des environnement multiple" et l'algorithme proposé pour la génération de la base de données...

The integration of diffusion models with reinforcement learning has recently gained attention due to its potential in improving sample efficiency and exploration (**?**). Our approach of using a structured database that includes both state and action information aligns with recent work on offline reinforcement learning with diffusion models (see Figure 9).

6.2 Multiple LTI System Training

In our SPRL (Setpoint Reinforcement Learning) approach, we initially aimed to target the setpoint at time $t+1$. However, to align with the diffusive approach, we slightly modified the state representation to form an image of dimensions $(1, 5, 32, 32)$. In this configuration, and given that the data is pre-generated, each channel of the image can represent a setpoint. To test our approach with an $N \times N$ environment, we employed three distinct configurations:

- **Direct generation from complete noise:** In this configuration, environment configurations are directly generated from complete noise, following the classical diffusion model approach (**?**).
- **Generation with a fixed "target" channel:** Here, environment configurations are generated with a fixed "target" channel, which remains unchanged throughout the generation process. This approach aligns with the concept of conditional generation in diffusion models (**?**).
- **Generation with a noisy "target" channel:** In this final configuration, environment configurations are generated with a noisy "target" channel, meaning this channel is perturbed by noise before the generation process. This method can be seen as a form of data augmentation, potentially improving the robustness of the model (**?**).

These configurations allow us to explore different aspects of the diffusion process in the context of control of multiple LTI systems. By varying the nature of the target channel, we can investigate the model's ability to generate coherent environment configurations under different constraints.

We implemented these methods using a state-of-the-art diffusion model architecture, similar to that proposed by (**?**) with HuggingFace tools, but adapted for our specific control task. The model was trained on a diverse dataset of LTI system trajectories, encompassing a wide range of initial conditions and setpoints. We used a batch size of 64 and trained for 100,000 iterations using the Adam optimizer with a learning rate of 1e-4. The diffusion process consisted of 1000 noise-prediction steps, following the approach outlined in (**?**).

Figure 10 illustrates the training curves for each of the three configurations, providing valuable insights into their respective learning dynamics and convergence properties. Our analysis reveals several key observations:

- **Convergence:** All three training curves demonstrate convergence, indicating the stability of our approach across different configurations.

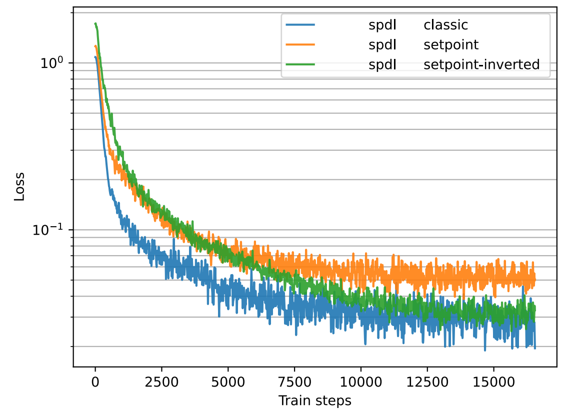

Figure 10. Train curves of DDPMS-Approach. In A...

- **Complete Generation Efficiency:** The complete generation method exhibits the highest efficiency, as evidenced by its lowest loss values. This aligns with findings in recent literature suggesting that unconstrained diffusion models often achieve superior performance (**?**).
- **Channel Generation Performance:** The channel generation training shows convergence comparable to full generation, albeit requiring more training steps. This trade-off between performance and computational cost is consistent with observations in conditional generation tasks (**?**).
- **Partial Channel Generation:** Training with the generation of all channels except one exhibits lower convergence compared to full generation for the same training duration. This phenomenon may be attributed to the increased complexity of learning with partially fixed inputs, a challenge noted in recent work on constrained generative models (**?**).

These results suggest that while complete generation offers the best performance in terms of final loss, there may be practical scenarios where the other configurations provide valuable trade-offs between performance, training time, and specific task requirements. The observed differences in convergence rates and final performance across configurations underscore the importance of carefully selecting the generation strategy based on the specific requirements of the learning task at hand (**?**).

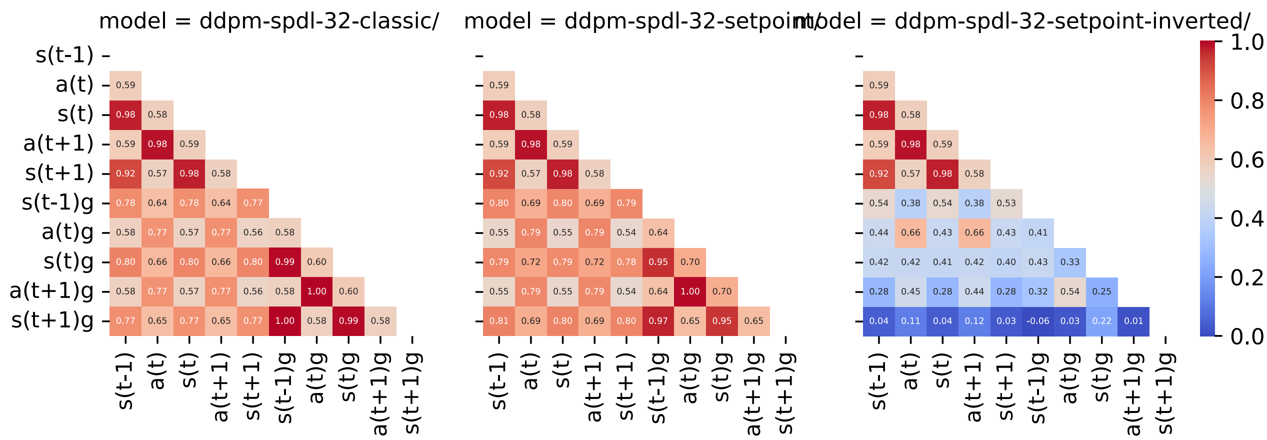

Figure 11. Correlation matrix of DDPMS prediction. In A... METTRE LES VALEUR DES CORRELATION DANS LE TABLEAUX

6.3 Evaluation of Diffusion Methods for LTI Control

To assess the efficacy of our diffusion-based approach in controlling multiple LTI environments, we conducted a comprehensive evaluation of the three generation methods: Complete Generation, Partial Channel Generation, and Pure Setpoint Learning. Our evaluation metrics focused on correlation scores, which measure the alignment between the generated control signals and the observed control trajectories, as well as precision, which quantifies the accuracy of the control actions. Our analysis of the diffusion methods yielded several significant insights:

- **Correlation Scores:** We measured positive correlation scores for each generation method. Notably, Complete Generation and Partial Channel Generation demonstrated the highest scores, indicating their superior performance in controlling multiple LTI environments in parallel across various configurations. This aligns with recent findings on the versatility of diffusion models in handling complex, multi-dimensional tasks (**?**).

- **Precision Limitations:** Despite the positive results, the precision achieved (approximately 80%) is not sufficiently high for direct application in efficient multiple control problems. This level of precision is comparable to bang-bang control, which, while useful in certain contexts, lacks the nuance required for fine-tuned control in complex systems (**?**).

- **Setpoint Learning Challenges:** We observed that pure setpoint learning was ineffective. Generating only one channel resulted in asymmetrical correlation tables, highlighting the importance of considering the full state space in diffusion-based control strategies. This observation is consistent with recent work emphasizing the need for comprehensive state representation in reinforcement learning tasks (**?**).

These results suggest that while setpoint diffusion methods show promise for controlling multiple LTI environments simultaneously, there is room for improvement, particularly in terms of precision. The challenges encountered with pure setpoint learning and single-channel generation underscore the complexity of applying diffusion models to control problems and the need for careful consideration of state representation.

7 Conclusions and perspective

lien vers le code

Utilisation de l'environnement construit dans ce papier pour construire des modele fondation avec plus d'hyperparametre

Utilisation de modèle Transformer pour predire l'action (TD) d'un ensemble d'environnement wrappé

DDPMS Part

This approach resonates with recent work on multi-scale reinforcement learning, where agents learn to operate at different levels of abstraction (**?**). The four configurations we've explored, particularly the divisible approach in (b), represent a novel way to address the challenges of generalization in reinforcement learning across multiple environments. By allowing for flexible division of both transfer functions and input/state variables, our method potentially offers a more robust approach to handling complex, multi-environment scenarios.

Les DDPMS pas tres efficace, utilisation de modèle SSM pour remedier au probleme ? (S4 https://arxiv.org/pdf/2111.00396 https://arxiv.org/abs/2405.21060 / Vision-MamBa) pour predire l'action d'une sequence d'image en limitant les calculs. Limitation : https://arxiv.org/abs/2404.08819 (illusion, repétition et permutation-composition)

Future work could explore the optimal division strategies for different types of environments and the potential for combining this approach with other generalization techniques in reinforcement learning.

Decision Mamba / Transformer https://arxiv.org/pdf/2403.19925

Alternative pour control End-to-End des environnement physique par des LLM plutot que des technique de function calling qui demande beaucoup de puissance de calcul (Eureka). Exemple avec les detections d'objects via Vision-GPT-Yolo https://arxiv.org/abs/2403.12415

References

Gymnasium documentation. URL https://gymnasium.farama.org/environments/classic_control/cart_pole/.

Rllib documentation. URL https://docs.ray.io/en/latest/rllib/index.html.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. doi: https://doi.org/10.1038/nature14236.